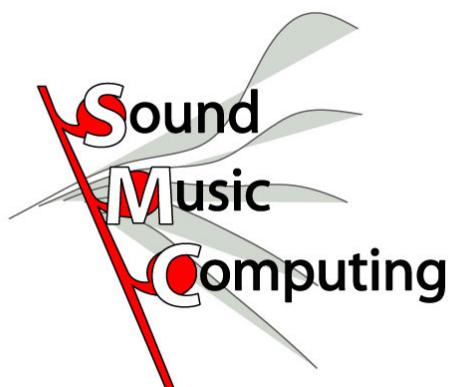


ALGORITHMS FOR



Book in progress

Version: August 28, 2018

Acknowledgments

...

“Path” in Digital Signal Processing:

- Ch. *Fundamentals of digital audio processing*. Fundamentals, representations, transforms, fft
- Ch. *Sound modeling: signal based approaches*. Spectral modeling, cosine transform (sms); 2nd order resonant filters, linear prediction (sub.synthesis and voice); nonlinear filtering and aliasing (nonlinear models)
- Ch. *Sound modeling: source based approaches*. Delays, combs, lowpass filters, fractional filters, lagrance filters, allpass filters
- Ch. *Sound in space*. Allpass combs, lowpass combs, nested allpass (perceptual reverberators); multichannel filters, approximate design of lowpass filters (fdns); LS design of pole-zero transfer functions, filter interpolation (synthetic hrtfs)

Students who contributed to the Octave examples

- Nicola Montecchio (chapter *Sound modeling: source based approaches*)
- Valentina Fabris (perceptual reverberators in *Sound in space*)
- Marco Forlin (FDN reverberators in *Sound in space*)

Contents

Acknowledgments	i
1 Fundamentals of digital audio processing	1
1.1 Introduction	1
1.2 Discrete-time signals and systems	1
1.2.1 Discrete-time signals	1
1.2.1.1 Main definitions	1
1.2.1.2 Basic sequences and operations	3
1.2.1.3 Measures of discrete-time signals	4
1.2.1.4 Random signals	5
1.2.2 Discrete-time systems	5
1.2.2.1 Basic systems and block schemes	5
1.2.2.2 Classes of discrete-time systems	6
1.2.3 Linear Time-Invariant Systems	7
1.2.3.1 Impulse response and convolution	7
1.2.3.2 Properties of LTI systems	8
1.2.3.3 Constant-coefficient difference equations	10
1.3 Signal generators	11
1.3.1 Digital oscillators	11
1.3.1.1 Table lookup oscillator	12
1.3.1.2 Recurrent sinusoidal signal generators	12
1.3.1.3 Control signals and envelope generators	13
1.3.1.4 Frequency controlled oscillators	15
1.3.2 Noise generators	17
1.3.2.1 White noise generators	17
1.3.2.2 Pink noise generators	18
1.4 Spectral analysis of discrete-time signals	19
1.4.1 The discrete-time Fourier transform	19
1.4.1.1 Definition	19
1.4.1.2 DTFT of common sequences	20
1.4.1.3 Properties	21
1.4.2 The sampling problem	22
1.4.2.1 Frequency aliasing	22
1.4.2.2 The sampling theorem and the Nyquist frequency	23
1.5 Short-time Fourier analysis	25
1.5.1 The Discrete Fourier Transform	25

1.5.1.1	Definitions and properties	25
1.5.1.2	Resolution, leakage and zero-padding	26
1.5.1.3	Fast computation of the DFT: the FFT algorithm	28
1.5.1.4	Iterative FFT algorithms and parallel realizations	29
1.5.2	The Short-Time Fourier Transform	30
1.5.2.1	Definition and examples	30
1.5.2.2	Windowing and the uncertainty principle	31
1.6	Digital filters	33
1.6.1	The z -Transform	33
1.6.1.1	Definitions	33
1.6.1.2	z -transforms of common sequences	34
1.6.1.3	Rational z -transforms	35
1.6.2	Transfer function and frequency response of a LTI system	36
1.6.2.1	Definitions	36
1.6.2.2	The concept of filtering	37
1.6.2.3	Stability, causality, and transfer functions	38
1.6.3	Digital filter structures and design approaches	39
1.6.3.1	Block diagram representations	39
1.6.3.2	Filter classification	39
1.7	Commented bibliography	41
2	Sound modeling: signal-based approaches	43
2.1	Introduction	43
2.2	Time-segment based models	44
2.2.1	Wavetable synthesis	44
2.2.1.1	Definitions and applications	44
2.2.1.2	Transformations: pitch shifting, looping	45
2.2.2	Overlap-Add (OLA) methods	47
2.2.2.1	Basic time-domain overlap-add	47
2.2.2.2	Synchronous overlap-add	48
2.2.2.3	Pitch synchronous overlap-add	50
2.2.3	Granular synthesis	52
2.2.3.1	Gaborets	52
2.2.3.2	Sound granulation	53
2.2.3.3	Synthetic grains	53
2.2.3.4	Corpus-based concatenative synthesis	55
2.3	Time-frequency models	56
2.4	Spectral models	57
2.4.1	Sinusoidal model	57
2.4.1.1	Time-varying partials	57
2.4.1.2	Time- and frequency-domain implementations	58
2.4.2	Synthesis by analysis	60
2.4.2.1	Magnitude and Phase Spectra Computation	61
2.4.2.2	A sinusoid tracking procedure	62
2.4.3	“Sines-plus-noise” models	63
2.4.3.1	Stochastic analysis	64
2.4.3.2	Stochastic modeling	65
2.4.3.3	Resynthesis and modifications	66

2.4.4	Sinusoidal description of transients	67
2.4.4.1	The DCT domain	67
2.4.4.2	Transient analysis and modeling	68
2.5	Source-filter models	69
2.5.1	Source signals and filter structures	70
2.5.1.1	Source signals	70
2.5.1.2	Bandlimited digital oscillators	72
2.5.1.3	Resonant filters	73
2.5.1.4	Subtractive synthesis of acoustic sounds	75
2.5.2	Voice modeling	76
2.5.2.1	Voice production mechanism and models	76
2.5.2.2	Formant synthesis	77
2.5.3	Linear prediction	79
2.5.3.1	Linear prediction equations	80
2.5.3.2	Short-time LP analysis	82
2.5.3.3	Linear Predictive Coding (LPC)	84
2.5.3.4	LP based audio effects	86
2.6	Non-linear models	87
2.6.1	Memoryless non-linear processing	87
2.6.1.1	Waveshaping and harmonic distortion	87
2.6.1.2	Aliasing and oversampling	89
2.6.1.3	Clipping, overdrive and distortion effects	90
2.6.1.4	Non-linear systems with memory	92
2.6.2	Multiplicative synthesis	93
2.6.2.1	Ring modulation	94
2.6.2.2	$ \omega_c \pm k\omega_m $ spectra	95
2.6.3	Frequency and phase modulation	95
2.6.3.1	A frequency-modulated sinusoidal oscillator	96
2.6.3.2	Simple modulation	97
2.6.3.3	Other basic FM schemes	98
2.6.3.4	FM synthesis of instrumental sounds	98
2.7	Commented bibliography	99
3	Sound modeling: source-based approaches	103
3.1	Introduction	103
3.2	Physical structures and models	105
3.2.1	Simple vibrating systems and normal modes	105
3.2.1.1	Oscillators	105
3.2.1.2	Impedance	106
3.2.1.3	Coupled oscillators and modal decomposition	108
3.2.2	Continuous vibrating systems and waves	108
3.2.2.1	The one-dimensional D'Alembert equation	108
3.2.2.2	Traveling wave solution	109
3.2.2.3	Waves and modes	110
3.3	Delays and oscillations	111
3.3.1	The Karplus-Strong algorithm	111
3.3.1.1	The comb filter	112
3.3.1.2	Synthesis of plucked strings	113

3.3.2	Fine tuning and fractional delays	116
3.3.2.1	FIR fractional delay filters	116
3.3.2.2	All-pass fractional delay filters	118
3.3.2.3	Time-varying delays	120
3.4	Distributed models: the waveguide approach	120
3.4.1	Basic waveguide structures	120
3.4.1.1	Wave variables and wave impedance	120
3.4.1.2	Delay lines	121
3.4.1.3	Boundary conditions	122
3.4.2	Modeling real world phenomena	123
3.4.2.1	Dissipation	124
3.4.2.2	Loss filter design	125
3.4.2.3	Dispersion	125
3.4.2.4	Dispersion filter design	126
3.4.3	Junctions and networks	127
3.4.3.1	The Kelly-Lochbaum junction	127
3.4.3.2	N-dimensional and loaded junctions	129
3.4.3.3	Non-cylindrical geometries	130
3.5	Lumped models and the modal approach	131
3.5.1	Numerical methods	131
3.5.1.1	Impulse invariant method	132
3.5.1.2	Finite differences and mappings “ <i>s</i> -to- <i>z</i> ”	132
3.5.1.3	Accuracy, stability, computability	133
3.5.1.4	Wave digital filters	135
3.5.2	Modal synthesis	136
3.5.2.1	Normal modes in finite dimensional systems	137
3.5.2.2	Normal modes in PDEs	138
3.5.2.3	Discrete-time mechanical oscillators	139
3.5.2.4	A modal synthesizer	142
3.5.3	Modal analysis	143
3.5.3.1	Simple 1-D shapes	143
3.5.3.2	Simple 2-D shapes	144
3.5.3.3	Experimental estimation	145
3.6	Non-linear physical models	146
3.6.1	Non-linear circuits	147
3.6.1.1	Non-linear capacities	147
3.6.1.2	Vacuum tubes	147
3.6.2	Mechanical interactions	148
3.6.2.1	Impacts	148
3.6.2.2	Stick-slip friction	149
3.6.2.3	Tension modulations	150
3.6.3	Acoustic interactions	151
3.6.3.1	Jets	151
3.6.3.2	Quasi-static reeds	151
3.6.3.3	Dynamic reeds	153
3.6.4	Computability issues	154
3.6.4.1	Non-linear systems and delay-free loops	154
3.6.4.2	Iterative methods	155

3.6.4.3	Sheared non-linearities	155
3.7	Commented bibliography	156
4	Sound in space	161
4.1	Introduction	161
4.2	Reverberation: physical and perceptual background	162
4.2.1	Basics of room acoustics	162
4.2.1.1	Sound waves in a closed space	162
4.2.1.2	Modal density	164
4.2.1.3	Sound sources and room impulse responses	164
4.2.1.4	Reverberation time	166
4.2.1.5	Geometrical room acoustics	168
4.2.2	Perceptual reverberation parameters	169
4.2.2.1	Reverberance	170
4.2.2.2	Early reflections and spatial impression	171
4.2.2.3	Clarity	172
4.2.2.4	Other perceptually relevant parameters	173
4.2.2.5	The Energy Decay Relief	174
4.3	Algorithms for synthetic reverberation: the perceptual approach	175
4.3.1	Late reverberation	175
4.3.1.1	Recirculating delays	176
4.3.1.2	Parameter tuning	176
4.3.1.3	Low-pass combs	178
4.3.1.4	Nested all-pass filters	179
4.3.2	Early reflections	180
4.3.2.1	FIR structures	180
4.3.2.2	Directional effects	182
4.4	Multidimensional reverberation structures	184
4.4.1	Feedback delay networks	184
4.4.1.1	A n-D generalization of the recursive comb filter	184
4.4.1.2	A general FDN reverberators	185
4.4.1.3	Designing the lossless prototype	187
4.4.1.4	Designing lossy components	188
4.4.2	Digital waveguide networks	190
4.4.2.1	The link between FDNs and DWNs	190
4.4.2.2	General lossless scattering matrices	191
4.4.2.3	Waveguide meshes	192
4.5	Spatial hearing	193
4.5.1	The sound field at the eardrum	194
4.5.1.1	Head	194
4.5.1.2	The external ear	196
4.5.1.3	Torso and shoulders	197
4.5.1.4	Head-related transfer functions	198
4.5.2	Perception of sound source location	200
4.5.2.1	Azimuth perception	200
4.5.2.2	Lateralization and externalization	201
4.5.2.3	Elevation perception	202
4.5.2.4	Distance perception	203

4.5.2.5	Dynamic cues	204
4.6	Algorithms for 3-D sound rendering	205
4.6.1	HRTF-based rendering	206
4.6.1.1	Measuring HRTFs and ITDs	206
4.6.1.2	Post-processing of measured HRTFs	207
4.6.1.3	Synthetic HRTFs: pole-zero models	209
4.6.1.4	Synthetic HRTFs: series expansions	210
4.6.1.5	Interpolation	212
4.6.2	Structural models	214
4.6.2.1	Head models	214
4.6.2.2	Modeling torso and pinna reflections	216
4.6.2.3	A complete structural model	217
4.7	Commented bibliography	218
6	Auditory processing	1
6.1	Introduction	1
6.2	Anatomy and physiology of peripheral hearing	2
6.2.1	Sound processing in the middle and inner ear	2
6.2.1.1	The middle ear	2
6.2.1.2	The cochlea and the basilar membrane	3
6.2.1.3	Spectral analysis in the basilar membrane	4
6.2.1.4	Cochlear traveling waves	6
6.2.1.5	The organ of Corti and the haircells	6
6.2.2	Non-linearities in the basilar membrane	8
6.2.2.1	Input-output functions and sensitivity	8
6.2.2.2	Tuning curves and frequency selectivity	9
6.2.2.3	Two-tone interactions	10
6.2.3	Active amplification in the cochlea	11
6.2.3.1	Experimental evidence for cochlear amplification	11
6.2.3.2	Reverse transduction and OHC electromotility	12
6.2.3.3	The cochlear amplifier at work	13
6.3	Elements of psychoacoustics	15
6.3.1	Loudness	15
6.3.1.1	Threshold in quiet	15
6.3.1.2	Equal loudness contours and loudness scales	16
6.3.1.3	Weighting curves	18
6.3.2	Masking	19
6.3.2.1	Simultaneous masking: noise-masking-tone	20
6.3.2.2	Simultaneous masking: tone-masking-noise	22
6.3.2.3	Simultaneous masking: tone-masking-tone	22
6.3.2.4	Temporal masking	23
6.3.3	Auditory filters and critical bands	24
6.3.3.1	The power spectrum model of masking	24
6.3.3.2	Estimating the auditory filter shape	25
6.3.3.3	Barks and ERBs	26
6.3.4	Pitch	28
6.3.4.1	Sinusoids and the mel scale	29
6.3.4.2	Harmonic sounds and pitch illusions	30

6.3.4.3	Inharmonic sounds	31
6.4	Commented bibliography	32
5	From audio to content	225
5.1	Sound Analysis	225
5.1.1	Time domain: Short-time analysis	227
5.1.1.1	Short-Time Average Energy and Magnitude	228
5.1.1.2	Short-Time Average Zero-Crossing Rate	230
5.1.1.3	Short-Time Autocorrelation Function	231
5.1.1.4	Short-Time Average Magnitude Difference Function	233
5.1.2	Audio segment temporal features	234
5.1.2.1	Temporal envelope estimation	234
5.1.2.2	ADSR envelope modelling	234
5.1.2.3	Pitch detection (F_0) by time domain methods	235
5.1.3	Frequency domain analysis	237
5.1.3.1	Energy features	237
5.1.3.2	Spectral shape features	237
5.1.3.3	Harmonic features	239
5.1.3.4	Pitch detection from the spectrum	240
5.1.4	Perceptual features	240
5.2	Spectral Envelope estimation	240
5.2.1	Filterbank	241
5.2.2	Spectral Envelope and Pitch estimation via Cepstrum	241
5.2.3	Analysis via mel-cepstrum	242
5.3	Mid-level features	244
5.3.1	Chromagram	244
5.3.2	Keystrength	244
5.3.3	Tempo	245
5.4	Onset Detection	246
5.4.1	Onset detection in frequency domain	246
5.4.2	Onset detection from Local Energy	247
5.4.3	Combining pitch and local energy information	249
5.5	Feature Selection	251
5.5.1	One-way ANOVA	252
5.5.2	Principal Component Analysis (PCA)	253
5.5.3	Further feature subset selection	254
5.6	Music Information Retrieval	254
5.6.1	Introduction	254
5.6.1.1	Digital Music and Digital Libraries	255
5.6.1.2	Music Information Retrieval	256
5.6.2	Issues of Content-based Music Information Retrieval	257
5.6.2.1	Peculiarities of the Music Language	257
5.6.2.2	The Role of the User	258
5.6.2.3	Formats of Music Documents	259
5.6.2.4	Dissemination of Music Documents	261
5.6.3	Approaches to Music Information Retrieval	262
5.6.3.1	Data-based Music Information Retrieval	262
5.6.3.2	Content-based Music Information Retrieval	263

5.6.3.3	Music Digital Libraries	264
5.6.4	Techniques for Music Information Retrieval	264
5.6.4.1	Terminology	264
5.6.5	Document Indexing	265
5.6.5.1	Query Processing	266
5.6.5.2	Ranking Relevant Documents	267
5.6.5.3	Measures for Performances of MIR Systems	267
5.6.5.4	An Example of Experimental Evaluation	269
5.6.6	Conclusions	272
5.7	Commented bibliography	272
6	Recognizing and communicating expressive information	275
6.1	The quest for expressiveness	275
6.1.1	Affective Computing: the American way to artificial emotions	276
6.1.2	The eastern approach: KANSEI Information Processing	277
6.1.3	Expressiveness in arts and culture	279
6.2	Expressive systems and interfaces	279
6.2.1	Affective channel	280
6.2.2	Affective interfaces	281
6.2.2.1	Properties of affective signals.	283
6.3	Mid-level representations of expressive information	284
6.3.1	Approaches to conceptualizing emotions	285
6.3.1.1	Valence-Arousal space	286
6.3.2	Laban Theory of Movement	287
6.3.3	Sensory expressive intention representation	289
6.3.3.1	Musical expression	289
6.3.3.2	Kinetics-Energy space	290
6.3.4	An action based metaphor	292
6.3.4.1	Similarities in the feature space	293
6.3.4.2	Similarities in the perceptual space.	295
6.3.4.3	Toward an action based interpretation of expressive intentions	296
6.4	Recognition of expression	300
6.4.1	Recognition of expression in music performance	301
6.4.1.1	Relevant features for expression recognition	302
6.4.1.2	From features to expressive intentions	303
6.4.2	Recognition of affect in speech	304
6.4.3	Expressive gestures	305
6.4.3.1	Recognition of expression in gestures	307
6.4.3.2	Local motion feature detection	307
6.4.3.3	Event motion feature detection	310
6.4.3.4	Mid-level motion features	312
6.4.4	Faces and emotional states	313
6.4.4.1	Targets and gestures associated with emotional expression	314
6.4.4.2	Facial expression recognition	316

7	Music information processing	319
7.1	Elements of music theory and notation	319
7.1.1	Pitch	319
7.1.1.1	Pitch classes, octaves and frequency	320
7.1.1.2	Musical scale.	321
7.1.1.3	Musical staff	322
7.1.2	Note duration	323
7.1.3	Tempo	325
7.1.4	Rhythm	325
7.1.5	Dynamics	326
7.1.6	Harmony	326
7.2	Organization of musical events	327
7.2.1	Musical form	327
7.2.1.1	Low level musical structure	327
7.2.1.2	Mid and high level musical structure	327
7.2.1.3	Basic patterns	327
7.2.1.4	Basic musical forms	328
7.2.2	Cognitive processing of music information	328
7.2.3	Auditory grouping	331
7.2.4	Gestalt perception	333
7.3	Basic algorithms for melody processing	338
7.3.1	Melody	338
7.3.1.1	Melody representation: melodic contour	338
7.3.1.2	Similarity measures	339
7.3.1.3	Edit distance	340
7.3.2	Melody segmentation	340
7.3.2.1	Gestalt based segmentation	341
7.3.2.2	Local Boundary Detection Model (LBDM)	342
7.3.3	Tonality: Key finding	344
7.3.3.1	Key finding algorithm	346
7.3.3.2	Modulation	348
7.4	Algorithms for music composition	349
7.4.1	Algorithmic Composition	349
7.4.2	Computer Assisted Composition	350
7.4.3	Categories of algorithmic processes	351
7.4.3.1	Mathematical models	351
7.4.3.2	Knowledge based systems	352
7.4.3.3	Grammars	353
7.4.3.4	Evolutionary methods	353
7.4.3.5	Systems which learn	354
7.4.3.6	Hybrid systems	355
7.4.4	Discussion	355
7.4.5	Emerging Trends	356
7.5	Markov Models and Hidden Markov Models	357
7.5.1	Markov Models or Markov chains	357
7.5.2	Hidden Markov Models	359
7.5.3	Markov Models Applied to Music	360
7.5.3.1	HMM models for music search: MuseArt	361

7.5.3.2	Markov sequence generator	363
7.5.4	Algorithms	364
7.5.4.1	Forward algorithm	364
7.5.4.2	Viterbi algorithm	366
7.6	Appendix	367
7.6.1	Generative Theory of Tonal Music of Lerdahl and Jackendorf	367
7.6.2	Narmour's implication realization model	371
7.7	Commented bibliography	373
8	Standards for audio and music representation	375
8.1	Digital audio compression	375
8.1.1	Bit Rate Reduction	375
8.1.2	Auditory Masking and Perceptual Coding	377
8.1.2.1	Auditory Masking	377
8.1.2.2	Perceptual Coding	380
8.1.3	MPEG 1 Layer-3 coding	382
8.1.3.1	The Psychoacoustic Model	382
8.1.3.2	The Problem of Stereo Coding	384
8.1.3.3	Discussion on MPEG Layer-3	385
8.2	MIDI representation of music	388
8.2.1	MIDI Messages	388
8.2.1.1	Channel messages	390
8.2.1.2	System Messages	392
8.2.2	MIDI files	393
8.2.2.1	Standard MIDI Files	393
8.2.2.2	General MIDI	395
8.2.2.3	MIDI Timing	396
8.2.3	Discussion on MIDI representation	397
8.2.3.1	MIDI limitations	397
8.2.3.2	Operation on music	397
8.3	MusicXML: a music interchange file format	398
8.3.1	MusicXML structure	399
8.3.2	MusicXML: a simple example	400
8.4	Object description: MPEG-4	404
8.4.1	Scope and features of the MPEG-4 standard	404
8.4.2	The utility of objects	405
8.4.3	Coded representation of media objects	406
8.4.3.1	Composition of media objects	406
8.4.3.2	Description and synchronization of streaming data for media objects .	411
8.4.4	MPEG-4 visual objects	411
8.4.5	MPEG-4 audio	412
8.4.5.1	Natural audio	412
8.4.5.2	Synthesized audio	413
8.4.5.3	Sound spatialization	417
8.4.5.4	Audio BIFS	418
8.5	Multimedia Content Description: Mpeg-7	419
8.5.1	Introduction	419
8.5.1.1	Context of MPEG-7	420

8.5.1.2	MPEG-7 objectives	420
8.5.2	MPEG-7 terminology	423
8.5.3	Scope of the Standard	423
8.5.4	MPEG-7 Applications Areas	426
8.5.4.1	Making audio-visual material as searchable as text	428
8.5.4.2	Supporting push and pull information acquisition methods	429
8.5.4.3	Enabling nontraditional control of information	430
8.5.5	Mpeg-7 description tools	431
8.5.6	MPEG-7 Audio	433
8.5.6.1	MPEG-7 Audio Description Framework	433
8.5.6.2	High-level audio description tools (Ds and DSs)	436
9	Multimodal interaction	439
9.1	Research paradigms on sound and sense	440
9.1.1	From music philosophy to music science	440
9.1.2	The cognitive approach	442
9.1.2.1	Psychoacoustics	442
9.1.2.2	Gestalt psychology	442
9.1.2.3	Information theory	442
9.1.2.4	Symbol-based modelling of cognition	443
9.1.2.5	Subsymbol-based modelling of cognition	443
9.1.3	Beyond cognition	443
9.1.3.1	Embodied music cognition	443
9.1.3.2	Music and emotions	444
9.1.3.3	Gesture modelling	444
9.1.3.4	Physical modelling	444
9.1.3.5	Motor theory of perception	444
9.1.4	Embodiment and mediation technology	445
9.1.4.1	An object-centered approach to sound and sense	445
9.1.4.2	A subject-centered approach to sound and sense	446
9.1.5	Music as innovator	447
9.2	Enaction, Arts and Creativity	449
9.3	Some core questions about creativity: a philosophical and linguistic point of view	451
9.3.1	Creativity: eight basic questions	451
9.4	Auditory displays and sound design	455
9.4.1	Warnings, Alerts and Audio Feedback	455
9.4.2	Earcons	457
9.4.3	Auditory Icons	457
9.4.4	Mapping	458
9.5	Sonification	459
9.5.1	Information Sound Spaces (ISS)	459
9.5.2	Interactive Sonification	461
9.6	Interactive sounds	461
9.6.1	Ecological acoustics	462
9.6.1.1	The ecological approach to perception	462
9.6.2	Everyday sounds and the acoustic array	464
9.7	Multimodal perception and interaction	467
9.7.1	Combining and integrating auditory information	467

9.7.2	Perception is action	468
9.8	Multimodal and Cross-Modal Approaches to Control of Interactive Systems	469
9.8.1	Introduction	469
9.8.2	Multisensory Integrated Expressive Environments	470
9.8.3	Cross-modal expressiveness	471
9.8.4	A conceptual framework	472
9.8.4.1	Syntactic layer	473
9.8.4.2	Semantic layer	474
9.8.4.3	Connecting syntax and semantics: Maps and spaces	475
9.8.5	Methodologies of gesture analysis	475
9.8.5.1	Bottom-up approach	475
9.8.5.2	Subtractive approach	476
9.8.6	Examples of multimodal and cross-modal analysis	476
9.8.6.1	Analysis of human full-body movement	476
9.8.7	Examples of Multisensory Integrated Expressive Environments	477
9.8.8	Perspectives	480
10	Musical cultural heritage: From preservation to restoration	483
10.1	Introduction	483
10.2	Audio Documents Preservation	485
10.2.1	“Two Legitimate Directions”	486
10.2.2	“To Save History, Not Rewrite It”	486
10.2.3	“Secondary Information”: the History of the Audio Document Transmission	487
10.2.4	The Audio Preservation Protocol	487
10.3	Passive Preservation	488
10.3.1	Mechanical Carriers	489
10.3.2	Magnetic Tape	489
10.4	Active Preservation	491
10.4.1	Carrier Analysis and Restorative Actions	491
10.4.2	Re-recording	492
10.4.3	Preservation Copy	492
10.4.3.1	Format for the Audio Files	494
10.4.3.2	Video Shooting and Photographic Documents	494
10.4.3.3	Audio Fingerprinting	496
10.4.3.4	Descriptive card	497
10.5	Automatic Metadata Extraction	497
10.5.1	Reel to Reel Magnetic Tape	497
10.5.2	Warped Phonographic Discs	498
10.5.3	Off-centered Phonographic Disc	500
10.5.4	Representing Metadata	501
10.6	Audio Data Extraction and Alignment from Phonographic Disc	502
10.6.1	Photos of GHOSTS (PoG)	502
10.7	Audio restoration	504
10.7.1	CREAK: A de-noise and de-click system dedicated to shellac discs	505
10.7.1.1	Bootstrap procedure	505
10.7.1.2	Forward/backward filtering	506
10.7.2	CMSR: A de-noise algorithm dedicated to wax and Amberol cylinders and shellac discs	506

10.7.3	PAR: A de-hiss perceptual algorithm dedicated to reel-to-reel tapes and cassettes	507
10.7.4	Experimental results	508
10.7.4.1	Comparison	509
10.7.5	Assessment	513
10.8	Concluding Remarks	514
10.9	Commented bibliography	516

List of Figures

1.1	(a) Analog, (b) quantized-analog, (c) discrete-time, and (d) numerical signals.	2
1.2	Block schemes of discrete-time systems; (a) a generic system $\mathcal{T}\{\cdot\}$; (b) ideal delay system \mathcal{T}_{n_0} ; (c) moving average system \mathcal{T}_{MA} . Note the symbols used to represent sums of signals and multiplication by a constant (these symbols will be formally introduced in Sec. 1.6.3).	6
1.3	Properties of LTI system connections, and equivalent systems; (a) cascade, and (b) parallel connections.	9
1.4	Block schemes for constant-coefficient difference equations representing (a) the accumulator system, and (b) the moving average system.	11
1.5	Controlling a digital oscillator; (a) symbol of the digital controlled in amplitude and frequency; (b) example of an amplitude control signal generated with an ADSR envelope.	14
1.6	Amplitude (a) and frequency (b) control signals	17
1.7	Example of frequency aliasing occurring for three sinusoids.	22
1.8	Examples of sampling a continuous time signal: (a) spectrum limited to the base band; (b) the same spectrum shifted by 2π ; (c) spectrum larger than the base band.	24
1.9	Examples of DFT applied to complex exponential sequences: (a) $N = 64, k_0 = 20$, the sequence is periodic and the DFT is a delta-sequence in the frequency domain; (b) $N = 64, k_0 = 20.5$, the sequence is not periodic and the DFT exhibits leakage; (c) $N = 128, k_0 = 41$, the sequence is the windowed exponential of Eq. (1.77) and the DFT is a shifted version of the rectangular window DFT.	27
1.10	Examples of DFT ($N = 128$) applied to a rectangular window: (a) $M = N/16$, (b) $M = N/8$, (a) $M = N/2$	28
1.11	(a) Tree of calls in an invocation of RECURSIVE-FFT: the leaves contain a bit-reversal permutation of $x[n]$; (b) parallel realization of ITERATIVE-FFT, where butterfly operations involve bit-reversed elements of $x[n]$	31
1.12	Short-time spectral analysis of a chirp signal: (a) initial portion of the chirp signal, with $\omega_0 = 2\pi \cdot 800 \text{ rad/s}^2$; (b) spectrogram obtained with $F_s = 8 \text{ kHz}$; (c) spectrogram obtained with $F_s = 4 \text{ kHz}$	32
1.13	Comparison of different windows; (a) time-domain window sequences, symmetrical with respect to $n = 0$; (b) window spectra in the vicinity of $k = 0$. Differences in terms of main lobe width and relative side lobe level can be appreciated.	33
1.14	Pole-zero plots and ROCs of some simple transforms: (a) finite-length exponential sequence; (b) right-sided exponential sequence with $ a > 1$; (c) left-sided exponential sequence with $ a > 1$; (d) exponentially damped, right-sided sinusoidal sequence. In all plots the dotted circle is the unit circle and the gray-shaded region is the ROC of the corresponding transform.	36

1.15	Comparing phase delay and group delay: (a) evaluation of phase delay and group delay for a generic non-linear phase response; (b) illustration of phase delay and group delay for a narrowband signal.	39
1.16	Block diagram representation of filters: (a) conventional pictorial symbols for delay, adder, and multiplier; (b) a general filter structure.	40
1.17	Classification of filters into basic categories, depending on their magnitude response $ H(\omega_d) $: (a) low-pass and high-pass filter; (b) band-pass and band-reject filter; (c) resonator and notch filter.	41
2.1	An example of OLA signal reconstruction, with triangular windowing.	48
2.2	The generic SOLA algorithmic step.	48
2.3	The generic PSOLA algorithmic step.	51
2.4	Representation of granular synthesis where grains derived from different sources are randomly mixed.	53
2.5	Example of a synthetic grain waveform, with frequency $\omega_k = 2\pi \cdot 500$ rad/s and standard deviation $\sigma = 0.2$.	54
2.6	Sum of sinusoidal oscillators with time-varying amplitudes and frequencies.	59
2.7	Beating effect: (a) frequency envelopes (f_1 dashed line, f_2 solid line) and (b) envelope of the resulting signal.	60
2.8	Fourier analysis of a saxophone tone: (a) frequency envelopes and (b) amplitude envelopes of the sinusoidal partials, as functions of time.	61
2.9	Block diagram of the sinusoid tracking process, where $s[n]$ is the analyzed sound signal and a_k , f_k are the estimated amplitude and frequency of the k th partial in the current analysis frame.	62
2.10	Block diagram of the stochastic analysis and modeling process, where $s[n]$ is the analyzed sound signal and a_k , f_k , ϕ_k are the estimated amplitude, frequency, and phase of the k th partial in the current analysis frame.	64
2.11	Example of residual magnitude spectrum (solid line) and its line-segment approximation (dashed line), in an analysis frame. The analyzed sound signal is the same saxophone tone used in figure 2.8.	65
2.12	Block diagram of the sines-plus-noise synthesis process.	66
2.13	Example of DCT mapping: (a) an impulsive transient (an exponentially decaying sinusoid) and (b) its DCT as a slowly varying sinusoid.	68
2.14	Block diagram of the transient analysis and modeling process, where $s[n]$ is the analyzed sound signal and a_k , f_k , ϕ_k are the estimated amplitude, frequency, and phase of the k th DCT-transformed transient in the current analysis frame.	68
2.15	Source-filter model.	70
2.16	Spectrally rich waveforms: (a) time-domain square, triangular, sawtooth, and impulse train waveforms; (b) corresponding spectra (first 20 partials).	71
2.17	Bandlimited synthesis of the square, triangular, and sawtooth waves, using 3, 8, and 13 sinusoidal components.	72
2.18	Example of a second-order resonator tuned on the center frequency $\omega_c = 2\pi 440/F_s$ and with bandwidth $B = 2\pi 100/F_s$; (a) impulse response; (b) magnitude response.	74
2.19	Parallel structure of digital resonators for the simulation of struck objects – the R_i 's have transfer functions of the form (2.28).	76
2.20	A schematic view of the phonatory system. Solid arrows indicates the direction of the airflow generated by lung pressure.	77
2.21	A general model for formant synthesis of speech.	78

2.22	Formant synthesis of voice: (a) spectra of two pulse trains with fundamental frequencies at 150 Hz and 250 Hz; (b) first three formants of the vowel /a/; (c) spectra of the two output signals obtained by filtering the pulse trains through a parallel combination of the three formants.	80
2.23	LP analysis: (a) the inverse filter $A(z)$, and (b) the prediction error $e[n]$ interpreted as the unknown input $gx[n]$	81
2.24	Example of LP analysis/synthesis, with prediction order $p = 50$; (a) target signal $s[n]$ (dotted line) and unit variance residual $x[n]$ (solid line); (b) magnitude spectra $ S(f) $ (thin line) and $ gH_{LP}(f) $ (thick line).	83
2.25	General scheme of a simple LP based codec.	84
2.26	Example of LP spectra for increasing prediction orders p (the target signal is a frame of voiced speech). For the sake of clarity each spectrum is plotted with a different offset.	85
2.27	Block scheme of a LP-based implementation of cross-synthesis (also known as vocoder effect) between two input sounds s_1 and s_2	86
2.28	Sound synthesis by non-linear distortion (or waveshaping).	88
2.29	Example of output signals from a linear and from a non-linear system, in response to a sinusoidal input; (a) in a linear system the input and output differ in amplitude and phase only; (b) in a non-linear system they have different spectra.	88
2.30	Example of quadratic distortion; (a) spectrum of a sinusoid $x[n]$ and (b) spectrum of the squared sinusoid $x^2[n]$	89
2.31	Two implementations of a memoryless non-linear system; (a) non-linear processing inserted between oversampling and downsampling; (b) non-linear processing on band-limited versions of the input.	90
2.32	Simulation of overdrive and distortion; (a) soft overdrive and exponential distortion (with $q = 6$); (b) asymmetric clipping (with $q = -0.2$ and $d = 8$).	92
2.33	Realization of a non-linear system with memory using the Volterra series (truncated at the order N).	93
2.34	Ring modulation with a sinusoidal carrier.	94
2.35	Simple modulation: (a) block scheme; (b) the first 10 Bessel functions; (c) spectra produced by simple modulation with $\omega_c = 2\pi 700$ Hz, $f_m = 2\pi 100$ Hz, and I varying from 0.5 to 3 (for the sake of clarity each spectrum is plotted with a different offset).	97
2.36	Basic FM schemes; (a) compound modulation, (b) nested modulation, and (c) feedback modulation.	99
3.1	Second order electrical, mechanical, and acoustic oscillators; (a) a RLC circuit; (b) a mass-spring-damper system; (c) a Helmholtz resonator.	106
3.2	Illustration of (a) cylindrical and (b) spherical coordinates.	109
3.3	Boundary conditions and wave reflections; (a) fixed string end and negative wave reflection; (b) free string end and positive wave reflection.	110
3.4	A comb filter; (a) block scheme and (b) magnitude response.	112
3.5	Spectrogram of a plucked A2 guitar string. Note the harmonic structure and the decay rates, which increases with increasing frequency.	114
3.6	Low-pass comb filter obtained through insertion of a low-pass element into the comb structure; (a) block scheme and (b) frequency response (the triangles mark the harmonic series $l\pi/L$, $l \in \mathbb{N}$).	115
3.7	Linear interpolation filters ($N = 1$) for $\tau_{ph} = 0, 0.1, \dots, 1$; (a) amplitude response and (b) phase delay.	117

3.8	First-order Thiran allpass filters for $\tau_{\text{ph}} = 0, 0.1, \dots, 1$; (a) phase response and (b) phase delay.	119
3.9	Lossless waveguide sections with observation points at position $x = 0$ and $x = mX_s = L$; (a) cylindrical section; (b) conical section.	122
3.10	Ideal waveguide terminations: (a) positive reflection; (b) negative reflection.	123
3.11	Waveguide simulation dissipation and dispersion phenomena through insertion of loss and dispersion filters.	124
3.12	Kelly-Lochbaum junction for two cylindrical bores with different areas.	127
3.13	Example of use of the Kelly-Lochbaum junction: (a) a parallel junction of two cylindrical bores; (b) realization with two waveguide sections and a Kelly-Lochbaum junction.	129
3.14	Example of a loaded junction: a waveguide structure for a string excited by an external force signal $f_J[n]$ (e.g. a hammer).	130
3.15	Boundary regions for (a) non-convex and (b) convex conical junctions.	131
3.16	Mapping of the vertical axis $s = j\omega$ (solid circle lines) and of the left-half s -plane (shaded regions) using the backward Euler method g_1 and the bilinear transform g_2	134
3.17	A linear discrete-time system; (a) delay-free path, (b) equivalent realization with no delay-free paths.	135
3.18	136
3.19	Analogies between continuous and discrete systems: (a) approximation of an ideal string with a mass-spring network; (b) modes of the discrete system for different numbers N of masses.	140
3.20	Amplitude responses of a second order oscillator with constant mass and quality factor, and $\omega_0 = 2, 4, 8$ kHz: continuous-time responses (solid lines) and discrete-time responses (dashed lines) with (a) impulse invariant method, (b) backward Euler method, (c) backward Euler method with centered scheme, (d) bilinear transform.	141
3.21	Modal description of the ideal bar: (a) ideal bar with various boundary conditions and (b) corresponding modes.	143
3.22	Modal description of ideal membranes: (a) ideal rectangular membrane with fixed ends and (b) corresponding modes; (c) ideal circular membrane with fixed ends and (d) corresponding modes.	145
3.23	Exciter-resonator interaction scheme for a musical instrument.	146
3.24	Non-linear behavior of (a) capacitance $C(v)$ and (b) charge $q(v)$ in the Chua-Felderhoff circuit.	148
3.25	The non-linear impact model (3.106): (a) phase portrait of a point mass hitting a hard surface; (b) the corresponding non-linear force during impact.	149
3.26	Stick-slip friction: (a) example of parametrization of a kinetic (static) friction curve; (b) Helmholtz motion resulting from stick-slip ideal string-bow interaction.	150
3.27	Length dL of a string at point x over the segment dx	151
3.28	Schematic representation of the reed-mouthpiece system.	152
3.29	Quasi-static approximation of a single reed; (a) u versus Δp and (b) rotated mapping p^+ versus p^-	153
3.30	Shear transformation of $f(x) = e^{-x^2}$ for various k values.	156
4.1	Plane wave loops $(1, 1, 0)$ and $(3, 2, 0)$, as seen on the (x, y) plane.	163
4.2	Estimation of modal density; (a) distribution of wavenumbers on a regular point lattice, (b) estimation of the amount of wavenumbers contained in a spherical octant of radius k	164

4.3	Room Impulse response and reverberation time: (a) RIR of a very reverberant environment (a cathedral); (b) a portion of the same RIR in dB, together with its EDC and a linear fit.	166
4.4	Acoustic rays from a source to a receiver (a) in a vertical room section and (b) in a horizontal room section. Solid lines represent the direct sound, dashed lines represent first-order reflections, dotted lines represent second-order reflections.	168
4.5	Schematical room response to an ideal impulse: the time axis is relative to the direct sound, which reaches the receiver at $t = 0$	169
4.6	Estimation of temporal reflection density through the image source method; (a) construction of two first-order and two second-order reflections, and (b) estimation of acoustic rays reaching a receiver within the time interval $(t, t + dt)$	170
4.7	Waterfall representation for the RIR of Fig. 4.3.	171
4.8	Energy Decay Relief for the RIR of Fig. 4.3, normalized at 0 dB and truncated at -60 dB.	175
4.9	Block scheme of a reverberator based on comb filters (the H_i blocks) and <i>all-pass comb</i> filters (the A_i blocks). The internal structure of the A_i filters is shown in the grey box.	176
4.10	A reverberator constructed with a series connection of all-pass filters and a low-pass filter in feedback.	179
4.11	Nested all-pass filters; (a) generalization of an all-pass structure (see Fig. 4.9), and (b) realization by means of a lattice structure.	180
4.12	Two realizations of a reverberator with early reflections; (a) late reverberation block receiving the delayed input signal, and (b) late reverberation block receiving the output of the early reverberation FIR filter, with additional control parameters D_1 , D_2 , g . The late reverberation block can be one of the structures examined in the previous sections.	182
4.13	Two stuctures that associate directional filters to early reflections, for binaural reverberation; (a) one directional filter for each reflection, and (b) two directional filters for two sets of reflections.	183
4.14	184
4.15	A Feedback Delay Network structure for artificial reverberation.	185
4.16	Lossless prototype network associated to the Feedback Delay Network of Fig. 4.15.	187
4.17	DWN reverberator	190
4.18	2D rectilinear digital waveguide mesh.	193
4.19	Estimate of ITD in the case of a distance sound source (plane waves) and spherical head.	194
4.20	Magnitude response $ H_{\text{sphere}}(\infty, \theta_{inc}), \mu $ of a sphere for an infinitely distant source.	195
4.21	External ear: (a) pinna, and (b) ear canal.	196
4.22	Effects of pinna: (a) direction-dependent reflections, and (b) resonances.	197
4.23	Effects of torso: (a) reflections, and (b) shadowing.	198
4.24	Spherical coordinate systems used in the definition of HRTFs: (a) vertical-polar coordinate system, and (b) interaural-polar coordinate system.	199
4.25	Example of magnitude of HRTFs (a) in the xy plane ($\theta \in [-\pi/2, \pi/2]$, $\phi = 0$) and (b) in the yz plane ($\theta = 0$, $\phi \in [-\pi/4, \pi]$). Interaural polar coordinates are used.	200
4.26	Time differences at the ears; (a) non ambiguous ITD, (b) ambiguous ITD, and (c) IED.	201
4.27	Cone of confusion.	202
4.28	Block scheme of a headphone 3-D audio rendering system based on HRTFs.	206
4.29	Example of principal component analysis: a two-dimensional data set with 0 mean, and the two basis vectors (principal axes) extracted using PCA.	210
4.30	Approaches to HRTF interpolation; (a) bilinear interpolation of the HRIRs, and (b) interpolation of zeros for pole-zero synthetic HRTFs	213
4.31	Block scheme of a headphone 3-D audio rendering system based on a structural model.	214

4.32	Spherical head model; (a) ideal response of Eq. (4.57) for $\rho \rightarrow +\infty$, (b) approximated response with the first-order filter of Eq. (4.75) with $\alpha_{min} = 0.1$ and $\theta_{min} = 170^\circ$	215
4.33	A schematic representation of the major features of the HRIR in the median plane ($\theta = 0$) for a human subject. White and black lines indicate ridges and throughs in the response, respectively.	216
4.34	A simple yet complete structural model.	217
6.1	Schematic, not-in-scale, drawing of the human peripheral auditory system.	2
6.2	Middle ear function; (a) scheme of the mechanical action , and (b) qualitative magnitude response (note that this plot does not report real measured data, it is an illustrative example in qualitative agreement with real data).	3
6.3	Linearized structure of the cochlea.	4
6.4	Qualitative responses to a 1 kHz sinusoidal stimulus at various sites on the basilar membrane.	5
6.5	Left: cross-section of the cochlea. Right: close-up on the organ of Corti and haircells.	7
6.6	Cochlear measurements: (a) example of input-output curve of the basilar membrane at CF; (b) examples of tuning curves of basilar membrane and haircells. Note that these plots do not report real measured data, they are just illustrative examples in qualitative agreement with real data.	9
6.7	Schematic representation of the positive feedback that causes cochlear amplification. The OHCs are involved in the loop while IHCs have no role in the amplification and are passive motion detectors	13
6.8	Loudness formation; (a) threshold in quiet as a function of frequency, measured with the method of Békésy-tracking; (b) equal loudness contours illustrating the variation in loudness with frequency (each curve represents one loudness level).	16
6.9	A-weighted dB scale: (a) inverse curve of the 40 phon equal loudness contour, and (b) magnitude response of the filter $H_{dBA}(s)$, digitized with the bilinear transform.	18
6.10	Masking threshold curves of a sinusoidal probe signal as a function of its frequency; (a) masking caused by white noise with density levels L_{mask} ; (b) masking caused by 1.1 kHz low-pass filtered white noise and 0.9 kHz high-pass filtered white noise with density levels L_{mask} . Here and in the following figures the dashed curve indicates threshold of hearing in quiet (see Sec. 6.3.1).	20
6.11	Masking threshold curves of a sinusoidal probe as a function of its frequency; (a) masking caused by narrow-band noise with $L_{mask} = 60$ dB and three different center frequencies f_0 ; (b) masking caused by narrow-band noise with $f_0 = 1$ kHz and five different levels L_{mask}	21
6.12	Masking threshold curves of a sinusoidal probe as a function of its frequency; (a) masking caused by a sinusoid at 1 kHz and 80 dB, with regions where beats and difference tone are audible; (b) The masked thresholds are given for sound pressure levels of 40 and 60 dB of each partial.	22
6.13	. Regions of premasking, simultaneous masking, and postmasking. Two different time scales are used: time relative to masker onset and time relative to masker cessation.	23
6.14	Qualitative psychophysical tuning curves for six different probe signals (probe frequencies and levels are indicated by circles), as a function of masker frequency.	25
6.15	Auditory filter estimation through a notched-noise experiment; (a) magnitude responses of sinusoidal probe, notched-noise masker, and auditory filter; (b) measured masking threshold as a function of notch half-bandwidth Δf	26

6.16 (a) Critical bandwidths, Eq. (6.17), and equivalent rectangular bandwidths, Eq. (6.19), as functions of center frequency; (b) the critical-band rate scale, Eq. (6.18), that maps Hz into Barks.	27
6.17 Constructing the mel scale: (a) relation between frequency f_1 of reference sinusoid and frequency f_2 of a comparison sinusoid producing half pitch sensation (solid curve), and absolute “ratio pitch” sensation as a function of frequency (the dashed line is the line $f_2 = 1.5 \cdot f_1$); (b) absolute “ratio pitch” sensation as a function of frequency in linear scales.	29
6.18 Qualitative plot of the existence region for virtual pitch.	31
 5.1 Scheme for supervised classification.	225
5.2 Features extraction process	227
5.3 Popular windows and their applications	228
5.4 <i>Short-Time Average Energy</i> and <i>Short-Time Average Magnitude</i>	229
5.5 Windowing affecting <i>Short-Time Average Energy</i>	230
5.6 Zero-Crossing Rate of the word /SONO/.	231
5.7 Autocorrelation of a voiced sound.	232
5.8 Short time AMDF of the voiced sound of fig. 5.7.	234
5.9 Frame of the phoneme /OH/ and its Short-Time Autocorrelation Function. The position of the second maximum at k_M indicates the pitch period.	236
5.10 AMDF of the frame of the phoneme /OH/ of fig. 5.9. The position of the second minimum at k_m indicates the pitch period.	236
5.11 Cepstrum example	242
5.12 Automatically formant estimation from cepstrally smoothed log Spectra [from Schaefer Rabiner].	243
5.13 (a) Transformation from Hz to mel. (b) Mel-scale filterbank.	244
5.14 Example of mel-cesptrum analysis of a clarinet tone: tone spectrum (high left); spectral envelope reconstructed with first 6 mel cepstral coefficients (low right), spectral envelope rebuilt from LPC analysis (low left); spectral envelope estimated with all mel cepstrum coefficients (low right).	245
5.15 Chromagram in the pitch range G2 - C8.	245
5.16 Wrapped chromagram.	246
5.17 Chromagram of a framed signal.	246
5.18 The Chromagram is compared with the profiles related to the different tonality candidate.	247
5.19 Keystrength computed on a musical excerpt.	247
5.20 Autocorrelation of the onset detection curve.	248
5.21 Tempo estimation of a framed audio signal.	248
5.22 Example of onset detector based on local energy: time-domain audio signal (a), 40ms windowed RMS smoothing with 75% overlap (b), peaks in slope of envelope (c).	249
5.23 caption index	250
5.24 envelope of a 5 second window of signal. The blue line represents the <i>RMS</i> temporal variation, while that the red line represents the dynamic threshold that follows the <i>RMS</i> of the signal. The crosses on signal <i>RMS</i> represent the detected onsets. The circles represent the zones of effective onsets.	251
5.25 The points P1 and P2 are two values of the <i>RMS</i> signal detected by threshold. These values define a valley between two peaks. The value for the onset is the minimum that can be find between these two points.	252

5.26	Detection of onsets for a five seconds window: the last figure represents the onset detection, where red lines indicate the instants for each detected onset.	253
5.27	Architecture of a music information retrieval system	256
5.28	Example of a melody	260
5.29	The phases of a methodology for MIR: Indexing, retrieval, and data fusion	268
6.1	Affective interface.	281
6.2	A sigmoid function is applied to the inputs to an emotional system (from Picard 1997).	284
6.3	Representation levels of expressive information.	284
6.4	Expressive content recognition.	285
6.5	Dimensional representation of emotions: the circumplex model of Russel also called Valence-Arousal space. The horizontal axis represents the Valence dimension, while the vertical axis represents the Arousal dimension.	287
6.6	Effort space with four components Space, Time, Weight, and Flow. Each effort component is measured on a bipolar scale, the extreme values of which represent opposite qualities along each axis.	288
6.7	Basic efforts: (left) components; (right) relationship between basic efforts.	289
6.8	Method to understand the the expressive content.	290
6.9	Position of the evaluation adjectives in the semantic space. Evaluation adjectives: black (nero), oppressive (greve), serious (grave), dismal (tetro), massive (massiccio), rigid (rigido), mellow (soffice), tender (tenero), sweet (dolce), limpid (limpido), airy (aereo), gentle (lieve), effervescent (spumeggiante), vaporous (vaporoso), fresh (fresco), abrupt (brusco), sharp (netto)	291
6.10	Kinetics-Energy space, as mid-level representation of expressive intentions.	292
6.11	Screen shot of the PerformanceWorm, showing an expression trajectory with horizontal axis tempo in beats per minute and vertical axis dynamics (loudness) in decibel. The darkest point represents the current instant, while instants further in the past appear fainter.	292
6.12	The Valence-Arousal space (left) and the Kinematics-Energy space (right), respectively, and placement of expressive intentions used in our experiments.	294
6.13	Feature space obtained by Principal Component Analysis of the features of expressive violin performances, according to adjectives from both affective and sensorial spaces (Fig. 6.12).	294
6.14	Perceptual space resulting from the listening experiment.	295
6.15	Behaviour of the basic linear mechanical systems: friction, inertia, elasticity.	297
6.16	Correspondence analysis on experiment with expressive performances. Dashed lines represent the outcome of the cluster analysis.	299
6.17	Correspondence analysis on experiment with Bigand musical excerpt. Dashed and continuous lines represent the outcome of the cluster analysis (with number of groups equal to 3 and 6, respectively).	299
6.18	The process of extracting expression.	301
6.19	Sequence of profiles for roughness (left) and attack time (right) according to suggested adjectives performed by three instruments ("Twinkle Twinkle Little Star").	302
6.20	Principal Component Analysis on the whole recorded audio from three instruments, according to adjectives from both affective and sensorial spaces.	303
6.21	Examples of where different type of musical gestures (sound producing, sound facilitationg and communicative) may be found in piano performance [from Leman-Godoy (2010)].	306
6.22	Silhouette extraction.	307

6.23	The Lucas - Kanade (LK) algorithm is used to track features in the input image.	308
6.24	Example of skin colour tracking to extract positions and trajectories of hands and head. .	308
6.25	(a) An example of SMI with time window of four frames. (b) Measure of internal motion in SMIs.	309
6.26	Computation of Contraction Index (CI).	309
6.27	Motion segmentation.	310
6.28	Temporal profile of motion features.	311
6.29	Directness index as a measure of how a trajectory is direct or flexible.	312
6.30	Mean values of the QoM (left) and CI (right) computed for each motion phase (the four graphs refer to four performances by the same dancer, each one expressing a different basic emotion: anger-solid line; fear- dashed line; joy-dash-dot line; grief-dotted line). The X-axis is the index of the motion phase in which the movement has been segmented (therefore, X is not the time axis).	312
6.31	Facial Definition Parameters (a) and Facial Animation Parameter Units (b) in MPEG-4: IRIS Diameter (by definition it is equal to the distance between upper and lower eyelid) in neutral face, Eye Separation, Eye - Nose Separation, Mouth - Nose Separation, Mouth - Width Separation	315
6.32	Archetypal face expressions in Mpeg-4: Sadness, Anger, Joy Fear, Disgust, Surprise. .	316
6.33	Face expression at its apex (Neutral, Happiness, Surprise, Anger and Disgust.[from Essa and Pentland, PAMI 97]	317
6.34	Eight sample face images from the CMU dataset of 20 persons showing neutral, angry, happy, and sad facial expressions.	317
7.1	One octave in a piano keyboard.	320
7.2	Example of a sharp (a) and a flat (b) note. Example of a key signature (c): D major. . .	321
7.3	Staff and note names.	322
7.4	Correspondence of keys and notes on the staff.	323
7.5	Symbols for note length.	324
7.6	Symbols used to indicate rests of different length.	324
7.7	Tie example: crotchet (quarter note) tied to a quaver (eighth note) is equivalent to the dotted crotchet (dotted quarter note).	324
7.8	(a) Example of a time signature: 3/4 indicates three quarter note beats per measure. (b) Example of a metronome marking: 120 quarters to the minute.	325
7.9	Dynamics notation indicating music starting moderately loud (<i>mezzo forte</i>), then becom- ing gradually louder (<i>crescendo</i>) and then gradually quieter (<i>diminuendo</i>).	326
7.10	Schema illustrating the various aspects of musical information processing [from McAdams 1996].	329
7.11	Auditory organization	331
7.12	Auditory scene analysis	331
7.13	Score of Frère Jacques.	332
7.14	(a) Pattern where successive notes are separated by large pitch jumps but alternate notes are close together in pitch, is probably heard as two separate and simultaneous melodies. (b) Excerpt from the Courante of Bach's First Cello Suite: two concurrent pitch patterns are heard.	333
7.15	Experiments of Proximity and Good Continuation	334
7.16	Experiments of Closure and Common Fate	335
7.17	Example of proximity gestalt rule	335
7.18	Example of similarity gestalt grouping principle.	336

7.19 Example of similarity gestalt grouping principle.	336
7.20 Examples of good continuation gestalt grouping principle.	337
7.21 Example of closure.	337
7.22 Rubin vase: example of figure/ground principle.	337
7.23 Horses by M. Escher. An artistic example of figure and ground interchange.	338
7.24 Melodic contour and Parson code.	339
7.25 Dynamic programming algorithm for computing EditDistance.	340
7.26 Beginning of Frère Jacques. Higher-level grouping principles override some of the local detail grouping boundaries (note that LBDM gives local values at the boundaries suggested by parallelism - without taking in account articulation.	343
7.27 Piano keyboard representation of the scales of C major and C minor. Notes in each scale are shaded. The relative importance of the first (tonic - C), fifth (dominant - G) and third (mediant - E) degrees of the scale is illustrated by the length of the vertical bars. The other notes of the scale are more or less equally important followed by the chromatic notes that are not in the scale (unshaded) [from McAdams 1996].	344
7.28 C Major and C minor profiles derived with the probe-tone technique from fittingness ratings by musician listeners.	345
7.29 Comparison between tonal hierarchies and statistical distribution of tones in tonal works. It is shown the frequency of occurrence of each of the 12 chromatic scale tones in various songs and other vocal works by Schubert, Mendelssohn, Schumann, Mozart, Richard Strauss and J. A. Hasse. and the key profile (scaled).	346
7.30 Example of Krumhansl-Schmuckler key finding algorithm: opening bar of <i>Yankee Doodle</i>	347
7.31 Example of Krumhansl-Schmuckler key finding algorithm: duration distribution of <i>Yankee Doodle</i>	347
7.32 State transition of the weather Markov model (from Rabiner 1999).	357
7.33 Block diagram of an isolated word recognizer (from Rabiner 1999).	360
7.34 A sung query (from Shifrin 2002)	361
7.35 Markov model for a scalar passage (from Shifrin 2002)	362
7.36 Markov model for Alouette fragment (from Shifrin 2002)	363
7.37 "Hymn tunes" generated by computer from an analysis of the probabilities of notes occurring in various hymns. From Brooks, Hopkins, Neumann, Wright. "An experiment in musical composition." IRE Transactions on Electronic Computers, Vol. 6, No. 1 (1957).	364
7.38 (a) Illustration of the sequence of operations required for the computation of the forward variable $\alpha_{t+1}(i)$. (b) Implementation of the computation of $\alpha_{t+1}(i)$ in terms of a lattice of observation t and states i	366
7.39 Example of Viterbi search.	367
7.40 Main components of Lerdahl and Jackendoff's generative theory of tonal music.	368
7.41 Example of a time-span tree for the beginning of the All of me ballad [from Arcos 1997].	369
7.42 Example of GTTM analysis of the first four bars of the second movement of Mozart's K.311: Metrical analysis (dots below the piece) and Time-Span analysis (tree-structure above the piece) [from Cross 1998].	369
7.43 Example of GTTM analysis of the first four bars of the second movement of Mozart's K.311: Prolongational analysis [from Cross 1998].	370
7.44 Top: Eight of the basic structures of the I/R model. Bottom: First measures of All of Me, annotated with I/R structures.	371
7.45 Example of Narmour analysis of the first four bars of the second movement of Mozart's K.311 [from Cross 1998].	373

8.1	Threshold in quiet and masking threshold. Acoustical events in the shaded areas will not be audible.	377
8.2	Masking threshold and signal-to-mask ratio (SMR). Acoustical events in the shaded areas will not be audible.	378
8.3	The absolute threshold of hearing in quiet. Across the audio spectrum, it quantifies the SPL required at each frequency such that an average listener will detect a pure tone stimulus in a noiseless environment. From (Painter-Spanias 2000).	378
8.4	When many simultaneous maskers, each has its own masking threshold, and a global masking threshold can be computed.	379
8.5	Temporal masking properties of the human ear. Backward (pre) masking occurs prior to masker onset and lasts only a few milliseconds; forward (post) masking may persist for more than 100 ms after masker removal.	379
8.6	Net effect of simultaneous and temporal masking. Acoustical events under the surface will not be audible.	380
8.7	Block diagram of a generic perceptual coder.	381
8.8	Block diagram of a generic perceptual decoder.	382
8.9	MPEG/Audio filter bandwidths versus critical bandwidths.	384
8.10	Block diagram of the MPEG 1 Layer-3 encoder.	384
8.11	Pre-Echo Example: (a) Uncoded Castanets. (b) Transform Coded Castanets, 2048-Point Block Size.	387
8.12	Changing the window length to match the signal properties of the input helps in avoiding pre-echoes. A long window is used to maximize coding gain and achieve good channel separation during segments identified as stationary, and a short window is used to localize time-domain artifacts when pre-echoes are likely.	387
8.13	Example of a connection of MIDI devices.	389
8.14	Midi messages.	389
8.15	Organization of MIDI messages.	390
8.16	Piano roll representation of Midi data.	398
8.17	Partwise (a) and timewise (b) score organization.	399
8.18	The partwise organozation of a score in MusicXML. Each part listed serially. A part consists of measures and the measures contain notes and attributes.	400
8.19	Example of note element.	400
8.20	A one-measure piece of music that contains a whole note on middle C, based in 4/4 time.	401
8.21	Mpeg-4 high level system architecture.	406
8.22	An example of an MPEG-4 scene.	407
8.23	Objects in a scene (a) and the corresponding BIrary Format for Scene (BIFS) representation (b).	409
8.24	Major components of an MPEG-4 terminal (receiver side).	410
8.25	Major components of an MPEG-4 terminal (receiver side).	413
8.26	In a traditional audio coder, the source model and perception model are defined outside of the transmission (for example, in a standards document). The codec designers do the best job they can at designing these models, but then they are fixed for all content (a). In Structured Audio, the source model is part of the content. It is transmitted in the bitstream and used to give different semantics to the signal representation for each piece of content. There can be a different source model, or multiple source models, for each different piece of content (b).	415

8.27	In SAOL, a metaphor of bus routing is employed that allows the concise description of complex networks. The output of the <code>tone</code> instrument is placed on the bus called <code>echo_bus</code> ; this bus is sent to the instrument called <code>echo</code> for further processing.	416
8.28	Two sound streams are processed and mixed using the AudioBIFS scene graph. A musical background is transmitted with the MPEG-4 Structured Audio system and reverb is added with an AudioFX node. Then the result is mixed with a speech sound transmitted with MPEG-4 CELP.	419
8.29	Scope of MPEG-7.	424
8.30	MPEG-7 main elements.	425
8.31	Abstract representation of possible applications using MPEG-7.	426
8.32	Abstract representation of possible relation between Descriptors and Description Schemes.426	
8.33	Abstract representation of possible applications using MPEG-7.	427
8.34	Overview of the MPEG-7 Multimedia DSs.	432
8.35	AudioSpectrumEnvelope description of a pop song. The required data storage is NM values where N is the number of spectrum bins and M is the number of time points.	436
8.36	A 10-basis component reconstruction showing most of the detail of the original spectrogram including guitar, bass guitar, hi-hat and organ notes. The left vectors are an AudioSpectrumBasis Descriptor and the top vectors are the corresponding AudioSpectrumProjection Descriptor. The required data storage is $10(M + N)$ values.	437
9.1	left: 8 steps in hue, saturation and lightness; right: The TBP prototype of an Information-Sound Space (ISS).	460
9.2	A map of everyday sounds. Complexity increases towards the center. Figure based on (Gaver, 1993).	466
9.3	The layered conceptual framework distinguishes between syntax and semantics, and in between, a connection layer that consists of affect/emotion expressiveness spaces and mappings.	471
9.4	The layered conceptual framework makes a distinction between syntax and semantics, and in between, a connection layer that consists of affect / emotion / expressiveness (AEE) spaces and mappings.	472
9.5	Taxonomy of musical syntactical cues.	473
9.6	The EyesWeb application for <i>Allegoria dell'opinione verbale</i> by R. Doati.	478
9.7	Connection between physical and spatial movements.	479
9.8	Trombone during the premiere performance of Adriano Guarnieri's Medea.	480
10.1	The schema of the most significant positions of the debate evolved since the Seventies inside the archivist community on the audio documents active conservation.	488
10.2	Representation of the A/D transfer protocol	493

10.3 (a) a sound postcard: it looked like a standard postcard on the back, but on the front an analogue recording was engraved in a thin layer of laminate. Sound postcards were usually made by small firms, and the recording quality was extremely low; in this case the importance of storing the picture in with the preservation copy is particularly evident. (b) displays a label of His Master's Voice disc: DK 119 (on the label, right) is the catalogue number; 2-054042 (on the label, left, and at the top of the mirror) is a second catalogue number (as its minor typographic importance, probably it is the first issue catalogue number: therefore here we have a reprint); A12804 (in the mirror, down) is the matrix number. It is possible to decode this information: DK = 30 cm diameter; Yellow label = "International Celebrítá" series, printed in Hayes; 2-054 prefix in catalogue number corresponds to a second issue (2), 30 cm diameter (0), Italian catalogue (5) and duet or trio as sound content (4); by means of a comparison between matrix number and published repertories we can deduce the recording date (17th, January, 1913). (c) and (d) show two typical corruptions in a tape and in a disc respectively: this information should be stored with the preservation copy also, in order to have a deep insight the artifacts of the audio signal.	495
10.4 Frame of a video recording of an open reel tape: the circle drawn in black marks a specific sound event. Often, in the electro-acoustic music field (in the works for tape and acoustic musical instruments) the marks on the tape are used as a synchronization means between live-electronics performer and the recorded tape music. If this information was not preserved, it would not be possible to perform the piece.	496
10.5 (a) and (b) show source frames from the video of a winding tape, while (c) and (d) show the corresponding processed images.	498
10.6 Automatic discontinuities extraction from a winding tape (splices, marks).	499
10.7 Processed frames from a video of a oscillating record player's arm. (a) Photo of the turntable arm; (b) Lowest position of the arm in an oscillation, (c) its highest position. (b) and (c) show Lucas-Kanade features detected on the arm's head and tracked through the oscillation. (d) shows the differences between lowest and highest positions.	500
10.8 Temporal evolution of the y coordinate of a Lucas-Kanade feature located on the arm's head. It can be seen clearly how the oscillations indicate a deformed disc.	500
10.9 Photos of GHOSTS schema.	503
10.10The audio signal transformation from "outer to "inner representation. The signal $x(n)$ is first windowed by the $w(n)$ window and transformed in the frequency domain. The short time spectral power is transformed from Hertz (f) to Bark (z) scale, band-limited and spread both in time and frequency.	507
10.11A sinusoid with broadband noise (top) and after the perceptual de-noise (bottom). Only the audible noise components are removed. X-axis: frequency normalized to the Nyquist frequency; Y-axis: Power Spectrum Magnitude (dB).	508
10.12Top: the waveform of the original (corrupted) audio extract. Bottom: the reconstructed data by CMSR. The increase of SNR can be noticed. X-axis: time (s). Y-axis: amplitude (normalized).	509
10.13Top: the waveform of the original (corrupted) audio extract. Bottom: the reconstructed data by CREAK. The click removal can be noticed. X-axis: time (s). Y-axis: amplitude (normalized).	510
10.14Top: the waveform of the original (corrupted) audio extract. Bottom: the restored data by PAR. X-axis: time (s). Y-axis: amplitude (normalized).	510
10.15Top: the spectrum of the original (corrupted) audio extract. Bottom: the restored data by PAR. X-axis: time (s). Y-axis: frequency (Hertz).	511

10.16Top: the waveform of the original (corrupted) audio extract. Middle: de-clicked and de-noised by CREAK. Bottom: de-noised by CMSR. X-axis: time (s). Y-axis: amplitude (normalized).	511
10.17Top: the spectrum of the original (corrupted) audio extract. Middle: de-clicked and de-noised by CREAK. Bottom: de-noised by CMSR. X-axis: time (s). Y-axis: frequency (Hertz).	512
10.18Gain trend introduced by the filters in the frequency domain at the varying of the input SNR (SNR_{out} - SNR_{in} vs. SNR_{in} in dB). The three filters have a good performance: CMSR for signal with low or medium SNR; CREAK for SNR [15 ÷ 30] dB; PAR for high SNR.	512

List of Tables

1.1	General properties of the discrete-time Fourier transform.	21
1.2	General properties of the z -Transform.	34
3.1	Summary of analogies in electrical, mechanical and acoustical systems.	107
6.1	Center frequencies and bandwidths for a critical-band filter bank, based on Eq. (6.17). . .	28
5.1	Retrieval effectiveness for correct queries	270
6.1	Three phases of information processing. [from S. Hashimoto 1997]	278
6.2	Correlation between coordinate axes and acoustic parameters.	290
6.3	Correlation between coordinate axes (INDSCAL - Group Space) and acoustic parameters (* $0.005 < p < 0.01$, ** $p < 0.005$).	296
6.4	Qualitative description of adjectives	304
6.5	Emotions and Speech Parameters (from Murray and Arnott, 1993).	304
6.6	Movement cues associated to different emotions.	313
6.7	Viseme and Related Phonemes.	315
6.8	Facial archetypal expressions.	316
7.1	Duration symbols for notes and rests.	323
7.2	Symbols for dynamics notation.	326
7.3	Correlation between the graph showing the durations of the various pitches in the Yankee Doodle excerpt and each of the major and minor key profiles.	348
7.4	Temperley key profiles. The note names refer to C major and C minor key.	348
8.1	Basic parameters for three classes of acoustic signals.	376
8.2	MIDI channel voice messages: c indicates the channel number.	390
8.3	MIDI System Real-Time Messages.	392
8.4	MIDI System Common Messages.	392
8.5	Example of a simple MIDI stream.	393
8.6	Examples of values and their variable-length equivalents.	394
8.7	Examples of midi file events.	395
10.1	Typologies of analogue mechanical carriers	489
10.2	Typology of magnetic tape carriers	490
10.3	Recommended climatic storage parameters for mechanical and tape characters	491
10.4	Mean for restored stimuli and anchors, 24 subjects. Stimuli: S1 = <i>My Mariuccia take-a steamboat</i> ; S2 = <i>La signorina sfinciosa</i> ; S3 = Unpublished tape recording	515

List of symbols

<i>Symbol</i>	<i>Quantity</i>	<i>Unit</i>
$t \in \mathbb{R}$	Continuous time	[s]
$n \in \mathbb{Z}$	Discrete time	
$s \in \mathbb{C}$	Complex variable in the Laplace domain	[rad/s]
$z \in \mathbb{C}$	Complex variable in the Z domain	
z^*	Conjugate of complex number z	
ω	Continuous frequency	$\text{Im}(s)$ [rad/s]
ω_d	Discrete frequency	$\arg(z)$ [rad]
F_s	Sampling rate	[Hz]
T_s	Sampling period	$1/F_s$ [s]
$s(t)$	Continuous-time signal	
$s[n]$	Discrete-time signal	
$S(s)$	Laplace-transformed signal	
$S(z)$	Z-transformed signal	
c	Sound speed in air	347 [m/s]
ρ_{air}	Air density	1.14 [Kg/m ³]

Chapter 1

Fundamentals of digital audio processing

Federico Avanzini and Giovanni De Poli

Copyright © 2005-2018 Federico Avanzini and Giovanni De Poli
except for paragraphs labeled as *adapted from <reference>*

This book is licensed under the CreativeCommons Attribution-NonCommercial-ShareAlike 3.0 license. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/>, or send a letter to Creative Commons, 171 2nd Street, Suite 300, San Francisco, California, 94105, USA.

1.1 Introduction

The purpose of this chapter is to provide the reader with fundamental concepts of digital signal processing, which will be used extensively in the remainder of the book. Since the focus is on audio signals, all the examples deal with sound. Those who are already fluent in DSP may skip this chapter.

1.2 Discrete-time signals and systems

1.2.1 Discrete-time signals

Signals play an important role in our daily life. Examples of signals that we encounter frequently are speech, music, picture and video signals. A signal is a function of independent variables such as time, distance, position, temperature and pressure. For examples, speech and music signals represent air pressure as a function of time at a point in space.

Most signals we encounter are generated by natural means. However, a signal can also be generated synthetically or by computer simulation. In this chapter we will focus our attention on a particular class of signals: The so called *discrete-time signals*. This class of signals is the most important way to describe/model the sound signals with the aid of the computer.

1.2.1.1 Main definitions

We define a signal x as a function $x : \mathcal{D} \rightarrow \mathcal{C}$ from a domain \mathcal{D} to a codomain \mathcal{C} . For our purposes the domain \mathcal{D} represents a time variable, although it may have different meanings (e.g. it may represent spatial variables). A signal can be classified based on the nature of \mathcal{D} and \mathcal{C} . In particular these sets can

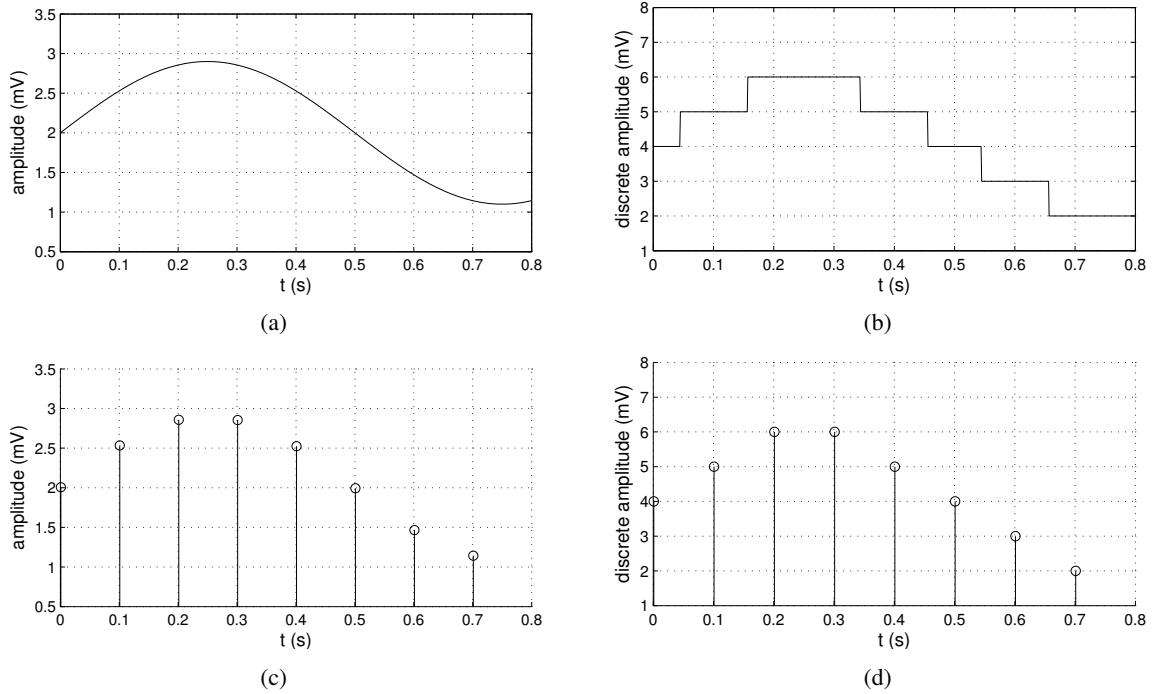


Figure 1.1: (a) Analog, (b) quantized-analog, (c) discrete-time, and (d) numerical signals.

be countable or non-countable. Moreover, \mathcal{C} may be a subset of \mathbb{R} or \mathbb{C} , i.e. the signal x may be either a real-valued or a complex-valued function.

When $\mathcal{D} = \mathbb{R}$ we talk of *continuous-time* signals $x(t)$, where $t \in \mathbb{R}$, while when $\mathcal{D} = \mathbb{Z}$ we talk of *discrete-time* signals $x[n]$. In this latter case $n \in \mathbb{Z}$ identifies discrete time instants t_n : the most common and important example is when $t_n = nT_s$, with T_s is a fixed quantity. In many practical applications a discrete-time signal x_d is obtained by *periodically sampling* a continuous-time signal x_c , as follows:

$$x_d[n] = x_c(nT_s) \quad -\infty < n < \infty, \quad (1.1)$$

The quantity T_s is called *sampling period*, measured in s. Its reciprocal is the *sampling frequency*, measured in Hz, and is usually denoted as $F_s = 1/T_s$. Note also the use of square brackets in the notation for a discrete-time signal $x[n]$, which avoids ambiguity with the notation $x(t)$ used for a continuous-time signal.

When $\mathcal{C} = \mathbb{R}$ we talk of *continuous-amplitude* signals, while when $\mathcal{C} = \mathbb{Z}$ we talk of *discrete-amplitude* signals. Typically the range of a discrete-amplitude signal is a finite set of M values $\{x_k\}_{k=1}^M$, and the most common example is that of a *uniformly quantized* signal with $x_k = kq$ (where q is called quantization step).

By combining the above options we obtain the following classes of signals, depicted in Fig. 1.1:

1. $\mathcal{D} = \mathbb{R}, \mathcal{C} = \mathbb{R}$: *analog* signal.
2. $\mathcal{D} = \mathbb{R}, \mathcal{C} = \mathbb{Z}$: *quantized analog* signal.
3. $\mathcal{D} = \mathbb{Z}, \mathcal{C} = \mathbb{R}$: *sequence*, or *sampled* signal.
4. $\mathcal{D} = \mathbb{Z}, \mathcal{C} = \mathbb{Z}$: *numerical*, or *digital*, signal. This is the type of signal that can be processed with the aid of the computer.



In these sections we will focus on discrete-time signals, regardless of whether they are quantized or not. We will equivalently use the terms discrete-time signal and sequence. We will always refer to a single value $x[n]$ as the n -th *sample* of the sequence x , regardless of whether the sequence has been obtained by sampling a continuous-time signal or not.

1.2.1.2 Basic sequences and operations

Sequences are manipulated through various basic operations. The *product* and *sum* between two sequences are simply defined as the sample-by-sample product sequence and sum sequence, respectively. *Multiplication by a constant* is defined as the sequence obtained by multiplying each sample by that constant. Another important operation is *time shifting* or *translation*: we say that a sequence $y[n]$ is a shifted version of $x[n]$ if

$$y[n] = x[n - n_0], \quad (1.2)$$

with $n_0 \in \mathbb{Z}$. For $n_0 > 0$ this is a *delaying* operation while for $n_0 < 0$ it is an *advancing* operation.

Several basic sequences are relevant in discussing discrete-time signals and systems. The simplest and the most useful sequence is the *unit sample sequence* $\delta[n]$, often referred to as *unit impulse* or simply *impulse*:

$$\delta[n] = \begin{cases} 1, & n = 0, \\ 0, & n \neq 0. \end{cases} \quad (1.3)$$

The unit impulse is also the simplest example of a *finite-length* sequence, defined as a sequence that is zero except for a finite interval $n_1 \leq n \leq n_2$. One trivial but fundamental property of the $\delta[n]$ sequence is that any sequence can be represented as a linear combination of delayed impulses:

$$x[n] = \sum_{k=-\infty}^{\infty} x[k] \delta[n - k]. \quad (1.4)$$

The *unit step sequence* is denoted by $u[n]$ and is defined as

$$u[n] = \begin{cases} 1, & n \geq 0, \\ 0, & n < 0. \end{cases} \quad (1.5)$$

The unit step is the simplest example of a *right-sided* sequence, defined as a sequence that is zero except for a right-infinite interval $n_1 \leq n < +\infty$. Similarly, *left-sided* sequences are defined as a sequences that are zero except for a left-infinite interval $-\infty < n \leq n_1$.

The unit step is related to the impulse by the following equalities:

$$u[n] = \sum_{k=0}^{\infty} \delta[n - k] = \sum_{k=-\infty}^n \delta[k]. \quad (1.6)$$

Conversely, the impulse can be written as the *first backward difference* of the unit step:

$$\delta[n] = u[n] - u[n - 1]. \quad (1.7)$$

The general form of the *real sinusoidal sequence* with constant amplitude is

$$x[n] = A \cos(\omega_0 n + \phi), \quad -\infty < n < \infty, \quad (1.8)$$

where A , ω_0 and ϕ are real numbers. By analogy with continuous-time functions, ω_0 is called *angular frequency* of the sinusoid, and ϕ is called the *phase*. Note however that, since n is dimensionless, the



dimension of ω_0 is radians. Very often we will say that the dimension of n is “samples” and therefore we will specify the units of ω_0 to be radians/sample. If $x[n]$ has been sampled from a continuous-time sinusoid with a given sampling rate F_s , we will also use the term *normalized angular frequency*, since in this case ω_0 is the continuous-time angular frequency normalized with respect to F_s .

Another relevant numerical signal is constructed as the sequence of powers of a real or complex number α . Such sequences are termed *exponential sequences* and their general form is

$$x[n] = A\alpha^n, \quad -\infty < n < \infty, \quad (1.9)$$

where A and α are real or complex constant. When α is complex, $x[n]$ has real and imaginary parts that are exponentially weighted sinusoid. Specifically, if $\alpha = |\alpha|e^{j\omega_0}$ and $A = |A|e^{j\phi}$, then $x[n]$ can be expressed as

$$x[n] = |A| |\alpha|^n \cdot e^{j(\omega_0 n + \phi)} = |A| |\alpha|^n \cdot (\cos(\omega_0 n + \phi) + j \sin(\omega_0 n + \phi)). \quad (1.10)$$

Therefore $x[n]$ can be expressed as $x[n] = x_{Re}[n] + jx_{Im}[n]$, with $x_{Re}[n] = |A| |\alpha|^n \cos(\omega_0 n + \phi)$ and $x_{Im}[n] = |A| |\alpha|^n \sin(\omega_0 n + \phi)$. These sequences oscillate with an exponentially growing magnitude if $|\alpha| > 1$, or with an exponentially decaying magnitude if $|\alpha| < 1$. When $|\alpha| = 1$, the sequences $x_{Re}[n]$ and $x_{Im}[n]$ are real sinusoidal sequences with constant amplitude and $x[n]$ is referred to as a the *complex exponential sequence*.

An important property of real sinusoidal and complex exponential sequences is that substituting the frequency ω_0 with $\omega_0 + 2\pi k$ (with k integer) results in sequences that are indistinguishable from each other. This can be easily verified and is ultimately due to the fact that n is integer. We will see the implications of this property when discussing the Sampling Theorem in Sec. 1.4.1, for now we will implicitly assume that ω_0 varies in an interval of length 2π , e.g. $(-\pi, \pi]$, or $[0, 2\pi]$.

Real sinusoidal sequences and complex exponential sequences are also examples of a *periodic sequence*: we define a sequence to be periodic with period $N \in \mathbb{N}$ if it satisfies the equality $x[n] = x[n + kN]$, for $-\infty < n < \infty$, and for any $k \in \mathbb{Z}$. The *fundamental period* N_0 of a periodic signal is the smallest value of N for which this equality holds. In the case of Eq. (1.8) the condition of periodicity implies that $\omega_0 N_0 = 2\pi k$. If $k = 1$ satisfies this equality we can say that the sinusoidal sequence is periodic with period $N_0 = 2\pi/\omega_0$, but this is not always true: the period may be longer or, depending on the value of ω_0 , the sequence may not be periodic at all.

1.2.1.3 Measures of discrete-time signals

We now define a set of useful metrics and measures of signals, and focus exclusively on digital signals. The first important metrics is *energy*: in physics, energy is the ability to do work and is measured in $\text{N}\cdot\text{m}$ or $\text{Kg}\cdot\text{m}^2/\text{s}^2$, while in digital signal processing physical units are typically discarded and signals are renormalized whenever convenient. The total *energy* of a sequence $x[n]$ is then defined as:

$$\mathcal{E}_x = \sum_{n=-\infty}^{\infty} |x[n]|^2. \quad (1.11)$$

Note that an infinite-length sequence with finite sample values may or not have finite energy. The rate of transporting energy is known as *power*. The average power of a sequence $x[n]$ is then defined as the average energy per sample:

$$\mathcal{P}_x = \frac{\mathcal{E}_x}{N} = \frac{1}{N} \sum_{n=0}^{N-1} |x[n]|^2. \quad (1.12)$$



Another common description of a signal is its *root mean square (RMS)* level. The RMS level of a signal $x[n]$ is simply $\sqrt{\mathcal{P}_x}$. In practice, especially in audio, the RMS level is typically computed after subtracting out any nonzero mean value, and is typically used to characterize periodic sequences in which \mathcal{P}_x is computed over a cycle of oscillation: as an example, the RMS level of a sinusoidal sequence $x[n] = A \cos(\omega_0 n + \phi)$ is $A/\sqrt{2}$.

In the case of sound signals, $x[n]$ will typically represent a sampled *acoustic pressure* signal. As a pressure wave travels in a medium (e.g., air), the RMS power is distributed all along the surface of the wavefront so that the appropriate measure of the strength of the wave is power per unit area of wavefront, also known as *intensity*.

Intensity is still proportional to the RMS level of the acoustic pressure, and relates to the sound level perceived by a listener. However, the usual definition of *sound pressure level (SPL)* does not directly use intensity. Instead the SPL of a pressure signal is measured in *decibels (dB)*, and is defined as

$$SPL = 10 \log_{10}(I/I_0) \quad (\text{dB}), \quad (1.13)$$

where I and I_0 are the RMS intensity of the signal and a reference intensity, respectively. In particular, in an *absolute dB scale* I_0 is chosen to be the smallest sound intensity that can be heard (more on this in Chapter *Auditory based processing*). The function of the dB scale is to transform ratios into differences: if I_2 is twice I_1 , then $SPL_2 - SPL_1 = 3 \text{ dB}$, no matter what the actual value of I_1 might be.¹

Because sound intensity is proportional to the square of the RMS pressure, it is easy to express level differences in terms of pressure ratios:

$$SPL_2 - SPL_1 = 10 \log_{10}(p_2^2/p_1^2) = 20 \log_{10}(p_2/p_1) \quad (\text{dB}). \quad (1.14)$$

Therefore, depending on the physical quantity which is being used the prefactor 20 or 10 may be employed in a decibel calculation. To resolve the uncertainty of which is the correct one, note that there are two kinds of quantities for which a dB scale is appropriate: “energy-like” quantities and “dynamical” quantities. An energy-like quantity is real and never negative: examples of such quantities are acoustical energy, intensity or power, electrical energy or power, optical luminance, etc., and the appropriate prefactor for these quantities in a dB scale is 10. Dynamical quantities may be positive or negative, or even complex in some representations: examples of such quantities are mechanical displacement or velocity, acoustical pressure, velocity or volume velocity, electrical voltage or current, etc., and the appropriate prefactor for these quantities in a dB scale is 20 (since they have the property that their squares are energy-like quantities).

1.2.1.4 Random signals

1.2.2 Discrete-time systems

Signal processing systems can be classified along the same lines used in Sec. 1.2 to classify signals. Here we are interested in discrete-time systems, that act on sequences and produce sequences as output.

1.2.2.1 Basic systems and block schemes

We define a discrete-time system as a transformation \mathcal{T} that maps an *input sequence* $x[n]$ into an *output sequence* $y[n]$:

$$y[n] = \mathcal{T}\{x\}[n]. \quad (1.15)$$

¹This is a special case of the Weber-Fechner law, which attempts to describe the relationship between the physical magnitudes of stimuli and the perceived intensity of the stimuli: the law states that this relation is logarithmic: if a stimulus varies as a geometric progression (i.e. multiplied by a fixed factor), the corresponding perception is altered in an arithmetic progression (i.e. in additive constant amounts).

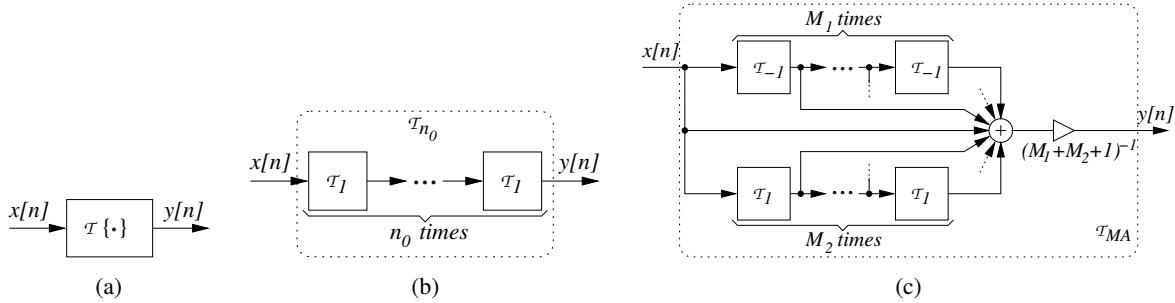


Figure 1.2: Block schemes of discrete-time systems; (a) a generic system $\mathcal{T}\{\cdot\}$; (b) ideal delay system \mathcal{T}_{n_0} ; (c) moving average system \mathcal{T}_{MA} . Note the symbols used to represent sums of signals and multiplication by a constant (these symbols will be formally introduced in Sec. 1.6.3).

Discrete-time systems are typically represented pictorially through *block schemes* that depict the signal flow graph. As an example the block scheme of Fig.1.2(a) represents the generic discrete-time system of Eq. (1.15).

The simplest concrete example of a discrete-time system is the *ideal delay* system \mathcal{T}_{n_0} , defined as

$$y[n] = \mathcal{T}_{n_0}\{x\}[n] = x[n - n_0], \quad (1.16)$$

where the integer n_0 is the delay of the system and can be both positive and negative: if $n_0 > 0$ then y corresponds to a time-delayed version of x , while if $n_0 < 0$ then the system operates a time advance. This system can be represented with the block-scheme of Fig. 1.2(b): this block-scheme also provides an example of *cascade connection* of systems, based on the trivial observation that the system \mathcal{T}_{n_0} can be seen as cascaded of n_0 unit delay systems \mathcal{T}_1 .

A slightly more complex example is the *moving average* system \mathcal{T}_{MA} , defined as

$$y[n] = \mathcal{T}_{MA}\{x\}[n] = \frac{1}{M_1 + M_2 + 1} \sum_{k=-M_1}^{M_2} x[n - k]. \quad (1.17)$$

The n -th sample of the sequence y is the average of $(M_1 + M_2 + 1)$ samples of the sequence x , centered around the sample $x[n]$, hence the name of the system. Fig. 1.2(c) depicts a block scheme of the moving average system: in this case all the branches carrying the shifted versions of $x[n]$ form a *parallel connection* in which they are all summed up and subsequently multiplied by the factor $1/(M_1 + M_2 + 1)$.

1.2.2.2 Classes of discrete-time systems

Classes of systems are defined by placing constraints on the properties of the transformation $\mathcal{T}\{\cdot\}$. Doing so often leads to very general mathematical representations.

We define a system to be *memoryless* if the output sequence $y[n]$ at every value of n depends only on the value of the input sequence $x[n]$ at the same value of n . As an example, the system $y[n] = \sin(x[n])$ is a memoryless system. On the other hand, the ideal delay system and the moving average system described in the previous section are not memoryless: these systems are referred to as *having memory*, since they must “remember” past (or even future) values of the sequence x in order to compute the “present” output $y[n]$.



We define a system to be *linear* if it satisfies the principle of superposition. If $y_1[n]$, $y_2[n]$ are the responses of a system \mathcal{T} to the inputs $x_1[n]$, $x_2[n]$, respectively, then \mathcal{T} is linear if and only if

$$\mathcal{T}\{a_1x_1 + a_2x_2\}[n] = a_1\mathcal{T}\{x_1\}[n] + a_2\mathcal{T}\{x_2\}[n], \quad (1.18)$$

for any pair of arbitrary constants a_1 and a_2 . Equivalently we say that a linear system possesses an additive property and a scaling property. As an example, the ideal delay system and the moving average system described in the previous section are linear systems. On the other hand, the memoryless system $y[n] = \sin(x[n])$ discussed above is clearly non-linear.

We define a system to be *time-invariant* (or *shift-invariant*) if a time shift of the input sequence causes a corresponding shift in the output sequence. Specifically, let $y = \mathcal{T}\{x\}$. Then \mathcal{T} is time-invariant if and only if

$$\mathcal{T}\{\mathcal{T}_{n_0}\{x\}\}[n] = y[n - n_0] \quad \forall n_0, \quad (1.19)$$

where \mathcal{T}_{n_0} is the ideal delay system defined previously. This relation between the input and the output must hold for any arbitrary input sequence x and its corresponding output. All the systems that we have examined so far are time-invariant. On the other hand, an example of non-time-invariant system is $y[n] = x[Mn]$ (with $M \in \mathbb{N}$). This system creates y by selecting one every M samples of x . One can easily see that $\mathcal{T}\{\mathcal{T}_{n_0}\{x\}\}[n] = x[Mn - n_0]$, which is in general different from $y[n - n_0] = x[M(n - n_0)]$.

We define a system to be *causal* if for every choice of n_0 the output sequence sample $y[n_0]$ depends only on the input sequence samples $x[n]$ with $n \leq n_0$. This implies that, if $y_1[n]$, $y_2[n]$ are the responses of a causal system to the inputs $x_1[n]$, $x_2[n]$, respectively, then

$$x_1[n] = x_2[n] \quad \forall n < n_0 \quad \Rightarrow \quad y_1[n] = y_2[n] \quad \forall n < n_0. \quad (1.20)$$

The moving average system discussed in the previous section is an example of a non-causal systems, since it needs to know M_1 “future” values of the input sequence in order to compute the current value $y[n]$. Apart from this, all the systems that we have examined so far are causal.

We define a system to be *stable* if and only if every bounded input sequence produces a bounded output sequence. A sequence $x[n]$ is said to be bounded if there exist a positive constant B_x such that

$$|x[n]| \leq B_x \quad \forall n. \quad (1.21)$$

Stability then requires that for such an input sequence there exists a positive constant B_y such that $|y[n]| \leq B_y \forall n$. This notion of stability is often referred to as *bounded-input bounded-output (BIBO)* stability. All the systems that we have examined so far are BIBO-stable. On the other hand, an example of unstable system is $y[n] = \sum_{k=-\infty}^n x[k]$. This is called the *accumulator* system, since $y[n]$ accumulates the sum of all past values of x . In order to see that the accumulator system is not stable it is sufficient to verify that $y[n]$ is not bounded when $x[n]$ is the step sequence.

1.2.3 Linear Time-Invariant Systems

Linear-time invariant (LTI) are a particularly relevant class of systems. A LTI system is any system that is both linear and time-invariant according to the definitions given in the previous section. As we will see in this section, LTI systems are mathematically easy to analyze and to characterize.

1.2.3.1 Impulse response and convolution

Let \mathcal{T} be a LTI system, $y[n] = \mathcal{T}\{x\}[n]$ be the output sequence given a generic input x , and $h[n]$ the *impulse response* of the system, i.e. $h[n] = \mathcal{T}\{\delta\}[n]$. Now, recall that every sequence $x[n]$ can be



represented as a linear combination of delayed impulses (see Eq. (1.4)). If we use this representation and exploit the linearity and time-invariance properties, we can write:

$$y[n] = \mathcal{T} \left\{ \sum_{k=-\infty}^{+\infty} x[k] \delta[n-k] \right\} = \sum_{k=-\infty}^{+\infty} x[k] \mathcal{T}\{\delta[n-k]\} = \sum_{k=-\infty}^{+\infty} x[k] h[n-k], \quad (1.22)$$

where in the first equality we have used the representation (1.4), in the second equality we have used the linearity property, and in the last equality we have used the time-invariance property.

Equation (1.22) states that a LTI system can be completely characterized by its impulse response $h[n]$, since the response to *any* input sequence $x[n]$ can be written as $\sum_{k=-\infty}^{+\infty} x[n]h[n-k]$. This can be interpreted as follows: the k -th input sample, seen as a single impulse $x[k]\delta[n-k]$, is transformed by the system into the sequence $x[k]h[n-k]$, and for each k these sequences are summed up to form the overall output sequence $y[n]$.

The sum on the right-hand side of Eq. (1.22) is called *convolution sum* of the sequences $x[n]$ and $h[n]$, and is usually denoted with the sign $*$. Therefore we have just proved that a LTI system \mathcal{T} has the property

$$y[n] = \mathcal{T}\{x\}[n] = (x * h)[n]. \quad (1.23)$$

Let us consider again the systems defined in the previous sections: we can find their impulse responses through the definition, i.e. by computing their response to an ideal impulse $\delta[n]$. For the ideal delay system the impulse response is simply a shifted impulse:

$$h_{n_0}[n] = \delta[n - n_0]. \quad (1.24)$$

The impulse response of the moving average system is easily computed as

$$h[n] = \frac{1}{M_1 + M_2 + 1} \sum_{k=-M_1}^{M_2} \delta[n-k] = \begin{cases} \frac{1}{M_1+M_2+1}, & -M_1 < n < M_2, \\ 0, & \text{elsewhere.} \end{cases} \quad (1.25)$$

Finally the accumulator system has the following impulse response:

$$h[n] = \sum_{k=-\infty}^n \delta[k] = \begin{cases} 1, & n \geq 0, \\ 0, & n < 0. \end{cases} \quad (1.26)$$

There is a fundamental difference between these impulse responses. The first two responses have a finite number of non-zero samples (1 and $M_1 + M_2 + 1$, respectively): systems that possess this property are called *finite impulse response (FIR)* systems. On the other hand, the impulse response of the accumulator has an infinite number of non-zero samples: systems that possess this property are called *infinite impulse response (IIR)* systems.

1.2.3.2 Properties of LTI systems

Since the convolution sum of Eq. (1.23) completely characterizes a LTI system, the most relevant properties of this class of systems can be understood by inspecting properties of the convolution operator. Clearly convolution is *linear*, otherwise \mathcal{T} would not be a linear system, which is by hypothesis. Convolution is also *associative*:

$$(x * (h_1 * h_2))[n] = ((x * h_1) * h_2)[n]. \quad (1.27)$$

Moreover convolution is *commutative*:

$$(x * h)[n] = \sum_{k=-\infty}^{\infty} x[n]h[n-k] = \sum_{m=-\infty}^{\infty} x[n-m]h[m] = (h * x)[n], \quad (1.28)$$



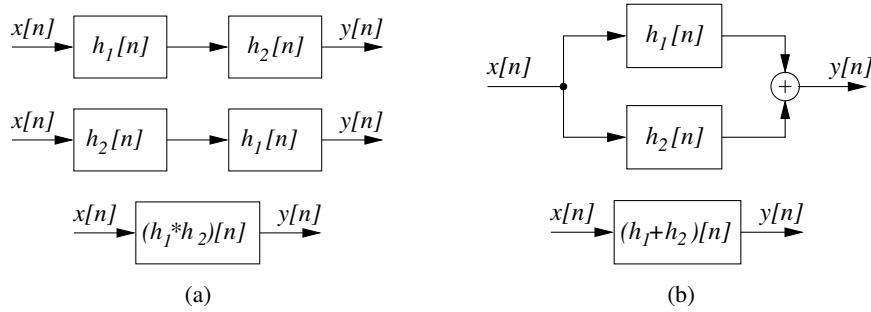


Figure 1.3: Properties of LTI system connections, and equivalent systems; (a) cascade, and (b) parallel connections.

where we have substituted the variable $m = n - k$ in the sum. This property implies that a LTI system with input $h[n]$ and impulse response $x[n]$ will have the same output of a LTI system with input $x[n]$ and impulse response $h[n]$. More importantly, associativity and commutativity have implications on the properties of *cascade connections* of systems. Consider the block scheme in Fig. 1.3(a) (upper panel): the output from the first block is $x * h_1$, therefore the final output is $(x * h_1) * h_2$, which equals both $(x * h_2) * h_1$ and $x * (h_1 * h_2)$. As a result the three block schemes in Fig. 1.3(a) represent three systems with the same impulse response.

Linearity and commutativity imply that the convolution is *distributive* over addition. From the definition (1.23) it is straightforward to prove that

$$(x * (h_1 + h_2))[n] = (x * h_1)[n] + (x * h_2)[n]. \quad (1.29)$$

Distributivity has implications on the properties of *parallel connections* of systems. Consider the block scheme in Fig. 1.3(b) (upper panel): the final output is $(x * h_1) + (x * h_2)$, which equals $x * (h_1 + h_2)$. As a result the two block schemes in Fig. 1.3(a) represent two systems with the same impulse response.

In the case of a LTI system, the notions of causality and stability given in the previous sections can also be related to properties of the impulse response. As for causality, it is a straightforward exercise to show that a LTI system is causal if and only if

$$h[n] = 0 \quad \forall n < 0. \quad (1.30)$$

For this reason, sequences that satisfy the above condition are usually termed *causal sequences*.

As for stability, recall that a system is BIBO-stable if any bounded input produces a bounded output. The response of a LTI system to a bounded input $x[n] \leq B_x$ is

$$|y[n]| = \left| \sum_{k=-\infty}^{+\infty} x[k]h[n-k] \right| \leq \sum_{k=-\infty}^{+\infty} |x[k]| |h[n-k]| \leq B_x \sum_{k=-\infty}^{+\infty} |h[n-k]|. \quad (1.31)$$

From this chain of inequalities we find that a sufficient condition for the stability of the system is

$$\sum_{k=-\infty}^{+\infty} |h[n-k]| = \sum_{k=-\infty}^{+\infty} |h[k]| < \infty. \quad (1.32)$$

One can prove that this is also a necessary condition for stability. Assume that Eq. (1.32) does not hold and define the input $x[n] = h^*[-n]/|h[n]|$ for $h[n] \neq 0$ ($x = 0$ elsewhere): this input is bounded by



unity, however one can immediately prove that $y[0] = \sum_{k=-\infty}^{+\infty} |h[k]| = +\infty$. In conclusion, a LTI system is stable if and only if h is absolutely summable, or $h \in L^1(\mathbb{Z})$. A direct consequence of this property is that FIR systems are always stable, while IIR systems may not be stable.

Using the properties demonstrated in this section, we can look back at the impulse responses of Eqs. (1.24,1.25,1.26), and we can immediately prove whether they are stable and causal.

1.2.3.3 Constant-coefficient difference equations

Consider the following *constant-coefficient difference equation*:

$$\sum_{k=0}^N a_k y[n-k] = \sum_{k=0}^M b_k x[n-k]. \quad (1.33)$$

Question: given a set of values for $\{a_k\}$ and $\{b_k\}$, does this equation define a LTI system? The answer is no, because a given input $x[n]$ does not univocally determine the output $y[n]$. In fact it is easy to see that, if $x[n], y[n]$ are two sequences satisfying Eq. (1.33), then the equation is satisfied also by the sequences $x[n], y[n] + y_h[n]$, where y_h is *any* sequence that satisfies the *homogeneous equation*:

$$\sum_{k=0}^N a_k y_h[n-k] = 0. \quad (1.34)$$

One could show that y_h has the general form $y_h[n] = \sum_{m=1}^N A_m z_m^n$, where the z_m 's are roots of the polynomial $\sum_{k=0}^N a_k z^k$ (this can be verified by substituting the general form of y_h into Eq. (1.34)).

The situation is very much like that of linear constant-coefficient differential equations in continuous-time: since y_h has N undetermined coefficients A_m , we must specify N additional constraints in order for the equation to admit a unique solution. Typically we set some *initial conditions*. For Eq. (1.33), an initial condition is a set of N consecutive “initial” samples of $y[n]$. Suppose that the samples $y[-1], y[-2], \dots, y[-N]$ have been fixed: then all the infinite remaining samples of y can be recursively determined through the recurrence equations

$$y[n] = \begin{cases} -\sum_{k=1}^N \frac{a_k}{a_0} y[n-k] + \sum_{m=0}^M \frac{b_m}{a_0} x[n-m], & n \geq 0, \\ -\sum_{k=0}^{N-1} \frac{a_k}{a_0} y[n+N-k] + \sum_{m=0}^M \frac{b_m}{a_0} x[n+N-m], & n \leq -N-1. \end{cases} \quad (1.35)$$

In particular, $y[0]$ is determined with the above equation using the initial values $y[-1], y[-2], \dots, y[-N]$, then $y[1]$ is determined using the values $y[-0], y[-1], \dots, y[-N+1]$, and so on.

Another way of guaranteeing that Eq. (1.33) specifies a unique solution is requiring that the LTI system is also *causal*. If we look back at the definition of causality, we see that in this context it implies that for an input $x[n] = 0 \forall n < n_0$, then $y[n] = 0 \forall n < n_0$. Then again we have sufficient initial conditions to recursively compute $y[n]$ for $n \geq n_0$: in this case we can speak of *initial rest conditions*.

All the LTI systems that we have examined so far can actually be written in the form (1.33). As an example, let us examine the accumulator system: we can write

$$y[n] = \sum_{k=-\infty}^n x[k] = x[n] + \sum_{k=-\infty}^{n-1} x[k] = x[n] + y[n-1], \quad (1.36)$$



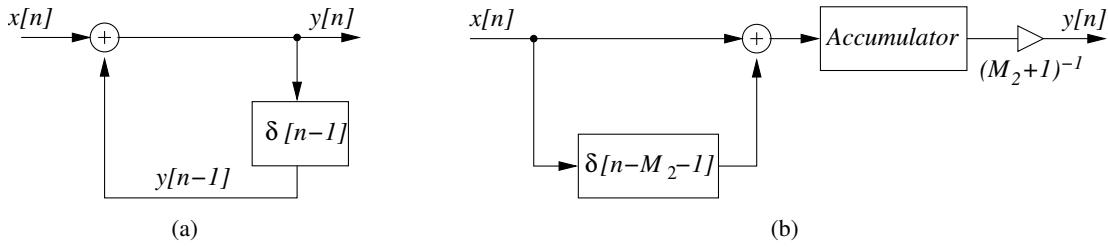


Figure 1.4: Block schemes for constant-coefficient difference equations representing (a) the accumulator system, and (b) the moving average system.

where in the second equality we have simply separated the term $x[n]$ from the sum, and in the third equality we have applied the definition of the accumulator system for the output sample $y[n - 1]$. Therefore, input and output of the accumulator system satisfy the equation

$$y[n] - y[n - 1] = x[n], \quad (1.37)$$

which is in form (1.33) with $N = 1$, $M = 0$ and with $a_0 = 1$, $a_1 = -1$, $b_0 = 1$. This also means that we can implement the accumulator with the block scheme of Fig. 1.4(a).

As a second example, consider the moving average system, with $M_1 = 0$ (so that the system is causal). Then Eq. (1.17) is already a constant-coefficient difference equation. But we can also represent the system with a different equation by noting that

$$y[n] = \frac{1}{M_2 + 1} \sum_{k=0}^{M_2} x[n - k] = \frac{1}{M_2 + 1} \sum_{-\infty}^n (x[n] - x[n - M_2 - 1]). \quad (1.38)$$

Now we note that the sum on the right-hand side represents an accumulator applied to the signal $x_1[n] = (x[n] - x[n - M_2 - 1])$. Therefore we can apply Eq. (1.37) and write

$$y[n] - y[n - 1] = \frac{1}{M_2 + 1} x_1[n] = \frac{1}{M_2 + 1} (x[n] - x[n - M_2 - 1]). \quad (1.39)$$

We have then found a totally different equation, which is still in the form (1.33) and still represents the moving average system. The corresponding block scheme is given in Fig. 1.4(b). This example also shows that different equations of the form (1.33) can represent the same LTI system.

1.3 Signal generators

In this section we describe methods and algorithms used to directly generate a discrete-time signal. Specifically we will examine periodic waveform generators and noise generators, which are both particularly relevant in audio applications.

1.3.1 Digital oscillators

Many relevant musical sounds are almost periodic in time. The most direct method for synthesizing a periodic signal is repeating a single period of the corresponding waveform. An algorithm that implements this method is called oscillator. The simplest algorithm consists in computing the appropriate value of the waveform for every sample, assuming that the waveform can be approximately described through a polynomial or rational truncated series. However this is definitely not the most efficient approach. More efficient algorithms are presented in the remainder of this section.



1.3.1.1 Table lookup oscillator

A very efficient approach is to pre-compute the samples of the waveform, store them in a table which is usually implemented as a circular buffer, and access them from the table whenever needed. If a copy of one period of the desired waveform is stored in such a *wavetable*, a periodic waveform can be generated by cycling over the wavetable with the aid of a circular pointer. When the pointer reaches the end of the table, it wraps around and points again at the beginning of the table.

Given a table of length L samples, the period T_0 of the generated waveform depends on the sampling period T_s at which samples are read. More precisely, the period is given by $T_0 = LT_s$, and consequently the fundamental frequency is $f_0 = F_s/L$. This implies that in order to change the frequency (while maintaining the sample sampling rate), we would need the same waveform to be stored in tables of different lengths.

A better solution is the following. Imagine that a single wavetable is stored, composed of a very large number L of equidistant samples of the waveform. Then for a given sampling rate F_s and a desired signal frequency f_0 , the number of samples to be generated in a single cycle is F_s/f_0 . From this, we can define the *sampling increment (SI)*, which is the distance in the table between two samples at subsequent instants. The *SI* is given by the following equation:

$$SI = \frac{L}{F_s/f_0} = \frac{f_0 L}{F_s}. \quad (1.40)$$

Therefore the *SI* is proportional to f_0 . Having defined the sampling increment, samples of the desired signal are generated by reading one every *SI* samples of the table. If the *SI* is not an integer, the closest sample of the table will be chosen (obviously, the larger L , the better the approximation). In this way, the oscillator resample the table to generate a waveform with different fundamental frequencies.

M-1.1

Implement in Matlab a circular look-up from a table of length L and with sampling increment SI .

M-1.1 Solution

```
phi=mod(phi +SI,L);
s=tab[round(phi)];
```

where `phi` is a state variable indicating the reading point in the table, `A` is a scaling parameter, `s` is the output signal sample. The function `mod(x,y)` computes the remainder of the division x/y and is used here to implement circular reading of the table. Notice that `phi` can be a non integer value. In order to use it as array index, it has to be truncated, or rounded to the nearest integer (as we did in the code above). A more accurate output can be obtained by linear interpolation between adjacent table values.

1.3.1.2 Recurrent sinusoidal signal generators

Sinusoidal signals can be generated also by recurrent methods. A first method is based on the following equation:

$$y[n+1] = 2 \cos(\omega_0) y[n] - y[n-1] \quad (1.41)$$

where $\omega_0 = 2\pi f_0/F_s$ is the normalized angular frequency of the sinusoid. Then one can prove that given the initial values $y[0] = \cos \phi$ and $y[-1] = \cos(\phi - \omega_0)$ the generator produces the sequence

$$y[n] = \cos(\omega_0 n + \phi). \quad (1.42)$$



In particular, with initial values $y[0] = 1$ and $y[-1] = \cos \omega_0$ the generator produces the sequence $y[n] = \cos(\omega_0 n)$, while with initial conditions $y[0] = 0$ and $y[-1] = -\sin \omega_0$ it produces the sequence $y[n] = \sin(\omega_0 n)$. This property can be justified by recalling the trigonometric relation $\cos \omega_0 \cdot \cos \phi = 0.5[\cos(\phi + \omega_0) + \cos(\phi - \omega_0)]$.

A second recursive method for generating sinusoidal sequence combines both the sinusoidal and cosinusoidal generators and is termed *coupled form*. It is described by the equations

$$\begin{aligned} x[n+1] &= \cos \omega_0 \cdot x[n] - \sin \omega_0 \cdot y[n], \\ y[n+1] &= \sin \omega_0 \cdot x[n] + \cos \omega_0 \cdot y[n]. \end{aligned} \quad (1.43)$$

With $x[0] = 1$ and $y[0] = 0$ the sequences $x[n] = \cos(\omega_0 n)$ and $y[n] = \sin(\omega_0 n)$ are generated. This property can be verified by noting that for the complex exponential sequence the trivial relation $e^{j\omega_0(n+1)} = e^{j\omega_0} e^{j\omega_0 n}$ holds. From this relation, the above equations are immediately proved by calling $x[n]$ and $y[n]$ the real and imaginary parts of the complex exponential sequence, respectively.

A major drawback of both these recursive methods is that they are not robust against quantization. Small quantization errors in the computation will cause the generated signals either to grow exponentially or to decay rapidly into silence. To avoid this problem, a periodic re-initialization is advisable. It is possible to use a slightly different set of coefficients to produce absolutely stable sinusoidal waveforms

$$\begin{aligned} x[n+1] &= x[n] - c \cdot y[n], \\ y[n+1] &= c \cdot x[n+1] + y[n], \end{aligned} \quad (1.44)$$

where $c = 2 \sin(\omega_0/2)$. With $x[0] = 1$ and $y[0] = c/2$ we have $x[n] = \cos(\omega_0 n)$.

1.3.1.3 Control signals and envelope generators

Amplitude and frequency of a sound are usually required to be time-varying parameters. Amplitude control can be needed to define suitable sound envelopes, or to create effects such as *tremolo* (quasi-periodic amplitude variations around an average value). Frequency control can be needed to simulate continuous gliding between two tones (*portamento*, in musical terms), or to obtain subtle pitch variations in the sound attack/release, or to create effects such as *vibrato* (quasi-periodic pitch variations around an average value), and so on. We then want to construct a digital oscillator of the form

$$x[n] = a[n] \cdot \text{tab}\{\phi[n]\}, \quad (1.45)$$

where $a[n]$ scales the amplitude of the signal, while the phase $\phi[n]$ relates to the *instantaneous frequency* $f_0[n]$ of the signal: if $f_0[n]$ is not constant, then $\phi[n]$ does not increase linearly in time. Figure 1.5(a) shows the symbol usually adopted to depict an oscillator with fixed waveform and varying amplitude and frequency.

The signals $a[n]$, and $f_0[n]$ are usually referred to as *control signals*, as opposed to *audio signals*. The reason for this distinction is that control signals vary on a much slower time-scale than audio signals (as an example, a musical *vibrato* usually have a frequency of no more than ~ 5 Hz). Accordingly, many sound synthesis languages define control signals at a different (smaller) rate than the audio sampling rate F_s . This second rate is called *control rate*, or *frame rate*: a frame is a time window with pre-defined length (e.g. 5 or 50 ms), in which the control signals can be reasonably assumed to have small variations. We will use the notation F_c for the control rate.

Suitable control signals can be synthesized using *envelope generators*. An envelope generator can be constructed through the table-lookup approach described previously. In this case however the table will be read only once since the signal to be generated is not periodic. Given a desired duration (in seconds) of the control signal, the appropriate sampling increment will be chosen accordingly.



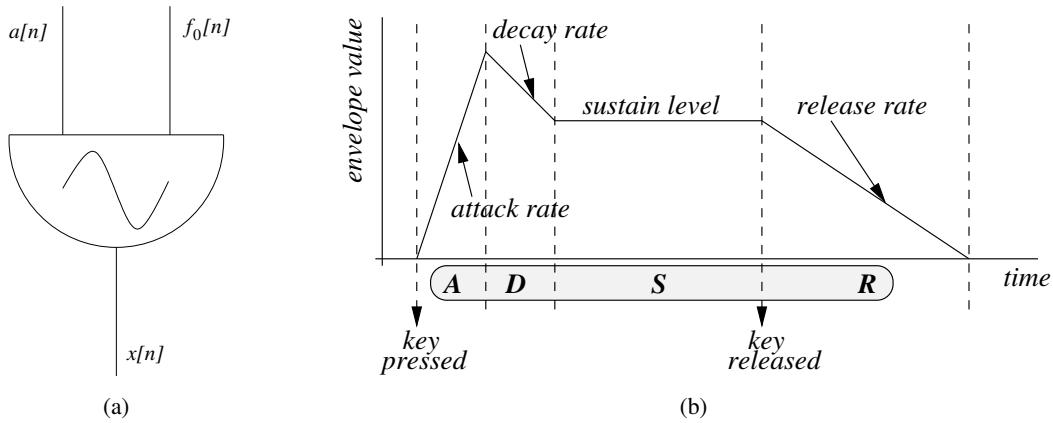


Figure 1.5: Controlling a digital oscillator; (a) symbol of the digital controlled in amplitude and frequency; (b) example of an amplitude control signal generated with an ADSR envelope.

Alternatively, envelope generators can be constructed by specifying values of control signals at a few control points and interpolating the signal in between them. In the simplest formulation, linear interpolation is used. In order to exemplify this approach, we discuss the so-called *Attack, Decay, Sustain, and Release (ADSR)* envelope typically used in sound synthesis applications to describe the time-varying amplitude $a[n]$. This envelope is shown in Fig. 1.5(b)): amplitude values are specified only at the boundaries between ADSR phases, and within each phase the signal varies linearly.

The attack and release phases mark the identity of the sound, while the central phases are associated with the steady-state portion of the sound. Therefore, in order to synthesize two sounds with the similar identity (or timbre) but different durations, it is advisable to only slightly modify the duration of attack and release, while the decay and especially sustain can be lengthened more freely.

M-1.2

Write a function that realizes a line-segment envelope generator. The input to the function are a vector of time instants and a corresponding vector of envelope values.

M-1.2 Solution

```
function env = envgen(t,a,method);
    %t= vector of control time instants
    %a= vector of envelope values

    global Fs; global SpF;      %global variables: sample rate, samples-per-frame

    if (nargin<3) method='linear'; end

    frt=floor(t*Fs/SpF+1);           %control time instants as frame numbers
    nframes=frt(length(frt));        %total number of frames
    env=interp1(frt,a,[1:nframes],method); %linear (or other method) interpolation
```

The envelope shape is specified by break-points, described as couples (time instant (sec) and amplitude). The function generates the envelope at frame rate. Notice that the interpolation function `interp1` allows to easily use cubic or *spline* interpolations.

The use of waveform and envelope generators allows to generate quasi periodic sounds with very limited hardware and constitutes the building block of many more sophisticated algorithms.



M-1.3

Assume that a function `sinosc(t0, a, f, ph0)` realizes a sinusoidal oscillator controlled in frequency and amplitude, with `t0` initial time, `a, f` frame-rate amplitude and frequency vectors, and `ph0` initial phase (see example M-1.4). Then generate a sinusoid with varying amplitude and constant frequency.

M-1.3 Solution

```
global Fs; global SpF; %global variables: sample rate, samples-per-frame

Fs=22050;
framelength=0.01;           %frame length (in s)
SpF=round(Fs*framelength); %samples per frame

%%% define controls %%%
slength=2;                  %sound length (in s)
nframes=slength*Fs/SpF;     %total no. of frames
f=50*ones(1,nframes);       %constant frequency (Hz)
a=envgen([0,.2,3,3.5,4],[0,1,.8,.5,0],'linear'); %ADSR amp. envelope

s=sinosc(0,a,f,0);         % compute sound signal
```

Note the structure of this simple example: in the “headers” section some global parameters are defined, that need to be known also to auxiliary functions; a second section defines the control parameters, and finally the audio signal is computed.

1.3.1.4 Frequency controlled oscillators

While realizing an amplitude modulated oscillator is quite straightforward, realizing a frequency modulated oscillator requires some more work. First of all we have to understand what is the instantaneous frequency of such an oscillator and how it relates to the phase function ϕ . This can be better understood in the continuous time domain. When the oscillator frequency is constant the phase is a linear function of time, $\phi(t) = 2\pi f_0 t$. In the more general case in which the frequency varies at frame rate, the following equation holds:

$$f_0(t) = \frac{1}{2\pi} \frac{d\phi}{dt}(t), \quad (1.46)$$

which simply says that the instantaneous angular frequency $\omega_0(t) = 2\pi f_0(t)$ is the instantaneous angular velocity of the time-varying phase $\phi(t)$. If $f_0(t)$ is varying slowly enough (i.e. it is varying at frame rate), we can say that in the k -th frame the following first-order approximation holds:

$$\frac{1}{2\pi} \frac{d\phi}{dt}(t) = f_0(t) \sim f_0(t_k) + F_c [f_0(t_{k+1}) - f_0(t_k)] \cdot (t - t_k), \quad (1.47)$$

where t_k, t_{k+1} are the initial instants of frames k and $k+1$, respectively. The term $F_c [f_0(t_{k+1}) - f_0(t_k)]$ approximates the derivative df_0/dt inside the k th frame. We can then find the phase function by integrating equation (1.47):

$$\phi(t) = \phi(t_k) + 2\pi f_0(t_k)(t - t_k) + 2\pi F_c [f_0(t_{k+1}) - f_0(t_k)] \frac{(t - t_k)^2}{2}. \quad (1.48)$$

From this equation, the discrete-time signal $\phi[n]$ can be computed within the k th frame, i.e. for the time indexes $(k-1) \cdot SpF + n$, with $n = 0 \dots (SpF - 1)$.

In summary, Eq. (1.48) allows to compute $\phi[n]$ at sample rate inside the k th frame, given the frame rate frequency values $f_0(t_k)$ and $f_0(t_{k+1})$. The key ingredient of this derivation is the linear interpolation (1.47).



M-1.4

Realize the `sinosc(t0,a,f,ph0)` function that we have used in M-1.3. Use equation (1.48) to compute the phase given the frame-rate frequency vector f .

M-1.4 Solution

```
function s = sinosc(t0,a,f,ph0);

global Fs; global SpF; %global variables: sample rate, samples-per-frame

nframes=length(a); %total number of frames
if (length(f)==1) f=f*ones(1,nframes); end
if (length(f) ~= nframes) error('wrong f length!'); end

s=zeros(1,nframes*SpF); %initialize signal vector to 0
lasta=a(1); lastf=f(1); lastph=ph0; %initialize amplitude, frequency, phase

for i=1:nframes %cycle on the frames
    naux=1:SpF; %count samples within frame
    %%%%%%%%%%%%%%% compute amplitudes and phases within frame %%%%%%
    ampl=lasta + (a(i)-lasta)/SpF.*naux;
    phase=lastph +pi/Fs.*naux.*(2*lastf +(1/SpF)*(f(i)-lastf).*naux);
    %%%%%%%%%%%%%%% read from table %%%%%%
    s(((i-1)*SpF+1):i*SpF)=ampl.*cos(phase); %read from table
    %%%%%%%%%%%%%%% save last values of amplitude, frequency, phase
    lasta=a(i); lastf=f(i); lastph=phase(SpF);
end
s=[zeros(1,round(t0*Fs)) s]; %add initial silence of t0 sec.
```

Both the amplitude a and frequency f envelopes are defined at frame rate and are interpolated at sample rate inside the function body. Note in particular the computation of the phase vector within each frame.

We can finally listen to a sinusoidal oscillator controlled both in amplitude and in frequency.

M-1.5

Synthesize a sinusoid modulated both in amplitude and frequency, using the functions `sinosc` and `envgen`.

M-1.5 Solution

```
global Fs; global SpF; %global variables: sample rate, samples-per-frame

Fs=22050;
framelength=0.01; %frame length (in s)
SpF=round(Fs*framelength); %samples per frame

%%% define controls %%%
a=envgen([0,.2,3,3.5,4],[0,1,.8,.5,0],'linear'); %ADSR amp. envelope
f=envgen([0,.2,3,4],[200,250,250,200],'linear'); %pitch envelope
f=f+max(f)*0.05*... %pitch envelope with vibrato added
sin(2*pi*5*(SpF/Fs)*[0:length(f)-1]).*hanning(length(f))';

%%% compute sound %%%
s=sinosc(0,a,f,0);
```

Amplitude a and frequency f control signals are shown in Fig. 1.6.



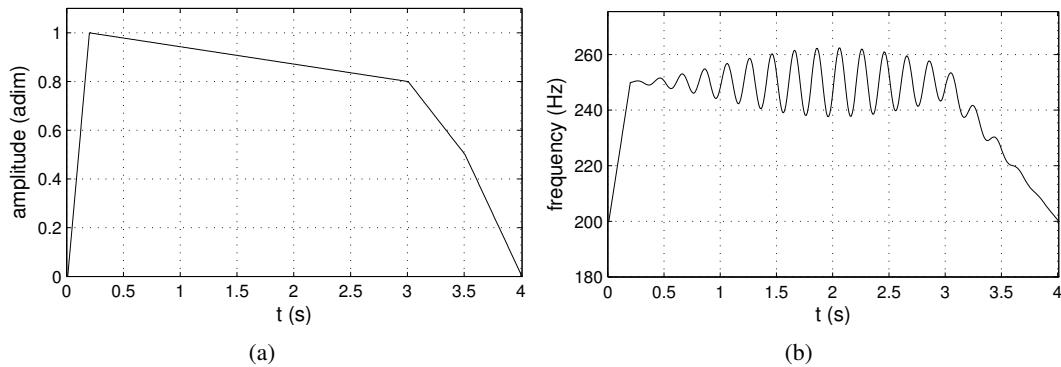


Figure 1.6: Amplitude (a) and frequency (b) control signals

1.3.2 Noise generators

Up to now, we have considered signals whose behavior at any instant is supposed to be perfectly knowable. These signals are called deterministic signals. Besides these signals, *random signals* of unknown or only partly known behavior may be considered. For random signals, only some general characteristics, called statistical properties, are known or are of interest. The statistical properties are characteristic of an entire signal class rather than of a single signal. A set of random signals is represented by a random process. Particular numerical procedures simulate random processes, producing sequences of random (or more precisely, pseudorandom) numbers.

Random sequences can be used both as signals (i.e., to produce white or colored noise used as input to a filter) and as control functions to provide a variety in the synthesis parameters most perceptible by the listener. In the analysis of natural sounds, some characteristics vary in an unpredictable way; their mean statistical properties are perceptibly more significant than their exact behavior. Hence, the addition of a random component to the deterministic functions controlling the synthesis parameters is often desirable. In general, a combination of random processes is used because the temporal organization of the musical parameters often has a hierarchical aspect. It cannot be well described by a single random process, but rather by a combination of random processes evolving at different rates. For example this technique is employed to generate $1/f$ noise.

1.3.2.1 White noise generators

The spread part of the spectrum is perceived as random noise. In order to generate a random sequence, we need a random number generator. There are many algorithms that generate random numbers, typically uniformly distributed over the standardized interval $[0, 1)$. However it is hard to find good random number generators, i.e. that pass all or most criteria of randomness. The most common is the so called *linear congruential* generator. It can produce fairly long sequences of independent random numbers, typically of the order of two billion numbers before repeating periodically. Given an initial number (seed) $I[0]$ in the interval $0 \leq I[0] < M$, the algorithm is described by the recursive equations

$$\begin{aligned} I[n] &= (aI[n-1] + c) \bmod M \\ u[n] &= I[n]/M \end{aligned} \quad (1.49)$$

where a and c are two constants that should be chosen very carefully in order to have a maximal length sequence, i.e. long M samples before repetition. The actual generated sequence depends on the initial



value $I[0]$; that is way the sequence is called pseudorandom. The numbers are uniformly distributed over the interval $0 \leq u[n] < 1$. The mean is $E[u] = 1/2$ and the variance is $\sigma_u^2 = 1/12$. The transformation $s[n] = 2u[n] - 1$ generates a zero-mean uniformly distributed random sequence over the interval $[-1, 1)$. This sequence corresponds to a white noise signal because the generated numbers are mutually independent. The power spectral density is given by $S(f) = \sigma_u^2$. Thus the sequence contains all the frequencies in equal proportion and exhibits equally slow and rapid variation in time.

With a suitable choice of the coefficients a and b , it produces pseudorandom sequences with flat spectral density magnitude (white noise). Different spectral shapes can be obtained using white noise as input to a filter.

M-1.6

A method of generating a Gaussian distributed random sequence is based on the central limit theorem, which states that the sum of a large number of independent random variables is Gaussian. As exercise, implement a very good approximation of a Gaussian noise, by summing 12 independent uniform noise generators.

If we desire that the numbers vary at a slower rate, we can generate a new random number every d sampling instants and hold the previous value in the interval (*holder*) or interpolate between two successive random numbers (*interpolator*). In this case the power spectrum is given by

$$S(f) = |H(f)|^2 \frac{\sigma_u^2}{d}$$

with

$$|H(f)| = \left| \frac{\sin(\pi f d / F_s)}{\sin(\pi f / F_s)} \right|$$

for the holder and

$$|H(f)| = \frac{1}{d} \left[\frac{\sin(\pi f d / F_s)}{\sin(\pi f / F_s)} \right]^2$$

for linear interpolation.

1.3.2.2 Pink noise generators

1/f noise generators A so-called *pink noise* is characterized by a power spectrum that fall in frequency like $1/f$:

$$S(f) = \frac{A}{f}. \quad (1.50)$$

For this reason pink noise is also called *1/f noise*. To avoid the infinity at $f = 0$, this behaviour is assumed valid for $f \geq f_{min}$, where f_{min} is a desired minimum frequency. The spectrum is characterized by a 3 db per octave drop, i.e. $S(2f) = S(f)/2$. The amount of power contained within a frequency interval $[f_1, f_2]$ is

$$\int_{f_1}^{f_2} S(f) df = A \ln \left(\frac{f_1}{f_2} \right)$$

This implies that the amount of power in any octave is the same. $1/f$ noise is ubiquitous in nature and is related to fractal phenomena. In audio domain it is known as pink noise. It represents the psychoacoustic equivalent of the white noise because it approximately excites uniformly the critical bands. The physical interpretation is a phenomenon that depends on many processes that evolve on different time scales. So a $1/f$ signal can be generated by the sum of several white noise generators that are filtered through first-order filters having the time constants that are successively larger and larger, forming a geometric progression.



M-1.7

In the Voss $1/f$ noise generation algorithm, the role of the low pass filters is played by the hold filter seen in the previous paragraph. The $1/f$ noise is generated by taking the average of several periodically held generators $y_i[n]$, with periods forming a geometric progression $d_i = 2^i$, i.e.

$$y[n] = \frac{1}{M} \sum_{i=1}^M y_i[n] \quad (1.51)$$

The power spectrum does not have an exact $1/f$ shape, but it is close to it for frequencies $f \geq F_s/2^M$. As exercise, implement a $1/f$ noise generator and use it to assign the pitches to a melody.

M-1.8

The music derived from the $1/f$ noise is closed to the human music: it does not have the unpredictability and randomness of white noise nor the predictability of brown noise. $1/f$ processes correlate logarithmically with the past. Thus the averaged activity of the last ten events has as much influence on the current value as the last hundred events, and the last thousand. Thus they have a relatively long-term memory.

$1/f$ noise is a fractal one; it exhibits self-similarity, one property of the fractal objects. In a self-similar sequence, the pattern of the small details matches the pattern of the larger forms, but on a different scale. In this case, is used to say that $1/f$ fractional noise exhibits statistical self-similarity. The pink noise algorithm for generating pitches has become a standard in algorithmic music. Use the $1/f$ generator developed in M-1.7 to produce a fractal melody.

1.4 Spectral analysis of discrete-time signals

Spectral analysis is one of the powerful analysis tool in several fields of engineering. The fact that we can decompose complex signals with the superposition of other simplex signals, commonly sinusoid or complex exponentials, highlights some signal features that sometimes are very hard to discover otherwise. Furthermore, the decomposition on simpler functions in the frequency domain is very useful when we want to perform modifications on a signal, since it gives the possibility to manipulate single spectral components, which is hard if not impossible to do on the time-domain waveform.

A rigorous and comprehensive tractation of spectral analysis is out the scope of this book. In this section we introduce the *Discrete-Time Fourier Transform (DTFT)*, which the discrete-time version of the classical Fourier Transform of continuous-time signals. Using the DTFT machinery, we then discuss briefly the main problems related to the process of sampling a continuous-time signal, namely frequency aliasing. This discussion leads us to the sampling theorem.

1.4.1 The discrete-time Fourier transform

1.4.1.1 Definition

Recall that for a continuous-time signal $x(t)$ the Fourier Transform is defined as:

$$\mathcal{F}\{x\}(\omega) = X(\omega) = \int_{-\infty}^{+\infty} x(t)e^{-j2\pi ft} dt = \int_{-\infty}^{+\infty} x(t)e^{-j\omega t} dt \quad (1.52)$$

where the variable f is *frequency* and is expressed in Hz, while the *angular frequency* ω has been defined as $\omega = 2\pi f$ and expressed in radians/s. Note that we are following the conventional notation by which time-domain signals are denoted using lowercase symbols (e.g., $x(n)$) while frequency-domain signals are denoted in uppercase (e.g., $X(\omega)$).

We can try to find an equivalent expression in the case of a discrete-time signal $x[n]$. If we think of $x[n]$ as the sampled version of a continuous-time signal $x(t)$ with a sampling interval $T_s = 1/F_s$,



i.e. $x[n] = x(nT_s)$, we can define the *discrete-time Fourier transform (DTFT)* starting from Eq. (1.52) where the integral is substituted by a summation:

$$\mathcal{F}\{x\}(\omega_d) = X(\omega_d) = \sum_{n=-\infty}^{+\infty} x(nT_s)e^{-j2\pi f \frac{n}{F_s}} = \sum_{n=-\infty}^{+\infty} x[n]e^{-j\omega_d n}. \quad (1.53)$$

There are two remarks to be made about this equation. First, we have omitted the scaling factor T_s in front of the summation, which would be needed to have a perfect correspondence with Eq. (1.52) but is irrelevant to our tractation. Second, we have defined a new variable $\omega_d = 2\pi f/F_s$: we call this the *normalized* (or *digital*) angular frequency. This is not to be confused with the angular frequency ω used in Eq. (1.52): ω_d is measured in radians/sample, and varies in the range $[-2\pi, 2\pi]$ when f varies in the range $[-F_s, F_s]$. In this book we use the notation ω to indicate the angular frequency in radians/s, and ω_d to indicate the normalized angular frequency in radians/sample.

As one can verify from Eq. (1.53), $X(\omega_d)$ is a periodic function in ω_d with a period 2π . Note that this periodicity of 2π in ω_d corresponds to a periodicity of F_s in the domain of the absolute-frequency f . Moreover $X(\omega_d)$ is in general a complex function, and can thus be written in terms of its real and imaginary parts, or alternatively in polar form as

$$X(\omega_d) = |X(\omega_d)| e^{\arg[X(\omega_d)]}, \quad (1.54)$$

where $|X(\omega_d)|$ is the *magnitude function* and $\arg[X(\omega_d)]$ is the *phase function*. Both are real-valued functions. Given the 2π periodicity of $X(\omega_d)$ we will arbitrarily assume that $-\pi < \arg[X(\omega_d)] < \pi$. We informally refer to $|X(\omega_d)|$ also as the *spectrum* of $x[n]$.

The *inverse discrete-time Fourier transform (IDTFT)* is found by observing that Eq. (1.53) represents the Fourier series of the periodic function $X(\omega_d)$. As a consequence, one can apply Fourier theory for periodic functions of continuous variables, and compute the Fourier coefficients $x[n]$ as

$$\mathcal{F}^{-1}\{X\}[n] = x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega_d) e^{j\omega_d n} d\omega_d. \quad (1.55)$$

Equations (1.53) and (1.55) together form a Fourier representation for the sequence $x[n]$. Equation (1.55) can be regarded as a *synthesis* formula, since it represents $x[n]$ as a superposition of infinitesimally small complex sinusoids, with $X(\omega_d)$ determining the relative amount of each sinusoidal component. Equation (1.53) can be regarded as an *analysis* formula, since it provides an expression for computing $X(\omega_d)$ from the sequence $x[n]$ and determining its sinusoidal components.

1.4.1.2 DTFT of common sequences

We can apply the DTFT definition to some of the sequences that we have examined. The DTFT of the unit impulse $\delta[n]$ is the constant 1:

$$\mathcal{F}\{\delta\}(\omega_d) = \sum_{n=-\infty}^{+\infty} \delta[n] e^{-j\omega_d n} = 1. \quad (1.56)$$

The unit step sequence $u[n]$ does not have a DTFT, because the sum in Eq. (1.53) takes infinite values. The exponential sequence (1.9) also does not admit a DTFT. However if we consider the *right sided exponential sequence* $x[n] = a^n u[n]$, in which the unit step is multiplied by an exponential with $|a| < 1$, then this admits a DTFT:

$$\mathcal{F}\{x\}(\omega_d) = \sum_{n=-\infty}^{+\infty} a^n u[n] e^{-j\omega_d n} = \sum_{n=0}^{+\infty} (ae^{-j\omega_d})^n = \frac{1}{1 - ae^{-j\omega_d}}. \quad (1.57)$$



Property	Time-domain sequences	Frequency-domain DTFTs
	$x[n], y[n]$	$X(\omega_d), Y(\omega_d)$
Linearity	$ax[n] + by[n]$	$aX(\omega_d) + bY(\omega_d)$
Time-shifting	$x[n - n_0]$	$e^{-j\omega_d n_0} X(\omega_d)$
Frequency-shifting	$e^{j\omega_0 n} x[n]$	$X(\omega_d - \omega_0)$
Frequency differentiation	$nx[n]$	$j \frac{dX}{d\omega_d}(\omega_d)$
Convolution	$(x * y)[n]$	$X(\omega_d) \cdot Y(\omega_d)$
Multiplication	$x[n] \cdot y[n]$	$\frac{1}{2\pi} \int_{-\pi}^{\pi} X(\theta)Y(\omega_d - \theta)d\theta$
Parseval relation	$\sum_{n=-\infty}^{+\infty} x[n]y^*[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega_d)Y^*(\omega_d)d\omega_d$	

Table 1.1: General properties of the discrete-time Fourier transform.

The complex exponential sequence $x[n] = e^{j\omega_0 n}$ or the real sinusoidal sequence $x[n] = \cos(\omega_0 n + \phi)$ are other examples of sequences that do not have a DTFT, because the sum in Eq. (1.53) takes infinite values. In general a sequences does not necessarily admit a Fourier representation, meaning with this that the series in Eq. (1.53) may not converge. One can show that $x[n]$ being absolutely summable (we have defined absolute summability in Eq. (1.32)) is a sufficient condition for the convergence of the series (recall the definition of absolute summability given in Eq. (1.32)). Note that an absolutely summable sequence has always finite energy, and that the opposite is not always true, since $\sum |x[n]|^2 \leq (\sum |x[n]|)^2$. Therefore a finite-energy sequence does not necessarily admit a Fourier representation.²

1.4.1.3 Properties

Table 1.1 lists a number of properties of the DTFT which are useful in digital signal processing applications. Time- and frequency-shifting are interesting properties in that they show that a shifting operation in either domain correspond to multiplication for an complex exponential function in the other domain. Proof of these properties is straightforward from the definition of DTFT.

The convolution property is extremely important: it says that a convolution in the time domain becomes a simple multiplication in the frequency domain. This can be demonstrated as follows:

$$\begin{aligned} \mathcal{F}\{x * y\}(\omega_d) &= \sum_{n=-\infty}^{+\infty} \left(\sum_{k=-\infty}^{+\infty} x[k]y[n-k] \right) e^{-j\omega_d n} = \sum_{m=-\infty}^{+\infty} \sum_{k=-\infty}^{+\infty} x[k]y[m]e^{-j\omega_d(k+m)} \\ &= \sum_{k=-\infty}^{+\infty} x[k]e^{-j\omega_d k} \cdot \sum_{m=-\infty}^{+\infty} y[m]e^{-j\omega_d m}, \end{aligned} \quad (1.58)$$

where in the second equality we have substituted $m = n - k$. The multiplication property is dual to the convolution property: a multiplication in the time-domain becomes a convolution in the frequency domain.

The Parseval relation is also very useful: if we think of the sum on the left-hand side as an inner product between the sequences x and y , we can restate this property by saying that the DTFT preserves

²For non-absolutely summable sequences like the unit step or the sinusoidal sequence, the DTFT can still be defined if we resort to the Dirac delta $\delta_D(\omega_d - \omega_0)$. Since this is not a function but rather a distribution, extending the DTFT formalism to non-summable sequences requires to dive into the theory of distributions, which we are not willing to do.



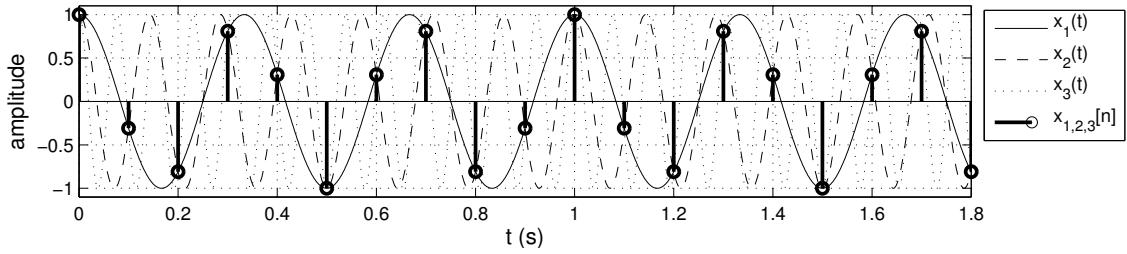


Figure 1.7: Example of frequency aliasing occurring for three sinusoids.

the inner product (apart from the scaling factor $1/(2\pi)$). In particular, when $x = y$, it preserves the energy of the signal x . The Parseval relation can be demonstrated by noting that the DTFT of the sequence $y^*[-n]$ is $Y^*(\omega_d)$. Then we can write:

$$\mathcal{F}^{-1}\{XY^*\}[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega_d)Y^*(\omega_d)e^{j\omega_d n} d\omega_d = \sum_{k=-\infty}^{+\infty} x[k]y^*[k-n], \quad (1.59)$$

where in the first equality we have simply used the definition of the IDTFT, while in the second equality we have exploited the convolution property. Evaluating this expression for $n = 0$ proves the Parseval relation.

1.4.2 The sampling problem

1.4.2.1 Frequency aliasing

With the aid of the DTFT machinery, we can now go back to the concept of “sampling” and introduce some fundamental notions. Let us start with an example.

Consider three continuous-time sinusoids $x_i(t)$ ($i = 1, 2, 3$) defined as

$$x_1(t) = \cos(6\pi t), \quad x_2(t) = \cos(14\pi t), \quad x_3(t) = \cos(26\pi t). \quad (1.60)$$

These sinusoids have frequencies 3, 7, and 13 Hz, respectively. Now we construct three sequences $x_i[n] = x_i(n/F_s)$ ($i = 1, 2, 3$), each obtained by sampling one of the above signals, with a sampling frequency $F_s = 10$ Hz. We obtain the sequences

$$x_1[n] = \cos(0.6\pi n), \quad x_2[n] = \cos(1.4\pi n), \quad x_3[n] = \cos(2.6\pi n). \quad (1.61)$$

Figure 1.7 shows the plots of both the continuous-time sinusoids and the sampled sequences: note that all sequences have exactly the same sample values for all n , i.e. they actually *are* the same sequence. This phenomenon of a higher frequency sinusoid acquiring the identity of a lower frequency sinusoid after being sampled is called *frequency aliasing*.

In fact we can understand the aliasing phenomenon in a more general way using the Fourier theory. Consider a continuous-time signal $x(t)$ and its sampled version $x_d[n] = x(nT_s)$. Then we can prove that the Fourier Transform $X(\omega)$ of $x(t)$ and the DTFT $X_d(\omega_d)$ of $x_d[n]$ are related via the following equation:

$$X_d(\omega_d) = F_s \sum_{m=-\infty}^{+\infty} X(\omega_d F_s + 2m\pi F_s). \quad (1.62)$$



This equation tells a fundamental result: $X_d(\omega_d)$ is a periodization of $X(\omega)$, i.e. it is a periodic function (of period 2π) made of a sum of shifted and scaled replicas of $X(\omega)$. The terms of this sum for $m \neq 0$ are *aliasing terms* and are said to alias into the so-called *base band* $[-\pi F_s, \pi F_s]$. Therefore if two continuous-time signals $x_1(t)$, $x_2(t)$ have Fourier transforms with the property $X_2(\omega) = X_1(\omega + 2m\pi F_s)$ for some $m \in \mathbb{Z}$, sampling these signals will produce identical DTFTs and therefore identical sequences. This is the case of the sinusoids in Eq. (1.61).

In the remainder of this section we provide a proof of Eq. (1.62). We first write $x(t)$ and $x_d[n]$ in terms of their Fourier transforms:

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(\omega) e^{j\omega t} d\omega, \quad x_d[n] = \frac{1}{2\pi} \int_{-\pi}^{+\pi} X_d(\omega_d) e^{j\omega_d n} d\omega_d. \quad (1.63)$$

The first integral can be broken up into an infinite sum of integrals computed on the disjoint intervals $[(2m - 1)\pi F_s, (2m + 1)\pi F_s]$, (with $m \in \mathbb{Z}$) each of length $2\pi F_s$. Then

$$\begin{aligned} x(t) &= \frac{1}{2\pi} \sum_{m=-\infty}^{+\infty} \int_{(2m-1)\pi F_s}^{(2m+1)\pi F_s} X(\omega) e^{j\omega t} d\omega \\ &= \frac{1}{2\pi} \int_{-\pi F_s}^{\pi F_s} e^{j\theta t} \sum_{m=-\infty}^{+\infty} X(\theta + 2m\pi F_s) e^{j2m\pi F_s t} d\theta, \end{aligned} \quad (1.64)$$

where in the second equality we have substituted $\omega = \theta + 2m\pi F_s$ in the integral, and we have swapped the integral and the series. If we sample this representation to obtain $x_d[n] = x(nT_s)$, we can write

$$\begin{aligned} x_d[n] = x(nT_s) &= \frac{1}{2\pi} \int_{-\pi F_s}^{\pi F_s} e^{j\theta n T_s} \sum_{m=-\infty}^{+\infty} X(\theta + 2m\pi F_s) e^{j2m\pi F_s n T_s} d\theta \\ &= \frac{1}{2\pi} \int_{-\pi F_s}^{\pi F_s} e^{j\theta n T_s} \left(\sum_{m=-\infty}^{+\infty} X(\theta + 2m\pi F_s) \right) d\theta, \end{aligned} \quad (1.65)$$

because the exponentials inside the sum are all equal to 1 ($e^{j2m\pi F_s n T_s} = e^{j2nm\pi} = 1$). If we finally substitute $\omega_d = \theta T_s$ we obtain

$$x_d[n] = \frac{F_s}{2\pi} \int_{-\pi}^{+\pi} e^{j\omega_d n} \left(\sum_{m=-\infty}^{+\infty} X(\omega_d F_s + 2m\pi F_s) \right) d\omega_d, \quad (1.66)$$

which proves Eq. (1.62).

1.4.2.2 The sampling theorem and the Nyquist frequency

Consider the three cases depicted in Fig. 1.8. The magnitude of the Fourier transform in Fig. 1.8(a) (upper panel) is zero everywhere outside the base band, and Eq. (1.62) tells us that the magnitude of the sampled signal looks like the plot in the lower panel. In Fig. 1.8(b) (upper panel) we have a similar situation except that the magnitude is non-zero in the band $[\pi F_s, 3\pi F_s]$. The magnitude of the corresponding sampled signal then looks like the plot in the lower panel, and is identical to the one in Fig. 1.8(a). Yet another situation is depicted in Fig. 1.8(c) (upper panel): in this case we are using a smaller sampling frequency F_s , so that the magnitude now extends to more than one band. As a result the shifted replicas of $|X|$ overlap and $|X_d|$ is consequently distorted.

These examples suggest that a “correct” sampling of a continuous signal $x(t)$ corresponds to the situation of Fig. 1.8(a), while for the cases depicted in Figs. 1.8(b) and 1.8(c) we lose information about



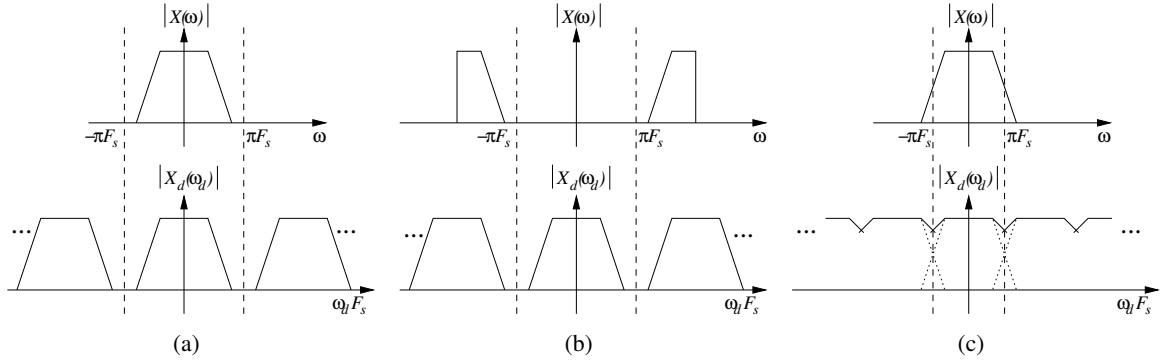


Figure 1.8: Examples of sampling a continuous time signal: (a) spectrum limited to the base band; (b) the same spectrum shifted by 2π ; (c) spectrum larger than the base band.

the original signal. The *sampling theorem* formalizes this intuition by saying that $x(t)$ can be exactly reconstructed from its samples $x[n] = x(nT_s)$ if and only if $X(\omega) = 0$ outside the base band (i.e. for all $|\omega| \geq \pi/F_s$). The frequency $f_{Ny} = F_s/2$ Hz, corresponding to the upper limit of the base band, is called *Nyquist frequency*.

Based on what we have just said, when we sample a continuous-time signal we must choose F_s in such a way that the Nyquist frequency is above any frequency of interest, otherwise frequencies above f_{Ny} will be aliased. In the case of audio signals, we know from psychoacoustics that humans perceive audio frequencies up to ~ 20 kHz: therefore in order to guarantee that no artifacts are introduced by the sampling procedure we must use $F_s > 40$ kHz, and in fact the most diffused standard is $F_s = 44.1$ kHz. In some specific cases we may use lower sampling frequencies: as an example it is known that the spectrum of a speech signal is limited to ~ 4 kHz, and accordingly the most diffused standard in telephony is $F_s = 8$ kHz.

In the remainder of this section we sketch the proof of the sampling theorem. If $X(\omega) \neq 0$ only in the base band, then all the sum terms in Eq. (1.62) are 0 except for the one with $m = 0$. Therefore

$$X_d(\omega_d) = F_s X(\omega_d F_s) \quad \text{for } \omega_d \in (-\pi, \pi). \quad (1.67)$$

In order to reconstruct $x(t)$ we can take the inverse Fourier Transform:

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(\omega) e^{j\omega t} d\omega = \frac{1}{2\pi} \int_{-\pi F_s}^{\pi F_s} X(\omega) e^{j\omega t} d\omega = \frac{1}{2\pi F_s} \int_{\pi F_s}^{+\pi F_s} X_d\left(\frac{\omega}{F_s}\right) e^{j\omega t} d\omega. \quad (1.68)$$

where in the second equality we have exploited the hypothesis $X \equiv 0$ outside the base band and in the third one we have used Eq. (1.67). If we now reapply the definition of the DTFT we obtain

$$x(t) = \frac{1}{2\pi F_s} \int_{-\pi F_s}^{\pi F_s} \left[\sum_{n=-\infty}^{+\infty} x(nT_s) e^{-j\omega T_s n} \right] e^{j\omega t} d\omega = \sum_{n=-\infty}^{+\infty} \frac{x(nT_s)}{2\pi F_s} \int_{-\pi F_s}^{\pi F_s} e^{j\omega(t-nT_s)} d\omega, \quad (1.69)$$

where in the second equality we have swapped the sum with the integral. Now look at the integral on the right hand side. We can solve it explicitly and write

$$\begin{aligned} \frac{1}{2\pi F_s} \int_{-\pi F_s}^{\pi F_s} e^{j\omega(t-nT_s)} d\omega &= \frac{1}{2\pi F_s} \frac{2}{2j(t-nT_s)} \left[e^{j\pi F_s(t-nT_s)} - e^{-j\pi F_s(t-nT_s)} \right] = \\ &= \frac{\sin[\pi F_s(t-nT_s)]}{\pi F_s(t-nT_s)} = \text{sinc}[F_s(t-nT_s)]. \end{aligned} \quad (1.70)$$



That is, the integral is a *cardinal sine* function, defined as $\text{sinc}(t) \triangleq \sin(\pi t)/\pi t$ (the use of π in the definition has the effect that the sinc function has zero crossings on the non-zero integers). In conclusion, we can rewrite Eq. (1.68) as

$$x(t) = \sum_{n=-\infty}^{+\infty} x(nT_s) \text{sinc}\left(\frac{t}{T_s} - n\right). \quad (1.71)$$

We have just proved that if $X \equiv 0$ outside the base band then $x(t)$ can be reconstructed from its samples through Eq. (1.71). The opposite implication is obvious: if $x(t)$ can be reconstructed through its samples it must be true that $X \equiv 0$ outside the base band, since a sampled signal only supports frequencies up to f_{Ny} by virtue of Eq. (1.62).

1.5 Short-time Fourier analysis

In these section we introduce the most common spectral analysis tool: the *Short Time Fourier Transform (STFT)*. Sounds are time-varying signals, therefore, it is important to develop analysis techniques to inspect some of their time-varying features. The STFT allows joint analysis of the temporal and frequency features of the sound signal, in other words it allows to follow the temporal evolution of the spectral parameters of a sound. The main building block of the STFT is the *Discrete Fourier Transform (DFT)*, which can be thought as a specialization of the DTFT for sequences of finite length.

1.5.1 The Discrete Fourier Transform

The *Discrete Fourier Transform (DFT)* is a special case of the DTFT applied to finite-length sequences. As such it is a useful tool for representing periodic sequences. as we said in the previous section, a periodic sequence does not have a DTFT in a strict sense. However periodic sequences are in one-to-one correspondence with finite-length sequences, meaning with this that a finite-length sequence can be taken to represent a period of a periodic sequence.

1.5.1.1 Definitions and properties

The *Discrete Fourier Transform (DFT)* is a special case of the DTFT applied to finite-length sequences $x[n]$ with $0 \leq n \leq N - 1$. Let us consider one such sequence: we define the DFT of $x[n]$ as the sequence $X[k]$ obtained by uniformly sampling the DTFT $X(\omega_d)$ on the ω_d -axis between $0 \leq \omega_d < 2\pi$, at points at $\omega_k = 2\pi k/N$, $0 \leq k \leq N - 1$. If $x[n]$ has been sampled from a continuous-time signal, i.e. $x[n] = x(nT_s)$, the points ω_k correspond to the frequency points $f_k = kF_s/N$ (in Hz).

From Eq. (1.53) one can then write

$$X[k] = X(\omega_d)|_{\omega_d=2\pi k/N} = \sum_{n=0}^{N-1} x[n] e^{-j \frac{2\pi}{N} kn} = \sum_{n=0}^{N-1} x[n] W_N^{kn}, \quad 0 \leq k \leq N - 1 \quad (1.72)$$

where we have used the notation $W_N = e^{-j2\pi/N}$. Note that the DFT is also a finite-length sequence in the frequency domain, with length N . The *inverse discrete Fourier Transform (IDFT)* is given by

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-kn}, \quad 0 \leq n \leq N - 1. \quad (1.73)$$



This relation can be verified by multiplying both sides by W_N^{ln} , with l integer, and summing the result from $n = 0$ to $n = N - 1$:

$$\sum_{n=0}^{N-1} x[n] W_N^{ln} = \frac{1}{N} \sum_{n=0}^{N-1} \sum_{k=0}^{N-1} X[k] W_N^{-(k-l)n} = \frac{1}{N} \sum_{k=0}^{N-1} X[k] \left[\sum_{n=0}^{N-1} W_N^{-(k-l)n} \right], \quad (1.74)$$

where the last equality has been obtained by interchanging the order of summation. Now, the summation $\sum_{n=0}^{N-1} W_N^{-(k-l)n}$ has the interesting property that it takes the value N when $k - l = rN$ ($r \in \mathbb{Z}$), and takes the value 0 for any other value of k and l . Therefore Eq. (1.74) reduces to the definition of the DFT, and therefore Eq. (1.73) is verified.

We have just proved that, for a length- N sequence $x[n]$, the N values of its DTFT $X(\omega_d)$ at points $\omega_d = \omega_k$ are sufficient to determine $x[n]$, and hence $X(\omega_d)$, uniquely. This justifies our definition of Discrete Fourier Transform of finite-length sequences given in Eq. (1.72). The DFT is at the heart of digital signal processing, because it is a *computable* transformation.

Most of the DTFT properties listed in Table 1.1 have a direct translation for the DFT. Clearly the DFT is linear. The time- and frequency-shifting properties still correspond to a multiplication by a complex number, however these properties becomes periodic with period N . As an example, the time shifting properties for the DFT becomes

$$x_m[n] = x[n - m] \Rightarrow X_m[k] = W_N^{km} X[k]. \quad (1.75)$$

Clearly any shift of $m + lN$ samples cannot be distinguished from a shift by m samples, since $W_N^{km} = W_N^{k(m+lN)}$. In other words, the ambiguity of the shift in the time domain has a direct counterpart in the frequency domain.

The convolution property also holds for the DFT and is stated as follows:

$$z[n] = (x * y)[n] \triangleq \sum_{m=0}^{N-1} x[n] y[n - m] \Rightarrow Z[k] = (X \cdot Y)[k], \quad (1.76)$$

where in this case the symbol $*$ indicates the *periodic convolution*. The proof of this property is similar to the one given for the DTFT.

1.5.1.2 Resolution, leakage and zero-padding

Consider the complex exponential sequence $x[n] = e^{j\omega_0 n}$ over a finite number of points $0 \leq n < N$. In Sec. 1.2 we have already shown that this sequence is periodic over the interval $[0, N]$ only if $\omega_0 N = 2\pi k$ for some integer k . This implies that there are exactly N periodic complex exponential sequences representable with N samples, i.e. those for which $\omega_0 = 2\pi k_0/N$, with $k_0 = 0 \dots N - 1$: these are the sequences $x[n] = e^{j2\pi k_0 n/N} = W_N^{-k_0 n}$. From the definitions of the DFT and the IDFT it follows immediately that $X[k] = \delta(k - k_0)$, i.e. the DFT associated to the sequence $W_N^{-k_0 n}$ takes the value 1 for $k = k_0$ and is zero everywhere else. As an example, Fig. 1.9(a) shows a 64-points DFT (computed numerically) for the sequence $x[n] = W_N^{-k_0 n}$ for $k_0 = 20$.

Since the *resolution* of the DFT is limited by the number of DFT points, one may wonder how the DFT looks like for complex exponential sequences that are *not* periodic over the interval $[0, N]$. An example is shown in Fig. 1.9(b) for the complex exponential sequence with $\omega_0 = 2\pi k_0/N$, where we have used a non-integer value $k_0 = 20.5$ and $N = 64$. Although the value of ω_0 is very similar to the one in Fig. 1.9(a), the DFT looks very different. This happens because we have chosen a value for ω_0 that falls in the crack between two DFT points, and consequently the DFT does a poor job in resolving



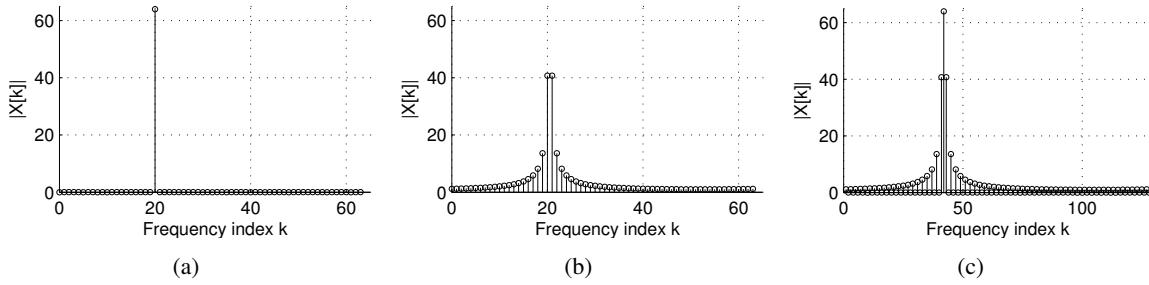


Figure 1.9: Examples of DFT applied to complex exponential sequences: (a) $N = 64$, $k_0 = 20$, the sequence is periodic and the DFT is a delta-sequence in the frequency domain; (b) $N = 64$, $k_0 = 20.5$, the sequence is not periodic and the DFT exhibits leakage; (c) $N = 128$, $k_0 = 41$, the sequence is the windowed exponential of Eq. (1.77) and the DFT is a shifted version of the rectangular window DFT.

the frequency of this particular signal. We call this effect *leakage*: since ω_0 does not line up with one of the “allowed” frequencies, the energy of the DFT leaks all over the base band.

In order to understand the leakage effect we now look at a third example. Figure 1.9(c) depicts the DFT of the sequence

$$x[n] = \begin{cases} e^{j2\pi k_0 n/N}, & 0 \leq n < \frac{N}{2}, \\ 0, & \frac{N}{2} \leq n < N, \end{cases} \quad (1.77)$$

with $k_0 = 41$ and $N = 128$. In other words, the sequence $x[n]$ is constructed by taking the complex exponential sequence $e^{j2\pi k_0 n/N}$ over 64 points, and by *zero-padding* this sequence over the remaining 64 points. Note that the complex exponential sequence of this example is clearly the same that we have considered in Fig. 1.9(b) (we have simply doubled the values of k_0 and N , so that ω_0 has the same value), and is clearly periodic over $N = 128$ points. Note also that the DFTs of Fig. 1.9(b) and 1.9(c) are identical, except that the one in Fig. 1.9(c) has twice the points and consequently a better resolution in frequency.

The sequence $x[n]$ of Eq. (1.77) can be also written as

$$x[n] = e^{j2\pi k_0 n/N} \cdot w_{N/2}[n] = W_N^{-k_0 n} w_{N/2}[n], \quad \text{where } w_{N/2}[n] = \begin{cases} 1, & 0 \leq n < \frac{N}{2}, \\ 0, & \frac{N}{2} \leq n < N, \end{cases} \quad (1.78)$$

where $w_{N/2}[n]$ is a rectangular window of length $N/2$. More in general, we call $w_M[n]$ a rectangular window of length M .

The advantage of this representation is that we can now compute explicitly the DFT $X[k]$, since we know that multiplying by $W_N^{-k_0 n}$ in time corresponds to shifting by k_0 samples in frequency. Therefore $X[k]$ equals the DFT of $w_{N/2}[h]$, shifted by k_0 samples. The DFT of the generic rectangular window $w_M[n]$ can be computed from the definition as

$$\mathcal{F}\{w_M\}[k] = \sum_{n=0}^{N-1} w_M[n] W_N^{kn} = \sum_{n=0}^{M-1} W_N^{kn} = \frac{1 - W_N^{kM}}{1 - W_N^k} = e^{-j\frac{\pi k}{N}(M-1)} \frac{\sin(\pi \frac{kM}{N})}{\sin(\pi \frac{k}{N})}, \quad (1.79)$$

where in the third equality we have used the property of the geometric series $\sum_{k=0}^{M-1} q^k = (1-q^M)/(1-q)$, and in the fourth equality we have applied some straightforward trigonometry. Figure 1.10 shows three plots of this DFT, for different window lengths M . In particular Fig. 1.10(c) shows the case

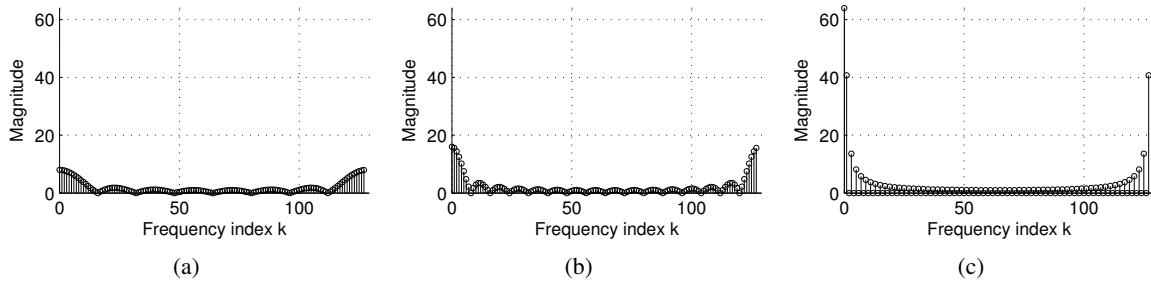


Figure 1.10: Examples of DFT ($N = 128$) applied to a rectangular window: (a) $M = N/16$, (b) $M = N/8$, (c) $M = N/2$.

$M = N/2$: note that this plot coincides with the one in Fig. 1.9(c), except for a shift in frequency, which is what we expected.

To summarize, in this section we have shown that (a) the frequency resolution of the DFT is limited by the number N of DFT points, (b) the DFT of a complex exponential sequence which is not periodic over N points exhibits poor resolution and leakage, and (c) the leakage effect can be interpreted as the effect of a rectangular window of length N applied to the sequence.

M-1.9

Compute the DFT of complex exponential sequence $x[n] = e^{j2\pi k_0 n/N}$ for integer and non-integer values of k_0 , in order to explore leakage effects. Then compute the DFT of zero-padded complex exponential sequences, in order to see how frequency resolution and leakage are affected.

1.5.1.3 Fast computation of the DFT: the FFT algorithm

A brute-force approach to DFT computation has $\mathcal{O}(N^2)$ running time, since we need to perform $\mathcal{O}(N)$ multiplications and additions to compute each of the N DFT samples: we need more efficient solutions to the problem. The *Fast Fourier Transform (FFT)* algorithm provides the most efficient computation of the DFT, as it runs in $\mathcal{O}(N \log N)$ times. The algorithm is based on a divide-and-conquer strategy, that allows to compute the DFT of $x[n]$ using the DFTs of two half-length subsequences of $x[n]$. Additionally it exploits some nice properties of the N -th roots of unity W_N^k .

Assume for simplicity that the sequence $x[n]$ has length $N = 2^s$ for some $s \in \mathbb{N}$. Then we can write the DFT sequence $X[k]$ as

$$\begin{aligned}
X[k] &= \sum_{n=0}^{\frac{N}{2}-1} x[2n]W_N^{2kn} + \sum_{n=0}^{\frac{N}{2}-1} x[2n+1]W_N^{k(2n+1)} \\
&= \sum_{n=0}^{\frac{N}{2}-1} x[2n]W_N^{2kn} + W_N^k \sum_{n=0}^{\frac{N}{2}-1} x[2n+1]W_N^{2kn} \\
&= \sum_{n=0}^{\frac{N}{2}-1} x[2n]W_{N/2}^{kn} + W_N^k \sum_{n=0}^{\frac{N}{2}-1} x[2n+1]W_{N/2}^{kn},
\end{aligned} \tag{1.80}$$

where in the first equality we have separated the terms involving even elements and odd elements, in the second one we have factorized a term W_N^k , and in the third one we have exploited the property

$W_{aN}^{ak} = W_N^k$ for any $a \in \mathbb{N}$. If we now look at the two sums in the last row, we see that they are the DFTs of the sequences $x_0[n] = \{x[0], x[2], \dots, x[N-2]\}$ and $x_1[n] = \{x[1], x[3], \dots, x[N-1]\}$, respectively. Both $x_0[n]$ and $x_1[n]$ have length $N/2$. Therefore the problem of computing $X[k]$ reduces to the problem of computing two half-length DFTs $X_0[k]$, $X_i[k]$, and then summing their values according to Eq. (1.80). The resulting computational procedure is detailed in Algorithm 1.1. It is quite obvious that the running time $T(N)$ of this algorithm is given by the recurrence equation $T(N) = 2T(N/2) + \mathcal{O}(N)$, therefore $T(N) = \mathcal{O}(N \log N)$.

Algorithm 1.1: RECURSIVE-FFT(x)

```

1  $N \leftarrow \text{length}(x)$  //  $N$  is a power of 2
2
3 if  $n = 1$  then return  $x$   $W_N \leftarrow e^{-2\pi j/N}$ 
4  $W \leftarrow 1$ 
5  $x_0[n] \leftarrow \{x[0], x[2], \dots, x[N-2]\}$ 
6  $x_1[n] \leftarrow \{x[1], x[3], \dots, x[N-1]\}$ 
7  $X_0[k] \leftarrow \text{RECURSIVE-FFT}(x_0)$ 
8  $X_1[k] \leftarrow \text{RECURSIVE-FFT}(x_1)$ 
9 for  $k \leftarrow 0$  to  $N/2 - 1$  do
10    $X[k] \leftarrow X_0[k] + W X_1[k]$ 
11    $X[k + N/2] \leftarrow X_0[k] - W X_1[k]$ 
12    $W \leftarrow W \cdot W_N$ 
13 return  $X$ 
```

M-1.10

Realize Algorithm 1.1 and assess its functioning by comparing it with the `fft` function.

1.5.1.4 Iterative FFT algorithms and parallel realizations

Note that in writing the last cycle of Algorithm 1.1 we have exploited a relevant property of the W_N^k coefficients, namely $W_N^{(k+N/2)} = -W_N^k$. Thanks to this property the value $WX_1[k]$ is used twice (it is a *common subexpression*): first it is added to $X_0[k]$ to compute $X[k]$, then it is subtracted from $X_0[k]$ to compute $X[k + N/2]$. This is known as *butterfly operation*, and is the key element in the construction of a more efficient, iterative implementation of the FFT algorithm.

Figure 1.11(a) depicts the tree of calls in an invocation of the recursive algorithm, in the case $N = 8$. Looking at the tree we observe that, if we could arrange the elements in the order in which they appear in the leaves, we could compute the DFT as follows: first we take the elements in pairs and combine each pair with one butterfly operation, thus obtaining four 2-element DFTs; second, we take these DFTs in pairs and combine each pair with two butterfly operations, thus obtaining two 4-element DFTs; finally, we combine these two DFTs with four butterfly operations, thus obtaining the final 8-element DFT. The resulting scheme is an iterative FFT implementation.

The only problem left is how to arrange the elements in the order in which they appear in the leaves. Luckily the solution is straightforward: this order is a *bit-reversal permutation*, that is $x[n]$ is placed in the position obtained by reversing the bits of the binary representation of n . We can then write Algorithm 1.2. Clearly the algorithm is still $\mathcal{O}(N \log N)$, since the total number of butterfly operations is $\mathcal{O}(N \log N)$, and since the bit-reversal permutation is also a $\mathcal{O}(N \log N)$ procedure (we have to reverse N integers,



each made of $\log N$ bits). Figure 1.11(b) shows an efficient parallel implementation of this algorithm: it is made of $\log N$ stages, each performing $N/2$ butterfly operations.

Algorithm 1.2: ITERATIVE-FFT(x)

```

1 BIT-REVERSE-COPY( $x, X$ )
2  $N \leftarrow \text{length}(x)$ 
3 for  $s \leftarrow 1$  to  $\log_2(N)$  do
4    $m \leftarrow 2^s$ 
5    $W_m \leftarrow e^{-2\pi j/m}$ 
6   for  $k \leftarrow 0$  to  $N - 1$  by  $m$  do
7      $W \leftarrow 1$ 
8     for  $l \leftarrow 0$  to  $m/2 - 1$  do
9        $t \leftarrow W X[k + l + m/2]$ 
10       $u \leftarrow X[k + l]$ 
11       $X[k + l] \leftarrow u + t$ 
12       $X[k + l + m/2] \leftarrow u - t$ 
13      $W \leftarrow W \cdot W_m$ 
14 return  $X$ 
```

M-1.11

Realize Algorithm 1.2 and assess its functioning by comparing it with the `fft` function.

1.5.2 The Short-Time Fourier Transform

1.5.2.1 Definition and examples

$$X_n(\omega_d) = \sum_{m=-\infty}^{+\infty} w[n-m]x[m]e^{-j\omega_d m} \quad (1.81)$$

If $w \equiv 1$, then this equation reduces to the DTFT of $x[n]$. However in practical applications we are interested in using finite-length windows, in order to analyze the spectral properties of $x[n]$ over a finite time interval. In such applications what we really do is computing, for each n , the DFT of the finite-length sequence $w[n-m]x[m]$, and what we obtain is a finite-length sequence $X_n[k] = X_n(\omega_d)|_{\omega_d=2\pi k/N}$.

$X_n(\omega_d)$ is the DTFT of the sequence $x_n[m] = w[n-m]x[m]$, therefore

$$w[n-m]x[n] = \frac{1}{2\pi} \int_{-\pi}^{+\pi} X_n(\omega_d)e^{j\omega_d m} d\omega_d, \quad (1.82)$$

from which, when $n = m$ and in the hypothesis of $x[0] \neq 0$

$$x[n] = \frac{1}{2\pi w[0]} \int_{-\pi}^{+\pi} X_n(\omega_d)e^{j\omega_d n} d\omega_d. \quad (1.83)$$

This equation shows that the sequence x can be reconstructed from its STFT.

The magnitude of STFT is called *spectrogram*. Since STFT is function of two variables (i.e., time n and frequency ω_d), the plot of the spectrogram lives in a 3-D space. A typical 2-D representation of



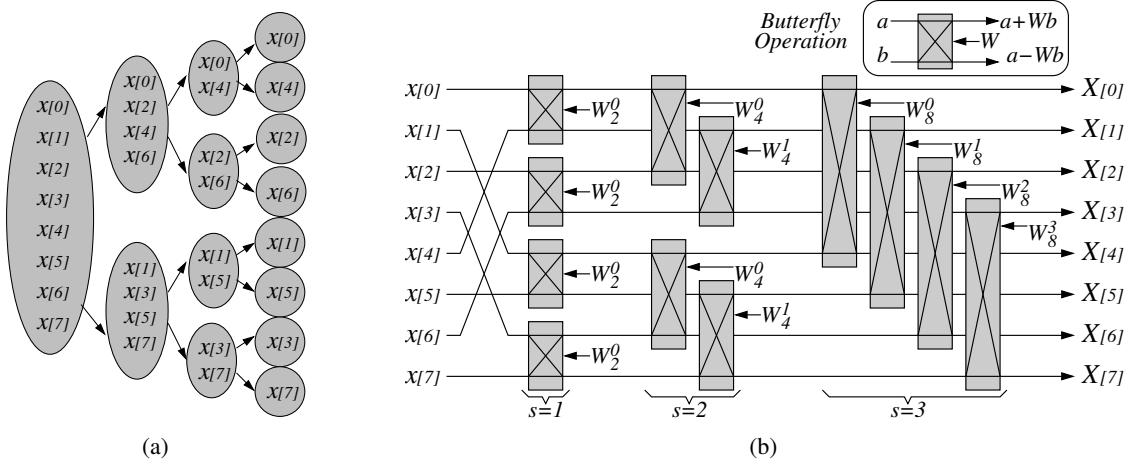


Figure 1.11: (a) Tree of calls in an invocation of RECURSIVE-FFT: the leaves contain a bit-reversal permutation of $x[n]$; (b) parallel realization of ITERATIVE-FFT, where butterfly operations involve bit-reversed elements of $x[n]$.

a spectrogram uses two axes for time and frequency, and magnitude values are represented with some color map (e.g. a greyscale map).

Figure 1.12 shows an example of short-time spectral analysis applied to a simple time-varying signal, the *chirp*. The chirp sequence is defined as

$$x[n] = A \cos \left[\frac{\omega_0}{2} (nT_s)^2 \right]. \quad (1.84)$$

We have already seen in Sec. 1.3.1 that in general the instantaneous frequency of a signal $\cos[\phi(t)]$ is $d\phi/dt(t)$: therefore the instantaneous frequency of this chirp signal is $\omega_0 n T_s$, i.e. it is not constant but increases linearly in time. A portion of a chirp signal with $\omega_0 = 2\pi \cdot 800 \text{ rad/s}^2$ is shown in Fig. 1.12(a). Now we segment this signal into a set of subsequences with short finite length, e.g. using a rectangular window $w[n]$, as in Eq. (1.81). If the window is sufficiently short, we can reasonably assume that the frequency of the chirp is approximately constant in each subsequence. In fact the resulting spectrogram, shown in Fig. 1.12(b), shows that for a given time index n the STFT is essentially the DFT of a sinusoidal sequence: the STFT magnitude has large non-zero values around the instantaneous frequency, and much smaller non-zero values at other frequency points. Moreover the instantaneous frequency increases linearly and after 5 seconds it reaches the value 4000 Hz ($= 800 \cdot 5 \text{ Hz}$), as expected.

So far in this example we have implicitly assumed that the sampling period T_s is sufficiently small, so that no aliasing occurs: in fact in drawing Fig. 1.12(b) we have used a sampling rate $F_s = 8 \text{ kHz}$. However, aliasing will occur if we use smaller sampling rates. Figure 1.12(c) shows the spectrogram obtained using $F_s = 4 \text{ kHz}$: in this case frequencies above $f_{Ny} = 2 \text{ kHz}$ are aliased, and this effect appears in the spectrogram when the black line starts moving down instead of increasing.

M-1.12

Synthesize a chirp signal and compute its STFT using different sampling rates, in order to verify the occurrence of aliasing.



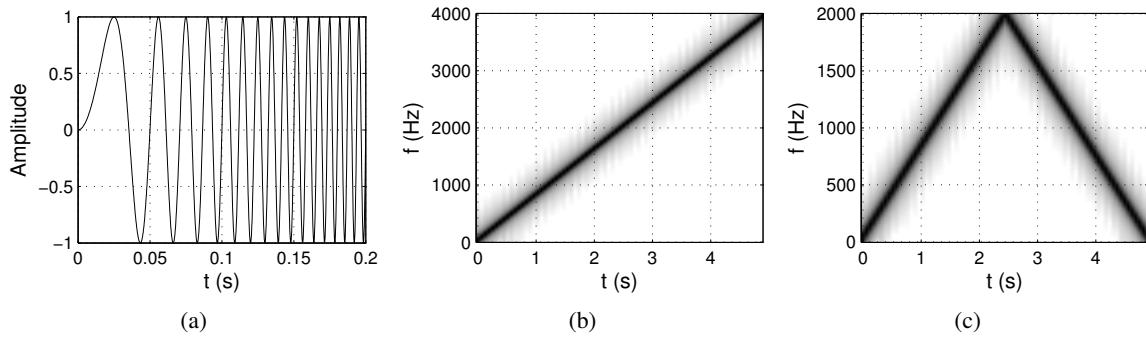


Figure 1.12: Short-time spectral analysis of a chirp signal: (a) initial portion of the chirp signal, with $\omega_0 = 2\pi \cdot 800 \text{ rad/s}^2$; (b) spectrogram obtained with $F_s = 8 \text{ kHz}$; (c) spectrogram obtained with $F_s = 4 \text{ kHz}$.

1.5.2.2 Windowing and the uncertainty principle

We have previously examined the resolution and leakage problems associated to the DFT: decreased frequency resolution and spectral energy leakage occur because the spectrum is convolved with that of a rectangular window. As the width of the rectangular window increases (see Fig. 1.10), the energy of its spectrum becomes more and more concentrated around the origin. In the limit, the spectrum of an infinite-width rectangular window is a $\delta[k]$ sequence, and no leakage occurs.

This qualitative analysis provides an example of the so-called *uncertainty principle*. Increasing resolution in frequency diminishes resolution in time, and viceversa. Although a certain trade-off between time resolution and frequency resolution is inevitable (and determined by the window length), one may wonder if such a trade-off can be improved by using windows with different shapes (and thus different spectra) from the rectangular window.

In fact the answer is yes. Some of the most commonly used window functions are listed below:

$$\begin{aligned}
 \text{(Hann)} \quad w[n] &= \frac{1}{2} \left[1 + \cos \left(\frac{2\pi n}{2M+1} \right) \right] \\
 \text{(Hamming)} \quad w[n] &= 0.54 + 0.46 \cos \left(\frac{2\pi n}{2M+1} \right) \\
 \text{(Blackman)} \quad w[n] &= 0.42 + 0.5 \cos \left(\frac{2\pi n}{2M+1} \right) + 0.08 \cos \left(\frac{4\pi n}{2M+1} \right)
 \end{aligned} \tag{1.85}$$

Figure 1.13(a) depicts the plots of these windows in the time-domain, while a portion of the corresponding spectra is shown in Fig. 1.13(b). Note that these spectra share some common characteristics: all have large main “lobe” at 0 frequency, plus side lobes with decreasing amplitude. More precisely, two main spectral features have to be taken into account when analyzing the properties of these windows: first, the ability to resolve two nearby spectral components of a windowed signal depend mostly on the *main lobe width*, i.e. the nearest zero crossings on both sides of the main lobe; second, the amount of leakage from one frequency component to neighbouring bands depends on the amplitude of the side lobes, and primarily on *relative side lobe level*, i.e. the difference in dB between the amplitudes of the main lobe and the largest side lobe.

Figure 1.13(b) shows that the rectangular window has the smallest main lobe width, therefore it is better than other windows in resolving nearby sinusoids. On the other hand, it has the largest relative side lobe level, therefore it causes considerable leakage. Other windows have different performances in terms



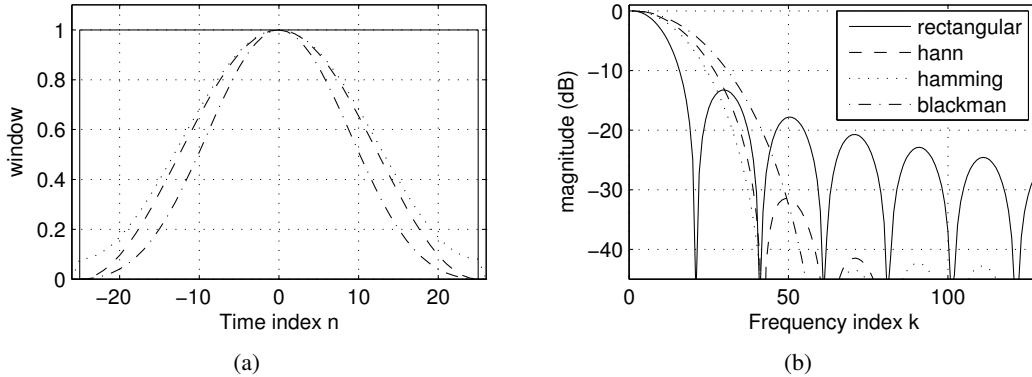


Figure 1.13: Comparison of different windows; (a) time-domain window sequences, symmetrical with respect to $n = 0$; (b) window spectra in the vicinity of $k = 0$. Differences in terms of main lobe width and relative side lobe level can be appreciated.

of main lobe width and relative side lobe level. The hamming window is often considered to provide the best trade-off for short-term spectral analysis of generic signals, however the choice of the window to use depends in general on many factors, including the characteristics of the signal to be analyzed.

M-1.13

Compute the DFT of complex exponential sequences $x[n] = e^{j2\pi k_0 n/N}$, windowed with rectangular, Hann, Hamming, and Blackman windows. Verify the performance of each window with respect to resolution and leakage effects, for integer and non-integer values of k_0 .

1.6 Digital filters

1.6.1 The z -Transform

1.6.1.1 Definitions

The z -Transform is an operator that maps a sequence of $x[n]$ into a function $X : \mathbb{C} \rightarrow \mathbb{C}$. Definition:

$$\mathcal{Z}\{x\}(z) = X(z) = \sum_{n=-\infty}^{+\infty} x[n]z^{-n}, \quad (1.86)$$

with $z \in \mathbb{C}$ complex variable.

Close relationship with the DTFT: if $z = e^{j\omega_d}$, i.e. if we restrict the variable z to the *complex unit circle*, then the z -Transform reduces to the DTFT. In particular the point $z = 1$ corresponds to frequency $\omega_d = 0$, and $z = -1$ corresponds to $\omega_d = \pi$. Therefore evaluation of the z -Transform on the upper half of the complex unit circle gives the DTFT up to the (normalized angular) Nyquist frequency. In general we can write:

$$\mathcal{F}\{x\}(\omega_d) = \mathcal{Z}\{x\}(z)|_{z=e^{j\omega_d}}. \quad (1.87)$$

For this reason we sometimes write $X(e^{j\omega_d})$ to indicate the DTFT of the sequence x .

The series in Eq. (1.86) does not converge for all values of z . For any sequence x , the set of values z for which the series converges is called *region of convergence (ROC)*. Since $|X(z)| \leq \sum_{n=-\infty}^{+\infty} |x[n]| |z|^{-n}$, if a point z_0 belongs to the ROC, then all points z that are on the complex circle with radius $|z_0|$ also



Property	Sequences	z -Transforms	ROCs
	$x[n], y[n]$	$X(z), Y(z)$	R_x, R_y
Linearity	$ax[n] + by[n]$	$aX(z) + bY(z)$	contains $R_x \cap R_y$
Time-shifting	$x[n - n_0]$	$z^{-n_0} X(z)$	R_x
z -scaling	$z_0^n x[n]$	$X(z/z_0)$	$ z_0 R_x$
z -differentiation	$nx[n]$	$-z \frac{dX}{dz}(z)$	R_x
Conjugation	$x^*[n]$	$X^*(z^*)$	R_x
Time-reversal	$x^*[-n]$	$X^*(1/z^*)$	$1/R_x$
Convolution	$(x * y)[n]$	$X(z) \cdot Y(z)$	contains $R_x \cap R_y$
Initial value theorem	If $x[n]$ causal (i.e. $x[n] = 0 \forall n < 0$), then $\lim_{z \rightarrow \infty} X(z) = x[0]$.		

Table 1.2: General properties of the z -Transform.

belong to the ROC. Therefore the ROC is in general a *ring* in the complex plane. Such ring may or may not include the unit circle, in other words the z -Transform of $x[n]$ may exist in a certain region of the complex plane even if the DTFT of $x[n]$ does not exist.

Table 1.2 lists a set of relevant properties of the z -Transform, that are particularly useful in the study of discrete-time signals and digital filters. Most of them have direct counterparts in DTFT properties (see Table 1.1), and can be proved from the definition (1.86).

The inverse z -transform is formally defined as

$$x[n] = \frac{1}{2\pi j} \oint_{\mathcal{C}} X(z) z^{n-1} dz, \quad (1.88)$$

where the integral is evaluated over the contour \mathcal{C} , which can be any closed contour that belongs to the ROC of $X(z)$ and encircles $z = 0$. Without entering into details, we remark that for the kinds of sequences and z -transforms typically encountered in digital signal processing applications, less formal procedures are sufficient and preferable. We propose some examples in Sec. 1.6.2.

1.6.1.2 z -transforms of common sequences

We illustrate the z -transform with some notable examples.

The unit impulse $\delta[n]$ has z -transform 1 and the ROC is the entire complex plane. The ideal delay $\delta[n - n_0]$ has z -transform z^{-n_0} and the ROC is the entire complex plane.

Consider the *finite-length exponential sequence*

$$x[n] = \begin{cases} a^n, & 0 \leq n < N \\ 0, & \text{elsewhere.} \end{cases} \quad (1.89)$$

The z -transform is

$$X(z) = \sum_{n=0}^{N-1} a^n z^{-n} = \sum_{n=0}^{N-1} (az^{-1})^{-n} = \frac{1 - a^N z^{-N}}{1 - az^{-1}}. \quad (1.90)$$

In particular since the series has only a finite number of terms the ROC is the entire complex plane. We can generalize and state that any finite-length sequence admits a z -transform whose ROC is the entire complex plane.



Slightly more complex example: the right-sided exponential sequence $x[n] = a^n u[n]$ already examined in Sec. 1.4.1. In this case

$$X(z) = \sum_{n=-\infty}^{+\infty} a^n u[n] z^{-n} = \sum_{n=0}^{+\infty} (az^{-1})^{-n} = \frac{1}{1 - az^{-1}}, \quad (1.91)$$

where the last equality can be written only if $|az^{-1}| < 1$, otherwise the series does not converge. In other words the ROC is the region $|z| > |a|$. It is easy to verify that the left-sided exponential sequence $x[n] = -a^n u[-n]$ also has a z -transform, identical to the one in (1.91), but with a different ROC (the region $|z| < |a|$).

This example shows that the algebraic expression of the z -transform does not completely specify the corresponding sequence, and that the ROC must also be specified. The example also shows a case of a sequence that has a z -transform but does not have a DTFT: for $a \geq 1$ the right-sided exponential sequence still admits the z -transform $1/(1 - az^{-1})$ in the region $|z| > a > 1$ although it increases exponentially in time and does not have a DTFT.

The *right-sided real sinusoidal sequence* $x[n] = \cos(\omega_0 n) u[n]$. Note that it can be written as a sum of two exponential sequences: $x[n] = (e^{j\omega_0 n} u[n] + e^{-j\omega_0 n} u[n])$. Therefore

$$X(n) = \frac{1}{2} \left[\frac{1}{1 - e^{j\omega_0} z^{-1}} + \frac{1}{1 - e^{-j\omega_0} z^{-1}} \right] = \dots = \frac{1 - [\cos \omega_0] z^{-1}}{1 - 2[\cos \omega_0] z^{-1} + z^{-2}}. \quad (1.92)$$

Since $|\cos \omega_0| \leq 1$, the ROC is clearly the region $|z| > 1$. This is a second example of a sequence that does not admit a DTFT but admits a z -transform.

A final important example is the *exponentially damped* right-sided sinusoidal sequence, defined as $x[n] = r^{-n} \cos(\omega_0 n) u[n]$, with $0 < r < 1$. In this case

$$X(n) = \frac{1 - [r \cos \omega_0] z^{-1}}{1 - 2[r \cos \omega_0] z^{-1} + r^2 z^{-2}}. \quad (1.93)$$

The ROC is the region $|z| > r$. Note that in this case the ROC includes the unit circle, therefore the sequence $x[n]$ also admits a DTFT. In fact as we will see this sequence represents the impulse response of a second-order resonating filter.

1.6.1.3 Rational z -transforms

A very important and useful family of z -transforms is that of *rational* transforms, i.e. those that can be written as

$$X(z) = \frac{P(z)}{Q(z)}, \quad \text{with } P(z), Q(z) \text{ polynomials.} \quad (1.94)$$

In fact in the previous section we have just examined a few examples of sequences with rational transforms, in particular the right-sided exponential sequence (1.91). Recalling that the z -transform is linear, we can say that any sequence that can be expressed as a linear combination of right-sided exponentials has a rational z -transform.

$$x[n] = \sum_{k=1}^N A_k a_k^n u[n] \Rightarrow X(z) = \frac{\sum_{k=1}^N A_k \prod_{m \neq k} (1 - a_m z^{-1})}{\prod_{k=1}^N (1 - a_k z^{-1})} \quad (1.95)$$

The values z for which the $P(z) = 0$ (and therefore $X(z) = 0$) are called *zeros* of $X(z)$. The values z for which $Q(z) = 0$ are called *poles* of $X(z)$. A number of important relations exist between the poles of a rational transform and its ROC.



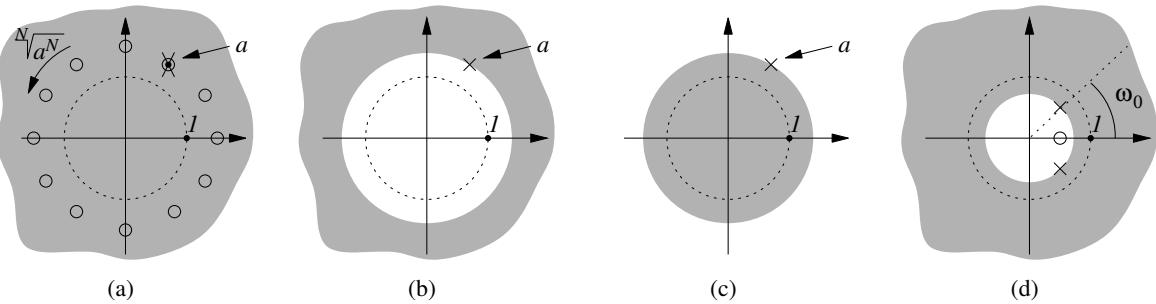


Figure 1.14: Pole-zero plots and ROCs of some simple transforms: (a) finite-length exponential sequence; (b) right-sided exponential sequence with $|a| > 1$; (c) left-sided exponential sequence with $|a| > 1$; (d) exponentially damped, right-sided sinusoidal sequence. In all plots the dotted circle is the unit circle and the gray-shaded region is the ROC of the corresponding transform.

First, the ROC cannot contain poles by definition, since $X(z)$ is not defined on the poles. It follows immediately that a finite-length sequence cannot have any poles. As an example, looking at Eq. (1.90) one notices that the pole at $z = a$ cancels with one of the zeros of the numerator $(1 - a^N z^{-N})$, therefore there are no poles. In a similar line of reasoning one can prove that for any right-sided sequence the ROC extends outwards from the pole with largest absolute value towards $|z| \rightarrow +\infty$, and that for any left-sided sequence the ROC extends inwards from the pole with smallest absolute value towards $z \rightarrow 0$. For a generic sequence that extends infinitely on both sides, the ROC consists of a ring bounded by a pole on the interior and exterior.

So-called *pole-zero plots* are typically used to represent z -transforms and their associated ROCs. Conventionally a zero is denoted with a “○” symbol and a pole is denoted with a “×” symbol. As an example, figure 1.14 shows the pole-zero plots for some of the transforms discussed in the previous section. Note in particular that the right-sided and left-sided exponential sequences have identical pole-zero patterns, but have different ROCs.

Since the pole-zero pattern does not completely define the corresponding sequence, it is sometimes convenient to specify some time-domain properties of the sequences, that implicitly define the ROC. As an example, consider the pole-zero plot of either Fig. 1.14(b) or Fig. 1.14(c) and assume that the ROC is not known. If one states that the corresponding sequence is absolutely summable, then this implies that it admits a DTFT and consequently implies that the ROC must be that of Fig. 1.14(c). Alternatively one may state that the corresponding sequence is causal: this implies that the ROC must extend towards $|z| \rightarrow +\infty$ and consequently implies that the ROC must be that of Fig. 1.14(b).

1.6.2 Transfer function and frequency response of a LTI system

1.6.2.1 Definitions

In Sec. 1.2.3 we have seen that a LTI system is completely characterized by its impulse response $h[n]$, since the response to any input sequence $x[n]$ can be written as the convolution between x and h (see Eqs. (1.22, 1.23)). Using the convolution property given in table 1.2, one can restate this by saying that for an LTI system an input sequence x is related to the corresponding output sequence y through the equation

$$Y(z) = H(z)X(z), \quad (1.96)$$



where $X(z)$, $Y(z)$, $H(z)$ are the z -transforms of $x[n]$, $y[n]$, $h[n]$, respectively. We call $H(z)$ the *transfer function* of the LTI system. Assuming that an appropriate ROC is specified for H , we can say that the LTI system is completely characterized by its transfer function.

If the ROC includes the unit circle, then $h[n]$ admits a Fourier representation. In this case we can also write

$$Y(e^{j\omega_d}) = H(e^{j\omega_d}) X(e^{j\omega_d}), \quad (1.97)$$

where $X(e^{j\omega_d})$, $Y(e^{j\omega_d})$, $H(e^{j\omega_d})$ are the DTFTs of $x[n]$, $y[n]$, $h[n]$, respectively. We call $H(e^{j\omega_d})$ the *frequency response* of the system. Assuming that the DTFTs are expressed in polar form (see Eq. (1.54)), we call $|H(e^{j\omega_d})|$ and $\arg[H(e^{j\omega_d})]$ the *magnitude response* and the *phase response* of the system, respectively.

If the LTI system under consideration has been expressed through a constant-coefficient difference equation (see Sec. 1.2.3), then one can immediately write the corresponding transfer function as a rational function:

$$\sum_{k=0}^N a_k y[n-k] = \sum_{k=0}^M b_k x[n-k] \Leftrightarrow \sum_{k=0}^N a_k z^{-k} Y(z) = \sum_{k=0}^M b_k z^{-k} X(z), \quad (1.98)$$

from which it follows immediately that

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^M b_k z^{-k}}{\sum_{k=0}^N a_k z^{-k}}. \quad (1.99)$$

Such z -transforms arise frequently in digital signal processing applications. By looking at a transform of this kind, one can easily find the corresponding sequence (this is an example of an informal procedure for determining the z -transform). First one can note that $H(z)$ can be written as

$$H(z) = \frac{b_0}{a_0} \cdot \frac{\prod_{k=1}^M (1 - c_k z^{-1})}{\prod_{k=1}^N (1 - d_k z^{-1})}, \quad (1.100)$$

where the c_k 's and the d_k 's are the zeros and the poles of H , respectively. If $M \leq N$ and the poles are all first-order, then by applying a *partial fraction expansion* the above equation can be rewritten as

$$H(z) = \sum_{k=1}^N \frac{A_k}{1 - d_k z^{-1}}, \quad \text{with } A_k = (1 - d_k z^{-1}) H(z)|_{z=d_k}. \quad (1.101)$$

Looking back at Eq. (1.91), we can conclude that $h[n]$ is a linear combination of right-sided exponential sequences (or left-sided exponential sequences, depending on the ROC).

1.6.2.2 The concept of filtering

The magnitude and phase responses of a LTI system describe how the system transforms a sinusoidal input $x[n] = A \cos(\omega_0 n) = A (e^{j\omega_0} + e^{-j\omega_0}) / 2$: the corresponding output is

$$y[n] = (h * x)[n] = A |H(e^{j\omega_0})| \cdot \cos(\omega_0 n + \arg[H(e^{j\omega_0})]), \quad (1.102)$$

i.e. the magnitude response at ω_0 defines the *gain* and the *phase delay* of the system at the frequency $\omega_d = \omega_0$. Similar considerations apply to a generic input $x[n]$: the corresponding output can be described in terms of system magnitude and phase response as

$$\begin{aligned} |Y(e^{j\omega_d})| &= |H(e^{j\omega_d})| \cdot |X(e^{j\omega_d})|, \\ \arg[Y(e^{j\omega_d})] &= \arg[H(e^{j\omega_d})] + \arg[X(e^{j\omega_d})]. \end{aligned} \quad (1.103)$$



The first equation in (1.103) says that frequency components of the input are emphasized or attenuated (or even suppressed) depending on the values of $|H(e^{j\omega_d})|$ at those frequencies. For this reason we typically refer to an LTI system as a *frequency selective filter*, or simply a *filter*. Thinking of audio, equalization is an obvious example of filtering an input sound by emphasizing certain frequency ranges and attenuating other ranges.

The second equation in (1.103) says that frequency components of the input are delayed in a frequency-dependent manner. The amount and type of tolerable *phase distortion* depends on the application. Often phase responses are disregarded in audio applications because phase distortions are to a large extent inaudible. However taking into account the phase response can be important in certain cases, e.g. when one wants to preserve the shape of the time-domain waveform.

A generally tolerable type of phase distortion is *linear distortion*. A filter with a linear phase response produces the same phase delay for all frequencies: as an example, the ideal delay system $h_{n_0} = \delta(n-n_0)$ has a linear phase response $\arg[H_{n_0}(e^{j\omega_d})] = \omega_d n_0$. A convenient measure of the linearity of the phase response of a filter is the *group delay*, defined as³

$$\tau(\omega_d) = -\frac{d}{d\omega_d} \{\arg[H(e^{j\omega_d})]\}. \quad (1.104)$$

The deviation of τ from a constant indicates the degree of phase non-linearity. Figure 1.15(a) provides a graphical comparison of the phase delay and the group delay.

The reason why τ is termed group delay is that this quantity relates to the effect of the phase on a quasi-sinusoidal signal. More precisely, consider the signal $x[n] = a[n]e^{\omega_0 n}$, and assume that $a[n]$ is varying slowly (equivalently, assume that the spectrum $A(e^{j\omega})$ is concentrated near $\omega = 0$). Then x will look like the signal in Fig. 1.15(b) (upper panel). The signal can also be rewritten as

$$x[n] = a[n]e^{\omega_0 n} = \left[\frac{1}{2\pi} \int_{-\epsilon}^{+\epsilon} A(e^{j\omega})e^{j\omega n} d\omega \right] e^{\omega_0 n}, \quad (1.105)$$

where $\epsilon \ll \pi$ is the upper limit of the band of A . Therefore x can be viewed as a superposition of neighboring sinusoidal components, or a *group* around ω_0 .

Since $X(e^{j\omega_d}) \neq 0$ only in the vicinity of ω_0 , the phase response can be approximated in that neighborhood as a line with slope $-\tau(\omega_0)$. With this approximation it is then quite straightforward to show that the output is the filter output $y[n]$ looks like in Fig. 1.15(b) (lower panel), i.e. $\tau(\omega_0)$ represents the delay applied to the slowly-varying amplitude $a[n]$.

1.6.2.3 Stability, causality, and transfer functions

We have defined in Sec. 1.2.3 the notion of BIBO stability, and we have proved that an LTI system is stable if and only if its impulse response is absolutely summable. This latter condition can be rewritten as

$$\sum_{n=-\infty}^{+\infty} |h[n]z^{-n}| < \infty, \quad (1.106)$$

for $|z| = 1$. Therefore the condition of stability is equivalent to the condition that the ROC of the transfer function includes the unit circle. The examples depicted in Fig. 1.14 confirm this finding.

Consider the relevant particular case of rational transfer functions associated to causal systems: for these transfer functions the condition of stability is equivalent to the condition that all the poles are inside the unit circle, because the ROC extends outwards from the pole with the largest absolute value.

³Our definition uses ω_d , so that τ is adimensional. Usual definitions employ ω , so that τ is in seconds.



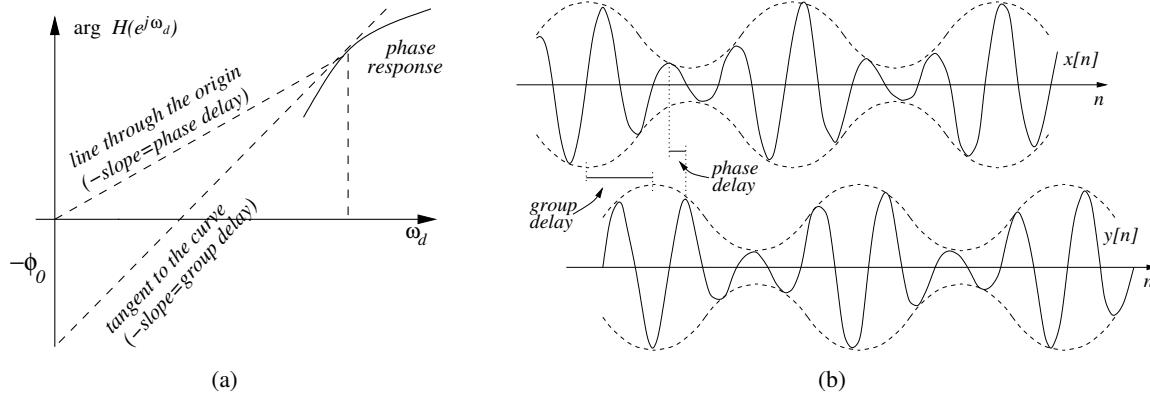


Figure 1.15: Comparing phase delay and group delay: (a) evaluation of phase delay and group delay for a generic non-linear phase response; (b) illustration of phase delay and group delay for a narrowband signal.

Another relevant particular case are FIR systems. We have already seen that a FIR system is always stable since its impulse response is always absolutely summable. This property can be “seen” in the z domain by noting that the transfer function of a FIR system does not have poles, therefore the ROC is always the entire z plane and includes the unit circle.

1.6.3 Digital filter structures and design approaches

In the following we only give a quick overview. In the next chapters we will discuss many specific filters. In particular we will examine the second-order resonant filter in Chapter *Sound modeling: signal based approaches*; comb and all-pass in Chapter *Sound modeling: source based approaches*; more comb and all-pass structures in Chapter *Sound in space*. We do not discuss filter structures (direct forms etc.), just a few words. Same with design techniques. In Chapter *Sound modeling: source based approaches* we talk about bilinear transform, s-to-z mappings, impulse response invariance, all techniques used to design a digital filter from an analog one. In Chapter *Sound in space* we will see pole-zero approximations.

1.6.3.1 Block diagram representations

Most the LTI systems that we are going to realize are represented as linear constant coefficient equations. As we have seen, the output of a LTI system represented in this way can be computed recursively provided that past values of input and output are available. These values will undergo two operations in the computation of the filter output: multiplication with coefficients, and addition.

Therefore the three basic elements for the implementation of a filter are memory for storing, past values, adders, and multipliers. A filter structure is created by properly interconnecting these elements. Figure 1.16 shows the pictorial symbols that are typically used for representing them. With these elements, a general filter

$$y[n] - \sum_{k=1}^N a_k y[n-k] = \sum_{k=0}^M b_k x[n-k] \Rightarrow H(z) = \frac{\sum_{k=0}^M b_k z^{-k}}{\sum_{k=1}^N a_k z^{-k}} \quad (1.107)$$

can be represented with the block diagram of Fig. 1.16(b).



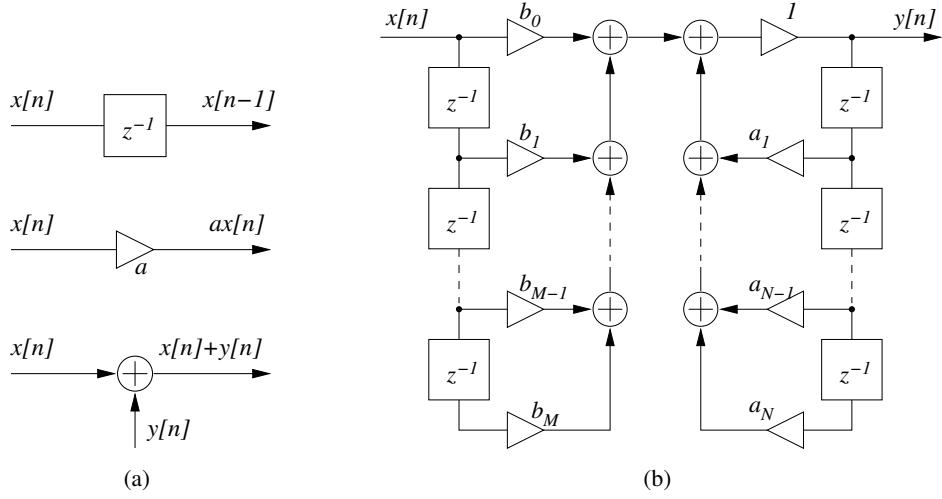


Figure 1.16: Block diagram representation of filters: (a) conventional pictorial symbols for delay, adder, and multiplier; (b) a general filter structure.

1.6.3.2 Filter classification

Filters can be classified according to most salient characteristics of their frequency responses, most typically their magnitude. Basic filter categories are represented in Fig. 1.17. Other types of filters can in general be described as a combination of these basic elements.

Low-pass filters (see Fig. 1.17(a), upper panel) select low frequencies up to a given *cut-off* frequency ω_c , and attenuate higher frequencies. *High-pass* filters (see Fig. 1.17(a), lower panel) have the opposite behavior: they select frequencies above ω_c , and attenuate lower frequencies.

Band-pass filters (see Fig. 1.17(b), upper panel) select frequencies within a frequency band, specified by two cut-off frequencies ω_{c1} and ω_{c2} , while frequencies outside this band are attenuated. *Band-reject* filters (see Fig. 1.17(b), lower panel) have the opposite behavior: they select frequencies outside the band $[\omega_{c1}, \omega_{c2}]$, and attenuate frequencies within the band.

Resonator filters (see Fig. 1.17(c), upper panel) amplify frequencies in a narrow band around a cut-off frequency ω_c . Conversely, *notch* filters (see Fig. 1.17(c), lower panel) attenuate frequencies in a narrow band around ω_c . Finally, when the magnitude response is perfectly flat the filter is called an *all-pass* filter, since all frequencies are passed. Note however that an all-pass filter modifies the phase of the input signal. In the next chapter we will see some important uses of all-pass filters.

In order to optimize their frequency selective properties, ideal filters should have magnitude responses exhibiting vertical transition between selected frequencies and rejected ones. Moreover they should have null or linear phase response in order not to introduce phase distortion. As an example, the *ideal low-pass filter* has the frequency response

$$H_{lp}(e^{j\omega}) = \begin{cases} 1, & |\omega| \leq \omega_c, \\ 0, & \omega_c < |\omega| \leq \pi. \end{cases} \quad (1.108)$$

However the corresponding impulse response is

$$h_{lp}[n] = \frac{\sin \omega_c n}{\pi n}, \quad -\infty < n < +\infty, \quad (1.109)$$



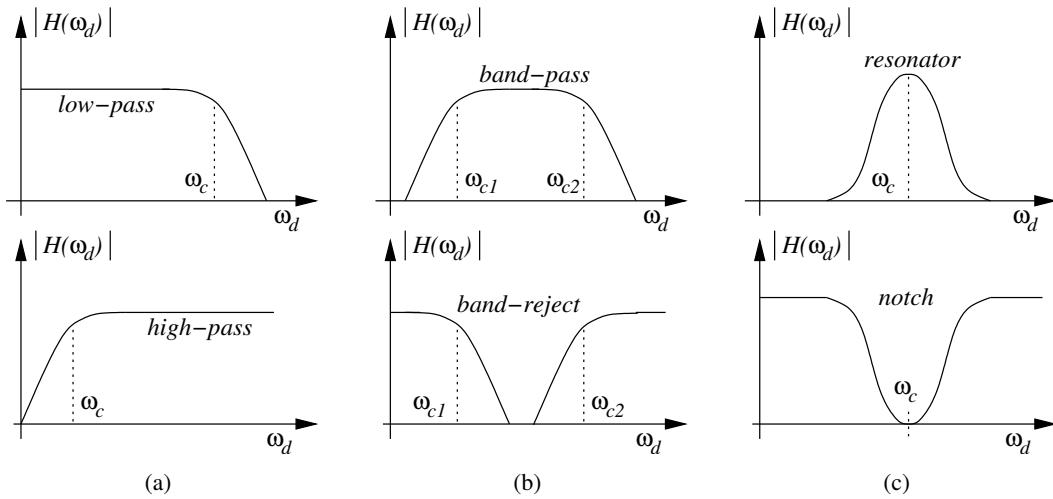


Figure 1.17: Classification of filters into basic categories, depending on their magnitude response $|H(\omega_d)|$: (a) low-pass and high-pass filter; (b) band-pass and band-reject filter; (c) resonator and notch filter.

i.e. the filter is non-causal and has a two-sided infinite impulse response. Therefore it is not possible to compute its output either recursively or non-recursively. In other words, the filter is *not realizable*. Similar considerations apply to other types of ideal filters.

The simplest examples of realizable filters are first-order filters (i.e. filters with no more than one pole and/or one zero). First-order FIR low-pass and high-pass filters are defined as follows:

$$H_{lp}(z) = \frac{1}{2} (1 + z^{-1}), \quad H_{hp}(z) = \frac{1}{2} (1 - z^{-1}). \quad (1.110)$$

They have a zero in $z = 1$ and $z = -1$, respectively. Therefore the magnitude responses decrease and increase monotonically, respectively. The low-pass filter in particular can be recognized to be a moving average filter that averages two contiguous samples.

First-order IIR low-pass and high-pass filters are defined as follows:

$$H_{lp}(z) = \frac{1 - \alpha}{2} \frac{1 + z^{-1}}{1 - \alpha z^{-1}}, \quad H_{hp}(z) = \frac{1 + \alpha}{2} \frac{1 - z^{-1}}{1 - \alpha z^{-1}}. \quad (1.111)$$

Both have a pole in $z = \alpha$, therefore $|\alpha| < 1$ for stability, and $\alpha \in \mathbb{R}$ in order for the impulse responses to be real-valued. They have a zero in $z = 1$ and $z = -1$, respectively. For $\alpha = 0$ they reduce to the FIR filters above, while for $\alpha > 0$ they have steeper responses.

M-1.14

Study the frequency responses of the first-order low-pass and high-pass filters of Eqs. (1.110, 1.111). Apply them to a broad-band audio signal (e.g. a snaredrum), and study their effect.

1.7 Commented bibliography

A general reference for digital signal processing is [Oppenheim et al., 1999]. Another classic is Mitra [2005]: more implementation oriented, with Matlab examples.



Primers on digital audio processing: Rocchesso [2003] and Steiglitz [1996]. Examples focused on audio are found also in [Zölzer, 2002, Chapter 1]

A useful reading about Fourier analysis for discrete-time signals is provided in [Smith, 2008]. Our discussion of FFT algorithms is based on [Cormen et al., 2001].

A discussion on recursive generators of sinusoidal signals is found e.g. in [Orfanidis, 1996]. Models for fractal signals are also partially discussed in [Orfanidis, 1996].

References

- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, Cambridge, MA, 2001.
- Sanjit K Mitra. *Digital Signal Processing*. McGraw-Hill, third edition, 2005.
- Alan V. Oppenheim, Ronald W. Schafer, and John R. Buck. *Discrete-Time Signal Processing*. Prentice Hall, Upper Saddle River, NJ, 1999.
- Sophocles J. Orfanidis. *Introduction to Signal Processing*. Prentice Hall, 1996.
- Davide Rocchesso. *Introduction to Sound Processing*. Mondo Estremo, Firenze, 2003. <http://profsci.univr.it/~rocchess/SP>.
- Julius O. Smith. *Mathematics of the Discrete Fourier Transform (DFT)*. <http://ccrma.stanford.edu/~jos/mdft/>, 2008. Online book, accessed Oct. 2008.
- Kenneth Steiglitz. *A Digital Signal Processing Primer - With Applications to Digital Audio and Computer Music*. Prentice Hall, 1996.
- Udo Zölzer, editor. *DAFX – Digital Audio Effects*. John Wiley & Sons, 2002.



Chapter 2

Sound modeling: signal-based approaches

Giovanni De Poli and Federico Avanzini

Copyright © 2005-2018 Giovanni De Poli and Federico Avanzini
except for paragraphs labeled as *adapted from <reference>*

This book is licensed under the CreativeCommons Attribution-NonCommercial-ShareAlike 3.0 license. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/>, or send a letter to Creative Commons, 171 2nd Street, Suite 300, San Francisco, California, 94105, USA.

2.1 Introduction

The sound produced by acoustic musical instruments is caused by the physical vibration of a certain resonating structure. This vibration can be described by signals that correspond to the time-evolution of the acoustic pressure associated to it. The fact that the sound can be characterized by a set of signals suggests quite naturally that some computing equipment could be successfully employed for generating sounds, for either the imitation of acoustic instruments or the creation of new sounds with novel timbral properties.

A wide variety of sound synthesis algorithms is currently available either commercially or in the literature. Each one of them exhibits some peculiar characteristics that could make it preferable to others, depending on goals and needs. Technological progress has made enormous steps forward in the past few years as far as the computational power that can be made available at low cost is concerned. At the same time, sound synthesis methods have become more and more computationally efficient and the user interface has become friendlier and friendlier. As a consequence, musicians can nowadays access a wide collection of synthesis techniques (all available at low cost in their full functionality), and concentrate on their timbral properties.

Each sound synthesis algorithm can be thought of as a digital model for the sound itself. Though this observation may seem quite obvious, its meaning for sound synthesis is not so straightforward. As a matter of fact, modeling sounds is much more than just generating them, as a digital model can be used for representing and generating a whole class of sounds, depending on the choice of control parameters. The idea of associating a class of sounds to a digital sound model is in complete accordance with the way we tend to classify natural musical instruments according to their sound generation mechanism. For example, strings and woodwinds are normally seen as timbral classes of acoustic instruments characterized by their sound generation mechanism. It should be quite clear that the degree of compactness of a class

of sounds is determined, on one hand, by the sensitivity of the digital model to parameter variations and, on the other hand, on the amount of control that is necessary for obtaining a certain desired sound. As an extreme example we may think of a situation in which a musician is required to generate sounds sample by sample, while the task of the computing equipment is just that of playing the samples. In this case the control signal is represented by the sound itself, therefore the class of sounds that can be produced is unlimited but the instrument is impossible for a musician to control and play. An opposite extremal situation is that in which the synthesis technique is actually the model of an acoustic musical instrument. In this case the class of sounds that can be produced is much more limited (it is characteristic of the mechanism that is being modeled by the algorithm), but the degree of difficulty involved in generating the control parameters is quite modest, as it corresponds to physical parameters that have an intuitive counterpart in the experience of the musician.

An interesting conclusion that could be already drawn in the light of what stated above is that the compactness of the class of sounds associated to a sound synthesis algorithm is somehow in contrast with the "playability" of the algorithm itself. One should remember that the "playability" is of crucial importance for the success of a specific sound synthesis algorithm as, in order for a sound synthesis algorithm to be suitable for musical purposes, the musician needs an intuitive and easy access to its control parameters during both the sound design process and the performance. Such requirements often represents the reason why a certain synthesis technique is preferred to others.

Some considerations on control parameters are now in order. Varying the control parameters of a sound synthesis algorithm can serve several purposes, the first one of which is certainly that of exploring a sound space, i.e. producing all the different sounds that belong to the class characterized by the algorithm itself. This very traditional way of using control parameters would nowadays be largely insufficient by itself. As a matter of fact, with the progress in the computational devices that are currently being employed for musical purposes, the musician's needs have turned more and more toward problems of timbral dynamics. For example, timbral differences between soft (dark) and loud (brilliant) tones are usually obtained through appropriate parameter control. Timbral expression parameters tend to operate at a note-level time-scale. As such, they can be suitably treated as signals characterized by a rather slow rate.

Another reason for the importance of time-variations in the algorithm parameters is that the musician needs to control the musical expression while playing. For example, staccato, legato, vibrato etc. need to be obtained through parameter control. Such parameter variations operate at a phrase-level time-scale. Because of that, they can be suitably treated as sequences of symbols events characterized by a very slow rate.

In conclusion, control parameters are signals characterized by their own time-scales. Controls signals for timbral dynamics are best described as discrete-time signals with a slow sampling rate, while controls for musical expression are best described by streams of asynchronous symbols events. As a consequence, the generation of control signals can once again be seen as a problem of signal synthesis.

2.2 Time-segment based models

2.2.1 Wavetable synthesis

2.2.1.1 Definitions and applications

Finding a mathematical model that faithfully imitates a real sound is an extremely difficult task. If an existing reference sound is available, however, it is always possible to reproduce it through recording. Such a method, though simple in its principle, is widely adopted by digital sampling instruments or



samplers and is called wavetable synthesis or sampling.¹

A samplers is a device which is able to store and play back a large quantity of recorded sounds: most typically these are musical, instrumental sounds, but they may also be non-musical, environmental sounds. In fact the most appealing quality of wavetable synthesis is the possibility of working with an unlimited variety of sounds. From the implementation viewpoint, computational simplicity is certainly an advantage of the technique, which contrasts with the need of huge memory capacities (as an example, a contemporary piano synthesizer based on sampling may require tens of Gigabytes to store the entire instrument).

The table look-up mechanism is in essence identical to that already outlined for the wavetable oscillator in Chapter *Fundamentals of digital audio processing*. The recorded sound is stored in a table of length L , and the synthesized sound $x[n]$ is the result of reading the table using a phase signal $\phi[n]$ with a certain length M :

$$x[n] = \text{tab}\{\phi[n]\}, \quad \phi : [0, M - 1] \rightarrow [0, L - 1]. \quad (2.1)$$

Simple playback one sound of the stored repertoire is obtained using the signal $\phi[n] = n$ and with $M = L$. In this case all the original samples of the stored sound are played back at the original speed.

Using different signals $\phi[n]$ will result in modifications of the original sound. However the possibilities of modification are rather limited, as it would be for the sound recorded by a tape deck. The most common modification is obtained by varying the speed when reproducing the sound, which results in a pitch transposition. Other simple transformations include time-reversal (i.e. table read out starting from the last sample), looping (i.e. periodically scanning the wavetable or part of it), cutting and possibly rearranging portions of the original sound. In particular, it is possible to adopt looping techniques with almost any stationary portion of sounds. One method of improving the expressive possibilities of samplers is store multiple instances of a single instrumental note for different dynamics (e.g. from soft to loud key velocities in the case of the piano), or even different pitches, and switching or interpolating between these upon synthesis. This approach is called *multisampling*. During synthesis, linear interpolation between stored sounds is performed as function of the desired dynamics.

Historically, the use of samplers may be traced back to a time where digital sound was still to come and magnetic tapes had just made their appearance. So-called *musique concrète*, first developed in Paris in late 1940's mainly through the work of french composer Pierre Schaeffer and colleagues, stemmed from an aesthetic practice that was centered upon the use of sound as a primary compositional resource. The development of this music was facilitated by the emergence of then new technologies such as microphones and magnetic tape recorders. Compositions were based on (usually non-musical) sounds acquired through a microphone, recorded on tape, manipulated and recombined in variety of ways, and finally played back from tape.

Nowadays wavetable synthesis techniques are presented as a method for reproducing natural instrumental sounds that are comparable in quality with the original instruments. This is the main reason why the most popular commercial digital keyboards adopt this synthesis technique. Of course, sampling cannot feature all the expressive possibilities of the original instrument.

2.2.1.2 Transformations: pitch shifting, looping

Pitch shifting, i.e. transposition of the original pitch of the recorded sound, is the simplest transformation obtainable with wavetable synthesis. Assume that the original sample rate and the playback sample rate are the same. If we read the entire table (of length L) with a signal ϕ of length M , then the overall sound duration is compressed or expanded by a factor M/L and the pitch is transposed by a factor L/M .

¹In musical jargon it has become customary to use the term *sample* to denote an entire sound, recorded and stored in a table. To avoid confusion, in this book we only use the term *sample* only with the meaning explained in Chapter *Fundamentals of digital audio processing*, i.e. a single value of a digital signal.



An overall transposition of a factor L/M can be obtained by reading the table with the signal

$$\phi[n] = \left\lfloor n \frac{L}{M} \right\rfloor, \quad n = 0, \dots, M - 1. \quad (2.2)$$

As an example, if $3M = 2L$ the sound will be transposed upwards by a perfect fifth (i.e. by a ratio of $3/2$). However, ϕ needs not to be a ramp that rises linearly in time. In the case where ϕ is still a monotonic signal, but changes its slope during time, the transposition will become time-dependent.

Although the pitch shifting outlined above is simple and straightforward to implement, it has to be noted that substantial pitch variations are generally not very satisfactory as a temporal waveform compression or expansion results in unnatural timbral modifications, which is exactly what happens when the playing speed is changed in a tape recorder. Satisfactory quality and timbral similarity between the original tone and the transposed one can be obtained only if small pitch variations (e.g. a few semitones) are performed. As an example, sampling a piano with a reasonable quality along the entire instrumental extension requires that many notes are stored (e.g. three for each octave). In this way, notes that have not been sampled can be obtained from the available ones through transposition of max. two semitones.

M-2.15

Import a .wav file of a single instrument tone. Scale it (compress and expand) to different extents and listen to the new sounds. Up to what scaling ratio are the results acceptable?

Often it is desired to vary the sound also in function of other parameters, the most important being the intensity. To this purpose it is not sufficient to change the sound amplitude by a multiplication, by it is necessary to modify the timbre of the sound. In general louder sounds are characterized by a sharper attack and by a brighter spectrum. In this case a technique could be to use a unique sound prototype (e.g. a tone played fortissimo) and then obtaining the other intensity by simple spectral processing, as low pass filtering. A different and more effective solution, is to use a set of different sound prototype, recorder with different intensity (e.g. tones played fortissimo, mezzo forte, pianissimo) and then obtaining the other dynamic values by interpolations and further processing.

This technique is thus characterized by high computational efficiency and high imitation quality, but by low flexibility for sounds not initially included in the repertoire or not easily obtainable with simple transformations. There is a trade-off of memory size with sound fidelity.

In order to employ efficiently the memory, often the sustain part of the tone is not entirely stored but only a part (or few significant parts) and in the synthesis this part is repeated (*looping*). Naturally the repeated part should not be too short, to avoid a static character of the resulting sound. For example to lengthen the duration of a note, first the attack is reproduced without modification, then the sustain part is cyclically repeated, with possible cross interpolation among the different selected parts, and finally the sound release stored part is reproduced. Notice that if we want to avoid artefacts in cycling, particular care should be devoted to choosing the points of the beginning and ending of the loop. Normally an integer number of periods is used for looping starting with a null value, to avoid amplitude or phase discontinuities. In fact these discontinuities are very annoying. To this purpose it may be necessary to process the recorded samples by slightly changing the phases of the partials.

M-2.16

Import a .wav file of a single instrument tone. Find the stationary (sustain) part, isolate a section, and perform the looping operation. Listen to the results, and listen to the artifacts when the looped section does not start/end at zero-crossings.

If we want a less static sustain, it is possible to individuate some different and significant sound segments, and during the synthesis interpolate (*cross-fade*) among subsequent segments. In this case the temporal evolution of the tone can be more faithfully reproduced.



2.2.2 Overlap-Add (OLA) methods

The definition Overlap-Add (OLA) refers in general to a family of algorithms that produce a signal by properly assembling a number of signal segments. OLA methods are developed both in the time domain and in the frequency domain. Here we are interested in reviewing briefly time-domain approaches, while in Sec. 2.3 we will address time-frequency representations.

2.2.2.1 Basic time-domain overlap-add

Given a sound signal $x[n]$, a sequence of windowed segments $x_m[n]$ can be constructed as

$$x_m[n] = x[n]w_a[n - mS_a], \quad (2.3)$$

where $w_a[n]$ is an analysis window and S_a is the *analysis hop-size*, i.e. the time-lag (in samples) between one analysis frame and the following one. If the window w_a is N samples long, then the *block size*, i.e. the length of each frame $x_m[n]$, will be N . In order for the signal segments to actually overlap, the inequality $S_a \leq N$ must be verified. When $S_a = N$ the segments are exactly juxtaposed with no overlap.

Given the above signal segmentation, time-domain overlap-add (OLA) methods construct an output signal $y[n] = \sum_m y_m[n]$, where the segments y_m are modified versions of the input segments x_m . As an example, they can be obtained by modifying S_a in Eq. (2.3), or by repeating/removing some of the input segments x_m . In the absence of modifications, this procedure reduces to the identity ($y[n] \equiv x[n]$) if the overlapped and added analysis windows w_a sum to unity:

$$y[n] = \sum_m x_m[n] = \sum_m x[n]w_a[n - mS_a] = x[n] \Leftrightarrow A_{w_a}[n] \triangleq \sum_m w_a[n - mS_a] \equiv 1. \quad (2.4)$$

If this condition does not hold, then the function A_{w_a} acts on the reconstructed signal as a periodic *amplitude modulation envelope*, with period S_a . Depending on the application and on the window length N , this amplitude modulation can introduce audible artifacts. This kind of frame rate distortion can be seen in the frequency domain as a series of sidebands with spacing F_s/S_a in a spectrogram of the output signal. In fact, one may prove that the condition $A_{w_a} \equiv 1$ is equivalent to the condition $W(e^{j\omega_k}) = 0$ at all harmonics of the frame rate F_s/S_a . Figure 2.1 illustrates an example of signal reconstruction using a triangular window, which satisfies the condition of Eq. (2.4).

A widely studied effect is *time-stretching*, i.e. contraction or expansion of the duration of an audio signal. Time-stretching algorithms useful in a number of applications: think about wavetable synthesis, post-synchronization of audio and video, speech technology at large, and so on. A time-stretching algorithm should ideally shorten or lengthen a sound file composed of N_{tot} samples to a new desired length $N'_{tot} = \alpha N_{tot}$, where α is the *stretching factor*. Note that a mere resampling of the sound signal does not provide the desired result, since it has the side-effect of transposing the sound: in this context resampling is the digital equivalent of playing the tape at a different speed.

What one really wants is a scaling of the perceived timing attributes without affecting the perceived frequency attributes. More precisely, we want the time-scaled version of the audio signal to be perceived as the same sequence of *acoustic events* as the original signal, only distributed on a compressed/expanded time pattern. As an example, a time-stretching algorithm applied to a speech signal should change the speaking rate without altering the pitch.

Time-domain OLA techniques are one possible approach to time-stretching effects. The basic OLA algorithm described above can be adapted to this problem by defining an *analysis hop size* S_a and a *synthesis hop size* $S_s = \alpha S_a$, scaled by the stretching factor, that will be applied to the output. An input signal $x[n]$ is then segmented into frames $x_k[n]$, each taken every S_a samples. The output signal $y[n]$



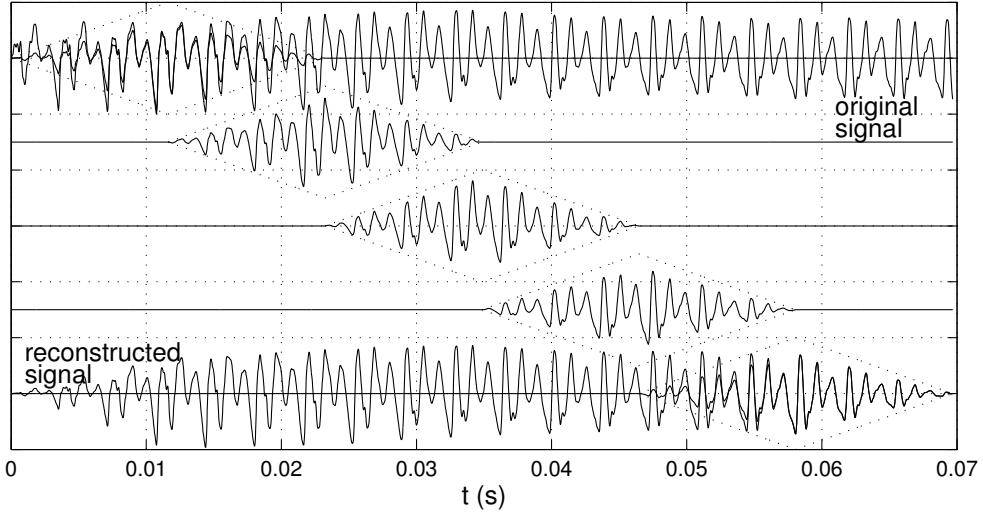


Figure 2.1: An example of OLA signal reconstruction, with triangular windowing.

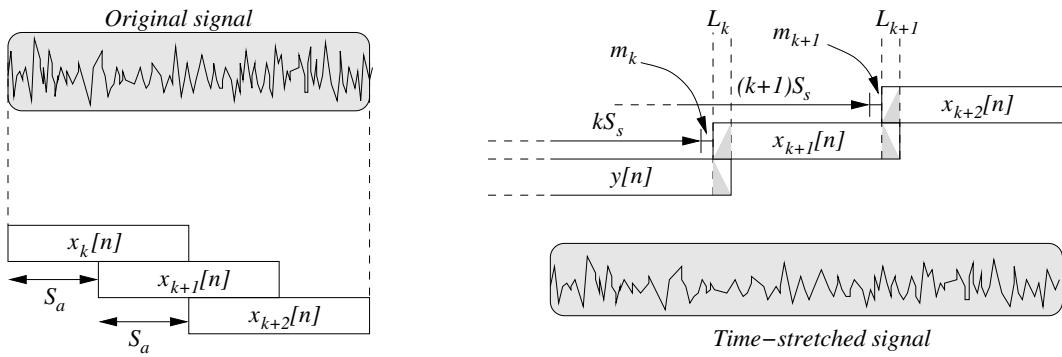


Figure 2.2: The generic SOLA algorithmic step.

is produced by reassembling the same frames $x_k[n]$, each added to the preceding one every S_s samples. However this repositioning of the input segments with respect to each other destroys the original phase relationships, and constructs the output signal by interpolating between these misaligned segments. This cause pitch period discontinuities and distortions that can produce heavily audible artifacts in the output signal.

2.2.2.2 Synchronous overlap-add

In order to avoid phase discontinuities at the boundaries between frames, a proper time alignment of the blocks has to be chosen. The *synchronous overlap-add (SOLA)* algorithm realizes such a proper alignment, and provides a good sound quality (at least for values of α not too far from 1) while remaining computationally simple, which makes it suitable even for real-time applications.

Let N be the analysis block length. In the initialization phase the SOLA algorithm copies the first N samples from $x_1[n]$ to the output $y[n]$, to obtain a minimum set of samples to work on:

$$y[j] = x_1[j] = x[j] \quad \text{for } j = 0 \dots N - 1. \quad (2.5)$$

Then, during the generic k th step the algorithm tries to find the optimal overlap between the last portion



of the output signal $y[n]$ and the incoming analysis frame $x_{k+1}[n]$. More precisely, $x_{k+1}[n]$ is pasted to the output $y[n]$ starting from sample $kS_s + m_k$, where m_k is a small discrete time-lag that optimizes the alignment between y and x_k (see Fig. 2.2). Note that m_k can in general be positive or negative, although for clarity we have used a positive m_k in Fig. 2.2.

When the optimal time-lag m_k is found, a *linear crossfade* is used within the overlap window, in order to obtain a gradual transition from the last portion of y to the first portion of x_k . Then the last samples of x_k are pasted into y . If we assume that the overlap window at the k th SOLA step is L_k samples long, then the algorithmic step computes the new frame of the input y as

$$y[kS_s + j] = \begin{cases} (1 - v[j])y[kS_s + j] + v[j]x_k[j] & \text{for } m_k \leq j \leq L_k \\ x_k[j] & \text{for } L_k + 1 \leq j \leq N \end{cases} \quad (2.6)$$

where $v[j]$ is a linear *smoothing function* that realizes the crossfade between the two segments. The effect of Eq. (2.6) is a local replication or suppression of waveform periods (depending on the value of α), that eventually results in an output signal $y[n]$ with approximately the same spectral properties of the input $x[n]$, and an altered temporal evolution.

At least three techniques are commonly used in order to find the optimal value for the discrete time lag m_k at each algorithmic step k :

1. Computation of the minimum vectorial inter-frame distance in an L_1 sense (cross-AMDF)
2. Computation of the maximum cross-correlation $r_k(m)$ in a neighborhood of the sample kS_s . Let M be the width of such neighborhood, and let $y_{M_k}[i] = y[kS_s + i]$ for $i = 1 \dots M - 1$, and $x_{M_k}[i] = x_{k+1}[i]$ for $i = 1 \dots M - 1$. Then the cross-correlation $r_k(m)$ is computed as

$$r_k[m] \triangleq \sum_{i=0}^{M-m-1} y_{M_k}[i] \cdot x_{M_k}[i+m], \quad m = -M + 1, \dots, M - 1. \quad (2.7)$$

Then m_k is chosen to be the index of maximal cross-correlation: $r_k[m_k] = \max_m r_k[m]$.

3. Computation of the maximum *normalized* cross-correlation, where every value taken from the cross-correlation signal is normalized by dividing it by the product of the frame energies.

The latter technique is conceptually preferable, but the second one is often used for efficiency reasons.

M-2.17

Write a function that realizes the time-stretching SOLA algorithm through segment cross-correlation.

M-2.17 Solution

```
function y = sola_timestretch(x,N,Sa,alpha,L)
%N: block length; Sa: analysis hop-size; alpha: stretch factor; L: overlap int.

Ss = round(Sa*alpha); %synthesis hop-size
if ( (Sa > N) || (Ss >= N) || Ss > N-L) error('Wrong parameter values!'); end
if (rem(L,2) ~= 0) L = L+1; end

M = ceil(length(x)/Sa); %number of frames
x(M*Sa+N)=0; %now x is exactly M*Sa samples
y(1:N,1) = x(1:N); %first frame of x is written into y;

for m=1:M-1 %loop over frames
    frame=x(m*Sa+1:N+m*Sa); %current analysis frame
    framecorr=xcorr(frame(1:L),y(m*Ss:m*Ss+(L-1)));

```



```
[corrmax,imax]=max(framecorr); %find point of max xcorrelation

fade = (m*Ss-(L-1)+imax-1):length(y); %points for crossfade
fadein = (0:length(fade)-1)/length(fade); %from 0 to 1
fadeout = 1 - fadein; %from 1 to 0
y=[y(1:(fade(1)-1)); (y(fade).*fadeout' +frame(1:length(fade)).*fadein'); ...
frame(length(fade)+1:length(frame))];
end
```

2.2.2.3 Pitch synchronous overlap-add

A variation of the SOLA algorithm for time stretching is the *pitch synchronous overlap-add* (PSOLA) algorithm, which is especially used for voice processing. PSOLA assumes that the input sound is pitched, and exploits the pitch information to correctly align the segments and avoid pitch discontinuities. The algorithm is composed of two phases: analysis/segmentation of the input sound, and resynthesis of the time-stretched output signal.

The analysis phase works in two main steps. The first *pitch estimation* step determines a set $\{n_i\}_i$ of time instants (“pitch marks”), such that signal portions between n_{i+1} and n_i contain an integer number of pitch periods (e.g., 2 or 4 periods), and the pitch can be considered constant within each of these portions. In this way a pitch function $P[n_i] = n_{i+1} - n_i$ is estimated. This function represents the time-varying period (in samples) of the original signal. As an example, in a voice signal pitch marks can be chosen to be points of maximum amplitude of the mouth pressure signal: they corresponds to instants of closure of the vocal folds, which occur periodically in a voiced signal. This step is clearly the most critical and computationally expensive one.

The second step in the analysis phase is a *segmentation* step. Segments $x_i[n]$ are created by windowing the input signal (typically a Hanning window is used): segments have a block length N of two pitch periods, and are centered at every pitch mark n_i . This procedure implies that the analysis hop-size is $S_a = N/2$.

In the synthesis phase, the segments $x_i[n]$ are realigned as follows. First, note that the time-stretched signal must have a pitch function \tilde{P} that is a stretched version of P . More precisely, the relation $\tilde{P}[\alpha n_i] = P[n_i]$ must hold (where for simplicity we are assuming that the time instants αn_i are still integers). This relation is then used to determine a new set $\{\tilde{n}_k\}_k$ of pitch marks for the output signal. The \tilde{n}_k 's are iteratively determined as follows:

$$\begin{aligned}\tilde{n}_1 &= n_1, \\ \tilde{n}_{k+1} &= \tilde{n}_k + P(n_{i(k,\alpha)}), \quad k > 1,\end{aligned}\tag{2.8}$$

where $i(k, \alpha)$ is the index of the input pitch marks that minimizes the distance $|\alpha n_i - \tilde{n}_k|$. Intuitively, this means that at the time instant \tilde{n}_k the time-stretched signal must have the same pitch possessed by the original signal at time n_i , with $\tilde{n}_k \sim \alpha n_i$.

Once the set $\{\tilde{n}_k\}_k$ has been determined in this way, for every k the segment $x_{i(k,\alpha)}[n]$ is overlapped and added at the point \tilde{n}_k . The algorithm is visualized in Fig. 2.3: note that with a stretching factor $\alpha > 1$ (time expansion, as in Fig. 2.3) some segments will be *repeated*, or equivalently the function $i(k, \alpha)$ will take identical values for some consecutive values of k . Similarly, when $\alpha < 1$ (time compression) some segments are discarded in the resynthesis.

The main advantage of the PSOLA algorithm with respect to SOLA is that it allows for a better alignment of segments, by exploiting information about pitch instead of using a simple cross-correlation. On



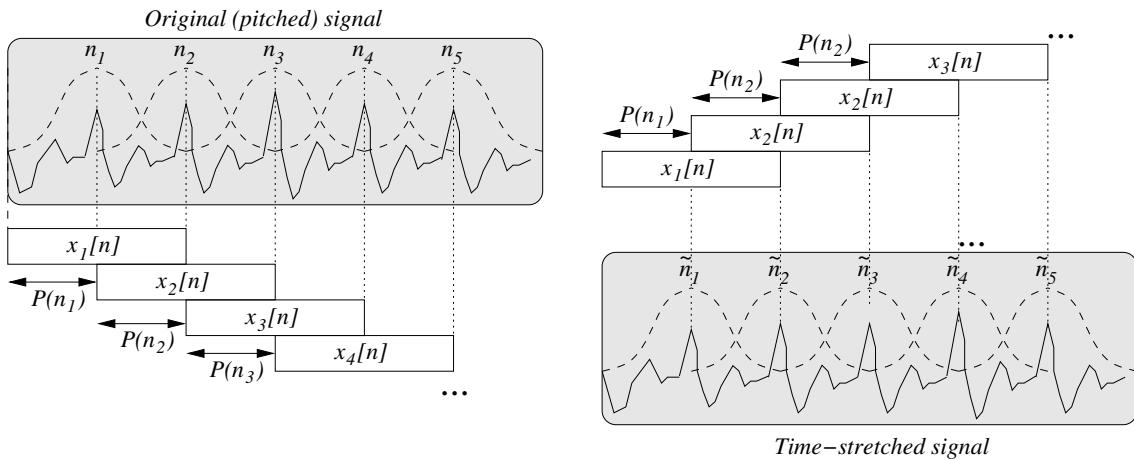


Figure 2.3: The generic PSOLA algorithmic step.

the other hand it has a higher complexity especially because of the pitch estimation procedure. Moreover, noticeable artifacts still appears for very small or large stretching factors. One problem is that when α is very large, identical segments will be repeated several times thus providing an unnatural character to the sound. A second more general problem is that the OLA algorithms examined here stretch an input signal uniformly, including possible transients, which instead should be preserved.

M-2.18

Write a function that realizes the time-stretching PSOLA synthesis algorithm, given a vector of input pitch marks n_i .

M-2.18 Solution

```

function y=psola_timestretch(x,nis,alpha)
%N: block length; nis: pitch marks; alpha: stretch factor;

P = diff(nis); %compute pitch periods
%%%%% remove first and last pitch marks %%%
if( nis(1)<=P(1) ) nis=nis(2:length(nis)); P=P(2:length(P)); end
if( nis(length(nis))+P(length(P))>length(x) ) nis=nis(1:length(nis)-1);
else P=[P P(length(P))]; end

y=zeros(1, ceil(length(x)*alpha +max(P)) ); %initialize output signal
nk = P(1)+1; %initialize output pitch mark
while round(nk)<length(y)
    [minimum i] = min( abs(alpha*nis - nk) ); %find analysis segment
    xi = x(nis(i)-P(i):nis(i)+P(i)).*hanning(2*P(i)+1); %take analysis frame
    %% overlap the frame to the output signal%%
    y(round(nk)-P(i):round(nk)+P(i)) = y(round(nk)-P(i):round(nk)+P(i))+xi;
    nk=nk+P(i);
    display('done')
end

```

Clearly we have omitted the most difficult part of the PSOLA approach, i.e. an algorithm that determines the input pitch marks (the vector `nis` in our code).



2.2.3 Granular synthesis

The term “granular synthesis” can be used to define a family of sound synthesis approaches that share the basic idea of building complex sounds from simple ones. Granular synthesis assumes that a sound can be considered as a sequence, possibly with overlaps, of elementary acoustic elements called grains. Granular synthesis constructs complex and dynamic acoustic events starting from a large quantity of grains. The features of the grains and their temporal location determine the sound timbre.

One can make an analogy with cinema, where a rapid sequence of static images gives the impression of objects in movement. Another possible analogy is with the technique of mosaicing, where the grains are single small monochromatic plugs, and their juxtaposition produces a complex image. In music, the use of granular synthesis techniques arises from the experiences of taped electronic music. In the early years of electronic music, the tools that composers had at disposal (e.g., fixed waveform oscillators and filters) did not allow for substantial variations of sound timbres. However they were able to obtain dynamic sounds by cutting tapes into short sections and then putting them together again. The rapid alternation of acoustic elements provides a certain variety to the resulting sound.²

2.2.3.1 Gaborets

The initial idea of granular synthesis can be traced back to the work of the hungarian physicist Dennis Gabor, which was aimed at pinpointing the physical and mathematical ideas needed to understand what a time-frequency spectrum is. He considered sound as a sum of elementary Gaussian functions that have been shifted in time and frequency. Gabor considered these elementary functions as *acoustic quanta*, the basic constituents of a sound. These works have been rich in implications and have been the starting point for studying time-frequency representations and wavelet theory.

The usual Gabor expansion on a rectangular time-frequency lattice of a signal $x(t)$ can be expressed as a linear combination of “grains” $g_{mk}(t)$, that are shifted and modulated versions of a synthesis window $w(t)$

$$x(t) = \sum_m \sum_k a_{mk} g_{mk}(t), \quad \text{with } g_{mk}(t) = w(t - m\alpha T) e^{jk\beta \Omega t}. \quad (2.9)$$

Other names for these grains, or acoustic quanta, are *gaborets*, or *Gabor functions*, or *Gabor atoms*.

In Gabor formulation $w(t)$ is a gaussian window of the form

$$w(t) = \frac{1}{\sigma \sqrt{2\pi}} e^{-t^2/2\sigma^2}, \quad (2.10)$$

where σ is the standard deviation of the gaussian. An important property of this function is that it is possibly the only smooth, nonzero function, known in closed form, that is transformed to itself in the Fourier domain:

$$\mathcal{F}\{w\}(\omega) = e^{-t^2/(1/\sigma)^2}. \quad (2.11)$$

Therefore the grain g_{mk} has a gaussian shape both in time and in frequency: it is a gaussian bell in the time-frequency domain. As a particular case of the uncertainty principle, the width of the time-domain gaussian is inversely proportional to that of the frequency-domain gaussian, as shown by Eq. (2.11).

Historically, the use of granular synthesis in musical applications has been classified into two main approaches. The first one is based on the use of sampled sounds to construct grains, while the second one is based on the use of abstract, entirely synthetic grains.

²The greek composer Iannis Xenakis is generally acknowledged to have provided the first formulation of granular synthesis in his compositions *Analogique A et B* (1958-59).



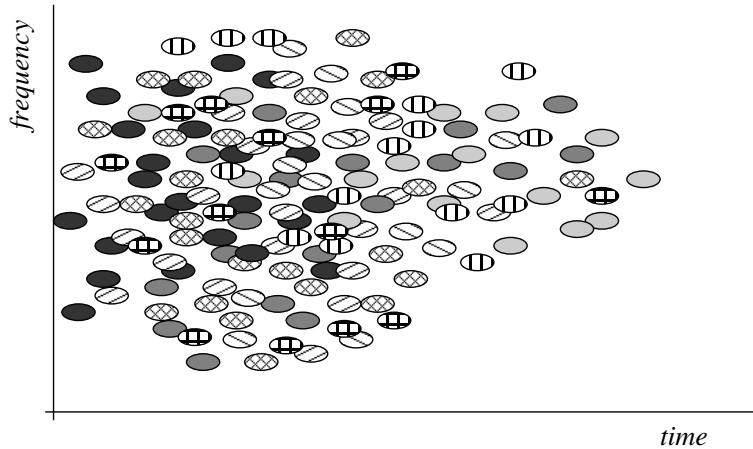


Figure 2.4: Representation of granular synthesis where grains derived from different sources are randomly mixed.

2.2.3.2 Sound granulation

The term *sound granulation* is typically used to identify the first of the two above mentioned approaches. Complex waveforms, extracted from real sounds or described by spectra, are organized in succession with partial overlap in time. In this way, it is possible both to reproduce accurately real sounds and modify them in their dynamic characteristics. The original sound $x[n]$ used to create the grains may have been previously recorded, or may be processed in real-time. In this respect granular synthesis can be viewed as an OLA technique in which segments $x_m[n]$ of a sound signal $x[n]$ represent the grains, and are processed both in time and frequency before being reassembled.

$$g_m[n] = x[n] \cdot w_m[n - n_m], \quad \text{with } n = n_m \dots n_m + L_m. \quad (2.12)$$

The time instant n_m indicates the point where the windowing starts and the segment is extracted; the length L_m of the window determines the amount of signal extracted; the shape of the window $w_m[n]$ should provide an amplitude envelope that ensures fade-in and fade-out at the border of the grain and affects the frequency content of the grain.

The length L_m is a critical parameter. Long grains tend to maintain the timbre identity of the portion of the input signal, while short ones acquire a pulse-like quality and frequency information is not perceivable any more. When the grain is long, the window has a flat top and it is used only to fade-in and fade-out the borders of the segment. Typical grain lengths are in the range 5 – 100 ms.

The organization of the choice of the frequencies is also very important, therefore in granular synthesis the proper timing organization of the grain is essential to avoid artifacts produced by discontinuities. This problem makes often the control quite difficult.

Notice that it is possible to extract grains from different sounds, to create hybrid textures, e.g. evolving from one texture to another. A schematic visualization of this approach is given in Fig. 2.4.

2.2.3.3 Synthetic grains

The second of the two above mentioned approaches is based on grains that consist of synthetic waveforms whose amplitude envelope is a short Gaussian function. Most typically, frequency modulated gaussian functions are used in order to localize the energy both in frequency and time domain, in the line of Gabor



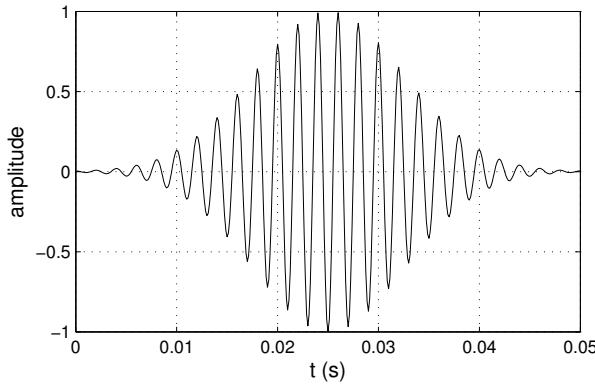


Figure 2.5: Example of a synthetic grain waveform, with frequency $\omega_k = 2\pi \cdot 500$ rad/s and standard deviation $\sigma = 0.2$.

work. In this case a single grain g_{mk} can be written as

$$g_{mk}[n] = w_{mk}[n - n_m] \cdot \cos\left(\omega_k \frac{n}{F_s} + \phi_k\right), \quad (2.13)$$

where the index m refers to the grain position in time, while the index k refers to its position in frequency. The window $w_{mk}[n]$ is a gaussian window shifted by n_m samples, and ω_k (in rad/s) is the frequency of the grain. A plot of a grain constructed in this way is provided in Fig. 2.5.

M-2.19

Write a function that computes a gaussian grain given its length, frequency, and standard deviation.

M-2.19 Solution

```
function yg = grain(L,f,sigma)
global Fs;
yg=(gausswin(L,1/sigma))'.*cos(2*pi*f*(1:L)/Fs);
```

The sound synthesized from these grains is again constructed according to Gabor formulation: it is an assemblage of grains scattered on the frequency-time plane in the form of “clouds”. The general synthesis expression is given by

$$s[n] = \sum_m \sum_k a_{mk} \cdot g_{mk}[n], \quad (2.14)$$

where a_{mk} is the amplitude coefficient of the corresponding grain. Every grain contributes to the total sound energy around the point (n_m, ω_k) of the time frequency plane.

In order to implement the above synthesis equation, the simplest approach amounts to define a constant density of grains in time, so that there is a constant time-step between the generation of a grain and the following one. This approach is often termed *synchronous granular synthesis*.

M-2.20

Write a script that realizes a synchronous granular synthesis scheme.

M-2.20 Solution



```

global Fs; Fs=22050;

gfreq=800;      frange=800; %grain freqs scattered around grainfreq
glength=0.05;   lrange=0.05; %grain lengths scattered around glength
gsigma=0.2;     sigrange=0.1; %gaussian st.dev. scattered around gsigma

slength=2;          %duration of the entire sound
gdens=100;          % grain density (=no. of grains per second)
totgrains=round(slength*gdens); %total no. of grains to be generated

y=zeros(1,(slength+2*glength)*Fs);
for m=0:totgrains-1
    f = gfreq +(rand(1)-.5)*frange;           % grain frequency
    t = round(Fs*(glength+(rand(1)-.5)*lrange)); % grain length (samples)
    d = gsigma +(rand(1)-.5)*sigrange;          %grain st.dev.
    yg = grain(t,f,d);                         %construct grain
    frame = round(m*Fs/gdens) +(1:t); %frame to be written (shifts with m)
    y(frame) = y(frame) + yg;                  %add current grain to sound
end

```

We have realized the synthesis in the time domain. Equivalent results could be obtained by working in the frequency domain.

However the most used type of granular synthesis is *asynchronous granular synthesis*, where grains are irregularly distributed in the time-frequency plane, e.g. they are scattered onto a mask that delimits specific portions of the time-frequency-amplitude space. This results in time-varying “clouds” of micro-sounds, or *sonic textures*, that can simulate even natural noisy sounds in which general statistical properties are more important than the exact sound evolution. Typical examples include the sound of numerous small objects (e.g., rice or sand) falling onto a resonating surface (e.g., a metal plate), or rain sounds composed by the accumulation of a large amount of water droplet micro-sounds, or even scratching/cracking sounds made by the accumulation of thousands of complex micro-sounds not necessarily deterministic. In general we can expect these types of sounds to occur in the real world when they are the result of multiple realizations of the same event or the same phenomenon.

Musical composers have tried to evaluate the effects of different control parameters from an aesthetic point of view. Grain duration affects the sonic texture: short duration (few samples) produces a noisy, particulate disintegration effect; medium duration (tens of ms) produces fluttering, warbling, gurgling; longer durations (hundreds of ms) produce aperiodic tremolo, jittering spatial position. When the grains are distributed on a large frequency region, the texture has a massive character, while when the band is quite narrow, it result a pitched sound. Sparse densities (e.g. 5 grains per second) give rise to a pointillistic texture.

2.2.3.4 Corpus-based concatenative synthesis

In recent years, synthesis methods conceptually similar to granular techniques have received a new impulse due to the availability of ever larger databases of sounds. Various definitions are used in the literature, including *concatenative synthesis*, *audio mosaicing*, and *musaicing* (neologism from music and mosaicing). All works in this direction share the general idea that a target sound can be approximated by samples taken from a pre-existing corpus of sounds.

Besides granular synthesis, the closest relative of this idea is in the field of speech synthesis: *concatenative speech synthesis* started to develop in the early sixties and is currently the most used synthesis approach in text-to-speech systems. In short, written text is automatically segmented into



elementary phonetic units that are subsequently synthesized using a large database of sampled speech sounds. These components are pieced together to obtain a synthesis of the text.

Central point: how to properly describe both the target sound and the sounds in the database, in order to define measures of similarity. We need high-level *sound descriptors*. Sounds in the database can be segmented into units (e.g. an instrument sound can be subdivided into attack, sustain, and release portions), and some kind of *unit selection algorithm* has to be realized that finds the sequence of units that match best the target sound or phrase to be synthesized. The selection will be performed according to the descriptors of the units. The selected units can then be transformed to fully match the target specification, and are concatenated.

2.3 Time-frequency models

Consider a sound signal $x[n]$. A time-frequency representation of $x[n]$ can be seen as a series of overlapping DFTs, typically obtained by windowing x in the desired frame. More precisely, we have that a frame $X_m(e^{j\omega_k})$ of the STFT is the DFT of the signal windowed segment $x_m[n] = x[n]w_a[n - mS_a]$, where $w_a[n]$ is the chosen analysis window and S_a is the *analysis hop-size*, i.e. the time-lag (in samples) between one analysis frame and the following one. If the window w_a is N samples long, then the *block size*, i.e. the length of each frame X_m , will be N . In order for the signal segments to actually overlap, the inequality $S_a \leq N$ must be verified. When $S_a = N$ the segments are exactly juxtaposed with no overlap.

Given the above signal segmentation, OLA methods are typically used to modify and reconstruct the signal in two main steps:

1. Any desired modification is applied to the spectra (e.g. multiplying by a filter frequency response function), and modified frame spectra $Y_m(e^{j\omega_k})$ are obtained.
2. Windowed segments $y_m[n]$ of the modified signal $y[n]$ are obtained by computing the inverse DFT (IDFT) of the frames Y_m .
3. The output is reconstructed by overlapping-adding the windowed segments: $y[n] = \sum_m y_m[n]$.

In their most general formulation OLA methods utilize a *synthesis* window w_s that can in general be different from the analysis window w_a . In this case the second step of the procedure outlined above is modified as follows:

2. Windowed segments $y_m[n]$ of the modified signal $y[n]$ are obtained by (**a**) computing the inverse DFT (IDFT) of the frames Y_m , (**b**) dividing by the analysis window w_a (assuming that it is non-zero for all samples), and (**c**) multiplying by the synthesis window.

This approach provides greater flexibility than the previous one: the analysis window w_a can be chosen only on the basis of its time-frequency resolution properties, but needs not to satisfy the “sum-to-unity” condition $A_{w_a} \equiv 1$. On the other hand, the synthesis window w_s is only used to cross-fade between signal segments, therefore one should only ensure that $A_{w_s} \equiv 1$. We will see in section 2.4.1 an application of this technique to frequency-domain implementation of additive synthesis.

Many digital sound effects can be obtained by employing OLA techniques. As an example, a *robotization* effect can be obtained by putting zero phase values on every FFT before reconstruction: the effect applies a fixed pitch onto a sound and moreover, as it forces the sound to be periodic, many erratic and random variations are converted into “robotic” sounds. Other effects are obtained by imposing a random phase on a time-frequency representation, with different behaviors depending on the block length N : if



N is large (e.g. $N = 2048$ with $F_s = 44.1$ kHz), the magnitude will represent the behavior of the partials quite well and changes in phase will produce an uncertainty over the frequency; if N is small (e.g. $N = 64$ with $F_s = 44.1$ kHz), the spectral envelope will be enhanced and this will lead to a whispering effect.

2.4 Spectral models

Since the human ear acts as a particular spectrum analyser, an important class of synthesis models aims at modeling and generating sound spectra. The Short Time Fourier Transform and other time-frequency representations provide powerful sound analysis tools for computing the time-varying spectrum of a given sound.

This section presents models that can be interpreted in the frequency domain. In computer music this models have been traditionally called *additive synthesis*, since they regard a time-varying sound as a sum of sinusoidal components with time-varying amplitudes and frequencies. Note that the idea behind additive synthesis is not new. As a matter of fact, it has been used for centuries in some traditional instruments such as organs. Organ pipes, in fact, produce relatively simple sounds that, when combined together, contribute to the richer spectrum of some registers. Particularly rich registers are created by using many pipes of different pitch at the same time.

In Sec. 2.4.1 we discuss sinusoidal modeling of the deterministic sound signal and introduce the main concepts of additive synthesis. In Sec. 2.4.2 we discuss a *synthesis-by-analysis* procedure that allows to fit parameters of a sinusoidal model to a target sound. Finally, in Secs. 2.4.3 and 2.4.4 we address the problem of including stochastic components and fast transients into the framework of additive models.

2.4.1 Sinusoidal model

Spectral analysis of the sounds produced by musical instruments, or by any physical system, shows that the spectral energy of the sound signals can be interpreted as the sum of two main components: a *deterministic* component that is concentrated on a discrete set of frequencies, and a *stochastic* component that has a broadband characteristics.

The deterministic –or sinusoidal– component normally corresponds to the main modes of vibration of the system. The stochastic residual accounts for the energy produced by the excitation mechanism which is not turned into stationary vibrations by the system, and for any other energy component that is not sinusoidal.

As an example, consider the sound of a wind instrument: the deterministic signal results from self-sustained oscillations inside the bore, while the residual noisy signal is generated by the turbulent flow components due to air passing through narrow apertures inside the instrument. Similar considerations apply to other classes of instruments, as well as to voice sounds, and even to non-musical sounds.

2.4.1.1 Time-varying partials

The term *deterministic* signal means in general any signal that is not noise. The class of deterministic signals that we consider here is restricted to sums of sinusoidal components with varying amplitude and frequency. For pitched sounds in particular, spectral energy is mainly concentrated at a few discrete (slowly time-varying) frequencies f_k . These frequency lines correspond to different sinusoidal components called *partials*. The amplitude a_k of each partial is not constant and its time-variation is critical for timbre characterization. If there is a good degree of correlation among the frequency and amplitude variations of different partials, these are perceived as fused to give a unique sound with its timbre identity.



Amplitude and frequency variations can be noticed e.g. in sound attacks: some partials that are relevant in the attack can disappear in the stationary part. In general, the frequencies can have arbitrary distributions: for quasi-periodic sounds the frequencies are approximately harmonic components (integer multiples of a common fundamental frequency), while for non-harmonic sounds (such as that of a bell) they have non-integer ratios.

The deterministic part of a sound signal can be represented by the *sinusoidal model*, which assumes that the sound can be modeled as a sum of sinusoidal oscillators whose amplitude $a_k(t)$ and frequency $f_k(t)$ are slowly time-varying:

$$\begin{aligned} s_s(t) &= \sum_k a_k(t) \cos(\phi_k(t)), \\ \phi_k(t) &= 2\pi \int_0^t f_k(\tau) d\tau + \phi_k(0), \end{aligned} \quad (2.15)$$

or, in the discrete-time domain:

$$\begin{aligned} s_s[n] &= \sum_k a_k[n] \cos(\phi_k[n]), \\ \phi_k[n] &= 2\pi f_k[n] T_s + \phi_k[n-1], \end{aligned} \quad (2.16)$$

where T_s is the sampling period. Equation (2.16) can also be written as

$$\phi_k[n] = \phi_{0k} + 2\pi T_s \sum_{j=1}^n f_k[j], \quad (2.17)$$

where ϕ_{0k} represents an initial phase value. These equations have great generality and can be used to faithfully reproduce many types of sound, especially in a “synthesis-by-analysis” framework (that we discuss in Sec. 2.4.2 below). If the sound is almost periodic, the frequencies of partials are approximately multiples of the fundamental frequency f_0 , i.e. $f_k(t) \simeq k f_0(t)$. In this sense Eqs. (2.16) are a generalization of the Fourier theorem to quasi-periodic sounds. Moreover the model is also capable of representing aperiodic and inharmonic sounds, as long as their spectral energy is concentrated near discrete frequencies (spectral lines).

As already noted, one limitation of the sinusoidal model is that it discards completely the noisy components that are always present in real signals. Another drawback of Eq. (2.16) is that it needs an extremely large number of control parameters: for each note that we want to reproduce, we need to provide the amplitude and frequency envelopes for all the partials. Moreover, the envelopes for a single note are not fixed, but depend in general on the intensity.

On the other hand, additive synthesis provides a very intuitive sound representation, and this is one of the reasons why it has been one of the earliest popular synthesis techniques in computer music.³ Moreover, sound transformations performed on the parameters of the additive representation (e.g., time-scale modifications) are perceptually very robust.

2.4.1.2 Time- and frequency-domain implementations

Additive synthesis with equation (2.16) can be implemented either in the time domain or in the frequency domain. The more traditional time-domain implementation uses the digital sinusoidal oscillator in wavetable or recursive form, as discussed in Chapter *Fundamentals of digital audio processing*. The instantaneous

³Some composers have even used additive synthesis as a compositional metaphor, in which sound spectra are reinterpreted as harmonic structures.



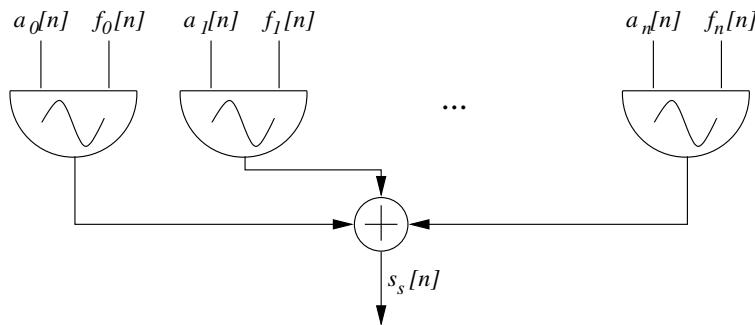


Figure 2.6: Sum of sinusoidal oscillators with time-varying amplitudes and frequencies.

amplitude and the instantaneous angular frequency of a particular partial are obtained by linear interpolation, as discussed there. Figure 2.6 provides a block diagram of such a time-domain implementation.

M-2.21

Use the sinusoidal oscillator realized in Chapter *Fundamentals of digital audio processing* to synthesize a sum of two sinusoids.

M-2.21 Solution

```
global Fs; global SpF; %global variables: sample rate, samples-per-frame

Fs=22050;
framelength=0.01; %frame length (in s)
SpF=round(Fs*framelength); %samples per frame

%%%%% define controls %%%%%%
a=envgen([0,.5,5,10,15,19.5,20],[0,1,1,1,1,1,0]); %fade in/out
f1=envgen([0,20],[200,200]); %constant freq. envelope
f2=envgen([0,1,5,10,15,20],... %increasing freq. envelope
           [200,200,205,220,270,300]);
%%%%% compute sound %%%%%%
s=sinosc(0,a,f1,0)+sinosc(0,a,f2,0);
```

The sinusoidal oscillator controlled in frequency and amplitude is the fundamental building block for time-domain implementations of additive synthesis. Here we employ it to look at the beating phenomenon. We use two oscillators, of which one has constant frequency while the second is given a slowly increasing frequency envelope. Figure 2.7 shows the f_1 , f_2 control signals and the amplitude envelope of the resulting sound signal: note the beating effect.

In alternative to the time-domain approach, a very efficient implementation of additive synthesis can be developed in the frequency domain, using the inverse FFT. As we have seen in Chapter *Fundamentals of digital audio processing*, the DFT of a windowed sinusoid is the DFT of the window, centered at the frequency of the sinusoid, and multiplied by a complex number whose magnitude and phase are the magnitude and phase of the sine wave:

$$s[n] = a \cos(2\pi f_0 n / F_s + \phi) \Rightarrow \mathcal{F}[w \cdot s](f) = ae^{j\phi}W(f - f_0). \quad (2.18)$$

If the window $W(f)$ has a sufficiently high sidelobe attenuation, the sinusoid can be generated in the spectral domain by calculating the samples in the main lobe of the window transform, with the appropriate magnitude, frequency and phase values. One can then synthesize as many sinusoids as desired,



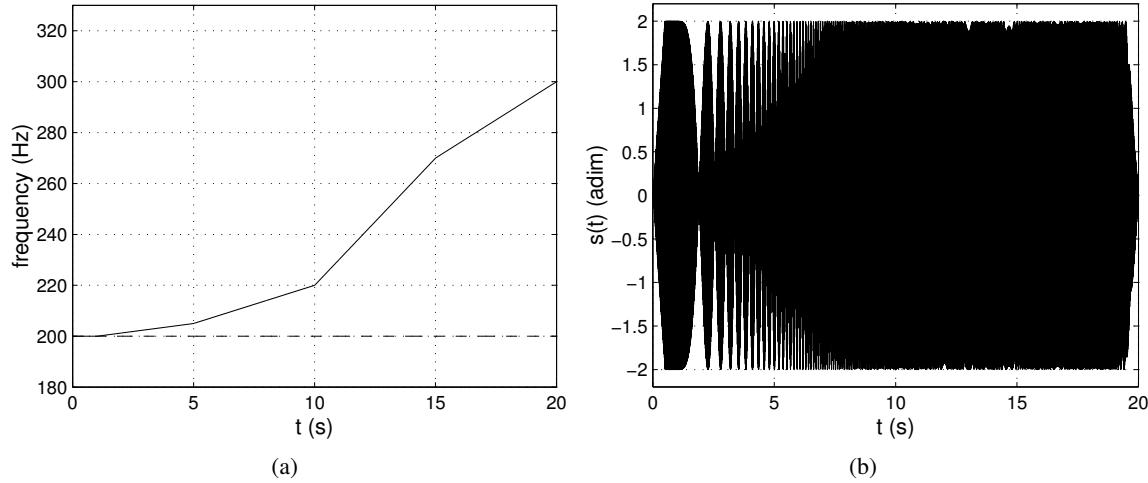


Figure 2.7: Beating effect: (a) frequency envelopes (f_1 dashed line, f_2 solid line) and (b) envelope of the resulting signal.

by adding a corresponding number of main lobes in the Fourier domain and performing a single IFFT to obtain the corresponding time-domain signal in a frame.

By an overlap-add process one then obtains the time-varying characteristics of the sound. Note however that, in order for the signal reconstruction to be free of artifacts, the overlap-add procedure must be carried out using a window with the property that its shifted copies overlap and add to give a constant. A particularly simple and effective window that satisfies this property is the triangular window.

The FFT-based approach can be convenient with respect to time-domain techniques when a very high number of sinusoidal components must be reproduced: the reason is that the computational costs of this implementation are largely dominated by the cost of the IFFT, which does not depend on the number of components. On the other hand, this approach is less flexible than the traditional oscillator bank implementation, especially for the instantaneous control of frequency and magnitude. Note also that the instantaneous phases are not preserved using this method. A final remark concerns the FFT size: in general one wants to have a high frame rate, so that frequencies and magnitudes need not to be interpolated inside a frame. At the same time, large FFT sizes are desirable in order to achieve good frequency resolution and separation of the sinusoidal components. As in every short-time based processes, one has to find a trade-off between time and frequency resolution.

2.4.2 Synthesis by analysis

As already remarked, additive synthesis allows high quality sound reproduction if the amplitude and frequency control envelopes are extracted from Fourier analysis of real sounds. Figure 2.8 shows the result of this kind of analysis, in the case of a saxophone tone. Using these data, additive resynthesis is straightforward.

M-2.22

Assume that two matrices `sinan_freqs` and `sinanamps` have been created from analysis of a real sound. These matrices contain frequency and amplitude envelopes of sinusoidal partials of the analyzed sound. Write a function that resynthesizes the sound.

M-2.22 Solution



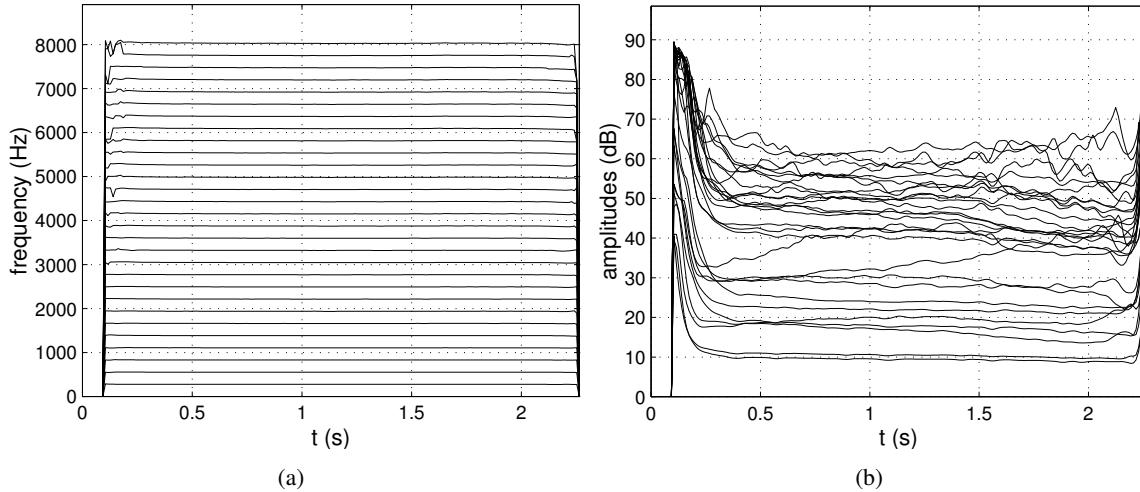


Figure 2.8: Fourier analysis of a saxophone tone: (a) frequency envelopes and (b) amplitude envelopes of the sinusoidal partials, as functions of time.

```
function s=sin_resynth(sinan_freqs, sinan_amps)

global Fs; global SpF; %global variables: sample rate, samples-per-frame

%%% define controls (analysis matrices sinan_freqs and
%%% sinan_amps have been created in the analysis phase)
npart=size(sinan_amps,1); %number of analyzed partials
t0=0; %initial time

%%% compute sound %%%
s=0; %initialize output signal
for (i=1:npart) %generate all partials and sum
    s=s+sinosc(t0,sinan_amps(i,:),sinan_freqs(i,:),0);
end
```

2.4.2.1 Magnitude and Phase Spectra Computation

The first step of any analysis procedure that tracks frequencies and amplitudes of the sinusoidal components is the frame-by-frame computation of the sound magnitude and phase spectra, through the STFT. The subsequent tracking procedure will be performed in the frequency domain. The control parameters for the STFT are the window-type and size, the FFT-size, and the frame-rate. These must be set depending on the sound to be processed.

Note that the analysis step is completely independent from the synthesis, therefore the observations made in Sec. 2.4.1 about FFT-based implementations (the window must overlap and add to a constant) do not apply here. Good resolution of the spectrum is needed in order to correctly resolve, identify, and track the peaks which correspond to the deterministic component.

If the analyzed sound is almost stationary, long windows (i.e. windows that cover several periods) that have good side-lobe rejection can be used, with a consequent good frequency resolution. Unfortunately most interesting sounds are not stationary and a trade-off has to be found. For harmonic sounds one can scale the actual window size as a function of pitch, thus achieving a constant time-frequency trade-off.



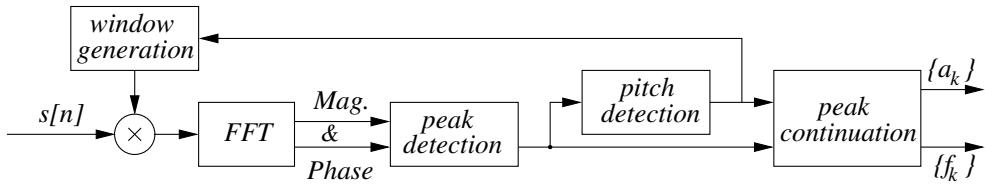


Figure 2.9: Block diagram of the sinusoid tracking process, where $s[n]$ is the analyzed sound signal and a_k , f_k are the estimated amplitude and frequency of the k th partial in the current analysis frame.

For inharmonic sounds the size should be set according to the minimum frequency difference that exists between partials.

The question is now how to perform automatic detection and tracking of the spectral peaks that correspond to sinusoidal components. In the next section we present the main guidelines of a general analysis framework, which is summarized in Fig. 2.9. First, the FFT of a sound frame is computed according to the above discussion. Next, the prominent spectral peaks are detected and incorporated into partial trajectories. If the sound is pseudo-harmonic, a pitch detection step can improve the analysis by providing information about the fundamental frequency information, and can also be used to choose the size of the analysis window.

Such a scheme is only one of the possible approaches that can be used to attack the problem. Hidden Markov Models (HMMs) are another one: a HMM can optimize groups of peaks trajectories according to given criteria, such as frequency continuity. This type of approach might be very valuable for tracking partials in polyphonic sounds and complex inharmonic tones.

2.4.2.2 A sinusoid tracking procedure

We now discuss in more detail the analysis steps depicted in Fig. 2.9.

The first step is *peak detection*. The most prominent frequency peaks (i.e., local maxima in the magnitude spectrum) in the current analysis frame are identified in this step. Real sounds are not periodic, do not have clearly spaced and defined spectral peaks, exhibit interactions between components. Therefore, the best that one can do at this point is to detect as many peaks as possible and postpone to later analysis steps the decision of which ones actually correspond to sinusoidal components. The peaks are then searched by only imposing two minimal constraints: they have to lie within a given frequency range, and above a given magnitude threshold. The detection of very soft peaks is hard: they have little resolution, and measurements are very sensitive to transformations because as soon as modifications are applied to the analysis data, parts of the sound that could not be heard in the original can become audible.

In ideal conditions, i.e. if the analyzed sound is very clean and has with the maximum dynamic range, the magnitude threshold can be set to the amplitude of the background noise floor. One possible strategy to decide whether a peak estimated in this way actually belongs to a sinusoidal partial or not is to measure how close the peak shape is to the ideal sinusoidal peak: the difference from the samples of the measured peak to the samples of the analysis window transform (centered at the measured frequency and scaled to the measured magnitude) provides a measure of how likely the peak is to belong to a sinusoidal component. Another refinement to the peak detection stage is to pre-process the sound in order to introduce *pre-emphasis* in the high frequency range: this allows to gain better resolution in the high frequency range. If applied, pre-emphasis will need to be compensated later on before the resynthesis.

The second step in Fig. 2.9 is *pitch detection*. Pitch is a valuable source of additional information in order to decide whether a particular peak belongs to a sinusoidal partial or not. If a fundamental

frequency is actually present, it can be exploited in two ways. First, it helps the tracking of partials. Second, the size of the analysis window can be set according to the estimated pitch in order to keep the number of periods-per-frame constant, therefore achieving the best possible time-frequency trade-off (this is an example of a *pitch-synchronous* analysis). There are many possible pitch detection strategies, which will be presented in Chapter *From audio to content*.

The third step in Fig. 2.9 is a *peak continuation* algorithm. The basic idea is that a set of “guides” advance in time and follow appropriate frequency peaks, forming trajectories out of them. A guide is therefore an abstract entity which is used by the algorithm to create the trajectories, and the trajectories are the actual result of the peak continuation process. The guides are turned on, advanced, and finally turned off during the continuation algorithm, and their instantaneous state (frequency and magnitude) is continuously updated during the process. If the analyzed sound is harmonic and a fundamental has been estimated, then the guides are created at the beginning of the analysis, with frequencies set according to the estimated harmonic series. When no harmonic structure can be estimated, each guide is created when the first available peak is found. In the successive analysis frames, the guides modify their status depending on the last peak values. This past information is particularly relevant when the sound is not harmonic, or when the harmonics are not locked to each other and we cannot rely on the fundamental as a strong reference for all the harmonics.

The main guidelines to construct a peak continuation algorithm can be summarized as follows. A peak is assigned to the guide that is closest to it and that is within an assigned frequency deviation. If a guide does not find a match, the corresponding trajectory can be turned off, and if a continuation peak is not found for a given amount of time the guide is killed. New guides and trajectories can be created starting from peaks of the current frame that have high magnitude and are not “claimed” by any of the existing trajectories. After a certain number of analysis frames, the algorithm can look at the trajectories created so far and adopt corrections: in particular, short trajectories can be deleted, and small gaps in longer trajectories can be filled by interpolating between the values of the gap edges.

One final refinement to this process can be added by noting that the sound attack is usually highly non-stationary and noisy, and the peak search is consequently difficult in this part. Therefore it is customary to perform the whole procedure backwards in time, starting from the end of the sound (which is usually a more stable part). When the attack is reached, a lot of relevant information has already been gained and non-relevant peaks can be evaluated and/or rejected.

2.4.3 “Sines-plus-noise” models

At the beginning of our discussion on additive modeling, we remarked that the spectral energy of the sound signals has a *deterministic* component that is concentrated on a discrete set of frequencies, and a *stochastic* component that has a broadband characteristics. So far we have discussed the problem of modeling the deterministic –or sinusoidal– component. Now we have to include the stochastic component into the model.

A sinusoidal representation may in principle be used also to simulate noise, since noise consists of sinusoids at every frequency within the band limits. It is clear however that such a representation would be computationally very demanding. Moreover it would not be a *flexible* sound representation. Therefore the most convenient sound model is of the form

$$s[n] = s_s[n] + e[n], \quad (2.19)$$

where $s_s[n]$ represents the deterministic part of the sound and has already been modeled with Eq. (2.16), while $e[n]$ represents the stochastic component and is modeled separately from $s_s[n]$.

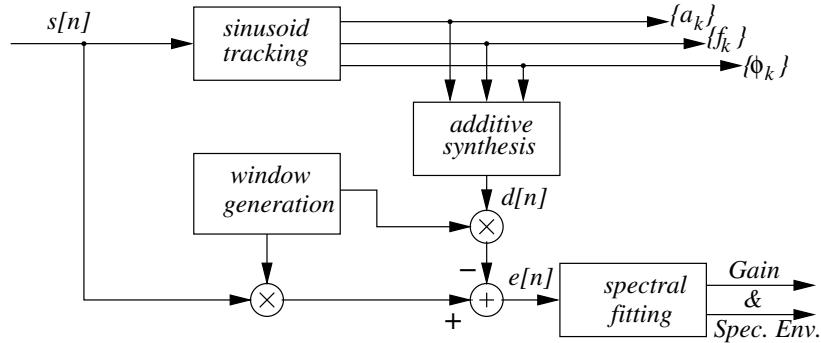


Figure 2.10: Block diagram of the stochastic analysis and modeling process, where $s[n]$ is the analyzed sound signal and a_k , f_k , ϕ_k are the estimated amplitude, frequency, and phase of the k th partial in the current analysis frame.

2.4.3.1 Stochastic analysis

The most straightforward approach to estimation of the stochastic component is through *subtraction* of the deterministic component from the original signal. Subtraction can be performed either in the time domain or in the frequency domain. Time domain subtraction must be done while preserving the phases of the original sound, and instantaneous phase preservation can be computationally very expensive. On the other hand, frequency-domain subtraction does not require phase preservation. However, time-domain subtraction provides much better results, and is usually favored despite the higher computational costs. For this reason we choose to examine time-domain subtraction in the remainder of this section. Figure 2.10 provides a block diagram.

Suppose that the deterministic component has been estimated in a given analysis frame, using for instance the general scheme described in section 2.4.2 (note however that in this case the analysis should be improved in order to provide estimates of the instantaneous phases as well). Then the first step in the subtraction process is the time-domain resynthesis of the deterministic component with the estimated parameters. This should be done by properly interpolating amplitude, frequency, and phase values in order to avoid artifacts in the resynthesized signal. The actual subtraction can be performed as

$$e[n] = w[n] \cdot [s[n] - d[n]], \quad n = 0, \dots, N - 1, \quad (2.20)$$

where $s[n]$ is the original sound signal and $d[n]$ is the re-synthesized deterministic part. The difference $(s - d)$ is multiplied by an analysis window w of size N , which deserves some discussion.

We have seen in 2.4.2 that high frequency resolution is needed for the deterministic part, and for this reason long analysis windows are used for its estimation. On the other hand, good time resolution is more important for the stochastic part of the signal, especially in sound attacks, while frequency resolution is not a major issue for noise analysis. A way to obtain good resolutions for both the components is to use two different analysis windows. Therefore w in equation (2.20) is not in general the same window used to estimate $d[n]$, and the size N is in general small.

Once the subtraction has been performed, there is one more step than can be used to improve the analysis, namely, test can be performed on the estimated residual in order to assess how good the analysis was. If the spectrum of the residual still contains some partials, then the analysis of the deterministic component has not been performed accurately and the sound should be re-analyzed until the residual is free of deterministic components. Ideally the residual should be as close as possible to a stochastic

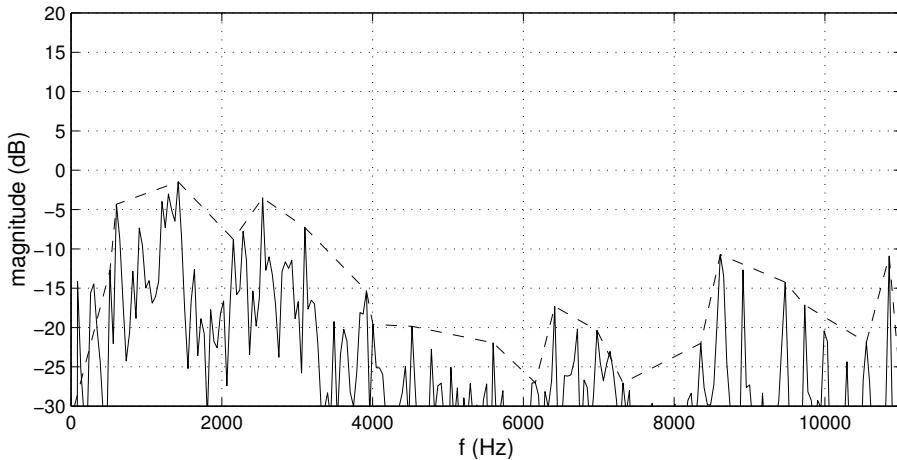


Figure 2.11: Example of residual magnitude spectrum (solid line) and its line-segment approximation (dashed line), in an analysis frame. The analyzed sound signal is the same saxophone tone used in figure 2.8.

signal, therefore one possible test is a measure of correlation of the residual samples.⁴

2.4.3.2 Stochastic modeling

The assumption that the residual is a stochastic signal implies that it is fully described by its amplitude and its spectral envelope characteristics. Information on the instantaneous phase is not necessary. Based on these considerations, a frame of the stochastic residual can be completely characterized by a filter that models the amplitude and general frequency characteristics of the residual. The representation of the residual for the overall sound will then be a time-varying filter.

Within a given frame we therefore assume that $e[n]$ can be modeled as

$$E[k] = H[k] \cdot U[k], \quad k = 0, \dots, N - 1, \quad (2.21)$$

where $U[k]$ is the DFT of a white noise sequence and $H[k]$ represents the frequency response of a filter which varies on a frame-by-frame basis. The stochastic modeling step is summarized in the last block of figure 2.10.

The filter design problem can be solved using different strategies. One approach that is often adopted uses some sort of curve fitting (line-segment approximation, spline interpolation, least squares approximation, and so on) of the magnitude spectrum of e in an analysis frame. As an example, line-segment approximation can be obtained by stepping through the magnitude spectrum, finding local maxima at each step, and connecting the maxima with straight lines. This procedure can approximate the spectral envelope with reasonable accuracy, depending on the number of points, which in turn can be set depending on the sound complexity. See Fig. 2.11 for an example.

Another possible approach to the filter design problem is Linear Prediction (LP) analysis, which is a popular technique in speech processing. In this context, however, curve fitting on the noise spectrum (e.g., line-segment approximation) is usually considered to be a more flexible approach and is preferred to LP analysis. We will return on Linear Prediction techniques in section 2.5.3.

The next question is how to implement the estimated time-varying filter in the resynthesis step.

⁴ Note that if the analyzed sound has not been recorded in silent and anechoic settings the residual will contain not only the stochastic part of the sound, but also reverberation and/or background noise.



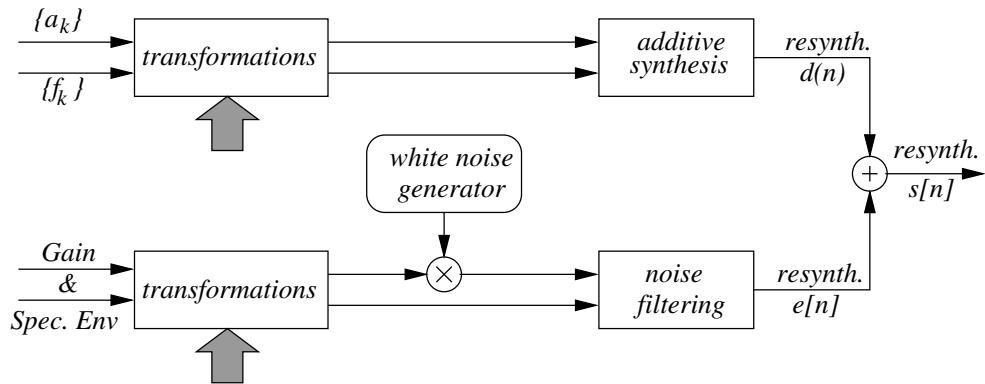


Figure 2.12: Block diagram of the sines-plus-noise synthesis process.

2.4.3.3 Resynthesis and modifications

Figure 2.12 shows the block diagram of the synthesis process. The deterministic signal, i.e., the sinusoidal component, results from the magnitude and frequency trajectories, or their transformation, by generating a sine wave for each trajectory (additive synthesis). As we have seen, this can either be implemented in the time domain with the traditional oscillator bank method or in the frequency domain using the inverse-FFT approach.

Concerning the stochastic component, a frequency-domain implementation is usually preferred to a direct implementation of the time-domain convolution (2.21), due to its computational efficiency⁵ and flexibility. In each frame, the stochastic signal is generated by an inverse-FFT of the spectral envelopes. Similarly to what we have seen for the deterministic synthesis in section 2.4.1, the time-varying characteristics of the stochastic signal is then obtained using an overlap-and-add process.

In order to perform the IFFT, a magnitude and a phase responses have to be generated starting from the estimated frequency envelope. Generation of the magnitude spectrum is straightforwardly obtained by first linearly interpolating the spectral envelope to a curve with half the length of the FFT-size, and then multiplying it by a gain that corresponds to the average magnitude extracted in the analysis. The estimated spectral envelope gives no information on the phase response. However, since the phase response of noise is noise, a phase response can be created from scratch using a random signal generator. In order to avoid periodicities at the frame rate, new random values should be generated at every frame.

The sines-plus-noise representation is well suited for modification purposes.

- By only working on the deterministic representation and modifying the amplitude-frequency pairs or the original sound partials, many kinds of frequency and magnitude transformations can be obtained. As an example, partials can be transposed in frequency. It is also possible to decouple the sinusoidal frequencies from their amplitude, obtaining pitch-shift effects that preserve the formant structure.
- Time-stretching transformations can be obtained by resampling the analysis points in time, thus slowing down or speeding up the sound while maintaining pitch and formant structure. Given the stochastic model that we are using, the noise remains noise and faithful signal resynthesis is possible even with extreme stretching parameters.

⁵ In fact, by using a frequency-domain implementation for both the deterministic and the stochastic synthesis one can add the two spectra and resynthesize both the components at the cost of a single IFFT per frame.

- By acting on the relative amplitude of the two components, interesting effects can be obtained in which either the deterministic or the stochastic parts are emphasized. As an example, the amount of “breathiness” of a voiced sound or a wind instrument tone can be adjusted in this way. One must keep in mind however that, when different transformations are applied to the two representations, the deterministic and stochastic components in the resulting signal may not be perceived as a single sound event anymore.
- Sound morphing (or *cross-synthesis* transformations can be obtained by interpolating data from two or more analysis files. This transformations are particularly effective in the case of quasi-harmonic sounds with smooth parameter curves.

2.4.4 Sinusoidal description of transients

So far we have seen how to extend the sinusoidal model by using a “sines-plus-noise” approach that explicitly describes the residual as slowly varying filtered white noise. Although this technique is very powerful, transients do not fit well into a filtered noise description, because they lose sharpness and are smeared. This consideration motivates us to handle transients separately.

One straightforward approach, that is sometimes used, is removing transient regions from the residual, performing the sines-plus-noise analysis, and adding the transients back into the signal. This approach obviously requires memory where the sampled transients must be stored, but since the transient residuals remain largely invariant throughout most of the range of an instrument, only a few residuals are needed in order to cover all the sounds of a single instrument. Although this approach works well, it is not flexible because there is no model for the transients. In addition, identifying transients as everything that is neither sinusoidal nor transient is not entirely correct. Therefore we look for a suitable transient model, that can be embedded in the additive description to obtain a “sines-plus-transients-plus-noise” representation.

2.4.4.1 The DCT domain

In the following we adopt a further modified version of the additive sound representation (2.16), in which the sound transients are explicitly modeled by an additional signal:

$$s[n] = s_s[n] + e_t[n] + e_r[n], \quad (2.22)$$

where $s_s[n]$ is the sinusoidal component, $e_t[n]$ is the signal associated to transients and $e_r[n]$ is the noisy residual. The transient model is based on a main undelying idea: we have seen that a slowly varying sinusoidal signal is impulsive in the frequency domain, and sinusoidal models perform short-time Fourier analysis in order to track slowly varying spectral peaks (the tips of the impulsive signals) over time. Transients are very much dual to sinusoidal components: they are impulsive in the time domain, and consequently they must be oscillatory in the frequency domain. Therefore, although transient cannot be tracked by a short-time analysis (because their STFT will not contain meaningful peaks), we can track them by performing sinusoidal modeling in a properly chosen frequency domain. The mapping that we choose to use is the one provided by the discrete cosine transform (DCT):

$$S[k] = \beta[k] \sum_{n=0}^{N-1} s[n] \cos \left[\frac{(2n+1)k\pi}{2N} \right], \quad \text{for } n, k = 0, 1, \dots, N-1, \quad (2.23)$$

where $\beta[0] = \sqrt{1/N}$ and $\beta[k] = \sqrt{2/N}$ otherwise. From equation (2.23) one can see that an ideal impulse $\delta[n - n_0]$ (i.e., a Kronecker delta function centered in n_0) is transformed into a cosine whose



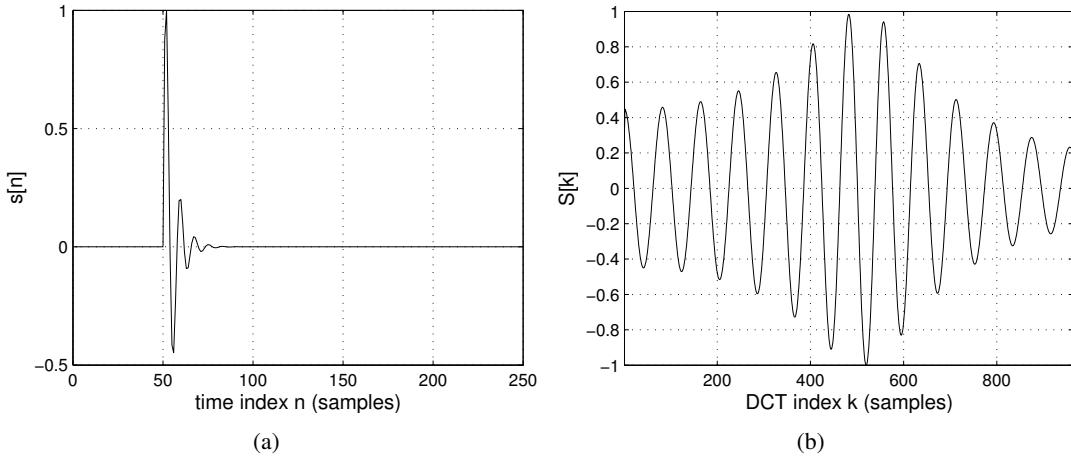


Figure 2.13: Example of DCT mapping: (a) an impulsive transient (an exponentially decaying sinusoid) and (b) its DCT as a slowly varying sinusoid.

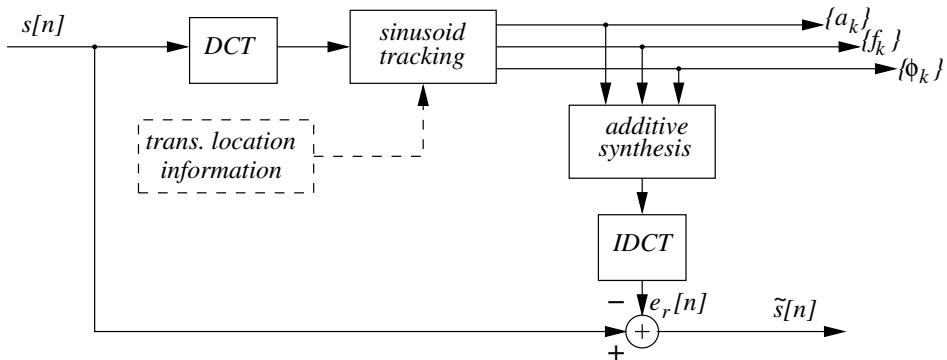


Figure 2.14: Block diagram of the transient analysis and modeling process, where $s[n]$ is the analyzed sound signal and a_k , f_k , ϕ_k are the estimated amplitude, frequency, and phase of the k th DCT-transformed transient in the current analysis frame.

frequency is monotonically related to n_0 . Figure 2.13(a) shows a more realistic transient signal, a one-sided exponentially decaying sine wave. Figure 2.13(b) shows the DCT of the transient signal: a slowly varying sinusoid. These considerations suggest that the time-frequency duality can be exploited to develop a transient model: the same kind of parameters that characterize the sinusoidal components of a signal can also characterize the transient components of a signal, although in a different domain.

2.4.4.2 Transient analysis and modeling

Having transformed the transient into the DCT domain, the most natural way to proceed is performing sinusoidal modeling in this domain: STFT analysis of the DCT-domain signal can be used to find meaningful peaks, and then the signal can be resynthesized in the DCT domain and back-transformed to the time domain with an inverse DCT transform (IDCT). This process is shown in figure 2.14. We now discuss the main steps involved in this block diagram.

First the input signal $s[n]$ is divided into non-overlapping blocks in which DCT analysis will be performed.

formed. The block length should be chosen so that a transient appears as “short”, therefore large block sizes (e.g., 1 s) are usually chosen. The block DCT is followed by a sinusoidal analysis/modeling process which is identical to what we have seen in section 2.4.2. The analysis can optionally embed some information about transient location within the block: there are many possible transient detection strategies, which we do not want to discuss here. Also, the analysis can perform better if the sinusoid tracking procedure starts from the end of the DCT-domain signal and moves backwards toward the beginning, because the beginning of a DCT frame is usually spectrally rich and this can deteriorate the performance of the analysis (similar considerations were made in Sec. 2.4.2 when discussing sinusoid tracking in the time domain).

The analysis yields parameters that correspond to slowly varying sinusoids in the DCT domain: each transient is associated to a triplet $\{a_k, f_k, \phi_k\}$, amplitude, frequency, and phase of the k th “partial” in each STFT analysis frame within a DCT block. By recalling the properties of the DCT one can see that f_k correspond to onset locations, a_k is the amplitude of the time-domain signal also, and ϕ_k is related to the time direction (positive or negative) in which the transient evolves. Resynthesis of the transients is then performed using these parameters to reconstruct the sinusoids in the DCT domain. Finally an inverse discrete cosine transform (IDCT) on each of the reconstructed signals is used to obtain the transients in each time-domain block, and the blocks are concatenated to obtain the transients for the entire signal.

It is relatively straightforward to implement a “fast transient reconstruction” algorithm. Without entering the details, we just note that the whole procedure can be reformulated using FFT transformations only: in fact one could verify that the DCT can be implemented using an FFT block plus some post-processing (multiplication of the real and imaginary parts of the FFT by appropriate cosinusoidal and sinusoidal signals followed by a sum of the two parts). Furthermore, this kind of approach naturally leads to a FFT-based implementation of the additive synthesis step (see Sec. 2.4.1).

One nice property of this transient modeling approach is that it fits well within the sines-plus-noise analysis examined in the previous sections. The processing block depicted in Fig. 2.14 returns an output signal $\tilde{s}[n]$ in which the transient components e_t have been removed by subtraction: this signal can be used as the input to the sines-plus-noise analysis, in which the remaining components (deterministic and stochastic) will be analyzed and modeled. From the implementation viewpoint, one advantage is that the core components of the transient-modeling algorithm (sinusoid tracking and additive resynthesis) are identical to those used for the deterministic model. Therefore the same processing blocks can be used in the two stages, although working on different domains.

2.5 Source-filter models

Some sound signals can be effectively modeled through a feed-forward source-filter structure, in which the source is in general a spectrally rich excitation signal, and the filter is a linear system that acts as a resonator and shapes the spectrum of the excitation.

A typical example is *voice*, where the periodic pulses or random fluctuations produced by the vocal folds are filtered by the vocal tract, that shapes the spectral envelope. The vowel quality and the voice color greatly depends on the resonance regions of the filter, called *formants*. In computer music, source-filter models are traditionally grouped under the label *subtractive synthesis*. A number of analog voltage controlled synthesizers in the 1960’s and 1970’s made use of subtractive synthesis techniques in which audio filters were applied to spectrally rich waveforms.

Source-filter models are often used in an analysis-synthesis framework, in which both the source signal and the filter parameters are estimated from a target sound signal, that can be subsequently resynthesized through the identified model. Moreover, transformations can be applied to the filter and/or the excitation before the reconstruction (see Fig. 2.15). One of the most common analysis techniques is



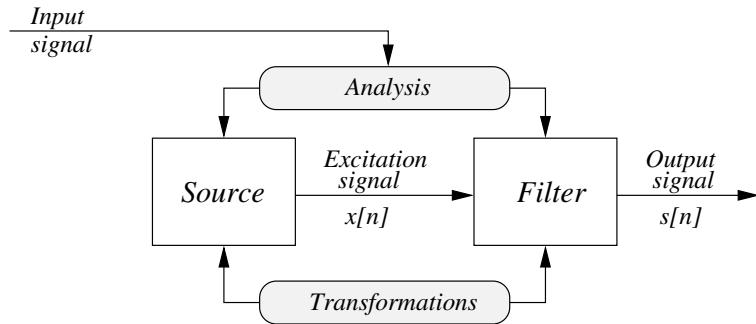


Figure 2.15: Source-filter model.

Linear Prediction, that we will address in Sec. 2.5.3.

2.5.1 Source signals and filter structures

We will assume the filter block to be linear and time-invariant (at least on a short-time scale), so that the excitation signal $x[n]$ and the output signal $s[n]$ are related through the difference equation

$$s[n] = \sum_{k=0}^M b_k x[n-k] - \sum_{k=1}^N a_k s[n-k], \quad (2.24)$$

or, in the z -domain,

$$S(z) = H(z)X(z), \quad \text{with} \quad H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{k=0}^M b_k z^{-k}}{\sum_{k=1}^N a_k z^{-k}}. \quad (2.25)$$

Equation (2.25) shows how the features of source and filter are combined: the spectral fine structure of the excitation signal is multiplied by the spectral envelope of the filter, which has a *shaping* effect on the source spectrum. Therefore, it is possible to control and modify separately different features of the signal: as an example, the pitch of a speech sound depends on the excitation and can be controlled separately from the formant structure, which instead depends on the filter. When the filter coefficients are (slowly) varied over time, the frequency response $H(e^{j\omega_d})$ changes. As a consequence, the output will be a combination of temporal variations of the input and of the filter (*cross-synthesis*).

2.5.1.1 Source signals

In order for the shaping effect of the filter to take place, the source signal must have a rich spectrum, that extends to a relevant portion of the audible frequency range. One important family of source signals are noise signals (see Chapter *Fundamentals of digital audio processing*), which have broadband spectral energy. One second important family are non-smooth periodic waveforms, whose spectral energy is concentrated in a (large) set of discrete spectral lines. This latter family includes square waves, sawtooth waves, and triangle waves, among others. Analog voltage-controlled synthesizers were typically equipped with a set of oscillators, which were able to synthesize these and possibly other waveforms.

An ideal *square wave* alternates periodically and instantaneously between two levels. An ideal *triangular wave* alternates periodically between a linearly rising portion and a linearly decreasing portion. An ideal *sawtooth wave* is a periodic series of linear ramps. Various formal definition of these signals (with



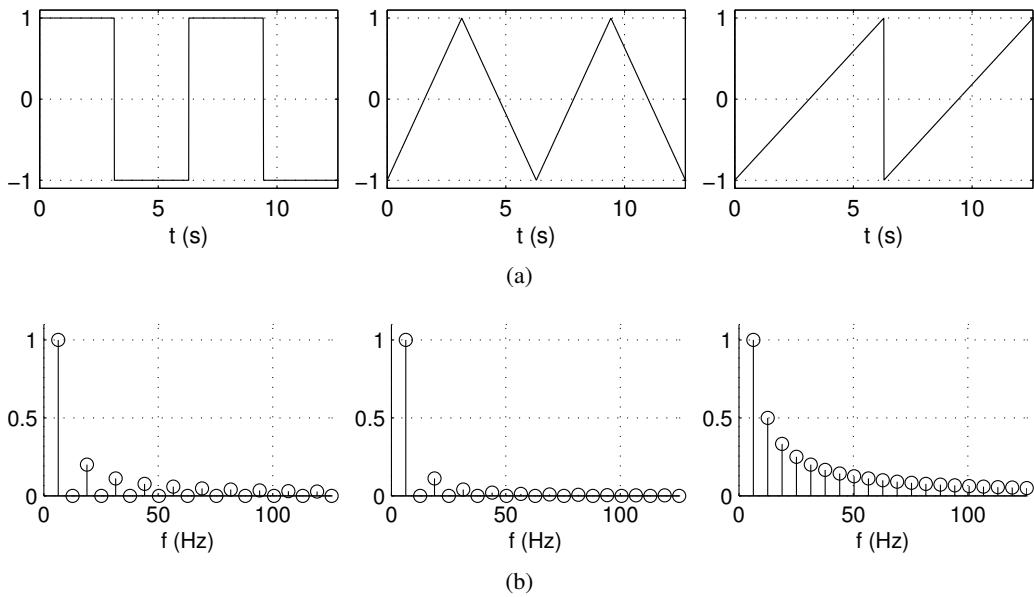


Figure 2.16: Spectrally rich waveforms: (a) time-domain square, triangular, sawtooth, and impulse train waveforms; (b) corresponding spectra (first 20 partials).

frequency f_0 Hz) can be conveniently given in the continuous-time domain.⁶ A set of compact possible definitions is the following:

$$\begin{aligned} x_{\text{square}}(t) &= \text{sgn} [\sin(2\pi f_0 t)], \quad x_{\text{triang}} = \frac{2}{\pi} \arcsin [\sin(2\pi f_0 t)] - 1, \\ x_{\text{saw}}(t) &= 2(f_0 t - \lfloor f_0 t \rfloor) - 1, \end{aligned} \quad (2.26)$$

where $\lfloor a \rfloor = \max\{n \in \mathbb{N} : n \leq a\}$ in the definition of the sawtooth wave indicates the floor function. These equations define waveforms that take values in the range $[-1, 1]$ and have zero average. The corresponding waveforms are depicted in Fig. 2.16(a).

These waveforms can be written in terms of the following sinusoidal expansions:

$$\begin{aligned} x_{\text{square}}(t) &= \frac{4}{\pi} \sum_{k=1}^{+\infty} \frac{\sin [(2k-1)2\pi f_0 t]}{2k-1}, \quad x_{\text{triang}}(t) = \frac{8}{\pi^2} \sum_{k=1}^{+\infty} (-1)^{k-1} \frac{\sin [(2k-1)2\pi f_0 t]}{k^2}, \\ x_{\text{saw}}(t) &= -\frac{2}{\pi} \sum_{k=1}^{+\infty} \frac{\sin (2\pi k f_0 t)}{k}. \end{aligned} \quad (2.27)$$

Note that here we are using an expansion on real sinusoids, which can be straightforwardly derived from the usual Fourier expansion on complex sinusoids. The corresponding spectra are shown in Fig. 2.16(b). As expected, all the waveforms are spectrally rich. In particular, the square and triangular waves contain only odd harmonics, with higher harmonics rolling off faster in the triangular than in the square wave (this is in accordance with the triangular wave being –and sounding– smoother than the square wave). On the other hand, the sawtooth wave has energy on all harmonics.

One more relevant source signal is the ideal *impulse train waveform*, a sequence of unit impulses spaced by the desired fundamental period. It is used especially for the simulation of voiced speech

⁶In fact continuous-time domain definitions are appropriate since they were used in analog synthesizers.

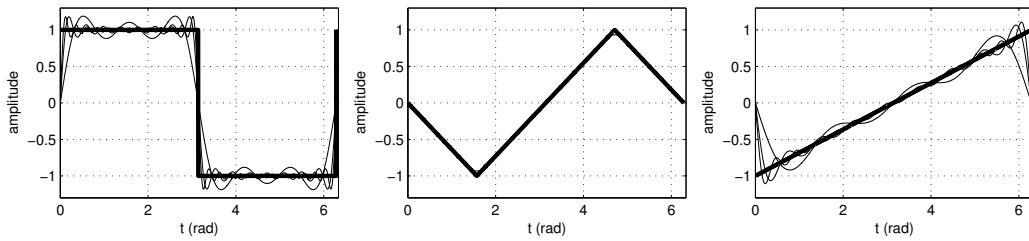


Figure 2.17: Bandlimited synthesis of the square, triangular, and sawtooth waves, using 3, 8, and 13 sinusoidal components.

sounds, and represents the periodic energy pulses provided by vocal fold movement to the vocal tract (we will return on this point in Sec. 2.5.2). It has a white spectrum.

2.5.1.2 Bandlimited digital oscillators

We have just seen that the spectra of square, triangular, and sawtooth waveforms are not bandlimited. For this reason, realizing digital oscillators able to synthesize these waveforms is not a completely trivial task because of potential aliasing problems. In fact, a simple wavetable implementation in which the waveforms are sampled from their theoretical definitions (2.26) would produce digital oscillators that are heavily affected by aliasing.

Spectral modifications introduced by aliasing deteriorate severely the sound quality, by introducing inharmonicity, beatings, and heterodyning. Inharmonicity is caused by the fact that aliased components will in general fall outside the harmonic series. In particular, some aliased components may have a smaller frequency than the fundamental frequency, thus altering the pitch. Some aliased components may instead fall in the vicinity of a harmonic component, thus generating beating. The phenomenon of heterodyning appears when the fundamental frequency of the oscillator is varied (e.g. in a vibrato): in this case the frequencies of some aliased components will move in the opposite direction with respect to the fundamental frequency, causing a characteristic timbral modulation.

It is therefore necessary to find aliasing-free implementations of such digital oscillator. The most straightforward approach amounts to exploiting the sinusoidal expansions given in Eqs. (2.27), in which only the partials with frequencies less than $F_s/2$ are employed. This is then a form of *bandlimited* additive synthesis, in which the number of sinusoidal components is chosen on the basis of the sampling rate.

Figure 2.17 shows an example of bandlimited additive synthesis in which different numbers of components have been used. Note that a low number of components already approximates closely the triangular wave, since this is a continuous function, while the square and sawtooth waves need higher number of components since they are discontinuous.

M-2.23

Write a function that realizes the generators for the square, triangular, sawtooth, and impulse train signals. The function will have parameters (t_0, a, f): initial time, amplitude envelope, and frequency envelope.

M-2.23 Solution

```
function s = waveosc(t0,a,f,ph0,wavetype,npart);
%%%%% param wavetype can be 'cos' | 'square' | 'triang' | 'saw' | 'imp' %%%%
global Fs; global SpF; %global variables: sample rate, samples-per-frame
```



```

npart = min(npart, floor(Fs/(2*max(f)))); % max. no of partials up to Nyquist
nframes=length(a); s=zeros(1,nframes*SpF); %initialize signal vector to 0
switch (wavetype)
    case {'cos'};
        s=sinosc(t0,a,f,ph0);
    case {'square'};
        for (k=1:npart) s=s+4/pi*sinosc(t0,a,(2*k-1)*f,ph0-pi/2)/(2*k-1); end
    case {'triang'};
        for (k=1:npart) s=s+8/pi^2*(-1)^k*sinosc(t0,a,(2*k-1)*f,ph0-pi/2)/(2*k-1)^2; end
    case {'saw'};
        for (k=1:npart) s=s-2/pi*sinosc(t0,a,k*f,ph0-pi/2)/k; end
    case {'imp'};
        for (k=1:npart) s= s + sinosc(t0,a,k*f,ph0)/(1+npart); end
end

```

This function utilizes the `sinosc` function written in Chapter *Fundamentals of digital audio processing*. We have used Eqs. (2.27) and for each waveform have summed up all the needed harmonic components up to the Nyquist frequency $F_s/2$.

2.5.1.3 Resonant filters

In Chapter *Fundamentals of digital audio processing* we have already examined some simple first-order low-pass and high-pass filters: these may be used in the “filter” block of Fig. 2.15. Another class of filters that is widely used in subtractive synthesis schemes is that of resonant filters. The second-order IIR filter is the simplest one, and is described by a transfer function with two complex conjugate poles:

$$H(z) = \frac{b_0}{1 + a_1 z^{-1} + a_2 z^{-2}} = \frac{b_0}{(1 - r \cdot e^{j\omega_c} z^{-1})(1 - r \cdot e^{-j\omega_c} z^{-1})}, \quad (2.28)$$

where r and $\pm\omega_c$ are the magnitude and phases of the poles, and the condition $r < 1$ must hold in order for the filter to be stable. If assume the filter to be causal then the impulse response is⁷

$$h[n] = \frac{b_0 r^n \sin [\omega_c(n+1)]}{\sin \omega_c} u[n], \quad (2.29)$$

where $u[n]$ is the unit step as usual. Therefore $h[n]$ is a right-sided, exponentially damped sinusoid, where r determines the decay time. An example of sequence $h[n]$ is shown in Fig. 2.18(a).

A pole $r \cdot e^{j\omega_c}$ causes a resonance to appear in the vicinity of ω_c , which becomes sharper and sharper as $r \rightarrow 1$. In order to choose appropriate parameter values for the filter, the first element to consider is the actual location ω_c of the *center frequency*, where the resonance occurs: it is close to ω_c but not exactly there. The resonance occurs when the first derivative of $|H(e^{j\omega_d})|$ goes to zero. By writing explicitly this condition (or, more conveniently, the equivalent $d/d\omega_d[1/|H|^2] = 0$), some straightforward algebra leads to the result

$$\cos \omega_c = \frac{1 + r^2}{2r} \cos \omega_c. \quad (2.30)$$

This equation provides a means to choose ω_c given ω_c . It says that ω_c is close to ω_c when r is close to 1, but becomes significantly different for small r .

The second element to consider is the sharpness of the resonance. A quantitative measure is the *half-power bandwidth* (*or simply bandwidth*) B of the resonance, defined as the width of the magnitude

⁷This equation can be derived by applying the partial fraction expansion technique seen in Chapter *Fundamentals of digital audio processing*.



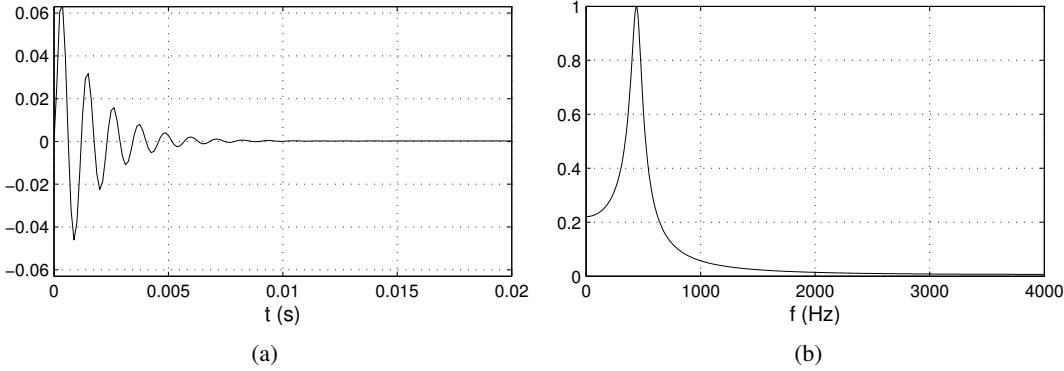


Figure 2.18: Example of a second-order resonator tuned on the center frequency $\omega_c = 2\pi 440/F_s$ and with bandwidth $B = 2\pi 100/F_s$; (a) impulse response; (b) magnitude response.

response at the two half-power points (the points where the magnitude response is $1/\sqrt{2}$ times smaller than at the peak value). An equivalent measure is the *quality factor* $q = \omega_c/B$.

A half-power point ω_{hp} is such that $|H(e^{j\omega_{hp}})/H(e^{j\omega_c})|^2 = 1/2$, by definition. If we assume that in the vicinity of one pole the effect of the second pole is negligible, we can derive a simple relation between B and r :

$$\begin{aligned} |H(e^{j\omega_c})|^2 &\sim \frac{b_o^2}{(1-r)^2}, \\ |H(e^{j\omega_{hp}})|^2 &\sim \frac{b_o^2}{|e^{j\omega_{hp}} - re^{j\omega_c}|^2} = \dots = \frac{b_o^2}{1 + r^2 - 2r \cos(\omega_{hp} - \omega_c)}. \end{aligned} \quad (2.31)$$

By noting that $\omega_{hp} - \omega_c \sim B/2$, and by applying the definition of half-power point, we can write

$$\frac{(1-r)^2}{1 + r^2 - 2r \cos(B/2)} = \frac{1}{2}, \quad \Rightarrow \quad \dots \quad \Rightarrow \quad \cos\left(\frac{B}{2}\right) = 2 - \frac{1}{2}\left(r + \frac{1}{r}\right). \quad (2.32)$$

This latter equation provides a means to choose r given B . A further useful approximation can be obtained for very sharp resonances, i.e. when $r = 1 - \epsilon$ with $\epsilon \ll 1$ and the poles are very close to the unit circle. Taylor expansions of the two sides of the equation give $\cos(B/2) \sim 1 - (B/2)^2$ and $2 - 1/2(r - 1/r)|_{r=(1-\epsilon)} \sim 1 - \epsilon^2/2$, respectively. Therefore in this limit one can write

$$B \sim 2\epsilon = 2(1-r). \quad (2.33)$$

In summary, given two values for ω_c and B , the poles can be determined using Eqs. (2.30) and (2.32) or (2.33). Then the coefficients can be written as functions of the parameters r , ω_c as

$$a_1 = -2r \cos(\omega_c), \quad a_2 = r^2, \quad b_0 = (1 - r^2) \sin^2(\omega_c), \quad (2.34)$$

where b_0 has been determined by imposing that $|H(e^{\pm j\omega_c})| = 1$. An example of magnitude response is shown in Fig. 2.18(b).

M-2.24

Write a function that computes the coefficients of a second-order resonant filter, given the normalized angular frequency ω_c (in radians) and the bandwidth B .



M-2.24 Solution

```
function [b,a]=reson2(omegac,B); %omegac and B are given in radians
r=(2-cos(B/2))-sqrt((2-cos(B/2))^2-1); % <= cos(B/2)=2-1/2*(r-1/r)
omega0=acos(2*r/(1+r^2)*cos(omegac));
a0=1; a1=-2*r*cos(omega0); a2=r^2;
b0=(1-r^2)*(sin(omega0));
a=[a0 a1 a2];
b=[b0 zeros(1,2)];
```

We have followed the Octave/Matlab convention in defining the coefficients b , a , but not the convention in defining normalized frequencies.

2.5.1.4 Subtractive synthesis of acoustic sounds

Subtractive synthesis has been mostly used for the generation of synthetic sounds which have no specific resemblance to acoustic instrumental sounds. However a few examples can be made.

We have already seen an example of subtractive synthesis when discussing spectral models: the stochastic component has been modeled as white noise passed through a time-varying filter. In this way noisy signals, like aeroacoustic noise produced by turbulent flow, can be effectively simulated.

Some of the source waveforms that we have examined previously in this section are already qualitatively similar to some instrumental sounds. As an example, the sawtooth wave has some similarity with the steady-state waveform produced by a bowed string, due to string-bow interaction mechanism. Consequently a simple subtractive synthesis scheme based on low-pass filtering of a sawtooth wave, and with the addition of suitable amplitude envelopes, can produce violin-like or cello-like sounds. Analogously, the spectrum of a clarinet sound is qualitatively similar to that of a square wave, since a cylindrical acoustic bore closed at one end and open at the other one supports all the odd harmonics of the fundamental frequency. Consequently the square wave is a good starting point for the subtractive synthesis of clarinet sounds.

One further example is modal synthesis of percussive sounds. A set of N second-order resonant filters R_i ($i = 1 \dots N$) of the form (2.28) can be grouped into a filterbank, where the same excitation signal x is fed to all the R_i 's, as depicted in Fig. 2.19. This specific source-filter structure is well suited to simulate percussive sounds.

In this case the excitation signal has an impulsive characteristics and represents a “striker” that a hammer or a mallet impart to a resonating object. Suitable “striker” excitation signals are e.g. a square impulse or a noise burst. The filter block represents the resonating object hit by the hammer: the center frequencies f_{ci} ($i = 1 \dots N$) of the resonant filters can be chosen to match a desired spectrum. As an example, a string will have a harmonic spectrum in which partials are regularly spaced on the frequency axis, therefore $f_{ci} = if_{c1}$ ($i = 2 \dots N$), where f_{c1} acts as the fundamental frequency. On the other hand, the partial distribution of a bar, or a bell, or the circular membrane of a drum, will be inharmonic. As an example, it is known that the partials of an ideal bar clamped at one end are approximately $f_{c2} \sim (2.998/1.994)^2 f_{c1}$, $f_{ci} \sim [(2i+1)/1.994]^2 f_{c1}$ ($i = 3 \dots N$).

The bandwidths B_i of the R_i 's determine the decay characteristics of each partial. A first possible choice is setting the same B (i.e. the same parameter r) for every filter. An alternative choice, that better describes the behavior of such resonant objects as strings, bars, and so on, amounts to setting the same quality factor $Q = B_i/f_{ci}$ for all the filters.

The structure depicted in Fig. 2.19 is also an example of *modal synthesis*. We will return on modal synthesis in Chapter *Sound modeling: source based approaches*, and will provide more physically sound foundations



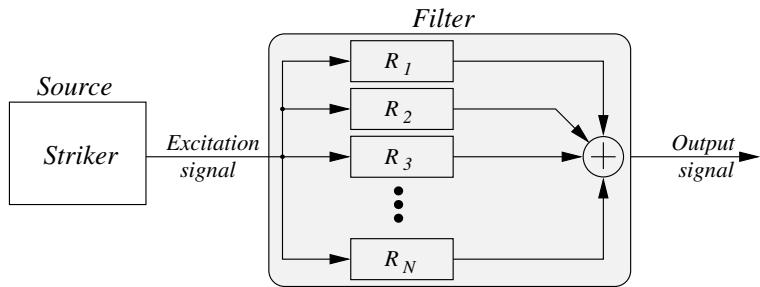


Figure 2.19: Parallel structure of digital resonators for the simulation of struck objects – the R_i 's have transfer functions of the form (2.28).

to this sound synthesis technique.

M-2.25

Write a function `modalosc(t0, tau, a, omegac, B)` that realizes the structure of Fig. 2.19. The parameter `t0` is the total duration of the sound signal, `tau`, `a` define duration and max. amplitude of the striker signal (e.g. a noise burst), and the vectors `omegac`, `B` define center frequencies and bandwidths of a bank of second-order oscillators.

2.5.2 Voice modeling

2.5.2.1 Voice production mechanism and models

Voice is an acoustic pressure wave created when air is expelled from the lungs through the trachea and vocal tract (see Fig. 2.20). The vocal tract starts at vocal fold opening (the glottis), and includes throat, nose, mouth and lips. As the acoustic wave passes through the vocal tract, its spectrum is altered by the resonances of the vocal tract (the *formants*).

Two basic types of sounds characterize speech, namely *voiced* and *unvoiced* sounds. Voiced sounds (vowels or nasals) result from a quasi-periodic excitation of the vocal tract caused by oscillation of the vocal folds in a quasi-periodic fashion. On the other hand, unvoiced sounds do not involve vocal fold oscillations and are typically associated to turbulent flow generated when air passes through narrow restrictions of the vocal tract (e.g. fricatives). In light of this description it is reasonable to describe voiced and unvoiced sounds as the effect of an acoustic filter (the vocal tract) applied to a source signal (the acoustic flow).

During voiced signals, the vocal folds oscillate in a very non-sinusoidal fashion. The quasi-periodic nature of the oscillations gives rise to an harmonic signal, and the frequency associated with the first harmonic partial is commonly termed the pitch of the voiced signal. The range of potential pitch frequencies can vary approximately from 50 Hz to 250 Hz for adult males, and from 120 Hz to 500 Hz for adult females. The frequency varies from speaker to speaker as well: every speaker has a “preferred pitch”, which is used naturally on the average. Additionally pitch is shifted up and down in speaking in response to factors relating to prosody and intonation (the pitch contour over time signals grammatical structure), but also stress and emotion.

Synthesized voice can be produced by several different approaches, all of which have some benefits and deficiencies. The methods are usually classified into three groups: *concatenative* synthesis, *formant* synthesis, and *articulatory* synthesis.

Concatenative synthesis uses pre-recorded samples of basic phonetic units, derived from natural speech. This technique is resemblant of the methods discussed in Sec. 2.2. Connecting pre-recorded



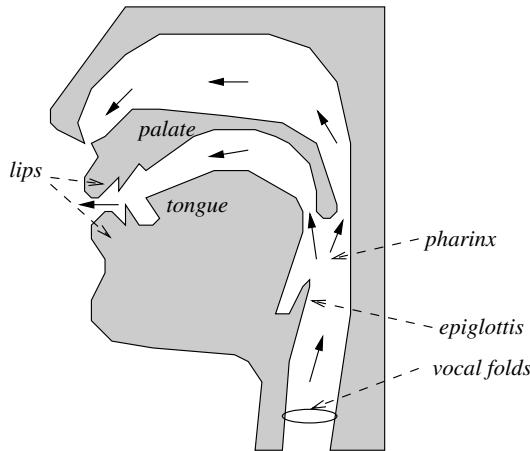


Figure 2.20: A schematic view of the phonatory system. Solid arrows indicates the direction of the airflow generated by lung pressure.

natural utterances is probably the easiest way and the most popular approach to produce intelligible and natural sounding synthetic speech. However, concatenative synthesizers are usually limited to one speaker and one voice and usually require more memory capacity than other methods.

Formant synthesis is based on the source-filter modeling approach described in Sec. 2.5 above: the transfer function of the vocal tract is typically represented as a series of resonant filters, each accounting for one formant. This was the most widely used synthesis method before the development of concatenative methods. Being based on a parametric model rather than on pre-recorded sounds, formant synthesis techniques are in principle more flexible than concatenative methods. We discuss formant synthesis in the next section.

Articulatory synthesis attempts to model the human voice production system directly and therefore belongs to the class of models discussed in Chapter *Sound modeling: source based approaches*. Articulatory synthesis typically involves models of the vocal folds, the vocal tract, and an associated set of *articulators* that define the area function between glottis and mouth. Articulators can be lip aperture, lip protrusion, tongue tip height and position, etc. Parameters associated to vocal folds can be glottal aperture, fold tension, lung pressure, etc. Although these methods promise high quality synthesis, computational costs are high and parametric control is arduous. At the time of writing no existing articulatory synthesizer can compare with a concatenative synthesizer.

2.5.2.2 Formant synthesis

Formant synthesis of voice realizes a source-filter model in which a broadband source signal undergoes multiple filtering transformations that are associated to the action of different elements of the phonatory system. Depending on whether voiced or unvoiced speech (see above) has to be simulated, two different models are used.

If $s[n]$ is a voiced signal, it can be expressed in the z -domain through the following cascaded spectral factors:

$$S(z) = g_v X(z) \cdot [G(z) \cdot V(z) \cdot R(z)], \quad (2.35)$$

where the source signal $X(z)$ is a periodic pulse train whose period coincides with the pitch of the signal, g_v is a constant gain term, $G(z)$ is a filter associated to the response of the glottis (the vocal folds) to pitch pulses, $V(z)$ is the vocal tract filter, and $R(z)$ simulates the radiation effect of the lips.

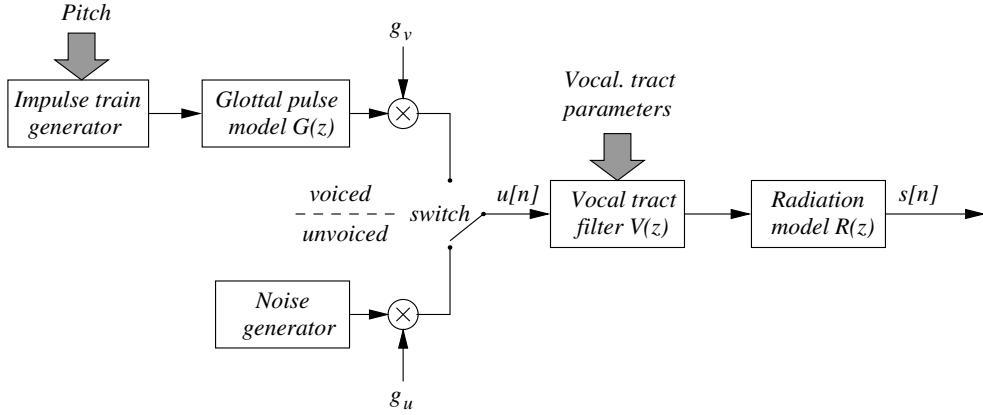


Figure 2.21: A general model for formant synthesis of speech.

If $s[n]$ is an unvoiced signal, vocal folds do not vibrate and turbulences is produced by the passage of air through a narrow constriction (such as the teeth). The turbulence can be modeled as white noise. In this case, the model is expressed in the Z-domain as

$$S(z) = g_u X(z) \cdot [V(z) \cdot R(z)], \quad (2.36)$$

where the source signal $X(z)$ is in this case a white noise sequence, while the gain term g_u is in general different from the voiced configuration gain, g_v . Note that the vocal fold response $G(z)$ is not included in the model in this case.

Any voiced or unvoiced sound is modeled by either Eq. (2.35) or (2.36). The complete transfer function $H(z) = S(z)/X(z)$ may or may not include vocal fold response $G(z)$ depending on whether the sound is voiced or unvoiced. The block structure of the resulting model is shown in Fig. 2.21.

The filter $G(z)$ shapes the glottal pulses. More specifically, since the input $x[n]$ is a pulse train, the output from this block is the impulse response $g[n]$ of this filter. We propose two historically relevant models. The first one is a FIR model with impulse response:

$$g_{\text{FIR}}[n] = \begin{cases} \frac{1}{2} \left[1 - \cos \left(\frac{\pi n}{N_1} \right) \right], & 0 \leq n \leq N_1, \\ \cos \left(\frac{\pi(n-N_1)}{2N_2} \right), & N_1 \leq n \leq N_1 + N_2, \\ 0 & \text{elsewhere.} \end{cases} \quad (2.37)$$

The second one is an IIR model with transfer function

$$G_{\text{IIR}}(z) = \frac{1}{[1 - \exp(-c/F_s)z^{-1}]^2}. \quad (2.38)$$

It represents a low-pass filter with one real pole of order 2, which integrates the pulse train.

The radiation filter $R(z)$ is a load that converts the airflow signal at the lips into an outgoing pressure wave (the signal $s[n]$). Under very idealized hypothesis, $R(z)$ can be approximated at least for low frequencies by a fixed differentiator:

$$R(z) = 1 - \rho z^{-1}, \quad (2.39)$$

where ρ is a lip radiation coefficient whose value is very close to 1.

The vocal tract filter $V(z)$ models vocal tract formants. A single formant can be modeled with a two-pole resonator (see Eq. (2.28)) which enables both the formant frequency and its bandwidth to be



specified. We denote the filter associated to the i th formant as $V_i(z)$, having center frequency f_i and bandwidth B_i . At least three vocal tract formants are generally required to produce intelligible speech and up to five formants are needed to produce high quality speech.

Two basic structures, parallel and cascade, can be used in general, but for better performance some kind of combination of these is usually adopted. A cascade formant synthesizer consists of band-pass resonators connected in series, i.e. the output of each formant resonator is applied to the input of the following one. A parallel formant synthesizer consists of resonators connected in parallel, i.e. the same input is applied to each formant filter and the outputs are summed. The corresponding vocal tract models are

$$V_{\text{casc}}(z) = g \prod_{i=1}^K V_i(z), \quad V_{\text{par}}(z) = \sum_{i=1}^K a_i \cdot V_i(z). \quad (2.40)$$

The cascade structure needs only formant frequencies as control information. The main advantage of this structure is that the relative formant amplitudes for vowels do not need individual controls. A cascade model of the vocal tract is considered to provide good quality in the synthesis of vowels, but is less flexible than a parallel structure, which enables controlling of bandwidth and gain for each formant individually.

M-2.26

Using the functions `waveosc` and `reson2`, realize a parallel formant synthesizer. Use 3 second-order IIR cells, corresponding to the first 3 vowel formants.

M-2.26 Solution

```
function s= formant_synth(a,f,vowel);

global Fs; global SpF;

%the vowel can be 'a' or 'e' or 'i'
if (vowel=='a')           fc=[700 1100 2500]; B=[50 75 100];
elseif (vowel=='e')        fc=[500 1850 2500]; B=[50 75 100];
elseif (vowel=='i')        fc=[300 2400 2500]; B=[50 75 100];
else error('Wrong vowel!'); end

%%% construct formant filters (numerators and denominators)
num=zeros(length(fc),3); den=zeros(length(fc),3);
for i=1:length(num)
    [num(i,:), den(i,:)] = reson2(2*pi/Fs*fc(i), 2*pi/Fs*B(i));
end

%%% compute sound %%
x=waveosc(0,a,f,0,'imp',30); %impulse train source
s=zeros(1,length(x));
for i=1:size(num,2) s= s + filter(num(i,:),den(i,:),x); end
```

Figure 2.22 shows an example of formant synthesis using a parallel formant filtering structure. In particular Fig. 2.22(c) shows that when the same vowel is uttered with two different pitches, only the fine spectral structure is affected, while the overall spectral envelope does not change its shape.

2.5.3 Linear prediction

The problem of extracting a spectral envelope from a signal spectrum is generally an ill-posed problem. If the signal contains harmonic partials only, one could state that the spectral envelope is the curve that



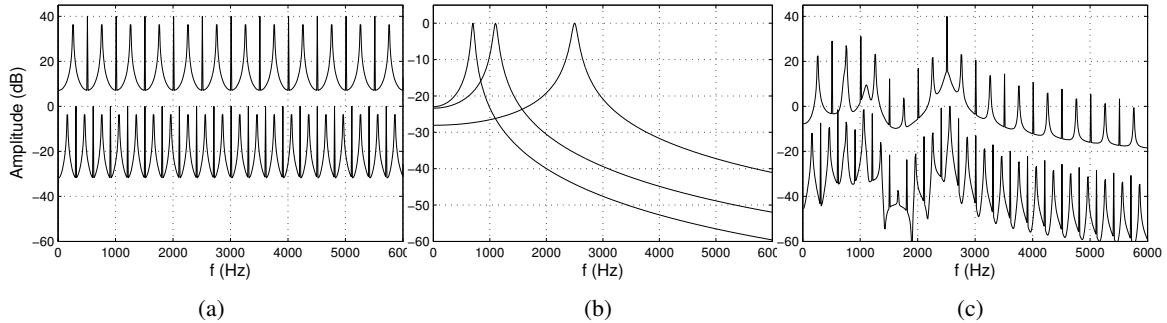


Figure 2.22: Formant synthesis of voice: (a) spectra of two pulse trains with fundamental frequencies at 150 Hz and 250 Hz; (b) first three formants of the vowel /a/; (c) spectra of the two output signals obtained by filtering the pulse trains through a parallel combination of the three formants.

passes through the partial peaks. This implies that 1) the peak values have to be retrieved, and 2) an interpolation scheme should be (arbitrarily) chosen for the completion of the curve in between the peaks. If the sound contains inharmonic partials or a noisy part, then the notion of a spectral envelope becomes completely dependent on the definition of what belongs to the “source” and what belongs to the “filter”.

Three main techniques, with many variants, can be used for the estimation of the spectral envelope. The *channel vocoder* uses frequency bands and performs estimations of the amplitude of the signal inside these bands and thus the spectral envelope. *Linear prediction* estimates an all-pole filter that matches the spectral content of a sound. When the order of this filter is low, only the formants are taken, hence the spectral envelope. *Cepstrum* techniques perform smoothing of the logarithm of the FFT spectrum (in decibels) in order to separate this curve into its slow varying part (the spectral envelope) and its quickly varying part (the source signal). In this section we present the basics of Linear Prediction (LP) techniques. We will return on cepstral analysis in Chapter *Auditory based processing*.

2.5.3.1 Linear prediction equations

Consider a general linear system that describes a source-filter model:

$$S(z) = gH(z)X(z), \quad \text{with } H(z) = \frac{1 + \sum_{k=1}^q b_k z^{-k}}{1 - \sum_{k=1}^p a_k z^{-k}}, \quad (2.41)$$

where g is a gain scaling factor and $X(z)$ and $S(z)$ are the Z-transforms of the source signal $x[n]$ and the output signal $s[n]$, respectively. This is often termed an *ARMA*(p, q) (Auto-Regressive Moving Average) model, in which the output is expressed as a linear combination of p past samples and $q + 1$ input values. LP analysis works on an approximation of this system, namely on an all-pole model:

$$S(z) = gH_{LP}(z)X(z), \quad \text{with } H_{LP}(z) = \frac{1}{1 - \sum_{k=1}^p a_k z^{-k}}. \quad (2.42)$$

The time-domain version of this equation reads $s[n] = gx[n] + \sum_{k=1}^p a_k s(n-k)$. Therefore the output $s[n]$ can be predicted using a linear combination of its p past values, plus a weighted input term. In statistical terminology, the output *regresses* on itself, therefore system (2.42) is often termed an *AR*(p) (Auto-Regressive) model

One justification of this approximation is that the input signal $x[n]$ is generally unknown together with the filter $H(z)$. A second more substantial reason is that any filter $H(z)$ of the form (2.41) can be



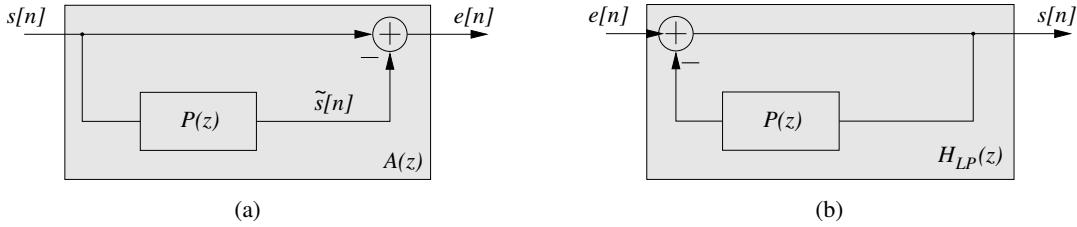


Figure 2.23: LP analysis: (a) the inverse filter $A(z)$, and (b) the prediction error $e[n]$ interpreted as the unknown input $gx[n]$.

written as $H(z) = h_0 H_{min}(z) H_{ap}(z)$, where h_0 is a constant gain, $H_{min}(z)$ is a minimum-phase filter, and H_{ap} is an all-pass filter (i.e. $|H_{ap}(e^{j\omega_d})| = 1, \forall \omega_d$). Moreover, the minimum-phase filter $H_{min}(z)$ can be expressed as an all-pole system of the form (2.42). Therefore we can say that LP analysis ideally represents the all-pole minimum-phase portion of the general system (2.41), and therefore yields at least a correct estimate of the magnitude spectrum.

Given an output signal $s[n]$, Linear Prediction analysis provides a method for determining the “best” estimate $\{\tilde{a}_i\}$ ($i = 1, \dots, p$) for the coefficients $\{a_i\}$ of the filter (2.42). The method can be interpreted and derived in many ways, here we propose the most straightforward one. Given an estimate $\{\tilde{a}_i\}$ of the filter coefficients, we define the *linear prediction* $\tilde{s}[n]$ of the output $s[n]$ as $\tilde{s}[n] = \sum_{k=1}^p \tilde{a}_k s(n-k)$. In the Z-domain we can write

$$\tilde{S}(z) = P(z)S(z), \quad \text{with } P(z) = \sum_{k=1}^p a_k z^{-k}, \quad (2.43)$$

and we call the FIR filter $P(z)$ a *prediction filter*. We then define the *prediction error* or *residual* $e[n]$ as the difference between the output $s[n]$ and the linear prediction $\tilde{s}[n]$. In the z -domain, the prediction error $e[n]$ is expressed as

$$E(z) = A(z)S(z), \quad \text{with } A(z) = 1 - \sum_{k=1}^p a_k z^{-k}. \quad (2.44)$$

Comparison of Eqs. (2.42) and (2.44) shows that, if the speech signal obeys the model (2.42) exactly, and if $\tilde{a}_k = a_k$, then the residual $e[n]$ coincides with the unknown input $x[n]$ times the gain factor g , and $A(z)$ is the inverse filter of $H_{LP}(z)$. Therefore LP analysis provides an estimate of the inverse system of (2.42):

$$e[n] = gx[n], \quad A(z) = [H_{LP}(z)]^{-1}. \quad (2.45)$$

This interpretation is illustrated in Fig. 2.23. If we assume that the prediction error has white (flat) spectrum, then the all-pole filter $H_{LP}(z)$ completely characterizes the spectrum of $s[n]$. For this reason $A(z)$ is also called a *whitening filter*, because it produces a residual with a flat power spectrum. Two kinds of residuals, both having a flat spectrum, can be identified: the pulse train and the white noise. If LP is applied to speech signals, the pulse train represent the idealized vocal-fold excitation for voiced speech, while white noise represents the idealized excitation for unvoiced speech.

The roots of $A(z)$ (i.e., the poles of $H_{LP}(z)$) are representative of the formant frequencies. In other words, the phases of these poles, expressed in terms of analog frequencies, can be used as an estimate of the formant frequencies, while the magnitude of the poles relate to formant bandwidths according to the equations written in Sec. 2.5.1 when discussing resonant filters.



We now describe the heart of LP analysis and derive the equations that determine the “best” estimate $\{\tilde{a}_i\}$ ($k = 1, \dots, p$). In this context “best” means best in a least-square sense: we seek the $\{\tilde{a}_i\}$ s that minimize the energy $E\{e\} = \sum_{m=-\infty}^{+\infty} e^2[m]$ of the residual, i.e. we set to zero the partial derivatives of $E\{e\}$ with respect to the a_i s:

$$0 = \frac{\partial E\{e\}}{\partial a_i} = 2 \sum_{m=-\infty}^{+\infty} e[m] \frac{\partial e[m]}{\partial a_i} = -2 \sum_{m=-\infty}^{+\infty} \left\{ \left[s[m] - \sum_{k=1}^p a_k s(m-k) \right] s[m-i] \right\}, \quad (2.46)$$

for $i = 1 \dots p$. If one defines the temporal autocorrelation of the signal $s[n]$ as the function $r_s[i] = \sum_{m=-\infty}^{+\infty} s[m]s[m-i]$, then the above equation can be written as

$$\sum_{k=1}^p a_k r_s[i-k] = r_s[i], \quad \text{for } i = 1 \dots p. \quad (2.47)$$

The system (2.47) is often referred to as the *normal equations*. Solving this system in the p unknowns a_i yields the desired estimates \tilde{a}_i .

2.5.3.2 Short-time LP analysis

In many applications of interest, and in particular analysis and resynthesis of speech signals, the coefficients a_k are not constant but slowly time-varying. Therefore the description (2.42) is only valid in a short-time sense, i.e. the a_k s can be assumed constant during an analysis frame. Therefore short-time analysis has to be used, in which the coefficients and the residual are determined from windowed sections, or frames, of the signal.

There are various efficient methods to compute the filter coefficients, the most common ones being the autocorrelation method, the covariance method, and the Burg algorithm. In this section we briefly describe the autocorrelation method, that simply amounts to substitute the autocorrelation function r_s of Eq. (2.47) with its short-time approximation:

$$r_s[i] \sim \sum_{m=1}^N u[m]u[m-i], \quad \text{where } u[m] = s[m]w[m] \quad (2.48)$$

is a windowed version of $s[m]$ in the considered frame ($w[m]$ is typically a Hamming window), and N is the length of the frame. Then the system (2.47) is solved within each frame. An efficient solution is provided by the so-called *Levinson-Durbin recursion*, an algorithm for solving the problem $Ax = b$, with A Toepliz, symmetric, and positive definite, and b arbitrary. System (2.47) is an instance of this general problem.

M-2.27

Write a function `lp_coeffs` that computes the LP coefficients of a finite-length signal $s[n]$, given the desired prediction order p .

M-2.27 Solution

```
% Compute LP coeffs using the autocorrelation method
% s is the (finite-length) signal, p is the prediction order
% a are the computed LP coefficients, g is the gain (sqrt of residual variance)

function [a,g] = lp_coeffs(s,p)
```



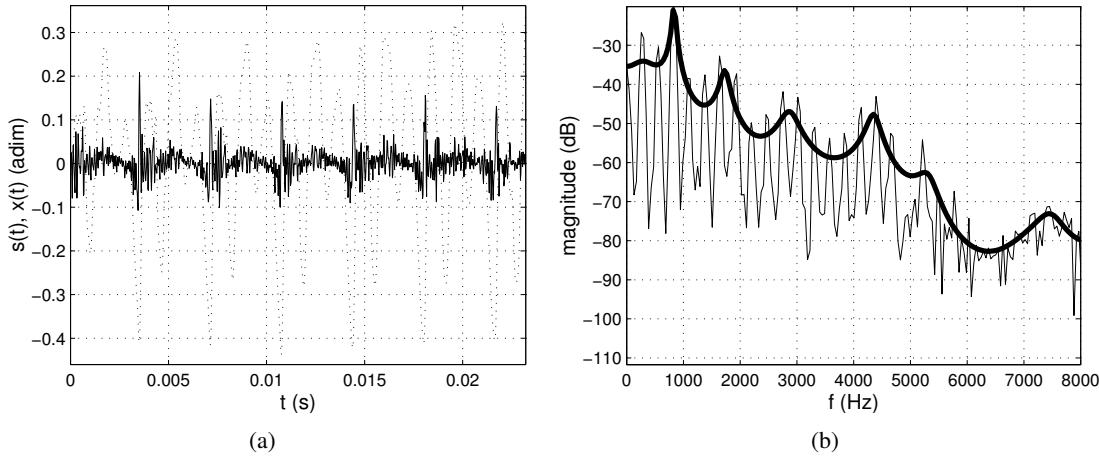


Figure 2.24: Example of LP analysis/synthesis, with prediction order $p = 50$; (a) target signal $s[n]$ (dotted line) and unit variance residual $x[n]$ (solid line); (b) magnitude spectra $|S(f)|$ (thin line) and $|gH_{LP}(f)|$ (thick line).

```
R=xcorr (s,p) ; % autocorrelation sequence R(k) with k=-p,...,p
R(1:p)=[]; % delete the first p samples
if norm(R) ~= 0
    [a,v] = levinson(R,p); % Levinson-Durbin recursion
    % a=[1,-a1,-a2,...,-ap], v = variance of the residual
else
    a=[1, zeros(1,p)];
end
g=sqrt (sum(a' .* R)); % gain factor (= sqrt(v))
```

Note that we are using the native function `levinson`, that computes the filter coefficients (as well as the variance of the residual) given the autocorrelation sequence and the prediction order.

Figure 2.24 shows an example of LP analysis and resynthesis of a single frame of a speech signal. As shown in Fig. 2.24(a), the analyzed frame is a portion of voiced speech and $s[n]$ is pseudo-periodic. Correspondingly, the estimated source signal $x[n]$ is a pulse train. Figure 2.24(b) shows the magnitude responses of the target signal and the estimated transfer function $gH_{LP}(z)$. A typical feature of LP spectral modeling can also be observed from this figure: the LP spectrum matches the signal spectrum much more closely in the region of large signal energy (i.e. near the spectrum peaks) than near the regions of low energy (spectrum valley).

M-2.28

Write an example script that analyzes frame-by-frame a voice signal using the LP model (2.42).

M-2.28 Solution

```
[s, Fs] = wavread('la.wav'); %%%% input file

%% analysis parameters
N=2048; %block length
Sa=256; %analysis hop-size
p=round(Fs/1000)+4 ; %prediction order
```



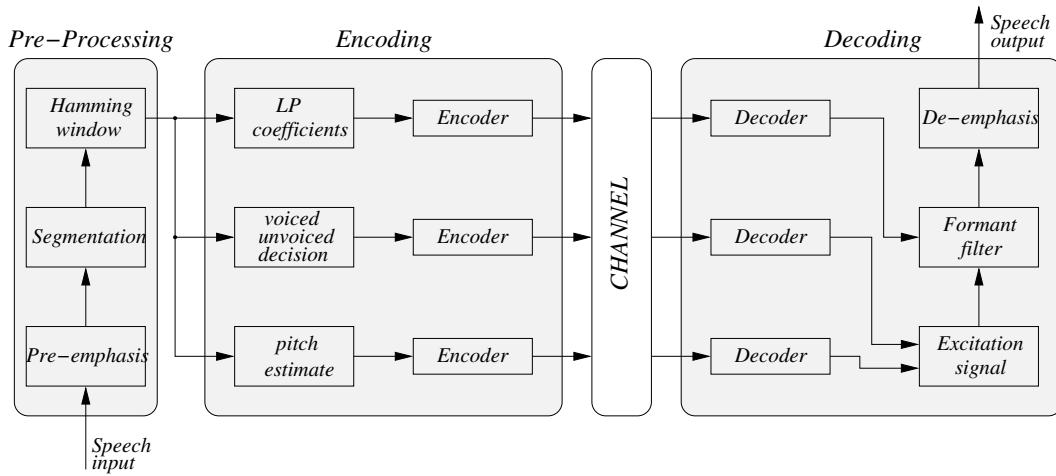


Figure 2.25: General scheme of a simple LP based codec.

```

win=hamming(N)'; %' window for input block

M = ceil(length(s)/Sa); %number of frames in the signal
s(M*Sa+N)=0; %now s is exactly M*Sa samples

for i=0:M-1 %% frame-by-frame analysis
    frame=s( (i*Sa+1):(i*Sa+N) );
    [a,g]=lp_coeffs(frame, p); % compute LP coeffs
    x=filter(a,g,frame); % X(z)=A(z)/g * S(z), i.e. x[n] = e[n]/g
    frameSpec= abs(fft(frame)); [H,F]=freqz(g,a,N,Fs);
    figure(1); clf; plot(frame); hold on; plot(x,'r'); grid on;
    figure(2); clf; plot(F,20*log10(frameSpec)); hold on; grid on;
    plot(F,20*log10(abs(H)),'r'); axis([0 5000 0 max(20*log10(frameSpec))]);
    pause;
end

```

Note that we have used the function `lp_coeffs` written in example M-2.27. The signals plotted in Fig. 2.24 have been computed from this script.

When formant parameters are extracted on a frame-by-frame basis, a lot of discontinuities and local estimation observation errors are found. Therefore, proper techniques have to be used in order to determine smooth formant trajectories over analysis frames. We have already encountered a conceptually similar problem in Sec. 2.4.2, when we have discussed a “sinusoid tracking” procedure.

M-2.29

Plot the formant frequencies as a function of the frame number, i.e., of time, in order to observe the time-evolution of the vocal tract filter. To this purpose, segment a speech signal $s[n]$ into M Hamming windowed frames $s_m[n]$, with a block length N and a hop-size $S_a = N/2$. Then, for each frame: a) compute the LP coefficients; b) find the filter poles and the corresponding formant frequencies; c) discard poles whose magnitude is less than 0.8, as these are unlikely to represent formants.

2.5.3.3 Linear Predictive Coding (LPC)

One of the most successful applications of LP analysis is in speech coding and synthesis, in particular for mobile phone communication. Figure 2.25 depicts a synthetic block diagram of a simple encoder-

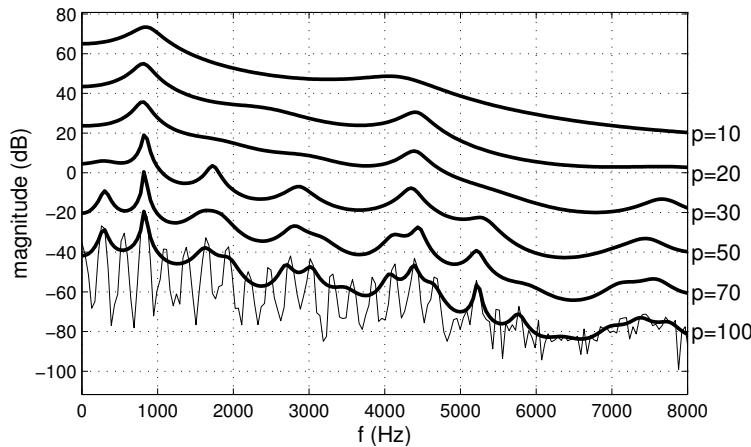


Figure 2.26: Example of LP spectra for increasing prediction orders p (the target signal is a frame of voiced speech). For the sake of clarity each spectrum is plotted with a different offset.

decoder system based on LP analysis. Speech is segmented in frames (typical frame lengths can range from 10 to 20 ms). In this phase a pre-emphasis processing can also be applied: since the lower formants contain more energy, they are preferentially modeled with respect to higher formants, and a pre-emphasis filter can compensate for this by boosting the higher frequencies (when reconstructing the signal the inverse filter should be used).

In its simplest formulation the encoder provides, for every frame, the coefficients a_k of the prediction filter, the gain g , a flag that indicates whether the frame corresponds to voiced or unvoiced speech, and the pitch (only in the case of voiced speech). The decoder uses this information to re-synthesize the speech signal. In the case of unvoiced speech, the excitation signal is simply white noise, while in the case of voiced speech the excitation signal is a pulse train whose period is determined by the encoded pitch information.

It is clear that most of the bits of the encoded signal are used for the a_k parameters. Therefore the degree of compression is strongly dependent on the order p of the LP analysis, which in turn has a strong influence on the degree of smoothness of the estimated spectral envelope, and consequently on the quality of the resynthesis (see Fig. 2.26). A commonly accepted operational rule for achieving reasonable intelligibility of the resynthesized speech is

$$p = \begin{cases} F_s + 4, & \text{for voiced speech,} \\ F_s, & \text{for unvoiced speech,} \end{cases} \quad (2.49)$$

where F_s is the sampling frequency in kHz, rounded to the nearest integer.

LPC-10 is an example of a standard that basically implements the scheme of Fig. 2.25. This standard uses an order $p = 10$ (hence the name), a sample rate $F_s = 8$ kHz (which is a common choice in telephone speech applications since most of the energy in a speech signal is in the range [300, 3400] Hz), and a frame length of about 22.5 ms. With these values, intelligible speech can be resynthesized.

However LPC-10, and in general similar early codecs, produced speech with very poor quality due to many artifacts: “buzzy” noise through parameter updates, jitter in the excitation signal, wide formant bandwidths, and so on. More recent and commonly used codecs are able to provide natural sounding speech at relatively low bit rates, thanks to an improved description of the excitation signal. Instead of applying a simple two-state voiced/unvoiced model, these codecs estimate the excitation signal through an analysis-by-synthesis approach: excitation waveforms are passed through the formant filter, and the



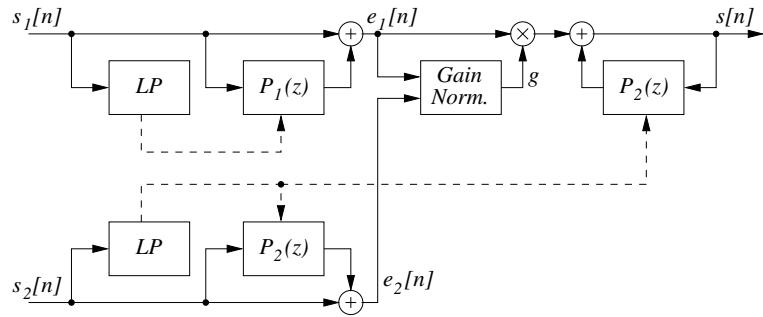


Figure 2.27: Block scheme of a LP-based implementation of cross-synthesis (also known as vocoder effect) between two input sounds s_1 and s_2 .

excitation which gives the minimum weighted error between the original and the reconstructed speech is then chosen by the encoder and used to drive the synthesis filter at the decoder. It is this ‘closed-loop’ determination of the excitation which allows these codecs to produce good quality speech at low bit rates, at the expense of a much higher complexity of the coding stage.

Within this family of analysis-by-synthesis codecs, many different techniques have been developed for the estimation of the excitation signal. Historically, the first one is the Multi-Pulse Excited (MPE) codec. Later the Regular-Pulse Excited (RPE), and the Code-Excited Linear Predictive (CELP) codecs were introduced. The “Global System for Mobile communications” (GSM), a digital mobile radio system which is extensively used throughout Europe, and also in many other parts of the world, makes use of a RPE codec.

2.5.3.4 LP based audio effects

Several musical effects can be realized based on spectral envelope estimation techniques, particularly linear prediction. These effects exploit the source-filter decomposition obtained through LP analysis of an input sound: modifications are applied to either the excitation, or the synthesis filter, or both.

Thinking of vocal signals, one simple effect can be obtained by modifying the filter block and shifting the vocal tract formants towards higher frequencies, without modifying the source signal. This results in a so-called “donald duck” effect, in which the vocal tract has been “shortened” without altering the pitch of the original voice.⁸

Time-stretching effects can be obtained by time-stretching the excitation signal and updating the synthesis filter at a correspondingly time-stretched rate, thus preserving the formant structure. Similarly, pitch-shifting can be obtained by modifying the pitch of the excitation signal: again, this approach allows to transpose pitch without affecting the formant structure of a signal.

A well-known effect is *cross-synthesis* between two sound signals $s_1[n]$ and $s_2[n]$: linear prediction (or any other deconvolution technique) is applied to both signals and then the excitation signal of $s_1[n]$ is used in combination with the LP filter of $s_2[n]$. Figure 2.27 shows the block scheme of a possible implementation of a musical vocoder. The cross-synthesis gives particularly pleasant results if $s_2[n]$ is a speech signal and $s_1[n]$ is a musical sound signal: this produces the popular “vocoder” musical effect, in which an instrumental sound becomes a “talking instrument” when filtered with a time-varying LP filter of a speaking voice. For musically interesting results, the two sounds should be synchronized.

⁸A similar result may be obtained by emitting voice after inhaling a light gas such as helium: the helium in the vocal tract changes its resonances, without affecting the oscillations of the vocal fold. Note that, because of the risk involved, it is safe not to try this experiment at home.

As an example, for the case of cross-synthesis between speech and music the played instrumental notes should fit to the rhythm of the syllables of the speech: this may be achieved if either speech or music is coming from a prerecorded source and the other sound is produced to match to the recording, or if a performer is both playing the instrument and speaking, thus producing both signals at the same time.

M-2.30

Realize the cross-synthesis effect depicted in Fig. 2.27.

2.6 Non-linear models

All the synthesis and processing models examined so far in this chapter are linear and time-invariant. A LTI system, which is completely characterized by its impulse response (or by its transfer function), cannot introduce spectral energy where it is not already present: a sinusoidal signal processed through a linear system remains a sinusoidal signal. When entering the domain of non-linear transformations, this picture changes radically: the spectrum of a signal processed through a non-linear system can be drastically modified, and spectral energy can be created even where it was not originally present. As a consequence, non-linear transformations can modify substantially the nature of the input sounds.

Spectral modifications introduced by non-linear transformations can be grouped into two main effects: spectrum enrichment and spectrum shift. The first effect is due to non-linear distortion of the signal, and reproduces to some extent the non-linearities and saturations found on real systems (e.g., analog amplifiers and electronic valves). The second effect is due to multiplication of the input signal by a sinusoidal carrier signal, which moves the spectrum to the vicinity of the carrier frequency. It derives from abstract mathematical properties of trigonometric functions as used in modulation theory applied to music signal. Therefore, it partially inherits – and simulates digitally – the processing blocks used in analog electronic music. Transformations that produce spectral shifts can produce very intriguing musical effects: complex harmonic and inharmonic spectra can be created starting from simple (sinusoidal) input sounds, and various harmonic relations among the partials can be established.

2.6.1 Memoryless non-linear processing

In general the output value of a non-linear system depends on present and past values of the input and the output, i.e. the system has *memory* (similarly to a generic LTI). However in many interesting audio signal processing applications *memoryless* non-linearities can be used, i.e. non-linear systems whose output depends only on the current input value and not on past values. In this cases the system can be described by a non-linear function $F : \mathbb{R} \rightarrow \mathbb{R}$, called *distortion function*, that maps the input into the output:

$$y[n] = F(x[n]). \quad (2.50)$$

By means of such a distortion function, a sound that with a rich spectral content can be obtained by processing a simple sinusoidal input. A block diagram of such a *non-linear distortion synthesis* scheme (or *waveshaping* scheme) is provided in Fig. 2.28, for the particular case of a sinusoidal input signal $x[n]$.

2.6.1.1 Waveshaping and harmonic distortion

In Chapter *Fundamentals of digital audio processing* we have seen that a sinusoidal input $x[n] = a \cos(\omega_0 n)$ which passes through a LTI system (a filter) produces an output signal $y[n]$ which is still a sinusoid with the same frequency ω_0 and amplitude and phase modified according to the transfer function (see



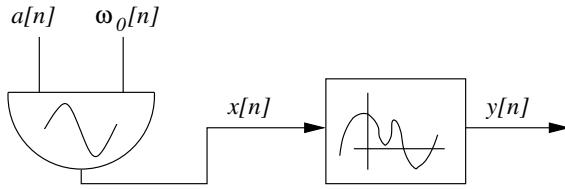


Figure 2.28: Sound synthesis by non-linear distortion (or waveshaping).

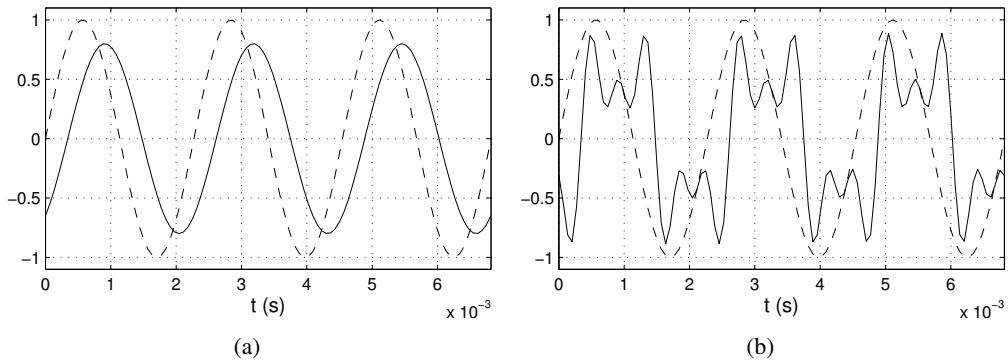


Figure 2.29: Example of output signals from a linear and from a non-linear system, in response to a sinusoidal input; (a) in a linear system the input and output differ in amplitude and phase only; (b) in a non-linear system they have different spectra.

Fig. 2.29(a)). On the other hand, if the signal is processed through a non-linear system of the form (2.50), more substantial modifications of the spectrum occur: the output has in general the form

$$y[n] = \sum_{k=0}^N a_k \cos(k\omega_0 n), \quad (2.51)$$

and therefore the spectrum of y possesses energy at higher harmonics of ω_0 (see Fig. 2.29(b)). This effect is termed *harmonic distortion*, and can be quantified through the *total harmonic distortion (THD)* parameter:

$$THD = \sqrt{\frac{\sum_{k=2}^N a_k^2}{\sum_{k=1}^N a_k^2}}. \quad (2.52)$$

In many cases one wants to minimize the THD in non-linear processing, but in other cases distortion is exactly what we want in order to enrich an input sound. An example is the effect of valves, as in amplifiers for electric guitars. There is no way to interpret harmonic distortion in terms of some transfer function, because the concept of transfer function itself cannot be defined for a non-linear system (equivalently, the impulse response of a non-linear system does not tell anything about its response to a generic input).

For a memoryless non-linear system, the THD has a straightforward interpretation if one rewrites the distortion function in terms of its (truncated) Taylor expansion around the origin:

$$y[n] = F(x[n]) = \sum_{i=0}^N a_i x^i[n]. \quad (2.53)$$



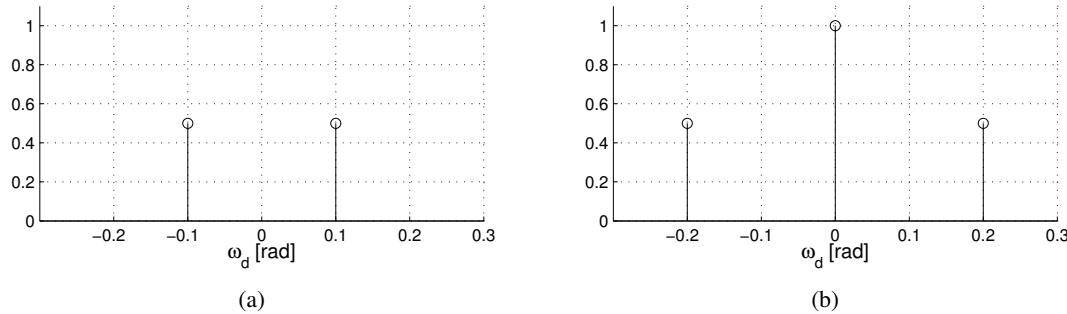


Figure 2.30: Example of quadratic distortion; (a) spectrum of a sinusoid $x[n]$ and (b) spectrum of the squared sinusoid $x^2[n]$.

Consider the effect of the quadratic term in this summation. The spectrum of $x^2[n]$ is the convolution of the spectrum of $x[n]$ with itself. Therefore, when $x[n] = a \cos(\omega_0 n)$, i.e. x is a sinusoidal signal with frequency ω_0 , the spectrum of $x^2[n]$ is $[X * X](\omega_d)$, and thus contains the frequency $2\omega_0$ as well as the 0 frequency. The same result may be derived looking at the time-domain signal: straightforward trigonometry shows that the squaring operation on a sinusoidal input signal produces the output signal

$$y[n] = x^2[n] = \frac{a^2}{2} [1 + \cos(2\omega_0 n)]. \quad (2.54)$$

Again, one can see that the output signal contains a DC component and the frequency $2\omega_0$. For a generic input $x[n]$, the squaring operation will in general double the bandwidth of the spectrum. In particular, for an input signal $x[n] = \sum_k a_k \cos(\omega_k n)$ the squaring operation produces an output signal that contains all the frequencies $2\omega_k$ and moreover all the frequencies $\omega_{k_1} \pm \omega_{k_2}$ (the so-called intermodulation frequencies, which arise from the cross terms in the square of the sum).

Similar considerations apply to higher-order terms of the Taylor expansion: raising a sinusoidal signal with frequency ω_0 to the i -th power will produce a spectrum that contains every other frequency up to $i\omega_0$. In particular, if the Taylor expansion of F contains only odd (or only even) powers, then the resulting spectrum will contain only odd (or only even) partials.

One specific application of waveshaping is in the generation of pure harmonics of a sinusoid. As an example suppose that, given the input $x[n] = \cos(\omega_0 n)$ one wants to generate the output $y[n] = \cos(5\omega_0 n)$, i.e. the fifth harmonic of the input. It is easily verified that waveshaping the input through the polynomial distortion function $F(x) = 16x^5 - 20x^3 + 5x$ provides the desired result. More in general, the polynomial that transforms the sinusoid $\cos(\omega_0 n)$ into the sinusoid $\cos(i\omega_0 n)$ is the i -th order *Chebyshev polynomial*.⁹ By combining Chebyshev polynomials, one can then produce any desired superposition of harmonic components in the output signal.

2.6.1.2 Aliasing and oversampling

Implementation of a memoryless non-linear system is apparently straightforward in the discrete-time domain: given a distortion function $F(x)$, either in analytical form or estimated from measurements, this can be pre-computed and stored in a look-up table. In order to process an incoming input sample $x[n]$, all that is needed is looking up the corresponding value $F(x[n])$ in the table, and possibly interpolating between adjacent points of the table in order to increase accuracy.

⁹Chebyshev polynomials are in general a sequence of orthogonal polynomials which can be defined recursively, and are widely used especially in approximation theory and polynomial interpolation.



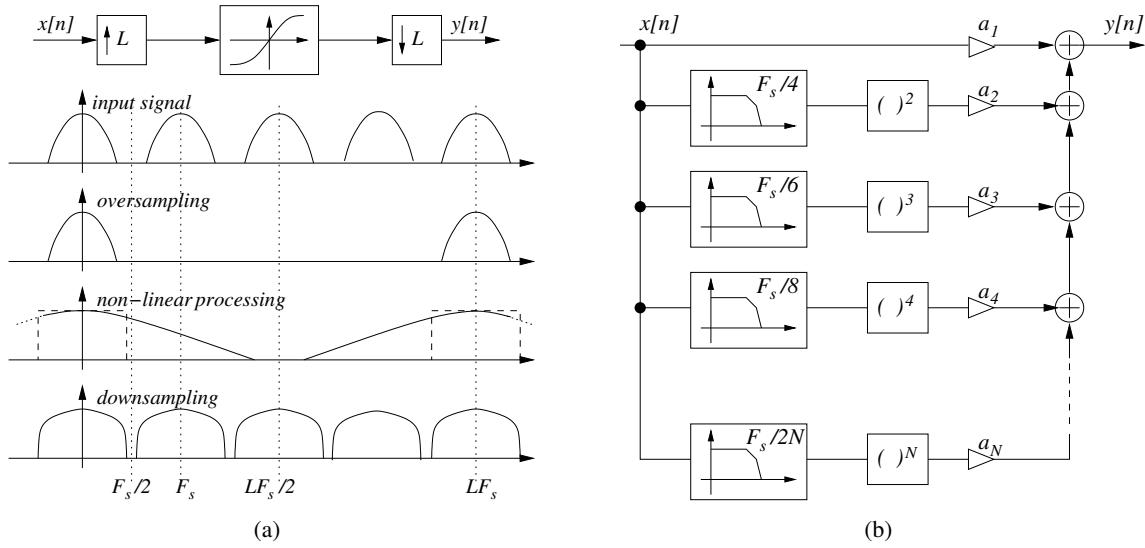


Figure 2.31: Two implementations of a memoryless non-linear system; (a) non-linear processing inserted between oversampling and downsampling; (b) non-linear processing on band-limited versions of the input.

From the discussion above we know that if $F(x)$ is a polynomial, or if its Taylor series expansion can be truncated, the bandwidth of the output spectrum induced by harmonic distortion remains limited. Nonetheless it can easily extend beyond f_{Ny} , and consequently cause aliasing in the output signal. Even though in some musical effects such an additional aliasing distortion can be tolerated,¹⁰ in general it has to be avoided as much as possible.

Solution: oversampling (see Fig. 2.31(a)). The procedure is illustrated in Fig. 2.31(a). The input is oversampled and interpolated to a higher sampling freqency, say LF_s with some $L > 1$. The distortion function is applied to this oversampled signal. The resulting output can have spectral energy up to the new Nyquist frequency $LF_s/2$. Finally the signal has to be converted back to the original sampling frequency: in order to avoid aliasing at this stage the signal is low-pass filtered back to the original Nyquist frequency.

If F is a polynomial or can be reasonably approximated by a polynomial through its truncated Taylor expansion, an alternative procedure can be designed, that avoids oversampling. This is illustrated in Fig. 2.31(b). The input signal is split into several low-pass versions, and each of them is processed through one term of the polynomial. In this stage no aliasing is generated by construction. Finally the output signal is constructed as the sum of the processed low-pass versions. This procedure is equivalent to the preceding one.

2.6.1.3 Clipping, overdrive and distortion effects

Clipping is a very common type of non-linear distortion. An ideal symmetric clipping distortion function is constructed as follows: it is the identity function, i.e. $F(x) = x$, as long as $|x| \leq x_{\max}$, and it is a constant function $F(x) = x_{\max}$ when $|x| > x_{\max}$. Therefore the input amplitude affects the output waveform in that the clipping function passes the input unchanged as long as its amplitude is small enough, while the output remains constant when the input amplitude grows beyond the limit. In

¹⁰It may be even considered to be helpful, e.g. for extreme metal guitar distortions.

particular, when the input has a decaying amplitude envelope (as in note played on a guitar) the output evolves from a nearly square waveform at the beginning to an almost pure sinusoid at the end.

This kind of effect will be well known to almost any guitarist or anyone who has played an instrument through an overdriven amplifier. In musical terms, *overdrive* refers to a nearly linear audio effect device which can be driven into the non-linear region of its distortion curve only by high input levels. The transition from the operating linear region to the non-linear region is smooth. *Distortion* instead refers to a similar effect, with the difference that the device operates mainly in the non-linear region of the distortion curve.

The sound of a valve amplifier is based on a combination of various factors: the main processing features of valves themselves are important, but the amplifier circuit as well as the chassis and loudspeaker combination have their influence on the final sound. Foot-operated pedal effects have simpler circuitry but always include a non-linear stage that introduces harmonic distortion on the input signal, in a faster way and at lower sound levels than valve amplifiers. The simplest digital emulations of overdrive and distortion effects can be obtained by using a static non-linearity that simulates some form of saturation and clipping.

M-2.31

Write a function that operates a distortion on a guitar input sound using a static non-linearity.

M-2.31 Solution

```
function y=distortion(x,dtype,params);
y=zeros(1,length(x));
for i=1:length(x)
    if dtype=='symm' y(i)=symm_overdrive(x(i));
    elseif dtype=='asymm' y(i)=asymm_overdrive(x(i),params(1),params(2));
    elseif dtype=='exp' y(i)=exp_distortion(x(i));
    end
end
```

Note that this is a naive implementation, that can potentially introduce aliasing in the output signal.

Let us now examine some specific non-linear functions that can be used to realize these effects. *Symmetric* distortion is based on static non-linearities that are odd with respect to the origin, are approximately linear for low input values, and saturate (i.e. progressively decrease their slope) with increasing input signals. As a consequence, these non-linearities produce a symmetric (with respect to positive and negative input values) clipping of the signal. A couple of possible parametrizations which have been proposed in the literature are the following:

$$F(x) = \begin{cases} 2x, & 0 \leq |x| \leq 1/3, \\ \operatorname{sgn}(x) \frac{3-(2-3|x|)^2}{3}, & \frac{1}{3} < |x| \leq \frac{2}{3}, \\ \operatorname{sgn}(x), & \frac{2}{3} < |x| \leq 1, \end{cases} \quad F(x) = \operatorname{sgn}(x) \left(1 - e^{-q|x|} \right), \quad (2.55)$$

where the parameter q in the second equation controls the amount of clipping (higher values provide faster saturation). Both functions are shown in Fig. 2.32(a). The first one is claimed to be well suited for implementing an soft overdrive effect, since it realizes a smooth transition of the linear behaviour for low level signal to saturation for high level sounds, resulting in a warm and smooth sound. The second one realizes a stronger clipping and is claimed to be more effective for implementing a distortion effect. Note that, since these functions are odd, their Taylor expansions only contain odd terms and consequently only odd harmonics are generated.



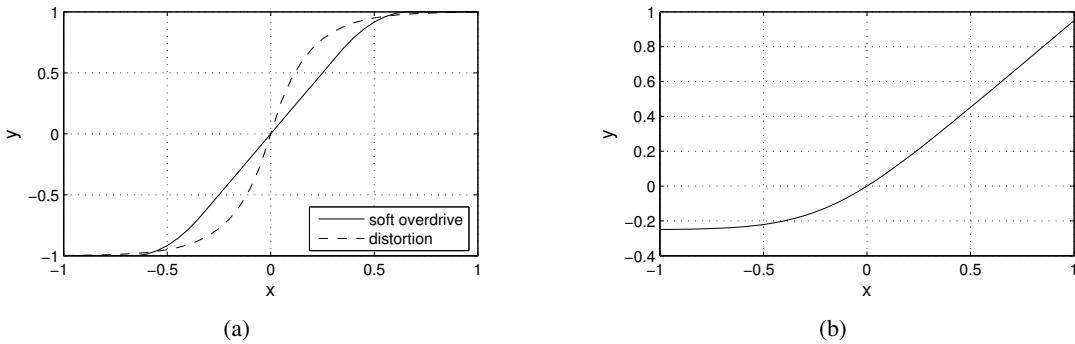


Figure 2.32: Simulation of overdrive and distortion; (a) soft overdrive and exponential distortion (with $q = 6$); (b) asymmetric clipping (with $q = -0.2$ and $d = 8$).

Asymmetric overdrive effects (that are resemblant of the effect of triode valves in the analog domain) are based on distortion curves that clip positive and negative input values in different ways. Since the distortion curve is no longer odd, also even harmonics are generated in this case. A proposal for a function that simulates asymmetric clipping is

$$F(x) = \frac{x - q}{1 - e^{-d(x-q)}} + \frac{q}{1 - e^{dq}}. \quad (2.56)$$

Note that this function is still linear for small input values ($f'(x) \rightarrow 1$ and $f(x) \rightarrow 0$ for $x \rightarrow 0$). The parameter q scales the range of linear behavior (more negative values increase the linear region of operation) and d controls the smoothness of the transition to clipping (higher values provide stronger distortions). A plot of this function is shown in Fig. 2.32(b).

M-2.32

Implement the three functions used in the previous example. For each of them, study the output spectrum when sinusoidal inputs with various amplitudes are provided.

M-2.32 Solution

```
function y=symm_overdrive(x);
if abs(x)<1/3 y=2*x;
elseif abs(x)<2/3 y=sign(x)*(3-(2-3*abs(x))^2)/3;
else y=sign(x);
end
```

2.6.1.4 Non-linear systems with memory

So far in this section we have only examined memoryless non-linear systems. However real non-linear systems are usually systems with memory. As an example, vacuum tubes and solid-state devices that realize analog guitar effects have their internal dynamics, although we have approximated them with simple non-linear instantaneous input-output relations.

In Chapter *Sound modeling: source based approaches* we will see that faithful simulations of non-linear systems can be obtained by writing a set of non-linear differential equations that describe the system dynamics



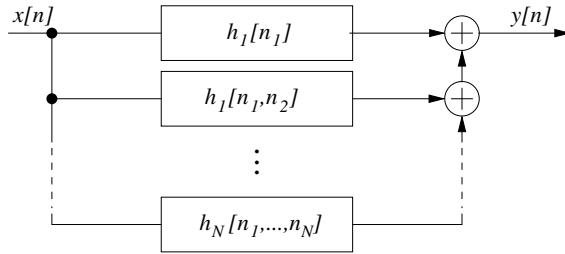


Figure 2.33: Realization of a non-linear system with memory using the Volterra series (truncated at the order N).

(e.g. current-voltage relations at each stage of the circuit) and then solving the system numerically. Here instead we stick to a signal based approach and discuss a generalization of the concepts examined so far.

The Volterra series is a model for non-linear systems, that can be seen on one hand as a generalization of the Taylor series for a non-linear function, and on the other hand as a generalization of the impulse response concept for a LTI system. Given an input signal $x[n]$, the output $y[n]$ of a discrete-time non-linear time-invariant system can be expanded in Volterra series as

$$y[n] = \sum_{n_1=0}^{+\infty} h_1[n_1]x[n - n_1] + \sum_{n_1=0}^{+\infty} \sum_{n_2=0}^{+\infty} h_2[n_1, n_2]x[n - n_1]x[n - n_2] + \dots + \dots + \sum_{n_1=0}^{+\infty} \dots \sum_{n_k=0}^{+\infty} h_k[n_1, \dots, n_k]x[n - n_1] \dots x[n - n_k] + \dots \quad (2.57)$$

The first term of the series corresponds to usual convolution of an impulse response with the input. However now higher terms are also present, all of which perform multiple convolutions and therefore depend in principle on the input at all past instants. Note also that if the multidimensional impulse responses h_k reduce to unit impulses, then the Volterra expansion reduces to a Taylor expansion.

The main advantage in representing a non-linear system through Eq. (2.57) is that various methods exist for estimating the responses h_k from measurements on real systems. If estimates for these responses are available, then Eq. (2.57) also suggests an implementation scheme, depicted in Fig. 2.33.

On the other hand, these kind of representations are useful only for representing systems with mild non-linearities, while for highly non-linear systems the Volterra series does not converge quickly enough and even with many terms it does not provide a sufficiently accurate representation of system behavior.

2.6.2 Multiplicative synthesis

In this section and in the next one we discuss sound synthesis techniques based on amplitude and frequency modulation, that are not derived from models of sound signals or sound production, and are instead based on abstract mathematical descriptions. These techniques provide versatile methods for producing many types of sounds, with great timbral variability, by using a very limited number of control parameters and with low computational costs.

The main drawback of these techniques is that they cannot be embedded in an analysis-synthesis scheme in which parameters of the synthesis model are derived from analysis of real sounds. No intuitive interpretation can be given to the parameter choice as this synthesis technique does not evoke any previous musical experience of the performer. For these reasons these techniques have progressively lost popularity over the years. However they still retain the attractiveness of their own peculiar timbral spaces



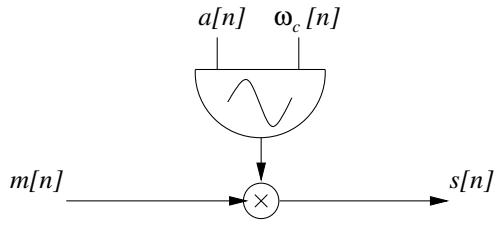


Figure 2.34: Ring modulation with a sinusoidal carrier.

and, besides their historical relevance, they still offer a wide range of original synthesis and processing schemes.

2.6.2.1 Ring modulation

The simplest modulation consists of the multiplication of two signals. In the analog domain, synthesis techniques based on this scheme have been called *ring modulation*. Although precise multiplication of two audio signals is not simple to obtain in the analog domain, it becomes a straightforward task for digital signals. Let $x_1[n]$ and $x_2[n]$ be two input signals, the resulting signal is

$$s[n] = x_1[n] \cdot x_2[n] \quad (2.58)$$

and its spectrum is the convolution of the two input signal spectra, i.e. $S(\omega_d) = [X_1 * X_2](\omega_d)$.

Most typically one of the two signals is a sinusoid with frequency ω_c , and is called the *carrier* signal $c[n]$, while the second signal is the input that will be transformed by the ring modulation and is called the *modulating* signal $m[n]$:

$$x_1[n] = c[n] = \cos(\omega_c n + \phi_c), \quad x_2[n] = m[n]. \quad (2.59)$$

This modulation scheme is shown in Fig. 2.34. Note that the resulting modulated signal $s[n]$ is formally identical to signals with time-varying amplitude examined in other occasions (e.g. sinusoidal oscillators controlled in amplitude). The only but fundamental difference is that in this case the amplitude signal is not a “slow” control signal, but varies at audio rate, and consequently it is perceived in a different way: due to the limited resolution of the human ear, a modulation slower than ~ 20 Hz will be perceived in the time-domain as a time-varying amplitude envelope, whereas a modulation faster than that will be perceived as distinct spectral components. More precisely, the spectrum of $s[n] = c[n]m[n]$ is

$$S(\omega_d) = \frac{1}{2} \left[M(\omega_d - \omega_c)e^{j\phi_c} + M(\omega_d + \omega_c)e^{-j\phi_c} \right], \quad (2.60)$$

i.e. $S(\omega_d)$ is composed of two copies of the spectrum of $M(\omega_d)$, symmetric around ω_c : a lower sideband (LSB), reversed in frequency, and an upper sideband (USB). When the bandwidth of $M(\omega_d)$ extends beyond ω_c , part of the LSB extends to the negative region of the frequency axis, and this part is aliased.

A variant of ring modulation is *amplitude modulation*:

$$s[n] = \{1 + \alpha m[n]\}c[n], \quad S(\omega_d) = C(\omega_d) + \frac{\alpha}{2} \left[M(\omega_d - \omega_c)e^{j\phi_c} + M(\omega_d + \omega_c)e^{-j\phi_c} \right], \quad (2.61)$$

where α is the amplitude modulation index. In this case the spectrum $S(\omega_d)$ contains also the carrier spectral line, plus side-bands of the form (2.60). From the expression for $S(\omega_d)$ one can see that α controls the amplitude of the sidebands.



2.6.2.2 $|\omega_c \pm k\omega_m|$ spectra

Let us consider an example of ring modulation composed by a sinusoidal carrier of the form (2.59) and a periodic modulating signal, $m[n] = \sum_{k=1}^N b_k \cos(k\omega_m n + \phi_k)$, with fundamental frequency ω_m and N harmonic partials with frequencies $k\omega_m$ ($k = 1, \dots, N$). In this case multiplicative synthesis causes every spectral line $k\omega_m$ to be replaced by two spectral lines, one in the LSB and the other one in the USB, with frequencies $\omega_c - k\omega_m$ and $\omega_c + k\omega_m$:

$$s[n] = \sum_{k=1}^N \frac{b_k}{2} \{ \cos [(\omega_c + k\omega_m)n + \phi_k] - \cos [(\omega_c - k\omega_m)n + \phi_k] \}. \quad (2.62)$$

If $\omega_c - k\omega_m < 0$ for some k , then the corresponding spectral line will be aliased around zero. The resulting spectrum has partials at frequencies $|\omega_c \pm k\omega_m|$ with $k = 1, \dots, N$, where the absolute value is used to take into account the possible aliasing around the origin.

Spectra of this kind can be characterized through the ratio ω_c/ω_m , sometimes also called *c/m* ratio. When this ratio is rational (i.e. $\omega_c/\omega_m = N_1/N_2$ with $N_1, N_2 \in \mathbb{N}$ and mutually prime), the resulting sound is periodic: more precisely all partials are multiples of the fundamental frequency $\omega_0 = \omega_c/N_1 = \omega_m/N_2$ and ω_c, ω_m coincide with the N_1 th and N_2 th harmonic partial, respectively. As a special case, if $N_2 = 1$ all the harmonics are present and the components with $k < -N_1$, i.e. with negative frequency, overlap some components with positive k . In general the N_1/N_2 ratio can be considered as an index of the harmonicity of the spectrum. The sound spectrum is more resemblant as a complete harmonic spectrum when the N_1/N_2 ratio is simple. The simplest possible *c/m* ratio is 1/2: in this case the effect of ring modulation is simply that of producing a sound whose fundamental frequency is half that of the modulating sound, with a limited distortion of the overall spectral envelope. This is a kind of *octave divider* effect.

The *c/m* ratios can be grouped in families. All ratios of the type $|\omega_c \pm k\omega_m|/\omega_m$ produce the same components. As an example, the ratios 2/3, 5/3, 1/3, 4/3, 7/3 and so on produce the same set of partials, in which only those that are multiples of 3 are missing. As a consequence, one family of ratio can be identified through a *normal form* ratio, i.e. the smallest ratio (the normal form ratio in the previous example is 1/3).

When the ω_c/ω_m ratio is irrational, the resulting sound is inharmonic. This configuration can be used to create inharmonic sounds, such as bells. As an example if $\omega_c/\omega_m = 1/\sqrt{2}$, the sound contains partials with frequency $\omega_c \pm k\sqrt{2}$ and no implied fundamental pitch is audible. Of particular interest is the case of an ω_c/ω_m ratio approximating a simple rational value, that is,

$$\frac{\omega_c}{\omega_m} = \frac{N_1}{N_2} + \epsilon, \quad \text{with } \epsilon \ll 1. \quad (2.63)$$

In this case the fundamental frequency is still $\omega_0 = \omega_m/N_2$, but partials are shifted from the harmonic series by $\pm\epsilon\omega_m$, so that the spectrum becomes slightly inharmonic. A small shift of ω_c does not change the pitch, but it slightly spread the partials and makes the sound more lively.

2.6.3 Frequency and phase modulation

The definition of *synthesis by frequency modulation (FM)* encompasses an entire family of techniques in which the instantaneous frequency of a *carrier* signal is itself a *modulating* signal that varies at audio rate. We have already discussed oscillators whose frequency varies slowly in time, and is consequently perceived as a varying pitch. In this case we are considering audio-rate frequency changes, which produce radically different effects.



2.6.3.1 A frequency-modulated sinusoidal oscillator

We have already seen in Chapter *Fundamentals of digital audio processing* how to compute the signal phase $\phi[n]$ when the instantaneous frequency $f_0[n]$ is varying at frame rate. We now face the problem of computing $\phi[n]$ when the instantaneous frequency varies at audio rate. A way of approximating $\phi[n]$ is through a first-order expansion.

Recalling that, in continuous time, phase and instantaneous frequency are related through $2\pi f_0(t) = d\phi/dt(t)$ (see Chapter *Fundamentals of digital audio processing*), we can approximate this relation over two consecutive discrete time instants as

$$\frac{d\phi}{dt}((n-1)T_s) \sim 2\pi \left[\frac{f_0(nT_s) + f_0((n-1)T_s)}{2} \right], \quad (2.64)$$

i.e. the phase derivative is approximated as the average of the instantaneous frequency at two consecutive instants. Using this approximation, a first-order expansion of the phase can be approximated in discrete time as

$$\phi[n] = \phi[n-1] + \frac{\pi}{F_s}(f_0[n] + f_0[n-1]). \quad (2.65)$$

M-2.33

Write a function that realizes a frequency-modulated sinusoidal oscillator, with input parameters t_0 (initial time), a (frame-rate amplitude vector) f (audio signal representing the instantaneous frequency vector), and ph_0 (initial phase).

M-2.33 Solution

```
function s=fm_osc(t0,a,f,ph0);

global Fs; global SpF; %global variables: sample rate, samples-per-frame

nframes=length(a); %total number of frames
s=zeros(1,nframes*SpF); %initialize signal vector to 0
lastf=f(1); lastph=ph0; %initialize frequency, phase

for (i=1:nframes)
    phase=zeros(1,SpF); %phase vector in a frame
    for(k=1:SpF) % work at sample rate
        phase(k)=lastph + pi/Fs*(f((i-1)*SpF+k)+lastf); %compute phase
        lastph=phase(k); lastf=f((i-1)*SpF+k); %save last values
    end
    s(((i-1)*SpF+1):i*SpF)=a(i).*sin(phase);
end
s=[zeros(1,round(t0*Fs)) s]; %add initial silence of t0 sec.
```

Compare this function with the `sinosc` function discussed in Chapter *Fundamentals of digital audio processing*. The only difference is that in this case the frequency is given at audio rate. Consequently the phase computation differs.

Although early realizations of FM synthesis were implemented in this fashion, in the remainder of this section we will follow an equivalent “phase-modulation” formulation, according to which the FM oscillator is written as:

$$s[n] = a[n] \cdot \sin(\omega_c[n]n + \phi[n]), \quad (2.66)$$



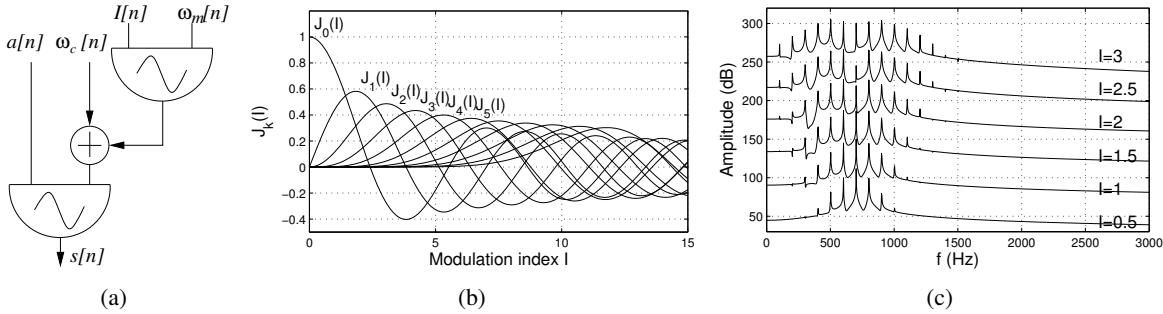


Figure 2.35: Simple modulation: (a) block scheme; (b) the first 10 Bessel functions; (c) spectra produced by simple modulation with $\omega_c = 2\pi 700 \text{ Hz}$, $f_m = 2\pi 100 \text{ Hz}$, and I varying from 0.5 to 3 (for the sake of clarity each spectrum is plotted with a different offset).

where $a[n]$ is the (frame rate) amplitude signal, $\omega_c[n]$ is the (frame rate) *carrier* frequency, and $\phi[n]$ is the (audio rate) modulating signal. In this case the iterative computation used in the example M-2.33 can be substituted by the following:

$$\begin{aligned}\varphi[n] &= \varphi[n-1] + \omega_c[n] + \phi[n], \\ y[n] &= a[n] \cdot \sin(\varphi[n]),\end{aligned}\quad (2.67)$$

where $\varphi[n]$ is a state variable representing the instantaneous phase of the oscillator.

2.6.3.2 Simple modulation

The simplest FM scheme (appropriately termed *simple modulation*) employs in Eq. (2.66) a sinusoidal modulating signal $\phi[n]$ with amplitude $I[n]$ (called *modulation index*) and frequency $\omega_m[n]$:

$$\phi[n] = I[n] \sin(\omega_m[n]n), \quad (2.68)$$

where $I[n]$, $\omega_m[n]$ vary at frame rate. This modulation (shown in Fig. 2.35(a)) produce the signal

$$s[n] = a[n] \sin [\omega_c[n]n + I[n] \sin(\omega_m[n]n)] = a[n] \sum_{k=-\infty}^{+\infty} J_k(I[n]) \sin [(\omega_c[n] + k\omega_m[n])n], \quad (2.69)$$

where $J_k(I[n])$ is the k -th order Bessel function of the first kind, evaluated in the point $I[n]$. From Eq. (2.69) we can see that the spectrum has partials at frequencies $|\omega_c \pm k\omega_m|$ (as already discussed for ring modulation, negative frequencies are aliased around the origin). Each partial has amplitude $J_k(I)$: a plot of the first Bessel functions is shown in Fig. 2.35(b), from which one can see that partial amplitudes are modulated in a very complex fashion when the modulation index I is varied.

Note that an infinite number of partials is generated, so that the signal bandwidth is not limited. In practice however only a few low-order Bessel functions take significantly non-null values for small values of I . As I increases, the number of significantly non-null Bessel functions increases too. A way of characterizing the bandwidth of $s[n]$ is by saying that the number M of lateral spectral lines $|\omega_c \pm k\omega_m|$ that are greater than 1/100 of the nonmodulated signal is given by $M(I) = I + 2.4 \cdot I^{0.27}$: therefore $M(I) \sim I$ for non small I values, and the bandwidth around ω_c is approximately $2I$. Manipulation of the modulation index produces an effect similar to low-pass filtering with varying cut-off frequency, and with smooth variation of the amplitude of partials. Figure 2.35(c) show the spectra produced by simple modulation, with varying modulation index values: as the index increases the energy of the carrier frequency is progressively transferred to the lateral bands, according to the predicted behaviour.



2.6.3.3 Other basic FM schemes

There are many variation of the simple modulation scheme examined above. If the modulating signal is composed of N sinusoids, $\phi[n] = \sum_{i=1}^N I_i[n] \sin(\omega_{m,i}[n]n)$, the corresponding FM scheme is termed *compound* (or *complex*) *modulation* (shown in Fig. 2.36(a)) and the following relation holds:

$$\begin{aligned} s[n] &= a[n] \sin \left[\omega_c[n]n + \sum_{i=1}^N I_i[n] \sin(\omega_{m,i}[n]n) \right] \\ &= a[n] \sum_{k_1, \dots, k_N} \prod_{i=1}^N J_{k_i}(I_i[n]) \sin \left[\left(\omega_c[n] + \sum_{i=1}^N k_i \omega_{m,i}[n] \right) n \right], \end{aligned} \quad (2.70)$$

where the integers k_1, \dots, k_N all vary between $-\infty$ and $+\infty$. Therefore $s[n]$ possesses all the partials with frequencies $|\omega_c \pm k_1 \omega_{m,1} \pm \dots \pm k_N \omega_{m,N}|$ with amplitudes given by the product of N Bessel functions. If the ratios between the $\omega_{m,i}$ s are sufficiently simple, then the spectrum is again of the type $|\omega_c \pm k\omega_m|$. Otherwise the spectrum is highly inharmonic (and takes a noisy character for high index values).

M-2.34

Synthesize a frequency modulated sinusoid in the case of compound modulation, and study the signal spectra when control parameters are varied.

A more complex FM scheme is *nested modulation* (shown in Fig. 2.36(b)), in which a sinusoidal modulator is itself modulated by a second one, i.e. $\phi[n] = I_1[n] \sin [\omega_{m,1}[n]n + I_2[n] \sin(\omega_{m,2}[n]n)]$. In this case the resulting signal is

$$\begin{aligned} s[n] &= a[n] \sin \{ \omega_c[n]n + I_1[n] \sin [\omega_{m,1}[n]n + I_2[n] \sin(\omega_{m,2}[n]n)] \} = \\ &= a[n] \sum_{k=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} J_k(I_1[n]) J_n(kI_2[n]) \sin \{ (\omega_c[n] + k\omega_{m,1}[n] + n\omega_{m,2}[n])n \}. \end{aligned} \quad (2.71)$$

The result can be interpreted as if each partial produced by the modulating frequency $\omega_{m,1}$ were modulated by $\omega_{m,2}$ with modulation index kI_2 . The spectral structure is similar to that produced by two sinusoidal modulators, but with larger bandwidth.

The last FM scheme that we examine is *feedback modulation* (shown in Fig. 2.36(c)), in which past values of the output signal are used as a modulating signal, i.e. $\phi[n] = \beta s[n - n_0]$. If $n_0 = 1$, the modulated signal is

$$s[n] = a[n] \sin (\omega_c[n]n + \beta s[n - 1]) = a[n] \sum_{k=-\infty}^{+\infty} \frac{2}{k\beta} J_k(k\beta) \sin(k\omega_c[n]n), \quad (2.72)$$

and β (called the *feedback factor*) acts as a scale factor or feedback modulation index. For increasing values of β the resulting signal is periodic of frequency ω_c and changes smoothly from a sinusoid to a sawtooth waveform. Moreover one may vary the delay n_0 in the feedback, and observe emergence of chaotic behaviors for suitable combinations of the parameters n_0 and β .

2.6.3.4 FM synthesis of instrumental sounds

As already mentioned earlier in this chapter, one of the main drawbacks of non-linear modulation synthesis approaches (ring modulation, frequency modulation) is that they cannot be embedded into a synthesis-by-analysis framework in which parameters of the synthesis models are derived from analysis of real



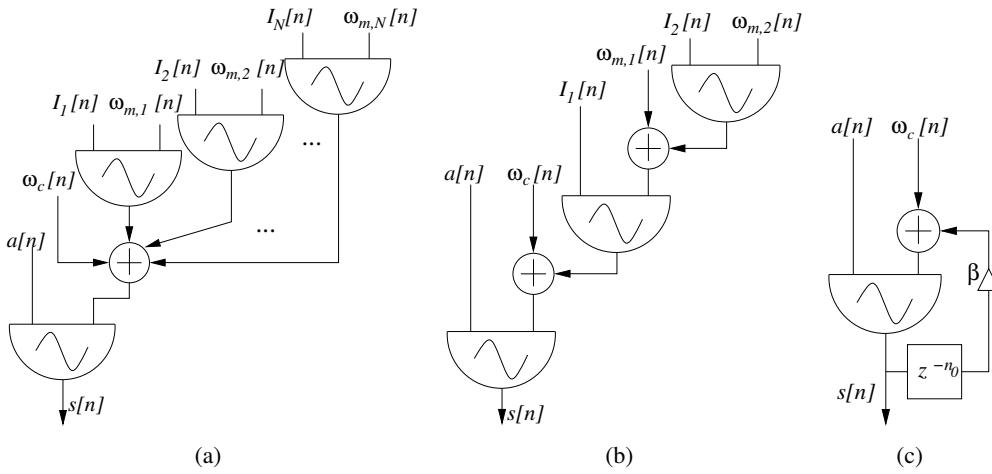


Figure 2.36: Basic FM schemes; (a) compound modulation, (b) nested modulation, and (c) feedback modulation.

sounds. In particular, tuning a FM synthesizer to simulate an acoustic instrumental sound is by no means a trivial task, which can only be accomplished through empirical exploration of the parameter space, intuition, and a lengthy trial and error process.

Nonetheless, since the early FM formulations researchers and musician have been experimenting with FM schemes, parameter values, and envelopes, with the goal of imitating familiar instrumental sounds. Several “FM instruments” have been proposed over the years. A simple example can be given using the basic FM schemes analyzed earlier in this section: specifically, the compound modulation scheme of Fig. 2.36(a) can be used to synthesize a FM piano. Taking $N = 2$ modulating signals, with $\omega_{m,1} \approx \omega_c$ and $\omega_{m,2} \approx 4\omega_c$, then the greatest common divisor is $\omega_{m,1}$ and the frequency modulated sound has a fundamental frequency around $\omega_{m,1}$ and a slightly inharmonic spectrum. If suitable ADSR envelopes are added to this scheme, piano tones can be simulated, using higher modulation indexes for lower tones (inharmonicity is generated especially in low piano strings).

During the 1980’s FM synthesis became a major industry with the introduction of several commercial synthesizers. In particular, it was implemented in the (possibly) most successful synthesizer of all times: the Yamaha DX7. This synthesizer, along with many others based on the same principles, utilized a set of so-called “operators” (6 in the DX7), i.e. amplitude controlled FM oscillators. These operators were connected in a number of different combinations (including simple, compound, nested, and feedback schemes), to produce several so-called “algorithms”. The user had control on all the synthesis parameters, particularly in terms of temporal envelopes, and could produce his/her own sounds besides the instrumental presets. Through the DX7 and its successors, FM synthesis became in a way a trademark of the sound of pop music in the 1980’s.

2.7 Commented bibliography

Among the plethora of available sound synthesis languages, one of the most widely used (and one of the most important historically) is Csound, developed by Barry Vercoe at the Massachusetts Institute of Technology. Csound descends from the family of Music-N languages created by Max Mathews at Bell Laboratories. See [Vercoe, 1993].

A second influential sound synthesis programming paradigm was developed starting from the early

1980's, mainly by Miller Puckette, and is today represented in three main software implementation: Max/MSP, jmax, and Pd. The "Max paradigm" (so named in honor of Max Mathews) is described by Puckette [Puckette, 2002] as a way of combining pre-designed building blocks into sound-processing "patches", to be used in real-time settings. This includes a scheduling protocol for both control- and audio-rate computations, modularization and component intercommunication, and a graphical environment to represent and edit patches.

About sampling and wavetable synthesis. Contemporary music synthesizer are still based on these techniques, and allow ever increasing quality thanks to the ever increasing availability of storage capacity. A multi-sampled instrument can occupy several Gb. From the point of view of music history these techniques are rooted in several works from the '50s, especially by composer Pierre Schaefer and coworkers, who experimented with the use of recorded environmental sound as sonic material in their compositions. This approach to musical composition has been termed *musique concrète*.

About granular synthesis. The scientific foundations of these approaches can be found in the work of hungarian physicist Dennis Gabor (see [Gabor, 1947]). The composer Iannis Xenakis developed this method in the field of analog electronic music. Starting from Gabor theory, Xenakis suggested a compositional method based on the organization of the grains by means of screen sequences, which specify frequency and amplitude parameters of the grains at discrete points in time. In this way a common conceptual approach is used both for micro and macro musical structure: "All sound, even continuous musical variation, is conceived as an assemblage of a large number of elementary sounds adequately disposed in time. In the attack, body and decline of a complex sound, thousands of pure sounds appear in a more or less short time interval of time Δt " [Xenakis, 1992].

The most widely treated case is (*asynchronous granular synthesis*), where simple grains are distributed irregularly. A classic introduction to the topic is [Roads, 1991]. In particular, figure 2.4 in this chapter is based on an analogous figure in [Roads, 1991]. In another classic work, Truax [Truax, 1988] describe the granulation of recorded waveforms.

About recent *corpus-based* concatenative synthesis techniques. A review is provided in [Schwarz, 2007]

About overlap-add techniques. The pitch-synchronous overlap-add algorithm for time-stretching was introduced by Moulines and Charpentier [1990] in the context of speech processing applications.

Additive synthesis was one of the first sound modeling techniques adopted in computer music and has been extensively used in speech applications as well. The main ideas of the synthesis by analysis techniques that we have reviewed date back to the work by McAulay and Quatieri [McAulay and Quatieri, 1986]. In the same period, Smith and Serra started working on "sines-plus-noise" representations, usually termed *SMS* (Spectral Modeling Synthesis) by Serra. A very complete coverage of the topic is provided in [Serra, 1997]. The extension of the additive approach to a "sines-plus-transients-plus-noise" representation is more recent, and has been proposed by Verma and Meng [Verma and Meng, 2000].

Subtractive synthesis techniques became extremely popular in the 1960's and 1970's, with the advent of analog voltage controlled synthesizers. The Moog synthesizers were especially successful and were based on a range of signal generators, filters, and control modules, which could be easily interconnected to each other. The central component was the voltage-controlled oscillator (VCO), which could produce a variety of waveforms and could be connected to other modules such as voltage-controlled amplifiers (VCA), voltage-controlled filters (VCF), envelope generators, and other devices. Moog's innovations were first presented in [Moog, 1965]. Several techniques for antialiasing digital oscillators, to be used in digital emulation of analog subtractive synthesis, are discussed in [Välimäki and Huovilainen, 2007].

A tutorial about filter design techniques, including normalization approaches that use L^1 , L^2 , and L^∞ norms of the amplitude response, is [Dutilleux, 1998]. Introductions to formant speech synthesis and linear prediction techniques and their applications in speech technology can be found in many textbook. See e.g. [Rabiner and Schafer, 1978] (our Fig. 2.21 is based on a similar figure in this book). Another



useful reference on the topic is [Deller et al., 1993]. One technique that is alternative to linear prediction and widely used is *digital all-pole modeling (DAP)* [El-Jaroudi and Makhoul, 1991].

The use of frequency modulation as a sound synthesis algorithm was first experimented by Chowning [1973] (later reprinted in [Roads and Strawn, 1985]), although these techniques had already been used for decades in electrical communications. While performing experiments on different extents of vibrato applied to simple oscillators, Chowning realized that when vibrato rates entered the audio range, dramatic timbral changes were produced. Soon after FM became very popular and were applied also to the simulation of real sounds: see [Schottstaedt, 1977] (also reprinted in [Roads and Strawn, 1985]). Our example of a synthetic piano tone at the end of Sec. 2.6.3 is taken from this latter work. The FM algorithms used for the DX7 synth are discussed at length in [Chowning and Bristow, 1986].

References

- John Chowning. The synthesis of complex audio spectra by means of Frequency Modulation. *J. Audio Engin. Soc.*, 21(7), 1973.
- John Chowning and David Bristow. *FM Theory and applications*. Yamaha Music Foundation, Tokio, 1986.
- John R Deller, John G. Proakis, and John. H.L. Hansen. *Discrete-Time Processing of Speech Signals*. Macmillan, New York, 1993.
- P. Dutilleux. Filters, Delays, Modulations and Demodulations: A Tutorial. In *Proc. COST-G6 Conf. Digital Audio Effects (DAFx-98)*, pages 4–11, Barcelona, 1998.
- Amro El-Jaroudi and John Makhoul. Discrete all-pole modeling. *IEEE Trans. Sig. Process.*, 39:411–423, Feb. 1991.
- Dennis Gabor. Acoustical quanta and the theory of hearing. *Nature*, 159(4044):591–594, 1947.
- R. McAulay and T. F. Quatieri. Speech Analysis/Synthesis Based on a Sinusoidal Speech Model. *IEEE Trans. Acoust., Speech, and Sig. Process.*, 34:744–754, 1986.
- Robert A. Moog. Voltage-controlled electronic music modules. *J. Audio Eng. Soc.*, 13(3):200–206, July 1965.
- E..... Moulines and F..... Charpentier. Pitch synchronous waveform processing techniques for text to speech synthesis using diphones. *Speech Communication*, 9(5/6):453–467, 1990.
- M. Puckette. Max at seventeen. *Computer Music J.*, 26(4):31–43, 2002.
- L. R. Rabiner and R. W. Schafer. *Digital Processing of Speech Signals*. Prentice-Hall, Englewood Cliffs, NJ, 1978.
- C. Roads. Asynchronous granular synthesis. In G. De Poli, A. Piccialli, and C. Roads, editors, *Representations of Musical Signals*, pages 143–186. MIT Press, 1991.
- C. Roads and J. Strawn, editors. *Foundations of Computer Music*. MIT Press, 1985.
- William Schottstaedt. The simulation of natural instrument tones using frequency modulation with a complex modulating wave. *Computer Music J.*, 1(4):46–50, 1977.
- Diemo Schwarz. Corpus-based concatenative synthesis. *IEEE Signal Processing Magazine*, pages 92–104–, Mar. 2007.
- X. Serra. Musical sound modeling with sinusoids plus noise. In C. Roads, S. Pope, A. Piccialli, and G. De Poli, editors, *Musical Signal Processing*, pages 91–122. Swets & Zeitlinger, 1997. <http://www.iua.upf.es/~xserra/articles/msm/>.
- B. Truax. Real-time granular synthesis with a digital signal processor. *Computer Music J.*, 12(2):14–26, 1988.
- Vesa Välimäki and Antti Huovilainen. Antialiasing oscillators in subtractive synthesis. *IEEE Signal Processing Magazine*, 24 (2):116–125, Mar. 2007.
- B. Vercoe. Csound: A manual for the audio processing system and supporting programs with tutorials. Technical report, Media Lab, M.I.T., Cambridge, Massachusetts, 1993. Software and Manuals available from <ftp://ftp.maths.bath.ac.uk/pub/dream/>.



T. S. Verma and T. H. Y. Meng. Extending Spectral Modeling Synthesis with Transient Modeling Synthesis. *Computer Music J.*, 24(2):47–59, 2000.

Iannis Xenakis. *Formalized music: Thought and Mathematics in Composition*. Pendragon Press, Stuyvesant, NY, 1992.



Chapter 3

Sound modeling: source-based approaches

Federico Avanzini

Copyright © 2005-2018 Federico Avanzini

except for paragraphs labeled as *adapted from <reference>*

This book is licensed under the CreativeCommons Attribution-NonCommercial-ShareAlike 3.0 license. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/>, or send a letter to Creative Commons, 171 2nd Street, Suite 300, San Francisco, California, 94105, USA.

3.1 Introduction

It was 1971 when Hiller and Ruiz envisioned the possibility of using numerical simulations of the wave equation for sound synthesis applications.

[...] This is a completely new approach to electronic sound synthesis insofar as the starting point is the physical description of the vibrating object [...]

A decade later McIntyre, Schumacher, and Woodhouse published their classic study on the use of non-linear maps for modeling the generation of self-sustained oscillations in musical instruments.

[...] a fast minicomputer could produce results at a cycle rate in the audible range. The result would perhaps have some novelty: an electronic musical instrument based on a mathematical model of an acoustic instrument [...]

Today the algorithms described by these authors can be easily implemented in real-time on general-purpose hardware, and it is common practice to use the term *physical modeling* to refer to sound modeling techniques in which the synthesis algorithms are designed based on a description of the physical phenomena involved in sound generation.

Direct sound representations, that are merely based on a description of the sound waveform, do not contain information about the way the sound has been generated and processed by the surrounding environment before arriving to the listener's ear. Sampling in time the sound signal does not assume any underlying structure, or process, or generative model, in sound representation. The symbolic description is extremely poor, and as a consequence very little interaction with the sound representations is allowed. Although signal processing techniques can provide meaningful modifications (e.g. pitch shift, time stretching), sampling is basically a *static*, low-level description of sound.

High level representations of sound signals are necessarily associated with some abstract paradigms that underlie sound production. As we have seen previously, when trying to develop a taxonomy of sound synthesis methods a first distinction can be traced between *signal models* and *source models*. Any algorithm which is based on a description of the sound pressure signal and makes no assumptions on the generation mechanisms belongs to the class of signal models. Additive synthesis is a good example of a signal model: as already mentioned, one major drawback of this technique is its enormous number of control parameters: at least one amplitude and one pitch envelopes have to be specified for each partial. Moreover, the sound representation has not a strong *semantic* interpretation, since these parameters do not have a high-level meaning. Subtractive synthesis with its source-filter structure provides in a sense a more semantic description of sound: in certain cases the two blocks can be given a physical interpretation in terms of an exciting action and a resonating object, respectively. As an example, in the case of LPC based speech synthesis the broadband input signal can be interpreted as a glottal source signal, and the shaping filter represents the action of the vocal tract. However, in many other cases this interpretation does not hold, and the control parameters in the model (e.g., the filter coefficients) do not have a high-level meaning.

Source models aim at describing the physical objects and interactions that have generated an acoustic event rather than the acoustic signal itself. This modeling approach often gives rise to rather complex descriptions, that can lead to computationally expensive numerical algorithms. Several modeling paradigms and techniques are available in the literature for deriving efficient implementations of such descriptions, including lumped/distributed modeling, waveguide structures, finite difference methods, and so on. The following sections describe in detail a few of these approaches. Here it is worth discussing another aspect, i.e. that of control. A direct consequence of assuming a source-based approach is that the resulting control parameters have a straightforward physical interpretation: typical parameters in the models are associated with masses, hardness/softness characteristics, blowing pressures, lengths: such a semantic representation can in principle allow more intuitive interaction.

Source-based sound modeling paradigms are often grouped into two broad categories, namely *lumped* and *distributed* models. Generally speaking, distributed models are more often used for describing vibrating bodies or air volumes where forces and matter depend on (and propagate along) both time and space. One-, two- and three-dimensional resonators (such as strings, bars, acoustical bores, membranes, plates, rooms, etc.) can be treated as continuous distributed systems, and mathematically described by means of Partial Differential Equations (*PDEs*). One of the most popular distributed modeling approaches is *waveguide modeling*, which will be discussed in detailed in Sec. 3.4.

Although waveguides are extremely successful in modeling nearly elastic mediums, where the D'Alembert equation or some of its generalizations hold, they are not equally good in dealing with systems where these hypothesis are not met. As an example, oscillations in a bar are governed by the so called Euler-Bernoulli equation, for which no traveling-waves schematization can be assumed. One possible approach for dealing with such systems is using *finite difference* or *finite elements* methods. These time-domain techniques are based on direct discretization of the PDEs and consequently have high computational costs. On the other hand, when properly applied they provide stable and very accurate numerical systems.

As opposed to distributed models, lumped models are used when a physical system can be conveniently described without explicitly considering its extension in space. As an example, a mechanical resonating body may be described in terms of ideal masses or rigid elements, connected to each other with spring and dampers, and possibly non-linear elements. Similar considerations may apply to electrical circuits and even to certain acoustic systems. The resulting models are naturally described in the time domain, in terms of Ordinary Differential Equations (*ODEs*). Sec. 3.5 discusses lumped modeling approaches, and includes an introduction to modal synthesis. Defining modal synthesis as a lumped modeling approach may be questionable, since the modal formalism incorporates a “spatial” representation



(e.g. it is possible to inject a force in a specific point of a modal resonator, or to measure its displacement in a specific point). On the other hand, representing a resonator as a combination of a finite number of modes corresponds to approximating the resonator as a mesh of point masses connected with strings and dampers, and in this sense modal synthesis may be regarded as a lumped modeling approach.

3.2 Physical structures and models

3.2.1 Simple vibrating systems and normal modes

Sound is produced by mechanical, acoustical, or electrical vibrations that ultimately generate an acoustic pressure signal that reaches our ear. In this section we review the most elementary oscillating systems and their properties.

3.2.1.1 Oscillators

The simplest physical oscillating system is the damped second-order (or harmonic) oscillator. A generic oscillator of this kind is described by the following linear differential equation:

$$\ddot{x} + 2\alpha\dot{x} + \omega_0^2 x = u_{\text{ext}}(t), \quad (3.1)$$

where u_{ext} is an external driving signal. The general solution of the homogeneous equation (i.e., Eq. (3.1) with $u_{\text{ext}} = 0$) is given by

$$x(t) = a_0 e^{-\alpha t} \cos(\omega_r t + \phi_0), \quad (3.2)$$

where $\omega_r = \sqrt{\omega_0^2 - \alpha^2}$. The parameters a_0 and ϕ_0 are uniquely determined by the initial conditions $x(0)$, $\dot{x}(0)$. In particular the impulse response of the system corresponds to initial conditions $x(0) = 0$ and $\dot{x}(0) = 1$, and is given by $h(t) = e^{-\alpha t} \sin(\omega_r t)/\omega_r$.

An electrical system representing a damped harmonic oscillator is the RLC circuit (Fig. 3.1(a)).

$$L \frac{d^2 i}{dt^2}(t) + R \frac{di}{dt} + \frac{1}{C} i = 0, \quad (3.3)$$

where i is the current in the circuit, L , R , C are the inductance, resistance, and capacitance of in the circuit, respectively. This is an equation of the form (3.1), with $\alpha = R/2L$, $\omega_0^2 = 1/LC$. Therefore it has a solution of the form (3.2).

In the mechanical case, an instance of damped harmonic oscillator is the mass-spring-damper system depicted in Fig. 3.1(b):

$$m\ddot{x}(t) + r\dot{x}(t) + kx(t) = 0, \quad (3.4)$$

where x is the displacement signal, m , r , k are the mass, mechanical resistance, and spring stiffness. Again this is an equation of the form (3.1), with $\alpha = r/2m$, $\omega_0^2 = k/m$. Therefore it has a solution of the form (3.2).

In certain situations, acoustic systems can also be described in terms of lumped elements that are equivalent to resistance, capacitance, and inductance. The variables involved in this case are air-flow (or volume velocity) $u(t)$, measured in m^3/s , and acoustic pressure $p(t)$, measured in Pa. When the dimensions of an acoustic element are much less than the sound wavelength, then the acoustic pressure, p can be assumed constant over the element. In this case, the acoustic behavior of the element is, at least at low frequencies, very simple. In particular, the Helmholtz resonator (depicted in Fig. 3.1(c)) behaves to a good degree of approximation as a second order oscillator. We analyze this systems in terms of three main elements: the opening, the neck, and the cavity.



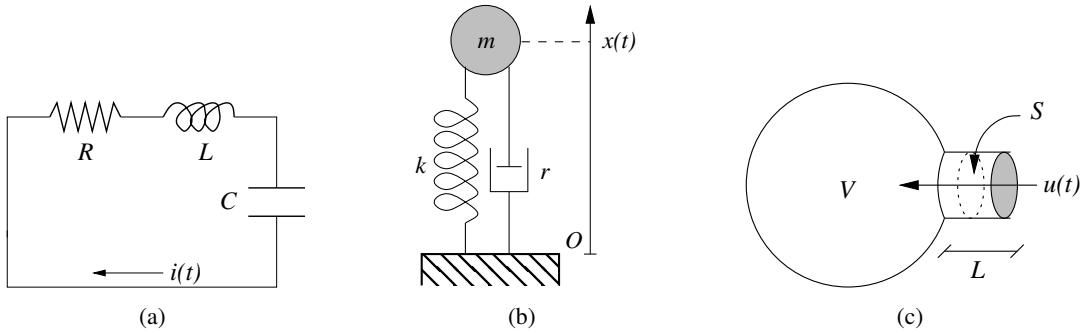


Figure 3.1: Second order electrical, mechanical, and acoustic oscillators; (a) a RLC circuit; (b) a mass-spring-damper system; (c) a Helmholtz resonator.

Resistive phenomena are observed during the passage of acoustic airflow through the opening, due to a pressure difference $\Delta p_{\text{op}}(t)$: the flow behavior is dominated by viscous and thermal losses and it is reasonably assumed to be in phase with the acoustic pressure. Therefore the relation $\Delta p_{\text{op}}(t) = Ru(t)$ holds at the opening where the constant R is termed *fluid-dynamic resistance*. Simple inertial behaviors are observed in the cylindrical neck. The air mass inside this tube is $m = \rho_{\text{air}}SL$ (ρ_{air} being the air density, S the cross-sectional area, and L the length). If a pressure difference $\Delta p_{\text{tube}}(t)$ is applied at the tube ends, the enclosed air behaves like a lumped mass driven by the force Δp_{tube} , and Newton's law implies

$$\Delta p_{\text{tube}}(t) = \rho_{\text{air}}SL \cdot \ddot{v}(t), \quad (3.5)$$

where the relation $u(t) = Sv(t)$ has been used, and $v(t)$ indicates particle velocity. Finally, the cavity has an elastic behavior. Consider the volume $V(t)$ of air inside the cavity: the contraction $dV(t)$ caused by a pressure difference $\Delta p_{\text{cav}}(t)$ is such that $-\rho_{\text{air}}c^2 \cdot dV/V = \Delta p_{\text{cav}}$. As a consequence, a new air volume $-dV$ can enter the cavity. By definition, this equals the integral of $u(t)$ over time, therefore

$$\begin{aligned} -dV(t) &= \int_0^t u(t')dt' = \frac{V}{\rho_{\text{air}}c^2} \Delta p_{\text{cav}}(t). \\ S\Delta p_{\text{cav}}(t) &= -\frac{\rho_{\text{air}}S^2c^2}{V} \int_0^t v(t')dt', \end{aligned} \quad (3.6)$$

which represent a linear spring with stiffness $\rho_{\text{air}}S^2c^2/V$. Both the air mass in the tube and the resistance at the opening impede the same flow u , and are therefore in a “series” connection. This flow u enters the cavity, so that the the volume is in series with the other two. The resulting equation for the particle displacement x is

$$(\rho_{\text{air}}SL) \cdot \ddot{x}(t) + R\dot{x}(t) + \frac{\rho_{\text{air}}S^2c^2}{V}x(t) = 0. \quad (3.7)$$

Again equation of the form (3.1). Therefore solution of the form (3.2).

3.2.1.2 Impedance

The examples in the previous section show that in a large class of systems it is possible to construct pairs of variables (often defined as *Kirchoff variables*) with the property that their product has the dimensions of power ($\text{Kg m}^2/\text{s}^3$). In electrical systems such a pair of variables is given by (v, i) , voltage and current. Integro-differential relations can be found that relate these two variables, in particular three elementary



Electrical		Mechanical		Acoustic	
Current i (A)		Velocity v (m/s)		Flow u (m^3/s)	
Voltage v (V)		Force f (N)		Pressure p (Pa)	
(Resistance) R	$\left(\frac{\text{Kg}\cdot\text{m}^2}{\text{s}}\right)$	(Damping) r	$\left(\frac{\text{Kg}}{\text{s}}\right)$	(Opening) R	$\left(\frac{\text{Kg}}{\text{m}^2\cdot\text{s}}\right)$
(Capacitance) $1/sC$		(Spring) k/s		(Cavity) $\rho_{air}c^2/Vs$	
(Inductance) s/L		(Mass) $m \cdot s$		(Bore) $\rho_{air}Ls/S$	

Table 3.1: Summary of analogies in electrical, mechanical and acoustical systems.

relations define the fundamental quantities resistance R , inductance L and capacitance C . In the Laplace domain, the integro-differential equations are turned into simple algebraic relations:

$$V(s) = R \cdot I(s), \quad V(s) = sL \cdot I(s), \quad V(s) = \frac{1}{sC} I(s). \quad (3.8)$$

These are particular examples of a more general relation in linear electric circuits:

$$V(s) = Z(s)I(s), \quad (3.9)$$

where the quantity $Z(s)$ is called *impedance* of the circuit and is defined as the ratio between the Laplace transforms of voltage and current intensity. The inverse of $Z(s)$ is called *admittance*, and it is usually denoted as $\Gamma(s) = Z(s)^{-1}$.

Similar considerations apply to mechanical systems. Force f (Kg m/s^2) and velocity v (m/s) are the mechanical Kirchhoff variables, since their product is a power. Again, the ratio of these two variables in the Laplace domain is defined as (mechanical) *impedance*, and its inverse is the (mechanical) admittance. In the mechanical oscillator described above we have already introduced the three mechanical equivalents of resistance, capacitance and inductance. The direct proportionality $f(t) = rv(t)$ defines ideal linear viscous forces, and by comparison with the first of Eqs. (3.8) r can be regarded as a mechanical resistance. The inertial mass m of a non-relativistic body is defined as the ratio between the total force acting on it and its acceleration, i.e. $f(t) = ma(t) = m\dot{v}(t)$, or $F(s) = msV(s)$ in the Laplace domain, and by comparison with the second equation in (3.8) m can be regarded as a mechanical inductance. Finally, in an ideal linear spring the elastic force is proportional to the elongation of the spring: $f(t) = kx(t) = k \int_0^t v(t')dt'$, or $F(s) = k/s V(s)$ in the Laplace domain, and by comparison with the third equation in (3.8) the stiffness k can be regarded as a mechanical capacitance. Therefore the aggregate impedance $Z(s)$ of a second-order mechanical oscillator is $Z(s) = ms + k/s + r$.

As far as acoustic systems are concerned, acoustic pressure p (Kg/ms^2) and volume velocity u (m^3/s) are the acoustic Kirchhoff variables, since their product is a power. Again, the ratio of these two variables in the Laplace domain is defined as (acoustic) *impedance*, and its inverse is the (acoustic) admittance. In the Helmholtz resonator described above we have already introduced the three acoustic equivalents of resistance, capacitance and inductance. More precisely, fluid-dynamic resistance is associated to viscous and thermal losses at narrow openings: $p(t) = Ru(t)$. Fluid-dynamic *inductance* is associated to short, open tubes: $p(t) = \rho_{air}L/S \cdot \dot{u}(t)$, or $P(s) = \rho_{air}Ls/S \cdot U(s)$ in the Laplace domain. Fluid-dynamic *capacitance* is associated with enclosed air volumes: $p(t) = \rho_{air}c^2/V \cdot \int u(t')dt'$, or $P(s) = \rho_{air}c^2/(Vs) \cdot U(s)$ in the Laplace domain.

Table 3.1 summarizes the main analogies between electrical, mechanical, and acoustic systems, that we have discussed throughout this section.



3.2.1.3 Coupled oscillators and modal decomposition

We have examined above the behavior of a single second-order oscillator. A way of describing more complex oscillating systems is to represent them as combinations of the simple elements described so far. As an example an oscillating mechanical system (a string, a membrane, etc.) can be described in terms of point masses coupled through linear springs and dampers. Therefore the system is described as a set of coupled second-order differential equations.

It is known that, under general hypotheses, one can find a change of variables such that the set of coupled equations is turned into a set of *uncoupled* second-order equations. In order to clarify this concept, let us look at the following simple mechanical example in which two point masses are connected to each other and to the “walls” through three springs:

$$\begin{aligned} m\ddot{x}_1(t) + kx_1(t) + k(x_1 - x_2) &= 0, \\ m\ddot{x}_2(t) + kx_2(t) + k(x_2 - x_1) &= 0. \end{aligned} \quad (3.10)$$

If we introduce a suitable set of variables $q_{1,2}$ in place of $x_{1,2}$, the above equations can be decoupled, or diagonalized:

$$\begin{aligned} \ddot{q}_1(t) &= -\omega_0^2 q_1(t), \\ \ddot{q}_2(t) &= -3\omega_0^2 q_2(t), \end{aligned} \quad (3.11)$$

with $q_1 = x_1 + x_2$, $q_2 = x_1 - x_2$, $\omega_0^2 = k/m$. The *normal modes* q_i ($i = 1, 2$) are uncoupled and the x_i are linear combinations of the q_i .

This simple example can be extended to more complicated systems, composed N masses coupled through springs and dampers. One can in general reformulate the system in terms of *normal modes* of oscillation, and the oscillation of each point mass can be seen as a linear combination of N normal modes, each of which obeys the equation of a second-order (damped) harmonic oscillator. We will return on these concept in Sec. 3.5.2, when discussing modal synthesis.

3.2.2 Continuous vibrating systems and waves

In the previous section we have examined oscillating systems constructed with lumped elements (e.g. resistances, capacitances, inductances, and their mechanical and acoustic counterparts), and are therefore represented by a finite and discrete set of points in space (e.g. a set of point masses). In this section we examine vibrating systems that are distributed continuously in space, and are therefore described by partial differential equations involving both space and time, rather than set of ordinary differential equations in time.

3.2.2.1 The one-dimensional D'Alembert equation

Vibrational phenomena in an ideal *elastic* medium are described by the D'Alembert equation, whose one-dimensional version is written as

$$\frac{\partial^2 y}{\partial x^2}(x, t) = \frac{1}{c^2} \frac{\partial^2 y}{\partial t^2}(x, t). \quad (3.12)$$

This equation holds, for instance, in an ideal string of length L , linear mass density μ and tension T . In this case the variable $x \in [0, L]$ stands for position along string length and y stands for *transversal* displacement of the string. The constant c has the value $\sqrt{T/\mu}$ and has the dimensions m/s of a velocity. A full derivation of Eq. (3.12) for the ideal string can be found in many textbooks: roughly speaking, the



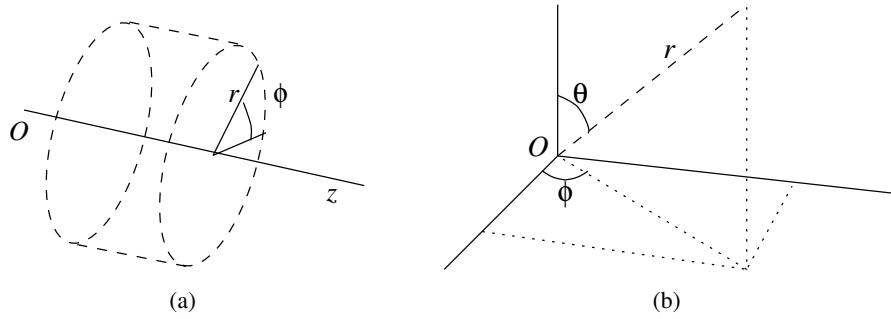


Figure 3.2: Illustration of (a) cylindrical and (b) spherical coordinates.

two main assumptions are that (i) the infinitesimal string segment dx moves only in the vertical direction, so that its acceleration can be computed using only the transverse component of the tension as the acting force; and (ii) the amplitude of the vibrations is very small.

There are interesting cases where acoustic disturbances can be assumed to be one-dimensional up to a reasonable approximation. Propagation of acoustic pressure in a cylindrical or in a conical tube is an example. Using cylindrical coordinates (see Fig. 3.2(a)), one can show that for cylindrical bores one-dimensional *longitudinal* pressure waves in the *z* direction are described using Eq. (3.12), with *z* in place of *x* and with *y* representing acoustic pressure. Using spherical coordinates (see Fig. 3.2(b)), one can show that for conical bores one-dimensional *spherical* pressure waves are described through the equation

$$\frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial R}{\partial r} \right) (r, t) = \frac{1}{c^2} \frac{\partial^2 R}{\partial t^2} (r, t), \quad (3.13)$$

in which $R(r)$ represents acoustic pressure, and the Laplacian operator is expressed in spherical coordinates as $\nabla^2 = \frac{1}{r^2} \frac{\partial}{\partial r} (r^2 \frac{\partial}{\partial r}) + \frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} (\sin \theta \frac{\partial}{\partial \theta}) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2}{\partial \phi^2}$. Using the substitution $R = \tilde{R}/r$, it is easily seen that Eq. (3.13) reduces to the one dimensional D'Alembert equation (3.12) for the variable \tilde{R} .

3.2.2.2 Traveling wave solution

A fundamental property of Eq. (3.12) is that it describes *propagation* phenomena. This statement can be proved by factoring the equation as follows:

$$\left(\frac{\partial}{\partial x} - \frac{1}{c} \frac{\partial}{\partial t} \right) \left(\frac{\partial}{\partial x} + \frac{1}{c} \frac{\partial}{\partial t} \right) y = 0. \quad (3.14)$$

From this factorization it is easily seen that generic solutions take the form

$$y(x, t) = y^+(ct - x) + y^-(ct + x). \quad (3.15)$$

This is the solution to Eq. (3.12) originally proposed by D'Alembert himself. The two functions y^\pm describe waveforms that translate rigidly with velocity *c*, in the right-going and left-going directions, respectively. Their shape is determined by the boundary conditions (in space) and the initial conditions (in time). As an example, if *y* represents the displacement of a vibrating string the initial conditions are represented by an initial displacement and an initial velocity:

$$y_0(x) = y(x, 0), \quad v_0(x) = \frac{\partial y}{\partial t}(x, 0). \quad (3.16)$$

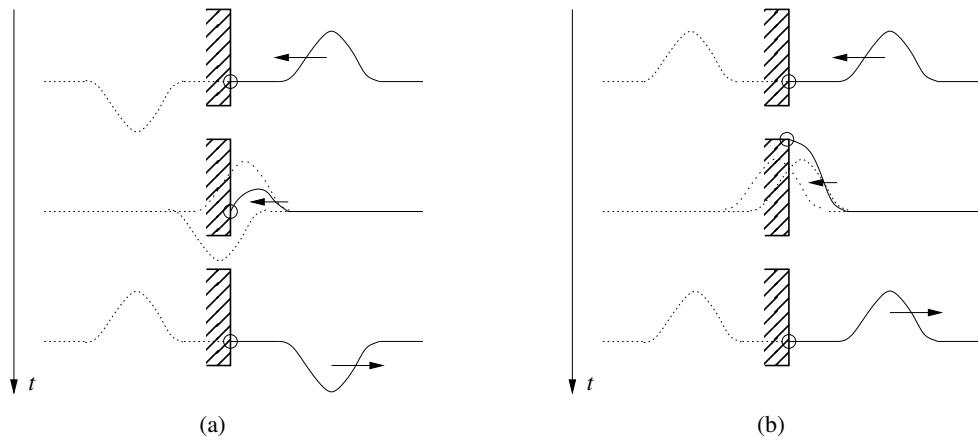


Figure 3.3: Boundary conditions and wave reflections; (a) fixed string end and negative wave reflection; (b) free string end and positive wave reflection.

Boundary conditions impose constraints on the solution at the boundary of its domain. As an example, if y represents the displacement of a vibrating string boundary conditions impose values for y and its derivatives at the boundary points $x = 0$ and $x = L$. The two most common boundary conditions for a string are the *fixed end* condition and the *free end* condition, which read as follows for the boundary point $x = 0$ (similar equations can be written for the boundary point $x = L$)

$$\begin{aligned} \text{Fixed end: } & y(x, t)|_{x=0} = 0, \quad \Rightarrow \quad y^+(ct) = -y^-(ct); \\ \text{Free end: } & \frac{\partial y}{\partial x}(x, t)\Big|_{x=0} = 0, \quad \Rightarrow \quad y^+(ct) = y^-(ct). \end{aligned} \quad (3.17)$$

These equations show that boundary conditions imply “reflection” conditions on the traveling waves y^\pm (see Fig. 3.3).

3.2.2.3 Waves and modes

A different analysis of the wave equation was proposed by Fourier, who proved that the general solution to Eq. (3.12) can be regarded as a superposition of a numerable set of so-called *stationary waves*.

We exemplify the Fourier analysis in the case of an ideal string with fixed-end boundary conditions $y(x, t)|_{x=0, L} \equiv 0$. We search for particular solutions $y(x, t) = s(x)q(t)$, which we call stationary waves, since they have a shape in space that is determined by the spatial function $s(x)$ and is modulated in time by the temporal function $q(t)$.

By substituting the generic stationary wave solution into Eq. (3.12), one finds that the functions s and q must satisfy suitable differential equations:

$$\frac{s''}{s}(x) = \frac{1}{c^2} \frac{\ddot{q}}{q}(t) \quad \Rightarrow \quad s''(x) = \alpha s(x), \quad \ddot{q}(t) = c^2 \alpha q(t), \quad (3.18)$$

for some $\alpha \in \mathbb{R}$. This last equation follows from the fact that s''/s is a function of space only, while \ddot{q}/q is a function of time only. Therefore these ratios must necessarily equal to a constant α .

Now look at the spatial equation. In order for the boundary conditions to be satisfied s has necessarily to be a non-monotonic function and consequently the condition $\alpha < 0$ must hold, so that s obeys the



equation of a second-order oscillator (otherwise $s(x)$ would be an exponential function). Moreover, since it has to be $s(0) = s(L) = 0$, only a numerable set of spatial frequencies are allowed for s :

$$s(x) = \sqrt{\frac{2}{L}} \sin(k_n x), \quad \text{with } k_n = \frac{n\pi}{L}, \quad (3.19)$$

where $\sqrt{2/L}$ is just a normalization factor.

Once the spatial equation has been solved, the temporal equation gives

$$q(t) = A \sin(\omega_n t + \phi), \quad \text{with } \omega_n = ck_n = \frac{n\pi c}{L}, \quad (3.20)$$

where A and ϕ depend on initial conditions. Again, only a numerable set of temporal frequencies $\omega_n = ck_n$ are allowed. Spatial and temporal frequencies are proportional to each other through the constant c .

In conclusion we have obtained the following stationary waves, or *normal modes*:

$$y_n(x, t) = \sqrt{\frac{2}{L}} \sin(\omega_n t + \phi_n) \sin(k_n x). \quad (3.21)$$

The general solution to Eq. (3.12) can be expressed as a linear combination of these modes:

$$y(x, t) = \sum_{n=1}^{+\infty} A_n y_n(x, t), \quad (3.22)$$

where A_n , ϕ_n are determined by the initial conditions. This latter equation re-states what we already know: a periodic signal, such as the one generated in an ideal string with ideal boundary conditions, can be expressed as a series of harmonically-related sinusoidal signals.

Note that the Fourier solution, expressed in term of normal modes, and the D'Alembert solution, expressed in terms of traveling waves, are equivalent. In fact a standing wave $y_n(x, t)$ can be viewed as a superposition of sinusoidal traveling waves. More precisely, using the Werner formulas¹ a standing wave can be written as

$$y_n(x, t) = \sqrt{\frac{1}{2L}} \{ \cos[k_n(ct - x) + \phi_n] - \cos[k_n(ct + x) + \phi_n] \}. \quad (3.23)$$

Therefore a standing wave is the sum of two sinusoidal waves y^\pm that translate rigidly with velocity c , in the right-going and left-going directions, respectively. This proves the equivalence of the D'Alembert and Fourier solutions.

Note however that normal-mode solutions are more general than traveling-wave solutions: already a simple system like a one-dimensional bar, described by a 4th order PDE, does not admit a solution in terms of traveling waves while its normal modes can be written analytically.

3.3 Delays and oscillations

3.3.1 The Karplus-Strong algorithm

This section reviews a sound synthesis algorithm which is relevant from many viewpoints. First, the Karplus-Strong (KS hereafter) algorithm is a famous one and deserves to be studied. Second, it contains many of the basic elements that are needed to provide a clear picture of what waveguide modeling is

¹ $2 \sin \alpha \sin \beta = \cos(\alpha - \beta) - \cos(\alpha + \beta)$.



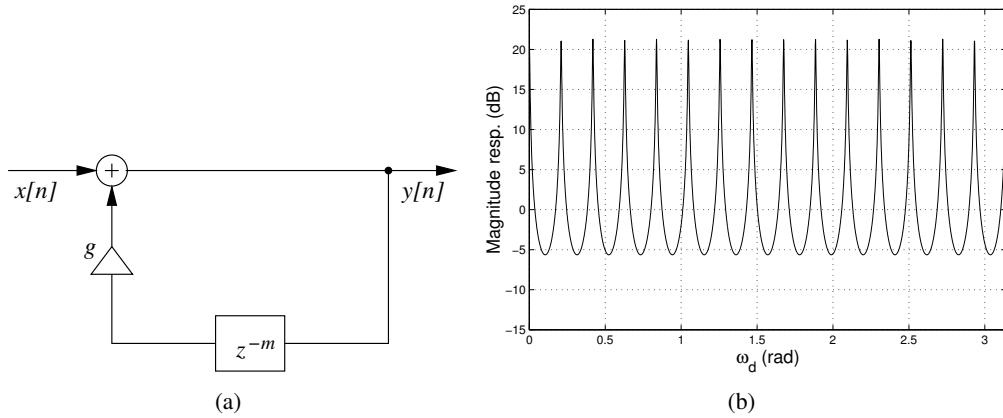


Figure 3.4: A comb filter; (a) block scheme and (b) magnitude response.

all about, and yet it is structurally simple enough to be discussed in a limited amount of pages. Finally, from a historical perspective it can be regarded as the first prototype of a waveguide approach to string modeling: it is true that the original formulation of the algorithm did not contain any physical interpretation. What is unquestionable, however, is that the KS algorithm is structurally identical to the simplest waveguide models that we are going to examine in the next sections.

3.3.1.1 The comb filter

The basic computational structure underlying the KS algorithm is the *IIR comb filter*, which is represented by the following difference equation (and transfer function):

$$y[n] = x[n] + gy[n-m], \quad \Rightarrow \quad H(z) = \frac{1}{1 - gz^{-m}}, \quad (3.24)$$

with $g \in \mathbb{R}$. The block structure of the filter is depicted in Fig. 3.4(a). The poles of $H(z)$ are found from $z^m = g$. Therefore the filter has m poles $p_l = \sqrt[m]{g} e^{j2l\pi/m}$ for $l = 0, \dots, m-1$, equally spaced around the circle of radius $\sqrt[m]{g}$. In order for the filter to be stable, the condition $|g| < 1$ must be satisfied.

The corresponding magnitude response is plotted in Fig. 3.4(b). Each pole p_l produces a peak in the magnitude response. We can apply the analysis seen in Chapter *Sound modeling: signal based approaches* for the second-order resonating filter in order to understand the relation between the poles p_l and the peaks in the response: in general, as g increases and grows closer to 1, each peak becomes higher and the associated bandwidth narrows down. Note also that the filter produces a harmonic spectrum in which frequency peaks are integer multiples of the “fundamental” frequency $f_0 = F_s/m$ Hz.

Figure 3.4(a) already provides us with an intuitive proto-physical interpretation: a disturbance (a wave) in a medium is propagated through that medium, is confined within a certain length, bounces back and forth due to some boundary conditions, has some energy dissipated at each bounce through the coefficient g . Note that if the sign of the wave is inverted at each reflection, the resulting filter spectrum is affected:

$$y[n] = x[n] - gy[n-m] \quad \Rightarrow \quad H(z) = \frac{1}{1 + gz^{-m}} \quad (3.25)$$

In this case the poles are $p_l = \sqrt[m]{g} e^{j(2l+1)\pi/m}$ for $l = 0, \dots, m-1$. This means that the corresponding frequency peaks have all been shifted by an angle π/m with respect to the previous case: now the frequency peaks are *odd* integer multiples of the “fundamental” frequency $f_0 = F_s/(2m)$ Hz. Section 3.4.3



will show that choosing a sign or another corresponds to describing two different boundary conditions (e.g., an open termination versus a closed termination in an acoustical bore).

M-3.35

Write a function that computes the output of the comb filter of Fig. 3.4, given a desired fundamental frequency f_0 and a factor $g < 1$.

M-3.35 Solution

```
function y = ks_simplecomb(f0,g);

global Fs;
m = round(Fs/f0); %length of the delay line
d= dline_init(m); % create a delay-line object

x=((rand(1, ceil(m))*2) - 1)/2;% define random input vector of length m
x=[x zeros(1,round(-3*m/log10(g)) )]; % zero-pad x to hear sound tail
y=zeros(1, length(x)); % initialize output signal

for n = 1:length(x) % audio cycle
    y(n) = x(n) + d.y*g; %read from delay line and update output
    d=dline_compute(d); %update delay line
    d.x = y(n);
end
```

The input signal x is defined in accordance to the KS algorithm specifications (see next section). Zero padding of x is chosen in such a way that the ouput signal has time to decay by 60 dB (see Chapter *Sound in space*). Matlab/Octave are very inefficient at computing long cycles, but we use this approach for coherence with next examples; in particular we have used two auxiliary functions that initialize a delay line structure

```
function f = dline_init(d); %initialize a dline structure of length d
% x is the current input value written into the line
% y=x(n-d) is the current output value read from the line
% in is a buffer containing d past input values

f.x = 0; f.y = 0;
if(floor(d) == d) f.d = d; % ok, d is a valid integer delay
else error('Not a valid delay');
end
f.in = zeros(1, d); % create buffer for past input values
```

and update the state of a delay line structure

```
function f = dline_compute(f);

f.y = f.in(1); % output is the first sample in the buffer
f.in = [f.in(2:length(f.in)), f.x]; % update buffer
```

3.3.1.2 Synthesis of plucked strings

The above observations suggest that the comb structure (3.24) may be employed to synthesize harmonic sounds, such as those produced by an ideal string. However, in order to obtain a complete formulation of the KS algorithm we still have to add some refinements to the structure. Specifically, what it is missing is a mean to control the spectral tilt of the filter magnitude response (i.e. the rate at which the response decays with increasing frequency), and to account for different decay rates for the sound partials.



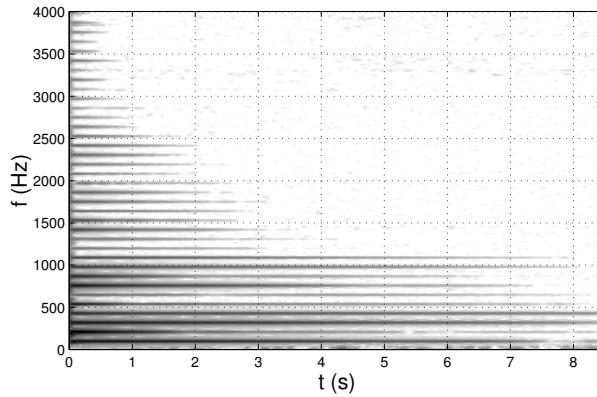


Figure 3.5: . Spectrogram of a plucked A2 guitar string. Note the harmonic structure and the decay rates, which increases with increasing frequency.

In the real world a nylon guitar string is one of the closest relative of an ideal string and exhibits an almost perfectly harmonic spectrum. Figure 3.5 shows the spectrogram of a plucked guitar string: as expected, a harmonic spectrum can be observed. However another relevant feature is that each harmonic partial decays at a different rate, with lower partials surviving longer than higher partials.

On the other hand we have just seen that the IIR comb filter produces a spectrum in which all harmonic peaks have the same magnitude, which means that the associated partials all decay in time at the same rate. In order to simulate a frequency-dependent decay, one can insert a low-pass filter H_{lp} into the feedback loop, as shown in Fig. 3.6(a): we call this structure a *low-pass comb* filter. Intuitively, at each passage the high-frequency components are attenuated more strongly than low-frequencies components. The simplest low-pass filter that can be employed is the first-order FIR already examined in Chapter *Fundamentals of digital audio processing*:

$$y[n] = \frac{1}{2} [x[n] + x[n - 1]] \quad \Rightarrow \quad H_{lp}(z) = \frac{1}{2} [1 + z^{-1}] . \quad (3.26)$$

Figure 3.6(b) shows the frequency response of the low-pass comb structure after the insertion of H_{lp} : as expected higher resonances are less peaked and have larger bandwidths, because now the filter poles have frequency-dependent magnitudes.

However the insertion on a low-pass filter in the structure has also a second effect: it introduces an additional half-sample delay, which can be observed if one looks at the phase response of $H_{lp}(z)$ and is qualitatively explained by the fact that this is filter averages the current sample with the previous one. A consequence of this additional delay is that the fundamental frequency generated by the low-pass comb structure is now $f_0 = F_s/(m + 1/2)$ Hz. Moreover, a closer analysis would also show that the upper partials are not anymore integer multiples of $f_0 = F_s/(m + 1/2)$, due to the insertion of H_{lp} in the loop. These deviations from the harmonic series can also be noticed from the plot in Fig. 3.6(b).

In many cases the deviations introduced by the low-pass filter are very small, especially for the lower partials and for values of g that are close to 1. However they can still be perceivable. As an example, if $F_s = 44.1$ kHz and $m = 100$, then a half sample delay corresponds to a delay in the order of 10^{-5} s: in this case the IIR comb produces a fundamental at $F_s/m = 441$ Hz, while the low-pass comb produces a fundamental at $F_s/(m + 1/2) \sim 439$ Hz.

M-3.36

Find the response of the complete system given in Fig. 3.6 and plot magnitude and phase responses for various values of g and m .



This book is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 license,

©2005-2018 by the authors except for paragraphs labeled as adapted from <reference>

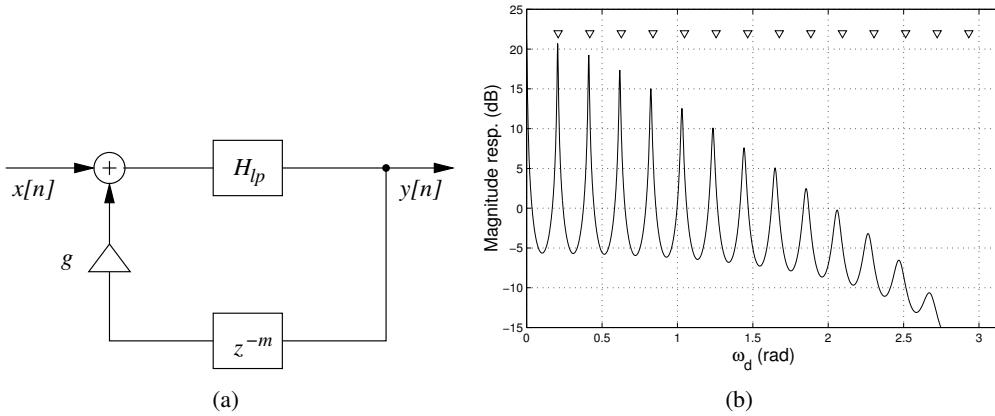


Figure 3.6: Low-pass comb filter obtained through insertion of a low-pass element into the comb structure; (a) block scheme and (b) frequency response (the triangles mark the harmonic series $l\pi/L$, $l \in \mathbb{N}$).

The low-pass comb structure discussed so far is the core of the KS algorithm. However we have not yet discussed what is the input signal to feed to the filter in order to obtain an output sound. Since the impulse response of the filter is the signal that is resemblant of a plucked string sound, an obvious choice is to inject the filter with an impulse. A second possible choice, originally suggested by Karplus and Strong, is to impose a random initial state (m past values of y) to the filter: although this choice has hardly any physical interpretation,² it has the benefit of providing significant initial excitation in the high-frequency region, with a consequent perceptual effect of an initial noisy transient followed by a harmonic steady-state signal.

M-3.37

Write a function that implements the KS algorithm using the low-pass comb of Fig. 3.6, given a desired fundamental frequency f_0 and a factor $g < 1$.

M-3.35 Solution

```
function y = ks_lpcmb(f0, g);

global Fs;
m = round(Fs/f0); %length of the delay line
d= dline_init(m); % create a delay-line object

x=((rand(1, ceil(m))-1)/2;% define random input vector of length m
x=[x zeros(1,round(-3*m/log10(g)) )]; % zero-pad x to hear sound tail
y=zeros(1, length(x)); % initialize output signal
a_past=0; %initialize auxiliary variable (input to lowpass filter)

for n = 1:length(x) % audio cycle
    a = x(n) + d.y*g; %read from delay line and sum to input
    y(n) = 1/2 * (a + a_past); %update output through lowpass filter
    a_past = a; % update auxiliary variable
    d=dline_compute(d); %update delay line
    d.x = y(n);
end
```

²It would be like imposing initial random displacements to points of a string, as we shall see in the next sections.



3.3.2 Fine tuning and fractional delays

Sound pitch (which we assume to coincide with fundamental frequency)³ in the KS algorithm is quantized: adding a unit delay in the comb filter modifies the fundamental period by $1/F_s$, which is a rather gross and perceivable quantization. In order to obtain a finer tuning of the delay loop, we need techniques to simulate *fractional delays*.

An ideal delay of m samples is a filter with transfer function $H_m(z) = z^{-m}$. Therefore its frequency, magnitude, and phase responses are

$$H_m(e^{j\omega_d}) = e^{-j\omega_d m}, \quad |H_m(e^{j\omega_d})| \equiv 1, \quad \arg[H_m(e^{j\omega_d})] = -m\omega_d. \quad (3.27)$$

We want to design a filter with the same characteristics, i.e. flat magnitude response and linear phase response (equivalently, with constant and coincident phase and group delays). However we want the slope of the phase response to be an arbitrary phase delay τ_{ph} , and not limited to integer values m . Moreover, since any real delay τ_{ph} can be written as the sum of an integer delay $\lfloor \tau_{ph} \rfloor$ and a *fractional delay* $0 \leq (\tau_{ph} - \lfloor \tau_{ph} \rfloor) < 1$, without loss of generality we restrict our attention to the design of fractional-delay filters $H_{\tau_{ph}}$ with $0 \leq \tau_{ph} < 1$.

Note that the impulse response of an ideal delay filter is

$$h_{\tau_{ph}}[n] = \frac{1}{2\pi} \int_{-\pi}^{+\pi} e^{-j\omega_d \tau_{ph}} e^{j\omega_d n} d\omega_d = \text{sinc}(n - \tau_{ph}). \quad (3.28)$$

If $\tau_{ph} = m \in \mathbb{N}$ this reduces to $h[n] = \delta[n - m]$. However, if τ_{ph} is not integer then this is a non-causal filter with infinite impulse response, i.e. a non-realizable filter. This remark makes it clear that we will not be able to find exact realizations of fractional-delay filters, and we will have to look for approximations.

3.3.2.1 FIR fractional delay filters

We first examine FIR fractional-delay filters, of the form

$$H_{\tau_{ph}}(z) = \sum_{k=0}^N b_k z^{-k}. \quad (3.29)$$

Starting from this general form, we have to design of an N th order FIR filter approximating a constant magnitude and linear phase frequency response. Several criteria can be adopted to drive this approximation problem. One approach amounts to minimizing some error distance between the FIR filter (3.29) and the ideal fractional-delay filter defined previously. Possibly the most intuitive realization of this approach is the minimization of the least squared (LS) error function, defined as the L^2 norm the error frequency response $E(e^{j\omega_d}) = H_{\tau_{ph}}(e^{j\omega_d}) - e^{-j\tau_{ph}}$ (i.e. E is the difference between the frequency responses of the FIR filter and the ideal fractional-delay filter).

A different approach, that we describe in some more details, amounts to setting the error function $E(e^{j\omega_d})$ and its N derivatives to zero at $\omega_d = 0$:

$$\left. \frac{d^l E}{d\omega_d^l}(e^{j\omega_d}) \right|_{\omega_d=0}, \quad l = 0, \dots, N. \quad (3.30)$$

³In chapter *Auditory based processing* we will see that pitch perception is a complex phenomenon, and that the perceived pitch does not necessarily coincide with the fundamental frequency.



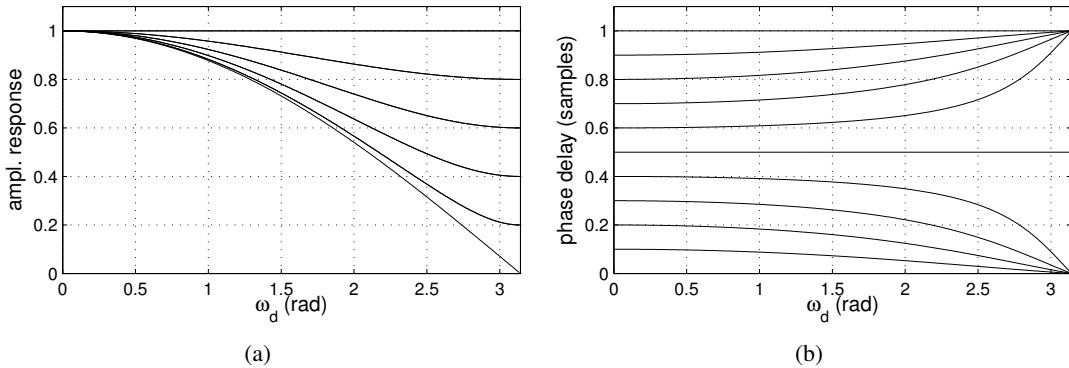


Figure 3.7: Linear interpolation filters ($N = 1$) for $\tau_{ph} = 0, 0.1, \dots, 1$; (a) amplitude response and (b) phase delay.

This is called the *maximally flat design* at $\omega_d = 0$, since it tries to make the error function as flat as possible around the value 0, in the vicinity of zero frequency. Substituting Eq. (3.29) in these latter $N + 1$ equations yields

$$\sum_{k=0}^N k^l b_k = \tau_{ph}^l, \quad l = 0, \dots, N \quad \Leftrightarrow \quad \mathbf{V} \mathbf{b} = \boldsymbol{\tau}, \quad (3.31)$$

where $\mathbf{b} = [b_0, b_1, \dots, b_N]$, $\boldsymbol{\tau} = [1, \tau_{ph}, \dots, \tau_{ph}^N]$, and \mathbf{V} is a Vandermonde matrix with elements $v_{i,j} = (j-1)^{i-1}$. Since \mathbf{V} is non-singular, the system has a unique solution which can be written in explicit form as

$$b_k = \prod_{l \neq k; l=0}^N \frac{\tau_{ph} - l}{k - l}, \quad k = 0, \dots, N. \quad (3.32)$$

It is interesting to notice that the FIR filter coefficients obtained by this method are equal to those of the *Lagrange interpolation* formula for equally spaced abscissas. In other words, the FIR filter determined by these coefficients estimates the value $x[n - \tau_{ph}]$ by interpolating a polynomial of order N over the $N + 1$ values $x[n], x[n - 1], \dots, x[n - N]$. This leads to Lagrange interpolation.⁴ For $N = 1$ one obtains simple linear interpolation, $b_0 = 1 - \tau_{ph}$, $b_1 = \tau_{ph}$. For the case $\tau_{ph} = 1/2$ we reobtain the first-order FIR low-pass filter.

Plots for $N = 1$ and different values of τ_{ph} are shown in Fig. 3.7. The phase delay remains reasonably constant up to high frequency values (and is exactly constant in the cases $\tau_{ph} = 0, 1/2, 1$). Note however that the magnitude response has always a low-pass character. This is a drawback of these FIR filters: high frequencies are attenuated due to non flat magnitude response. Using higher orders N allows to keep the magnitude response close to unity and a phase response close to linear in a wider frequency band. Of course, this is paid in terms of computational complexity.

M-3.38

Implement a fractional delay line using Lagrange interpolation.

M-3.38 Solution

⁴We are not interested here in deriving the Lagrange interpolation method, which is reviewed in many textbooks of numerical analysis.



Same approach as before. One function to initialize the line

```
function f = lagrangedline_init(d,N); %uses Nth order lagrange interpolation

f.x = 0; f.y = 0;
f.d = d; % set delay (not necessarily integer)
f.in = zeros(1, floor(d)+2); % create buffer for past input values
f.b=ones(1,N+1); %coefficients of the Lagrange interpolator
tau=d-floor(d); %fractional delay to be simulated
for k=1:length(f.b)
    for l=1:length(f.b)
        if (l~=k); f.b(k)=f.b(k)*(tau-(l-1))/( (k-1)-(l-1)); end
    end
end
```

and one to update the state

```
function f = lagrangedline_compute(f);

f.y = f.b * f.in(1:length(f.b))'; % output is lagrange interpolation of buffer
f.in = [f.in(2:length(f.in)), f.x]; % update buffer
```

These functions can be tested in the KS algorithm (examples M-3.35 and M-3.37) in place of the integer delay lines.

3.3.2.2 All-pass fractional delay filters

We now examine IIR fractional-delay filters, of the form

$$H_{\tau_{ph}}(z) = \frac{z^{-N} A(z)}{A(z^{-1})} = \frac{a_n + a_{n-1}z^{-1} + \dots + a_1z^{-(N-1)} + z^{-N}}{1 + a_1z^{-1} + \dots + a_{N-1}z^{-(N-1)} + a_nz^{-N}}. \quad (3.33)$$

This is not the transfer function of a generic IIR filter. It represents the transfer function of a N th order *all-pass* filter. According to the definition already given in Chapter *Fundamentals of digital audio processing*, an all-pass filter is a filter with a perfectly flat magnitude response. The filter in the above equation satisfies the property $|H_{\tau_{ph}}(e^{j\omega_d})| \equiv 1$ by construction: this property can be proved by noting that, since the numerator polynomial is a mirrored version of the denominator polynomial A , the poles of a stable all-pass filter are located inside the unit circle and its zeros are located outside the unit circle with the same angle and with the inverse radius of the corresponding poles.

Since the above IIR filter satisfies by construction one of the two properties of an ideal delay filter (flat magnitude response), we can now focus on the second one (linear phase response). The phase response of an all-pass filter is found to be

$$\arg[H_{\tau_{ph}}(e^{j\omega_d})] = N\omega_d + 2 \arg\left[\frac{1}{A(e^{-j\omega_d})}\right] = N\omega_d + 2 \arctan\left[\frac{\sum_{k=0}^N a_k \sin(k\omega_d)}{\sum_{k=0}^N a_k \cos(k\omega_d)}\right]. \quad (3.34)$$

Therefore the phase response, the phase delay, and the group delay are all highly non-linear functions of the filter coefficients. This means that one cannot expect as simple design formulas for the all-pass filter coefficients as for FIR filters. Instead, one can almost exclusively find only iterative optimization techniques for minimization of traditional error criteria.

Possibly the only design technique that has a closed-form solution is the *maximally flat group delay* design. Let us start considering an all-pole low-pass filter with transfer function $1/A(z^{-1})$. It has been



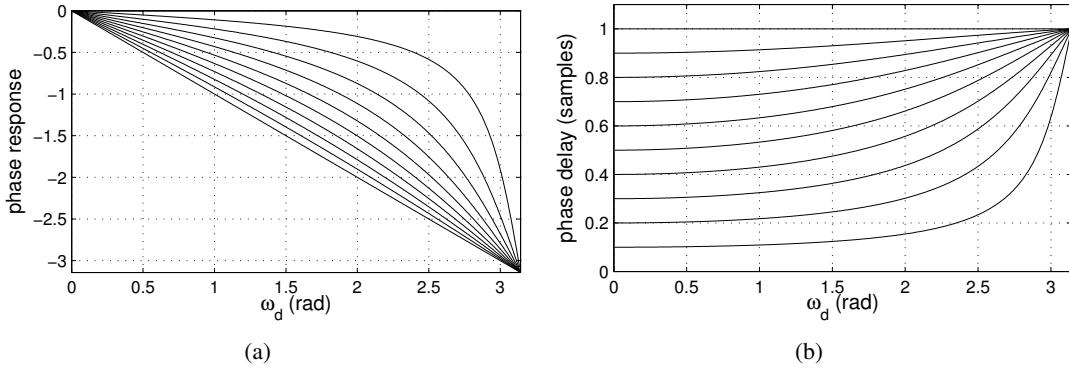


Figure 3.8: First-order Thiran allpass filters for $\tau_{ph} = 0, 0.1, \dots, 1$; (a) phase response and (b) phase delay.

shown that the condition of maximally flat group delay at $\omega_d = 0$ for this filter yields the following analytic solution:

$$a_k = (-1)^k \binom{N}{k} \prod_{l=0}^N \frac{2\tau_{ph} + l}{2\tau_{ph} + k + l}, \quad (3.35)$$

where $\binom{N}{k}$ is the binomial coefficient. When $\tau_{ph} > 0$ then the filter is stable. This result can be applied to our problem using Eq. (3.34): since the fractional phase delay of $H_{\tau_{ph}}$ is twice those of $1/A$, a maximally flat all-pass filter with coefficients

$$a_k = (-1)^k \binom{N}{k} \prod_{l=0}^N \frac{\tau_{ph} + l}{\tau_{ph} + k + l}, \quad (3.36)$$

approximates the ideal delay filter with total delay $N + \tau_{ph}$. This is known as *Thiran all-pass filter* approximation.

As an example let us look at the first-order all-pass filter

$$H_{\tau_{ph}}(z) = \frac{a_1 + z^{-1}}{1 + a_1 z^{-1}}, \quad (3.37)$$

with $a_1 < 1$ for stability. The plots of its phase response and phase delay are shown in Fig. 3.8. In the low-frequency region, the phase response can be approximated as follows:

$$\arg [H_{\tau_{ph}}(e^{j\omega_d})] \sim -\frac{\sin \omega_d}{a_1 + \cos(\omega_d)} + \frac{a_1 \sin \omega_d}{1 + a_1 \cos \omega_d} \sim -\omega_d \frac{1 - a_1}{1 + a_1}, \quad (3.38)$$

i.e. the phase response is approximately linear with phase and group delay approximately equal to $(1 - a_1)/(1 + a_1)$. Therefore given a desired phase delay τ_{ph} one chooses

$$a_1 = \frac{1 - \tau_{ph}}{1 + \tau_{ph}}. \quad (3.39)$$

This corresponds to the Thiran approximation with $N = 1$.

Thiran filters have complementary drawbacks with respect to Lagrange filters: although they provide flat magnitude response, detuning of higher frequencies occurs due to phase non-linearity. In order to have phase response approximately linear in a wider frequency range one has to use higher orders, at the expense of higher complexities.



M-3.39

Implement a fractional delay line using Thiran filters.

M-3.39 Solution

Same approach as before. One function to initialize the line

```
function f = thirandline_init(d, N); %uses a Nth order Thiran filter

f.x = 0; f.y = 0;
f.d = d; % set delay (not necessarily integer)
f.in = zeros(1, floor(d)-N); % create buffer for past input values
% the Thiran filter account for the remaining N+(d-floor(d))
f.state=zeros(1,N); %state of the Thiran filter
f.a = zeros(1, N+1); % coefficients of the Thiran filter
tau = d-floor(d); %fractional delay to be simulated
for k = 0:N
    ak = 1; for l=0:N; ak = ak * (tau+l)/(tau+k+l); end
    f.a(k+1) = (-1)^k * nchoosek(N,k) * ak;
end
```

and one to update the state

```
function f = thirandline_compute(f);

[out,state] = filter(fliplr(f.a), f.a, f.in(1), f.state);
f.state=state;
f.y=out;
f.in = [f.in(2:length(f.in)), f.x];
```

These functions can be tested in the KS algorithm (examples M-3.35 and M-3.37) in place of the integer delay lines.

3.3.2.3 Time-varying delays

3.4 Distributed models: the waveguide approach

This section introduces the basic concepts of waveguide modeling. Discussion is focused on one-dimensional resonators, and no attention is devoted here to higher dimensional waveguide structures.

In their simplest form, waveguide models exploit the existence of an analytical solution to the D'Alembert wave equation, which can be seen as a superposition of traveling waves (rigidly translating waveforms). Such a solution can be simulated in the discrete space-temporal domain using delay lines, and the resulting numerical algorithms are extremely efficient and accurate. Moreover, physical phenomena such as frequency dependent losses and dispersion can be included in the models by incorporating low-pass and all-pass filters in the delay line scheme. Again, careful design of such filters allows for very accurate and relatively low-cost simulations.

3.4.1 Basic waveguide structures

3.4.1.1 Wave variables and wave impedance

So far, only displacement y (for a string) and acoustic pressure p (for a cylindrical bore) have been considered in the wave equation. However, alternative wave variables can be used in strings and acoustical



bore. As an example, the force acting on a string section dx is defined as

$$f(x, t) = -T \frac{\partial y}{\partial x}(x, t) = -T \left[\frac{\partial y^+}{\partial x}(ct - x) + \frac{\partial y^-}{\partial x}(ct + x) \right] = \frac{T}{c} [\dot{y}^+(ct - x) - \dot{y}^-(ct + x)]. \quad (3.40)$$

Therefore, using this equation force waves f^\pm can be defined as $f^\pm := \mp \frac{T}{c} \dot{y}^\pm$. On the other hand, the transversal velocity in the same string is given by

$$v(x, t) = \frac{\partial y}{\partial t}(x, t) = \dot{y}^+(ct - x) + \dot{y}^-(ct + x). \quad (3.41)$$

From this, velocity waves v^\pm are defined as $v^\pm := \dot{y}^\pm$. As we have seen in Sec. 3.2.1, the force-velocity variable pair represent the mechanical Kirchhoff variables, in analogy with voltage and current in electrical systems. From the previous equations it immediately follows that

$$f^\pm(ct \mp x) = \pm Z_0 v^\pm(ct \mp x), \quad \text{with} \quad Z_0 = T/c = \sqrt{T\mu}. \quad (3.42)$$

The quantity Z_0 takes the name of *wave* (or *characteristic*) *impedance* of the string, and its reciprocal $\Gamma_0 = Z_0^{-1}$ is termed *wave admittance*. Note that using Z_0 both the force f and the velocity v can be related to the force waves f^\pm . Namely, the following relations hold:

$$\begin{aligned} f &= f^+ + f^-, & v &= \frac{1}{Z_0} [f^+ - f^-], \\ f^+ &= \frac{f + Z_0 v}{2}, & f^- &= \frac{f - Z_0 v}{2}, \end{aligned} \quad (3.43)$$

that transform the pair (f, v) into the pair (f^+, f^-) , and vice versa.

Wave impedance can be defined also in a cylindrical bore. In this case the Kirchhoff variables are taken to be pressure p and flow u (volume velocity). These can be related through the wave impedance Z_0 : $p^\pm(ct \pm x) = \pm Z_0 u^\pm(ct \pm x)$, where $Z_0 = \rho_{air} c / S$ and S is the constant cross-sectional area of the bore. For conical geometries, the cross-section S is not constant and the definition of Z_0 has to be generalized. The wave impedance is then defined as a function $Z_0(s)$ such that the relations $P^\pm(r, s) = \pm Z_0(s) U^\pm(r, s)$ hold in the Laplace domain. It can be seen that $Z_0(s) = \rho_{air} c / S \cdot [rs/(rs + c)]$.

In summary, Kirchhoff and wave variables in elastic media obeying the D'Alembert equation are related through wave impedance and Eqs. (3.43). This results provide the basis for developing 1-D waveguide structures.

3.4.1.2 Delay lines

Waveguide models exploit the existence of the solution (3.15) to the D'Alembert equation and discretize this solution instead of the differential equation itself. This remark explains to a large extent why waveguide structures are much more efficient than finite difference methods in simulating vibrations of elastic media, at least in the 1-D case.

As a starting example, consider a pressure distribution $p = p^+ + p^-$ inside an ideal lossless cylindrical bore. We want to discretize p both in time and in space. If T_s is the sampling period, a suitable choice for the spatial sampling step is $X_s = cT_s$. Assume for simplicity that the length L is a multiple of the spatial step, $L = mX_s$. Then a discretized version of p is obtained through the variable substitution $x \mapsto mX_s$ and $t \mapsto nT_s$ (with $m, n \in \mathbb{N}$), and leads to

$$p(mX_s, nT_s) = p^+(ncT_s - mX_s) + p^-(ncT_s + mX_s) = p^+((n - m)cT_s) + p^-((n + m)cT_s).$$



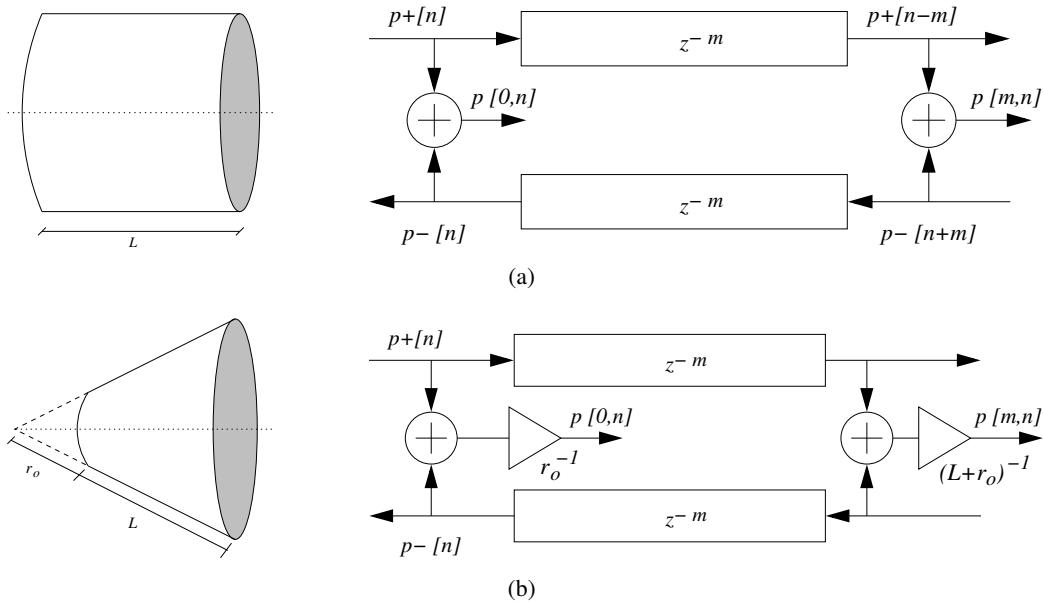


Figure 3.9: Lossless waveguide sections with observation points at position $x = 0$ and $x = mX_s = L$; (a) cylindrical section; (b) conical section.

Removing the constant sampling steps yields

$$p[m, n] = p^+[n - m] + p^-[n + m]. \quad (3.44)$$

The term $p^+[n - m]$ in Eq. (3.44) can be thought of as the output from a digital delay line of length m , whose input is $p^+[n]$. Analogously, the term $p^-[n + m]$ can be thought of as the input of a digital delay line with the same length, whose output is $p^-[n]$. This remark leads to the definition of a *waveguide section* as a bidirectional delay line, as depicted in Fig. 3.9(a). The horizontal direction of this structure has a straightforward physical interpretation: it corresponds to the position x along the axis of the cylindrical bore. In the example depicted in Fig. 3.9(a), two “observation points” have been chosen at $x = 0$ and $x = mX_s = L$. At these points, the pressure signal at time n is reconstructed by summing the corresponding pressure waves p^\pm .

A very similar structure can be outlined for numerically simulating a pressure distribution in an ideal lossless conical bore. In this case, propagation is described by the one-dimensional equation (3.13), whose general solution is given by

$$R(r, t) = \frac{1}{r} [\tilde{R}^+(ct - r) + \tilde{R}^-(ct + r)]. \quad (3.45)$$

The conical waveguide is therefore defined as in Fig. 3.9(b). Observation points can be chosen analogously to the cylindrical case.

At the beginning of this discussion we have assumed for simplicity that $L = mX_s$. However this quantization of the allowed lengths is too coarse for our purposes: with a sampling rate $F_s = 44.1$ kHz and with a wave velocity $c = 347$ m/s (sound velocity in air at 20 °C), the resulting spatial step is $X_s = 7.8 \cdot 10^{-3}$ m. Length differences of this magnitude produce perceivable pitch variations in a wind instrument. One way to overcome this limitation is to include in the structure a fractional-delay filter (see Sec. 3.3.2) that provide fine tuning of the length of a waveguide section.



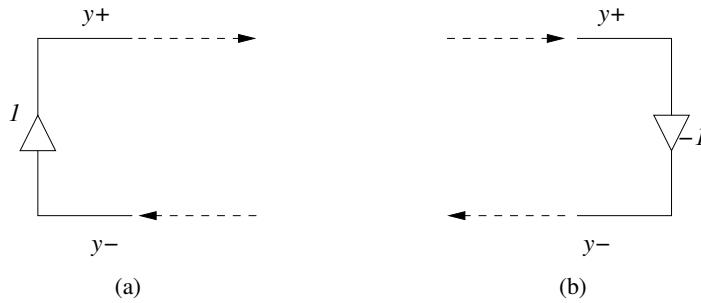


Figure 3.10: Ideal waveguide terminations: (a) positive reflection; (b) negative reflection.

3.4.1.3 Boundary conditions

Looking at Fig. 3.9 we immediately realize that we still one element in order to come out with a computational structure that describes e.g. a string with fixed ends or a cylindrical tube section with open ends: boundary conditions.

In Sec. 3.2.2 we have briefly discussed fixed-end and free-end boundary conditions for the displacement $y(x, t)|_{x=0, L}$ of a vibrating string. These can be immediately turned into *reflection conditions* for both velocity waves and force waves. As an example, a fixed-end condition implies that the velocity is 0 at the boundaries, therefore the reflection conditions $v^+ = -v^-$ applies at both points. By looking at Eq. (3.43), one also see that the 0 velocity condition translates into the reflection condition $f^+ = f^-$ at both points. Therefore wave variables at the boundaries are multiplied by either 1 or -1 (see Fig. 3.10).

More in general, reflection conditions can be derived by formulating boundary conditions for Kirchhoff variables and then using Eq. (3.43) to relate Kirchhoff variables to wave variables. A second relevant example is that of a cylindrical bore of length L , with a closed end at $x = 0$ and an open end at $x = L$. The first condition implies $u = u^+ + u^- = [p^+ - p^-]/Z_0 = 0$ at $x = 0$ (no flow through a closed end), which in turn implies the reflection conditions $u^+ = -u^-$ and $p^+ = p^-$. The second condition implies $p = p^+ + p^- = 0$ at $x = L$ (p matches the atmospheric pressure at the open boundary), which in turn implies the reflection conditions $p^- = -p^+$ and $u^+ = u^-$.

With these concepts in mind we can now go back to Sec. 3.3.1 and reinterpret the IIR comb structure used to construct the KS algorithm. The IIR comb can be viewed as a pair of waveguide sections of length $m/2$ samples in which traveling waves circulate and reflect at the boundaries according to some reflection condition. If the coefficient g has a positive sign, as in Eq. (3.24), the corresponding condition is that of a string fixed at both ends. The signal traveling into the filter can be interpreted either as a velocity wave (two sign inversions at the boundaries) or as a force wave (no sign inversions at the boundaries). As a result a harmonic spectrum is generated that contains all the partials. On the other hand, if the coefficient g has a negative sign, as in Eq. (3.25), the corresponding condition is e.g. that of a cylindrical bore with one open end and one closed end. The signal traveling into the filter can be interpreted either as a flow wave or as a pressure wave (both with one sign inversion at the boundaries). As a result a harmonic spectrum is generated that contains only the odd partials.

3.4.2 Modeling real world phenomena

As already mentioned, the waveguide structures introduced above describe *ideal* systems, i.e. ideally elastic media, where the D'Alembert equation (3.12) or its spherical version (3.13) hold. Real systems exhibit more complex behaviors. Two phenomena are particularly relevant for sound production: dissipation



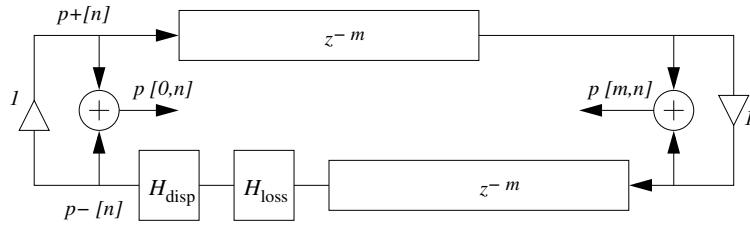


Figure 3.11: Waveguide simulation dissipation and dispersion phenomena through insertion of loss and dispersion filters.

pation and dispersion. Both can be accounted for by adding proper time, space or time-space derivatives of different orders to the ideal wave equation. Correspondingly the basic waveguide structure is modified by inserting appropriate loss and dispersion filters in the loop, as in Fig. 3.11

3.4.2.1 Dissipation

Energy *dissipation* occurs in any real vibrating medium. In an acoustical bore this is due to air viscosity, thermal conduction and wall losses. Dissipation in a string comes from internal losses related to elastic properties of the material, energy transfer through terminations, and friction with air. For clarity, consider the pressure distribution in a cylindrical bore. In the simplest approximation, all of the dissipation phenomena can be incorporated in the D'Alembert equation by including an additional term proportional to the first time derivative. As an example, a first-order approximation of a string with linear density μ , tension T , and dissipation is given by the modified D'Alembert equation

$$\mu \frac{\partial^2 p}{\partial t^2}(x, t) = T \frac{\partial^2 p}{\partial x^2}(x, t) - d_1 \frac{\partial p}{\partial t}(x, t). \quad (3.46)$$

In the limit of small d_1 , Eq. (3.46) still admits a traveling wave solution, which can be digitized with the same procedure described in the ideal case:

$$\begin{aligned} p(x, t) &= e^{-\frac{d_1 x}{2c}} p^+(ct - x) + e^{\frac{d_1 x}{2c}} p^-(ct + x), \quad \text{then} \\ p[m, n] &= g^m p^+[n - m] + g^{-m} p^-[n + m], \quad \text{with} \quad g = e^{-\frac{d_1 T_s}{2}} < 1. \end{aligned} \quad (3.47)$$

Thus the traveling waves are exponentially damped along the propagation direction, and this phenomenon can be incorporated in the waveguide structure. In many real-world phenomena, however, losses increase with frequency. As an example, the dissipative force exerted by the air on a moving string section is, to a first approximation, directly proportional to the frequency of oscillation. Similar remarks apply to the effects of internal material losses. A better approximation of dissipation phenomena in a string is provided by the equation

$$\mu \frac{\partial^2 p}{\partial t^2}(x, t) = T \frac{\partial^2 p}{\partial x^2}(x, t) - d_1 \frac{\partial p}{\partial t}(x, t) + d_2 \frac{\partial^3 p}{\partial t \partial^2 x}(x, t), \quad (3.48)$$

where d_1 introduces frequency-independent dissipation and d_2 introduces frequency-dependent dissipation. This frequency dependence can be accounted for by substituting the constant factor g with a loss filter, which will have a low-pass characteristics. This is shown in Fig. 3.11, where losses have been consolidated, or *lumped*, in a single loss filter $H_{\text{loss}}(z)$ cascaded to the delay line. This filter summarizes the distributed losses occurring in the spatial interval $[0, 2mX_s]$.



With these concepts in mind we can go back again to Sec. 3.3.1 and reinterpret the the comb structures. In the simple IIR comb filter, the coefficient $g < 1$ plays the role of the loss factor g^m , and accordingly introduces equal decay times to all partials. In the low-pass comb filter, the low-pass transfer function H_{lp} plays the role of the loss filter $H_{\text{loss}}(z)$, and accordingly introduces frequency-dependent decay times to the partials.

3.4.2.2 Loss filter design

There are many techniques for designing a loss filter H_{loss} to fit a real object. In this section we outline a relatively simple approach to fit a lossy waveguide model to a real string sound.

First the sound of the target string has to be recorded and analyzed. This can be done using e.g. the sinusoidal peak detection/continuation algorithms discussed in Chapter *Sound modeling: signal based approaches*. As a result from the analysis stage, the frequencies f_k and the decay times τ_k ($k = 1, \dots, N$) of the first N partials can be estimated. In particular τ_k is defined as the time required by the amplitude of the k th partial to decay by $1/e$ with respect to its initial amplitude. A robust way of calculating the τ_k 's is fitting a line by linear regression on the logarithm of the amplitude envelopes derived from the peak continuation algorithm.

The estimated parameters f_k, τ_k specify the magnitude of H_{loss} over a set of N points:

$$\left| H_{\text{loss}} \left(e^{j \frac{2\pi f_k}{F_s}} \right) \right| = e^{-\frac{k}{f_k \tau_k}}, \quad k = 1, \dots, N. \quad (3.49)$$

Given this magnitude specification, a common technique to design H_{loss} is through minimization of the squared error $\sum_{k=1}^N (H_{\text{loss}}(e^{j 2\pi f_k / F_s}) - e^{-k/f_k \tau_k})^2$. However one problem with these techniques is that one may find a filter whose magnitude exceeds unity, which would result in an unstable waveguide structure. Moreover, in order to avoid frequency dependent delay, $H_{\text{loss}}(z)$ should be ideally a linear-phase filter (and the length of the delay line should be reduced correspondingly, in order to obtain the desired overall delay).

A more straightforward design approach amount to choose a first order IIR low-pass filter:

$$H_{\text{loss}}(z) = g \frac{1 + \alpha}{1 + \alpha z^{-1}}, \quad (3.50)$$

with $-1 < \alpha < 0$ and $g < 1$. One can show that in this case the approximate analytical formulas for the decay times are

$$\frac{1}{\tau_k} \simeq a + b \left(\frac{2\pi f_k}{F_s} \right)^2, \quad \text{with } a = f_0(1-g), \quad b = -f_0 \frac{\alpha}{2(\alpha+1)}^2, \quad (3.51)$$

and where f_0 is the fundamental frequency. Therefore the decay rate $1/\tau_k$ is a second-order polynomial of f_k with even order terms. Consequently a and b can be straightforwardly determined by polynomial regression from the prescribed decay times, and finally g and α are computed from a and b via the inverse of Eqs. (3.51). In most cases, the one-pole loss filter yields good results. Nevertheless, when precise rendering of the partial envelopes is required, higher-order filters have to be used.

M-3.40

Realize a complete loss filter design procedure, to be applied to a guitar sound. Use the spectral analysis tools to estimate the decay times of the guitar string partials. Use Eq. (3.51) to design the filter (3.50).



3.4.2.3 Dispersion

A second important phenomenon in natural wave propagation is *dispersion*. In a string, dispersion is introduced by string stiffness, i.e. the phenomenon by which a string opposes resistance to bending. Such a shearing force can be modeled as a fourth spatial derivative, which is introduced as an additional term in the D'Alembert equation:

$$\mu \frac{\partial^2 p}{\partial t^2}(x, t) = T \frac{\partial^2 p}{\partial x^2}(x, t) - D \frac{\partial^4 p}{\partial^4 x}(x, t), \quad (3.52)$$

where the dispersive correction term D is usually termed “bending stiffness” of the string, and is proportional to the string Young’s modulus. If D is sufficiently small, its first-order effect is to increase the wave propagation speed with frequency:

$$c(\omega) = c_0 \left(1 + \frac{D\omega^2}{2Tc_0^2} \right), \quad (3.53)$$

where $c_0 = \sqrt{T/\mu}$ is now the wave propagation speed in the absence of dispersion. Equation (3.53) states that a traveling wave is no longer a rigid shape that translates at constant speed. Instead, frequencies “disperse” as they propagate with different velocities.⁵ As a consequence, the frequencies f_k of the allowed partials are not harmonic, instead they are stretched onto an inharmonic series according to the equation

$$f_k = k f_0 I_k, \quad \text{where } I_k \approx \sqrt{1 + Bk^2}, \quad (3.54)$$

and where $B = \pi^2 D / TL^2$. The quantity I_k is usually termed *index of inharmonicity*. Dispersion is particularly important in piano strings, where the lower tones exhibit significant inharmonicity.

Having a non-uniform wave velocity $c(\omega)$ implies that it is not possible to define a sampling step as $X_s = c_0 T_s$. Instead, it can be said that a component with frequency $f = \omega/(2\pi)$ travels a distance $c_0 T_s$ in the time interval $c_0 T_s / c(\omega)$. As a consequence, each unitary delay z^{-1} in the waveguide structure has to be substituted with an all-pass dispersion filter with unitary magnitude response and a non-linear phase response approximates the frequency-dependent phase delay $c_0 T_s / c(\omega)$.

Similarly to dissipative low-pass filters, these all-pass delays can be *lumped* into a single product filter. Moreover, the linear and non-linear parts of the phase response can be treated separately. In conclusion the dispersion filter that substitutes $2m$ unitary delays can be written as $z^{-2m} H_{\text{disp}}(z)$, where z^{-2m} accounts for the linear part of the phase response and the all-pass filter $H_{\text{disp}}(z)$ approximates the non-linear part. The resulting dispersive waveguide structure is then as in Fig. 3.11.

M-3.41

Implement the waveguide structure of Fig. 3.11, including the loss filter (3.50) and an all-pass filter to simulate dispersion.

3.4.2.4 Dispersion filter design

Similarly to the discussion on loss filter design, the effects of dispersion in a real sound can be estimated from analysis using e.g. the sinusoidal peak detection/continuation algorithms discussed in Chapter *Sound modeling: signal based approaches*. The estimated series of partial frequencies f_k provide an indication of the degree of inharmonicity in the sound, and thus of dispersion.

⁵Dispersion can be sometimes experienced when hiking on the mountains, by imparting an impulse on a long metallic cable such as that of a cableway: after some seconds the impulse will bounce back and one will feel that it has “unraveled” into a smoother step with high-frequency ripples running out ahead.



In this section we outline one possible approach to the dispersion filter design. The total phase delay over a waveguide loop of length $2m$, with loss and dispersion filters is

$$\tau_{\text{ph}}(f_k) = \frac{kF_s}{f_k} = 2m + \tau_{\text{loss}}(f_k) + \tau_{\text{disp}}(f_k). \quad (3.55)$$

With everything else known, this equation provide a phase delay specification for the dispersion filter:

$$\tau_{\text{disp}}(f_k) = \frac{kF_s}{f_k} - 2m - \tau_{\text{loss}}(f_k). \quad (3.56)$$

Given L estimated partial frequencies $\{f_k\}_{k=1,\dots,L}$, one can then design an all-pass filter of order $N < L$ as follows. First, for each partial compute the quantities

$$\beta_k = -\frac{1}{2} [\tau_{\text{disp}}(f_k) - 2N\pi f_k], \quad k = 1, \dots, L. \quad (3.57)$$

Then, filter coefficients are computed by solving the system

$$\sum_{j=1}^N a_j \sin(\beta_k + 2j\pi f_k) = \sin \beta_k, \quad k = 1, \dots, L. \quad (3.58)$$

This is an overdetermined system. It can be solved with a LS error criterion.

Note that this design approach is not based on the fitting of the relative positioning of the partials, but on the absolute values of the f_k 's. Therefore the resulting all-pass filter accounts both for the simulation of dispersion and for the fine-tuning of the string (fractional delay).

M-3.42

Realize a complete dispersion filter design procedure, to be applied to a piano sound. Use the spectral analysis tools to estimate the frequencies of the piano partials. Use Eq. (3.58) to design the all-pass filter.

3.4.3 Junctions and networks

The last section has introduced the main concepts of waveguide modeling for a signal propagating in a *uniform* medium. When discontinuities are encountered, the wave impedance changes and signal *scattering* occurs, i.e. a traveling wave is partially reflected and partially transmitted.

Examples of non-uniform media are a cylindrical bore where the cross-sectional area changes abruptly, or a string where the value of the linear mass density jumps changes discontinuously. In order to model these discontinuities, appropriate junctions have to be developed, that connect two (or more) waveguide sections. The boundary reflection conditions that we have examined at the end of Sec. 3.4.1 can be regarded as special cases of junctions, as discussed in the following paragraphs.

3.4.3.1 The Kelly-Lochbaum junction

Consider two cylindrical bores, with cross-sectional areas $S_{1,2}$ and wave admittances $\Gamma_{1,2} = Z_{1,2}^{-1} = S_{1,2}/\rho_{air}c$, connected to each other. Analysis of this problem leads to the derivation of the well known *Kelly-Lochbaum* junction.

The derivation is based on imposing appropriate physical constraints on the Kirchhoff variables p, u at the junction. Specifically, continuity requires that pressures $p_{1,2}$ have the same value p_J at the junction. Moreover, the flows $u_{1,2}$ from the two sides must sum to zero (simply said, the air entering one side of

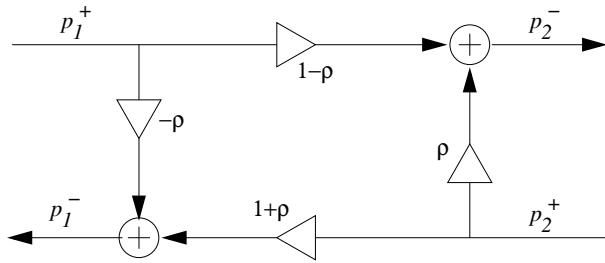


Figure 3.12: Kelly-Lochbaum junction for two cylindrical bores with different areas.

the junction and the air coming out from the other side must be the same). These two requirements lead to the following conditions at the junction:

$$u_1 + u_2 = 0, \quad p_1 = p_2 = p_J. \quad (3.59)$$

Using the Kirchhoff analogy $p \leftrightarrow v$ (voltage) and $u \leftrightarrow i$ (current), Eqs. (3.59) can be regarded as describing a parallel junction. If pressure wave variables are introduced as in Eq. (3.43) (with p^+ and p^- denoting incoming and outgoing waves, respectively), and the junction pressure p_J is used, then the relation $p_l^- = p_J - p_l^+$ (for $l = 1, 2$) holds. Substitution in the first of Eqs. (3.59) yields

$$\begin{aligned} 0 &= (u_1^+ + u_1^-) + (u_2^+ + u_2^-) = \Gamma_1(p_1^+ - p_1^-) + \Gamma_2(p_2^+ - p_2^-) = \\ &= \Gamma_1(2p_1^+ - p_J) + \Gamma_2(2p_2^+ - p_J). \end{aligned} \quad (3.60)$$

From this, the junction pressure p_J can be expressed in terms of the incoming pressure waves $p_{1,2}^+$ as

$$p_J = 2 \frac{\Gamma_1 p_1^+ + \Gamma_2 p_2^+}{\Gamma_1 + \Gamma_2}. \quad (3.61)$$

Using this latter expression, the outgoing pressure waves $p_{1,2}^-$ can be written as

$$\begin{aligned} p_1^- &= p_J - p_1^+ = -\frac{\Gamma_2 - \Gamma_1}{\Gamma_2 + \Gamma_1} p_1^+ + \frac{2\Gamma_2}{\Gamma_2 + \Gamma_1} p_2^+, \\ p_2^- &= p_J - p_2^+ = +\frac{2\Gamma_1}{\Gamma_2 + \Gamma_1} p_1^+ + \frac{\Gamma_2 - \Gamma_1}{\Gamma_2 + \Gamma_1} p_2^+. \end{aligned} \quad (3.62)$$

And finally

$$\begin{aligned} p_1^- &= -\rho p_1^+ + (1 + \rho) p_2^+, \\ p_2^- &= (1 - \rho) p_1^+ + \rho p_2^+, \end{aligned} \quad \text{with } \rho \triangleq \frac{\Gamma_2 - \Gamma_1}{\Gamma_2 + \Gamma_1}, \quad (3.63)$$

These equations describe the Kelly-Lochbaum junction. The quantity ρ is called the *reflection coefficient* of the junction. A scattering diagram is depicted in Fig. 3.12.

This junction has been extensively used in so-called “multitube lossless models” of the vocal tract. These are articulatory models where the vocal tract shape is approximated as a series of concatenated cylindrical sections. Pressure wave propagation in each section is then described using digital waveguides, and interconnections are treated as Kelly-Lochbaum junctions. However this very same junction can be used to describe not only acoustic, but also mechanical structures. As an example, consider two strings with different densities, connected at one point: this can be thought of as a series junction, since the physical constraints impose that velocity (i.e., “current”) has to be the same on the left and right



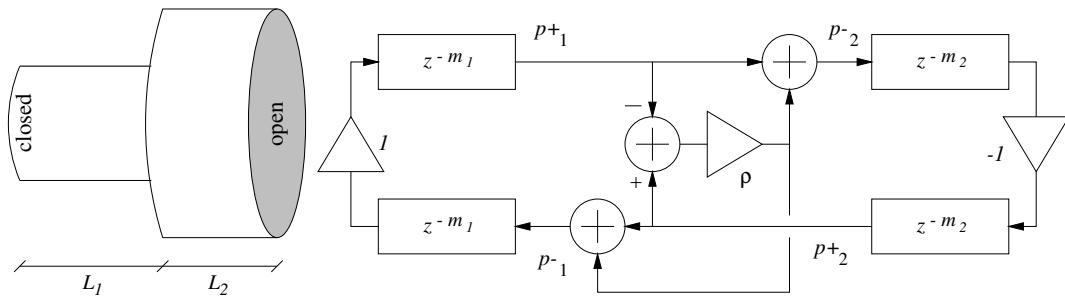


Figure 3.13: Example of use of the Kelly-Lochbaum junction: (a) a parallel junction of two cylindrical bores; (b) realization with two waveguide sections and a Kelly-Lochbaum junction.

sides, and the sum of forces (i.e., “voltages”) from the two sides must be zero. Analogously to the above analysis, a series Kelly-Lochbaum junction can be derived in this case.

Terminations of a waveguide model are an interesting particular case of junctions. Consider an ideal cylindrical bore, closed at one end: this boundary condition corresponds to an infinite impedance $Z_2 = \infty$ (i.e., $S_2 = 0$), and thus to a reflection coefficient $\rho = -1$. In other words, complete reflection occurs and the relation $p_1^-(0, t) = p_1^+(0, t)$ holds. Similarly, an ideally open end can be seen to correspond to $Z_2 = 0$ (i.e., $S_2 = \infty$), and thus to $\rho = 1$: this is a second case where complete reflection occurs, namely the relation $p_1^-(0, t) = -p_1^+(0, t)$ holds. These reflection conditions are identical to those derived in Sec. 3.4.1 (similar considerations hold for string terminations).

Figure 3.13 shows an example where different junctions have been used and combined into a waveguide model. Note that in this example the scattering junction between the two cylindrical sections is not in the original Kelly-Lochbaum form; instead, a *one-multiply scattering junction* is used, which allows more efficient implementation of Eqs. (3.63). Open- and closed-tube terminations are modeled according to the above remarks.

M-3.43

Implement the waveguide structure of Fig. 3.13. Add a loss filter (3.50) to each WG section.

3.4.3.2 N-dimensional and loaded junctions

The result expressed in Eq. (3.63) can be readily extended to higher dimensions. Consider a parallel junction of N acoustic bores. In this case a *scattering matrix* can be found, and Eq. (3.63) is generalized to

$$\mathbf{p}^- = \mathbf{A} \cdot \mathbf{p}^+, \quad (3.64)$$

where \mathbf{p}^\pm are n -dimensional vectors whose elements are the incoming and outgoing pressure waves in the n bores. The physical constraints expressed in Eq. (3.59) are also generalized as

$$\begin{aligned} p_1 &= p_2 = \dots = p_N = p_J, \\ u_1 + u_2 + \dots + u_N &= 0. \end{aligned} \quad (3.65)$$



Calculations analogous to those outlined for the Kelly-Lochbaum junction yield

$$\mathbf{A} = \begin{bmatrix} \frac{2\Gamma_1}{\Gamma_J} - 1, & \frac{2\Gamma_2}{\Gamma_J}, & \dots & \frac{2\Gamma_N}{\Gamma_J} \\ \frac{2\Gamma_1}{\Gamma_J}, & \frac{2\Gamma_2}{\Gamma_J} - 1, & \dots & \frac{2\Gamma_N}{\Gamma_J} \\ \vdots & & \ddots & \vdots \\ \frac{2\Gamma_1}{\Gamma_J}, & \frac{2\Gamma_2}{\Gamma_J}, & \dots & \frac{2\Gamma_N}{\Gamma_J} - 1 \end{bmatrix}, \quad \text{where } \Gamma_J = \sum_{l=1}^N \Gamma_l. \quad (3.66)$$

As an example, a 3-dimensional junction can be used to model an acoustic hole in a wind instrument: in this case, two waveguide sections represents the two sides of the acoustic bore with respect to the hole, and the third one represents the hole itself. Note also that when $N = 2$ Eq. (3.64) reduces to the Kelly-Lochbaum equations.

A second relevant extension of the Kelly-Lochbaum junction is the *loaded junction*, in which an external signal is injected into the system. A simple example is that of a string that is excited (e.g. hammered) at a given point. For continuity, the velocity of the string in this contact point will be the same at both sides. Moreover, during the contact this velocity will be equal to the velocity of the hammer. Finally, the sum of the forces at the contact point equals the hammer force. The following equations of continuity are then derived:

$$v_1 = v_2 = v_J, \quad f_1 + f_2 + f_J = 0. \quad (3.67)$$

With the Kirchhoff analogies this is a series junction with an external load (the “currents” at the junction are the same, and the potentials at the junction sum to the driving potential). Then

$$\begin{aligned} f_J &= -f_1 - f_2 = \dots = -2Z_0(v_1^+ + v_1^- - v_J), \\ \Rightarrow v_J &= v_1^+ + v_2^+ - \frac{1}{2Z_0}f_J. \end{aligned} \quad (3.68)$$

This yields the scattering equations for the loaded junction:

$$\begin{aligned} v_1^-[n] &= v_J[n] - v_1^+[n] = v_2^+[n] + \frac{1}{2Z_0}f_J[n], \\ v_2^-[n] &= v_J[n] - v_2^+[n] = v_1^+[n] + \frac{1}{2Z_0}f_J[n]. \end{aligned} \quad (3.69)$$

The corresponding computational structure is shown in Fig. 3.14. This structure may be further extended to the case of N -dimensional parallel or series loaded junctions.

M-3.44

Implement the waveguide structure of Fig. 3.14. Add a loss filter (3.50) to each WG section.

3.4.3.3 Non-cylindrical geometries

A final remark is concerned with junctions of conical elements. Generalizing the cylindrical case is not straightforward, since the derivation of Kelly-Lochbaum equations is based on the implicit assumption of plane wave propagation. This assumption permits imposition of the constraints (3.59) on a flat scattering boundary, which is a wavefront for both p_1 and p_2 . But wavefronts in conical sections are spherical and this circumstance makes it impossible to define a unique surface on which boundary conditions can be applied: Fig. 3.15(a) shows that there is a region between the two spherical wavefronts which is within neither conical segment. This ambiguity in the definition of the scattering boundary is usually overcome



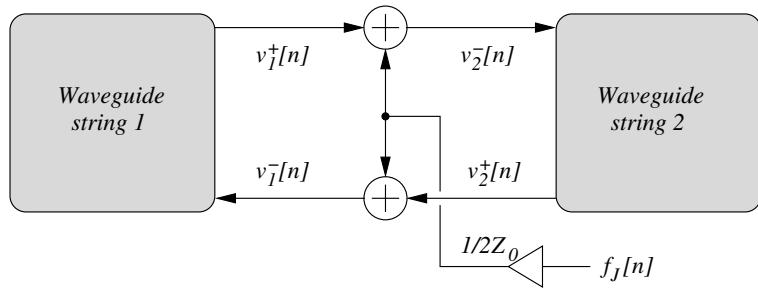


Figure 3.14: Example of a loaded junction: a waveguide structure for a string excited by an external force signal $f_J[n]$ (e.g. a hammer).

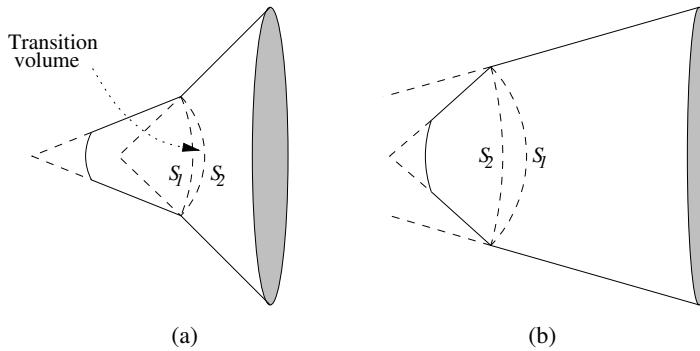


Figure 3.15: Boundary regions for (a) non-convex and (b) convex conical junctions.

by assuming that the transition volume is small and thus pressure is constant inside the volume. Under this assumption, continuity conditions analogous to (3.59) are imposed and the reflection coefficient ρ is generalized to a first order filter $R(s)$.

However, a second and more serious problem arises when one looks at the nature of $R(s)$. This filter turns out to be unstable (non-causal growing exponential) in the case of the convex configuration depicted in Fig. 3.15(b). While this circumstance is physically consistent (in the continuous-time domain the scattered waves can grow exponentially only for a limited time because they are cancelled out by subsequent multiple reflections), in a numerical simulation the system can turn out unstable, due to the approximations introduced by the discretization process and to round-off errors introduced by finite-precision.

3.5 Lumped models and the modal approach

Lumped modeling approaches can be applied in variety of contexts where the physical system under exam can be represented with ideal lumped elements. These include *electrical circuits*, with linear elements like capacities, resistances, and inductances, connected in series and parallel; *mechanical systems* viewed as ideal point masses connected through springs and dampers, representing mechanical resonators; *acoustic systems* viewed as networks of linear acoustic elements like bores, cavities, and acoustic holes (like the Helmholtz resonator examined previously).

Lumped models are particularly suited for describing systems whose spatial dimensions are small compared to acoustic wavelengths. As an example, pressure-controlled valves, such as single, double

or lip reeds, can be conveniently described using the lumped modeling paradigm. Although these systems are quite complicated, due to their limited spatial extensions they can be modeled using lumped elements, and it is widely accepted that such a simplified description captures the basic behavior of pressure controlled valves. Similar remarks hold for hammers and mallets: during collision, they are deformed and subject to internal losses and non-linear restoring forces. However, interactions with strings and bars have been modeled and efficiently implemented in sound synthesis algorithms by assuming the hammer/mallet to be a lumped mass.

3.5.1 Numerical methods

Unlike waveguide structures, lumped models are developed in the continuous-time domain, and are described through sets of ordinary differential equations (ODEs). In order to be implemented as numerical algorithms for sound synthesis, the differential equations have to be discretized in an efficient and effective manner. In most cases, a trade-off has to be found between accuracy of the discretization technique and efficiency of the resulting algorithms.

3.5.1.1 Impulse invariant method

When dealing with linear time-invariant systems, the most elementary technique to turn a continuous-time system into a discrete-time one is sampling its impulse response.

If a continuous-time LTI system is described in terms of Kirchhoff variables, then it is possible to define a transfer function which coincides with the admittance $\Gamma(s)$ of the system. As an example, in a mechanical lumped system this corresponds to defining the input as a driving force and the output as the resulting velocity. The inverse Laplace transform $\gamma(t)$ is the continuous-time impulse response. The linear system can thus be digitized by defining the discrete response as $\gamma_d[n] \triangleq \gamma(nT_s)$, i.e. by sampling $\gamma(t)$. This technique is widely used in the context of digital filter design, and it is usually termed the *Impulse invariant method*.

Assume that the continuous-time system has a rational transfer function $\Gamma(s)$. This can be rewritten using a partial fraction expansion (similarly to what we have done in Chapter *Fundamentals of digital audio processing* for discrete-time systems):

$$\Gamma(s) = \frac{B(s)}{A(s)} = \frac{\sum_{k=0}^M b_k s^{M-k}}{\sum_{k=0}^N a_k s^{N-k}}, \quad \Rightarrow \quad \Gamma(s) = \sum_{k=1}^N \frac{K_k}{s - p_k}, \quad (3.70)$$

where the p_k 's are the poles of the system. By taking the inverse Laplace transform of this latter equation, one can see that the impulse response $\gamma(t)$ is a combination of complex exponentials. This impulse response is then sampled to obtain its digital counterpart:

$$\gamma(t) = \sum_{k=1}^N K_k e^{p_k t}, \quad \Rightarrow \quad \gamma_d[n] \triangleq \gamma(nT_s) = \sum_{k=1}^N K_k (e^{p_k T_s})^n. \quad (3.71)$$

Finally, taking the Z -transform of γ_d yields:

$$\Gamma_d(z) = \sum_{k=1}^N \frac{K_k}{1 - p_{d,k} z^{-1}} = \frac{B_d(z)}{A_d(z)}, \quad \text{with } p_{d,k} = e^{p_k T_s}. \quad (3.72)$$

This equation tells that the transfer function $\Gamma_d(z)$ of the discretized system is still rational, with N poles $p_{d,k}$ uniquely determined by the continuous-time poles p_k .



One quality of the method is that stability is guaranteed at any sampling rate: if the continuous-time system is stable, i.e. $\text{Re}(p_k) < 0$ for all k , then Eq. (3.72) tells that $|p_{d,k}| < 1$ for all k , i.e. the discrete time system is also stable. On the other hand, a drawback of the method is *aliasing*. Since $\gamma_d[n]$ has been obtained by sampling $\gamma(t)$, then the discrete-time response Γ_d is a periodization of Γ :

$$\Gamma_d(e^{j\omega}) = \sum_{k=-\infty}^{+\infty} \Gamma\left(\frac{j\omega}{T_s} + j\frac{2k\pi}{T_s}\right). \quad (3.73)$$

As a consequence, aliasing can occur in Γ_d if the bandwidth of Γ exceeds the Nyquist frequency.

3.5.1.2 Finite differences and mappings “*s-to-z*”

An alternative approach to the discretization of ODEs amounts to replacing time derivatives with *finite differences*, thus turning the differential equations directly into difference equations. Since in the Laplace domain the derivation operator is turned to a multiplication by s , and since in the z -domain the unit delay is turned into a multiplication by z^{-1} , approximating derivatives with finite differences corresponds to finding appropriate *s-to-z mappings*. Let $s = g(z)$ be such a mapping, then if the original continuous-time system is LTI with impulse response $\Gamma(s)$, the discrete-time response is found as $\Gamma_d(z) = \Gamma(g(z))$.⁶

The simplest possible mapping is obtained by replacing the derivative with an incremental ratio. Let $x(t)$ be a generic smooth function of time, then

$$\frac{dx}{dt}(nT_s) = \lim_{h \rightarrow 0+} \frac{x(nT_s) - x(nT_s - h)}{h} \approx \frac{x[n] - x[n-1]}{T_s} \Rightarrow s \approx \frac{1 - z^{-1}}{T_s} \triangleq g_1(z). \quad (3.74)$$

The mapping $g_1(z)$ is known in numerical analysis as the *backward Euler method*. The adjective “backward” is used because the first derivative of x at time n is estimated through the values of x at time n and $n-1$. Higher-order derivatives can be estimated through iterate application of Eq. (3.74). As an example, the second derivative is computed as

$$\frac{d^2x}{dt^2}(nT_s) \approx \frac{1}{T_s} \left[\frac{x[n] - x[n-1]}{T_s} - \frac{x[n-1] - x[n-2]}{T_s} \right] = \frac{x[n] - 2x[n-1] + x[n-2]}{T_s^2}. \quad (3.75)$$

Alternatively, a centered estimate is also often used in combination with the backward Euler method. In this case the second derivative is computed as:

$$\frac{d^2}{dt^2}x(t_n) \approx \frac{x[n+1] - 2x[n] + x[n-1]}{T_s^2}. \quad (3.76)$$

A second, widely used *s-to-z* mapping is provided by the *bilinear transform*. Like the backward Euler method, it can be seen as a finite approximation of the time derivative, but in this case the incremental ratio is assumed to approximate the value of $\dot{x}(t)$ averaged on time instants nT_s and $(n-1)T_s$:

$$\frac{\dot{x}(nT_s) + \dot{x}((n-1)T_s)}{2} \approx \frac{x[n] - x[n-1]}{T_s}, \Rightarrow s \approx 2F_s \frac{1 - z^{-1}}{1 + z^{-1}} \triangleq g_2(z). \quad (3.77)$$

The mapping $g_2(z)$ is known in numerical analysis as the one-step *Adams-Moulton method*.

Both the backward Euler method and the bilinear transform are *implicit* numerical methods. This means that both methods turn a generic first-order differential equation $\dot{x}(t) = f(x(t), t)$ into a difference

⁶Note however that, unlike the impulse invariant method, finite differences do not assume linearity and time invariance of the original system, and are therefore more general methods.



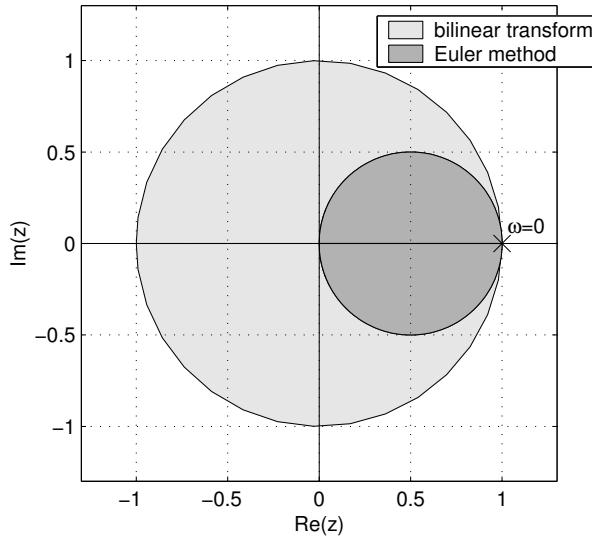


Figure 3.16: Mapping of the vertical axis $s = j\omega$ (solid circle lines) and of the left-half s -plane (shaded regions) using the backward Euler method g_1 and the bilinear transform g_2 .

equation of the form $x[n] = f_d(x[n], x[n-1], n)$, in which $x[n]$ depends implicitly on itself through the function f_d . This is a source of problems for the resulting discrete-time system, since the difference equation is not computable explicitly due to the instantaneous dependence of a variable on itself. Below we discuss briefly this computability problem in the case of linear systems. Note that one advantage of the centered estimate (3.76) is that when it is applied in conjunction with the Euler method to a second-order ODE it leads to an *explicit* difference equation.

3.5.1.3 Accuracy, stability, computability

A comparison between the first estimate in Eq. (3.77) and the first in Eq. (3.74), gives the intuition that the bilinear transform provides a more accurate approximation than the Euler method. A rigorous analysis would show that the order of accuracy of the bilinear transform is two, while that of the backward Euler method is one.

Another way of comparing the two techniques consists in studying how the frequency axis $s = j\omega$ and the left-half plane $\text{Im}(s) < 0$ are mapped by $g_{1,2}$ into the discrete domain. This provides information on both stability and accuracy properties of $g_{1,2}$. As shown in Fig. 3.16, both the methods define one-to-one mappings from $s = j\omega$, onto two circles. Therefore no frequency aliasing is introduced. Second, both the methods are stable, since the left-half s -plane is mapped inside the unit circle by both g_1 and g_2 . However we also see that both mappings introduce *frequency warping*, i.e. the frequency axis is distorted. The bilinear transform g_2 maps the axis $s = j\omega$ exactly onto the unit circle $z = e^{j\omega_d}$, and the mapping between the continuous frequency ω and the digital frequency ω_d can be written analytically:

$$j\omega = \frac{2}{T_s} \frac{1 - e^{-j\omega_d}}{1 + e^{-j\omega_d}} = \frac{2j}{T_s} \tan\left(\frac{\omega_d}{2}\right), \quad \Rightarrow \quad \omega_d = 2 \arctan\left(\frac{\omega T_s}{2}\right). \quad (3.78)$$

At low frequencies ω_d increases almost linearly with ω , while higher frequencies are progressively compressed (warped) and $\omega_d \rightarrow \pm\pi$ as $\omega \rightarrow \pm\infty$. This warping phenomenon is the main drawback of the bilinear transform.



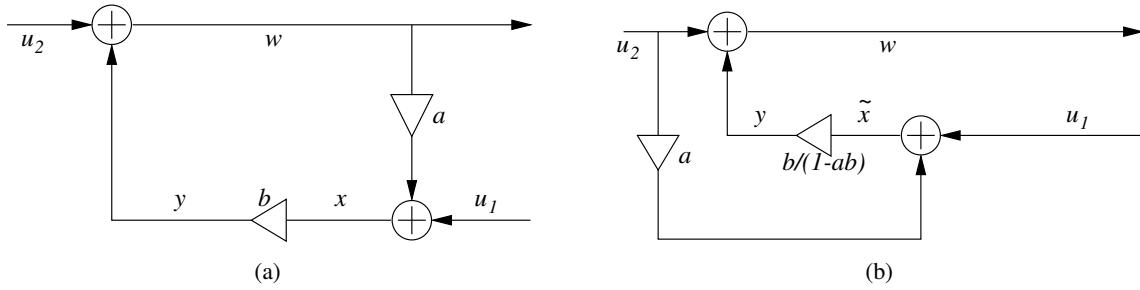


Figure 3.17: A linear discrete-time system; (a) delay-free path, (b) equivalent realization with no delay-free paths.

For the Euler method no analytic mapping can be found from ω to ω_d . The function g_1 “doubly” warps the frequency axis: there is a progressive warping in the direction of increasing frequency (similarly to the bilinear transform), and there is also warping normal to the frequency axis. Figure 3.16 also shows that the poles of the discrete-time system obtained with g_1 are more “squeezed” inside the unit circle than those obtained with g_2 . Furthermore, it can happen that continuous-time poles with positive real-part are turned by g_1 into discrete-time poles with modulus less than unity: in other words g_1 can turn unstable continuous systems into stable discrete systems. This *numerical damping* is a second major drawback of the Euler method.

One more relevant aspect to discuss is the *computability* of the discrete-time systems obtained when discretizing a system of ODEs with either $g_{1,2}$ (or other mappings). As already stated, being these implicit methods the resulting difference equations are implicit. In order to clarify this point, let us consider the simple example depicted in Fig. 3.17(a). This system can be written as

$$\begin{cases} w[n] = \tilde{w}[n] + y[n], & \text{with } \tilde{w} = u_2, \\ x[n] = \tilde{x}[n] + ay[n], & \text{with } \tilde{x} = u_1 + au_2, \\ y[n] = bx[n], & \Rightarrow y[n] = b[u_1[n] + au_2[n] + ay[n]], \end{cases} \quad (3.79)$$

where we have defined tilded variables \tilde{w} and \tilde{x} than only depend on the external inputs $u_{1,2}$, and are therefore known at each time n .

The signals y and x are connected through a *delay-free loop* and the resulting set of difference equations is implicit: in particular the last of Eqs. (3.79) shows that $y[n]$ depends implicitly on itself. It is easy, however, to rearrange the computation in order to solve this problem: the last of Eqs. (3.79) can be inverted, yielding

$$y[n] = \frac{b}{1-ab}[u_1[n] + au_2[n]]. \quad (3.80)$$

This new equation relates y to the computable vector \tilde{x} . Therefore, an equivalent realization of the system is obtained as shown in Fig. 3.17(b). The key point in this example is that the discrete-time system is linear, which allows explicit inversion of the last equation in (3.79).

This simple example is an instance of the so-called *delay-free loop* problem. In the linear case the literature of digital signal processing provides techniques for the restoring computability by rearrangement of the structure.

3.5.1.4 Wave digital filters

The bilinear transform finds application in the theory of Wave Digital Filters (*WDFs*). These structures are digital equivalents of the lumped circuit elements described in Sec. 3.2.1. WDF theory has been developed primarily for electric circuits but can be applied as well to mechanical and acoustic systems using Kirchhoff analogies.

Wave digital filters are constructed in two steps. The first step amounts to converting the continuous-time lumped circuits in wave variables. Here the definition of wave variables is identical to that used for waveguides models (see Eq. (3.43), namely:

$$f^+ = \frac{f + Z_0 v}{2}, \quad f^- = \frac{f - Z_0 v}{2}, \quad (3.81)$$

where the mechanical Kirchhoff variables force f and velocity v have been used for clarity. The only and fundamental difference with Eq. (3.43) is that here Z_0 is a reference impedance that can be given any value and has no direct physical interpretation. The variables f^\pm themselves do not have a clear physical interpretation since in a lumped model they cannot be interpreted as traveling waves. Therefore Eqs. (3.81) have to be regarded as a mere change of coordinates.

Using wave variables, circuit elements can be converted into *one-port elements*. Given one of the elementary lumped elements analyzed in Sec. 3.2.1 (resistance, inductance, capacity) and its associated impedance $Z(s)$, the new variables f^\pm are related to each other through a *reflectance* $R(s)$:

$$F(s) = Z(s)V(s), \quad \Rightarrow \quad F^-(s) = R(s)F^+(s), \quad \text{with} \quad R(s) \triangleq \frac{Z(s) - Z_0}{Z(s) + Z_0}. \quad (3.82)$$

The circuit element can then be visualized as a black box with a port consisting of two terminals, with a port voltage applied across them, and an associated flowing current, as in Fig. 3.18(a). A linear system can then be modeled through series and parallel connections of one-port elements: as an example, Fig. 3.18(b) visualizes a series connection of two ports representing the mechanical system

$$m\ddot{x}(t) + kx(t) = f(t). \quad (3.83)$$

The second step in WDF design is the discretization of $R(s)$. The equivalent wave digital filter is obtained using the bilinear transform as $R(g_2(z))$. Note that since the reference impedance Z_0 can be given any value, this provides an additional degree of freedom in the design. In particular, Z_0 can be chosen such that the WDF has no delay-free paths from input to output, therefore guaranteeing computability when connecting more than one element. As an example, consider the three elementary mechanical impedances $Z_{\text{mass}}(s) = ms$, $Z_{\text{spring}}(s) = k/s$, $Z_{\text{loss}}(s) = r$. For the mass, the reflectance is $R_{\text{mass}}(s) = (ms - Z_0)/(ms + Z_0)$, therefore the equivalent WDF is

$$R_{\text{mass}}(z) = \frac{(2F_s - Z_0/m) - (Z_0/m + 2F_s)z^{-1}}{(2F_s + Z_0/m) - (Z_0/m - 2F_s)z^{-1}}, \quad \Rightarrow R_{\text{mass}}(z) = z^{-1} \quad \text{with} \quad Z_0 = 2F_s m. \quad (3.84)$$

Therefore choosing $Z_0 = 2F_s m$ leads to the interesting result that no delay-free path is present in the corresponding WDF. Similarly, one can prove that $R_{\text{spring}}(z) = z^{-1}$ with $Z_0 = k/2F_s$, and $R_{\text{loss}} = 0$ with $Z_0 = r$.

This brief section has shown that WDFs can be used to digitize lumped element networks using wave variables and adapted impedances in such a way that delay-free computational loops are avoided in the resulting numerical structure. We have shown a single example of a series connection between two elements. The concept of connection is generalized in WDF theory with the concept of *adaptors*, which are N -port elements that model interconnection between arbitrary numbers of elements.



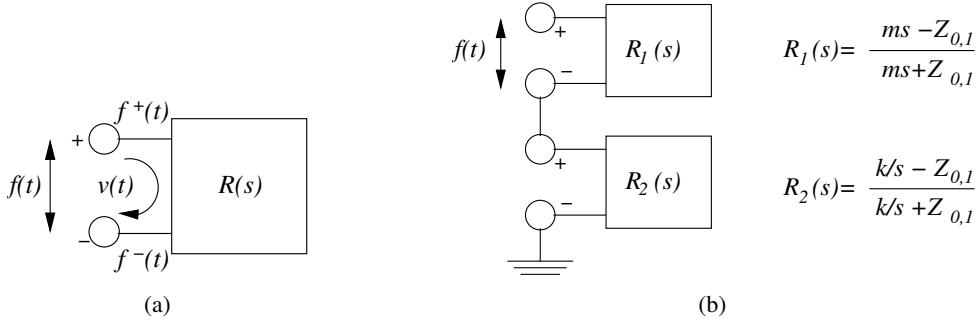


Figure 3.18:

3.5.2 Modal synthesis

Modal synthesis is conceptually simple: the sound of a resonating object is represented as a linear combination of the outputs of N second order oscillators, each of which represents one mode of oscillation of the object excited by a driving force or acoustic pressure: in this sense modal synthesis can be regarded as a lumped physical modeling approach, and can also be interpreted a source-filter approach in which the source is the driving signal and the filter is bank of second-order resonators.

Understanding the mathematical and physical basis of modal theory is a bit less straightforward. In the next sections we sketch the main concepts for both discrete systems (e.g. discrete networks of masses, springs, and dampers) and continuous systems (i.e. partial differential equations in space and time). We show that modal theory is fundamentally the same for these classes of systems.

Therefore the power of modal synthesis is that it is a very general technique that can be applied to a large class of sounding physical systems (while e.g. waveguide techniques are suited only for elastic systems that obey some perturbed version of the D'Alembert equation).

3.5.2.1 Normal modes in finite dimensional systems

In Sec. 3.2.1 we have studied the simple example of two coupled mechanical oscillators, and we have seen that the resulting system can be viewed as the combination of two *uncoupled* oscillators, whose frequencies depend on those of the original ones. This approach can be extended to a generic network of N linear undamped oscillators:

$$\mathbf{M}\ddot{\mathbf{y}}(t) + \mathbf{K}\mathbf{y}(t) = \mathbf{f}_{\text{ext}}(t). \quad (3.85)$$

In this equation \mathbf{y} is a vector containing the displacements of the N points of the network, while \mathbf{M} is the mass matrix: typically (but not necessarily) it is diagonal and contains the masses m_l ($l = 1 \dots N$) of each point of the network. \mathbf{K} is the stiffness matrix and is in general not diagonal because the points are coupled through springs.

Now we consider the homogeneous equation ($\mathbf{f}_{\text{ext}} \equiv 0$) and look for a factorized solution of the form $\mathbf{y}(t) = \mathbf{s} \cdot \sin(\omega t + \phi)$. By substituting this into Eq. (3.85), one finds

$$\mathbf{K}\mathbf{s} = \omega^2 \mathbf{M}\mathbf{s}. \quad (3.86)$$

This is a generalized eigenvalue problem for the matrix \mathbf{K} : more precisely, ω^2 is an eigenvalue of $\mathbf{M}^{-1}\mathbf{K}$ and \mathbf{s} is the associated eigenvector. In general one will find N distinct eigenvalues and eigenvectors ω_i^2 and \mathbf{s}_i (for simplicity we consider normalized \mathbf{s}_i 's). The key property of these eigenvectors

is that they are orthogonal with respect to the mass and the stiffness matrix:

$$\mathbf{s}_j^T \mathbf{M} \mathbf{s}_i = \delta_{i,j} m_i, \quad \mathbf{s}_j^T \mathbf{K} \mathbf{s}_i = \delta_{i,j} k_i, \quad (3.87)$$

where m_i and $k_i = \omega_i^2 m_i$ are real positive scalars. The orthogonality condition also implies that the *modal shapes* \mathbf{s}_i are linearly independent. The \mathbf{s}_i 's can be used to define a *modal transformation*, i.e. a change of spatial coordinates that transforms system (3.85) into a set of N uncoupled oscillators:

$$\mathbf{y} = \mathbf{S} \mathbf{q} \quad \mathbf{q} = \mathbf{S}^T \mathbf{y}, \quad \text{with } \mathbf{S} = [\mathbf{s}_1 | \mathbf{s}_2 | \dots | \mathbf{s}_N]. \quad (3.88)$$

Substituting this into Eq. (3.85) and premultiplying by \mathbf{S}^T yields

$$\mathbf{M}_q \ddot{\mathbf{q}} + \mathbf{K}_q \mathbf{q} = \mathbf{S}^T \mathbf{f}_{\text{ext}}(t), \quad \text{with } \mathbf{M}_q = \mathbf{S}^T \mathbf{M} \mathbf{S}, \quad \mathbf{K}_q = \mathbf{S}^T \mathbf{K} \mathbf{S}. \quad (3.89)$$

By virtue of the orthogonality property, the matrices \mathbf{M}_q and \mathbf{K}_q are diagonal and contain the elements m_i and k_i on their diagonals, respectively. Therefore this is a system of uncoupled oscillators with frequencies ω_i , the quantities m_i and k_i represent the masses and the stiffnesses of these modes.

The matrix \mathbf{S}^T of the modal shapes defines how a driving force \mathbf{f}_{ext} acts on the modes: as a particular case, consider a scalar force acting only on the l th point of the network, i.e. $\mathbf{f}_{\text{ext}}(t) = [0, \dots, f_{\text{ext}}(t), \dots, 0]^T$ (where the only non-null element is in the l th index). This force is applied to the generic i th mode, scaled by the factor $s_{i,l}$, i.e. the shape of the i th mode at the l th point of the network. If this factor is 0, i.e. if the i th mode has a *node* at the l th point of the network, then no force is transmitted to the mode.

The oscillation $y_l(t)$ of the system at the l th spatial point will be the sum of the modal oscillation weighted by the modal shapes, according to Eq. (3.88): $y_l(t) = \sum_{i=1}^N s_{i,l} q_i(t)$. Again, if the i th mode has a *node* at the l th point of the network, that mode will not be “heard” in this point. In conclusion the motion of the network is determined by the motion of N second-order mechanical oscillators and by the transformation matrix \mathbf{S} .

This formalism can be extended to systems that include damping, i.e. where we add a term $R\dot{\mathbf{y}}$ in Eq. (3.85).

3.5.2.2 Normal modes in PDEs

Now look at the concept of normal modes from a different perspective: a distributed object is not modeled as a network of lumped elements, but instead as a partial differential equation that describes the displacement $y(x, t)$ as a continuous function of space and time. We can reformulate the modal description even in this case: as we will see, there are strict analogies with the case of finite dimensional systems outlined above.

We use a concrete example, a string with fixed ends, to derive the modal formulation in the case of continuous systems. In analogy with the case of finite dimensional systems, we now state that a normal mode is a factorized solution $y(x, t) = s(x)q(t)$. For the example under exam, we already know that the D'Alembert equation with fixed boundary conditions admits the factorized solutions $y_n(x, t) = s_n(x)q_n(t)$ of the form (3.21). If a force density $f_{\text{ext}}(x, t)$ is acting on the string, the equation is

$$\mu \frac{\partial^2 y}{\partial t^2}(x, t) - T \frac{\partial^2 y}{\partial x^2}(x, t) = f_{\text{ext}}(x, t). \quad (3.90)$$

If one substitutes in this equation the mode $y_n(x, t)$, and then multiplies by $s_n(x)$ and integrates over the string length, the following equation is found:

$$\left[\mu \int_0^L s_n^2(x) dx \right] \ddot{q}_n(t) - \left[T \int_0^L s_n''(x) s_n(x) dx \right] q_n(t) = \int_0^L s_n(x) f_{\text{ext}}(x, t) dx. \quad (3.91)$$



The second integral can be integrated by parts to obtain

$$\left[\mu \int_0^L s_n^2(x) dx \right] \ddot{q}_n(t) - T \left[s'_n(x) s_n(x) \Big|_0^L - \int_0^L [s'_n(x)]^2 dx \right] q_n(t) = \int_0^L s_n(x) f_{\text{ext}}(x, t) dx, \quad (3.92)$$

where the term $s'_n(x) s_n(x) \Big|_0^L$ is identically zero for fixed (or even free) boundary conditions. Therefore the equation for the n th mode is that of a second-order oscillator with mass $m_n = \mu \int_0^L s_n^2(x) dx$ and stiffness $k_n = T \int_0^L [s'_n(x)]^2 dx$. For the ideal string the modal shapes are simply $s_n(x) = \sin(n\pi x/L)$, therefore $m_n = \mu L/2$ and $k_n = TL/2$.

The shape also defines how a driving force acts on the mode. As a particular case, consider a force density that is ideally concentrated in a single point x_{in} of the string, i.e. $f_{\text{ext}}(x, t) = \delta_D(x - x_{in}) u(t)$ (where the function $\delta_D(\cdot)$ is the Dirac delta): then the force acting on the n th mode is $s_n(x_{in}) u(t)$, and if x_{in} is a node of the mode then no force is transmitted to it. We already know that the oscillation $y(x_{out}, t)$ of the system at the spatial point x_{out} will be the sum of the modal oscillations weighted by the modal shapes: $y(x_{out}, t) = \sum_{n=1}^{+\infty} s_n(x_{out}) q_n(t)$. Again, if the n th mode has a *node* at the point x_{out} , that mode will not be “heard” in this point.

This analysis can be extended to include dispersion and dissipation. As an example, we know that for a dissipative string we have to add the terms $d_1 \partial y / \partial t - d_2 \partial / \partial t (\partial^2 y / \partial x^2)$ on the left-hand side of Eq. (3.90). Again, by substituting the mode $y_n(x, t)$ in the equation, and then multiplying by $s_n(x)$ and integrating over the string length, one finds that the term $\left[d_1 \int_0^L s_n^2(x) dx + d_2 \int_0^L [s'_n(x)]^2 dx \right] \dot{q}(t)$ has to be added to Eq. (3.92), which represents a viscous damping term for the second order oscillator.

M-3.45

Compute modal parameters for a string with linear dissipation.

M-3.45 Solution

```
function [omega, alpha, m, s]=modal_string(L, T, mu, d1, d2, N, M);
xstep=L/(M-1); xpoints=0:xstep:L;
s=zeros(N,M); omega=zeros(1,N); alpha=omega; m=omega;
for i=1:N % i is mode number
    s(i,:)= sin(i*pi*xpoints/L); %spatial shape
    m(i) = mu*xstep*sum(s(i,:).^2); %=mu*L/2; modal mass
    dsdx= i*pi/L*cos(i*pi*xpoints/L);
    k = T*xstep*sum(dsdx.^2); %=T*L/2*(i*pi/L)^2; modal stiffness
    omega(i)=sqrt(k/m(i)); %=i*pi*c/L; modal frequency
    alpha(i)=(d1*xstep*sum(s(i,:).^2)-d2*xstep*sum(dsdx.^2))/(2*m(i)); %loss
end
```

Parameters are functions of the string tension T , linear density μ , loss factors $d_{1,2}$. One can choose the number N of modes to compute, and the number M of spatial points for the shape computation. Two remarks. First, we are using the ideal spatial shapes: this is not correct for the dissipative string, but is acceptable for small $d_{1,2}$ values. Second, we are computing the integrals numerically, although for the ideal string shapes these have analytical solutions: in Sec. 3.5.3 we will examine less trivial shapes.

In conclusion the modal representation of continuous systems described by PDEs is in strict analogy with that of discrete systems described as networks of masses and springs. Here we have obtained similar equations, where the discrete spatial index $l = 1 \dots N$ indicating the points of the network (3.85) has become a continuous spatial variable x , sums over l have become integrals over x , and a numerable



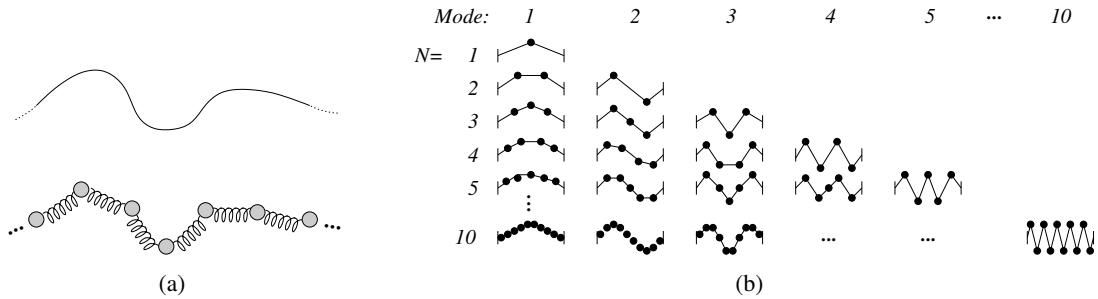


Figure 3.19: Analogies between continuous and discrete systems: (a) approximation of an ideal string with a mass-spring network; (b) modes of the discrete system for different numbers N of masses.

infinity of modes has been found instead of a finite set of N modes. These strict analogies reflect the fact that continuous systems can be seen as the limit of discrete systems when the number of masses becomes infinite. As an example, a string can be approximated with the discrete network of Fig. 3.19(a) made of N masses and $N + 1$ springs. Figure 3.19(b) shows that for a given N the system has N modes, whose shapes resemble closely those of the first N modes of the continuous string. Moreover the approximation grows closer and closer as N increases. One could also show that the modal frequencies of the continuous system are underestimated by those of the discrete system, due to the spatial discretization.

3.5.2.3 Discrete-time mechanical oscillators

We have seen that each mode of either a discrete or a continuous system is a second order oscillator:

$$\begin{aligned} \ddot{q}(t) + 2\alpha\dot{q}(t) + \omega_0^2 q(t) &= \frac{1}{m} f_{\text{mode}}(t), \\ Q(s) = H(s)F_{\text{mode}}(s), \quad \text{with } H(s) &= \frac{m^{-1}}{s^2 + 2\alpha s + \omega_0^2}. \end{aligned} \tag{3.93}$$

The frequency $\omega_0 = k/m$ and the loss factor $\alpha = r/m$ depend on the geometry and the material of the object. The force f_{mode} that is “felt” by a single mode depends on the modal shape and on the spatial force distribution, and is scaled by the modal mass m . The displacement $y(x, t)$ at a certain point x of the structure is a linear combination of the modes $q(t)$, where the coefficients of the linear combination are the modal shapes $s(x)$ at the point x . This is true whether we have a discrete set of points x_i or a continuous domain, although in practice the spatial domain will be always discretized.

In order to construct a modal synthesizer, the first step to perform is to construct a discrete-time equivalent of the second order oscillator (3.93). We can discretize the differential equation with the numerical methods examined previously in Sec. 3.5.1. The impulse invariant method yields:

$$H(z) = \frac{\left[T_s \left(\frac{e^{-\alpha T_s}}{m\omega_r} \right) \sin(\omega_r T_s) \right] z^{-1}}{1 - [2e^{-\alpha T_s} \cos(\omega_r T_s)] z^{-1} + e^{-2\alpha T_s} z^{-2}}. \quad (3.94)$$

The presence of a z^{-1} factor at the numerator indicates that this is an explicit numerical scheme (there is no instantaneous dependence on the input). The backward Euler method yields

$$H(z) = \frac{\frac{1}{m(F_s^2 + 2\alpha F_s + \omega_0^2)}}{1 - \frac{2F_s(\alpha + F_s)}{F^2 + 2\alpha F_s + \omega_0^2} z^{-1} + \frac{F_s^2}{F^2 + 2\alpha F_s + \omega_0^2} z^{-2}}. \quad (3.95)$$

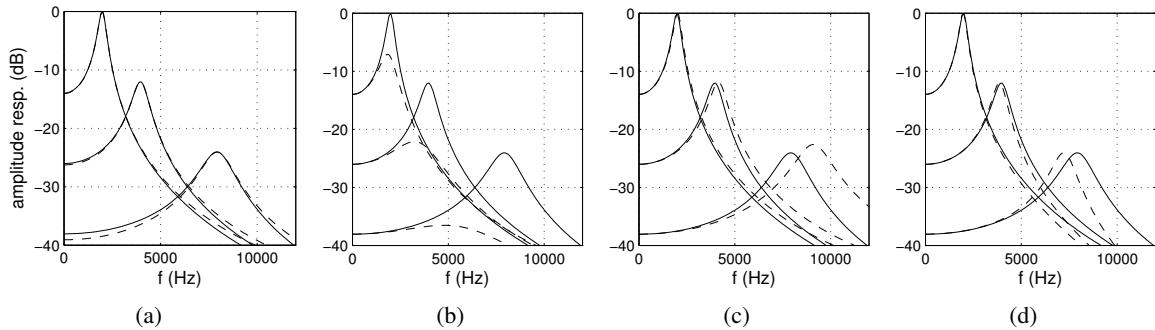


Figure 3.20: Amplitude responses of a second order oscillator with constant mass and quality factor, and $\omega_0 = 2, 4, 8 \text{ kHz}$: continuous-time responses (solid lines) and discrete-time responses (dashed lines) with (a) impulse invariant method, (b) backward Euler method, (c) backward Euler method with centered scheme, (d) bilinear transform.

This instead is an implicit numerical scheme. The backward Euler method with centered scheme yields

$$H(z) = \frac{\frac{T_s^2}{m} z^{-1}}{1 + [\omega_0^2 T_s^2 + 2\alpha T_s - 2] z^{-1} + [1 - 2\alpha T_s] z^{-2}}. \quad (3.96)$$

Like in the impulse invariant case, this is an explicit numerical scheme. By looking at the poles of this discrete-time system one can see that it can become unstable depending on the mechanical parameters and on the sampling period: the scheme is not *unconditionally stable*. Finally, the bilinear transform yields

$$H(z) = \frac{\left[\frac{1}{m(4F_s^2 + 4\alpha F_s + \omega_0^2)} \right] (1 + 2z^{-1} + z^{-2})}{1 + \frac{2(\omega_0^2 - 4F_s^2)}{4F_s^2 + 4\alpha F_s + \omega_0^2} z^{-1} + \frac{4F_s^2 - 4\alpha F_s + \omega_0^2}{4F_s^2 + 4\alpha F_s + \omega_0^2} z^{-2}}. \quad (3.97)$$

Like in the case of the backward Euler method, this is an implicit numerical scheme.

The resulting amplitude responses are shown in Fig. 3.20. As expected, the impulse invariant method exhibits aliasing, the Euler method exhibits warping and numerical damping, the Euler method with centered scheme tends to become unstable for high ω_0 values, and the bilinear transform exhibits warping (but not numerical damping).

M-3.46

Write a function that computes the filter coefficients of the mechanical oscillator discretized with (a) the impulse invariant method, (b) the Euler method $g_1(z)$, (c) Euler method with the centered estimate (3.76), and (d) the bilinear transform. Compare the frequency responses of the resulting discrete-time systems.

M-3.46 Solution

```
function [B, A]=modal_oscillator(m, alpha, omega, method)

global Fs; Ts=1/Fs;

switch method
case 'impinv'
    omegar=sqrt(omega^2-alpha^2); eaTs=exp(-alpha*Ts);
    B= [0 Ts*eaTs/ (m*omegar)*sin(omegar*Ts) 0];
    A= [1 -2*alpha/Ts -Ts^2/m];
end
```

```

A= [1 -2*eaTs*cos(omegar*Ts) eaTs^2];
case 'euler'
    delta= Fs^2+2*alpha*Fs+omega^2;
    B= [1/ (m*delta) 0 0];
    A= [1 -2*Fs*(alpha+Fs)/delta Fs^2/delta];
case 'eulercenter'
    B= [0 Ts^2/m 0];
    A= [1 (omega^2*Ts^2 +2*alpha*Ts -2) (1-2*alpha*Ts) ];
case 'bilin'
    delta=4*Fs^2 +4*alpha*Fs +omega^2;
    B= [1/(m*delta) 2/(m*delta) 1/(m*delta)];
    A= [1 2*(omega^2 -4*Fs^2)/delta (4*Fs^2 -4*alpha*Fs +omega^2)/delta];
otherwise error('unknown numerical method');
end

```

3.5.2.4 A modal synthesizer

A simple modal synthesizer can be constructed as a parallel connection of N numerical oscillators. By choosing a different center frequency ω_0 and damping factor α for each oscillator, it is possible to account for a set of partials and decay times of the resonator spectrum. Moreover, the modal shapes determine both how a force signal is injected into the modal oscillator and how the modal oscillations are combined

M-3.47

Write a function that computes the output of a modal resonator given an input force signal.

M-3.47 Solution

```

function y = modal_synth(x, omega, alpha, m, s, in, out, method);

global Fs;
N=length(omega);% it must be size(omega)=size(alpha)=size(m)=N
                 % it must be size(s,1)=N; 0<in<size(s,2); 0<out<size(s,2);
y=zeros(1,length(x));
for i= 1:N
    [B,A]=modal_oscillator(m(i),alpha(i),omega(i),method);
    y_i=filter(B,A,s(i,in)*x);
    y=y +s(i,out)*y_i;
end

```

We have assumed that the force distribution is concentrated in a single point, represented by the index `in`. We “pick-up” the resonator signal at another point, represented by the index `out` (like we were using a contact mike attached to the object at the point `out`).

The input modal parameters can be chosen to match those of an arbitrary object. Moreover, morphing between different shapes and material can be obtained by designing appropriate trajectories for these parameters.

M-3.48

Synthesize the sound of a dissipative string using the modal approach.

M-3.48 Solution



```

global Fs; Fs=44100;
slength=8; %sound length (s)

%%%%% Physical parameters for a E3 guitar nylon string %%%%
mu=5.25e-3;
T = 60; % string tension (N)
L= 0.65; %string length (m)
d1=mu*.65;
d2=-T*9e-8;
%%%%%%%%%%%%%%%
fmax=30; % impulsive force (N)
x=[fmax zeros(1,round(slength*Fs))]; %force signal zero-padded to sound length

N= 120; M=round(L/5e-3); %no. of modes and spatial points (incl.ends)
in=round(M/20); out=round(M/20); %input and output points
[omega,alpha,m,s]=modal_string(L,T,mu,d1,d2,N,M); %compute modal parameters
y=modal_synth(x,omega,alpha,m,s,in,out,'impinv'); %compute output signal

```

3.5.3 Modal analysis

The modal synthesizer that we have constructed needs to know the modal parameters for the specific resonator under exam. The question is then how to determine these parameters.

In the case of a discrete system of N point masses with linear interaction forces, modal parameters are exactly found through standard matrix calculations. Most systems of interest of course do not fit these assumptions. For some distributed systems, particularly for symmetrical problems with simple boundary conditions, the partial differential equation describing the system can be solved analytically, giving the modal parameters. Alternatively, either accurate numerical simulations (e.g. wave-guide mesh methods) or “real” physical measurements can be used.

3.5.3.1 Simple 1-D shapes

The simplest tractable case is the ideal string: we have already discussed the modal solution in this case. There are other tractable cases: one interesting example is the ideal bar, with various boundary conditions. Bars are almost as relevant as strings for musical applications: mallet percussion instruments, such as the marimba, the xylophone, the vibraphone, and so on, are based on the oscillations of bars.

Transverse vibrations in a bar are due to internal elastic force generated when the bar is bent. One can show that for a bar with constant cross-section the equation governing the bar transversal displacement y is the Euler-Bernoulli equation:

$$\frac{\partial^2 y}{\partial t^2}(x, t) = -\frac{EK^2}{\rho} \frac{\partial^4 y}{\partial x^4}(x, t), \quad (3.98)$$

where E is the Young modulus of the material, K is the radius of gyration,⁷ and ρ is the volume density. Note that the fourth-order term is the one that we used to describe a stiff (and dispersive) string. The modal solutions $y(x, t) = s(x)q(t)$ are in this case

$$y(x, t) = [A \cosh kx + B \sinh kx + C \cos kx + D \sin kx] \cos(\omega t + \phi), \quad \text{with } k = c\omega, \quad (3.99)$$

⁷This would need some explanation. In short: $K^2 = \frac{1}{S} \int z^2 dS$, where $S = \int dS$ is the total cross-section of the bar and z is the distance from the neutral axis, i.e. the axis along the bar which does not change its length when the bar is bent (at one side of the neutral axis there is elongation, at the other side there is compression). Everything clear???



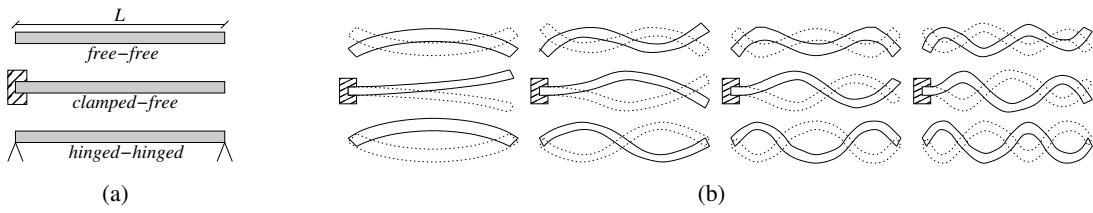


Figure 3.21: Modal description of the ideal bar: (a) ideal bar with various boundary conditions and (b) corresponding modes.

and where $c^2 = \omega K \sqrt{E/\rho}$. This modal solution cannot be interpreted in terms of traveling waves, therefore waveguide methods fall short here, while modal synthesis can be successfully employed.

The constants A, B, C, D as well as the allowed frequencies are determined depending on four boundary conditions (two at each end). The conditions for a *free* end are $\partial^2 y / \partial x^2 = \partial^3 y / \partial x^3 = 0$ (no torque and no shearing force); those for a *supported* (hinged) end are $y = \partial^2 y / \partial x^2 = 0$ (no displacement and no torque); and those for a *clamped* end are $y = \partial y / \partial x = 0$ (no displacement and zero slope). Three notable examples are shown in Fig. 3.21(a). For these cases, numerical solution of the equations resulting from boundary conditions yields

$$\begin{aligned} \text{(free-free)} \quad \{\omega_n\} &= \frac{\pi^2 K}{4L^2} \sqrt{\frac{E}{\rho}} [3.011^2, 5^2, 7^2, \dots, (2n+1)^2, \dots], \\ \text{(clamped-free)} \quad \{\omega_n\} &= \frac{\pi^2 K}{4L^2} \sqrt{\frac{E}{\rho}} [1.194^2, 2.988^2, 5^2, \dots, (2n-1)^2, \dots], \\ \text{(hinged-hinged)} \quad \{\omega_n\} &= \frac{2\pi^2 K}{L^2} \sqrt{\frac{E}{\rho}} n^2. \end{aligned} \quad (3.100)$$

Note that in the first two cases the frequencies are strongly inharmonic, while in the third case they are harmonically related: the corresponding lowest modes are shown in Fig. 3.21(b). Mallet percussions most typically use bars with (approximately) free-free conditions. However in many cases their bars do not have constant cross-sections, instead their are cut with an arch on the underside in such a way that the theoretical partials of the free-free series in Eq. (3.100) are shifted and aligned to an almost harmonic series.

M-3.49

Compute modal parameters for a bar with the three boundary conditions examined here, and with linear dissipation.

M-3.49 Solution

Like Example M-3.45, but using the modal shapes of the ideal bar.

3.5.3.2 Simple 2-D shapes

The first example of a musically relevant 2-D shape is a rectangular membrane with fixed ends, like the one depicted in Fig. 3.22(a). The ideal membrane obeys the 2-D D'Alembert equation:

$$\sigma \frac{\partial^2 z}{\partial t^2}(x, y, t) = T \left[\frac{\partial^2 z}{\partial x^2}(x, y, t) + \frac{\partial^2 z}{\partial y^2}(x, y, t) \right] = T \nabla^2 z(x, y, t), \quad (3.101)$$



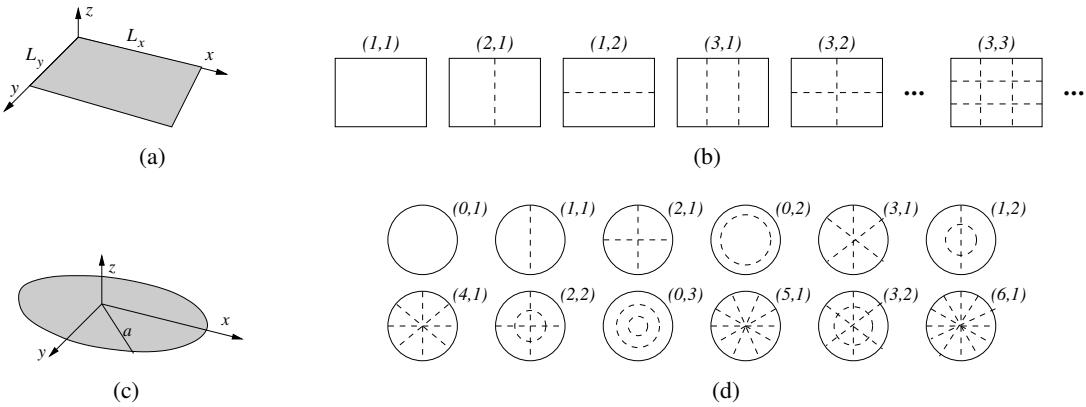


Figure 3.22: Modal description of ideal membranes: (a) ideal rectangular membrane with fixed ends and (b) corresponding modes; (c) ideal circular membrane with fixed ends and (d) corresponding modes.

where z is the membrane vertical displacement and the constants T and σ are the membrane surface tension (in N/m) and surface density (in Kg/m²). The symbol $\nabla^2 = \partial^2/\partial x^2 + \partial^2/\partial y^2$ stands here for the 2-dimensional Laplacian operator. Modal solutions $z(x, y, t) = s^{(x)}(x)s^{(y)}(y)q(t)$ are found with the same procedure used for the ideal string:

$$z_{n,m}(x, y, t) = \sqrt{\frac{2}{L_x}} \sqrt{\frac{2}{L_y}} \sin\left(k_n^{(x)} x\right) \sin\left(k_m^{(y)} y\right) \cos(\omega_{n,m} t + \phi_{n,m}), \quad (3.102)$$

with $k_n^{(x)} = \frac{n\pi}{L_x}$, $k_m^{(y)} = \frac{m\pi}{L_y}$, $\omega_{n,m} = c \sqrt{\left[k_n^{(x)}\right]^2 + \left[k_m^{(y)}\right]^2}$.

Note that the modal frequencies $\omega_{n,m}$ are not harmonically related in this case. The modal shapes $s_{n,m}(x, y) = s_n^{(x)}(x)s_m^{(y)}(y)$ have straight nodal lines: the lowest modes are shown in Fig. 3.22(b).

A second example, even more relevant for musical applications, is the circular membrane with fixed ends, like the one in Fig. 3.22(c). In this case the 2-D D'Alembert equation is more conveniently written in circular coordinates $x = r \sin \theta$ and $y = r \cos \theta$ and the laplacian becomes $\nabla^2 = \partial^2/\partial r^2 + 1/r(\partial/\partial r) + 1/r^2(\partial/\partial\theta)$. Accordingly, one looks for modal solutions of the form $z(r, \theta, t) = s^{(r)}(r)s^{(\theta)}(\theta)q(t)$.

Substituting this into the 2-D D'Alembert equation results in two differential equations for $s^{(r)}$ and $s^{(\theta)}$. One finds the angular shapes $s_m^{(\theta)}(\theta) = \cos(m\theta)$. Then for each m , the radial shapes are $s_n^{(r)}(r) = J_m(k_{m,n}^{(r)}r)$, i.e. they are the first-kind Bessel functions of order m , with radial frequencies $k_{m,n}^{(r)}$. The allowed values for $k_{m,n}^{(r)}$ are found as usual by imposing that $s_n^{(r)} = 0$ at the fixed boundary, therefore are determined by the n th zero of J_m . In conclusion the m, n mode has m nodal diameters (determined by the function $s_m^{(\theta)}$) and n nodal circles (determined by the function $s_n^{(r)}$). The lowest modal frequencies $\omega_{n,m}$ are

$$\{\omega_{n,m}\} = \frac{2.405c}{a} [1, 1.594, 2.136, 2.296, 2.653, 2.918, 3.156, 3.501, 3.6, 3.652, 4.06, 4.154], \quad (3.103)$$

and are highly inharmonic. The corresponding modes are shown in Fig. 3.22(d).

M-3.50

Compute modal parameters for a rectangular and a circular bar with fixed boundary conditions, and with linear dissipation.



M-3.50 Solution

Like Example M-3.45, but using the modal shapes of the ideal rectangular and circular membrane.

3.5.3.3 Experimental estimation

When the modal solution cannot be written analytically, modal parameters can still be estimated. One approach is to extract modal data from a recorded audio signal. Various methods are known that can estimate resonances (center frequencies and quality factors) from a signal: these include linear prediction techniques and partial tracking techniques examined in Chapter *Sound modeling: signal based approaches*. However there are various problems to deal with. First, modal frequencies are often very closely spaced and one needs high-resolution methods that are able to discriminate nearby resonances. Second, estimates derived from analysis of a single sound lack information about the spatial shapes of the modes. Third, there are many inaccuracies related to technical difficulties in the recording of object responses: ideally one should record the impulse response of an object, for many different excitation points and many different pick-up points. In practice one will strike the object and record the response in air, with consequent spatially distributed interactions, and sound radiation through air.

Modal shapes may be observed through more sophisticated measurement devices, e.g. by using holographic interferometry. A relatively simple experimental technique amounts to place some finely divided material (e.g. fine sand or flour) on the resonating object (e.g. a plate of arbitrary shape), and then setting the object into forced oscillation (most typically through mechanical or electromechanical means) with a sinusoidal driving signal which is tuned to the frequency of the desired mode. As a consequence one will observe the sand on the object bouncing and moving about, and only at or near the nodal lines of the mode the sand will be stationary. Thus the sand is either bounced off the object or else collects at the nodes, forming so-called *Chladni patterns* (from the name of the german physicist and musician who first observed nodal patterns through this technique). Variations of this technique have been commonly used by acoustic instrument makers, especially for the design and construction of the resonating bodies of violins, guitars, cellos, etc.

An alternative “experimental” approach amounts to simulate the response of an object with finite difference or finite element methods. This implies spatial discretization, which means that only a finite amount of modes can be estimated. Moreover, modal data obtained in this way suffers from underestimation of modal frequencies, due to errors introduced by spatial discretization.

3.6 Non-linear physical models

So far in this Chapter we have examined linear models, mostly employed to simulate physical resonators. However musical oscillators are often strongly non-linear.

Non-linearities must be present for a system to reach stable self-sustained oscillations. As an example, self-sustained oscillations in the acoustic bore of a woodwind or brass instrument can only be explained in terms of a non-linear, persistent excitation mechanism. More precisely, the valve (a single or double-reed, or the player’s lips) at the bore termination acts as a non-linear element that injects energy into the system. A very similar description holds for bowed string instruments, where the bow and its non-linear friction force is the exciting element. In other cases the instrument is non-linearly excited only for a limited amount of time: a struck string or bar interacts with the hammer or mallet through a non-linear contact force. Values for the contact time are typically a few milliseconds, and after this short excitation the system evolution is linear and the oscillations decay away.

Generalizing from the above examples, we may schematize a musical instrument (or any sound-producing physical system) by means of two main functional blocks, as in Fig. 3.23. The *resonator* is



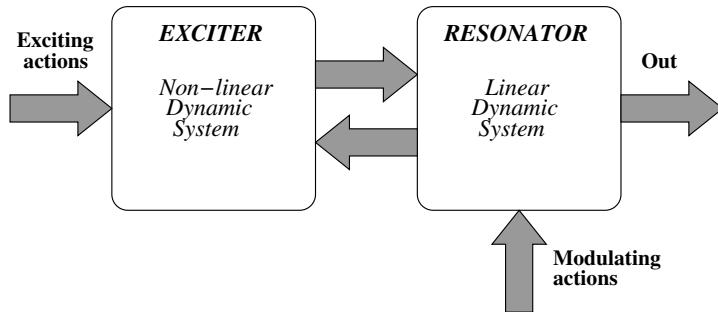


Figure 3.23: Exciter-resonator interaction scheme for a musical instrument.

where the oscillations actually take place (an acoustic bore, a string, a bar, etc) and is therefore related to such sound attributes as pitch, spectral envelope, and so on. The *exciter* controls the way energy is injected into the system, thus initiating and possibly sustaining the oscillations, and relates in particular to properties of attack transients. A simple yet striking demonstration of the effectiveness of the exciter/resonator schematization is provided by mounting a clarinet mouthpiece on a flute.⁸ The bore boundary conditions are changed from open-open to closed-open, so that it plays one octave lower, and the resulting instrument is perceived as a bad sounding clarinet. In other words, the excitation mechanism defines sound identity (“it’s a clarinet”), while the resonator is mostly associated to sound quality (“it’s a *bad* clarinet”).

The interaction between the two blocks is a two-way interaction, where the state of each block influences the other. As an example, the impact force between a hammer and a string depends on the displacements and velocities of both hammer and string, and affects both. Clearly there are also examples where non-linearities in the excitation are negligible: plucked string instruments can be conveniently treated as linear systems (strings and instrument body), where the “pluck” is described as a non-equilibrium initial condition (i.e., the pluck gives a string a non-zero displacement distribution and a null velocity distribution).

Finally, note that non-linearities are not necessarily related to excitation mechanisms only: even resonators, that are assumed to be linear in a first approximation, can exhibit non-linear behaviors. As an example, when a string vibrates outside the limit of small oscillations its length cannot be anymore assumed to be constant, but varies (together with string tension) during an oscillation cycle: this length- and tension-modulation mechanism can produce perceivable pitch glides in the sound. Similar considerations apply to other systems (e.g. non-linear circuit elements).

3.6.1 Non-linear circuits

3.6.1.1 Non-linear capacities

Consider the well known *Chua-Felderhoff* electrical circuit: this is a *RLC* circuit, made of a series connection of a resistor R , an inductor L and a capacitor C . The elements R and L are constant, while this is not the case for C . More precisely, the characteristic of the capacitance is a function of the voltage v ,

⁸The author has enjoyed a live demonstration with such a “flarinet”, performed by Joe Wolfe while giving a seminar in Venice, 2000.

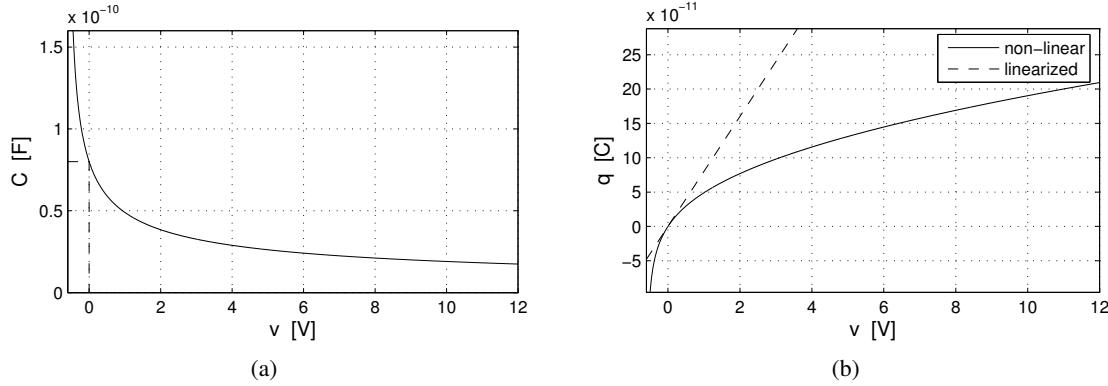


Figure 3.24: Non-linear behavior of (a) capacitance $C(v)$ and (b) charge $q(v)$ in the Chua-Felderhoff circuit.

so that the system is described as follows:

$$v(q) = \frac{1}{2v_0C_0^2} \left(q^2 + q\sqrt{q^2 + 4v_0^2C_0^2} \right), \quad \Leftrightarrow \quad C(v) = \frac{C_0}{\sqrt{1 + \frac{v}{v_0}}}, \quad (3.104)$$

$$v(q) + R\dot{q}(t) + L\ddot{q}(t) = v_e(t), \quad (v > v_0).$$

The variable $q(t)$ stands for the charge on the capacitor, and $v_e(t)$ is an applied voltage. Note that $C(v) \sim C_0$ when $v \rightarrow 0$, i.e. the system is a linear RLC circuit in the limit of small oscillations. However, for larger voltage v this approximation does not hold, and $C(v)$, $q(v)$ behave as depicted in Fig. 3.24(a) and (b), respectively. There is no easy way to translate the non-linear relation (3.104) into the Laplace domain, because the definition of impedance given in Sec. 3.2.1 assumes linearity of the circuit elements.

The Chua-Felderhoff circuit has been extensively studied and is one of the classical systems used for exemplifying transition to chaotic behavior: when the peak of the voltage generator is increased, the behavior of the charge $q(t)$ on the capacitor undergoes successive bifurcations.

3.6.1.2 Vacuum tubes

.....

3.6.2 Mechanical interactions

3.6.2.1 Impacts

Several musical and non musical classes of sounds are produced by a single impact of two objects, one of which (at least) resonates as a consequence of the collision. Moreover, impact is at the basis of other more complex mechanical contacts: as an example, scraping and rolling can be seen as temporal sequences of micro-impacts between non-smooth surfaces.

The ideal impact is a force signal shaped like a Dirac delta in time. It imparts to the resonator an ideal force impulse in an infinitesimal time. If the resonator is initially at rest, such force impulse imparts to the resonator initial conditions given by null initial displacement and a non-zero initial velocity whose magnitude depends on the magnitude of the delta.



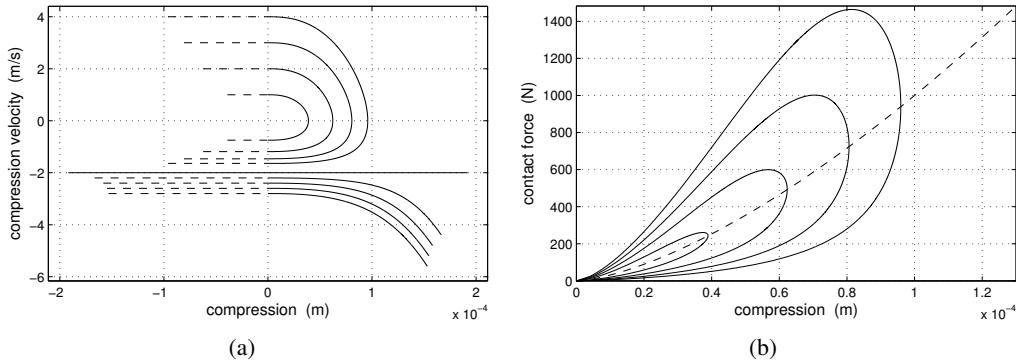


Figure 3.25: The non-linear impact model (3.106): (a) phase portrait of a point mass hitting a hard surface; (b) the corresponding non-linear force during impact.

In a less ideal impact model one would assume that the impact force is non-null over a finite duration or time (the *contact time* between the colliding objects) and takes finite values. The force magnitude is related to the impact energy (e.g. the impact velocity of the hammer hitting the resonator), while the contact time is related to the hardness of the impact. A simple signal model of the impact force is the following:

$$f(t) = \begin{cases} \frac{f_{\max}}{2} \left[1 - \cos\left(\frac{2\pi t}{\tau}\right) \right], & 0 \leq t \leq \tau, \\ 0, & \text{otherwise,} \end{cases} \quad (3.105)$$

where τ is the contact time and f_{\max} is the maximum force value.

More complex models must take into account other effects. There is dissipation of energy during contact. The contact force itself is a function of the relative compression $x(t)$ between the two contacting objects (which may be thought as the difference between the displacements of the two objects during the contact), and also of the compression velocity $v(t) = \dot{x}(t)$. Accordingly, a more physically-based model of the impact force is the following:

$$f(x(t), v(t)) = \begin{cases} kx(t)^\alpha + \lambda x(t)^\alpha v(t), & x > 0, \\ 0, & \text{otherwise,} \end{cases} \quad (3.106)$$

where k is the force *stiffness*, λ is the force damping weight, and the exponent α depends on the local geometry around the contact area. As an example, according to Herz theory of contact an ideal impact between two spherical objects obeys this equation with $\alpha = 3/2$ and $\lambda = 0$.

Figure 3.25(a) depicts the simulation of a point mass hitting a rigid surface with the impact model (3.106): the phase portrait shows that due to dissipation the mass velocity after the impact is always lower in magnitude than the initial impact velocity, and converges to a limit value. Figure 3.25(a) shows the corresponding impact force: it has a non-linear characteristics that depends on the exponent α , and it exhibits a hysteresis effect that is associated to the dissipative component $\lambda x^\alpha v$. This plot is qualitatively resemblant of what one would observe by measuring the contact force during a real impact of a small mass against a rigid surface.

M-3.51

Simulate a modal oscillator excited by the non-linear impact force (3.106).



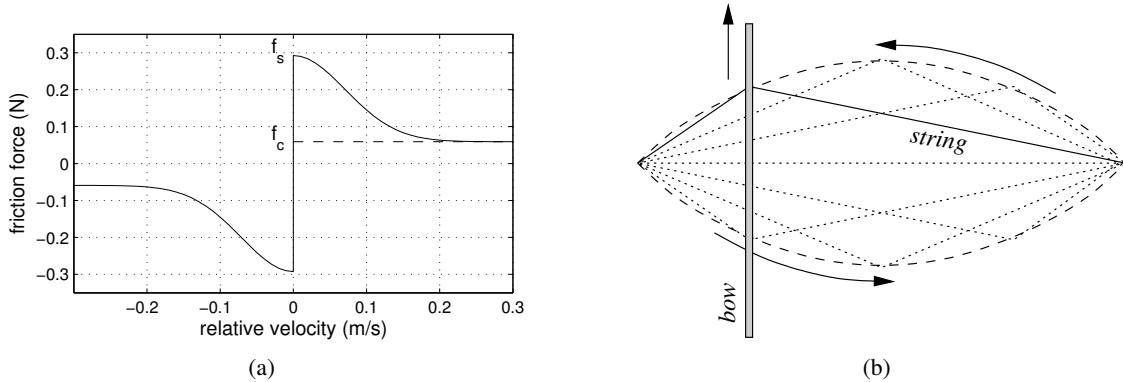


Figure 3.26: Stick-slip friction: (a) example of parametrization of a kinetic (static) friction curve; (b) Helmholtz motion resulting from stick-slip ideal string-bow interaction.

3.6.2.2 Stick-slip friction

Stick-slip friction is a second relevant mechanical interaction in sound production. A typical musical example is the interaction between bow and string in a violin. A non musical example is the sound produced by a finger rubbing on a moist window or on a glass.

We know from physics that static friction is higher than dynamic friction: the simplest model assumes that the friction force is proportional to the normal force f_N between two contacting objects, but the coefficient of proportionality is higher if there is no relative motion and is lower if there is relative motion. More refined models define the coefficient of proportionality as a function of the relative velocity. These are called *kinetic* models (as the friction force is assumed to be a function of velocity only), or *static* models (since the force-velocity dependence is derived under stationary conditions). One possible parametrization of a kinetic friction force model is:

$$f(v(t)) = \begin{cases} \text{sgn}(v) \left[f_c + (f_s - f_c)e^{-(v/v_s)^2} \right], & f_N > 0, \\ 0, & \text{otherwise,} \end{cases} \quad (3.107)$$

where f_c, f_s are the Coulomb force and the stiction (short for static friction) force respectively, while v_s is named Stribeck velocity. The Coulomb force and the stiction force are related to the normal force through the equations $f_s = \mu_s f_N$ and $f_c = \mu_d f_N$, where μ_s and μ_d are the static and dynamic friction coefficients. If $f_N \leq 0$ this means that there is no contact. The dependence of the friction force on velocity, as given in Eq. (3.107), is shown in Fig. 3.26(a).

When two objects in relative motion interact through a friction force of this kind, a *stick-slip* phenomenon is generated in which the two objects remain in static contact for a certain amount of time (the “stick” phase) and suddenly detach (the “slip” phase). Sound generation occurs when this alternation of stick and slip phases occurs in an almost periodic fashion and with an audio rate, typically locked to some of the proper resonance frequencies of the interacting objects.

An example of stick-slip interaction is the *Helmholtz motion* occurring in an ideal, rigidly terminated, bowed string (see Fig. 3.26(b)). Assuming the bow to be perfectly rigid and to be in contact with the string in a single point, the string motion at the contact point is a sawtooth signal in which the string remains stuck to the bow hair for a considerable fraction of each vibratory cycle, and slips back abruptly when its displacement becomes large enough, to begin the next cycle. In normal playing condition the resulting frequency of oscillation is almost coincident with the first-mode frequency of the string. Further



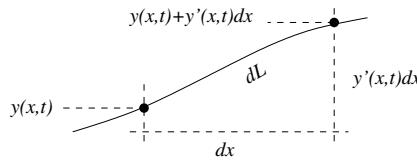


Figure 3.27: Length dL of a string at point x over the segment dx .

analysis of this Helmholtz motion would reveal that at every instant the shape of the string consists of two line segments joined by a corner, and the corner travels on an envelope composed of two parabolas.

M-3.52

Simulate a modal oscillator excited by the non-linear friction force (3.107).

More refined, *dynamic* friction models include some “memory”. The dependence of friction on the relative sliding velocity is modeled using a differential equation. These models are able to take into account presliding behavior, where the friction force increases gradually for small displacement values. Static and dynamic friction models have the same behavior at high or stationary relative velocities, but dynamic models provide more accurate simulation of transients, which is particularly relevant for realistic sound synthesis.

3.6.2.3 Tension modulations

The phenomenon of tension modulation is qualitatively different from the previous examples. This non-linear effects is not generated from an external excitation force. It is a non-linear correction to the D'Alembert equation when the limit of small oscillations of the elastic medium is not valid.

The simplest example of tension modulation is encountered in a vibrating string with fixed ends. When the string is significantly displaced from equilibrium, its length and therefore also its tension are increased. When it returns closer to its equilibrium state, its length and tension are decreased. Clearly the rate of this tension modulation is twice the rate of the transversal vibration, since minimum tension occurs at equilibrium, and maximum tension occurs at both extreme displacements.

$$\mu \frac{\partial^2 p}{\partial t^2}(x, t) - T[y(x, t)] \frac{\partial^2 p}{\partial x^2}(x, t) + EI \frac{\partial^4 p}{\partial^4 x}(x, t) + d_1 \frac{\partial p}{\partial t}(x, t) - d_2 \frac{\partial^3 p}{\partial t \partial^2 x}(x, t) = 0, \quad (3.108)$$

where $T[y(x, t)]$ is the string tension and is now a function of the string displacement. More precisely, it is proportional to the string length, which in turns depends on $y(x, t)$. From the theorem of Pithagoras, the length dL at point x over the segment dx is (see Fig. 3.27) $dL[y(x, t)] = \sqrt{dx^2 + (y'(x, t)dx)^2}$. Then the total string length deviation ΔL from the length L_0 at equilibrium is

$$\Delta L[y(x, t)] = L[y(x, t)] - L_0 = \int_0^{L_0} dL[y(x, t)] - L_0 = \int_0^{L_0} \sqrt{1 + y'(x, t)^2} dx - L_0. \quad (3.109)$$

Then the tension is

$$T[y(x, t)] = T_0 + \frac{EA\Delta L[y(x, t)]}{L_0}, \quad (3.110)$$

where A is the string section.

Tension modulation in a waveguide model can be simulated by using all-pass filters with time-varying coefficients that account for length modulation. Tension modulation in modal synthesis can be simulated

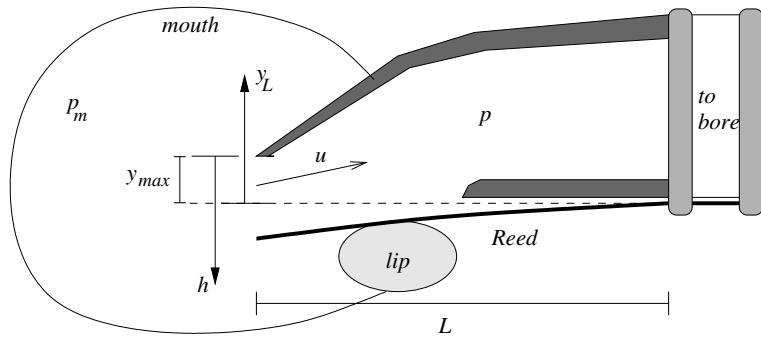


Figure 3.28: Schematic representation of the reed-mouthpiece system.

by finding the modes of Eq. (3.108). As an example, in the simple case where $d_2 = E = 0$ in Eq. (3.108), one can find the modes

$$\ddot{q}_i(t) + d_1\dot{q}_i(t) + i^2 \left[\omega_0^2 + \omega_1^2 \sum_{l=1}^{+\infty} l^2 q_l^2(t) \right] q_i(t) = 0, \quad i = 1, \dots, +\infty, \quad (3.111)$$

that can be interpreted as mechanical oscillators in which the frequency of oscillation depends on the modal displacement. In the general case of Eq. (3.108) including dispersion and frequency-dependent dissipation, a similar modal description can still be found.

3.6.3 Acoustic interactions

3.6.3.1 Jets

.....

3.6.3.2 Quasi-static reeds

Reeds are acoustic valves that oscillate due to pressure differences at the two sides. The simplest example is the *single reed*, schematically represented in Fig. 3.28. The reed dimensions are small with respect to typical wavelengths in the resonator, thus pressure can be thought of as constant along the reed surfaces; under normal playing conditions, the first mode of oscillation of the reed is well above the main frequency components of the pressure signal in the resonator. Oscillations occur mainly in the vertical direction, and a single degree of freedom can be reasonably assumed, i.e. the vertical displacement y_L of the reed tip from the equilibrium. These considerations justify the choice of a *lumped* modeling approach for the reed.

The simplest possible lumped model regards the reed as a system with stiffness only, neglecting inertia and damping properties. In this approximation the reed moves in phase with the pressure difference $\Delta p(t) = p_m(t) - p(t)$ across the reed:

$$ky_L(t) = S_d \Delta p(t) \quad \Rightarrow \quad k_a y_L(t) = \Delta p(t), \quad (3.112)$$

where p_m is the pressure inside the performer's mouth, p is the (oscillating) acoustic pressure inside the instrument bore, k is the effective reed stiffness, S_d is an effective driving surface on which the pressure Δp acts, and $k_a = k/S_d$ is the stiffness per unit area. Equation (3.112) is called a *quasi-static approximation* since it can be determined experimentally in static conditions where a constant pressure

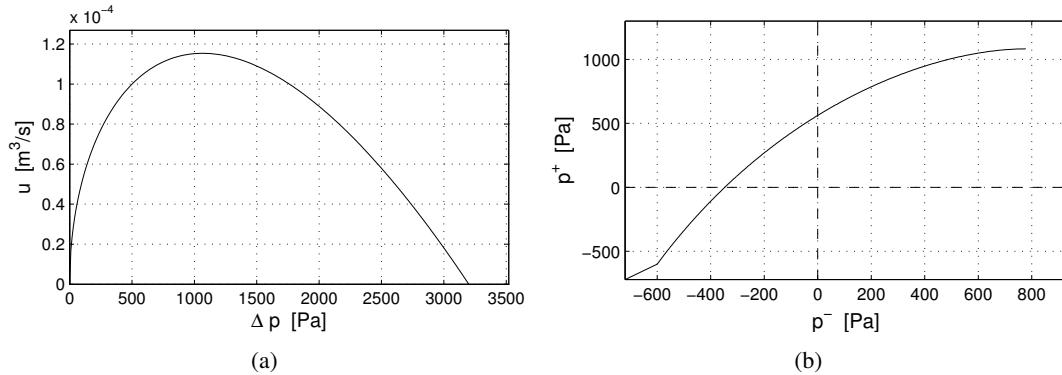


Figure 3.29: Quasi-static approximation of a single reed; (a) u versus Δp and (b) rotated mapping p^+ versus p^- .

difference δp is injected into the system and the corresponding constant displacement y_L is measured after an initial transient.

As far as aerodynamics is concerned, the relation between the reed opening $h(t)$, the airflow $u(t)$ through the slit, and the pressure drop $\Delta p(t)$ can be approximated through the equation

$$\Delta p(t) = f(u(t), h(t)) = \text{sgn}[u(t)] \frac{\rho_{\text{air}}}{2} \frac{|u(t)|^2}{wh(t)}, \quad (3.113)$$

where w is the reed width. This equation is derived from the Bernoulli law.⁹ Using Eq. (3.112), the reed opening h is computed as $h = y_{\max} - y_L = y_{\max} - \Delta p/k_a$, and by substituting this relation into Eq. (3.113) one finds

$$u(t) = \begin{cases} w \text{sgn}[\Delta p(t)] \left(y_{\max} - \frac{\Delta p(t)}{k_a} \right) \sqrt{\frac{2|\Delta p(t)|}{\rho_{\text{air}}}}, & \Delta p < k_a y_{\max}, \\ 0, & \text{otherwise.} \end{cases} \quad (3.114)$$

Figure 3.29(a) shows the plot of the resulting relation between u and Δp . For low Δp values, u increases until a maximum is reached at $\Delta p = k_a y_{\max}/3$. For higher Δp values, the flow starts to drop due to reed closure, and reaches the value $u = 0$ at $\Delta p = k_a y_{\max}$. Beyond this value the reed is completely closed.

This non-linear map can be used to construct a quasi-static reed model. If wave variables p^\pm are introduced in the cylindrical bore, i.e. $p = p^+ + p^-$ and $u = p^+ - p^-$, then these relations can be substituted into Eq. (3.114). As a consequence this non-linearity can be turned in a new one in which p^+ depends on p^- through a non-linear reflection function R_{nl} , i.e. $p^+ = R_{nl}(p^-)$. This is depicted in Fig. 3.29(b).

Despite its simplicity, the quasi-static model is able to capture the basic non-linear mechanisms of self-sustained oscillations in a single reed instrument. Due to its compactness and low number of parameters, this model has been also used for sound synthesis purposes.

M-3.53

⁹ The Bernoulli law holds for incompressible non-viscous fluids in stationary conditions, and states the relation $u = A \cdot x \cdot \Delta p^{1/2} \text{sgn}(\Delta p)$ between the flow u and the pressure difference Δp through an aperture of width x . Some authors adopt for the single reed the generalized equation $u = [A \cdot x \Delta p^{1/2} \text{sgn}(\Delta p)]^{1/\alpha}$, with an experimentally determined value $\alpha = 3/2$.

Write a function that computes the pressure wave $p^+[n]$ reflected into the bore from the wave $p^-[n]$ arriving from the bore, according to the quasi-static model (3.114). Implement a quasi-static clarinet model in which the quasi-static reed is coupled to a waveguide cylindrical bore and driven by a mouth pressure signal p_m . The bell can be modeled as a low-pass filter, that radiates frequencies above its cut-off (typically around 1500 Hz) and reflects low frequencies back inside the bore.

M-3.53 Solution

Further refinements to this model should include propagation losses, fractional-delay filters in order to allow for fine tuning of the bore length, and acoustic holes modeled as scattering filters connected through 3-port junctions to the main waveguide structure.

3.6.3.3 Dynamic reeds

More refined reed models need to take into account the dynamics of the reed. A reasonably accurate description is obtained through a second-order mechanical oscillator, driven by the pressure drop Δp between mouth and mouthpiece:

$$\begin{cases} m\ddot{y}_L(t) + r\dot{y}_L(t) + k(y_L(t) - y_0) = S_d\Delta p(t), & y_L < y_{max}, \\ y_L(t) = y_m \quad \text{and} \quad \dot{y}_L(t) = 0, & y_L \geq y_{max}, \end{cases} \quad (3.115)$$

where m and r represent the reed mass and damping, while other parameters and variables are defined as before. The constant y_0 represents the reed displacement at rest.

This modeling approach is reasonable for the same reasons mentioned before: small reed dimensions compared to typical wavelengths in the resonator, reed oscillation mainly in the vertical direction, and high frequencies of the transversal reed modes (only the first mode is relevant). Note that in Eq. (3.115) the phenomenon of reed beating (i.e. complete closure of the reed) is here incorporated in the lumped model in a non-physical way, by imposing an ideal “stop” when the reed tip reaches its maximum allowed displacement y_m . Note also that the quasi static approximation examined in the previous section corresponds to approximating the transfer function of this system with its value at 0 frequency.

Another refinement amounts to taking into account an additional component affecting the total flow inside the instrument: the reed motion generates the flow $S_d\dot{y}_L(t)$, proportional to the reed tip velocity. If we now call u the flow inside the instrument and u_f the flow entering from the slit, these are related through the following equation:

$$u(t) = u_f(t) + u_r(t), \quad \text{with} \quad u_r(t) = S_d\dot{y}_L(t). \quad (3.116)$$

Incorporating this dynamics into the model results in more convincing sound synthesis, especially as far as transients are concerned. Realistic effects can be obtained, such as transitions to high regimes of oscillation. Both the resonance and the damping of the reed oscillator (3.115) and g play a role in helping transition to the second register of a single reed instrument. As an example, the clarion register in the clarinet plays a twelfth above the fundamental register and is usually obtained with the aid of a register hole. However the clarion register can be produced also without opening the register hole if the reed resonance matches a low harmonic of the playing frequency and the damping is small enough. Another playing regime in single reed instruments is the so-called reed regime (“squeaks”): this can be obtained by imparting an extremely low damping to the reed oscillator, so that the oscillation is governed by the reed resonance.

M-3.54

Implement a dynamic clarinet model in which the dynamic reed is coupled to a waveguide cylindrical bore and driven by a mouth pressure signal p_m .



This book is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 license,

©2005-2018 by the authors except for paragraphs labeled as adapted from <reference>

Finally, additional degrees of freedom must be taken into account when simulating other types of reeds. *Double reeds* (such as those found in oboe and bassoon) are composed of two reeds that oscillate independently, and even if one assumes perfect symmetry of oscillation the flow model differs from the one examined previously, due to the smallness of the aperture. In so-called *lip reeds* the role of the reed is taken by the performer's lips, that are constrained into the mouthpiece and vibrate at the fundamental frequency: at least two degrees of freedom are needed to simulate lip vibration.

3.6.4 Computability issues

We have examined in Sec. 3.5.1 the concept of delay-free loop in the case of linear systems, and have mentioned some strategies for dealing with it. However, more severe computability problems can arise when simulating non-linear elements.

3.6.4.1 Non-linear systems and delay-free loops

It should be clear that in the non-linear case one cannot perform a rearrangement such as in (3.79), because a non-linear equation is not always analytically invertible. The question is then how to deal with the delay-free loop problem in the non-linear case.

One can use an *explicit* numerical method, that produces a system of difference equations in which there are no delay-free loops. This choice solves the computational problem but can introduce more severe artifacts in the numerical system: explicit methods have lower orders of accuracy with respect to implicit methods, and more importantly are not unconditionally stable, i.e. are not stable for any sampling frequency F_s and for any values of the system parameters.

A rudimentary solution, that is nonetheless often encountered in the literature of physical modeling, amounts to inserting fictitious delay elements z^{-1} in the computational scheme. In practice this is a variant of the previous approach: instead of using an explicit method from the beginning, one makes the computation explicit *a posteriori*, through the insertion of delay elements. While this "trick" can be acceptable at significantly high sampling rates, the insertion of delay elements can again deteriorate the accuracy and stability properties of the numerical system. Even worse, in this case one cannot determine analytically the stability range of the system.

3.6.4.2 Iterative methods

Numerical analysis provides iterative methods to find solutions of non-linear systems of algebraic equations: examples of such methods include fixed-point iteration and Newton iteration, and each of them requires specific hypothesis on the non-linear system to hold.

Using an iterative solver is advantageous over the previous approaches in that one can exploit the accuracy and stability properties of an implicit method without introducing additional numerical errors in the system. One major drawback, however, is that one does not know in advance the number of iterations that are needed for the solver to converge to the solution $\mathbf{y}[n]$: this can be a problem for real-time applications, where one wants to know the time needed to compute one sound sample.

See [Fontana and Avanzini, 2008] for details about Newton-Raphson and fixed-point iteration for the simulation of non-linear systems.

3.6.4.3 Sheared non-linearities

In many practical cases the delay-free loop problem takes the form of the implicit dependence

$$\mathbf{y}[n] = \mathbf{f}(\tilde{\mathbf{x}}[n] + \mathbf{K}\mathbf{y}[n]), \quad (3.117)$$



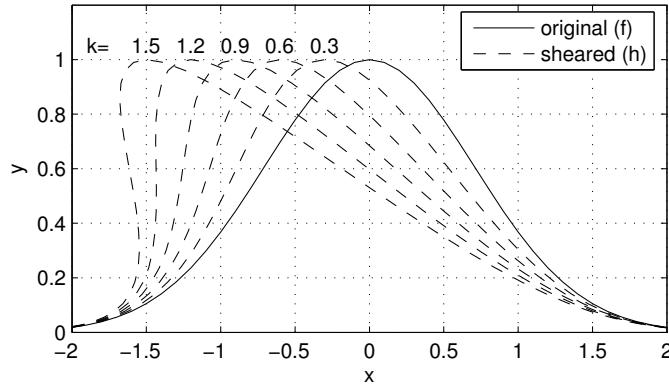


Figure 3.30: Shear transformation of $f(x) = e^{-x^2}$ for various k values.

where \mathbf{f} is a non-linear function, and $\tilde{\mathbf{x}}[n]$ is a vector of variables that are known at time n . The variables $\mathbf{y}[n]$ depend instantaneously onto themselves in the above equation. If one could turn this implicit dependence into a new explicit dependence $\mathbf{y}[n] = \mathbf{h}(\tilde{\mathbf{x}}[n])$, this would solve the delay-free loop problem.

This is achieved using the *implicit mapping theorem*. Define the function \mathbf{g} as

$$\mathbf{g}(\tilde{\mathbf{x}}, \mathbf{y}) = \mathbf{f}(\tilde{\mathbf{x}} + \mathbf{K}\mathbf{y}) - \mathbf{y}, \quad (3.118)$$

and assume that there is a point $(\tilde{\mathbf{x}}_0, \mathbf{y}_0)$ such that $\mathbf{g}(\tilde{\mathbf{x}}_0, \mathbf{y}_0) = 0$. Moreover, assume that the following condition holds

$$\det[\mathbf{J}_{\mathbf{y}}(\mathbf{g})(\tilde{\mathbf{x}}_0, \mathbf{y}_0)] = \det \left[\frac{g_i}{y_j}(\tilde{\mathbf{x}}_0, \mathbf{y}_0) \right]_{i,j} \neq 0, \quad (3.119)$$

where $\mathbf{J}_{\mathbf{y}}(\cdot)$ denotes the Jacobian matrix with respect to the \mathbf{y} variables. From the definition of \mathbf{g} , it is seen that $\mathbf{J}_{\mathbf{y}}(\mathbf{g}) = \mathbf{J}_{\mathbf{x}}(\mathbf{f})\mathbf{K} - \mathbf{I}$. Therefore, condition (3.119) implies that the matrix $[\mathbf{J}_{\mathbf{x}}(\mathbf{f})\mathbf{K} - \mathbf{I}]$ must be non-singular at the point $(\tilde{\mathbf{x}}_0, \mathbf{y}_0)$. If these conditions are fulfilled, then the implicit mapping theorem states that a function $\mathbf{h}(\tilde{\mathbf{x}})$ exists locally (i.e. for points $\tilde{\mathbf{x}}$ in a neighborhood of $\tilde{\mathbf{x}}_0$), with the properties

$$\mathbf{h}(\tilde{\mathbf{x}}_0) = \mathbf{y}_0 \quad \text{and} \quad \mathbf{g}(\tilde{\mathbf{x}}, \mathbf{h}(\tilde{\mathbf{x}})) = 0. \quad (3.120)$$

If the above conditions are fulfilled globally rather than in a neighborhood of $(\tilde{\mathbf{x}}_0, \mathbf{y}_0)$, then \mathbf{h} is defined globally.

A few geometrical considerations can help understanding the shape of the new function \mathbf{h} . Consider the coordinate transformation

$$\begin{bmatrix} \tilde{\mathbf{x}} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & -\mathbf{K} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}. \quad (3.121)$$

This defines a *shear* that leaves the \mathbf{y} axes unchanged and distorts the \mathbf{x} axis into the $\tilde{\mathbf{x}}$ axis. The plot of the function $\mathbf{y} = \mathbf{f}(\mathbf{x})$ “lives” in the (\mathbf{x}, \mathbf{y}) space. Then the plot of $\mathbf{y} = \mathbf{h}(\tilde{\mathbf{x}})$ is obtained by applying the coordinate transformation (3.121), and is therefore a sheared version of the former.

In order to understand this shear effect, consider the following example with a scalar function $f : \mathbb{R} \rightarrow \mathbb{R}$:

$$y[n] = f(x[n]) = e^{-(x[n]^2)}, \quad \text{with} \quad x[n] = \tilde{x}[n] + ky[n]. \quad (3.122)$$

Condition (3.119) translates in this case in the condition $f'(x) \neq 1/k$, which has a straightforward geometrical interpretation: the shear transformation defined in Eq. (3.121) is such that the vector $[x, y]^T =$



$[k, 1]^T$ (i.e. a point with tangent $1/k$) is transformed into the vector $[\tilde{x}, y]^T = [0, 1]^T$ (i.e. a point with vertical tangent). This explains why the derivative of f cannot equal $1/k$.

Figure 3.30 shows the original function $f(x)$, together with the sheared one $h(\tilde{x})$, for various k values. It can be seen that the horizontal coordinate is distorted when applying the shearing transformation. Moreover, note that for $k = 1.5$ the new function $h(\tilde{x})$ cannot be defined globally, because the condition $f'(x) \neq 1/k$ is not fulfilled globally in this case.

M-3.55

Simulate a modal oscillator excited by the non-linear impact force $f(x(t)) = kx(t)^\alpha$ (i.e. the impact model (3.106) with $\lambda = 0$) as follows: use an implicit numerical scheme (e.g. the bilinear transform), find the implicit dependence in the form (3.117), and construct the corresponding sheared non-linear function.

3.7 Commented bibliography

Sound modeling techniques can be classified according to many criteria. Two general references that address these issues are [De Poli, 1991, Smith III, 1991]. Specifically, the taxonomy based on *signal models* and *source models*, and their subclasses, proposed at the beginning of this chapter is based on [De Poli, 1991].

Seminal ideas that eventually lead to the definition of physically-based sound modeling techniques are to be found in research on musical instrument acoustics. Some classic papers in this area are [Hiller and Ruiz, 1971a,b, Schumacher, 1981, McIntyre et al., 1983]. In particular, the two citations at the beginning of the Introduction are taken from Hiller and Ruiz [1971a], McIntyre et al. [1983], respectively. A book that covers the topic of musical acoustics exhaustively is [Fletcher and Rossing, 1991]. In particular our discussion of the analogies between electrical, mechanical, and acoustic systems, given in Sec. 3.2.1 is based on an analogous discussion in [Fletcher and Rossing, 1991, Ch.1].

A general overview on approaches and techniques used in physical modeling, with an emphasis on structural and computational aspects, is provided by De Poli and Rocchesso [1998]. Figure 3.23 in this chapter (typical block scheme of a musical instrument model) is based on an analogous scheme in [De Poli and Rocchesso, 1998]. Two more recent and very complete tutorial papers on the topic of physical modeling are [Smith III, 2004] and [Välimäki et al., 2006].

About waveguide modeling approaches. The theory of 1-D waveguide models is now well established. An exhaustive introduction to the topic is given by Smith III [1998], who provides full derivations of waveguide structures and examples of musical instrument modeling, together with a vast bibliography. A more recent and even more exhaustive overview is given by the same author in [Smith III, 2008].

The basic principles of waveguide models were already present in the work of Kelly and Lochbaum [1962] on speech synthesis, where a so-called “transmission-line modeling” approach was used to simulate the human vocal tract through delay lines and scattering junctions. The definition of “digital waveguide modeling” was introduced later by Smith III [1985] in the context of musical applications, because of an analogy to the concept of waveguide that has been used, for example, in microwave technology. The Karplus-Strong algorithm, which we have regarded as the first step toward the development of digital waveguide structures, was originally proposed by Karplus and Strong [1983]. Fractional-delay filters: detailed discussion is provided by Laakso et al. [1996]. Modeling of dissipation and dispersion: Bank [2006] discusses the topic at length, with application to physically-based synthesis of the piano. In particular, the frequency-dependent dissipation model reported in Eq. (3.48) was first proposed by Bensa et al. [2003], although in the context of finite-difference simulations, as an improvement of the dissipation model by Hiller and Ruiz [1971a]. About waveguide junctions. Many textbooks on digital speech processing discuss multitube lossless models of the vocal tract, which are basically cylindrical waveguide sections connected by Kelly-Lochbaum junctions: see e.g. [Deller et al., 1993]. We have not addressed



the topic of higher dimensional (2- and 3-D) waveguide structures: seminal ideas were first presented by van Duyne and Smith III [1993].

About lumped modeling approaches. Numerical and computational aspects: most of the techniques described in Sec. 3.5.1 are found in DSP textbooks: see e.g. [Mitra, 2005]. In the field of numerical analysis, a comprehensive discussion on numerical methods for ordinary differential equations is given by Lambert [1993]. The example illustrated in Fig. 3.17 about delay-free computational paths in linear systems is adapted from [Mitra, 2005, Sec. 6.1.3, Fig. 6.5]. A classic reference to the theory of Wave Digital Filters (*WDF*) theory is [Fettweis, 1986].

Finite difference schemes have been applied to also to the explicit numerical simulation of partial differential equations, e.g. for modeling idiophones [Chaigne and Doutaut, 1997] and single reed systems [Stewart and Strong, 1980]. A recent book about the applications of finite difference methods to numerical sound synthesis is [Bilbao, 2009], which discusses the fundamentals of finite differences and shows how they can be employed to simulate strings, bars, plates, membranes, acoustic tubes. Among other lumped modeling approaches, in the early nineties Cadoz and coworkers have introduced the CORDIS-ANIMA model [Florens and Cadoz, 1991], which describes vibrating bodies as a set of interconnected mass-spring-damper cells.

Modal synthesis. A classic presentation of modal synthesis techniques is [Adrien, 1991]. Cook [1997] developed a series of “physically-informed” approaches to the modeling of percussion sounds, which are based on a modal description. The use of modal sound synthesis to virtual reality applications is discussed in [van den Doel and Pai, 2004]. A corpus of relevant contributions in this field has been provided by Rabenstein and coworkers [Trautmann and Rabenstein, 2003], who have proposed the so-called functional transformation method (FTM): in essence, the method exploits the existence of an analytical form of the modal parameters for a set of relevant multidimensional differential systems, including strings and membranes with various boundary conditions. Our examples of modal analysis for simple 1-D and 2-D shapes is based on [Fletcher and Rossing, 1991, Ch.2-3]. The same book also shows experimental results of modal analysis on several musical instruments, including modal shapes and Chladni patterns. In addition to linear prediction techniques and partial tracking methods, already discussed in Chapter *Sound modeling: signal based approaches*, a method for high-resolution estimate of modal parameters from sound analysis has been proposed in [Esquef et al., 2003].

About non-linear physical models. The non-linear impact model of Eq. (3.106) was first proposed by Hunt and Crossley [1975]. . Concerning stick-slip friction models, an overview of traditional models in the context of sound synthesis applications (bowed strings) is provided by Serafin [2004]. More complex dynamic stick-slip models, typically used in the literature of automatic control, have been recently applied to sound synthesis by Avanzini et al. [2005]. We have seen that the reed mechanism is that of pressure-controlled valves: a classic paper on the topic is [Fletcher, 1993]. The quasi-static single reed examined in Sec. 3.6.3 was first studied by Schumacher [1981] and has been used extensively in the literature. Other types of reeds: for the double reed see [Guillemain, 2004], for the lip reed see [Adachi and aki Sato, 1996]. Lip reeds have some similarities with vocal fold functioning: a classic example of a vocal fold model applied to voice synthesis is [Ishizaka and Flanagan, 1972].

We have seen that new problems are encountered when non-linear elements are present in the delay-free computational path: Borin et al. [2000] provides a discussion of these issues, together with a proposed non-iterative solution (in brief, a set of hypotheses and techniques to pre-compute a “sheared” non-linear function that makes the numerical scheme computable), and applications to the simulation of acoustic systems.



References

- Seiji Adachi and Masa aki Sato. Trumpet Sound Simulation Using a Two-dimensional Lip Vibration Model. *J. Acoust. Soc. Am.*, 99(2):1200–1209, Feb. 1996.
- Jean-Marie Adrien. The missing link: Modal synthesis. In Giovanni De Poli, Aldo Piccialli, and Curtis Roads, editors, *Representations of Musical Signals*, pages 269–297. MIT Press, Cambridge, MA, 1991.
- Federico Avanzini, Stefania Serafin, and Davide Rocchesso. Interactive simulation of rigid body interaction with friction-induced sound generation. *IEEE Trans. Speech Audio Process.*, 13(6):1073–1081, Nov. 2005.
- Balasz Bank. *Physics-based Sound Synthesis of String Instruments Including Geometric Nonlinearities*. PhD thesis, Budapest University of Technology and Economics, Dep. of Measurement and Information Systems, Budapest, 2006.
- Julien Bensa, Stefan Bilbao, Richard Kronland-Martinet, and Julius O. Smith III. The simulation of piano string vibration: From physical models to finite difference schemes and digital waveguides. *J. Acoust. Soc. Am.*, 114(2):1095–1107, Aug. 2003.
- Stefan Bilbao. *Numerical sound synthesis - Finite difference schemes and simulation in musical acoustics*. John Wiley & Sons, Chichester, 2009.
- Gianpaolo Borin, Giovanni De Poli, and Davide Rocchesso. Elimination of delay-free loops in discrete-time models of nonlinear acoustic systems. *IEEE Trans. Speech Audio Process.*, 8(5):597–606, Sep. 2000.
- Antoine Chaigne and Vincent Doutaut. Numerical Simulations of Xylophones. I. Time-domain Modeling of the Vibrating Bar. *J. Acoust. Soc. Am.*, 101(1):539–557, Jan. 1997.
- Perry R. Cook. Physically informed sonic modeling (PhISM): Synthesis of percussive sounds. *Computer Music J.*, 21(3): 38–49, 1997.
- Giovanni De Poli. A Tutorial on Digital Sound Synthesis Techniques. In Curtis Roads, editor, *The Music Machine*, pages 429–447. MIT Press, 1991.
- Giovanni De Poli and Davide Rocchesso. Physically Based Sound Modelling. *Organized Sound*, 3(1):61–76, Apr. 1998.
- John R Deller, John G. Proakis, and John. H.L. Hansen. *Discrete-Time Processing of Speech Signals*. Macmillan, New York, 1993.
- Paulo A. A. Esquef, Matti Karjalainen, and Vesa Välimäki. Frequency-zooming arma modeling for analysis of noisy string instrument tones. *EURASIP Journal on Applied Signal Processing*, 2003(10):953–967, 2003.
- Alfred Fettweis. Wave Digital Filters: Theory and Practice. *Proceedings of the IEEE*, 74(2):270–327, Feb. 1986.
- Neville H. Fletcher. Autonomous Vibration of Simple Pressure-Controlled Valves in Gas Flows. *J. Acoust. Soc. Am.*, 93(4): 2172–2180, Apr. 1993.
- Neville H. Fletcher and Thomas D. Rossing. *The physics of musical instruments*. Springer-Verlag, New York, 1991.
- Jean Luc Florens and Claude Cadoz. The physical model: modeling and simulating the instrumental universe. In Giovanni De Poli, Aldo Piccialli, and Curtis Roads, editors, *Representations of Musical Signals*, pages 227–268. MIT Press, Cambridge, MA, 1991.
- Federico Fontana and Federico Avanzini. Computation of delay-free nonlinear digital filter networks. Application to chaotic circuits and intracellular signal transduction. *IEEE Trans. Sig. Process.*, 56(10):4703–4715, Oct. 2008.
- Philippe Guillemin. A digital synthesis model of double-reed wind instruments. *EURASIP Journal on Applied Signal Processing*, 2004(1):990–1000, Jan. 2004.
- Lejaren Hiller and Paul Ruiz. Synthesizing Musical Sounds by Solving the Wave Equation for Vibrating Objects: Part I. *J. Audio Eng. Soc.*, 19(6):462–470, June 1971a.
- Lejaren Hiller and Paul Ruiz. Synthesizing Musical Sounds by Solving the Wave Equation for Vibrating Objects: Part II. *J. Audio Eng. Soc.*, 19(7):542–551, July 1971b.



- Kenneth H. Hunt and F. R. Erskine Crossley. Coefficient of restitution interpreted as damping in vibroimpact. *ASME J. Applied Mech.*, 42:440–445, June 1975.
- Kenzo Ishizaka and James L. Flanagan. Synthesis of voiced sounds from a two-mass model of the vocal cords. *Bell Syst. Tech. J.*, 51:1233–1268, 1972.
- Kevin Karplus and Alexander Strong. Digital Synthesis of Plucked String and Drum Timbres. *Computer Music J.*, 7(2):43–55, 1983.
- John L. Kelly and Carol C. Lochbaum. Speech synthesis. In *Proc. 4th Int. Congr. Acoustics*, pages 1–4, Copenhagen, Sep. 1962.
- Timo I. Laakso, Vesa Välimäki, Matti Karjalainen, and Unto K. Laine. Splitting the Unit Delay Tools for Fractional Delay Filter Design. *IEEE Signal Processing Magazine*, 13(1):30–60, Jan. 1996.
- John D. Lambert. *Numerical Methods for Ordinary Differential Systems*. John Wiley & Sons, 1993.
- Michael E. McIntyre, Robert T. Schumacher, and James Woodhouse. On the Oscillations of Musical Instruments. *J. Acoust. Soc. Am.*, 74(5):1325–1345, Nov. 1983.
- Sanjit K Mitra. *Digital Signal Processing*. McGraw-Hill, third edition, 2005.
- Robert T. Schumacher. *Ab Initio* Calculations of the Oscillations of a Clarinet. *Acustica*, 48(2):71–85, 1981.
- Stefania Serafin. *The sound of friction: real-time models, playability and musical applications*. PhD thesis, Stanford University, Center for Computer Research in Music and Acoustics, Stanford, 2004.
- Julius O. Smith III. A new approach to digital reverberation using closed waveguide networks. In *Proc. Int. Computer Music Conf. (ICMC'85)*, pages 47–53, Vancouver, 1985.
- Julius O. Smith III. Viewpoints on the History of Digital Synthesis. In *Proc. Int. Computer Music Conf. (ICMC'91)*, pages 1–10, Montreal, Oct. 1991.
- Julius O. Smith III. Principles of digital waveguide models of musical instruments. In Mark Kahrs and Karl-Heinz Brandenburg, editors, *Applications of Digital Signal Processing to Audio and Acoustics*, pages 417–466. Kluwer Academic Publishers, New York, Mar. 1998.
- Julius O. Smith III. Virtual acoustic musical instruments: Review and update. *Journal of New Music Research*, 33(3):283–304, Autumn 2004.
- Julius O. Smith III. *Physical Audio Signal Processing: for Virtual Musical Instruments and Digital Audio Effects, December 2008 Edition*. <http://ccrma.stanford.edu/~jos/pasp/>, 2008. Accessed 15/12/2008.
- Stephen E. Stewart and William J. Strong. Functional Model of a Simplified Clarinet. *J. Acoust. Soc. Am.*, 68(1):109–120, July 1980.
- Lutz Trautmann and Rudolf Rabenstein. *Digital Sound Synthesis by Physical Modeling Using the Functional Transformation Method*. Kluwer Academic, New York, 2003.
- Vesa Välimäki, Jyri Pakarinen, Cumhur Erkut, and Matti Karjalainen. Discrete-time modelling of musical instruments. *Rep. Prog. Phys.*, 69(1):1–78, 2006.
- Kees van den Doel and Dinesh K. Pai. Modal Synthesis for Vibrating Objects. In Ken Greenebaum, editor, *Audio Anecdotes*. AK Peters, Natick, MA, 2004.
- S. A. van Duyne and J. O. Smith III. The 2-D Digital Waveguide Mesh. In *Proc. IEEE Workshop on Applications of Sig. Process. to Audio and Acoustics (WASPAA'93)*, pages 177–180, New Paltz (NY), Oct. 1993.



Chapter 4

Sound in space

Federico Avanzini

Copyright © 2005-2018 Federico Avanzini

except for paragraphs labeled as *adapted from <reference>*

This book is licensed under the CreativeCommons Attribution-NonCommercial-ShareAlike 3.0 license. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/>, or send a letter to Creative Commons, 171 2nd Street, Suite 300, San Francisco, California, 94105, USA.

4.1 Introduction

If we think at the process of sound production in the light of the classic source-medium-receiver model of communication theory, we can say that in Chapters *Sound modeling: signal based approaches* and *Sound modeling: source based approaches* we have studied models for the source of sound signals. We now move a step further and examine the effects of the *medium* in which sound propagates, and the *receiver*, specifically a human receiver with two ears.

One of the most frequently effects produced during sound propagation in a medium is reverberation, which is caused by physical surfaces that partly absorb and partly reflect sound waves in air. We will first examine in Sec. 4.2 the physical and perceptual background of reverberation. The knowledge gained on these aspects will enable us to study some of the most known reverberation algorithms in Sec. 4.3. Finally we will review in Sec. 4.4 more recent approaches to synthetic reverberation, that are based on feedback delay networks and waveguide meshes.

A similar path will be followed in examining the receiver block. We will first examine in Sec. 4.5 how and to what extent a human receiver with two ears can gain information about the incoming direction and distance of an emitted sound, and what are the most relevant perceptual effects involved in *spatial hearing*. Armed with this knowledge we will address in Sec. 4.6 the most popular *3-D sound* processing techniques by which a virtual sound source can be positioned in some point of the space around a listener. We will in particular focus on *binaural techniques*, which assume that two independent sound signals are delivered to the two ears, e.g. through headphones.

4.2 Reverberation: physical and perceptual background

Almost any sound of our everyday life is produced in a reverberant environment, be it the office at work, the living room, or a concert hall. An emitted sound is therefore always accompanied by delayed versions, caused by reflecting surfaces and coming from many different directions. We talk about reverberation when the reflections occur soon after the emitted sound, so that they are not perceived as separate sound events, and instead have the effect of “coloring” the original sound and modifying its spatial characteristics. In this section we first review the physical process of reverberation, then we examine the most perceptually salient characteristics of reverberation. Having knowledge of both these aspects are essential in order to develop algorithms for synthetic reverberation.

4.2.1 Basics of room acoustics

For our purposes a room is a physical enclosure that contains an elastic medium (generally, air) through which acoustic disturbances can be propagated. It also has a boundary (the room walls) that limit the propagation of these acoustic disturbances. In this view a room is simply an acoustic resonator, similar to the string that we have examined in Chapter *Sound modeling: source based approaches*, but with at least two important differences: first, it is a 3-D resonator, because sound can propagate in all spatial directions, and second, its physical dimensions are much larger than typical dimensions of a string in a musical instrument. Put in another way, its physical dimensions are much larger than typical acoustic wavelengths.

4.2.1.1 Sound waves in a closed space

We have analyzed in Chapter *Sound modeling: source based approaches* the D'Alembert equation which describes sound propagation within a perfectly elastic medium. While the 1-D D'Alembert equation can be used to model strings or acoustic tubes, the 3-D equation describes sound propagation in space:

$$\nabla^2 p(\mathbf{x}, t) = \frac{1}{c^2} \frac{\partial^2 p}{\partial t^2}(\mathbf{x}, t), \quad (4.1)$$

where \mathbf{x} represents Euclidean coordinates in space and p is the acoustic pressure. The symbol $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$ stands for the 3-dimensional Laplacian operator. As opposed to mechanical vibrations in a string or membrane, acoustic vibrations are *longitudinal* rather than transversal, i.e. the air particles are displaced in the same direction of the wave propagation. The constant c has the dimensions m/s of a velocity and indeed is sound velocity in air.

By adding suitable boundary conditions we can gain a description of waves of particle velocity within a three-dimensional enclosure. Let us start with the simplest possible 3-D enclosure, a rectangular room with perfectly smooth and rigid walls. More precisely, we define the domain \mathcal{D} of the problem to be a parallelepiped with edges of length L_x, L_y, L_z :

$$\mathcal{D} = \{\mathbf{x} = (x, y, z); 0 \leq x \leq L_x, 0 \leq y \leq L_y, 0 \leq z \leq L_z\} \quad (4.2)$$

Let \mathcal{B} be the boundary of \mathcal{D} , i.e. the rigid walls of the parallelepiped. The boundary conditions require the air velocity perpendicular to each wall to be zero on \mathcal{B} . Equivalently, if we consider acoustic pressure p then the conditions on the boundary are $\partial p / \partial \mathbf{x}(\mathcal{B}) = 0$. Then one can provide an analytical solution of Eq. (4.1) in terms of stationary modes of the kind

$$p(\mathbf{x}, t) = s(\mathbf{x})q(t) \quad (4.3)$$



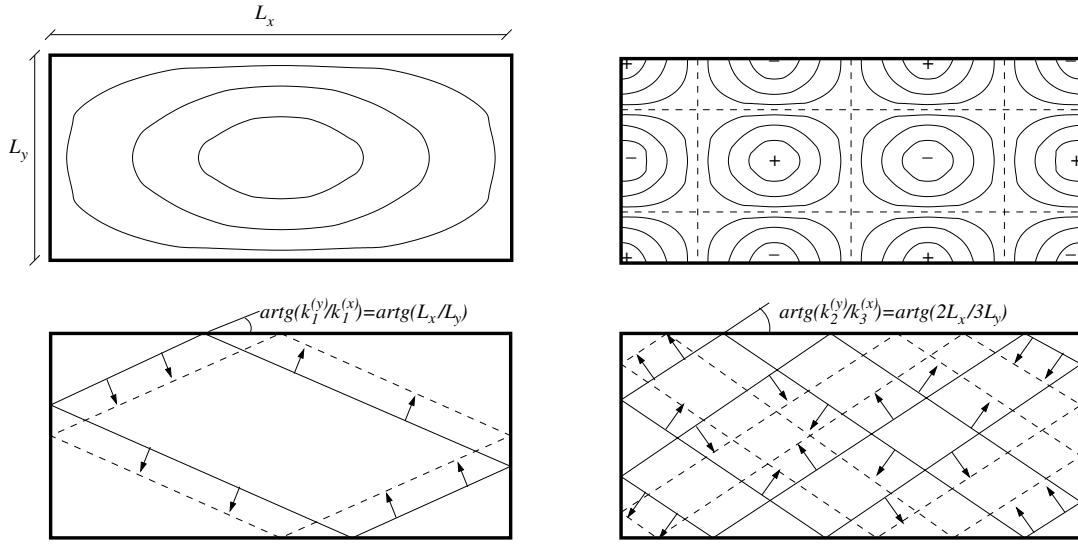


Figure 4.1: Plane wave loops $(1, 1, 0)$ and $(3, 2, 0)$, as seen on the (x, y) plane.

Following a line of reasoning analogous to the 1-D case, one can determine the spatial shape $s(\mathbf{x})$ of the modes as

$$s_{n,m,l}(\mathbf{x}) = \sqrt{\frac{2}{L_x}} \sqrt{\frac{2}{L_y}} \sqrt{\frac{2}{L_z}} \cos(k_n^{(x)} x) \cos(k_m^{(y)} y) \cos(k_l^{(z)} z), \quad (4.4)$$

where we can define the *wavenumbers* $\mathbf{k}_{n,m,l}$ as

$$\mathbf{k}_{n,m,l} = (k_n^{(x)}, k_m^{(y)}, k_l^{(z)}) \quad \text{with} \quad k_n^{(x)} = \frac{n\pi}{L_x}, \quad k_m^{(y)} = \frac{m\pi}{L_y}, \quad k_l^{(z)} = \frac{l\pi}{L_z}. \quad (4.5)$$

Analogously to the 1-D case discussed for modal synthesis in *Sound modeling: source based approaches*, these functions are a orthonormal basis for the space $L^2(\mathcal{D})$. The temporal part is subsequently derived as

$$q_{n,m,l}(t) = \cos(\omega_{n,m,l} t + \phi_{n,m,l}), \quad \text{with} \quad \omega_{n,m,l} = c \sqrt{\left[k_n^{(x)} \right]^2 + \left[k_m^{(y)} \right]^2 + \left[k_l^{(z)} \right]^2}. \quad (4.6)$$

Differently from the 1-D case, and analogously to the 2-D case of the rectangular membrane, the frequencies $\omega_{n,m,l}$ are a non-harmonic series. However each of the three spatial directions (where only one of the three indexes (n, m, l) is varying) is associated to a harmonic subseries. Analogously to the 1-D case a mode (n, m, l) has nodal surfaces, which corresponds to the regions where $s_{n,m,l}(\mathbf{x}) = 0$. It is easy to see that these are planes parallel to the walls of the room.

A normal mode $p_{n,m,l}(\mathbf{x}, t) = s_{n,m,l}(\mathbf{x}) q_{n,m,l}(t)$ can be written as a superposition of waves traveling in different directions. This can be seen through multiple application of Werner formulas¹, which yields

$$p_{n,m,l}(\mathbf{x}, t) = \dots = \sqrt{\frac{2}{L_x}} \sqrt{\frac{2}{L_y}} \sqrt{\frac{2}{L_z}} \sum \cos \left[\mathbf{k}_{n,m,l}^{\pm\pm\pm} \cdot \mathbf{x} \pm (\omega_{n,m,l} t + \phi_{n,m,l}) \right], \quad (4.7)$$

where we have defined $\mathbf{k}_{n,m,l}^{\pm\pm\pm} = (\pm k_n^{(x)}, \pm k_m^{(y)}, \pm k_l^{(z)})$, and where the summation has to be extended over the sixteen possible combinations of signs in the argument. This means that for each mode there are eight directions of wave propagation, each one associated to one $\mathbf{k}_{n,m,l}^{\pm\pm\pm}$ vector. Figure 4.1 visualizes the wavefronts for the modes $(1, 1, 0)$ and $(3, 2, 0)$: these result in plane wave loops having constant length.

¹ $2 \cos \alpha \cos \beta = \cos(\alpha - \beta) + \cos(\alpha + \beta)$.

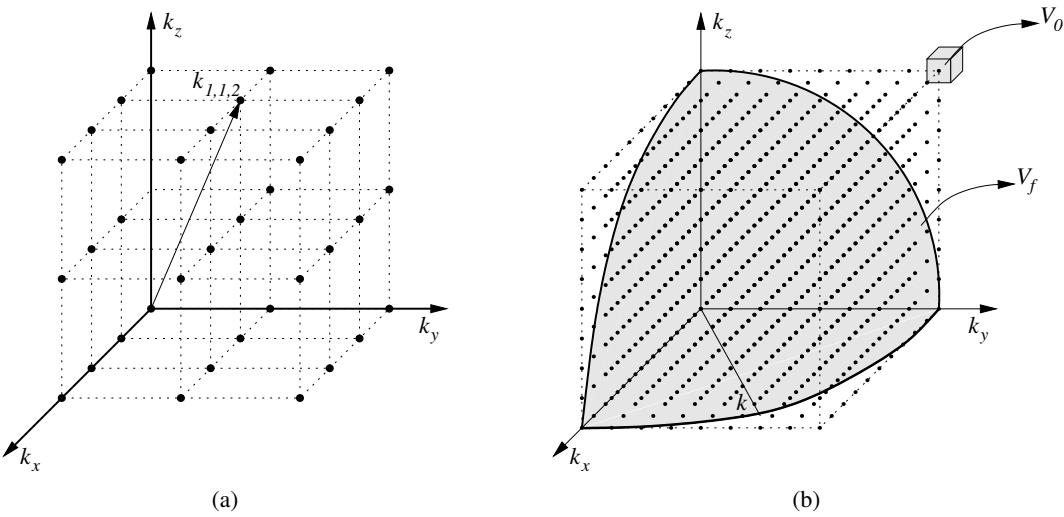


Figure 4.2: Estimation of modal density; (a) distribution of wavenumbers on a regular point lattice, (b) estimation of the amount of wavenumbers contained in a spherical octant of radius k .

4.2.1.2 Modal density

We now want to derive an estimate of the *modal density*, i.e. the average density of eigenfrequencies on the frequency axis.

From Eq. (4.5) one observes that the allowed values for the wave numbers k are distributed on a regular point lattice in the 3-D space depicted in Fig. 4.2(a). The number N_f of eigenfrequencies in the frequency range from 0 to f equals the number of lattice points contained in the sphere octant of radius $k = c \cdot 2\pi f$ depicted in Fig. 4.2(b). Therefore, $N_f = V_f/V_0$, where V_f is the volume of the sphere octant of radius k and V_0 is the average volume per lattice point. The former is one octave of the sphere volume, $V_f = \pi k^3/6$, while the latter can be estimated as the volume of the cube depicted in Fig. 4.2(b), whose edges have lengths $\pi/L_x, \pi/L_y, \pi/L_z$, respectively (recall Eq. (4.5) for the wavenumbers $k_{n,m,l}$). Therefore $V_0 = \pi^3/V$, where $V = L_x L_y L_z$ is the room volume. One finally obtains

$$N_f = \frac{\pi k^3/6}{\pi^3/V} = \frac{4\pi}{3} V \left(\frac{f}{c} \right)^3. \quad (4.8)$$

The modal density is estimated as the derivative of N_f with respect to frequency:

$$D_f(f) = \frac{dN_f}{df} = \frac{4\pi V}{c^3} f^2 \quad (4.9)$$

In order to gain a quantitative understanding of these equations, let us consider a hypothetical medium-small auditorium with dimensions $(L_x, L_y, L_z) = (35, 20, 14)$ meters, which means $V = 9800 \text{ m}^3$. From Eq. (4.8) we see that there are approximately 10^9 normal modes with frequencies between 0 and 10 kHz. From Eq. (4.9) we see that at 1 kHz the modal density per Hz is approximately 3500, which means that the average spacing between modes is less than 3×10^{-4} Hz.

4.2.1.3 Sound sources and room impulse responses

Let us now move from the mathematical analysis sketched in the previous sections towards a more realistic situation. First, we assume that a sound source is located within the domain \mathcal{D} . The distribution

in space of the source is described by a continuous density function $\bar{f}(\mathbf{x})$, while the time-domain signal emitted by the source is described by a function $\bar{q}(t)$: this means that $\bar{q}(t) \cdot \bar{f}(\mathbf{x})dV$ is the volume velocity of a volume element dV at time t .

As a second hypothesis, we consider complex, non-ideal, boundary conditions in which walls are not perfectly rigid and instead absorption occurs. This can be restated by assuming that the normal modes have now complex eigenvalues $k_{n,m,l}$:

$$k_{n,m,l} = \omega_{n,m,l}/c + j\delta_{n,m,l}/c, \quad \delta_{n,m,l} \ll \omega_{n,m,l}. \quad (4.10)$$

We want to find the solution of the wave equation in \mathcal{D} under these two hypotheses. The wave equation in the presence of a sound source can be written as

$$\nabla^2 p(\mathbf{x}, t) = \frac{1}{c^2} \frac{\partial^2 p}{\partial t^2}(\mathbf{x}, t) - \rho_{air} \bar{f}(\mathbf{x}) \frac{d\bar{q}}{dt}(t). \quad (4.11)$$

Since the $f_{n,m,l}$ functions of Eq. (4.4) are a basis for $L^2(\mathcal{D})$, we can project both the source density function \bar{f} and the solution p of Eq. (4.11) on this basis:

$$\bar{f}(\mathbf{x}) = \sum_{n,m,l} \bar{F}_{n,m,l} f_{n,m,l}(\mathbf{x}), \quad P(\mathbf{x}, s) = \sum_{n,m,l} P_{n,m,l}(s) f_{n,m,l}(\mathbf{x}). \quad (4.12)$$

Note that in the second of the above equations we have implicitly assumed to work in the Laplace domain instead of the time domain. If we can find the unknown coefficients $P_{n,m,l}(s)$ as functions of the known coefficients $\bar{F}_{n,m,l}$, then we have the solution $P(\mathbf{x}, s)$ or equivalently $p(\mathbf{x}, t)$. If one inserts both series into Eq. (4.11) the result is

$$P_{n,m,l}(s) = s\rho_{air}c^2 Q(s) \frac{\bar{F}_{n,m,l}}{s^2 + c^2 k_{n,m,l}^2}. \quad (4.13)$$

We can find a solution $P(\mathbf{x}, s)$ if we consider the special case of a point source located at a certain point \mathbf{x}_0 of the room and emitting an impulsive sound signal. Under this assumption one has $\bar{f}(\mathbf{x}) = \delta_D(\mathbf{x} - \mathbf{x}_0)$, where the function $\delta_D(\cdot)$ is the Dirac delta. This implies that the coefficients $\bar{F}_{n,m,l}$ are in this case $\bar{F}_{n,m,l} = f_{n,m,l}(\mathbf{x}_0)$. Moreover, if the sound source is emitting an impulse $\bar{q}(t) = \delta(t)$, then the corresponding spectrum is $Q(s) = 1$. If one substitutes the coefficients (4.13) into the second of Eqs. (4.12), the result is

$$P(\mathbf{x}, s) := H_{x_0,x}(s) = s\rho_{air}c^2 \sum_{n,m,l} \frac{f_{n,m,l}(\mathbf{x}) f_{n,m,l}(\mathbf{x}_0)}{s^2 + c^2 k_{n,m,l}^2}. \quad (4.14)$$

This is the acoustic pressure generated in \mathbf{x} by a point source located at \mathbf{x}_0 and emitting an impulse. If we take the inverse Laplace transform, $h_{x_0,x}(t) = \mathcal{L}^{-1}\{H_{x_0,x}\}(t)$, this is what we call a *Room Impulse Response (RIR)*, measured at point \mathbf{x} after an impulse emitted in \mathbf{x}_0 . Equation (4.14) is telling us that the RIR is a superposition of numerous second-order resonant systems, each with center frequency very close to $\omega_{n,m,l}$ and damping constant very close to $\delta_{n,m,l}$:

$$h_{x_0,x}(t) = \begin{cases} 0 & t < 0 \\ \sum_{n,m,l} A_{n,m,l}(\mathbf{x}_0, \mathbf{x}) e^{-\delta'_{n,m,l} t} \cos(\omega'_{n,m,l} t + \phi_{n,m,l}) & t \geq 0 \end{cases} \quad (4.15)$$

The function $h_{x_0,x}(t)$ completely describes the room response for a source in \mathbf{x}_0 and a receiver in \mathbf{x} : if the emitted sound is not an impulse but a generic signal $\bar{q}(t)$, then the response will be –as usual– the convolution of the signal with the impulse response: $s(t) = [\bar{q} * h_{x_0,x}](t)$.



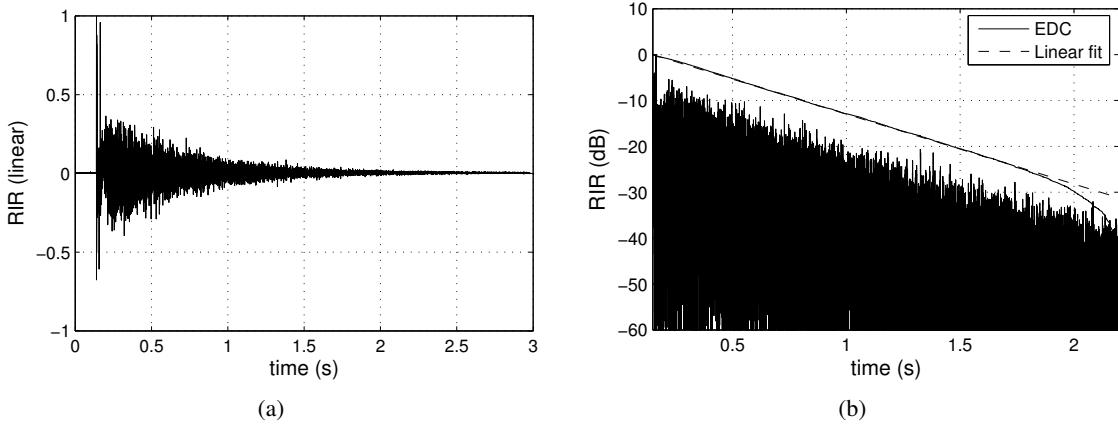


Figure 4.3: Room Impulse response and reverberation time: (a) RIR of a very reverberant environment (a cathedral); (b) a portion of the same RIR in dB, together with its EDC and a linear fit.

Now in normal rooms damping constants typically lie between 1 and 20 Hz: this justifies the assumption of very small δ coefficients in Eq. (4.10), and moreover tells that half-widths of these resonant systems are of the order of 1 Hz. If we compare this finding to the modal density estimate given in Eq. (4.9), we see that the average spacing of eigenfrequencies is smaller by several orders of magnitude than half-widths. Therefore each single resonant peak always covers many others and it is practically impossible to excite a single room resonance e.g. with a sinusoidal signal.

4.2.1.4 Reverberation time

From Eq. (4.15) we see that room reverberation adds a decaying tail to a source signal. One of the most important parameters derived from this equation is the *reverberation time* T_r , which is defined as the time required for the sound pressure to decay 60 dB. Clearly T_r is related to the absorption coefficients $\delta_{n,m,l}$. An approximate description of T_r can be derived as follows.

Given a source signal $\bar{q}(t)$ in x_0 , the resulting room response $s_x(t)$ at a point x will have the form

$$p(x, t) = [\bar{q} * h_{x_0, x}](t) = \dots = \sum_{n,m,l} c_{n,m,l} e^{-\delta'_{n,m,l} t} \cos(\omega'_{n,m,l} t + \psi_{n,m,l}) = \sum_{n,m,l} c_{n,m,l} s_{n,m,l}(t), \quad (4.16)$$

where the $c_{n,m,l}$'s and the $\psi_{n,m,l}$'s will vary depending on the signal \bar{q} , and where we have introduced the notation $s_{n,m,l}(t) = e^{-\delta'_{n,m,l} t} \cos(\omega'_{n,m,l} t + \psi_{n,m,l})$ for brevity. The energy density of the response (or actually a quantity proportional to the energy density) is obtained by squaring $s(t)$:

$$w(t) = [s(t)]^2 = \sum_{n,m,l} \sum_{n',m',l'} s_{n,m,l}(t) s_{n',m',l'}(t). \quad (4.17)$$

We can derive an estimate of how $w(t)$ decays by averaging $w(t)$ over time and exploiting the circumstance that the exponential terms vary slowly (as the δ 's are small). By averaging the cosine products only, the products with $(n, m, l) \neq (n', m', l')$ cancel on the average, and the products with $(n, m, l) = (n', m', l')$ give a value 1/2. If one makes the further assumption of nearly uniform damping, i.e. $\delta_{n,m,l} \sim \delta_0$, then we obtain the following result:

$$\langle w(t) \rangle = \sum_{n,m,l} c_{n,m,l}^2 e^{-2\delta_{n,m,l} t} \sim e^{-2\delta_0 t} \sum_{n,m,l} c_{n,m,l}^2. \quad (4.18)$$

This equation tells that for uniform damping the energy of the reverberation tail decays exponentially. In particular the reverberation time T_r is in this case derived as

$$-60 = 10 \log \left(e^{-2\delta_0 T_r} \right), \quad \Rightarrow T_r = \frac{6.91}{\delta_0}. \quad (4.19)$$

In general one cannot assume uniform damping, and as a consequence T_r is a function of frequency. However, the reverberation level falls in many practical cases in a fairly exponential fashion and therefore an overall reverberation time T_r can be defined and measured. Figure 4.3(a) shows a RIR measured in a very reverberant environment, a cathedral. Note that, apart from the initial spikes, the overall decay is fairly exponential.

The accuracy with which T_r can be determined directly from RIR signals is in general severely limited by random fluctuations in the decay curves, which result from mutual beating of normal modes of different frequencies at the moment the excitation signal ceases. Instead T_r is more reliably estimated by looking at another function, the *Energy Decay Curve* (*EDC*, also called the Schroeder integral for historical reasons), defined as follows:

$$EDC(t) = \int_t^\infty h^2(\tau) d\tau, \quad (4.20)$$

where $h(t)$ is a RIR. The value $EDC(t)$ provides a measure of the reverberation energy that is left in the RIR at time t . The advantage is that this function has a much more regular behavior than $h(t)$, therefore T_r can be determined by fitting the decay of $EDC(t)$ through linear regression (on a dB scale), and looking at the time needed for this linear fit to drop by 60 dB. Figure 4.3(b) shows this procedure applied to the RIR of Fig. 4.3(a). From the linearly fitted *EDC* one can see that T_r is in this case close to 4 s, which is a quite large value as one would expect in a cathedral.

M-4.56

Write a function that computes the reverberation time T_r given a signal representing a RIR.

M-4.56 Solution

```
function [Tr,edc] = revtime(rir,Fs,ti,tf);
%Returns an estimate of Tr and of the edc; rir is a row vector representing
%a RIR, ti and tf are the initial and final times on which edc is computed

rir=rir(round(ti*Fs):round(tf*Fs)); % chunk the RIR on the interval [ti,tf]
edc = 10*log10(fliplr(cumsum(fliplr(rir.^2)))); % compute EDC
edc = edc-max(edc); %normalize at 0 dB

%linearly fit edc in the first half and find Tr as
%the instant where the linear fit drops below -60dB
c = polyfit(1:round(length(edc)/2),edc(1:round(length(edc)/2)),1);
Tr = (-60-c(2))/(c(1)*Fs); % edc=c2 + c1*Fs*t; edc=-60 => this formula
```

The choice of the initial and final instants is critical: they have to cover the range after the initial impulse and where the decay is almost linear. Moreover we have decided to estimate T_r on the first half of the EDC curve only, in order to avoid the error of the last EDC samples. Of course many techniques exist to choose these parameters automatically, this is just a toy example.



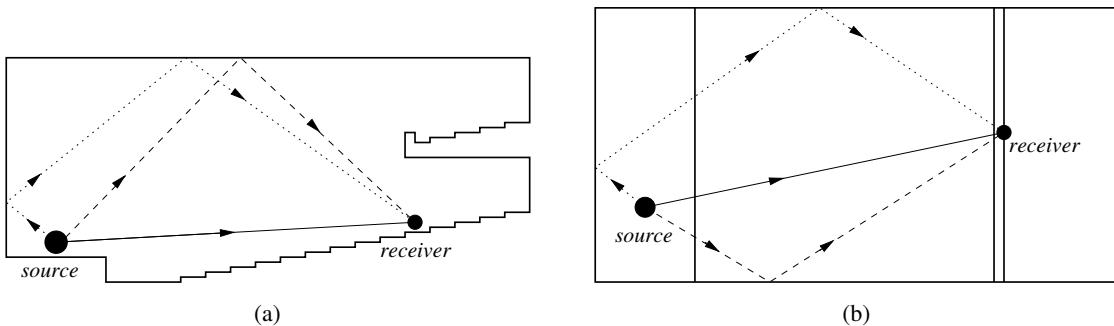


Figure 4.4: Acoustic rays from a source to a receiver (a) in a vertical room section and (b) in a horizontal room section. Solid lines represent the direct sound, dashed lines represent first-order reflections, dotted lines represent second-order reflections.

4.2.1.5 Geometrical room acoustics

Few results of practical use are obtained from direct manipulation of the D'Alembert equation, as we did in the previous sections. This is especially true when we consider rooms of arbitrary shapes instead of parallelepipeds: in that case even the computation of a single normal mode can become extremely difficult. An alternative description can be employed if we consider extremely high acoustic frequencies. In this limit, the concept of sound waves can be replaced by the concept of *acoustic rays*. By sound ray, we mean a vanishingly small portion of a spherical wave emitted by a point source in a room. This ray has well-defined direction and velocity of propagation, and conveys a total energy which remains constant (provided that it propagates within an ideal medium with no losses).

This simplified description based on acoustic rays takes the name of *geometrical acoustics* and has strict similarities with geometrical optics, although typical wavelengths and propagation velocities are very different in the two cases. Note that the assumption of extremely high frequencies is practically met in many cases of interest in room acoustics: a frequency of 1 kHz corresponds to a wavelength of approximately 34 cm, which is one or two orders of magnitude smaller than typical linear dimensions of rooms, as well as typical distances traveled by sound waves in a room.

Similarly to an optic ray, an acoustic ray that strikes a plane surface is reflected according to the following principles: (a) the reflected ray remains in the plane identified by the incident ray and the normal to the surface, and (b) the angles of the incident and reflected rays with the normal are equal. Figure 4.4 shows a room with a non-trivial shape (something like an auditorium), in which we have positioned a sound source and a receiver. All the paths from the source to the receiver can be characterized according to the number of reflections involved. The single source-receiver path with 0 reflections is the *direct sound*, and is followed by a small number of *first-order reflections* that involve one reflection on the room boundary, a larger number of *second-order reflections* that involve two reflections, and so on. In Fig. 4.4 we have drawn two examples of first- and second-order reflections.

Geometrical room acoustics can be used to provide a qualitative description of a RIR. Assume that an ideal impulse shot from a point source reaches a receiver at time $t = 0$. Each reflected ray will then arrive with a certain time delay and also with a certain attenuation, which depends on the path length (absorption in the medium) and on the number of reflections (wall absorption). The first reflections are strong and sporadic, but the temporal density of reflections increases rapidly while the average reflection energy decays accordingly. A qualitative reflection diagram is given in Fig. 4.5. Except for the first few isolated reflections, the weaker and denser reflections arriving at later times merge into a unitary percept.

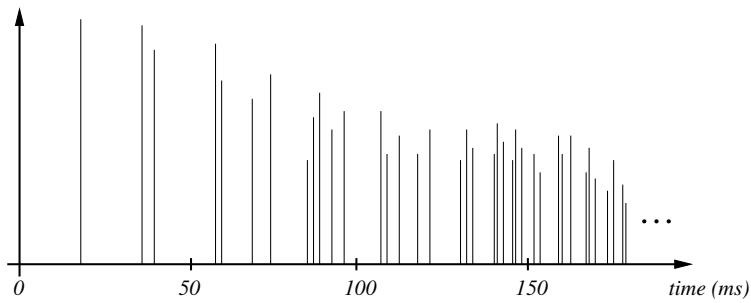


Figure 4.5: Schematic room response to an ideal impulse: the time axis is relative to the direct sound, which reaches the receiver at $t = 0$.

This description of room reverberation as a temporal sum of reflected rays is complementary to the view of reverberation as the sum of free decaying normal modes.

We now want to derive an estimate of the temporal structure of reflections. To this end we employ the usual prototype room, i.e. the parallelepiped, and we introduce the concept of *image sources*. If the reflecting surface is a plane the reflection of a sound ray can be simulated by constructing an *image source*. This process is illustrated in Fig. 4.6(a): given a sound source A and a receiver B , the path of a reflected ray r from the wall to B is the same path of the direct ray r' emitted by the *image source* A' . The process can be iterated in order to take into account higher-order reflections, and results in the construction of a grid of image sources that replace the walls altogether.

Now suppose that at time $t = 0$ all the sources emit an impulse. During the time interval from t to $t + dt$, the impulses that reach a receiver in the center of the room are those emitted by image sources whose distance from the receiver lies between ct and $c(t + dt)$. These sources are located within the spherical shell with radius ct , thickness cdt , and volume $4\pi c^3 t^2 dt$ illustrated in Fig. 4.6(b). Therefore the volume V of an image room is contained $4\pi c^3 t^2 dt/V$ times in the spherical shell, and for t large enough (i.e. for high reflection densities) this number coincides with the number dN_r of image sources contained in the shell. The temporal density of reflections arriving at time t is then

$$D_r(t) = \frac{dN_r}{dt}(t) = 4\pi \frac{c^3 t^2}{V}. \quad (4.21)$$

One could show that this result applies not only to a parallelepiped but to rooms of arbitrary shapes.

4.2.2 Perceptual reverberation parameters

In the previous section we have analyzed reverberation from a purely physical point of view. However in many applications it is important to correlate physical measurements to subjective judgements of acoustical quality, obtained from psychophysical experimentation. This is especially true in the domain of concert hall acoustics, where researchers have tried to isolate the objective parameters that are most relevant in determining the perception of acoustical quality of a hall. Subjective attributes are typically derived from perceptual experiments with musicians and listeners, who answer to detailed interviews, and subsequent comparison of the results with measured objective parameters.

In this section we enter, for the time in this book, the domain of psychoacoustics, and review some of the subjective attributes most commonly used in establishing the acoustical quality of reverberant environments. The literature on this topic is vast and the terminology is not always fully consistent, therefore we try to cluster together similar or equivalent concepts whenever possible.



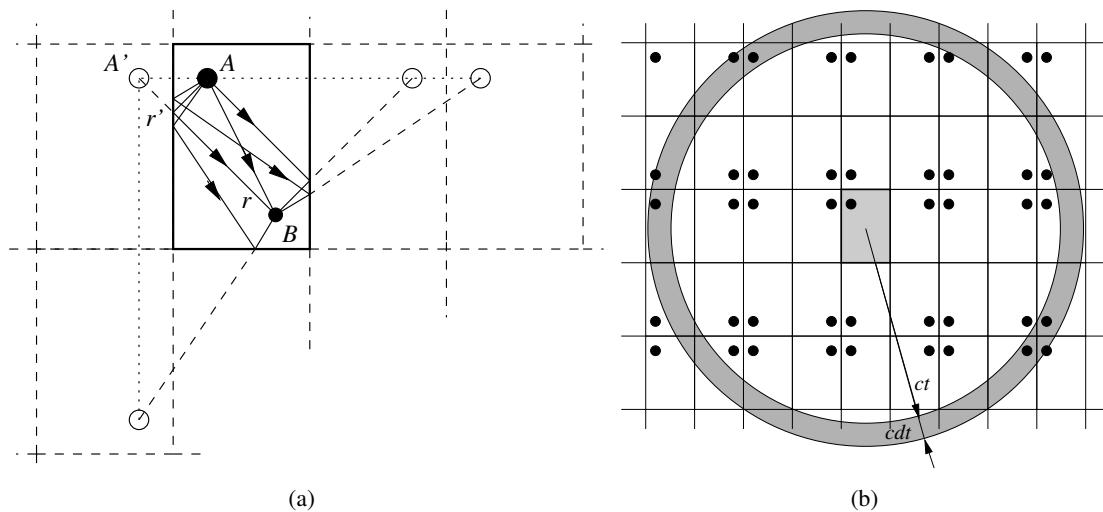


Figure 4.6: Estimation of temporal reflection density through the image source method; (a) construction of two first-order and two second-order reflections, and (b) estimation of acoustic rays reaching a receiver within the time interval $(t, t + dt)$.

Clearly the perceptual attributes of reverberation are of great importance also for the design of reverberation algorithms. The ultimate goal is to determine an orthogonal set of subjective attributes, using e.g. multidimensional scaling techniques, and provide reverberation algorithms with a set of knobs each of which controls a different perceptual attribute. A fundamental problem with this kind of approach is that the number of perceptual dimensions is not known *a priori*, and moreover it is hard to assign relevance to dimensions that are added.

4.2.2.1 Reverberance

We have defined the reverberation time T_r in Sec. 4.2.1 as the time required for the sound pressure to decay 60 dB.² This is one of the most important parameters for the perception of the *reverberance*, i.e. the property of the environment of adding fullness and loudness to a dry sound, and of giving the listener a sense of being enveloped by the sound. Some use the term “liveness” to refer to a similar concept, and by contrast call “dead” an environment that is not reverberant.

We have already mentioned that T_r is in general a function of frequency, because absorption in materials is typically higher at higher frequencies. A confirmation of this is given in Figure 4.7, which shows a waterfall representation of a RIR: one can see that each frequency bin decays with a different rate. This dependence of T_r on frequency is also important perceptually. In general the mid-frequency T_r can be considered to be the best measure of the overall reverberant characteristics of a room.

Clearly the audibility of reverberation depends greatly on the sound source. For music or speech, the early portion of the reverberant decay contributes more to the perception of reverberance than does late reverberation, because it is audible during pauses and gaps between notes, syllables, and words. For this reason an *early decay time (EDT)* parameter is also used as a complementary measure of reverberance. The EDT is defined as the time required for the sound pressure to decay from 0 to -10 dB, multiplied by a factor of 6 (which merely serves to facilitate comparison with T_r).

²An alternative and not completely equivalent definition commonly used in the domain of concert hall acoustics is the following: T_r is the time required for the sound pressure to decay from -5 to -35 dB, multiplied by a factor of 2.

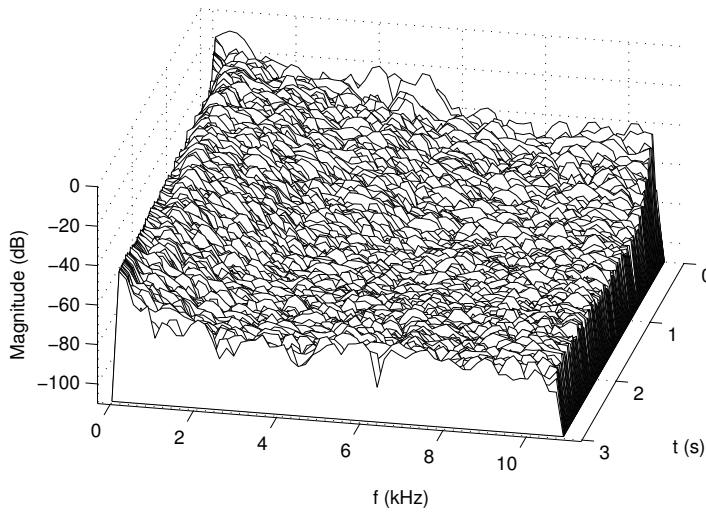


Figure 4.7: Waterfall representation for the RIR of Fig. 4.3.

One might wonder what is the “optimal” T_r for a reverberant environment. The answer depends first of all on the source signal: in the case of speech a relatively short T_r is generally preferred, since when listening to speech we generally want to understand what the speaker is saying and thus we need to perceive each element of the sound signal. For music on the contrary, a longer T_r can make the listening experience more pleasant by masking small imperfections and blending musical sounds. Given this remark, it is not surprising that T_r ’s in (good) concert halls are usually in the range 1.8 to 2.2 s, while in opera houses values are usually in the range 0.9 to 1.5 s because the listener has to be able to enjoy the music as well as to understand the text. Note however that T_r ’s of renowned opera theaters are more scattered than those of equally renowned concert halls.

4.2.2.2 Early reflections and spatial impression

The subjective attribute of *spatial impression* refers to the sense of a listener of being in close communication with the sound source and surrounded by the sound. Other terms often found in the literature and referred to similar concepts are spaciousness, envelopment, ambience, apparent source width. Subjective judgements about this property appear to be strongly correlated to the structure of the early reflections of the environment, with two elements being of specific importance.

A first commonly accepted result is that the degree of spatial impression depends on the *initial time-delay gap* t_I , the difference in arrival times between the direct sound and the first reflection. A lack of early reflections (i.e. a long t_I) has the effect of making the sound source perceived as remote and disconnected from the listener, while a short t_I provides the desired sense of envelopment. Some studies suggest that a parameter t_I defined as above becomes useless if the first reflection is much weaker than the following ones.

A second physical correlate of spatial impression is the fraction of lateral energy to the total energy within the early reverberation: a significant amount of *lateral* early reflections, i.e. reflections coming from the sidewalls, provides the listener with the impression of being enveloped by the sound. A quantitative estimate of this property is the so-called *lateral energy fraction* LF , defined as

$$LF_t = \frac{\int_0^t h_{lat}^2(\tau) d\tau}{\int_0^t h^2(\tau) d\tau}, \quad (4.22)$$



where $h(t)$ is the room impulse response measured with an omnidirectional microphone while $h_{lat}(t)$ is the one measured with a dipole microphone (with null axis facing forward this captures lateral energy in the $\pm 20^\circ \pm 90^\circ$ range). A typical integration time is $t = 80$ ms.

The LF_t measure has been superceded by another parameter, the *early interaural cross-correlation coefficient $IACC_E$* . Let us first define the interaural cross-correlation function $IACF(t)$ as

$$IACF(t) = \frac{\int_{t_1}^{t_2} h^{(l)}(\tau)h^{(r)}(\tau+t)d\tau}{\sqrt{\int_{t_1}^{t_2} h^{(l)}(\tau)d\tau \int_{t_1}^{t_2} h^{(r)}(\tau)d\tau}}, \quad (4.23)$$

where $h^{(l),(r)}(t)$ are the so-called Head-Related Impulse Responses at the entrance of the left and right ear canals, respectively (measured e.g. with a “dummy-head” such as those described later on in Sec. 4.6.1), with the listener facing the sound source. Therefore $IACF(t)$ is a *binaural* attribute of reverberation, while all the parameters previously examined in this section are *monoural* attributes.

The interaural cross-correlation coefficient $IACC$ is the maximum of $IACF(t)$ in a range ± 1 ms:

$$IACC = \max_{t \in (-1,1) \cdot 10^{-3}} IACF(t). \quad (4.24)$$

In particular, if the integration times $t_1 = 0$, $t_2 = 80$ ms are used then the above equations provide the early interaural cross-correlation coefficient $IACC_E$. This is a measure of the similarity of the sound signals arriving at the two ears during the first 80 ms. If the sounds are equal then $IACC_E = 1$, while if they are two independent random signals then $IACC_E = 0$. The $IACC_E$ is a measure of spatial impression because it scales with the fraction of lateral early reflections arriving at the ears: as the number of reflections from outside the median plane increases, the $IACF(t)$ function broadens and consequently $IACC_E$ takes smaller values.

In concert halls initial time-delay gap t_I and the amount of lateral energy are correlated parameters. Measures of t_I in real concert halls show a high correlation of this parameter with the hall width: in a narrow hall it can be shorter than 30 ms, while in a wide hall it can be longer than 50 ms. On the other hand, the hall width is clearly correlated with the fraction of lateral energy arriving at the listener, which will increase as the hall narrows. It is a common finding in the literature of concert hall acoustics that subjective rankings of the acoustic quality of halls scale with their width.

As a final remark, it has to be noted that the perception of spatial impression is largely independent of the reverberation time: halls with similar T_r values but different t_I and $IACC_E$ values sound very different from each other. This finding supports the commonly accepted assumption that early reflections and late reverberation play rather separate roles in the perception of reverberation.

4.2.2.3 Clarity

The subjective attribute of *clarity* refers to the “transparency” of a reverberant environment. If the source signal is music, then clarity is associated to the ability of a listener to perceive musical details, while if the source signal is speech then clarity correlates to speech intelligibility. An alternative term which is sometimes found in the literature is that of distinctness.

Single reflections of a reverberant environment are not perceived as individual events, except for exceptional (and generally undesirable) cases. Roughly speaking, early reflections have the effect of making the sound source appear more extended and to increase the apparent loudness of the direct sound. On the contrary, reflections arriving with longer delays are considered to be detrimental for the transmission of information, since they cause different portions of the direct sound signal to merge.



A quantitative measure of clarity is the *clarity index*, or early-to-reverberant energy ratio C_t :

$$C_t = 10 \log_{10} \left(\frac{\int_0^t h^2(\tau) d\tau}{\int_t^\infty h^2(\tau) d\tau} \right), \quad (4.25)$$

measured in dB. The integration time t is ideally the time instant where late reverberation starts, and is typically selected to be $t = 80$ ms. Thus C_t is a measure of early to late energy ratio.

It is sometimes recommended that $C_t|_{t=0.008}$ for concert halls takes values in the range of -2 to $+1$ dB. Note however that this parameter is not an independent measurable quality, since it correlates to the initial time-delay gap t_I and also on the early decay time EDT . Therefore, subjectively “good” values of the clarity index will also depend on t_I and EDT values. In other words, the subjective attribute of clarity is not orthogonal to reverberance and spaciousness.

Note also that C_t is strongly dependent on the distance between source and listener: the direct sound falls off 6 dBs for each distance doubling, whereas the reverberant level remains approximately constant throughout the room. For this reason, the ratio of direct to reverberated energy is one of the most important cues for the perception of distance, as we will see in Sec. 4.5.

A second objective parameter that relates to the subjective attribute of clarity is the *center time* t_s , defined as the center of gravity time of the sound field:

$$t_s = \frac{\int_0^\infty \tau \cdot h^2(\tau) d\tau}{\int_0^\infty h^2(\tau) d\tau}. \quad (4.26)$$

Obviously a single reflection with a given strength will contribute the more to t_s the longer it is delayed with respect to the direct sound. Therefore high clarity is associated to low values of t_s . It has to be noted however that many studies report a high correlation of t_s with C_t , in the range $50 < t < 80$ ms. Therefore this parameter does not add new information with respect to the clarity index.

4.2.2.4 Other perceptually relevant parameters

In Sec. 4.2.1 we have discussed room acoustics in terms of rays and normal modes, and we have not considered other real-world phenomena. One of the most relevant of these is sound *diffusion* of sound waves: very roughly, diffusion is due to irregularities (at various scales) of reflecting surfaces, that cause scattering of reflected acoustic energy in many directions. This physical concept has a direct perceptual counterpart. If one listens to music in a rectangular hall with perfectly flat sidewalls, the sound takes on an undesirable harsh character. In order to produce the effect of a mellower sound and to increase spaciousness during late reverberation, diffusion should be physically realized at fine and large scales. A commonly accepted measure of diffusion is the *late interaural cross-correlation coefficient IACCL*. This is defined from Eqs. (4.23, 4.24) using integration times $t_1 = 80$ ms and $t_2 = 3$ s, i.e. by estimating the IACF function in the late reverberation portion. Similarly to the *IACC_E* parameter, *IACCL* is a *binaural* attribute of reverberation. It provides a measure of the correlation of the signals at the two ears during late reverberation.

Loudness (or *strength*) is often mentioned as a relevant subjective attribute. Of course the overall loudness depends on the power output of the sound source and not only on the reverberation of the environment. Nonetheless it is useful to introduce a measure of loudness of the environment, which is normalized with respect to the source power. Such a measure can be used e.g. as a complementary parameter to the clarity index (see Eq. (4.25) above), since high clarity is of no use if the sound cannot be heard at proper loudness. A normalized measure of environmental loudness is given by the following quantity, often called *strength index* G :

$$G = 10 \log_{10} \left(\frac{\int_0^\infty h^2(\tau) d\tau}{\int_0^\infty h_0^2(\tau) d\tau} \right), \quad (4.27)$$



where $h(t)$ is as usual the room impulse response and $h_0(t)$ is the response to the same non-directional impulse measured in an anechoic environment at a distance of 10 m. Note however that subjective loudness increases with reverberation time and is affected by the structure of early reflections. Therefore G is not an independent correlate of loudness.

Finally, the most elusive subjective attributes are those related to timbral qualities of a reverberant environment. Roughly speaking, many of the attributes in this family are related to the frequency-dependent shape of the reverberation time. One such attribute is *warmth*, or sometimes *timbre*, which characterizes the musicians' judgement of "richness in bass". This attribute correlates with the variation of the reverberation time in the low- and mid-frequency range: as an example, a quantitative measure of warmth can be the ratio of the average T_r in the range 250 – 500 Hz to that in the range 500 – 1000 Hz, or alternatively the slope of a linear interpolation of the *EDT* function in the range 125 – 2000 Hz. Other timbre-related attributes are *heaviness* and *liveness*, which roughly relate to low-frequency and high-frequency variations of the reverberation time, respectively.

4.2.2.5 The Energy Decay Relief

A compact representation of the perceptually relevant features of a room impulse response is the so-called *Energy Decay Relief (EDR)* function, which is a time-frequency representation of the reverberation energy. The EDR function is in a way a generalization of the Energy Decay Curve (EDC) discussed previously, and is constructed as follows: given a RIR $h(t)$, this is bandpass filtered into a number N of frequency bands, and the EDC of each of the bandpassed responses $h_i(t)$ ($i = 1 \dots N$) is computed. The resulting function $EDR(t, \omega)$ can be displayed as a surface in the 3-D space. The section $EDR(0, \omega)$ provides the power gain as a function of frequency. A section $EDR(t, \omega_0)$ shows the energy decay curve for a given frequency bin around ω_0 .

M-4.57

Write a function that computes the EDR given a RIR.

M-4.57 Solution

```
function [EDR,F,T] = compute_edr(rir,Fs,frameSizeMS,overlap);
% adapted from http://ccrma.stanford.edu/%7Ejos/vguitar/

% define STFT parameters
minFrameLen = Fs*frameSizeMS/1000; %frameSizeMS is the framelen in ms
frameLen = 2^nextpow2(minFrameLen); % frame length = fft size
frameWindow = hann(frameLen);

%compute spectrogram and energy
[B,F,T] = specgram(rir,frameLen,Fs,frameWindow,round(overlap*frameLen));
[nBins,nFrames] = size(B);
B_energy = B.*conj(B);

%compute EDR (in dB)
EDR = zeros(nBins,nFrames);
for i=1:nBins %compute EDC for each frequency band
    EDR(i,:) = 10*log10(abs(fliplr(cumsum(fliplr(B_energy(i,:))))));
end
EDR = EDR - max(max(EDR)); %normalize at 0 dB
```

The time-frequency EDR function can be parametrized through two functions of frequency only. The first one is $T_r(\omega)$, the frequency-dependent reverberation time. The second one is the *frequency*



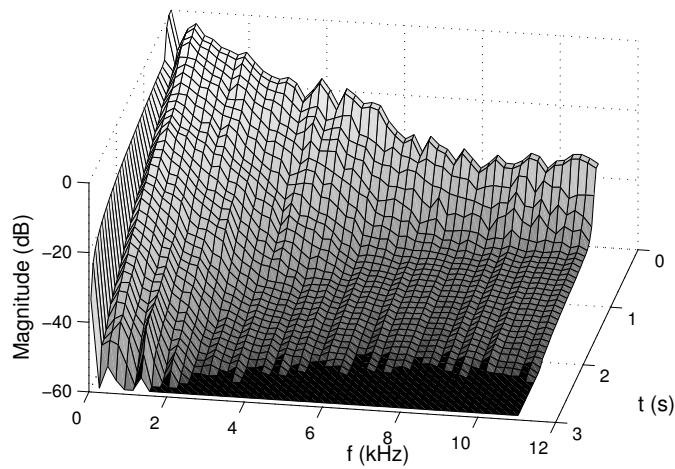


Figure 4.8: Energy Decay Relief for the RIR of Fig. 4.3, normalized at 0 dB and truncated at –60 dB.

response envelope, $G(\omega)$. This latter function is constructed by backward interpolating up to $t = 0$ the exponential decay time. For an ideally diffuse reverberation that decays exponentially, one has the equality $G(\omega) = EDR(0, \omega)$ and G coincides with the power gain of the room. In non-ideal cases, $G(\omega)$ only represent a “conceptual” $EDR(0, \omega)$ of the late reverberation, and the parametrization through $T_r(\omega)$ and $G(\omega)$ is only valid for the late portion of $EDR(t, \omega)$.

The EDR is sometimes regarded as a perceptual “signature” of a RIR, meaning with this that a large number of measures of independent perceptual factors can be categorized as energy ratios or energy decay slopes computed in different time-frequency regions of the EDR. Figure 4.8 shows an example of EDR. Note that, in accordance to our predictions, T_r is shorter at higher frequencies.

4.3 Algorithms for synthetic reverberation: the perceptual approach

If a RIR signal is available, the most straightforward approach to synthetic reverberation is to convolve an anechoic input signal with such a RIR. We do not review techniques for impulse response measurement in this chapter, nor we address numerical techniques for convolution. We only observe that *direct convolution*, obtained by storing each sample of the impulse response as a coefficient of an FIR filter whose input is the dry signal, becomes easily impractical if the length of the response exceeds few tenths of a second, as it translates into a FIR filter of order $N \sim 10^4$. But even if we have enough computational resources for direct convolution, or use fast convolution techniques, a real recorder RIR has the disadvantage that it is not easily modified to simulate changes in room attributes.

In order to overcome these limitations, in the second half of the 20th century several engineers and acousticians developed electronic devices, models, and algorithms for synthetic reverberation that are based on a *perceptual approach*, in which efficient filter representations are used, and only the perceptually salient features of reverberation are simulated and controllable.

4.3.1 Late reverberation

We have previously seen that a RIR can be seen as made of two components, early reflections and late reverberation. In this section we discuss perceptual models for late reverberation, and we postpone early reflection modeling to Sec. 4.3.2.

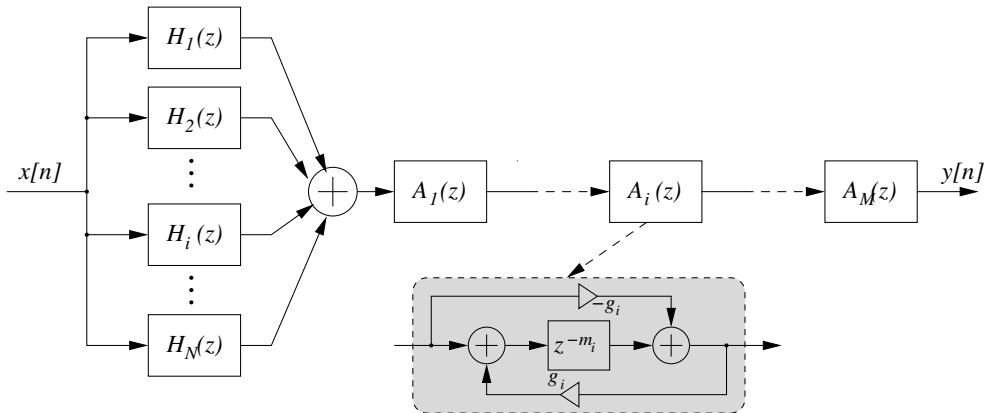


Figure 4.9: Block scheme of a reverberator based on comb filters (the H_i blocks) and all-pass comb filters (the A_i blocks). The internal structure of the A_i filters is shown in the grey box.

4.3.1.1 Recirculating delays

The two main computational structures that can be used for the inexpensive simulation of complex patterns of echoes associated to late reverberation are the recursive comb filter $H(z)$ (see Karplus-Strong in Ch. *Sound modeling: source based approaches*) and the so-called *all-pass comb filter* $A(z)$:

$$H(z) = \frac{z^{-m}}{1 - gz^{-m}}, \quad A(z) = \frac{z^{-m} - g}{1 - gz^{-m}}. \quad (4.28)$$

It is easily seen that $A(z)$ is an all-pass structure, since each of the m poles is the reciprocal of one of the m zeros and the amplitude response $|A(z)|$ is therefore flat. For $m = 1$ the structure reduces to the first-order all-pass filter examined in Ch. *Sound modeling: source based approaches*. The (positive) gain g in $A(z)$ has to be less than unity in order to ensure stability.

Figure 4.9 depicts a reverberator constructed using comb-filters and all-pass comb filters, together with a realization of the all-pass comb (see the grey box). The general idea behind this structure is the following. First, the parallel combination of comb filters generates a frequency response that contains peaks contributed by each comb. In theory we can obtain an arbitrary modal density by using a sufficiently large number N of comb filters. Second, the series combination of all-pass combs that receives the output of the parallel combination of combs has the effect of dramatically increasing the temporal density of reflections, because each echo generated by $A_i(z)$ will create a set of echoes in $A_{i+1}(z)$. Again, an arbitrarily high reflection density can be in principle obtained by using a sufficiently large number M of all-pass combs.

4.3.1.2 Parameter tuning

The choice of a proper set of parameter values is critical in order to obtain convincing results. In the remainder of this section we provide a list of commonly accepted guidelines. The sample delays m_i of the combs should be mutually coprime (or incommensurate), in order to reduce the superimposition of echoes in the impulse response, thus maximizing the modal density and reducing the so-called flutter echoes.

The gains g_i of the combs can be chosen as functions of the sample delays m_i , given a desired reverberation time T_r , as follows: we want to find the number R of loops in the i th comb after which the dB amplitude of an unitary impulse has become -60 dB; the amplitude after R loops is $g^R =$

Moreover R loops are completed in the time $T_r = Rm_i/F_s$; therefore the following equation holds for the reverberation time of a single comb:

$$\frac{F_s \cdot 20 \log_{10}(g_i)}{m_i} = -\frac{60}{T_r} \Rightarrow g_i = 10^{-3 \frac{m_i}{F_s T_r}}. \quad (4.29)$$

Note that this choice ensures that the pole moduli $\sqrt[m_i]{g_i} = 10^{-3 \frac{1}{F_s T_r}}$ have the same value for all the combs. If this condition was not verified, then the poles with largest moduli would resonate longer and would add an undesired tonal coloration in the late decay.

A quantitative estimate of the modal density provided by the parallel comb structure can be easily obtained. If the m_i 's of the combs are mutually coprime, then the modal density D_f (which is number of frequency peaks per Hz) can be estimated as

$$D_f = \sum_{i=1}^N \frac{m_i}{F_s} = \frac{N\bar{m}}{F_s}, \quad (4.30)$$

where \bar{m} is the mean sample delay length. Note that this modal density is constant for all frequencies, unlike in real rooms (see Eq. (4.9)). A too low D_f can introduce audible beating between two neighboring modes, especially in response to narrowband signals. In order to avoid this effect, a good rule of thumb is to choose the m_i 's such that $D_f \geq T_r$: this ensures that the average beat period is at least equal to the reverberation time.

In a similar way we can estimate quantitatively the temporal reflection density provided by the parallel combination of combs: each filter outputs one echo every m_i/F_s seconds, therefore the combined reflection density (number of reflections per second) is

$$D_r = \sum_{i=1}^N \frac{F_s}{m_i} \approx \frac{NF_s}{\bar{m}}, \quad (4.31)$$

where the last approximation only holds when the m_i are similar. Again, the reflection density is constant as a function of time, unlike real rooms (see Eq. (4.21)). A value $D_r = 10^3$ is sometimes considered to be sufficient to sound indistinguishable from diffuse reverberation, although higher values (e.g. $D_r = 10^4$) are preferable.

From the two estimates (4.30) and (4.31) provide an estimate of the number of comb filters needed in order to achieve desired modal and reflection densities:

$$N = \sqrt{D_f D_r}. \quad (4.32)$$

Note however that this estimate does not consider the effect of the cascaded series of all-pass comb filters A_i : as already mentioned, the A_i provide a dramatic increase of the reflection density and allow to a number N of comb filters that is smaller than the one estimated from Eq. (4.32).

M-4.58

Realize the reverberant structure of Fig. 4.9. The reverberator can be tried with $N = 4$, $M = 2$, and the following settings: time delays m_i/F_s ($i = 1 \dots 4$) of the comb filters between 30 and 45 ms, time delays m_i/F_s ($i = 5, 6$) of the all-pass combs between 1.7 and 5 ms, modal density $D_f = 1000$, gains of the all-pass combs $g_i = 0.7$ ($i = 5, 6$). With these settings the structure is known as Schroeder reverberator (see bibliography).

M-4.58 Solution



```

function y = reverb_schroeder(x, Tr, m_H, m_A, g_A);
% x: input signal; m_H: N-dim array of delays (in samples) of combs H_i;
% m_A: M-dim array of delays (in samples) of all-passes A_i;
% g_A: M-dim array of all-pass gains

global Fs;
y = zeros(length(x),1); %output signal updated after each single filter
for i=1:length(m_H)           %parallel comb filtering
    g_H = 10^(-3*m_H(i)/(Fs*Tr));          % gain of ith comb
    num_H = [zeros(1,m_H(i)),1];            % numerator of ith comb
    den_H = [1,zeros(1,m_H(i)-1),-g_H];    % denominator of ith comb
    y = y + filter(num_H, den_H, x);        % update comb parallel
end
for i=1:length(m_A)           %series all-pass filtering
    num_A = [-g_A(i),zeros(1,m_A(i)-1),1]; % numerator of ith all-pass
    den_A = [1,zeros(1,m_A(i)-1),-g_A(i)]; % denominator of ith all-pass
    y = filter(num_A, den_A,y);             % update all-pass series
end

```

4.3.1.3 Low-pass combs

The reverberators discussed above sound reasonably well especially for short reverberation times and low reverberation levels. For different settings however they suffer from a number of problems. First, the reverberation is not dense enough at the beginning, resulting in a “grainy” sound quality (especially with impulsive sounds). Second, late reverberation tends to exhibit an already mentioned “fluttering” effect. Third, especially for long T_r ’s a “ringing” effect can be heard, which gives an undesired metallic quality to the reverberation. Fourth, the modal density is not sufficiently large and, as already mentioned, does not increase with frequency. Fifth, the reverberation time T_r does not depend on frequency, unlike in real rooms (see Sec. 4.2.2 and the EDR function there discussed).

A first obvious way of improving the modal density is to increase the number of comb filters in parallel, especially when long reverberation times need to be simulated. A second more substantial improvement amounts to employ, in place of comb filters, a *low-pass comb* filter, where a low-pass filter H_{lp} is inserted in the feedback loop of the comb together with the scalar gain g . The purpose of this modification is to simulate the attenuation effects of higher frequencies, due to air viscosity, heat conduction, and energy losses at reflection. As a result, T_r now decreases at higher frequencies and makes the reverberation sound more realistic. In addition, the response to impulsive sounds is also improved, due to the smoothing effect of the low-pass filtering.

If a simple one-pole low-pass filter H_{lp} is used, then the low-pass comb filter is given as

$$H(z) = \frac{z^{-m}}{1 - H_{lp}(z)z^{-m}}, \quad \text{with } H_{lp}(z) = \frac{g_2}{1 - g_1 z^{-1}}. \quad (4.33)$$

One could verify that in order for $H(z)$ to be stable the condition $\max_z |H_{lp}(z)| = g_2/(1 - g_1) < 1$ must hold. A practical choice is $g_2 = g(1 - g_1)$, with $g < 1$. In this way the overall T_r is still controlled by the parameter g as in Eq (4.29)

Note that we have already introduced the low-pass comb filter for the Karplus-Strong algorithm in Ch. *Sound modeling: source based approaches*, although here we are using a different low-pass filter H_{lp} .

Coefficients of the low-pass combs: g_2 can be determined as a function of the delay length and the desired T_r , as explained in the previous section. The g_1 coefficient can also be related with decay time at a specific frequency or fine tuned by direct experimentation.



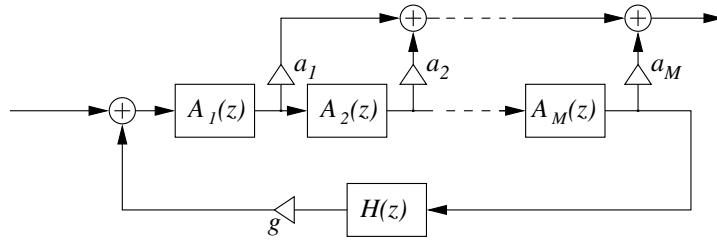


Figure 4.10: A reverberator constructed with a series connection of all-pass filters and a low-pass filter in feedback.

M-4.59

Realize the reverberant structure of Fig. 4.9 using low-pass combs of the form (4.33). The reverberator can be tried with $N = 6$, $M = 1$, and with the following settings: time delays m_i/F_s ($i = 1 \dots 6$) of the combs distributed between 50 and 78 ms, coefficients $g_{1,i}$ of the low-pass filter distributed between 0.40 and 0.48 (at $F_s = 44.1$ kHz), time delay of the all-pass comb $m_7/F_s = 6$ ms, gain of the all-pass comb $g_7 = 0.7$. With these settings the structure is known as Moorer reverberator (see bibliography)

M-4.59 Solution

```

function y = reverb_moorer(x, Tr, m_H, g1_H, m_A, g_A)
% x: input signal; m_H: N-dim array of delays (in samples) of combs H_i;
% g1_H: N-dim array of coefficients of the H_lp's; m_A : M-dim array of
% delays (in samples) of all-passes A_i; g_A: M-dim array of all-pass gains

global Fs;
y = zeros(length(x),1); %output signal updated after each single filter
for i=1:length(m_H)           %parallel comb filtering
    g_H = 10^(-3*m_H(i)/(Fs*Tr));          % gain of ith comb
    num_H = [zeros(1,m_H(i)),1,g1_H(i)]; % numerator of ith lowp. comb
    den_H = [1,-g1_H(i),zeros(1,m_H(i)-2),-g_H*(1-g1_H(i))]; % denominator
    y = y + filter(num_H, den_H, x);        % update comb parallel
end
for i=1:length(m_A)           %series all-pass filtering
    num_A = [-g_A(i),zeros(1,m_A(i)-1),1]; % numerator of ith all-pass
    den_A = [1,zeros(1,m_A(i)-1),-g_A(i)]; % denominator of ith all-pass
    y = filter(num_A, den_A, y);            % update all-pass series
end

```

Clearly the filter coefficients num_H and den_H have been determined by combining Eqs. (4.33).

4.3.1.4 Nested all-pass filters

Despite the improvements provided by this latter reverberator, some problems remain. First, it is not possible to tune the reverberator to a desired $T_r(\omega)$ function. Second, the modal density is still constant with respect to frequency and the ringing quality and the fluttering effect in the reverberation tail remain, although reduced to some extent. In order to overcome this problems some researchers have proposed reverberators with entirely different structures than the one shown in Fig. 4.9. One such structure is shown in Fig. 4.10.

As before, the cascaded all-pass filters $A_i(z)$ provide a high temporal density of reflections, because each echo generated by a filter will create a set of echoes in the following one. In this case however, the

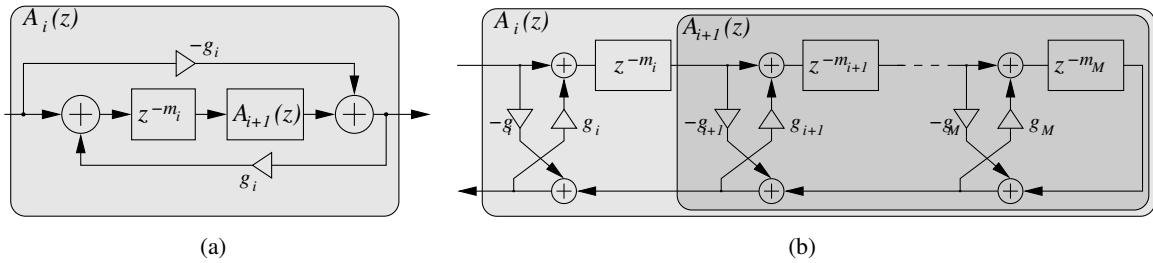


Figure 4.11: Nested all-pass filters; (a) generalization of an all-pass structure (see Fig. 4.9), and (b) realization by means of a lattice structure.

output from the last one is recirculated to the series connection through a low-pass filter $H(z)$ and an attenuating gain g . The resulting system is stable, if the condition $|gH(e^{j\omega})| < 1 \forall \omega$ is verified.

The low-pass filter $H(z)$ can be interpreted as simulating frequency-dependent absorptive losses, and the gain g provides control over the reverberation time. An important effect of this outer feedback loop is that the characteristic metallic sound of the series all-pass is drastically reduced. Another peculiarity of this structure is that the output is constructed as a linear combination of the all-pass outputs. Since each each tap outputs a different response shape, the coefficients a_i can be adjusted in order to shape the amplitude envelope of the reverberant decay.

A final remark concerns the possibility of generating a reflection density that increases with time, as in real rooms. A structure that achieves this goal is a *nested all-pass filter* $A_1(z)$, which can be defined recursively as follows:

$$\begin{aligned} A_{M+1}(z) &= 1, \\ A_i(z) &= \frac{z^{-m_i} A_{i+1}(z) - g}{1 - g z^{-m_i} A_{i+1}(z)}, \quad \text{for } i = 1 \dots M. \end{aligned} \quad (4.34)$$

Figure 4.11(a) shows that this structure can be seen as a generalization of the all-pass comb, in which part of the delay line has been substituted by an all-pass filter. Figure 4.11(b) explodes this structure into a lattice realization. It is easy to verify that each of the nested filters $A_i(z)$ are all-pass. Moreover, Fig. 4.11(a) shows that each echo generated by the inner all-pass $A_{i+1}(z)$ is recirculated to itself through the outer feedback path of $A_i(z)$: this intuitively explains why this structure provides a reflection density that increases with time.

M-4.60

Realize the reverberant structure of Fig. 4.10 using nested all-pass filters of the form (4.34).

4.3.2 Early reflections

So far we have only examined perceptually-based algorithms for the simulation of late reverberation. In this section we address the simulation of early reflections, which have great importance in the perception of the acoustic space as we have seen in Sec. 4.2.2.

4.3.2.1 FIR structures

As previously discussed, the early response of a room is sparsely populated with attenuated impulses. These can be straightforwardly simulated using a direct-form FIR filter that reproduces these impulses



explicitly and accurately. For the determination of the filter parameters, a good rule of thumb is to apply to the early reflections delays the same criterion of “mutually-primeness” used before for the comb delays. A better strategy is to derive the parameters from some geometric modeling technique, e.g. the source image method discussed in Sec. 4.2.1.

M-4.61

Write a function that computes a signal containing the first R early reflections

M-4.61 Solution

```
function y = reverb_earlyrefl(x, m_E, a_E);
% x: input signal; m_E: R-dim array of delays (in samples) of early
% reflections; a_E: R-dim array of gains of early reflections

num = zeros(1,max(m_E)+1); %empty FIR numerator
num(m_E+1) = a_E; % populate numerator with early reflection gains
y = filter(num,1,x);
```

The delays in this script have a slightly different meaning than those in Fig. 4.12, since they are not cascaded.

Figure 4.12 shows an example of early reflection modeling, in which the FIR filter simulates the first R reflections and has been realized using a direct form structure. The early reflection filter has to be connected to a late reverberation block: Fig. 4.12(a) and 4.12(b) show two possible connections. In Fig. 4.12(a) the late reverberator receives the delayed input signal, and therefore the FIR response will always occur before the late response in the final output. Figure 4.12(b) shows a more complex coupling between the two blocks. In this case the late reverberator is driven by the output of the FIR filter, with the result of increasing the reflection density in the late reverberation. Moreover, additional control parameters are available: the gain g can be adjusted in order to balance the early/late reverberation ratio, while the delays D_1 , D_2 can be tuned so that the start of the late reverberator output coincides with the last pulse output from the FIR filter, thus avoiding undesired gaps in the overall response.

M-4.62

Realize the reverberator depicted in Fig. 4.12(a), where the early reflection FIR filter has to be coupled to one of the late reverberation structures discussed in the previous sections.

M-4.62 Solution

```
function y = reverb_schroeder_earlyrefl(x, Tr, m_E, a_E, m_H, m_A, g_A);

global Fs;
y_E = [reverb_earlyrefl(x,m_E,a_E); zeros(max(m_E),1)]; %early refl.
y_L = reverb_schroeder([zeros(max(m_E),1); x],Tr,m_H,m_A,g_A); %late rev.
y=y_E+y_L;
```

M-4.63

Realize the reverberator depicted in Fig. 4.12(b), where the early reflection FIR filter has to be coupled to one of the late reverberation structures discussed in the previous sections. Compare the resulting impulse responses with the ones obtained from M-4.62.

M-4.63 Solution



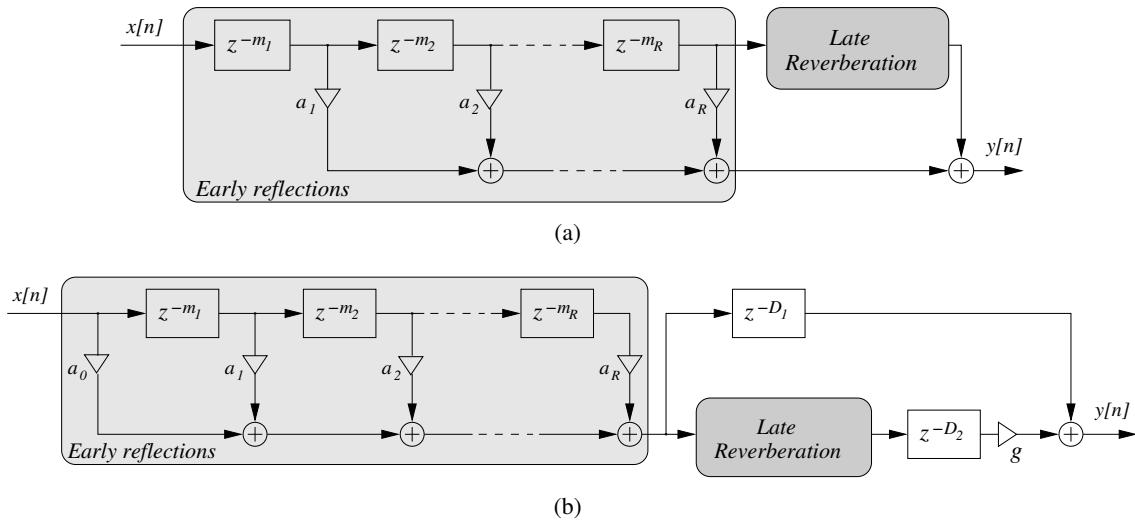


Figure 4.12: Two realizations of a reverberator with early reflections; (a) late reverberation block receiving the delayed input signal, and (b) late reverberation block receiving the output of the early reverberation FIR filter, with additional control parameters D_1 , D_2 , g . The late reverberation block can be one of the structures examined in the previous sections.

```

function y = reverb_moorer_earlyrefl(x, Tr, m_E, a_E, m_H, g1_H, m_A, g_A, g_mix);

global Fs;
y_E = reverb_earlyrefl(x, m_E, a_E); %early refl.
y_L = reverb_moorer(y_E, Tr, m_H, g1_H, m_A, g_A); %late rev.

delaydiff = max(m_E) - min(m_H); % diff. in delay between early
if (delaydiff>0) % refl. and late rev.
    y = [y_E; zeros(delaydiff,1)] + g_mix*[zeros(delaydiff,1); y_L];
else
    y = [zeros(delaydiff,1); y_E] + g_mix*[y_L; zeros(delaydiff,1)];
end

```

In order to improve the quality of the FIR structure described above, one has to include some form of low-pass filtering that models frequency dependent losses. One possibility is to substitute each of the gains a_i with a low-pass filter, composed by considering the history of reflections for each echo. Early reflections are not perceived as individual events however, therefore it is not necessary to model accurately the spectral content of each single reflection. A cheaper, and often satisfactory, choice is to sum sets of reflections together and and to filter them through the same low-pass.

4.3.2.2 Directional effects

In this brief section we anticipate some concepts that will be addressed in Secs. 4.5 and 4.6, where we will address the topic of rendering the location in space of a sound source.

Effects due to reverberation and spatial perception of sound are related in many respects. On the one hand, reverberation has a relevant role in the perception of the location of a sound source, as we will see in Sec. 4.5. On the other hand, the subjective attribute of *spatial impression* is extremely important in the perception of reverberation, and should be accounted for in any synthetic reverberation algorithm: in

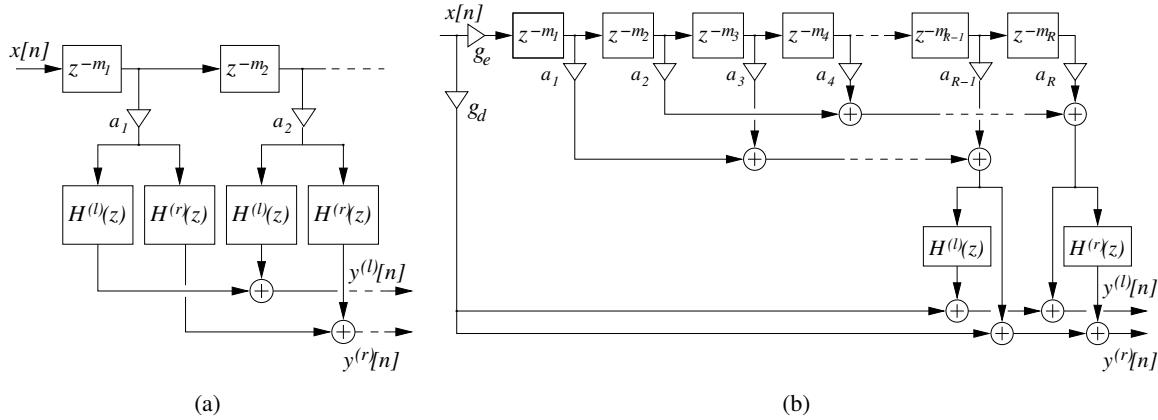


Figure 4.13: Two structures that associate directional filters to early reflections, for binaural reverberation; (a) one directional filter for each reflection, and (b) two directional filters for two sets of reflections.

Sec. 4.2.2 we have seen that early reflections in particular have a primary role in the formation of spatial impression (see the definition of the early interaural cross-correlation coefficient).

An early reflection reaches the two ears with different intensities and at different times, because of the shadowing effect of the head, the different distance traveled, the filtering properties of the pinna, and so on. For this reason early reverberation is most effective if it is presented *binaurally*, i.e. by taking into account these effects and presenting different early reflections to the two ears (e.g. via headphones). In this case one can associate with each early reflection a directional filter intended to reproduce localization cues. One structure that realizes this idea is shown in Fig. 4.13(a). $H^{(l),(r)}$ are the so-called Head-Related Transfer Functions,³ that represent the transfer function between the sound source and the entrance of the ear canals. These directional filters are associated to early reflections in a structure analogous to those shown in Fig. 4.12.

Another possibility is to sum sets of early reflections together and process each set with the same directional filter, so that all the reflections in a single set will be rendered with the same spatial location. This approach can still produce a convincing sensation of spatial impression, while being far more efficient. Various realizations of this general idea have been proposed. Figure 4.13(b) shows one possible realization: two sets of echoes are formed and each set is processed with the same directional filters. Various degrees of spatial impression can be obtained by playing with the gain g_e , and convincing results are obtained already with $R = 6$ reflections.

As a conclusion to this section it is worth mentioning that if no binaural processing is performed the addition of early reflections can in certain cases deteriorate the quality of a reverberator, as they cause tonal coloration of the sound without producing spatial impression.

³The transforms of the Head-Related Impulse Responses already introduced in Eq. (4.23).

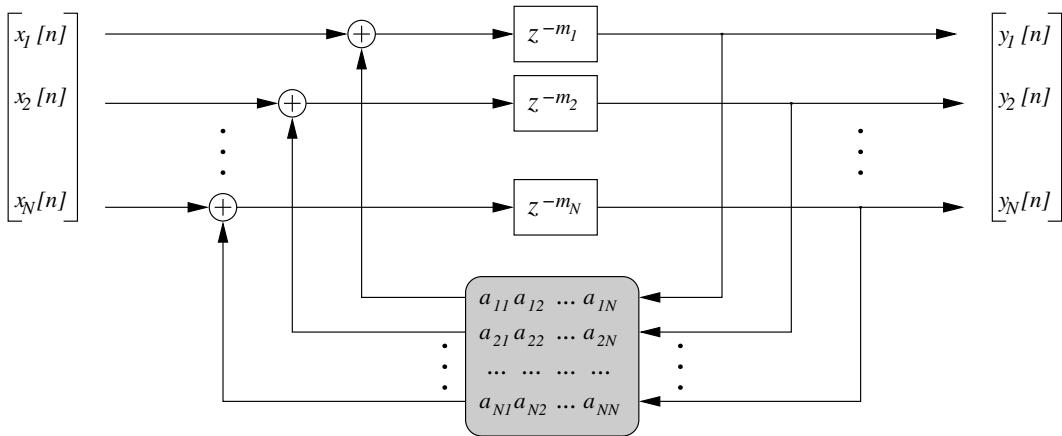


Figure 4.14: ...

4.4 Multidimensional reverberation structures

4.4.1 Feedback delay networks

4.4.1.1 A n-D generalization of the recursive comb filter

In the previous section we have seen that the recursive comb filter of Eq. (4.28) has been extensively used as the main building block of perceptual reverberators, as an inexpensive way to generate patterns of resonances. Now the question is: can we generalize the comb structure in order to achieve higher modal densities? The filter structure depicted in Fig. 4.14 provides a first answer. First, it is easily seen to be a vector generalization of the recursive comb filter, as it reduces to a parallel combination of ordinary comb filters when the feedback matrix $A = [a_{ij}]$ is diagonal. Second, and more interesting, it recirculates the output of the i th delay line to the input of the j th delay line, for every non-null element a_{ij} . This observation gives the intuition that when A is non-diagonal this structure is capable of much higher modal densities than a simple parallel of comb filters.

The generalization extend also to stability conditions. While the comb filter of Eq. (4.28) is stable if $|g| < 1$, the multidimensional structure of Fig. 4.14 is stable if $\|A\|_2 < 1$, where $\|\cdot\|_2$ is the spectral norm of a matrix.⁴ This can be easily verified by applying the conditions for Lyapunov stability, i.e. that the output $y[n]$ decreases in time when the input signal x is zero. Since

$$\|y[n]\|_2 = \left\| A \begin{bmatrix} y_1[n - M_1] \\ \vdots \\ y_N[n - M_N] \end{bmatrix} \right\|_2, \quad (4.35)$$

stability is guaranteed whenever the feedback matrix satisfies

$$\|Ay\|_2 < \|y\|_2 \quad \forall y. \quad (4.36)$$

In other words, a sufficient condition for stability is that the feedback matrix decreases the L² norm of its input vector. Since in general $\|Ay\|_2 < \|A\|_2 \cdot \|y\|_2$, we conclude that stability is guaranteed for $\|A\|_2 < 1$.

⁴The matrix norm corresponding to any vector norm $\|\cdot\|$ may be defined for any matrix A as $\|A\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|}$. The spectral norm $\|\cdot\|_2$ is the matrix norm induced by the L² vector norm.

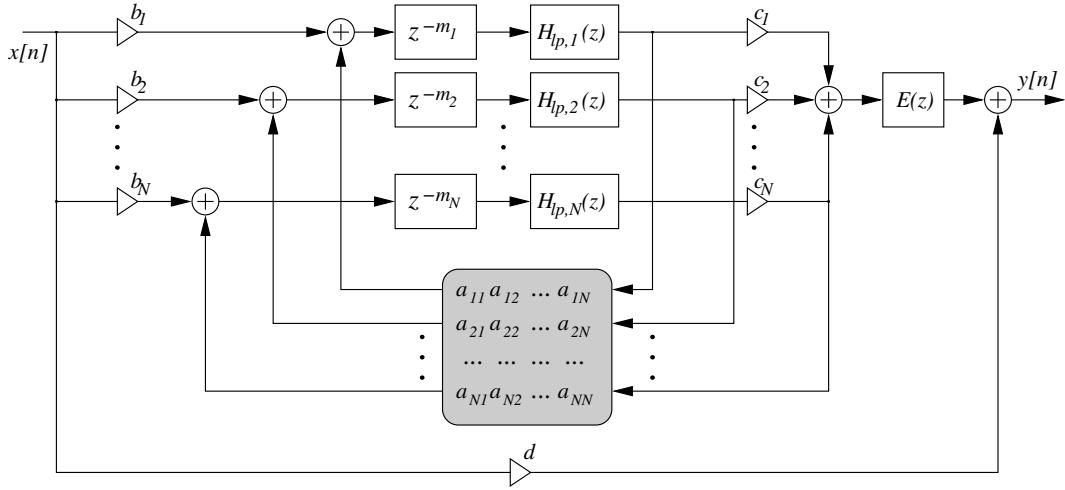


Figure 4.15: A Feedback Delay Network structure for artificial reverberation.

A class of matrices that satisfy the stability condition is

$$\mathbf{A} = \boldsymbol{\Gamma} \mathbf{Q}, \quad \text{where} \quad \boldsymbol{\Gamma} = \begin{bmatrix} g_1 & 0 & \cdots & 0 \\ 0 & g_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & g_N \end{bmatrix}, \quad |g_i| < 1, \quad (4.37)$$

and where \mathbf{Q} is an orthogonal matrix. Recall that (1) the spectral norm $\|\mathbf{A}\|_2$ is the square root of the largest eigenvalue of $\mathbf{A}\mathbf{A}^T$, and that (2) by definition \mathbf{Q} is orthogonal if and only if $\mathbf{Q}\mathbf{Q}^T = \mathbb{I}$. Then $\|\mathbf{A}\|_2 = \|\boldsymbol{\Gamma}\mathbf{Q}\| = \max_i |g_i|$.

The above analysis justifies the use of the structure of Fig. 4.14 as a multichannel reverberator in which N input signals $x[n]$ (or N replicas of a single input signal $x[n]$) produce N outputs $y[n]$ that are approximately mutually incoherent and thus can be used in a N -channel loudspeaker system to render a diffuse soundfield. A possible choice for the matrix \mathbf{A} is

$$\mathbf{A} = g \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & 1 & 1 & 0 \\ -1 & 0 & 0 & -1 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \end{bmatrix}, \quad |g| < 1, \quad (4.38)$$

which is immediately seen to belong to the class (4.37).

4.4.1.2 A general FDN reverberators

The “vector comb filter” that we have analyzed in the previous section is an example of a class of filter networks, known as *Feedback Delay Networks (FDNs)*. Figure 4.15 shows a more general FDN structure for artificial reverberation, that extends in many ways the one depicted in Fig. 4.14. First, it is a Single-Input, Single-Output structure which uses two $N \times 1$ vectors $\mathbf{b} = [b_i]$ and $\mathbf{c} = [c_i]$ to split the input into N channels and to combine the N outputs in one channel. Second, low-pass filters $H_{lp,i}(z)$ are cascaded to the delay lines. Third, the final output y is corrected with an additional filter $E(z)$ plus an additive term d . The transfer function of the system is almost immediately found to be:

$$\frac{Y(z)}{X(z)} = \mathbf{c}^T \left\{ [\mathbb{I} - \mathbf{D}(z)\mathbf{A}]^{-1} \mathbf{D}(z) \right\} \mathbf{b} \cdot E(z) + d = \mathbf{c}^T [\mathbf{D}(z^{-1}) - \mathbf{A}]^{-1} \mathbf{b} \cdot E(z) + d, \quad (4.39)$$

where $\mathbf{A} = [a_{ij}]$ is the *feedback matrix* of the system, and

$$\mathbf{D}(z) = \begin{bmatrix} z^{-m_1} H_{lp,1}(z) & 0 & \cdots & 0 \\ 0 & z^{-m_2} H_{lp,2}(z) & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & z^{-m_N} H_{lp,N}(z) \end{bmatrix}$$

is the *delay matrix* of the system. We shall see that this structure allows to orthogonalize to a great extent the reverberation parameters, as the various blocks can be independently tuned to fit desired values of different reverberation parameters.

M-4.64

Realize the reverberant structure of Fig. 4.15. With the 4×4 matrix given in Eq. (4.38), the structure of Fig. 4.14 is a special case of this.

M-4.64 Solution

```
function y = reverb_fdn(x,Fs,Tr,A,b,c,d,m);
% x: input signal; A: NXN feedback matrix; b: Nx1 array of input weights;
% c: 1XN array of output weights; d: scalar weight for input-to-output contrib.
% m: N-dim array of line delays (in samples)

N=size(A,1); %dimension of the FDN
delaylines = zeros(max(m),N); %create and initialize N delay lines
[num_H,den_H,num_E,den_E] = lossy_fdn(Fs,Tr,m); %initialize lossy components

y = zeros(size(x)); %initialize output signal
for n = 1:length(x) %audio cycle
    Y=zeros(N,1); % Y is the array of N signals after the lowpass filters
    for i=1:N Y(i)= filter(num_H(i), den_H(i), delaylines(m(i),i) ); end
    y(n)= filter(num_E, den_E, c*Y) +d*x(n); %compute output
    linein =b*x(n) + A*Y; %compute new input to lines
    delaylines = circshift(delaylines,1); %circular shift lines
    delaylines(1,:)=linein; %write lines
    if(mod(n,round(length(x)/20))==0) fprintf('%d%\n',round(n/length(x)*100)); end
end
```

Note that in this case we had to write an audio loop, since an explicit formulation of the rational transfer function (4.39) is not available in general: for this reason this realization is extremely inefficient. Note also that we are using an auxiliary transfer function `lossy_fdn`: see M-4.66 below.

Since an “ideal” late reverberation impulse response should resemble exponentially decaying noise, it is useful to start designing a lossless reverberator (with infinite reverberation time) and work on making it a good noise generator. Once this *lossless prototype* has been designed, one can work on obtaining the desired reverberation time in each frequency band. We associate to the FDN of Fig. 4.15 the lossless prototype of Fig. 4.16.

What does the losslessness requirement imply to the feedback matrix \mathbf{A} ? We know that by definition of losslessness the equality $\int_{\omega} \left\{ \sum_{i=1}^n |Y_i(e^{j\omega})|^2 \right\} d\omega = \int_{\omega} \left\{ \sum_{i=1}^n |X_i(e^{j\omega})|^2 \right\} d\omega$ must hold. Moreover it is a general result that a multidimensional filter is lossless if and only if its frequency response matrix $\mathbf{H}(e^{j\omega})$ is unitary, i.e. $\mathbf{H}(e^{j\omega})\mathbf{H}^*(e^{j\omega}) = \mathbb{I}$ (where $*$ denotes the complex-conjugate transpose as usual). In our case, it is quite straightforward to prove that \mathbf{A} being unitary is a sufficient condition for the overall frequency response matrix to be unitary. Moreover the entries a_{ij} have to be real in order for the system to output a real signal $y[n]$, and a unitary matrix with real entries is an orthogonal matrix.



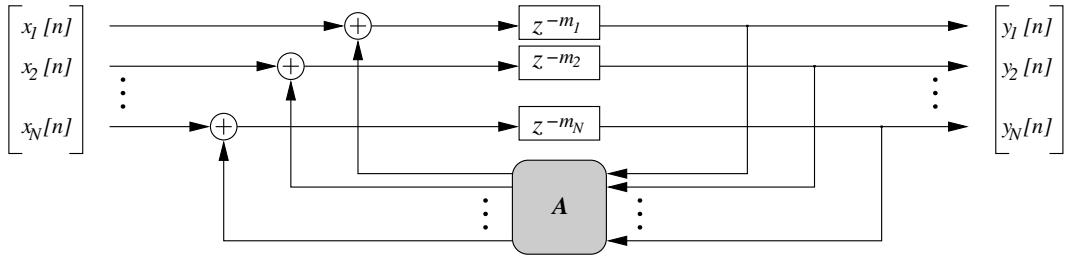


Figure 4.16: Lossless prototype network associated to the Feedback Delay Network of Fig. 4.15.

In conclusion, if \mathbf{A} is orthogonal then the network of Fig. 4.16 is lossless. Note however that this condition is sufficient but not necessary, thus the system may be lossless even with a non-orthogonal feedback matrix. We will return to this point in Sec. 4.4.2.

4.4.1.3 Designing the lossless prototype

Designing the lossless prototype means choosing the dimension N , the m_i 's, and the feedback matrix \mathbf{A} . Let us start with the dimension N and the delay lengths m_i . Together with the feedback matrix these parameters determine the buildup of reflection density. The criteria that we have examined in Sec. 4.3 (see in particular Eqs. (4.30, 4.31) can be applied also in this case with satisfactory results. Note however that Eqs. (4.30, 4.31) are no longer valid here, since, a non-diagonal feedback matrix increases the modal and reflection densities. Therefore in general the parameters have to be chosen on the basis of empirical observations. It is generally noted that $N = 8$ to 16 lines with a total delay $\sum_i m_i / F_s$ of 1 to 2 seconds already produce a response perceptually undistinguishable from white noise.

Let us now consider the lossless feedback matrix \mathbf{A} . The simplest orthogonal matrix is a diagonal matrix whose diagonal elements (which are the eigenvalues) have unit modulus: as already seen this choice corresponds to a parallel of ordinary comb filters. A more interesting family of orthonormal matrices are *Householder reflection matrices*. A specific Householder matrix is defined given the reference vector $\mathbf{u} = [1, \dots, 1]^T$:

$$\mathbf{A} = \mathbb{I} - \frac{2}{N} \mathbf{u} \mathbf{u}^T, \quad \text{then } \mathbf{A} \mathbf{x} = \begin{bmatrix} x_1 - \frac{2}{N} \sum_i x_i \\ \vdots \\ x_N - \frac{2}{N} \sum_i x_i \end{bmatrix}, \quad (4.40)$$

for any input vector \mathbf{x} . We will see in Sec. 4.4.2 that \mathbf{u} can be interpreted as the specific vector about which an input vector is reflected by the matrix \mathbf{A} in an N -dimensional space. A more general formulation may be obtained by replacing the identity matrix in Eq. (4.40) with any $N \times N$ permutation matrix.

The explicit expression for $\mathbf{A} \mathbf{x}$ in Eq. (4.40) shows that applying a Householder matrix to a vector requires $N - 1$ additions and one multiplication to obtain the term $\frac{2}{N} \sum_i x_i$, plus N additions to subtract this term from \mathbf{x} . Therefore the matrix-times-vector operation is only $\mathcal{O}(N)$ as opposed to the usual $\mathcal{O}(N^2)$.

Another interesting feature of the Householder feedback matrix is that \mathbf{A} does not have null entries for $N \neq 2$. This is a desirable property since it implies that every delay line feeds back to every other delay line, reinforcing the build-up of reflection density. The case $N = 4$ is especially nice, since the matrix entries all have the same magnitude and \mathbf{A} is therefore “balanced”. For larger N the diagonal becomes larger than the off-diagonal elements, and \mathbf{A} approaches a diagonal matrix as $N \rightarrow \infty$. Due to

the elegant balance of the $N = 4$ case, a larger ($N = 16$) feedback matrix can be constructed as follows:

$$\mathbf{A} = \frac{1}{2} \begin{bmatrix} \mathbf{A}_4 & -\mathbf{A}_4 & -\mathbf{A}_4 & -\mathbf{A}_4 \\ -\mathbf{A}_4 & \mathbf{A}_4 & -\mathbf{A}_4 & -\mathbf{A}_4 \\ -\mathbf{A}_4 & -\mathbf{A}_4 & \mathbf{A}_4 & -\mathbf{A}_4 \\ -\mathbf{A}_4 & -\mathbf{A}_4 & -\mathbf{A}_4 & \mathbf{A}_4 \end{bmatrix}, \quad \text{where } \mathbf{A}_4 := \frac{1}{2} \begin{bmatrix} 1 & -1 & -1 & -1 \\ -1 & 1 & -1 & -1 \\ -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & 1 \end{bmatrix} \quad (4.41)$$

is the 4×4 Householder matrix.

Other types of unitary matrices may be used. In particular, unitary feedback matrices can be derived from Hadamard matrices. A Hadamard matrix \mathbf{H} is defined as an $N \times N$, $(-1, 1)$ -matrix (i.e. a matrix whose elements consist only of the numbers -1 or 1) with the additional property that $\mathbf{H}\mathbf{H}^T = N\mathbb{I}$. This means that $\mathbf{A} = \mathbf{H}/\sqrt{N}$ is an orthogonal matrix whose entries all have the same magnitude $1/\sqrt{N}$. In Sec. 4.4.2 we discuss other classes of feedback matrices.

4.4.1.4 Designing lossy components

So far we have designed the lossless prototype. Now we have to correct it by inserting the low-pass filters $H_{lp,i}$ and the correction filter E . The $H_{lp,i}$'s set the reverberation time from infinity to a finite value, by moving the poles slightly inside the unit circle. More precisely, they can be chosen to tune the reverberator to a desired, frequency-dependent reverberation time $T_r(\omega)$.

The following analysis assumes that the filters $H_{lp,i}$ are all defined as $H_{lp,i}(z) = [G(z)]^{m_i}$. This is conceptually equivalent to substituting each delay z^{-1} in the lines with a “damped delay” $G(z)z^{-1}$, where the factor $G(z)$ represents a *filtering per sample* in the propagation medium. We also make the simplifying hypotheses that (1) the response $G(e^{j\omega})$ is zero-phase and that (2) the magnitude $|G(e^{j\omega})|$ is close to 1. Now assume that the lossless prototype has poles $e^{j\omega_i/F_s}$, $i = 1, \dots, N$, then the insertion of the low-pass filters moves the poles to

$$p_i \approx R_i e^{j\omega_i/F_s}, \quad \text{with } R_i = G\left(R_i e^{j\omega_i/F_s}\right) \approx G\left(e^{j\omega_i/F_s}\right), \quad (4.42)$$

where we have exploited our first simplifying hypothesis in assuming that the filters affect the radius of the poles and not their angles, and we have exploited our second simplifying hypothesis in the last approximation for R_i .

We know that the component of the impulse response arising from the i th pole of the system decays like R_i^n , as a function of discrete time n . Therefore the time needed for this response to decay by 60 dB (i.e. $T_r(\omega_i)$) satisfies the relation $20 \log_{10} (R_i^{T_r(\omega_i)F_s}) = -60$ dB. From Eq. (4.42), and recalling that $H_{lp,i} = G^{m_i}$, we conclude that the ideal low-pass filter satisfies the relation

$$20 \log_{10} \left| H_{lp,i} \left(e^{j\omega_i/F_s} \right) \right| = -60 \frac{m_i}{F_s T_r(\omega_i)}. \quad (4.43)$$

Having been derived in the assumption of zero-phase, this expression disregards the phase response of the $H_{lp,i}$'s, which has the effect of slightly modifying the effective length of the delay m_i . It is usually assumed that in practice this correction has no perceivable effect and can therefore be ignored.

A consequence of incorporating the filters $H_{lp,i}(z)$ into the delay lines is that the energy of each decaying mode of the system response will be affected, i.e. the envelope of the frequency response of the system will no longer be flat. In particular, for exponentially decaying reverberation the envelope is proportional to the reverberation time at all frequencies. The role of the filter $E(z)$ (often referred to as the *tonal correction filter*) is to compensate for this effect: a flat frequency response envelope is restored if the magnitude response of $E(z)$ is inversely proportional to the reverberation time:

$$\left| E \left(e^{j\omega/F_s} \right) \right| \sim \frac{1}{\sqrt{T_r(\omega)}}. \quad (4.44)$$



Having specified ideal filter responses for the $H_{lp,i}$'s and for E , any number of filter-design methods can be used to find low-order filters that reasonably approximate Eqs. (4.43, 4.44). Note that this design effectively decouples the control over reverberation time from the overall reverberator gain.

M-4.65

Write a function that computes filter coefficients for $H_{lp,i}(z)$ and $E(z)$, given a function $T_r(\omega)$ specified on a set of points $\{\omega_k\}$, and given the filter order k .

Since the function $T_r(\omega)$ is typically very smooth and slowly varying with respect to ω , the filters $H_{lp,i}(z)$ can be chosen to have low order. In particular, first-order filters of the form (4.33) can be used:

$$H_{lp,i}(z) = \frac{g_{1,i}}{1 - g_{2,i}z^{-1}}. \quad (4.45)$$

In this case one can use Eq. (4.43) to find the gains (we only report results):

$$g_{2,i} = \frac{\ln(10)}{4} \log_{10}(a_i) \left(1 - \frac{T_r(0)^2}{T_r(\pi F_s)^2} \right), \quad g_{1,i} = a_i(1 - g_{2,i}) \quad (4.46)$$

where $a_i = 10^{-\frac{m_i}{F_s T_r(0)}}$ is determined from the desired reverberation time at $\omega = 0$, while $g_{2,i}$ sets the reverberation time at high frequencies.

If first-order low-pass filters of the form (4.45) are used, then one can use a correction filter which is also first-order and is determined as follows (we only report results):

$$E(z) = \frac{1 - bz^{-1}}{1 - b}, \quad \text{with } b = \frac{1 - \frac{T_r(\pi F_s)}{T_r(0)}}{1 + \frac{T_r(\pi F_s)}{T_r(0)}}. \quad (4.47)$$

M-4.66

Write a function that computes filter coefficients for $H_{lp,i}(z)$ and $E(z)$ in the first-order case described above, given a function $T_r(\omega)$ specified on a set of points $\{\omega_k\}$.

M-4.66 Solution

```

function [num_H,den_H,num_E,den_E] = lossy_fdn(Fs,Tr,m);
%computes (first-order) lowpass filters H and correction filter E for a fdn,
%given an array of frequency-dependent Tr and the array m of fdn delays

N=length(m);
for i=1:N
    a=10^(-3*m(i)/(Fs*Tr(1)));
    g2=(log(10)/4)*log10(a)*(1-(Tr(1)^2/Tr(length(Tr))^2));
    num_H(i,:)=[a*(1-g2), 0];
    den_H(i,:)=[1, -g2];
end
b=(1-(Tr(length(Tr))/Tr(1)))/(1+(Tr(length(Tr))/Tr(1)));
num_E=[1, -b]; den_E=[1-b, 0];

```



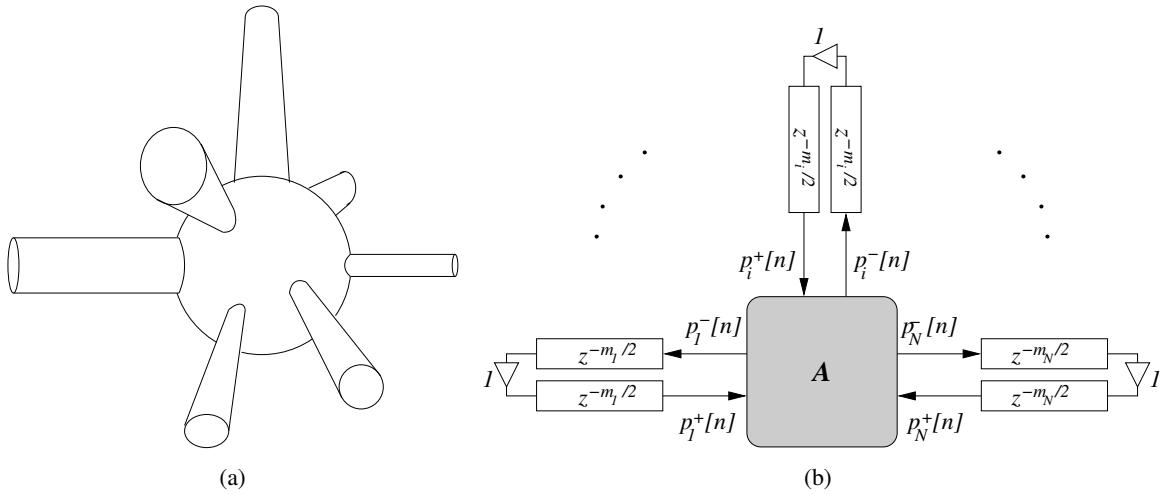


Figure 4.17: DWN reverberator

4.4.2 Digital waveguide networks

4.4.2.1 The link between FDNs and DWNs

In Eq. (4.40) we have introduced a specific Householder reflection matrix, constructed from the reference vector $\mathbf{u} = [1, \dots, 1]^T$. In fact a Householder matrix can be constructed given any reference vector \mathbf{u} . We now want to provide a geometric interpretation of this family of matrices.

Consider the *projection matrix* \mathbf{P}_u , which orthogonally projects any vector \mathbf{x} onto the vector \mathbf{u} :

$$\mathbf{P}_u = \frac{\mathbf{u} \mathbf{u}^T}{\|\mathbf{u}\|^2} = \frac{\mathbf{u} \mathbf{u}^T}{\mathbf{u}^T \mathbf{u}}, \quad \text{then} \quad \mathbf{x}_u := \mathbf{P}_u \mathbf{x} = \mathbf{u} \frac{\langle \mathbf{u}, \mathbf{x} \rangle}{\|\mathbf{u}\|^2} \quad (4.48)$$

is the orthogonal projection of \mathbf{x} onto \mathbf{u} . Now consider the vector $\mathbf{x}_u^\perp := (\mathbb{I} - \mathbf{P}_u)\mathbf{x}$: this is the projection of \mathbf{x} onto the hyperplane orthogonal to \mathbf{u} , since it is easily verified that $\mathbf{x}_u^\perp \perp \mathbf{x}_u$ and that $\mathbf{x}_u^\perp + \mathbf{x}_u = \mathbf{x}$.

Finally consider the vector \mathbf{y} obtained by *reflecting* \mathbf{x} about \mathbf{u} . Elementary geometrical considerations allow to conclude that this vector is the difference between \mathbf{x}_u and \mathbf{x}_u^\perp :

$$\mathbf{y} = \mathbf{x}_u - \mathbf{x}_u^\perp = \mathbf{P}_u \mathbf{x} - (\mathbb{I} - \mathbf{P}_u)\mathbf{x} = (2\mathbf{P}_u - \mathbb{I})\mathbf{x}. \quad (4.49)$$

The matrix $(2\mathbf{P}_u - \mathbb{I})$ is a Householder matrix as defined in Eq. (4.40), except for a sign. Therefore we conclude that given a reference vector \mathbf{u} the corresponding Householder matrix reflects any vector \mathbf{x} about \mathbf{u} .

Having understood the meaning of Householder matrices, we now construct a digital waveguide network (DWN) that is equivalent to the FDN lossless prototypes considered in the previous section. We start by considering the physical resonator depicted in Fig. 4.17(a). It is composed by N acoustic bores connected in parallel. In Chapter *Sound modeling: source based approaches* we have derived the $N \times N$ scattering matrix \mathbf{A} that relates the incoming pressure waves \mathbf{p}^+ to the outgoing pressure waves \mathbf{p}^- . In this section we reconsider that matrix when the pressure waves in the i th bore are defined as

$$p_i^+ = \sqrt{\Gamma_i} \frac{p_i + Z_i u_i}{2}, \quad p_i^- = \sqrt{\Gamma_i} \frac{p_i - Z_i u_i}{2}, \quad (4.50)$$

where Z_i and $\Gamma_i = 1/Z_i$ are the wave impedance and admittance of the i th bore. These are often referred to as *normalized* waves, and differ from our previous definition of wave variables uniquely for the scaling



factor $\sqrt{\Gamma_i}$. It is straightforward to see that normalized pressure waves are scattered as $\mathbf{p}^- = \mathbf{A}\mathbf{p}^+$, where

$$\mathbf{A} = \begin{bmatrix} \frac{2\Gamma_1}{\Gamma_J} - 1, & \frac{2\sqrt{\Gamma_1\Gamma_2}}{\Gamma_J}, & \dots & \frac{2\sqrt{\Gamma_1\Gamma_N}}{\Gamma_J} \\ \frac{2\sqrt{\Gamma_2\Gamma_1}}{\Gamma_J}, & \frac{2\Gamma_2}{\Gamma_J} - 1, & \dots & \frac{2\sqrt{\Gamma_2\Gamma_N}}{\Gamma_J} \\ \vdots & \ddots & \ddots & \vdots \\ \frac{2\sqrt{\Gamma_N\Gamma_1}}{\Gamma_J}, & \frac{2\sqrt{\Gamma_N\Gamma_2}}{\Gamma_J}, & \dots & \frac{2\Gamma_N}{\Gamma_J} - 1 \end{bmatrix}, \quad \text{where } \Gamma_J = \sum_{l=1}^N \Gamma_l. \quad (4.51)$$

This normalized scattering matrix is immediately recognized as a Householder matrix:

$$\mathbf{A} = \frac{2}{\|\boldsymbol{\Gamma}\|} \boldsymbol{\Gamma} \boldsymbol{\Gamma}^T - \mathbb{I}, \quad \text{with } \boldsymbol{\Gamma} := [\sqrt{\Gamma_1}, \sqrt{\Gamma_2}, \dots, \sqrt{\Gamma_N}], \quad (4.52)$$

so we have this interesting geometrical interpretation: scattering of normalized pressure waves corresponds to a reflection around the vector $\boldsymbol{\Gamma}$.

If the acoustic bores are lossless and with ideal closed terminations, and if the length (in samples) of the i th bore is $m_i/2$, then the physical resonator of Fig. 4.17(a) can be modeled with the digital waveguide network given in Fig. 4.17(b). Now compare this scheme with the lossless FDN of Fig. 4.16: apart from the input signals $x_i[n]$, the two schemes implement the same computational structure. The incoming pressure waves $p_i^+[n]$ correspond to the output signals $y_i[n]$, and the outgoing pressure waves $p_i^-[n]$ correspond to the feedback signals generated by the feedback matrix.

4.4.2.2 General lossless scattering matrices

Showing the equivalence between DWNs and FDNs is more than a mere intellectual exercise: we can now design an entire new class of lossless FDN prototypes, in which the feedback matrix \mathbf{A} is given by Eq. (4.51) and have a straightforward physical interpretation.

Note that the matrix in Eq. (4.51) is still orthogonal (it is easy to verify that $\mathbf{A}\mathbf{A}^T = \mathbb{I}$). We can push the generalization further by generalizing our definition of losslessness, and consequently define new classes of lossless feedback matrices that are neither physical nor orthogonal. Consider a Hermitian, positive-definite $N \times N$ matrix $\boldsymbol{\Gamma}$ (we use this notation because we interpret $\boldsymbol{\Gamma}$ as a generalized junction admittance). This matrix induces a norm $\|\cdot\|_\Gamma$, defined as follows: $\|\mathbf{x}\|_\Gamma := \mathbf{x}^T \boldsymbol{\Gamma} \mathbf{x}$ for any real valued N -dimensional vector \mathbf{x} . We can then define a waveguide scattering matrix \mathbf{A} to be “lossless” if the scattering preserves the norm, i.e. the equality $\|\mathbf{p}^+\|_\Gamma = \|\mathbf{p}^-\|_\Gamma$ holds. This condition is clearly equivalent to the condition

$$\mathbf{A}^T \boldsymbol{\Gamma} \mathbf{A} = \boldsymbol{\Gamma} \quad (4.53)$$

for the scattering matrix \mathbf{A} . In the case $\boldsymbol{\Gamma} = \mathbb{I}$, the norm $\|\cdot\|_\Gamma$ is the euclidean norm and the above equation reduces to the condition of \mathbf{A} being orthogonal. In the general case $\boldsymbol{\Gamma} \neq \mathbb{I}$ it can be shown that Eq. (4.53) holds if and only if \mathbf{A} has eigenvalues with modulus 1 and N linearly independent eigenvectors. We do not provide a proof of this characterization: intuitively it means that when such a feedback matrix is used in a lossless FDN prototype the system poles all have unit modulus and thus the system response consists of non-decaying eigenmodes.

Clearly orthogonal matrices are lossless in this sense, since they have unitary eigenvalues and pairwise orthogonal eigenvectors. Another class of matrices that satisfy this condition are triangular matrices: designing a triangular matrix with unitary eigenvalues is straightforward since we know from linear algebra that they lie on the diagonal. Additional care is required in order to ensure that the triangular matrix possesses N independent eigenvectors.



4.4.2.3 Waveguide meshes

So far we have seen DWNs in analogy with FDNs. In this section we discuss a new multidimensional waveguide structure, named *waveguide mesh*, that can be used to physically simulate resonating enclosures. What follows is only a quick and qualitative introduction to the subject, the interested reader can refer to the bibliography.

Consider again the N-D D'Alembert equation (4.1). Similarly to what we have done in the 1-D case (Chapter *Sound modeling: source based approaches*), we can simulate the traveling wave solution by using delay lines. In this case the delay lines are arranged in a mesh, that represents waves propagating in the x, y, z directions. At each node of the mesh continuity constraints must be satisfied, namely the pressure waves in each direction must provide the same pressure value.⁵ This means that at each node of the mesh the incoming pressure waves are scattered by a matrix identical to the matrix \mathbf{A} given in Chapter *Sound modeling: source based approaches*, in which all the incoming branches share the same impedance:

$$\mathbf{A} = \begin{bmatrix} \frac{2}{N} - 1 & \frac{2}{N} & \dots & \frac{2}{N} \\ \frac{2}{N} & \frac{2}{N} - 1 & \dots & \frac{2}{N} \\ \vdots & \ddots & \ddots & \vdots \\ \frac{2}{N} & \frac{2}{N} & \dots & \frac{2}{N} - 1 \end{bmatrix}. \quad (4.54)$$

In order to clarify this idea, let us examine the 2-D case shown in Fig. 4.18. The outgoing pressure waves at each node are computed as $\mathbf{p}^- = \mathbf{A}\mathbf{p}^+$, i.e.

$$p_i^-[n] = p_J[n] - p_i^+[n] \quad (i = 1 \dots 4) \quad \text{where } p_J[n] = \frac{\sum_{i=1}^4 p_i^+[n]}{2} \quad (4.55)$$

is the junction pressure. It can be shown that this rectangular waveguide mesh is equivalent to a finite-difference numerical solution of the the 2-D D'alembert equation, in which the pressure at a certain node is expressed in terms of the pressures at its neighboring nodes one sample earlier, and itself two samples earlier.

The rectangular layout depicted in Fig. 4.18 is not the only possible one: other geometries may be used for assembling the mesh, like triangular, hexagonal, and so on. The choice of the geometry has a major influence on the *dispersion* error in the mesh, i.e the error in propagation speed as a function of frequency and direction along the mesh. It can be shown that the triangular waveguide mesh is the simplest 2-D mesh geometry with the least dispersion variation as a function of direction of propagation. In other words, the triangular mesh is closer to isotropic than all other known elementary geometries. Isotropy can be obtained also through interpolation, i.e. by using non integer propagation delays, but computational costs are higher. As far as frequency dispersion is concerned, frequency-warping methods can be used to minimize it in the mesh.

The waveguide meshes analyzed so far simulates lossless propagation in an infinite medium. In order to model something similar to a real resonating enclosure we must add losses and boundary conditions into the structure. The techniques discussed in Chapter *Sound modeling: source based approaches* to simulate lossless in 1-D wave propagation can be extended to the waveguide mesh: the basic idea is once again that wave propagation during one sampling interval (in time) is associated with linear filtering by $G(z)$. The problem of modeling mesh boundaries is particularly important in the context of artificial reverberation: in order to obtain high temporal reflection densities, maximally *diffusing* boundaries have to be modeled.

As efficient solutions are found to deal with the above mentioned problems, 3-D waveguide meshes are being more and more used for the simulation of acoustic spaces.

⁵In this section we are using waveguide meshes to simulate resonating enclosures and thus we work with pressure waves and consider parallel junctions. Waveguide meshes can also be used to simulate mechanical resonators, e.g membranes, and in that case it is natural to choose velocity waves and to consider series junctions at mesh nodes.



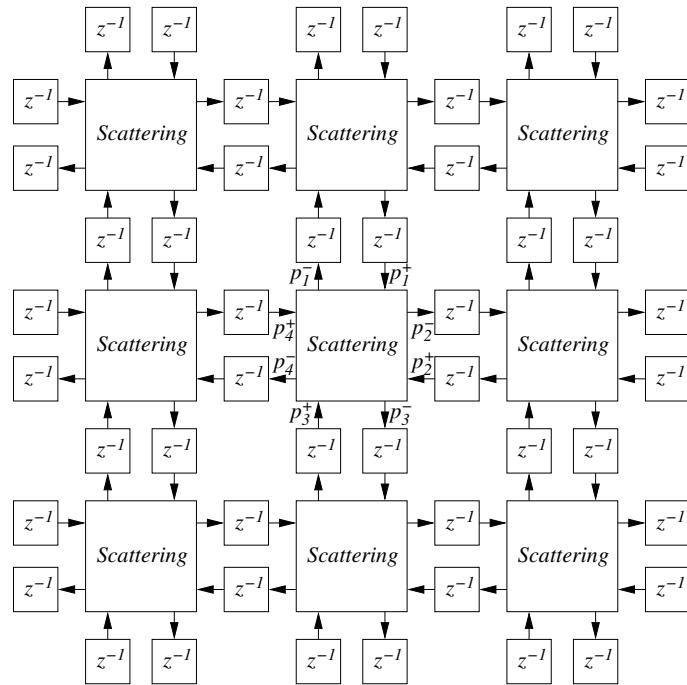


Figure 4.18: 2D rectilinear digital waveguide mesh.

4.5 Spatial hearing

In the previous sections we have learned how a sound produced by an acoustic source is affected by the surrounding environment. So far we have assumed that the receiver is a point in the space, which is reasonable e.g. for a omnidirectional microphone. We now want to study a different type of receiver, i.e. a human receiver with two ears and one head in between.

Throughout the next sections our assumption will be that the two acoustic pressure signals at the two eardrums contain all the information that is used by a human listener to elaborate his/her auditory perception. In other words we assume that if different acoustic events (e.g. different sounds, or different sound/listener positions in the environment, etc.) produce the same pair of acoustic pressures at the eardrums, they will be perceived by a human listener as the same acoustic event.⁶ In particular these signal will provide the listener with *spatial information*, about the location of the sound source relative to the listener.

With this assumption, our goal is to understand and simulate how sound is transformed in his path to the eardrum by neighboring parts of the body (such as head and shoulders), by the pinna (the visible portion of the outer ear), and by the ear canal (the meatus that leads to the eardrum).

4.5.1 The sound field at the eardrum

Spatial attributes of the sound field are coded into temporal and spectral attributes of the acoustic pressure at the eardrum, via the filtering effect of three main elements: head, external ear, and torso/shoulders.

⁶In fact this is not entirely correct. Sound reaches our ears also through bone conduction. Moreover auditory perception interacts with other types of information (e.g., conflicting visual cues) and is affected by adaptation and expectations.



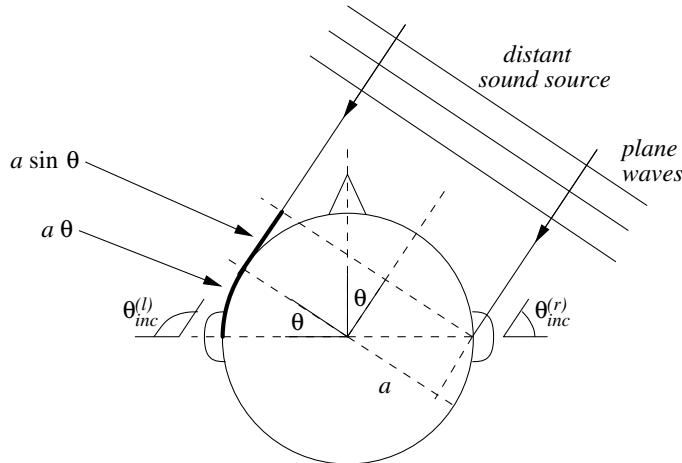


Figure 4.19: Estimate of ITD in the case of a distance sound source (plane waves) and spherical head.

4.5.1.1 Head

Our ears are not isolated objects in space. They are located, at the same height, on opposite sides of an acoustically rigid object: the head. This acts as an obstacle to the free propagation of sound and has two main effects: (1) it introduces an *interaural time difference* (ITD), because a sound wave has to travel an extra distance in order to reach the farthest ear, and (2) it introduces an *interaural level difference* (ILD) because the farthest ear is acoustically “shadowed” by the presence of the head.

An approximate yet quite accurate description of the ITD can be derived using a few simplifying assumptions, in particular by considering the case of “distant” sound sources and a spherical head: this situation is depicted in Fig. 4.19. The first assumption implies that the sound waves that strike the head are plane waves. Then the extra-distance Δx needed for a sound ray to reach the farthest ear is estimated from elementary geometrical considerations, as shown in Fig. 4.19, and the ITD is simply $\Delta x/c$. Therefore

$$\text{ITD} \sim \frac{a}{c}(\theta + \sin \theta), \quad (4.56)$$

where a is the head radius and θ is the *azimuth* angle that defines the direction of the incoming sound on the horizontal plane. This formula shows that the ITD is zero when the source is directly ahead ($\theta = 0$), and is a maximum of $a/c(\pi/2 + 1)$ when the source is off to one side ($\theta = \pi/2$). This represents an ITD of more than 0.6 ms for a head radius $a = 8.5$ cm, which is a realistic value.

While it is acceptable to approximate the ITD as a frequency independent parameter, as we did in Eq. (4.56), the ILD is highly frequency dependent: at low frequencies (i.e., for wavelengths that are long relative to the head diameter) there is hardly any difference in sound pressure at the two ears, while at high frequencies differences become very significant. Again, the ILD can be studied in the case of an ideal spherical head of radius a , with a point sound source located at a distance $r > a$ from the center of the sphere. It is customary to use the normalized variables $\mu = \omega a/c$ (normalized frequency) and $\rho = r/a$ (normalized distance). If we consider a point on the sphere, then the diffraction of an acoustic wave by the sphere seen on the chosen point is expressed with the transfer function

$$H_{\text{sphere}}(\rho, \theta_{\text{inc}}, \mu) = -\frac{\rho}{\mu} e^{-i\mu\rho} \sum_{m=0}^{+\infty} (2m+1) P_m(\cos \theta_{\text{inc}}) \frac{h_m(\mu\rho)}{h'_m(\mu)}, \quad (4.57)$$

where P_m and h_m are the m th order Legendre polynomial and spherical Hankel function, respectively,

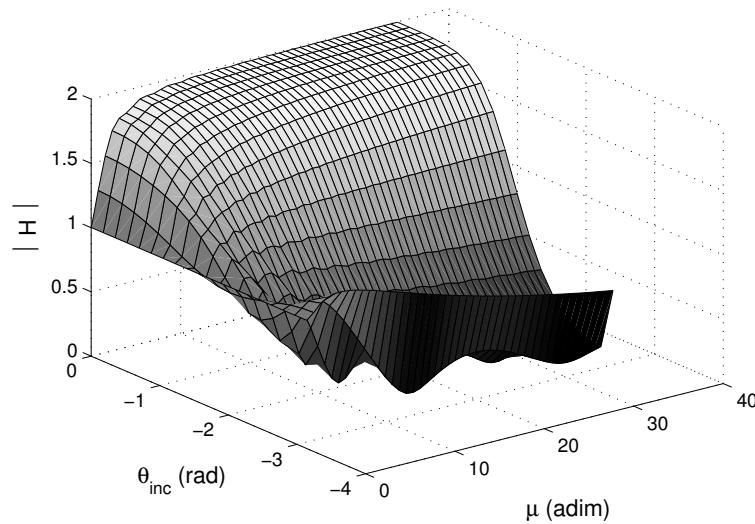


Figure 4.20: Magnitude response $|H_{\text{sphere}}(\infty, \theta_{\text{inc}}, \mu)|$ of a sphere for an infinitely distant source.

and θ_{inc} is the angle of incidence, i.e. the angle between the ray from the center of the sphere to the source and the ray to the measurement point on the surface of the sphere.⁷ Normal incidence corresponds to $\theta_{\text{inc}} = 0$, while the sphere point opposite to the source is at $\theta_{\text{inc}} = \pi$.

It is known that the Hankel function $h_m(x)$ admits an asymptotic approximation as the argument x goes to infinity. By exploiting this approximation one can study the behavior of the transfer function $H_{\text{sphere}}(\infty, \theta_{\text{inc}}, \mu)$ as the distance r between the source and the sphere becomes arbitrarily large. The approximate solution $|H_{\text{sphere}}(\infty, \theta_{\text{inc}}, \mu)|$ is plotted in Fig. 4.20.

At low frequencies the transfer function is not directionally dependent and the magnitude $|H_{\text{sphere}}|$ is essentially unity for any angle θ_{inc} . When μ exceeds 1 the dependence on θ_{inc} becomes noticeable. The response increases around the front of the sphere, and in particular exhibits a 6 dB boost at high frequencies near the front of the sphere ($|H_{\text{sphere}}(\infty, 0, \infty)| = 2$), consistently with the requirement that in this limit the solution must reduce to that of a plane wave normally incident on a rigid plane surface. $|H_{\text{sphere}}|$ is approximately flat when θ_{inc} is around 100 degrees, and progressively decreases around the back of the sphere. Note however that the minimum response does not occur at the very back ($\theta_{\text{inc}} = \pi$). Instead, this point exhibits a so-called “bright spot” effect, which is due to the fact that all the waves propagating around the sphere arrive at that point in phase. At very high frequencies the bright-spot lobe becomes extremely narrow, and the back of the sphere is effectively in a sound shadow. Finally, note that interference effects caused by waves propagating in various directions around the sphere introduce ripples in the response that are quite prominent on the shadowed side.

4.5.1.2 The external ear

The external ear consists of the *pinna* and the *ear canal* until the eardrum. Beyond the eardrum are the middle ear and the internal ear. For the purpose of this chapter we are interested in the external ear only. In Chapter *Auditory based processing* we will study the middle and internal ear.

The pinna, schematically depicted in Fig. 4.21(a), has a characteristic “bas-relief” form with features

⁷We are using a different notation with respect to the azimuth angle θ used previously, in order to avoid confusion. Given a 2-D reference system like that in Fig. 4.19, the transfer functions (4.57) at the right and left ear will use the angles $\theta_{\text{inc}}^{(r)} = \pi/2 - \theta$ and $\theta_{\text{inc}}^{(l)} = \pi/2 + \theta$, respectively.

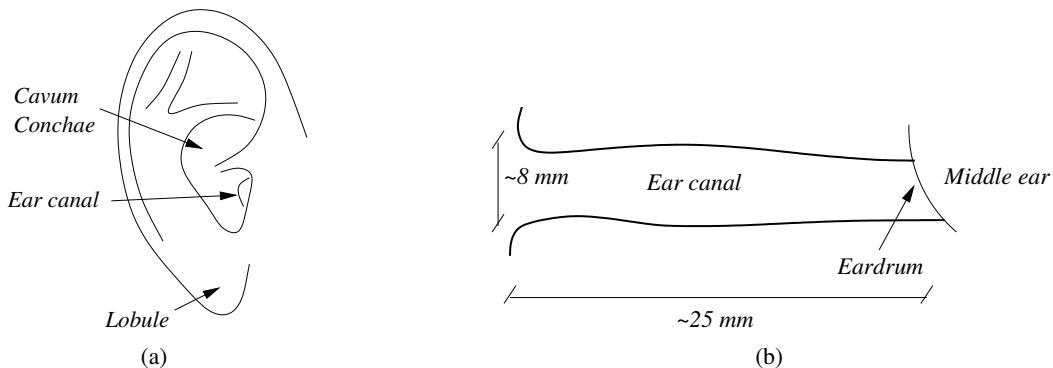


Figure 4.21: External ear: (a) pinna, and (b) ear canal.

that differ greatly from one individual to another (just look at people's ears). The pinna is connected to the ear canal, depicted in Fig. 4.21(b). It can be approximately described as a tube of constant width, with walls of high acoustic impedance. At the end opposite to the pinna, the ear canal is terminated by the eardrum diaphragm.

At a first approximation the acoustic behaviour of the ear canal is easily understood: it behaves like a one-dimensional resonator. On the other hand the pinna has much more complex effects, as it basically acts like an acoustic antenna. Its resonant cavities amplify some frequencies, and its geometry leads to interference effects that attenuate other frequencies. Moreover, its frequency response is directionally dependent. Acoustically it acts like a filter whose transfer function depends in general on the distance and direction of the sound source relative to the ear. Like for any other resonator, we can interpret these filtering effect either by looking at reflections of sound rays or in the frequency domain.

First approach: external ear as a sound reflector. Figure 4.22(a) shows two different directions of arrival. In each case there are two paths from the source to the ear canal – a direct path and a longer path following a reflection from the pinna. At moderately low frequencies, the pinna essentially collects additional sound energy, and the signals from the two paths arrive in phase. However, at high frequencies, the delayed signal is out of phase with the direct signal, and destructive interference occurs. The greatest interference occurs when the difference in path length is a half wavelength: this produces a “pinna notch”. Since the pinna is a more effective reflector for sounds coming from the front than for sounds from above, the resulting notch is much more pronounced for sources in front than for sources above. In addition, the path length difference changes with elevation.

However reflection models are suspect whenever the dimensions of the reflecting surfaces are comparable to (or even smaller than) the acoustic wavelengths under exam. At the very least, the reflection coefficients should be frequency dependent. A more thorough approach is modal analysis of the external ear resonator, through measurements of frequency responses using an imitation pinna and a ear canal with high impedance termination. Such measurements give results like those depicted in Fig. 4.22(b). First resonance is that of a open-closed tube $\sim 33\%$ longer than the ear canal: the pinna acts as a prolongation of the ear canal with an aperture effect. Second resonance is a resonance of the *cavum concha* alone: the pressure distribution is similar to what would be obtained if the canal were plugged. The higher resonances instead are again associated to longitudinal standing waves: these are not very widely spaced and are quite dependent on the individual, therefore it can combine in a single broad peak of the magnitude response.

The synthetic conclusion of this section is then that the pinna and the ear canal form a systems of acoustic resonators, whose resonances are excited to different extents depending on the direction and

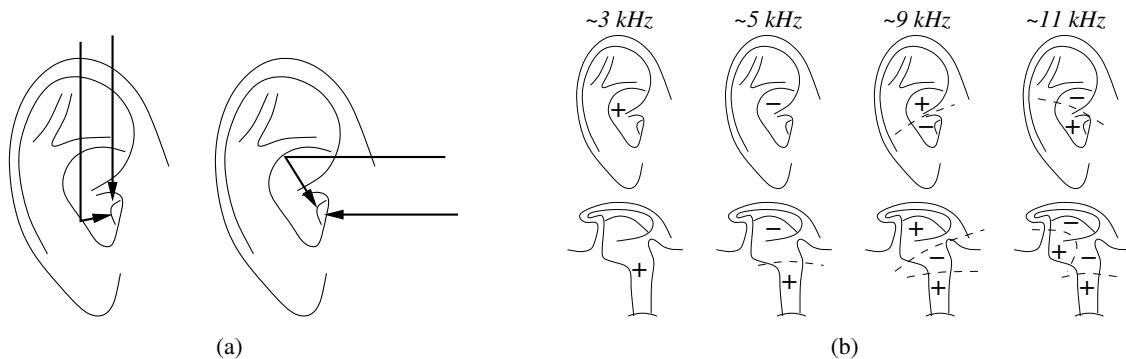


Figure 4.22: Effects of pinna: (a) direction-dependent reflections, and (b) resonances.

distance of the sound source.

4.5.1.3 Torso and shoulders

In the discussion up to now we have not considered a third element that, together with the head and the external ear, contributes to the shaping of the sound field at the eardrum: the torso. Torso and shoulders affect incident sound waves in two main respects. First, they provide additional reflections that sum up with the direct sound. Second, they have a shadowing effect for sound rays coming from below.

The geometry of the torso is quite complicated. However a simplified description can be derived by considering an ellipsoidal torso below a spherical head. These kind of approximate descriptions are sometimes called “snowman models”, for obvious reasons. Figure 4.23(a) depicts a snowman model and shows the main effects of the ellipsoidal torso on the sound field at the ear.

Reflections: Fig. 4.23(a). If we measured the impulse response at the right ear for the sound source locations depicted in Fig. 4.23(a) we would see that the initial pulse is followed by a series of subsequent pulses, whose delays increase and then decrease with elevation. These additional pulses are caused by reflections on the torso.

We could exploit the simplified geometry of the snowman model to compute analytically the delay of the reflected rays, given the model parameters and the sound source position. However some important remarks can already be made from a qualitative analysis. First, the delay between the direct sound and the reflected ray does not vary much if the sound source moves on a circumference in the horizontal plane (especially if its radius is large compared to the head radius). Second, the delay varies considerably if the sound source moves vertically, and in particular the reflected pulses are maximally delayed for sound source locations right above the listener. If we consider that the distance from the ear canal to the shoulder is roughly 16 cm, then a reflected ray from a source right above the subject has to travel an extra distance of approximately 32 cm, which corresponds to a delay of almost 1 ms.

In the frequency domain the torso reflections act as a comb filter, introducing periodic notches in the spectrum. The frequencies at which the notches occur are inversely related to the delays, and thus produce a pattern that varies with the elevation of the source. The lowest notch frequency corresponds to the longest delay. Delays longer than a sixth of a millisecond will produce one or more notches below 3 kHz, which is approximately the lowest frequency where pinna effects start to be noticeable.

Modeling the effects of the torso as specular reflections means accounting for only a part of the story. First, reflection is a high frequency concept. Second, and perhaps more important, as the source descends in elevation, a point of grazing incidence is reached, below which torso reflections disappear

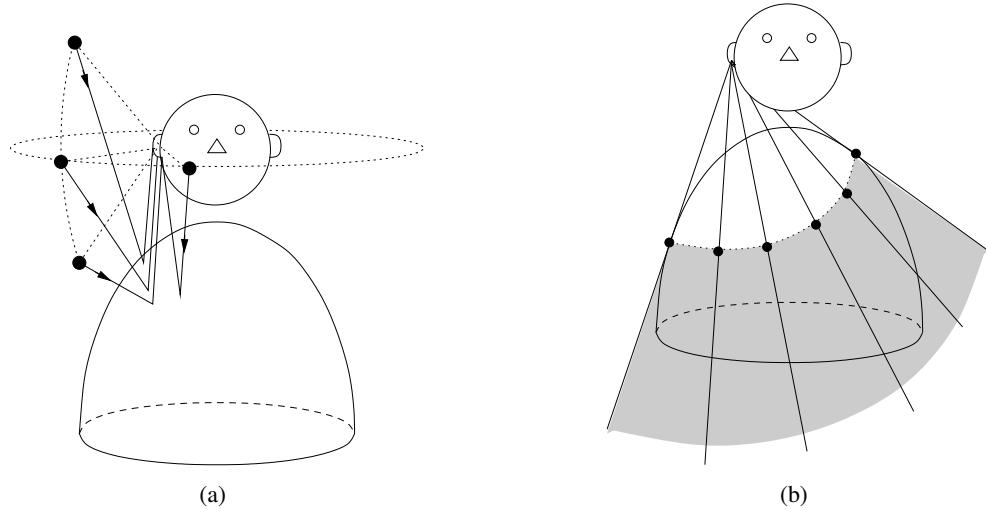


Figure 4.23: Effects of torso: (a) reflections, and (b) shadowing.

and *torso shadowing* emerges. As shown in Fig. 4.23(b), rays drawn from the ear to points of tangency around the upper torso define a torso-shadow cone. Clearly, the specular reflection model does not apply within the torso shadow cone. Instead, diffraction and scattering produce a qualitatively different behavior, characterized by a stronger attenuation for high frequencies (i.e. for wavelength comparable to or smaller than the size of the torso).

Although the acoustic effects of torso and shoulders are not as strong as those introduced by the pinna, they are important because they appear at lower frequencies, where typical sound signals have most of their energy and where the response of the pinna is essentially flat. In terms of frequency ranges the effects provided by the torso are therefore complementary to those provided by the pinna.

4.5.1.4 Head-related transfer functions

In the preceding sections we have investigated the influence of head, torso and external ear on the sound field at the eardrum. All the effects that we have examined are linear, which means that (1) they can be described by means of transfer functions, and (2) they combine additively. Therefore the sound pressure produced by an arbitrary sound source at the eardrum is uniquely determined by the impulse response from the source to the eardrum. This is called *Head-Related Impulse Response (HRIR)*, and its Fourier transform is called *Head Related Transfer Function (HRTF)*. The HRTF captures all of the physical effects that we have examined separately in the previous sections.

The HRTF is a function of three spatial coordinates and frequency. Given the approximately spherical shape of the head, it is customary to use the spherical coordinates depicted in Fig. 4.24, which use slightly different notations and conventions with respect to more traditional definition (see our definition of spherical coordinates in Chapter *Sound modeling: source based approaches*). Specifically, in this context the vertical and horizontal angular coordinates *azimuth* and *elevation* are noted as θ and ϕ , respectively, while the radial coordinate is named *range* and noted as r . Moreover, two different spherical coordinate systems are used in the literature. Figure 4.24(a) show the most popular one, sometimes called *vertical polar* coordinate system: in this system the azimuth is measured as the angle from the yz plane to a vertical plane containing the source and the z axis, and the elevation is measured as the angle up from the xy plane. With this choice, surfaces of constant azimuth are planes through the z axis, and surfaces

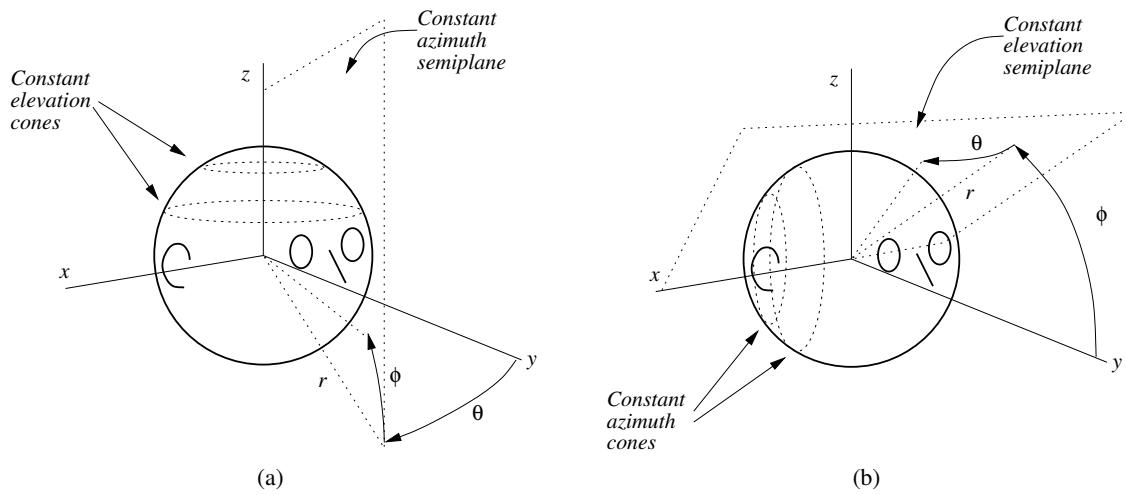


Figure 4.24: Spherical coordinate systems used in the definition of HRTFs: (a) vertical-polar coordinate system, and (b) interaural-polar coordinate system.

of constant elevation are cones concentric about the z axis.

In alternative the so-called *interaural-polar* coordinate system, shown in Fig. 4.24(b), is sometimes used. In this case the elevation is measured as the angle from the xy plane to a plane containing the source and the x axis, and the azimuth is then measured as the angle from the yz plane. With this choice, surfaces of constant elevation are planes through the x axis, and surfaces of constant azimuth are cones concentric with the x axis. One advantage of this system is that it makes it significantly simpler to express interaural differences at all elevations (in particular the constant-azimuth cones are the loci of points that share equals ILD and ITD values for a spherical head).

In the remainder of this chapter we will specify, when necessary, whether we are using the vertical-polar or the interaural-polar coordinate system. In any case we will indicate the HRTFs as $H^{(l),(r)}(r, \theta, \phi, \omega)$, where superscripts $(l), (r)$ indicate the HRTF at the left and right ear, respectively. When $r \rightarrow +\infty$ (which in practice means $r > 1$ m, a condition that is met in most applications), the source is said to be in the *far field*. In this case we will use the notation $H^{(l),(r)}(\theta, \phi, \omega)$. Finally, in the hypothesis of a perfectly symmetrical geometry will simply write $H(\theta, \phi, \omega)$, with $H^{(r)}(\theta, \phi, \omega) = H(\theta, \phi, \omega)$ and $H^{(l)}(\theta, \phi, \omega) = H(-\theta, \phi, \omega)$.

We formally define the HRTF at one ear as the frequency-dependent ratio between the sound pressure level (SPL) $\Phi^{(l),(r)}(\theta, \phi, \omega)$ at the corresponding eardrum and the free-field SPL at the center of the head $\Phi_f(\omega)$ as if the listener were absent:

$$H^{(l)}(\theta, \phi, \omega) = \frac{\Phi^{(l)}(\theta, \phi, \omega)}{\Phi_f(\omega)}, \quad H^{(r)}(\theta, \phi, \omega) = \frac{\Phi^{(r)}(\theta, \phi, \omega)}{\Phi_f(\omega)}. \quad (4.58)$$

Figures 4.25(a) and 4.25(b) show two examples of HRTFs (magnitude response only): all the effects examined in this section combine to form a surprisingly complicated function of θ and ϕ .

4.5.2 Perception of sound source location

This is complicate matter. Many competing and interfering effects can influence auditory perception of sound source location. In this section we provide a brief summary, but we warn the reader to be cautious when dealing with this matter and always to be aware of limitations and simplifying hypotheses.

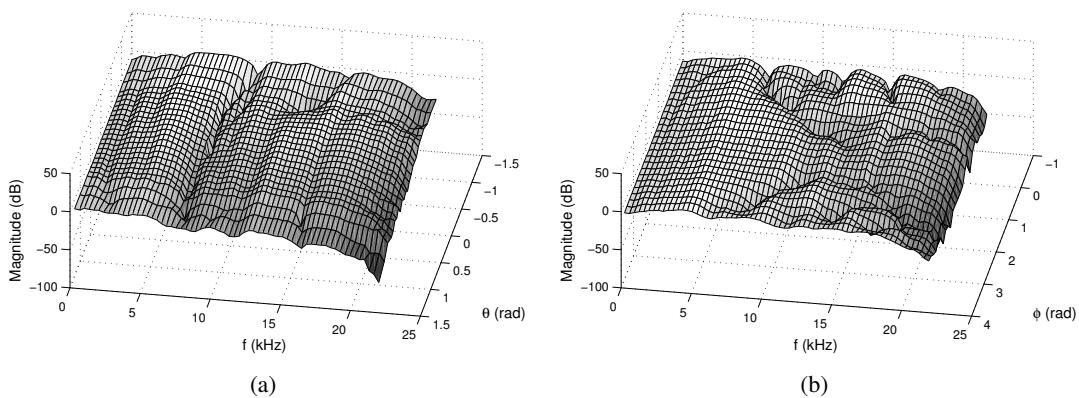


Figure 4.25: Example of magnitude of HRTFs (a) in the xy plane ($\theta \in [-\pi/2, \pi/2]$, $\phi = 0$) and (b) in the yz plane ($\theta = 0$, $\phi \in [-\pi/4, \pi]$). Interaural polar coordinates are used.

4.5.2.1 Azimuth perception

The horizontal placement of the ears maximizes differences for sound events occurring around the listener, rather than from below or above, enabling audition of sound sources at the terrain level and outside the visual field of view. The ITD and the ILD are considered to be the key parameters for azimuth perception, in what is sometimes referred to as the *Duplex Theory* of localization.

For the sake of clarity, consider a sine wave reaching the left and right ear. At low frequencies the ITD shifts the waveform a fraction of a cycle, which is easily detected: see Fig 4.26(a). Qualitatively one can say that if the half wavelength is larger than the size of the head, then it is possible for the auditory system to detect the phase of these waveforms unambiguously, and the ITD cue can function. On the other hand, at high frequencies there is ambiguity in the ITD, since there can be several cycles of shift: see Fig 4.26(b). Qualitatively, we can consider the critical point to be the point where the half wavelength becomes shorter than the head size: for shorter wavelengths, the phase information in relation to relative time of arrival at the ears can no longer convey which is the leading wavefront. The critical point in frequency is usually assumed to be a value around 1.5 kHz.

If we now look at the ILD the situation is reversed. As we have seen in Sec. 4.5.1 (see in particular Fig. 4.20), at low frequencies the head transfer function is essentially flat and therefore there is little ILD information. On the other hand, at high frequencies the ILD is more marked and can become very large. For this reason the Duplex Theory asserts that the ILD and the ITD are complementary cues to azimuth perception, and that taken together they provide azimuth perception throughout the audible frequency range.

This is not completely true, though. In fact timing information can be exploited for azimuth perception also in the high frequency range because the timing differences in amplitude envelopes are detected. Again, for the sake of clarity consider a sine wave that is modulated in amplitude as in Fig. 4.26(c). Then an ITD envelope cue, sometimes referred to as *Interaural Envelope Difference (IED)* can be exploited, based on the hearing system's extraction of the timing differences from the transients of amplitude envelopes, rather than from the timing of the waveform within the envelope. This is demonstrated by the so-called Franssen Effect: if a sine wave is suddenly turned on and a high-pass-filtered version is sent to a loudspeaker "A" while a low-pass filtered version is sent to a loudspeaker "B", most listeners will localize the sound at A. This is true even if the frequency of the sine wave is sufficiently low that in steady state most of the energy is coming from B.

The information provided by ITD and ILD can be ambiguous. If we assume the spherical geometry of

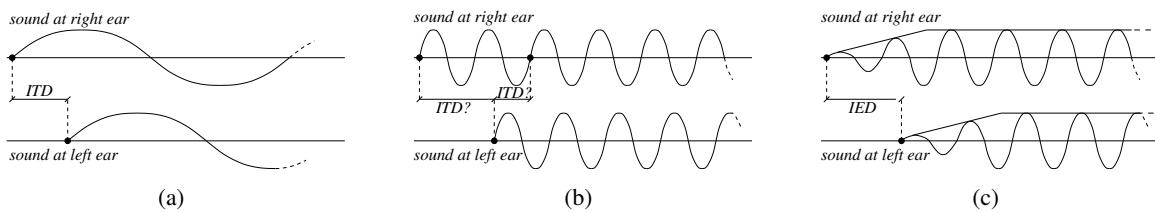


Figure 4.26: Time differences at the ears; (a) non ambiguous ITD, (b) ambiguous ITD, and (c) IED.

Fig. 4.19, a sound source located in front of the listener at a certain θ , and a second one located at the rear, at $\pi - \theta$, provide identical ITD and ILD values. In reality ITD and ILD will not be exactly identical at θ and $\pi - \theta$ because (1) human heads are not spherical, (2) there are asymmetries and other facial features, and (3) ears are not positioned as in Fig. 4.24 but lie below and behind the x axis. Nonetheless the values will be very similar, and *front-back confusion* is in fact often observed experimentally: listeners operate *reversals* in azimuth judgements, erroneously locating sources at the rear instead of at the front, or viceversa. The former reversal occurs more often than the latter. Some argue that this asymmetry may originate from a sort of ancestral “survival mechanism”, according to which if something (a predator?) can be heard but not seen then it must be at the rear (danger!).

The Duplex Theory essentially works in anechoic conditions. But in everyday conditions reverberation can severely degrade especially ITD information. As we know, in a typical room reflections begin to arrive a few milliseconds after the direct sound. Below a certain sound frequency, the first reflections reach the ear before one oscillation period is completed. Before the auditory system estimates the frequency of the incoming sound wave, and consequently infers the ITD, the number of reflections at the ear has increased exponentially and the auditory system is not able to estimate the ITD. Therefore sounds that possess energy in the low-frequency range only (indicatively below 250 Hz) are essentially impossible to localize in a reverberant environment.⁸ Instead the IED is used, because the starting transient provides unambiguous localization information, while the steady-state signal is very difficult to localize. In conclusion we can state –with some risk of oversimplification– that high-frequency energy only is important for localization in reverberant environments.

4.5.2.2 Lateralization and externalization

In Sec. 4.6 we will see that the simplest systems for spatial sound rendering are based on manipulation of the interaural cues examined above, and on headphone-based auditory display. These systems can be used in applications where only two-dimensional localization –in the horizontal plane– is required.

In this context, the term *lateralization* is typically used to indicate a special case of localization, where the spatial percept is heard inside the head, mostly along the interaural axis (the x of Fig. 4.24), and the means of producing the percept involves manipulation of ITD and/or ILD over headphones. Lateralization illustrates a fundamental example of virtual, as opposed to actual, sound source position. When identical monaural sounds are delivered from stereo headphones, the listener does not hear two distinct sounds coming from the transducers, and instead perceives a single virtual sound source which appears to be positioned at the center of the head. As ITD and ILD are increased, the perceived position of the virtual sound source will start to shift toward one ear, along an imaginary line. Once a critical value of the ITD or the ILD is reached, the perceived sound source will stop moving along the interaural axis and will be located at one of the ears. This effect is sometimes termed *inside-the-head localization*

⁸This is why surround systems use many small loudspeakers for high frequencies and one subwoofer for low frequencies.



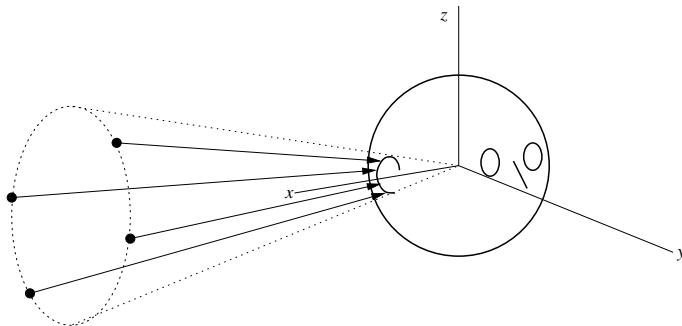


Figure 4.27: Cone of confusion.

(IHL). Having knowledge of this effect is important since headphone playback is otherwise superior to loudspeakers for transmitting virtual acoustic imagery in three dimensions.

Achieving *externalization* of the sound (i.e. in removing the IHL effect) is in many respects the “sacred graal” of headphone-based spatial audio systems. It is not completely clear what additional cues are most effective in producing sound externalization. However it has been observed by many that externalization increases as the stimulation approximates more closely a stimulation that is natural and that especially reverberation, either natural or artificial, can enhance dramatically externalization. In general, IHL is not an inevitable consequence of headphone listening, simply because externalized sounds can be heard through headphones in many instances.

4.5.2.3 Elevation perception

While the relevant cues for the localization of a sound source in the horizontal plane are relatively well understood, things become more complicated when we consider non-null elevations.

Figure 4.27 shows that sound sources located anywhere on a conical surface extending out from the ear of a spherical head produce identical values of ITD and ILD. These surfaces are often referred to as *cones of confusion*, and extend the concept of front/back confusion that we have examined above. Of course this situation is only theoretical: in reality ITD and ILD will never be completely identical on the cone of Fig. 4.27, because of the facial features and asymmetries already mentioned. Nonetheless, when ITD and ILD cues are maximally similar between two locations, a potential for confusion between the positions exists in the absence of other spatial cues.

The directional effects of the pinnae can disambiguate this confusion, and are considered to be particularly important for vertical localization. The role of the pinnae in improving vertical localization can be evaluated experimentally e.g. by comparing judgments made under normal conditions to a condition where the pinnae are bypassed or occluded. In fact vertical localization can be achieved even when one ear is completely occluded. This evidence supports the idea that the spectral cues provided by the pinnae work mainly monaurally.

There are many theories about the role of pinnae spectral cues. Very roughly, all of them suggest that a major cue for elevation involves movement of spectral notches and/or peaks, that change as a function of source and listener orientation. A way of appreciating the pinnae spectral cues is to examine the case of sound sources along the yz plane of the listener: note that this is the locus of the points where not only IID and ITD are null, but also spectral differences between the left and right HRTFs are null as long as the left and right pinnae are identical. If we look back at Fig. 4.25(b), we can notice a moving spectral notch that is thought to be important for elevation perception.

In general it is difficult without extensive psychoacoustic evaluation to ascertain how importantly

these changes function as spatial cues. In particular, it is unclear if localization cues are derived from a particular spectral feature such as a peak or a notch, or from the overall spectral shape. Also, it is generally considered that a sound source has to contain substantial energy in the high-frequency range for accurate judgment of elevation, because the pinna has limited dimensions in space and wavelengths longer than the size of the pinna are not affected (see also Fig. 4.22(a)). One could roughly state that the pinnae have a relatively little effect below 3 kHz.

While the role of the pinna in vertical localization has been extensively studied, the role of the torso is less well understood. We have seen in Sec. 4.5.1 that the torso disturbs incident sound waves at frequencies lower than those affected by the pinna. However, the effects of the torso are relatively weak, and experiments to establish the perceptual importance of low-frequency cues have produced mixed results.

4.5.2.4 Distance perception

It is an unanimous claim that auditory estimation of azimuth is more accurate than elevation estimation, and that distance estimation is the most difficult task. Accordingly, the cues for azimuth are quite well understood, those for elevation are less well understood, and those for distance are least well understood. Distance perception involves a process of integrating multiple cues, any of which can be rendered ineffective by the summed result of other potential cues.

In the absence of other information, *intensity* is the primary distance cue used by listeners, who learn from experience to correlate the physical displacement of sound sources with corresponding increases or reductions in intensity. Under anechoic conditions, sound intensity reduction with increasing distance is predicted by the inverse square law: an omnidirectional sound source's intensity will fall approximately 6 dB for each distance doubling (see also our discussion of the clarity index parameter in Sec. 4.2.2). However this law is not well motivated perceptually: it expresses the ratio of a sound source's intensity to a reference level, whereas the *perceived* magnitude of intensity is called *loudness*. Thus a mapping where the relative estimation of doubled distance follows "half-loudness" rather than "half-intensity" seems preferable: the two scales are different.⁹

Loudness (or intensity) increments can only operate effectively as distance cues in the absence of other information, in particular reverberation. When reverberation is present the overall loudness at a listener's ear does not change much for very close and very distant sources: the distance-dependent scaling applies only to the direct sound whereas the *reflected energy* remains approximately constant. The change in the proportion of reflected to direct energy, the so-called *R/D ratio*, seems to function as a stronger cue for distance than intensity scaling. In particular a sensation of changing distance can occur if the overall loudness remains constant but the R/D ratio is altered. Note however that in some cases the possible R/D ratio variation can be limited by the size of the particular environmental context, causing the cue to be less robust (e.g. in a small, acoustically treated room, the ratio would vary between smaller limits than in a large room like a gymnasium).

Estimation of distance with anechoic stimuli is usually worse than in experiments with "optimal" reverberation conditions. Many experimental results show an overall underestimation of the apparent distance of a sound source in an anechoic environment, which may be explained by the absence of reverberation. It can be said that reverberation provides the "spatiality" that allows listeners to move from the domain of loudness inferences to the domain of distance inferences, i.e. from an analytic listening attitude to an *everyday listening* attitude.

Distance perception is also affected by expectation or *familiarity* with the sound source. If the sound is completely synthetic (e.g., pulsed white noise), then a listener will typically focus on parametric

⁹We will return on the concept of loudness in Chapter *Auditory based processing*.



changes in loudness and R/D ratio (in this case loudness probably plays a more important role than reverberation effects). On the other hand, if the sound source is cognitively associated with a typical distance range, that range will be more easily perceived than unexpected or unfamiliar distances. This is especially true for speech: as an example, it is easier to simulate a whispering voice 20 cm away from your ear than it is to simulate an unnaturally loud whisper 10 m away.

Spectral effects can also affect distance perception, although to a lesser extent than the cues discussed above. Atmospheric conditions and air absorption play a role: with increasing distance, higher frequencies of a complex sound are increasingly attenuated by air humidity and temperature. There is little experimental evidence this cue is actually used by listeners in forming the distance of an auditory event, although some experimental results suggest that, in the absence of other cues, a low-frequency emphasis applied to a stimulus would be interpreted as “more distant” compared to an untreated stimulus. A second spectral effect is produced in the so-called *near field*, i.e. for distances less than approximately 1 m. Within this range it is not possible to assume the sound wavefronts to be planar, and the effect of their curvature must be taken into account. As the source approaches, emphasis is added to lower frequencies. This phenomenon corresponds to the “darkening” of tone color that occurs as a sound source is moved very close to one’s ear.

Note that all the cues discussed above are essentially monaural cues. An open question is whether binaural listening improves the perception of distance. This could indeed be the case again in the near-field limit. The spherical head model shows that in this limit both the ILD and the ITD at low frequencies are emphasized, especially for very lateralized sound sources ($\theta \sim \pm\pi/2$). This effect is sometimes termed *auditory parallax*, and has been interpreted by some to mean that the accuracy of estimation of a sound from the side should be improved when compared to distance perception on the median plane. There are numerous discrepancies in the literature, however, and the question of the influence binaural cues to distance perception is still unresolved.

4.5.2.5 Dynamic cues

So far we have examined sound source perception in the implicit assumption of static conditions, i.e. with both listener and source not moving. However in everyday perception we use also *dynamic* cues in addition to static ones to reinforce localization. These arise from active, sometimes unconscious, motions of listeners, who change their position relative to the source. When we hear a sound that we want to localize, we move e.g. in order to minimize the interaural differences, using our head as a sort of “pointer”. Animals use movable pinnae for the same purpose (think of a cat).

When moving their head, listeners apparently integrate some combination of the changes in ITD, ILD, and movement of spectral notches and peaks that occur with head movement over time, and subsequently use this information to improve localization ability. Perhaps the most clear example is represented by front/back confusions, which are common in static listening tests (see our discussion about cones of confusion), and instead disappear when listeners are allowed to turn their heads during a localization task: a listener who is trying to localize a source at, e.g. $\theta = 30^\circ$, $\phi = 0^\circ$ will probably attempt to center the auditory image by moving his head to the right. If the sound becomes increasingly centered –i.e. interaural differences are minimized– consequently to head motion, then it must be in the front. If instead it becomes increasingly lateralized –i.e., the sound becomes louder and arrives sooner at the right ear relative to the left– then it must be to the rear.

Dynamic cues are important also for externalization. IHL, which can be experienced with headphone reproduction as discussed previously, is less likely to occur when head movement is allowed, probably for the same reason that front/back confusion is avoided: dynamic cues arising from head motion are used to disambiguate locations, while static conditions can potentially lead to judgments at a “default” position inside or at the edge of the head. A very undesirable situation is when the sound scene is pre-



sented through headphones without tracking of head/body motion, *and* the listener can move: in this case dynamic cues are absent and the scene rotates together with the user, creating discomfort and preventing externalization. When visual cues are supplied however, e.g. one can move in a fully immersive virtual environment and can see the virtual sound source, it is quite likely that the combination of vestibular and visual cues will enable externalization. In fact externalization can occur even when listening to a television with a single earpiece: this is because vision is more reliable than audition in spatial location, and therefore our brain “trusts” visual rather than auditory feedback (the general mechanism underlying this phenomenon is known as “visual capture”).

Finally, active listener motion provides cues for distance perception. One is the motion-induced rate of change in intensity the so-called *acoustic τ* ,¹⁰ by which a listener who moves e.g. towards the sound can infer distance information. Another is the so-called *motion parallax*, which indicates the rate of change in angular direction resulting from listener translation: for a very close source, a small shift of the head causes a large change in angular direction, while for a very distant source the change is almost null irrespective of the amount of shift. The rate of change of ITD, ILD, and spectral notches/peaks will therefore be affected by the distance. This dynamic cue is in many respects similar to its visual counterpart (a large, distant sphere and a small, near sphere look the same, but if we move the different changes in perspective reveal the different distances).

4.6 Algorithms for 3-D sound rendering

Before examining processing algorithms for 3-D sound rendering we have to understand that the techniques to be developed depend on the type of system that is going to be used: the type of the effectors (e.g. loudspeakers vs. headphones), as well as their number and geometric arrangement (e.g. stereo systems vs. 5.1 surround systems, etc.).

Stereo is the simplest system involving “spatial” sound. In order to place a sound to the left or to the right, its signal is sent to the corresponding loudspeaker. If the same signal is sent to both speakers, the speakers are wired “in phase”, and the listener is approximately equidistant from the speakers, then the listener will perceive a “phantom source” located midway between the two loudspeakers. By crossfading the signal from one speaker to the other, one can create the impression of the source moving continuously between the two loudspeaker positions. With this technique however the perceived source will never move outside the line segment between the two speakers.

Multichannel systems are the next step in complexity. The idea is to have a separate channel for every desired direction, possibly including above and below. Commercial home-theater systems are based on this idea. In typically reverberant environments, one can exploit the limitations of our perception (see in Sec. 4.5.2 our discussion about azimuth perception in reverberant environments) and use small loudspeakers everywhere, except for one large speaker (the “subwoofer”) that provides the nondirectional, low-frequency content.

Headphone-based systems have some disadvantages compared to loudspeakers: headphones are invasive and can be uncomfortable to wear for long periods of time; they have non-flat frequency responses that can compromise spatialization effects; they tend to provide the impression of too close sources, and do not compensate for listener motion unless a tracking system is used. On the other hand they have two main advantages: first, they eliminate reverberation of the listening space; second, and more important, they allow to deliver distinct signals to each ear, which greatly simplifies the design of 3-D sound rendering techniques. On the contrary loudspeaker based systems suffer from “cross-talk”, i.e. the sound emitted by one loudspeaker will be always heard by both ears. If one ignores the effects of the listening

¹⁰This name comes from studies in visual perception, where the *optical τ* specifies the time-to-contact estimated by a subject in relative motion with respect to a target.



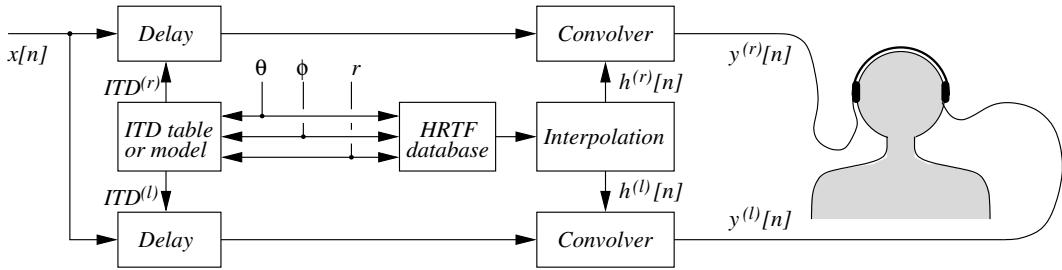


Figure 4.28: Block scheme of a headphone 3-D audio rendering system based on HRTFs.

environment, headphone listening conditions can be roughly approximated from stereo loudspeakers using *cross-talk cancellation* techniques, which try to pre-process the stereo signals in such a way that the sound emitted from one loudspeaker is cancelled at the opposite ear. Using these techniques the phantom source can be placed significantly outside of the line segment between the two loudspeakers and in particular elevation effects can be produced. The main problem is that the result will depend on where the listener is relative to the speakers: cross-talk cancellation is obtained only near the so-called “sweet spot”, a specific listener location assumed by the system.

In this section we will focus on techniques for headphone-based systems. We will implicitly assume that a single (point) sound source is rendered in space: if multiple sound sources have to be rendered, then each one has to be processed with a different replica of the rendering scheme, with consequent increases in the computational costs.

4.6.1 HRTF-based rendering

The general idea in HRTF-based 3-D audio systems is to use measured HRIRs and HRTFs. Given an anechoic signal and a desired virtual sound source position (θ, ϕ) , a left and right signals are synthesized by (1) delaying the anechoic signal by appropriate amount, in order to introduce the desired ITD, and (2) convolving the anechoic signal with the corresponding left and right head-related impulse responses. A synthetic block scheme is given in Fig. 4.28. In the remainder of this section we summarize the main steps involved in the development of a HRTF-based 3-D audio system, including HRTF measurement and processing, approximation through synthetic HRTFs, and interpolation.

4.6.1.1 Measuring HRTFs and ITDs

The typical setting for HRTF measurement is the following: an anechoic chamber, a set of speakers mounted on a geodesic sphere (with a radius of at least one meter in order to avoid near-field effects), at fixed intervals in azimuth and elevation. The listener is at the center of the sphere, with microphones placed in each ear. HRIRs are measured by playing an analytic signal and recording the responses produced at the ears, for each desired virtual position.¹¹ Listener and speakers do not need to be moved, facilitating the collection of measurements. Microphone placing is an issue: it can be placed at the entrance of a plugged ear canal, or near the eardrum to account for the response of the ear canal.

Measured HRTFs can be analyzed in order to estimate ITD values and derive a table to be subsequently used in the rendering stage (see the first processing block in Fig. 4.28). ITD estimation can be performed through various methods, including cross-correlation methods (where ITD is computed as the offset in cross-correlation maxima of $h^{(l)}$ and $h^{(r)}$), leading-edge methods (where the time difference

¹¹There is a plethora of sophisticated techniques for Impulse Response estimation, which we do not discuss here.

of the start of the impulses is estimated), and so on. Some approaches allow to derive a frequency-dependent ITD, while in other cases frequency-independent estimates are derived. Alternatively, theoretical ITD models can also be used instead of empirically estimated values. We have already examined a frequency-independent ITD model, given in Eq. (4.56). Other models exist, that introduce frequency dependence or even elevation dependence of ITD.

In most 3-D sound applications one typically wants to use a single set of HRTFs for every user. One approach might be to use the features of a person who has “desirable” HRTFs, based on some criteria. A set of HRTFs from a good localizer could be used if the criterion were localization performance. An alternative approach is to construct *generalized HRTFs*, that represent the common features of a number of individuals. Binaural impulse responses from many individuals can be “spectrally averaged” in the Fourier domain. However this can cause the resultant HRTF to have diminished spectral features with respect to individual ones. In the extreme case, one person has a 20 dB notch at 8 kHz, and another has a 20 dB peak – the average is no spectral feature at all.

Generalized HRTFs can also be obtained through the use of so-called “dummy heads”, which are mannequins constructed from averaged anthropometric measures and represent standardized heads with average pinnae and torso. The most widely used one is probably the *KEMAR* (Knowles Electronics Manikin for Auditory Research), although many others are commercially available. Measurements with dummy heads are easier, since they are often part of integrated measurement and analysis systems. The low frequency response of the microphones built into the head will be better than that of probe mics, and the results will be more replicable. Moreover, 3-D sound systems based on dummy head HRTFs will be closely matched to recordings made by the same binaural head, allowing compatibility between the two different types of processing. One dummy head might sound more natural to a particular set of users than another, depending on the microphones, the technique used for simulating the ear canal, the head’s dimensions, and so on. The head size (and correspondingly, its diffraction effects and overall ITD) is a major component in the suitability of one dummy head versus another.

4.6.1.2 Post-processing of measured HRTFs

Measured HRTFs undergo a series of processing steps. A typical procedure is post-equalization of HRTFs to eliminate potential spectral nonlinearities originated from the loudspeaker, the measuring microphone, and the headphones used for playback. As an example, probe microphones are usually small and are especially inefficient at low (< 400 Hz) frequencies, making high-pass filtering or “bass boosting” a fairly common HRTF post-equalization procedure. A frequency curve approximating the ear canal filter, usually derived from some standard equalization, can be applied if it was not part of the impulse response measurement. Since this filter is independent on the angle of incidence, it needs to be compensated only once. For most applications, the listener’s own ear canal resonance will be present during headphone listening; this requires removal of the ear canal resonance that may have been present in the original measurement, to avoid a “double resonance”.

One more post-processing procedure is often applied to reduce redundancy in HRTF data. Spectral features that are common to raw HRTFs at all locations do not contain important directional cues, and do not need to be encoded in each single HRTF. Therefore a so-called *Common Transfer Function (CTF)* is often estimated, by computing the mean log-magnitude of the HRTFs measured at several spatial locations. The CTF will include the direction-independent spectral features shared by all HRTFs (e.g., the ear canal filter). It will also include systematic measurement artifacts, if any. During postprocessing, the CTF can be removed from the raw HRTFs to yield the *Directional Transfer Function (DTF)*. The DTF is a function of θ , ϕ , and is the quantity that contains spectral cues responsible for spatial hearing. Let $C(\omega)$ be the known CTF and $D^{(l),(r)}(\theta, \phi, \omega)$ be the unknown left and right ear DTFs respectively.



Then $D^{(l),(r)}$ are estimated from $H^{(l),(r)}$ and C with the equality

$$H^{(l),(r)}(\theta, \phi, \omega) = C(\omega)D^{(l),(r)}(\theta, \phi, \omega). \quad (4.59)$$

The CTF captures the overall structure and dynamic range of the HRTFs, allowing each DTF to operate over a smaller dynamic range. This division allows us to vary a smaller parameter set (corresponding to only the DTF) to achieve space-varying HRTF approximations. Many of the algorithms described in the next sections can be applied either to the “raw” HRTFs or to the DTFs.

M-4.67

Write a script that computes the Common Transfer Function $C(\omega)$ and the Directional Transfer Functions $D(\theta_k, \phi_k, \omega)$ given a set of HRTFs $H(\theta_k, \phi_k, \omega)$ measured on M directions θ_k, ϕ_k ($k = 1 \dots M$).

A third post-processing procedure is minimum-phase reconstruction of the HRTF filters. Recall that a minimum-phase reconstruction of a filter is a filter with the same magnitude response of the original one, in which all zeros and poles are inside the unit circle. Minimum-phase filters have many benefits in terms of realization, coefficient interpolation, and so on. Various studies show that this processing step does not have any perceptual consequences, provided that ITD is introduced before convolution with the minimum-phased reconstructed HRTF (as in Fig. 4.28), as detailed phase information is not perceptually relevant.

Having acquired HRTF magnitude responses, one can design *synthetic HRTFs*, low-order filters that approximate the original HRTFs in a perceptually motivated way while providing significant computational advantages. In fact direct use of measured HRTFs requires a convolution with long FIR filters: assuming a duration of ~ 10 ms for a measured HRIR (reported durations vary across studies), the corresponding HRIR filter length is ~ 440 samples for $F_s = 44.1$ kHz. Despite the ever increasing computational power at our disposal, such filter sizes can make it difficult to synthesize complex acoustic environments in real time, particularly when multiple sound sources and reverberant environments have to be rendered.

Developing perceptually appropriate low-order representations of the HRTFs may also provide insight into sound localization mechanisms and into the usefulness of various cues embodied in the HRTF, which is incompletely understood. Moreover, such representations can be used to improve our understanding of the physical mechanisms that produce certain features in the HRTF.

We can schematically synthetic HRTF design techniques into two families. In *pole-zero models* the problem is viewed as one of filter design, which has several classical solutions. One drawback is that the coefficients are usually complicated functions of azimuth and elevation, and have to be tabulated, which hinders the usefulness of the model. *Series expansions* let one represent the HRTF as a weighted sum of simpler basis functions. While this is useful for inspecting the data, the run-time complexity of such models can limit their usefulness. In both cases, the original HRTFs can be further processed prior to the design of the synthetic HRTFs. Usually some form of *auditory smoothing* is used, that performs a non-uniform frequency-dependent smoothing of the responses based on psychoacoustic models. This produces more regular magnitude responses without any relevant perceptual consequences, and filter approximation is easier. We postpone the description of auditory smoothing techniques to Chapter *Auditory based processing*. In the next sections we discuss both pole-zero and series expansion approaches to synthetic HRTFs.



4.6.1.3 Synthetic HRTFs: pole-zero models

Given a direction (θ, ϕ) , a *pole-zero model* (or an ARMA model)¹² approximates the corresponding HRTF, $H(z)$, with the rational transfer function

$$\tilde{H}(z) = \frac{b_0 + \sum_{k=1}^q b_k z^{-k}}{1 - \sum_{k=1}^p a_k z^{-k}} = \frac{B(z)}{A(z)}. \quad (4.60)$$

For brevity, here and in the following we omit in the notation any dependence on (θ, ϕ) : in particular the coefficient vectors $\mathbf{b} = \{b_k\}$, $\mathbf{a} = \{a_k\}$ will depend on θ, ϕ .

In the particular case $p = 0$, Eq. (4.60) is an *all-zero* (FIR) model: in this case the most straightforward approximation consists in windowing the impulse response h to a shorter length. This approach can be since further refined to account for frequency-domain weighting that models the non-uniform frequency resolution of the ear.¹³ Various studies report of synthetic all-zero HRTF models obtained with this approach, with filter orders between 20 and 64.

In the particular case $q = 0$, Eq. (4.60) is an *all-pole* model: we have already seen in Chapter *Sound modeling: signal based approaches* that linear prediction can be used in this case to estimate the coefficients $\{a_k\}$ that allow \tilde{H} to best approximate H .

In the general case $q, p \geq 1$, traditional digital filter design techniques still state the problem as one of minimizing the difference between \tilde{H} and the target response H , which is typically known on a set of L “design frequencies” $\{\omega_k\}_{k=1}^L$ (e.g. $\omega_k = 2k\pi/LF_s$ if the ω_k are evenly distributed along the frequency axis). Usually this difference is expressed as a weighted error function \mathcal{E} given by

$$\mathcal{E}(\omega_d) = W(e^{j\omega_d}) [H(e^{j\omega_d}) - \tilde{H}(e^{j\omega_d})], \quad (4.61)$$

where W is some positive weighing function specified in the design. Moreover the error is usually estimated with regard to the magnitude response while the phase response is disregarded since, as already mentioned, the effect of the ITD is rendered separately and minimum phase transfer functions are used (see Fig. 4.28). Commonly used error functions are based on the L^p norm of the function \mathcal{E} . The most straightforward choice is the L^2 norm, which is generally known as the *Least-Squares Error* and corresponds to the energy of the difference signal:

$$E_{LS}\{\mathcal{E}\} = \frac{1}{2\pi} \int_{-\pi}^{\pi} |\mathcal{E}(\omega_d)|^2 d\omega_d \sim \frac{1}{L} \sum_{k=1}^L [\mathcal{E}(\omega_k)]^2. \quad (4.62)$$

Minimizing the error $E_{LS}\{\mathcal{E}\}$ means finding the coefficient vectors \mathbf{b}, \mathbf{a} for which the gradient of $E_{LS}\{\mathcal{E}\}$ is null, that is solving the set of equations

$$\nabla_a E_{LS}\{\mathcal{E}\} = \nabla_b E_{LS}\{\mathcal{E}\} = \mathbf{0}, \quad (4.63)$$

where the notation $\nabla_x E_{LS}$ stands for the gradient of E_{LS} with respect to the vector x . One of the main advantages of the least squares formulation is that the error function has a global minimum, since it is quadratic. We do not enter into the mathematics involved in writing and solving these equations and refer the reader to the literature on linear Least-Squares Error estimation.

M-4.68

Write a function that computes a LS pole-zero approximation of a target impulse response (representing e.g. a HRIR), given the orders p and q .

¹²See linear prediction in Chapter *Sound modeling: signal based approaches*.

¹³See Chapter *Auditory based processing*



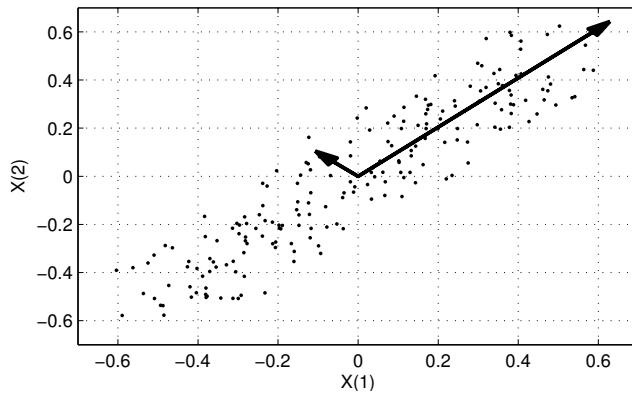


Figure 4.29: Example of principal component analysis: a two-dimensional data set with 0 mean, and the two basis vectors (principal axes) extracted using PCA.

Another choice is to minimize the L^∞ norm of the difference function:

$$E_\infty\{\mathcal{E}\} = \max_{-\pi < \omega_d < +\pi} |\mathcal{E}(\omega_d)|. \quad (4.64)$$

This is often referred to as Chebyshev or minimax criterion. Since this norm tries to minimize the maximum error, it should be able to provide good approximation of peaks and valleys of the HRTFs, which are relevant for localization as we know. On the other hand one drawback is that this error surface may not always be convex and thus may lead to unstable or locally optimal results.

As already mentioned for the all-zero case, for our particular filter design problem it is desirable to account for frequency-domain weighing that models the non-uniform frequency resolution of the ear. In this sense, error metrics that utilize absolute LS error on a linear scale are not the best choice, whereas an error criteria based on the difference in log magnitude might be perceptually more appropriate. Since both spectral peaks and spectral notches provide relevant information about the sound source location, minimizing the error on a log scale ensures that the solution is not biased toward peaks relative to notches. An example of such a perceptually motivated error criterion is

$$E_{\log}\{H, \tilde{H}\} = \frac{1}{L} \sum_{k=1}^L \left(\ln |H(\omega_k)| - \ln |\tilde{H}(\omega_k)| \right)^2, \quad (4.65)$$

A drawback of this kind of error functions is their minimization is a nonlinear problem, whose solution can be found only with iterative numerical solvers. Another way to construct a perceptually motivated error criterion is to choose the weighing function W in order to model auditory frequency resolution.

4.6.1.4 Synthetic HRTFs: series expansions

Based on the notions given in Sec. 4.5.1, one can argue on a physical basis that HRTFs should be completely determined by a relatively small number of physical parameters: average head radius, maximum pinna diameter, etc. This suggests that the intrinsic dimensionality of the HRTFs might be small, and that their complexity primarily reflects the fact that we are not viewing them correctly.

Among the statistical procedures used to provide a “simpler” representation of a set of correlated measures, a powerful and popular one is *principal component analysis (PCA)*, also known as Karhunen-Loëve transformation. The central idea of PCA is to reduce the dimensionality of a large dataset while retaining as much as possible of the variation present in the data. A small set of *basis vectors* is derived,



and these are used to compute the *principal components*, i.e. the sets of weights that reflect the relative contributions of each basis vector to the original data.

To start, assume we wish to represent M N -dimensional column vectors $\mathbf{x}_1 \dots \mathbf{x}_M$ with a 1-dimensional projection (a line) through their mean. The vector will then be represented as

$$\mathbf{x}_k \sim \mathbf{m} + a_k \mathbf{e} \quad k = 1, \dots, M, \quad (4.66)$$

where \mathbf{e} is a unit vector in the direction of the line, and a_k is a constant coefficient that estimates the distance of \mathbf{x}_k from the sample mean $\mathbf{m} = 1/M \sum_{k=1}^M \mathbf{x}_k$. The optimal coefficient a_k can be obtained by minimizing the “squared error criterion function”

$$E(a_1 \dots, a_k, \mathbf{e}) = \sum_{k=1}^M \|(\mathbf{m} - a_k \mathbf{e}) - \mathbf{x}_k\|^2. \quad (4.67)$$

For a given direction \mathbf{e} , the optimal coefficients are clearly $a_k = \mathbf{e}^T (\mathbf{x}_k - \mathbf{m})$, i.e. they are obtained by projecting the data vectors onto the line \mathbf{e} that passes through the sample mean. The question is now: what is the optimal direction \mathbf{e} ? By exploiting the expression written above for the optimal a_k 's, the error E can be rewritten after some straightforward algebra as

$$E(a_1 \dots, a_k, \mathbf{e}) = \sum_{k=1}^M \|(\mathbf{m} - a_k \mathbf{e}) - \mathbf{x}_k\|^2 = \dots = -\mathbf{e}^T \mathbf{S} \mathbf{e} + \sum_{k=1}^M \|\mathbf{x}_k - \mathbf{m}\|^2, \quad (4.68)$$

where $\mathbf{S} = \sum_{k=1}^M (\mathbf{x}_k - \mathbf{m})(\mathbf{x}_k - \mathbf{m})^T$ is the $N \times N$ *scattering matrix* of the data (which coincides with the covariance matrix except for a multiplying factor $1/(N-1)$). Therefore minimizing E means maximizing the function $f(\mathbf{e}) = \mathbf{e}^T \mathbf{S} \mathbf{e}$, with the constraint $\|\mathbf{e}\| = 1$. This can be done using Lagrange multipliers.¹⁴ For our PCA problem we have $L(\mathbf{e}, \lambda) = \mathbf{e}^T \mathbf{S} \mathbf{e} - \lambda(1 - \mathbf{e}^T \mathbf{e})$, and $\nabla_{\mathbf{e}} L(\mathbf{e}, \lambda) = 2\mathbf{S} \mathbf{e} - 2\lambda \mathbf{e}$. In conclusion the points \mathbf{e} that maximize $f(\mathbf{e})$ are those for which

$$\mathbf{S} \mathbf{e} = \lambda \mathbf{e}, \quad (4.69)$$

i.e. are the eigenvectors of \mathbf{S} for the eigenvalue λ . The single “best” line that represents the data is found by picking the eigenvector corresponding to the largest eigenvalue of \mathbf{S} so to ensure that $\mathbf{e}^T \mathbf{S} \mathbf{e} = \lambda$ is maximized. This can be readily extended to larger dimensions. If we wish to represent the \mathbf{x}_k 's on a q -dimensional hyperplane through the sample mean, written as

$$\mathbf{x}_k \sim \mathbf{m} + \sum_{i=1}^q a_{k,i} \mathbf{e}_i, \quad (4.70)$$

then we project the data onto the q eigenvectors of \mathbf{S} corresponding to the q largest eigenvalues. If we choose to use all eigenvectors, that is $q = M$ in Eq. (4.70), we will get the original data back (with no dimensionality reduction). From a geometrical standpoint, eigenvectors of \mathbf{S} represent the *principal axes* along which the data exhibit largest variance. The weight coefficients $a_{k,i}$ are called the *principal components*. Moreover the basis vectors are derived in such a way that the first one captures the majority of common variation present in the data and that the remaining vectors reflect decreasing common variation and increasing unique variation. The number q of principal axes required to provide an adequate representation of the data is largely a function of the amount of redundancy or correlation present in the data. The greater the redundancy, the smaller the number q needed.

¹⁴Recall that in order to find the extremum of a function $f(\mathbf{x})$ subject to a constraint $g(\mathbf{x}) = 0$, one can construct the Lagrange function $L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \lambda g(\mathbf{x})$ and look for a zero of the gradient $\nabla_{\mathbf{x}} L(\mathbf{x}, \lambda)$.



Now suppose we have measured directional transfer functions $D(\theta_k, \phi_k, \omega_j)$, on M directions θ_k, ϕ_k ($k = 1 \dots M$) and on N frequency points ω_j ($j = 1 \dots N$). We can apply PCA to the particular set of $M N$ -dimensional vectors x_k constructed as $x_{k,j} = \log |D(\theta_k, \phi_k, \omega_j)|$, i.e. we work on the log magnitudes of the DTFs (as already remarked, approximation of log-magnitudes is perceptually more appropriate than approximation of linear magnitudes). The result is a set of q basis vectors e_i (where $e_{i,j} = e_i(\omega_j)$), such that for the k th direction (θ_k, ϕ_k) the DTF can be approximated as

$$\log |D(\theta_k, \phi_k, \omega_j)| \sim \sum_{i=1}^q a_i(\theta_k, \phi_k) e_i(\omega_j). \quad (4.71)$$

Studies on the evaluation of this procedure have shown that the first five basis functions ($q = 5$) can accurately represent the magnitudes of the DTF set, and listening tests have shown a high correlation between responses to the synthesized and measured conditions. Moreover the dependence on space and frequency have been decoupled in Eq. (4.71), with consequent computational advantages.

M-4.69

Write a function that computes the first q principal axes e_i ($i = 1 \dots q$) and components $a_{i,k}$ for a set of DTFs $D(\theta_k, \phi_k, \omega_j)$ measured on M directions θ_k, ϕ_k ($k = 1 \dots M$) and on N frequency points ω_j ($j = 1 \dots N$).

4.6.1.5 Interpolation

HRTF measurements can only be made at a finite set of locations, and when a sound source at an intermediate location must be rendered, the HRTF must be *interpolated*. If interpolation is not applied (e.g.. if a nearest neighbor approach is used) audible artifacts like clicks and noise are generated in the sound spectrum when the source position changes.

A straightforward way to perform interpolation directly on the HRIR samples is the bilinear method, which simply consists of computing the response at a given point (θ, ϕ) as a weighted mean of the measured responses associated with the four nearest points. More precisely, if the corresponding set of HRIRs has been measured over a spherical grid with steps θ_{grid} and ϕ_{grid} , the estimate \hat{h} of the HRIR at an arbitrary point (θ, ϕ) can be obtained as (see Fig. 4.30(a))

$$\hat{h}[n] = (1 - c_\theta)(1 - c_\phi)h_1[n] + c_\theta(1 - c_\phi)h_2[n] + c_\theta c_\phi h_3[n] + (1 - c_\theta)c_\phi h_4[n], \quad (4.72)$$

where $h_k[n]$ ($k = 1, \dots, 4$) are the HRIRs associated with the four nearest points to the desired position. The parameters c_θ and c_ϕ are computed as

$$c_\theta = \frac{\theta \mod \theta_{\text{grid}}}{\theta_{\text{grid}}}, \quad c_\phi = \frac{\phi \mod \phi_{\text{grid}}}{\phi_{\text{grid}}}. \quad (4.73)$$

Several refinements can be applied to this simple technique, in order to improve efficiency. In particular, reduced-order HRIR such as those described earlier in this section can be used. Also, interpolation can be performed using only three grid points (those which form a triangle around the desired position). However, since some HRTF features arise due to coherent addition or cancelation of reflected and diffracted waves, interpolation may not preserve these features and produce perceptually poor results. Moreover, the interpolating filters are required to be minimum-phase: if this requirement is not satisfied, severe comb-filtering effects in the frequency domain can be produced when the phase delays of the interpolating filters vary considerably. Also, to capture fine details of the HRTF the sampling must be fine enough, i.e. satisfy a spatial Nyquist criterion. Interpolation can be performed in the frequency domain as well (i.e. estimate the DFT of \hat{h} by interpolating the DFTs of the h_k 's). Besides linear approaches, geometric and spline interpolation can be used as well.



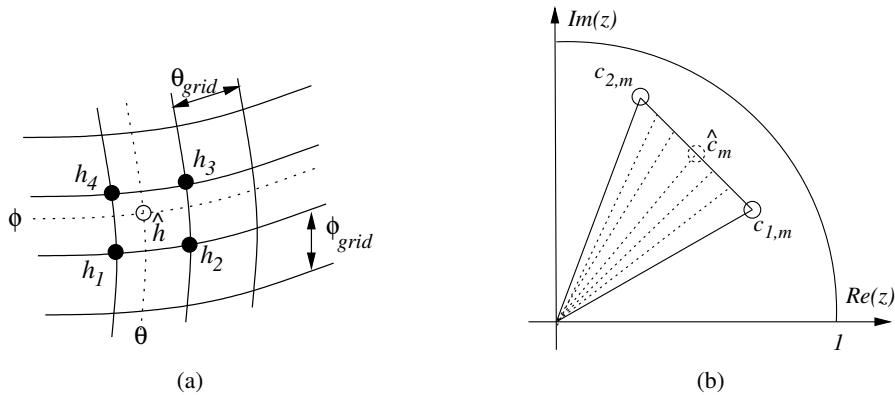


Figure 4.30: Approaches to HRTF interpolation; (a) bilinear interpolation of the HRIRs, and (b) interpolation of zeros for pole-zero synthetic HRTFs

M-4.70

Realize bilinear interpolation.

If synthetic HRTFs in the form of pole-zero filters are being used, interpolation can be performed on the poles and the zeros themselves. The case of an all-zero filter is relatively straightforward. Suppose that we want to interpolate between two transfer functions $H_k(z)$ ($k = 1, 2$) of the form

$$H_k(z) = 1 + \sum_{m=1}^q b_{k,m} z^{-m} = \prod_{k=0}^q (1 - c_{k,m} z^{-1}), \quad k = 1, 2, \quad (4.74)$$

where $b_{k,0} = 1$ without loss of generality, and where we are assuming that the zeros of both filters are sorted according to their phases. Then an interpolated filter $\hat{H}(z) = \prod_{m=0}^q (1 - \hat{c}_m z^{-1})$ can be obtained by (1) pairing the zeros according to angular proximity, and (2) computing the interpolated zeros $\hat{c}_m = (1 - \rho)c_{1,m} + \rho c_{2,m}$ ($m = 1, \dots, q$). Note that if the H_k are minimum-phase the interpolated filter is also minimum-phase (see also Fig. 4.30(b))

If we use pole-zero synthetic HRTFs of the form (4.60) with $p > 0$, then interpolation becomes more complicated. One can still use convex combinations of pole and zero values from neighbouring DTF approximations (note in particular that linear combination of stable poles is guaranteed to be stable). However a naive realization of this approach can result in erratic and occasionally large errors of the interpolated filters. In order to achieve regularity in the interpolation, more refined algorithms are needed that provide pairing and ordering on the entire HRTF database.

Synthetic HRTFs based on PCA expansions, in the form (4.71) are well suited for interpolation, since the dependence on frequency is decoupled from the dependence on spatial variables: therefore only the spatially-dependent coefficients $a_i(\theta, \phi)$ need to be interpolated while the frequency-dependent basis-vectors are not involved in the interpolation process. Functional representations of the a_i 's can be obtained through standard techniques such as spline interpolation.

In general, reconstruction of the underlying continuous coefficient functions from the samples obtained is an inherently ill-posed problem, because the samples do not uniquely define the functions in the absence of additional assumptions, and because the samples can be corrupted by the presence of noise. Some form of *smoothness constraints* must be used, so that a small change in θ, ϕ induces a small change in the coefficients.

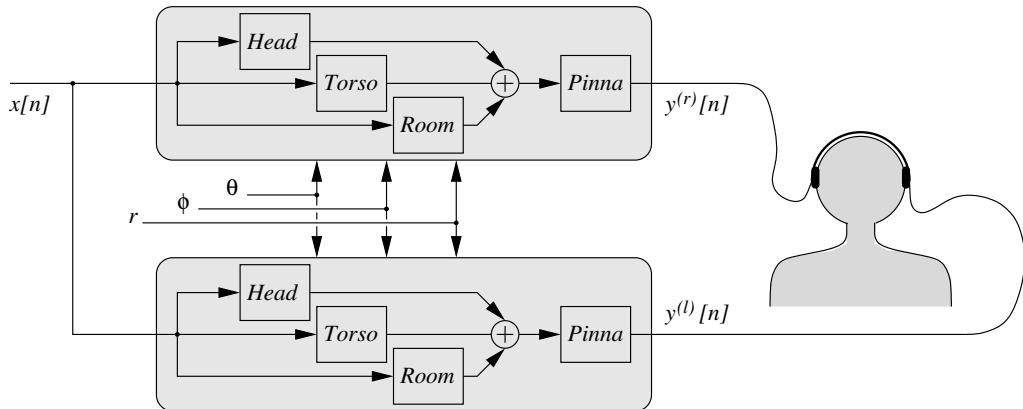


Figure 4.31: Block scheme of a headphone 3-D audio rendering system based on a structural model.

4.6.2 Structural models

As opposed to the HRTF-based rendering approach discussed above, the structural approach presented in this section is based on the modeling of the separate effects of the torso, head, and pinna, which combine to form the head related transfer function.

The HRTF is then modeled as a combination of filter blocks, each accounting for the contribution of one anatomical structure. The parameters of each block can in principle be related to anthropometric measures (e.g. the interaural distance, or the diameter of the cavum conchae), with the advantage that a generic structural HRTF model can be adapted to a specific listener and can account for posture-related effects. Another advantage is that room effects can be incorporated into the rendering scheme, specifically early reflections can be processed through the pinna model.

It is clear that separating the effects of various anatomical structures into perfectly independent filter structures is a heuristic approximation that disregards interactions due to waves scattered from one structure to another. However research in this direction has shown that structural models are able to provide good approximations of real HRTFs.

A synthetic block scheme of a generic structural model is given in Fig. 4.31. In the remainder of this section we describe modeling approaches for each of the three main components depicted in the figure, namely head, torso, and pinna. Room effects can be also accounted for in this structure: in particular early reflections can be convolved with a pinna model, depending on their incoming direction (see also our discussion on directional effects with early reflection modeling in Sec. 4.3.2).

4.6.2.1 Head models

In Sec. 4.5.1 we have analyzed the effects of the head on the sound field at the eardrum by approximating the head with a sphere. We have seen that given a sphere of radius a , a point sound source at a distance $r > a$ from the center of the sphere, and a point on the sphere, then the diffraction of an acoustic wave by the sphere seen on the chosen point can be expressed with a transfer function $H_{\text{sphere}}(\rho, \theta_{\text{inc}}, \mu)$ (where we are using the normalized frequency $\mu = \omega a / c$ and the normalized distance $\rho = r/a$, and θ_{inc} is the angle of incidence). We have also studied this transfer function in the limit of $\rho \rightarrow +\infty$.

In this limit the response H_{sphere} can be approximated with a parametric filter $\tilde{H}_{\text{sphere}}(\theta_{\text{inc}}, \mu)$, whose parameters depend on θ_{inc} only. In fact already a first order filter can provide reasonable results, if



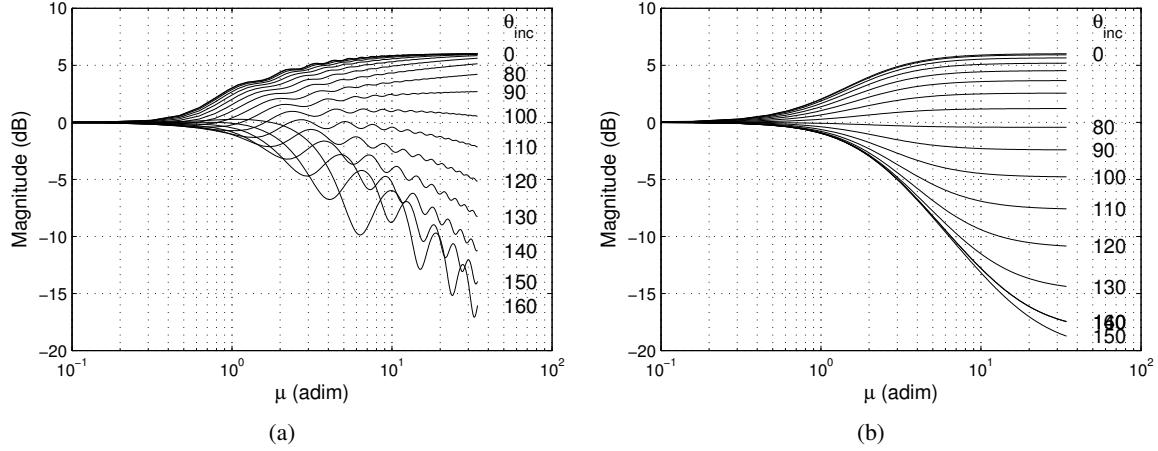


Figure 4.32: Spherical head model; (a) ideal response of Eq. (4.57) for $\rho \rightarrow +\infty$, (b) approximated response with the first-order filter of Eq. (4.75) with $\alpha_{min} = 0.1$ and $\theta_{min} = 170^\circ$.

properly parametrized. Some authors have proposed the following form:

$$\tilde{H}_{\text{sphere}}(\theta_{inc}, \mu) = \frac{1 + \frac{j}{2}\mu \cdot \alpha(\theta_{inc})}{1 + \frac{j}{2}\mu}, \quad 0 \leq \alpha(\theta_{inc}) \leq 2. \quad (4.75)$$

The idea behind this equation is that the θ_{inc} -dependent parameter α controls the location of the zero in the numerator: for $\alpha = 2$ the filter gives a 6 dB boost at high frequencies (which corresponds to the behavior of H_{sphere} for $\theta_{inc} = 0$), while for $\alpha < 1$ there is a low pass effect. Moreover, in order for $\tilde{H}_{\text{sphere}}$ to match the behavior of H_{sphere} at values $\theta_{inc} \neq 0$, the parameter α must depend in a nonlinear way on θ_{inc} . A possible choice is

$$\alpha(\theta_{inc}) = \left(1 + \frac{\alpha_{min}}{2}\right) + \left(1 - \frac{\alpha_{min}}{2}\right) \cos\left(\frac{\theta_{inc}}{\theta_{min}}\right) \quad (4.76)$$

where values of the auxiliary parameters α_{min} , θ_{min} can be chosen in order to tune the dependence of α on θ_{inc} . The result can be seen in Fig. 4.32.

The filter $\tilde{H}_{\text{sphere}}$ can already produce fairly convincing azimuth effects, even though it only matches the gross magnitude characteristics of the spectrum. In order to enhance its effectiveness, an all-pass section has to be cascaded to it, to account for the interaural time difference: we can implement this additional block as a fractional delay filter¹⁵ $F_{ITD}(\theta_{inc}, z)$, so that the complete head models is $\tilde{H}_{\text{sphere}}(\theta_{inc}, z) \cdot F_{ITD}(\theta_{inc}, z)$. The ITD values used to parametrize the filter F_{ITD} can be values derived from measured HRTFs, or values derived from theoretical ITD models. We have already discussed this point at the beginning of Sec. 4.6.1.

M-4.71

Write a function that realizes the first-order filter (4.75).

It is clear that a sphere provides only a first approximation to a human head. Better approximation can be already obtained by introducing two simple refinements. First, one can use a non-spherical shape:

¹⁵Recall that we have discussed fractional delay filters in Chapter *Sound modeling: source based approaches*.



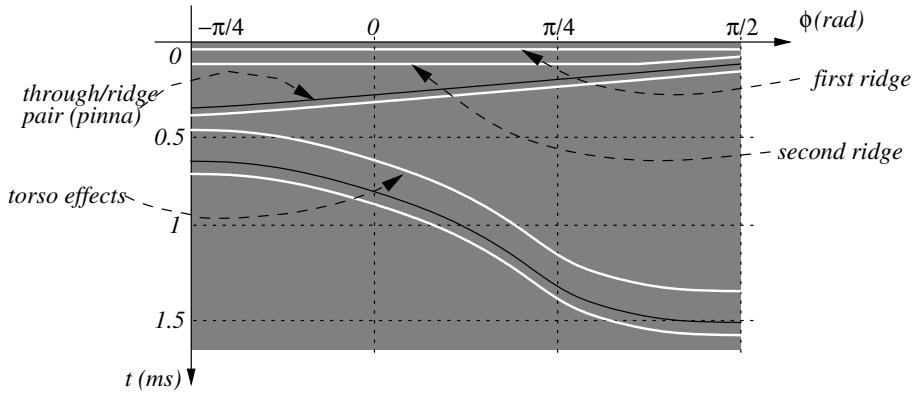


Figure 4.33: A schematic representation of the major features of the HRIR in the median plane ($\theta = 0$) for a human subject. White and black lines indicate ridges and throughs in the response, respectively.

an ellipsoid is an obvious choice. Second, one can note that the ears are not positioned across a diameter, but are displaced behind and below the center of the head. As already remarked in Sec. 4.5.2, these two anatomical details have the consequence that the ITD is a function of elevation as well as azimuth. In fact analysis on measured HRTFs shows that for a fixed value of θ and varying values of ϕ ,¹⁶ the ITD can vary by almost 20% of its maximum value, with noticeable perceptual effects.

4.6.2.2 Modeling torso and pinna reflections

Based on what we have said in the previous sections, we can assume that the main effects of torso and pinna that need to be accounted for are reflections. This means that both torso and pinna will be modeled as FIR comb filters, in which each reflection determines a comb series in the spectrum. We should be aware however that reflection is a short-wavelength or high-frequency concept, and modeling the effects of torso and pinna by specular reflections is only a first approximation.

In order to realize a model for the torso and the pinna, everything reduces down to estimating reflection delays and their dependence on θ and ϕ , either through analysis of measured HRIRs/HRTFs, or through numerical simulations. As remarked by many authors, a general trend can be observed in measured HRTFs. A schematic representation is given in Fig. 4.33, where only elevations in the range $[-\pi/4, \pi/2]$ have been considered: for values $\phi < -\pi/4$ head shadowing effects start to appear, while HRIR features (end especially pinna related features) are less clear for $\phi > \pi/2$.

The initial ridge due to the direct impulse is followed by a sequence of ridges and troughs. A second ridge occurs roughly 50μs after the initial ridge and varies only slightly with elevation. It is followed by a very prominent trough and ridge pair whose latency varies nearly monotonically with elevation from about 400μs at $\phi = -\pi/4$ to 100μs at $\phi = \pi/2$. The sharply positive sloping diagonal events are due to a torso reflection and its replication by pinna effects. The delay between the direct and the reflected sound from the torso is maximum above the head and decreases with elevation, as one would expect from geometrical considerations.

Note that the scheme depicted in Fig 4.33 corresponds to HRIRs measured in the median plane ($\theta = 0$). Torso echoes vary significantly with azimuth also. On the contrary, pinna events exhibit very limited azimuth dependence.

The main torso reflection is relatively straightforward to estimate, either from measured HRIRs, or from numerical simulations of simplified models where both head and torso are approximated as

¹⁶We are using the interaural polar coordinate system here.

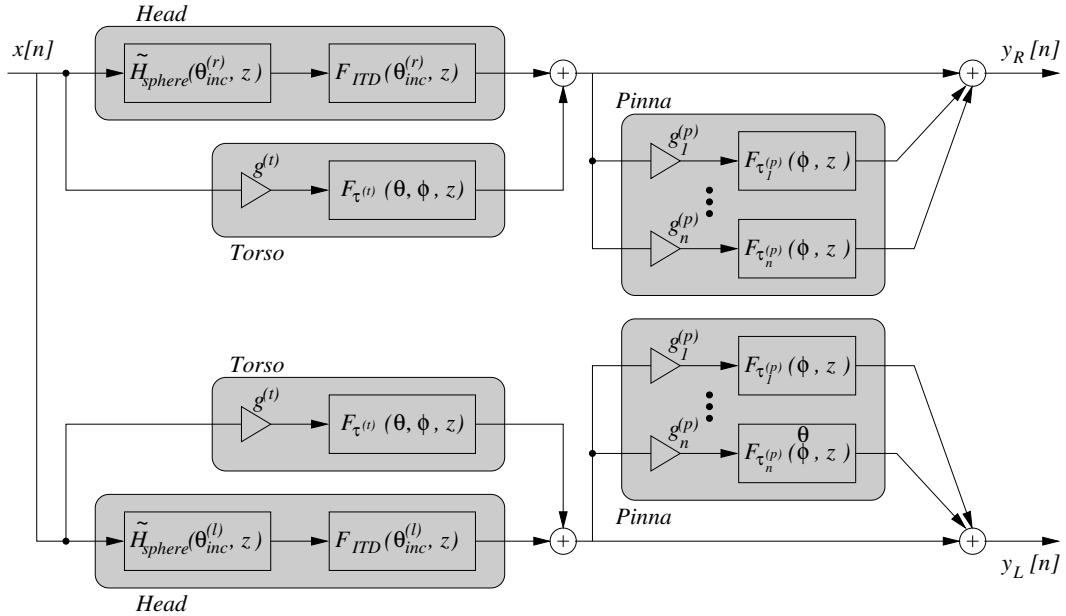


Figure 4.34: A simple yet complete structural model.

ellipsoids (so-called “snowman” models). With these methods one can estimate the θ - and ϕ -dependent delay $\tau^{(t)}(\theta, \phi)$ of the main torso reflection. In conclusion, torso effects can be modeled with a single fractional delay filter $g^{(t)}F_{\tau^{(t)}}(\theta, \phi, z)$, where $g^{(t)}$ is the torso reflection coefficient. Note that this model is only valid for positive ϕ values: as the source descends in elevation, a point of grazing incidence is reached, below which torso reflections disappear and torso shadowing emerges.

Pinna effects are harder to model, since it is more difficult to automatically extract filter parameters from measured data. Time-domain analysis (i.e., identification of reflections in the HRIR) is in this case not reliable. Frequency-domain analysis is preferable, and consists in identifying notch series in the HRTF. If such series can be identified, they can then be related to ear anatomy. More precisely, the delay $\tau_i^{(p)}(\phi)$ of the i th pinna reflection causes periodic notches in the spectrum,¹⁷ with frequencies $\omega_{i,n}^{(p)}(\phi) = 2\pi(2n + 1)/\tau_i^{(p)}(\phi)$ (with $n \geq 0$). Moreover, $\tau_i^{(p)}(\phi) = 2d_i^{(p)}(\phi)/c$, where $d(\phi)$ is the distance between the i th reflecting surface of the pinna (e.g. the cavum conchae) and the ear canal. The frequency $\omega_{i,0}^{(p)}(\phi)$ of the first notch and $d_i^{(p)}(\phi)$ are then related through the equation

$$\omega_{i,0}^{(p)}(\phi) = \pi c / 2d_i^{(p)}(\phi). \quad (4.77)$$

Therefore, given an estimate of the function $\omega_{i,0}^{(p)}(\phi)$ obtained from analysis of HRTFs, the function $d_i^{(p)}(\phi)$ can be estimated through this equation and consequently the measured notches can be related to anatomical details of the pinna. In conclusion, pinna effects can be modeled with a set of n fractional delay filters $g_i^{(p)}F_{\tau_i^{(p)}}(\phi, z)$.

4.6.2.3 A complete structural model

The components that we have analyzed in the previous sections can be combined to form the simple yet complete structural model depicted in Fig. 4.34, which explores the general block scheme presented

¹⁷See comb filters in *Sound modeling: source based approaches*.

in Fig. 4.31. The rationale for this structure is that sound can reach the ear pinna via two major paths: diffraction around the head, and reflection from the torso. In both cases, the sound waves that reach the pinna are altered by pinna reflections before entering the ear canal.

M-4.72

Realize the structural model of Fig. 4.34.

This model can be refined in many respects. The first-order head-shadow filter $\tilde{H}_{\text{sphere}}$ can be replaced by more accurate filters. In particular, $\tilde{H}_{\text{sphere}}$ is derived in the far-field limit: in order to model near-field effects we should substitute it with a filter that approximates Eq. (4.57) directly, and takes into account dependence on range.

Some parameters can be made direction-dependent: in particular, careful examination of torso echo patterns reveals that torso reflection coefficients $g^{(t)}$ vary with elevation. Finally, note that in this model sound diffracted from the head and sound reflected from the torso are processed through the same pinna models: this is not entirely correct since the torso echoes arrive at the ear from a different direction than the direct sound, and therefore they should really pass through a different pinna model. On the other hand the actual perceptual relevance of torso reflections is not clear, as already mentioned in Sec. 4.5.2, therefore this approximate description can be considered to be acceptable.

4.7 Commented bibliography

Wallace C. Sabine has in a way invented the science of concert hall acoustics in the early '900s. For a review of his work and early literature on concert hall acoustics see [Sabine, 1939] (note that the Paul E. Sabine author of this paper is the cousin of Wallace). A very complete discussion of physical aspects of room acoustics is provided by Kuttruff [1991]: Section 4.2.1 is almost entirely based on this book. We have not discussed techniques for impulse response (and particularly RIR) measurement, for a review see e.g. [Stan et al., 2002]. Farina and coworkers have worked extensively on RIR measurements but also on the simulation of the acoustics of closed spaces; the RIR plotted in Fig. 4.3 is one of the publicly available RIRs on the group webpage.¹⁸

Concerning the research on perceptual attributes of reverberation, the tutorial paper by Beranek [1992] summarizes the main results obtained up to 1992. Research at IRCAM tried to provide a minimal set of independent parameters that give an exhaustive characterization of room acoustic quality [Jot, 1999]. These parameters are divided into three categories, that relate to room perception, source/room interaction, and source perception, respectively.

The first artificial reverberators were electromechanical devices such as *plate reverberators* and *spring reverberators*, in which mechanical elements like plates and springs were fed with a dry sound signal, and an output signal was read at a different point of the element. Despite their limited ability in simulating real environments, plate and spring reverbs have become through the years some of the most sought after effects in digital audio [Bilbao and Parker, 2010].

The first artificial reverberator based on filters was proposed by Manfred Schroeder in the early '60's. The reverberator realized in our example M-4.58 is in fact the Schroeder [1962] reverberator. Schroeder also provided a method for measuring the reverberation time [Schroeder, 1965], which can be used to realize the code in example M-4.56. Moreover, Schroeder [1970] proposed the combination of early reflections and late reverberation depicted in our Fig. 4.12(a).

An extensive experimentation on structures for artificial reverberation was conducted by Andy Moorer in the late '70's. He extended the Schroeder's work in relating some basic computational structures (e.g.,

¹⁸See <http://pcfarina.eng.unipr.it/>.



tapped delay lines, comb and allpass filters) with the physical behavior of actual rooms. The reverberator realized in our example M-4.59 is in fact the Moorer [1979] reverberator. He also proposed the combination of early reflections and late reverberation depicted in our Fig. 4.12(b).

Gardner [1998] has explored the use of structures based on all-pass and nested all-pass filters (see in particular Figs. 4.10 and 4.11). This reference, together with [Rocchesso, 2002], also provides an general extensive overview of reverberation algorithms, including binaural reverberation. Research on binaural reverberation techniques includes work by [Begault, 1994, Chapter 4] and by Griesinger [1997]. Our Fig. 4.13(b) is based on this latter reference.

Feedback Delay Networks were first suggested for artificial reverberation by Gerzon [1971, 1972], who noted that several comb filters could “sound good” when cross-coupled. He proposed an orthogonal matrix feedback around a parallel bank of delay lines, as a means of maximizing cross-coupling. Some years later Stautner and Puckette [1982] independently suggested similar ideas and proposed a four-channel FDN reverberator based on the feedback matrix given in our Eq. (4.38). Jot [Jot and Chaigne, 1991, Jot, 1991, 1997] developed a systematic FDN design methodology allowing largely independent setting of reverberation time in different frequency bands. Rocchesso and Smith [1997] have provided further insights about the structures of feedback matrices in FDNs, and discussed analogies between FDNs and DWNs. General discussions of the use of FDNs for artificial reverberation are provided by Gardner [1998], Rocchesso [2002], Smith [2008]

Waveguide meshes were first studied by Van Duyne and Smith [1993, 1995]. Since then many studies have focused on techniques for reducing dispersion errors. Savioja and Välimäki [2000, 2003] have proposed interpolation and frequency-warping techniques to reduce dispersion as function of both frequency and propagation direction. Fontana and Rocchesso [1998, 2001] have focused on 2-D meshes, and provided results both about applications to membrane modeling and about general numerical aspects: they compared square, triangular, and hexagonal meshes in terms of sampling efficiency and dispersion error. Bilbao [2004] has also investigated in details many numerical and computational properties of the waveguide mesh, in particular he analyzed dispersion properties of various mesh topologies using von Neumann analysis and he provided a unified view of the digital waveguide mesh and wave digital filters as particular classes of energy invariant finite difference schemes. Finally, another topic addressed in the literature is the design of mesh boundaries, with a special focus on modeling diffusion. This problem was addressed by Laird et al. [1999], and later by Lee and Smith [2004], who used quadratic residue sequences to design maximally diffusing boundaries.

Three general and valuable books on spatial hearing are [Blauert, 1996], which is the traditional reference on the psychophysics of three-dimensional hearing, [Carlile, 1996], which not surveys the physics and psychophysics of 3-D auditory perception and also addresses the synthesis of spatial sound, and [Begault, 1994], which is focused on 3-D sound rendering techniques and applications to virtual reality and multimedia.

One of the pioneers in spatial hearing research was John Strutt, better known as Lord Rayleigh. He first described quantitatively the shadow effects of a sphere in [Strutt, 1904], and subsequently presented in [Strutt, 1907] the Duplex Theory that we have described in Sec. 4.5.2. The acoustic effects of the pinna have been studied in later years. Edgar A. G. Shaw and coworkers developed mechanical models of the external ear and measured their acoustic properties in several works (see e.g. [Teranishi and Shaw, 1968]). In the same years Dwight W. Batteau studied the role of the pinna in sound localization [Batteau, 1967]. More recently pinna effects have been studied through computational models, e.g. by Katz [2001].

As already mentioned, auditory cues for distance perception are still not completely understood. A recent review on the subject is provided by Zahorik et al. [2005]. The perceptual relevance of intensity scaling with distance han been known for a long time (see [Coleman, 1963]). Begault [1991] has shown that the preferred scaling of intensity with distance depends on the stimulus type. The R/D ratio have been



cited in many studies as a relevant cue to distance since Rabinovich [1936]. Other relevant studies about the role of reverberation, familiarity, and expectation in distance perception include those by Mershon and Bowers [1979] and by Gardner [1969]. Butler et al. [1980] have studied distance-dependent spectral effects due to air absorption. Sound source localization in the near-field is another open research topic. Recent studies include the work by Shinn-Cunningham et al. [2000] and by Brungart [2002].

Studies on the importance of dynamic cues for sound localization date back to Wallach [1940]. Since then many studies have shown that active motion helps especially in azimuth estimation and to a lesser extent in elevation estimation [Thurlow and Runge, 1967, Perrett and Noble, 1997]. Wightman and Kistler [1999] have provided evidence of the disappearing of front-back reversal when listeners are allowed to turn their heads during the localization task. Loomis et al. [1998] have studied the role of dynamic cues, specifically motion parallax and acoustic tau, on the perception of distance.

A tutorial of HRTF-based rendering techniques is [Cheng and Wakefield, 2001], while a review paper more focused on the evaluation of 3-D sound systems is [Martens, 2003]. Huopaniemi [1999] also provides an extensive overview, especially on synthetic HRTFs and pole-zero models. The first attempt to develop a pole-zero HRTF model is reportedly Asano et al. [1990]. Other relevant contributions include work by Wakefield and coworkers (see e.g. [Durant and Wakefield, 2002]), and by Kulkarni and Colburn [2004]. The Interface Lab. at UC Davis has created a public-domain database of high-spatial-resolution HRTF measurements for 45 different subjects, including the KEMAR mannequin with both small and large pinnae. The database is described in Algazi et al. [2001]. The HRTFs plotted in our Fig. 4.25 have been taken from this database.

The first attempt to apply PCA techniques to series expansions of HRTFs appears to be [Martens, 1987]. Other relevant contributions include in particular work by Kistler and Wightman [1992]. Middlebrooks and Green [1992] have studied the relation between basis vectors obtained from PCA and anthropometric data. PCA is the oldest technique in multivariate analysis. It was originally developed by Pearson [1901] and further generalized by other authors. A general introduction to PCA can be found e.g. in [Duda et al., 2000].

Concerning HRTF interpolation: direct bilinear interpolation on FIR coefficients is described by Huopaniemi [1999]. Other recent contribution include [Zotkin et al., 2004], where an interpolation method that uses only three grid points is proposed, and [Freeland et al., 2004], where an interpolation procedure similar to the bilinear method, but based on auxiliary “interpositional transfer functions” (IPTFs), is proposed. Interpolation of pole-zero HRTF models is addressed e.g. by Hacihabiboglu et al. [2005] and by Larcher [2001]. Interpolation of HRTF models based on PCA expansions has been investigated by Chen et al. [1995].

The origin of research on structural HRTF models is probably to be found in the work of Genuit [1984]. Even though it was based on very crude approximations of human geometries, the model incorporated static features of the HRTF (ear-canal resonance and eardrum impedance), as well as azimuth-dependent (ITD, IID) and elevation-dependent (pinna and torso reflections) features. The Interface Lab. at UC Davis has been working on the topic since the early ’90’s and has produced a number of relevant research papers. Much of our Sec. 4.6.2 is based on their work. For a start, see [Brown and Duda, 1998]. An interesting work that relates resonant properties of the pinna to anthropometry is [Raykar et al., 2005].

References

- V. Ralph Algazi, Richard O. Duda, Dennis M. Thompson, and Carlos Avendano. The CIPIC HRTF database. In *Proc. IEEE Workshop Appl. Signal Process. to Audio and Acoust. (WASPAA01)*, pages 99–102, Mohonk Mountain House, New Paltz, NY, Oct. 2001.
- Futoshi Asano, Yoiti Suzuki, and Toshio Sone. Role of spectral cues in median plane localization. *J. Acoust. Soc. Am.*, 88(1): 159–168, July 1990.



- Dwight W. Batteau. The role of the pinna in human localization. *Proc. R. Soc. London. Series B, Biological Sciences*, 168, (1011):158–180, Aug. 1967.
- Durand R. Begault. Preferred sound intensity increase for sensation of half distance. *Perceptual and Motor Skills*, 72:1019–1029, June 1991.
- Durand R. Begault. *3-D Sound for Virtual Reality and Multimedia*. Academic Press Inc., 1994.
- Leo L. Beranek. Concert hall acoustics. *J. Acoust. Soc. Am.*, 92(1):1–39, July 1992.
- Stefan Bilbao. *Wave and Scattering Methods for Numerical Simulation*. John Wiley and Sons, Inc., New York, 2004.
- Stefan Bilbao and Julian Parker. A virtual model of spring reverberation. *IEEE Trans. Speech Audio Process.*, 2010. In press.
- Jens Blauert. *Spatial Hearing: Psychophysics of Human Sound Localization*. MIT Press, Cambridge, Mass., 2nd edition, 1996.
- C. Phillip Brown and Richard O. Duda. A structural model for binaural sound synthesis. *IEEE Trans. Speech Audio Process.*, 6(5):476–488, Sep. 1998.
- Douglas S. Brungart. Near-field virtual audio displays. *Presence: Teleoperators and Virtual Environment*, 11(1):93–106, Feb. 2002.
- Robert A. Butler, Elena T. Levy, and William D. Neff. Apparent distance of sounds recorded in echoic and anechoic chambers. *J. Experimental Psychology*, 6(4):745–750, Nov. 1980.
- Simon Carlile. *Virtual Auditory Space: Generation and Applications*. Chapman and Hall, New York, 1996.
- Jiashu Chen, Barry D. Van Veen, and Kurt E. Hecox. A spatial feature extraction and regularization model for the head-related transfer function. *J. Acoust. Soc. Am.*, 97(1):439–452, Jan. 1995.
- Corey I. Cheng and Gregory H. Wakefield. Introduction to Head-Related Transfer Functions (HRTFs): Representations of HRTFs in time, frequency, and space. *J. Audio Eng. Soc.*, 49(4):231–249, Apr. 2001.
- Paul D. Coleman. An analysis of cues to auditory depth perception in free space. *Psychological Bulletin*, 60(3):302–315, May 1963.
- Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. John Wiley & Sons, Nov. 2000.
- Eric A. Durant and Gregory H. Wakefield. Efficient model fitting using a genetic algorithm: pole-zero approximations of HRTFs. *IEEE Trans. Speech Audio Process.*, 10(1):18–27, Jan. 2002.
- Federico Fontana and Davide Rocchesso. Physical modeling of membranes for percussion instruments. *Acta Acustica united with Acustica*, 84(3):529–542, May 1998.
- Federico Fontana and Davide Rocchesso. Signal-Theoretic Characterization of Waveguide Mesh Geometries for Models of Two-Dimensional Wave Propagation in Elastic Media. *IEEE Trans. Speech Audio Process.*, 9(2):152–161, Feb. 2001.
- Fabio P. Freeland, Luiz W. P. Biscainho, and Paulo S. R. Diniz. Interpositional transfer function for 3d-sound generation. *J. Audio Eng. Soc.*, 52(9):915–930, Sep. 2004.
- Mark B. Gardner. Distance estimation of 0° or apparent 0°-oriented speech signals in anechoic space. *J. Acoust. Soc. Am.*, 45 (1):47–53, Jan. 1969.
- William G. Gardner. Reverberation algorithms. In Mark Kahrs and Karl-Heinz Brandenburg, editors, *Applications of Digital Signal Processing to Audio and Acoustics*, pages 85–131. Kluwer Academic Publishers, New York, Mar. 1998.
- Klaus Genuit. *A model for the description of outer-ear transmission characteristics*. PhD thesis, Rheinisch-Westfälischen Technischen Hochschule Aachen, Aachen, Germany, Dec. 1984.
- Michael A. Gerzon. Synthetic stereo reverberation, Part I. *Studio Sound*, 13:632–635, Dec. 1971.
- Michael A. Gerzon. Synthetic stereo reverberation, Part II. *Studio Sound*, 14:24–28, Jan. 1972.



David Griesinger. The psychoacoustics of apparent source width, spaciousness and envelopment in performance spaces. *Acta Acustica united with Acustica*, 83(4):721–731, 1997.

Hüseyin Hacıhabiboglu, Banu Günel, and Ahmet M. Kondoz. Head-related transfer function interpolation by root displacement. In *Proc. IEEE Workshop Appl. Signal Process. to Audio and Acoust. (WASPAA05)*, pages 134–137, Mohonk Mountain House, New Paltz, NY, Oct. 2005.

Jyri Huopaniemi. *Virtual acoustics and 3-D sound in multimedia signal processing*. PhD thesis, Helsinki University of Technology, Faculty of Electrical and Communications Engineering, Laboratory of Acoustics and Audio Signal Processing, Espoo, 1999.

Jean-Marc Jot. An analysis/synthesis approach to real-time artificial reverberation. In *Proc. IEEE Int. Conf. Acoust. Speech and Signal Process.*, volume 2, pages 221–224, S. Francisco, Feb. 1991.

Jean-Marc Jot. Efficient models for reverberation and distance rendering in computer music and virtual audio reality. In *Proc. Int. Computer Music Conf.*, pages 236–243, Thessaloniki, 1997.

Jean-Marc Jot. Real-time spatial processing of sounds for music, multimedia, and interactive human-computer interfaces. *Multimedia Systems*, 7(1):55–69, Jan. 1999.

Jean-Marc Jot and Antoine Chaigne. Digital delay networks for designing artificial reverberators. In *Proc. Audio Engineering Society Convention*, Paris, Feb. 1991. Preprint 3030.

Brian F. G. Katz. Boundary element method calculation of individual head-related transfer function. I. Rigid model calculation. *J. Acoust. Soc. Am.*, 110(5):2440–2448, Nov. 2001.

Doris J. Kistler and Frederic L. Wightman. A model of head-related transfer functions based on principal components analysis and minimum-phase reconstruction. *J. Acoust. Soc. Am.*, 91(3):1637–1647, Mar. 1992.

Abhijit Kulkarni and H. Steven Colburn. Infinite-impulse-response models of the head-related transfer function. *J. Acoust. Soc. Am.*, 115(4):1714–1728, Apr. 2004.

Heinrich Kuttruff. *Room Acoustics*. Elsevier Applied Science, London and New York, 3rd edition, 1991.

Joel Laird, Paul Masri, and Nishan Canagarajah. Modelling diffusion at the boundary of a digital waveguide mesh. In *Proc. Int. Computer Music Conf.*, pages 492–495, Beijing, Oct. 1999.

Véronique Larcher. *Techniques de spatialisation des sons pour la réalité virtuelle*. PhD thesis, Université de Paris VI, Paris, May 2001.

Kyogu Lee and Julius O. Smith. Implementation of a highly diffusing 2-D digital waveguide mesh with a quadratic residue diffuser. In *Proc. Int. Computer Music Conf.*, Miami, Nov. 2004.

Jack M. Loomis, Roberta L. Klatzky, John W. Philbeck, and Reginald G. Golledge. Assessing auditory distance perception using perceptually directed action. *Perception and Psychophysics*, 60(6):966–980, 1998.

William L. Martens. Principal components analysis and resynthesis of spectral cues to perceived direction. In *Proc. Int. Computer Music Conf. (ICMC87)*, pages 274–281, Champaign-Urbana, IL, Sep. 1987.

William L. Martens. Perceptual evaluation of filters controlling source direction: Customized and generalized HRTFs for binaural synthesis. *Acoust. Sci. and Tech.*, 24(5):220–232, 2003. Special Issue on Spatial Hearing.

Donald H. Mershon and John N. Bowers. Absolute and relative cues for the auditory perception of egocentric distance. *Perception*, 8(3):311–322, Mar. 1979.

John C. Middlebrooks and David M. Green. Observations on a principal components analysis of head-related transfer functions. *J. Acoust. Soc. Am.*, 92(1):597–599, July 1992.

Jame A. Moorer. About this reverberation business. *Computer Music J.*, 3(2):13–18, Summer 1979.

Karl Pearson. On lines and planes of closest fit to systems of points in space. *Philos. Mag.*, 2(6):559–572, 1901.

Stephen Perrett and William Noble. The effect of head rotations on vertical plane sound localization. *J. Acoust. Soc. Am.*, 102(4):2325–2332, Oct. 1997.



- A. V. Rabinovich. The effect of distance in the broadcasting studio. *J. Acoust. Soc. Am.*, 7(3):199–203, Jan. 1936.
- Vikas C. Raykar, Ramani Duraiswami, and B. Yegnanarayana. Extracting the frequencies of the pinna spectral notches in measured head related impulse responses. *J. Acoust. Soc. Am.*, 118(1):364–374, July 2005.
- Davide Rocchesso. Spatial effects. In Udo Zölzer, editor, *Digital Audio Effects*, pages 137–200. John Wiley & Sons, Chichester Sussex, UK, 2002.
- Davide Rocchesso and Julius O. Smith. Circulant and elliptic feedback delay networks for artificial reverberation. *IEEE Trans. Speech Audio Process.*, 5(1):51–63, Jan. 1997.
- Paul E. Sabine. Architectural acoustics: Its past and its possibilities. *J. Acoust. Soc. Am.*, 11(1):21–28, July 1939.
- Lauri Savioja and Vesa Välimäki. Reducing the dispersion error in the digital waveguide mesh using interpolation and frequency-warping techniques. *IEEE Trans. Speech Audio Process.*, 8(2):184–194, Mar. 2000.
- Lauri Savioja and Vesa Välimäki. Interpolated rectangular 3-d digital waveguide mesh algorithms with frequency warping. *IEEE Trans. Speech Audio Process.*, 11(6):783–790, Nov. 2003.
- Manfred R. Schroeder. Natural-sounding artificial reverberation. *J. Audio Eng. Soc.*, 10(3):219–233, July 1962.
- Manfred R. Schroeder. New method of measuring reverberation time. *J. Acoust. Soc. Am.*, 37(6):1187–1188, June 1965.
- Manfred R. Schroeder. Digital simulation of sound transmission in reverberant spaces. *J. Acoust. Soc. Am.*, 47(2):424–431, Feb. 1970.
- Barbara G. Shinn-Cunningham, Scott Santarelli, and Norbert Kopco. Tori of confusion: Binaural localization cues for sources within reach of a listener. *J. Acoust. Soc. Am.*, 107(3):1627–1636, Mar. 2000.
- Julius O. Smith. *Physical Audio Signal Processing: for Virtual Musical Instruments and Digital Audio Effects, December 2008 Edition*. <http://ccrma.stanford.edu/~jos/pasp/>, 2008. Accessed 15/12/2008.
- Guy-Bart Stan, Jean-Jacques Embrechts, and Dominique Archambeau. Comparison of different impulse response measurement techniques. *J. Audio Eng. Soc.*, 50(4):249–262, Apr. 2002.
- John Stautner and Miller Puckette. Designing multichannel reverberators. *Computer Music J.*, 3(2):52–65, 1982. Reprinted in *The Music Machine*, Curtis Roads (Ed.). Cambridge, The MIT Press, 1989. (pp. 569–582).
- (Lord Rayleigh) John W. Strutt. On the acoustic shadow of a sphere. *Philos. Trans. R. Soc. London*, A-203:87–89, 1904.
- (Lord Rayleigh) John W. Strutt. On our perception of sound direction. *Philos. Mag.*, 13:214–232, 1907.
- R. Teranishi and Edgar A. G. Shaw. External-ear acoustic models with simple geometry. *J. Acoust. Soc. Am.*, 44(1):357–263, July 1968.
- Willard R. Thurlow and Philip S. Runge. Effect of induced head movements on localization of direction of sounds. *J. Acoust. Soc. Am.*, 42(2):480–488, Aug. 1967.
- Scott A. Van Duyne and Julius O. Smith. Physical modeling with the 2-d digital waveguide mesh. In *Proc. Int. Computer Music Conf.*, pages 40–47, Tokio, 1993.
- Scott A. Van Duyne and Julius O. Smith. The tetrahedral digital waveguide mesh. In *Proc. IEEE Workshop Appl. Signal Process. to Audio and Acoust.*, pages 234–237, Mohonk, Oct. 1995.
- Hans Wallach. The role of head movement and vestibular and visual cues in sound localization. *J. Experimental Psychology*, 27:339–368, 1940.
- Frederic L. Wightman and Doris J. Kistler. Resolution of front-back ambiguity in spatial hearing by listener and source movement. *J. Acoust. Soc. Am.*, 105(5):2841–2853, May 1999.
- Pavel Zahorik, Douglas S. Brungart, and Adelbert W. Bronkhorst. Auditory distance perception in humans: A summary of past and present research. *Acta Acustica united with Acustica*, 91(3):409–420, May 2005.
- Dmitry N. Zotkin, Ramani Duraiswami, and Larry S. Davis. Rendering localized spatial audio in a virtual auditory space. *IEEE Trans. Multimedia*, 6(4):553–562, Aug. 2004.



Chapter 7

From audio to content

Giovanni De Poli - Luca Mion - Nicola Orio

Copyright © 2005-2018 Giovanni De Poli - Luca Mion - Nicola Orio
except for paragraphs labeled as *adapted from <reference>*

This book is licensed under the CreativeCommons Attribution-NonCommercial-ShareAlike 3.0 license. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/>, or send a letter to Creative Commons, 171 2nd Street, Suite 300, San Francisco, California, 94105, USA.

7.1 Sound Analysis

Models for the *representation* the information from sound are necessary for the description of the information, from the perceptive and operative point of view. Beyond the models, analysis methods are needed to discover the parameters which allow sound description, possibly lossless from the physical and perceptual properties description.

Audio features extraction. When aiming to the *extraction* of information for sound, we need to discard every feature which is non relevant. This process of feature extraction consists of various steps, starting from pre-processing the sound, then windowing, extraction, and post-processing procedures. An audio signal classification system can be generally represented as represented in Figure 5.1.

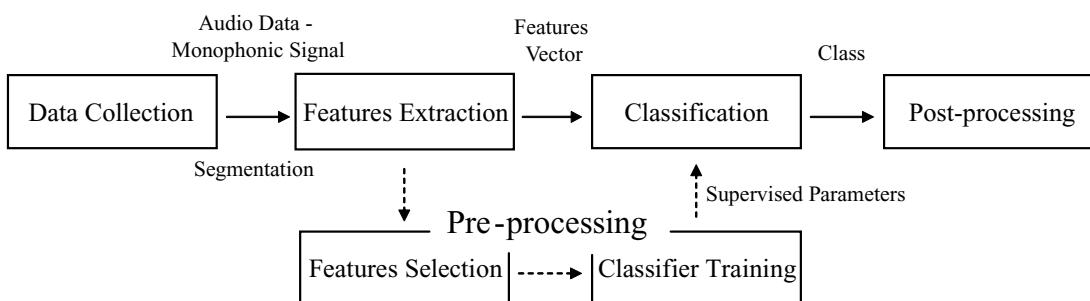


Figure 7.1: Scheme for supervised classification.

Pre-processing: The pre-processing consists of noise reduction, equalization, low-pass filtering. In speech processing (voice has a low-pass behavior) a pre-emphasis is applied by high-pass filtering the signal to smooth the spectrum, to achieve an uniform energy distribution spectrum.

Windowing: Second step is to create frames from a continuous waveform signal. Frames are partially overlapping, as discuss in section 5.1.1

Extraction: Third step consists in obtaining a vector of features for each frame.

Post-processing: In the post-processing phase, the most relevant cues of the features vector are selected. For instance, earlier mel-cepstral coefficients can be lightly weighted when having low frequency noise.

The origins of audio features extraction derive from the research related to speech processing and representation. Initially, they proposed their representation for compressing data (i.e. for telephone applications), and later for speech recognition and synthesis. Features-based representation for non-speech audio was introduced at the end of the 90s, exploring different directions of research from the typical Short Time Fourier Transform; for example, representations based on Cochlear Models, Wavelets, and the MPEG audio compression filterbank. Audio features differ not only in how effective they are but also in their performance requirements, for instance FFT-based features might be more appropriate for a real time applications than a computationally intensive but perceptually more accurate features based on a cochlear filterbank.

Many features have been proposed by different communities in previous studies, e.g. from the musical instrument classification or psycho-acoustical studies. Features can be grouped according to various aspects; *steadiness*, because the features can represent either a value derived from signal, or a parameter from a model of signal behavior (e.g. mean, standard deviation); *time extent* of the description provided by the feature (global or instantaneous descriptors); *abstractness* of the feature, that is how the feature represents (e.g. spectral envelope can be represented via cepstrum or LPC on two different levels of abstraction); *extraction process*, which groups features according if features extracted from waveform, features extracted from transformed signal, features related to models, features based on auditory models.

Audio framework in MPEG-7 Mpeg7 became ISO/IEC 15398 standard in Fall 2001. It is the standard for describing multimedia content that provides the richest multimedia content description tools for applications ranging from content management, organization, navigation, and automated processing. The MPEG-7 standard defines a large library of core description tools, and a set of system tools provides the means for deploying the description in specific storage and transport environments. MPEG-7 addresses many different applications in many different environments, which means it needs to provide a flexible framework for describing multimedia data, including extensibility (using the Description Definition Language) and restrictability (via the MPEG-7 Profiles under specification). The Mpeg7 descriptors for audio documents content consist of low-level and high-level descriptors. The task of supplying suitable descriptors for the extraction of significant characteristic is very interesting and it is also pretty hard. Audio signals can belong to different musical genres, played with various instruments, and can refer to speech-non speech sound, environmental sounds, mechanical sounds etc.

According to the features description proposed for the MPEG-7 standard, the extraction process can be summarized as depicted in Fig. 5.2. In the following, we will present many audio cues that are useful; some of them are also used as descriptors for standard MPEG-7 files, in that cases the Mpeg7 descriptor name will be indicated along with the feature description.



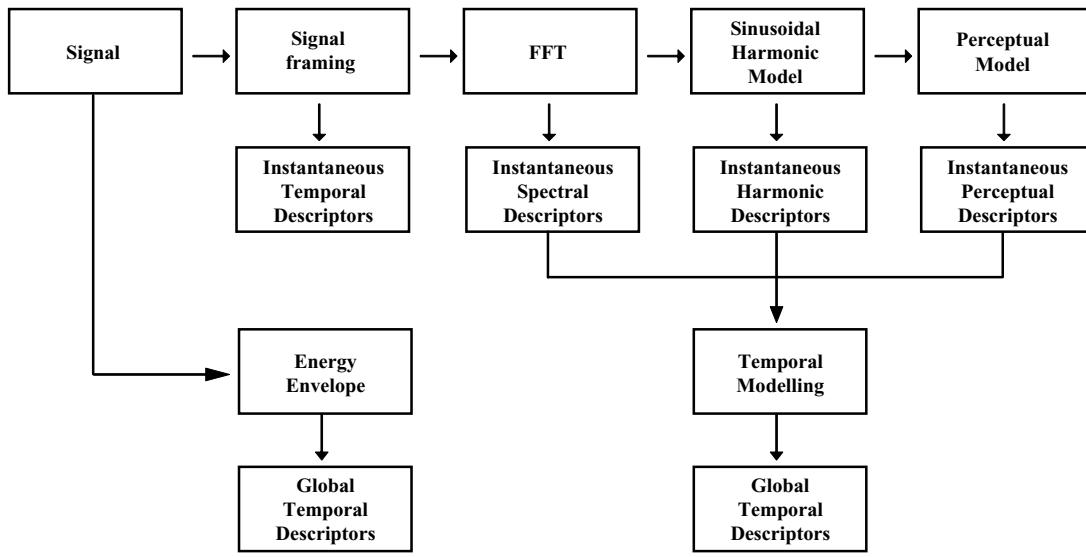


Figure 7.2: Features and extraction processes as proposed by Cuidado Project for MPEG-7 sound descriptors.

7.1.1 Time domain: Short-time analysis

Time domain methods divide according to the time extent of the description provided by the feature. Short and long time analysis techniques apply to different parts of the object (i.e. a sounding note) to describe the attack of the sound (short extent) or the loudness of a note (long extent).

In the case of short-time analysis, an hypothesis of stationarity of sound is now needed, with respect to the sampling rate. That is, we assume the signal to be slowly time-varying over intervals of a few milliseconds to manage the parameters as they were constants within a single frame. Considering the vocal signal for instance, this assumption can be justified by the fact that the words generation is affected by both the vocal chords and the entire phonation apparatus (larynx, mouth, tongue) with modifications not much quick, such to be considered constants within 100-200 ms. Therefore, the signal is divided into frames of e.g. 10 ms. The number of frames computed per second is called frame rate. A tapered window function (e.g. a Gaussian or Hanning window) is applied to each frame to minimize the discontinuities at the beginning and at the end. Consecutive frames are usually considered with some overlap for smoother analysis, and the duration between two frames defines the temporal accuracy, which is chosen according to the target accuracy. This analysis step is called *hop-size H*. On short time analysis, the signal is multiplied by a function $w[k]$ on $k = 0, \dots, N - 1$ which is null out the temporal window.

Windowing Temporal window defines the frame duration. The choice of the window depends on three factors: (1) it has to be short enough to be under the stationarity assumption; (2) it has to be long enough to comfortably allow the parameter computation and to reduce noise if affecting the signal; (3) windows have to entirely cover the parameter, that is the *frame rate* of the parameter has to be at least the inverse of the window duration.

The simplest window is the rectangular window, which gives the poorest quality result of all standard windows and requires a very large number of coefficients to yield a result equivalent to a higher quality window:

$$r[n] = \begin{cases} 1 & \text{for } 0 \leq n \leq N - 1 \\ 0 & \text{otherwise} \end{cases} \quad (7.1)$$

Many applications use longer windows to satisfy the hypothesis of stationarity, but also changing the shape to emphasize the central samples (see Fig. 5.3). An alternative to the rectangular window 5.1 is the Hamming window, which is formulated from cosines and follow sinusoidal curves:

$$h[n] = \begin{cases} 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right) & \text{for } 0 \leq n \leq N-1 \\ 0 & \text{otherwise} \end{cases} \quad (7.2)$$

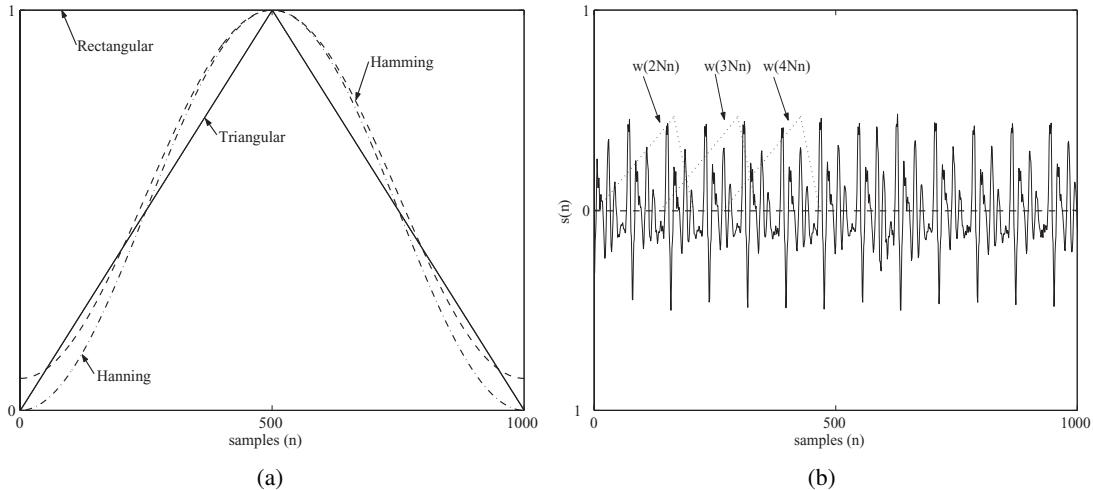


Figure 7.3: Various windows on the left; on the right, three windows over the signals[n], shifted from origin by $2N$, $3N$ e $4N$ samples

Preprocessing Some parameters in the time domain can also be represented by following formulation:

$$Q[n] = \sum_{m=-\infty}^{\infty} T[s[m]]w[n-m] = T[s] * w[n] \quad (7.3)$$

where $T[\cdot]$ is a (even non-linear) transformation weighted by a window $w[n]$. Before being processed, signal can be filtered to select the correct frequency band. In eq. 5.3, $w[n]$ can be a finite impulse response filter (FIR), which allows us to decrease the frame rate (less computational load), or alternatively an IIR filter; an example of IIR window is:

$$w[n] = \begin{cases} a^n & \text{for } n \geq 0 \\ 0 & \text{for } n < 0 \end{cases} \quad (7.4)$$

where $0 < a < 1$;

7.1.1.1 Short-Time Average Energy and Magnitude

For a discrete signal, the *Short-Time Average Energy* is defined as follows:

$$E[n] = \frac{1}{N} \sum_{i=n-N+1}^n s[i]^2 \quad (7.5)$$



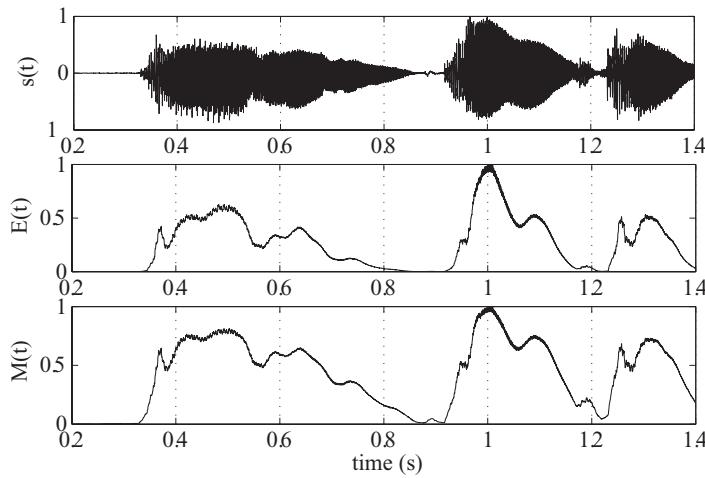


Figure 7.4: On the top: a short excerpt from violin performance of Handel's Flute Sonata in E minor. Other diagrams represent normalized Short-Time Average Energy and Short-Time Average Magnitude, computed using rectangular windows with $N=100$ samples and frame rate equal to signal sampling rate (8kHz).

thus is equivalent to $Q[n]$ in equation 5.3 if $T[\cdot] = (\cdot)^2$.

A drawback of *Short-Time Average Energy* is to be affected when large signals occur. To face this problem, one solution is to define *Short-Time Average Magnitude* as follows:

$$M[n] = \frac{1}{N} \sum_{i=n-N+1}^n |s[i]| \quad (7.6)$$

which is equivalent to 5.3 when $T[\cdot] = |\cdot|$.

M-7.3

Write two MATLAB functions to compute Short-Time Average Energy e Magnitude.

M-5.73 Solution

```

Nframe=100; % numero di campioni per frame
Ns=max(size(s)); % numero di campioni del segnale

for n=1:Ns; % calcola la Short-Time Average Energy
    E(n,1)=sum(s(max(1,n-Nframe+1):n).*...
        s(max(1,n-Nframe+1):n))/Nframe;
end;

for n=1:Ns; % calcola la Short-Time Average Magnitude
    M(n,1)=sum(abs(s(max(1,n-Nframe+1):n)))/Nframe;
end;

% disegna E(t) e M(t)
E=E/max(E); % normalizza E(t)
tempi = (1/fS)*[1:max(size(E))]; subplot(2,1,1);

```

```

plot(tempi,E); xlabel('time (s)'); ylabel('E(t)');
M=M/max(M); % normalizza M(t)
tempi = (1/fS)*[1:max(size(M))]; subplot(2,1,2);
plot(tempi,M); xlabel('time (s)'); ylabel('M(t)');

```

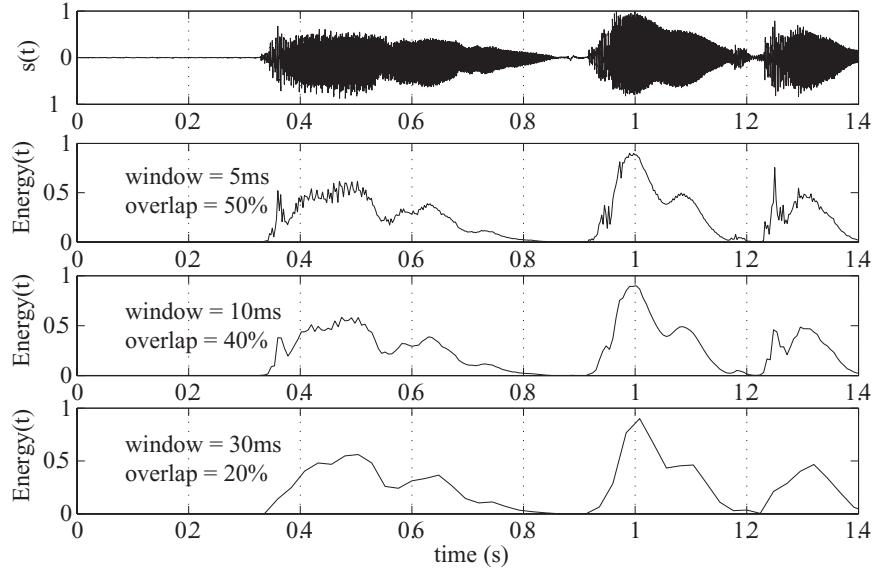


Figure 7.5: On the top: a short excerpt from violin performance of Handel's Flute Sonata in E minor. Other diagrams represent Short-Time Average Energy computed with different Hamming windows.

7.1.1.2 Short-Time Average Zero-Crossing Rate

Zero-Crossing Rate (*ZCR*) can yield useful information about the spectrum with low computational costs. *ZCR* represents the number of crossing through the zero signal, and it is mathematically described as the changing of the sign of two following samples. For narrow-band signals (e.g. sinusoids or band-pass filter output), from *ZCR* we obtain the fundamental frequency (*F0*) of the signal:

$$F0 = \frac{ZCR * F_S}{2} \quad (7.7)$$

where F_S is the signal sampling rate and ZCR is expressed as *zero crossing* for sample. We can have $ZCR = Q[n]$ when in 5.3 we use $T[s[n]] = |\text{sign}(s[n]) - \text{sign}(s[n-1])|/2$, and scaling the window $w[n]$ by a factor $1/N$; thus, we have

$$Z[n] = \frac{1}{N} \sum_{m=n-N+1}^n \frac{|\text{sign}(s[m]) - \text{sign}(s[m-1])|}{2} w[n-m] \quad (7.8)$$

where the sign of $s[n]$ is defined as follows:

$$\text{sign}(s[n]) = \begin{cases} 1 & \text{for } s[n] \geq 0 \\ -1 & \text{otherwise} \end{cases} \quad (7.9)$$



In Figure 5.6 the Zero crossing Rate of the word /sono/ is shown. Notice the high ZCR values at the beginning in correspondence of the unvoiced /S/ and low values for the voiced part. This properties can be exploited to distinguish voiced (periodic) from unvoiced sounds.

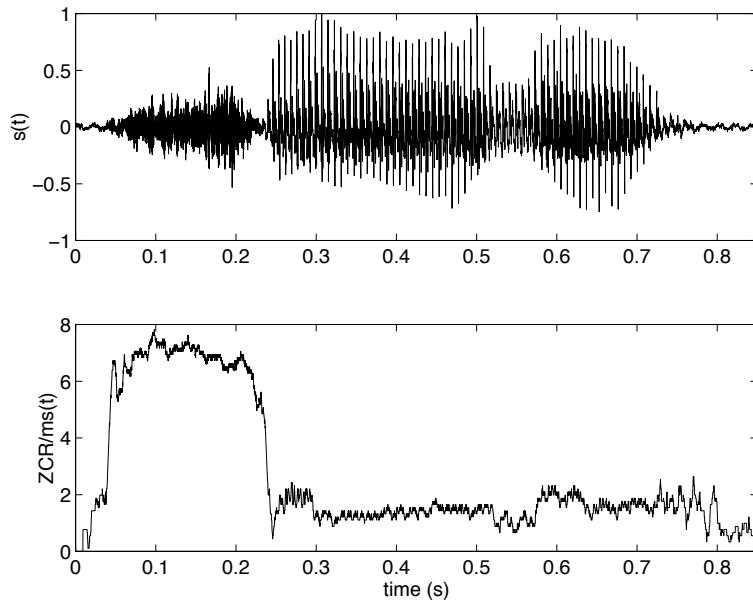


Figure 7.6: Zero-Crossing Rate of the word /SONO/.

M-7.4

Write a MATLAB function for Zero Crossing Rate computation.

M-5.74 Solution

```
Nframe = 100; % numero di campioni per frame
Ns = max(size(s));

for n = 1+Nframe:Ns; % calcola la Short-Time Average ZCR
    Z(n,1) = sum(abs(sign(s(n-Nframe+1:n))-...
        sign(s(n-Nframe:n-1))))/2/Nframe;
end;

Z=Z*fS/1000; % Zero-Crossing per ms

% disegna Z(t):
t = (1/fS)*[1:max(size(Z))];
plot(t,Z); xlabel('time (s)'); ylabel('ZCR/ms(t)');
```

7.1.1.3 Short-Time Autocorrelation Function

Signal *autocorrelation* is the inverse Fourier transform of the spectral density of the signal energy $C_s(f)$. It is formulated as follows:

$$\mathcal{F}[\phi[k]] = C_s(f) = |S(f)|^2 \quad (7.10)$$

For a discrete signal, it is defined as

$$\phi[k] = \sum_{m=-\infty}^{\infty} s[m]s[m+k] \quad (7.11)$$

Autocorrelation preserves the information related to harmonics, formants amplitude and their frequencies. Equation 5.11 shows that $\phi[k]$ is somehow representing the signal likeness to its shifted version. So, it will assume higher values when occurring delays k such that $s[m]$ and $s[m+k]$ have similar waveforms.

Some important properties of $\phi[k]$ are the followings:

1. it is an even function: $\phi[k] = \phi[-k]$
2. when $k = 0$ $\phi[k]$ takes its maximum value, $\phi[0] \geq |\phi[k]| \ \forall k$
3. $\phi[0]$ corresponds to the signal energy (or to the average power if the signal is periodic or non-deterministic)
4. if the signal is periodic with period P , the autocorrelation is periodic with the same period of the analyzed signal: $\phi[k] = \phi[k + P]$ (this is an important property when the signal periodicity has to be estimated). In fact it has maximal values at time lag P , $2P$, and so on. If the sound is quasi-periodic, we will have local maxima at lag P , $2P$, and so on (see Fig. 5.7).

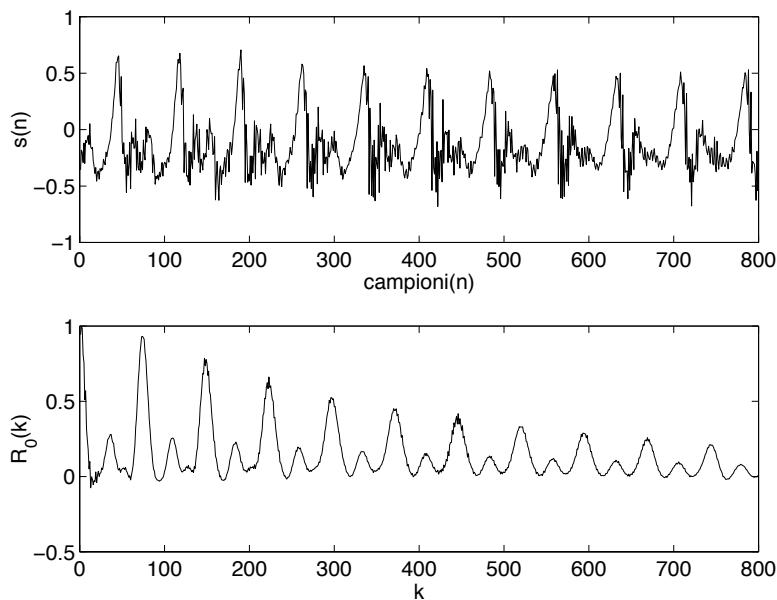


Figure 7.7: Autocorrelation of a voiced sound..

M-7.5

Write a MATLAB function for computing the *Short-Time Autocorrelation Function*.

M-5.75 Solution



```

Ns = max(size(s)); %no. of samples
window = ones(Ns,1); %rectangular window
s_w = s.*window;

for k = 1:Ns-1; %compute ST autocorrelation
    R0(k) = sum(s_w(1:Ns-k) .* s_w(k+1:Ns));
end;
%plots
R0=R0/max(abs(R0)); %normalize R0
plot(1:max(size(R0)),R0); xlabel('k'); ylabel('R_0(k)');

```

Short-Time Autocorrelation Function (STAF) is applied for pitch extraction applications and speech-non speech signal discriminations. It is yielded by Eq. 5.11 after filtering the signal with windows $w[n]$:

$$R_n[k] = \sum_{m=-\infty}^{\infty} s[m]w[n-m]s[m+k]w[n-k-m] \quad (7.12)$$

This equation can be seen in the form:

$$R_n[k] = \sum_{m=-\infty}^{\infty} [s[n+m]w'[m]] \cdot [s[n+m+k]w'[k+m]] \quad (7.13)$$

where $w'[n] = w(-n)$; if we assume that $w'[n]$ has finite duration N we obtain:

$$R_n[k] = \sum_{m=0}^{N-1-k} [s[n+m]w'[m]] \cdot [s[n+m+k]w'[k+m]] \quad (7.14)$$

7.1.1.4 Short-Time Average Magnitude Difference Function

Beyond the *Short-Time Autocorrelation Function*, the detection of F0 can also be faced by means of the *Short-time Average Magnitude Difference Function* (AMDF). For a periodic signal with period P , succession $d[n] = s[n] - s[n - k]$ is zero when $k = 0, \pm P, \pm 2P, \dots$, so we can consider the absolute value of the difference of $s[m]$ and $s[m - k]$ instead of their product:

$$\gamma_n[k] = \sum_{m=-\infty}^{\infty} |s[n+m]w[m] - s[n+m-k]w[m-k]| \quad (7.15)$$

We can have a simpler formulation when $w[n]$ is rectangular, with duration N:

$$AMDF[k] = \sum_{m=k}^{N-1} |s[m] - s[m-k]| \quad (7.16)$$

The AMDF of the signal of Fig. 5.7 is shown in fig. 5.8.

M-7.6

Write a MATLAB function for *Short-time Average Magnitude Difference Function* computing.

M-5.76 Solution



```

Ns=max(size(s));      % numero di campioni

window=ones(ceil(Ns/2)+1,1);    % finestra rettangolare

for k=1:floor(Ns/2)-1;  % calcola la Short-Time AMDF
    STAMDF(k) = sum(abs(s(floor(Ns/2):Ns).* window - ...
        s(floor(Ns/2)-k:Ns-k).* window));
end;

% disegna STAMDF(t):
STAMDF=STAMDF/max(STAMDF);      % normalizza STAMDF(t)
plot(1:max(size(STAMDF)),STAMDF); xlabel('k'); ylabel('AMDF(k)');

```

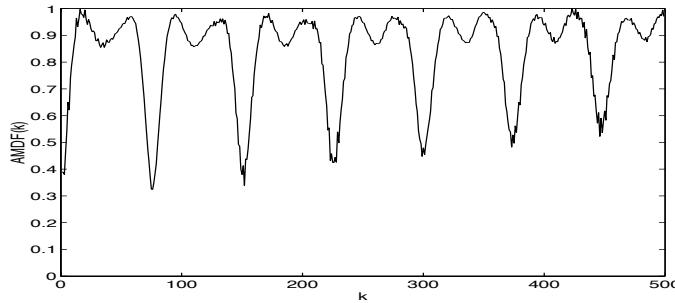


Figure 7.8: Short time AMDF of the voiced sound of fig. 5.7.

7.1.2 Audio segment temporal features

Descriptors computed after long-time localization of sound events (segmentation). These descriptors are extracted from the whole signal.

7.1.2.1 Temporal envelope estimation

Amplitude envelope can be defined as the variation of the amplitude of a note while sounding. This is often described through four phases called *attack*, *decay*, *sustain* *release* (ADSR). The basic idea to extract the temporal envelope starts with a low-pass filtering (i.e. 10 - 30 Hz), in order to extract the slow time changing components. One method to detect the envelope is based on *Short-Time Average Energy*, where window acts as a low-pass filter:

$$env[n] = \sqrt{E[n]}$$

7.1.2.2 ADSR envelope modelling

Modelling Attack, Decay, Sustain and Release of a note comes from the time representation of energy envelope. This feature is not always easily achievable because of overlapping notes with noise or with other notes. The estimation of attack can be achieved by means of either fixed or adaptive thresholds techniques, often empirically tuned. Fixed threshold methods consists of taking into account the possible presence of noise setting a threshold (e.g. to 20%) on energy envelope. Also, in order to take into account the possibility that the maximum of the envelope does not occur at the end of the attack, another threshold is set (e.g. to 90%). Adaptive techniques are based on the behavior of the signal during the attack; the



best threshold is chosen along multiple notes by repeated tunings, according to the slope of the threshold crossing. I will discuss some onset detection techniques in Sec. 5.4. This set of features consists of:

Log-Attack Time [In Mpeg7 is LogAttackTime]"

$$LAT = \log_{10}(attack_time)$$

where *attack_time* is the time duration of note attack. This feature has been proven to be strongly related to perceptual description of timbres.

Temporal Centroid [In Mpeg7 is TemporalCentroid]"

$$TC = \frac{\sum_t env(t) \cdot t}{\sum_t env(t)}$$

where *env(t)* is the temporal envelope. This is a useful parameter to distinguish percussive sounds from sustained sounds.

Effective Duration is a measure of the time the signal is perceptually meaningful. It is approximately given by the time the energy is above a given threshold (e.g. 40%).

7.1.2.3 Pitch detection (*F0*) by time domain methods

The general problem of fundamental frequency estimation is to take a portion of signal and to find the dominant frequency of repetition.

Many applications are based on the detection of pitch, i.e. the dominant perceived frequency of a sound. The basic problem is to extract from a sound signal the fundamental frequency *F0*, which is the lowest sinusoidal component, or partial, which relates well to most of the other partials.

Difficulties arise from: (i) Not all signals are periodic; (ii) Those that are periodic may be changing in fundamental frequency over the time of interest; (iii) Signals may be contaminated with noise, even with periodic signals of other fundamental frequencies; (iv) Signals that are periodic with interval *T* are also periodic with interval *2T*, *3T* etc, so we need to find the smallest periodic interval or the highest fundamental frequency; (v) Even signals of constant fundamental frequency may be changing in other ways over the interval of interest.

In a pitched signal, most partials are harmonically related, meaning that the frequency of most of the partials are related to the frequency of the lowest partial by a small whole-number ratio. The frequency of this lowest partial is *F0* of the signal. The simplest approach to periodicity evaluation is based on the investigation of the time domain waveform. We may test each *F0* hypothesis by testing how well the signal will resemble a delayed version of itself. An evaluation criteria can be either the correlation or the difference between the signal and its delayed version.

From the Short-Time Autocorrelation Function we obtain the information on the signal periodicity (fig. 5.9) by means of *k_M*, that is the first maximum after the one related to *k* = 0:

$$F0 = \frac{F_S}{k_M} \quad (7.17)$$

where *F_S* is the signal sampling rate. On the other side, if we use the Short-Time AMDF we have to consider the first minimum *k_m* after the one related to *k* = 0 (fig. 5.10). However sometimes we get a harmonic or sub-harmonic, depending on the shape of the spectrum. Whitening the signal by center-clipping is effective in minimizing this problem.

A pitch estimator normally proceeds in three steps:



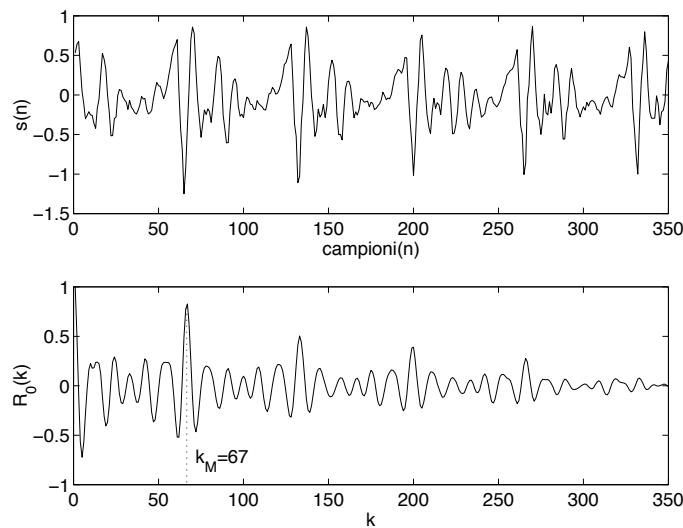


Figure 7.9: Frame of the phoneme /OH/ and its Short-Time Autocorrelation Function. The position of the second maximum at k_M indicates the pitch period.

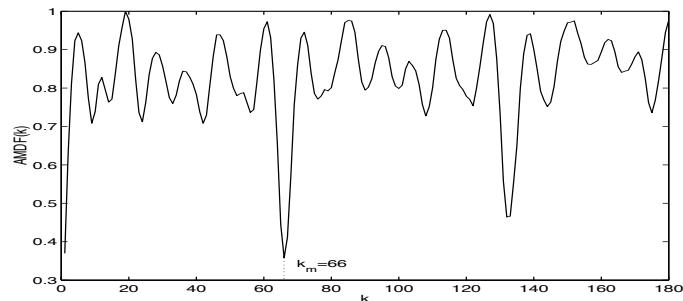


Figure 7.10: AMDF of the frame of the phoneme /OH/ of fig. 5.9. The position of the second minimum at k_m indicates the pitch period.

- pre-processing: the signal is filtered to remove high frequency components;
- pitch extraction;
- post-processing to correct possible errors.

M-7.7

Compute the pitch with *Short-Time Autocorrelation Function*.

M-5.77 Solution

```
inizio=floor(fS*0.001); % salta il primo massimo
[massimo,kM] = max(R0(inizio:max(size(R0)))); 
kM=kM + inizio -1;
F0=fS/kM;
```

M-7.8

Compute the pitch with *Short-time Average Magnitude Difference Function*.



M-5.78 Solution

```
inizio=floor(fS*0.001); % salta il primo minimo
[minimo,km] = min(STAMDF(inizio:max(size(STAMDF)))); 
km=km + inizio -1;
F0=fS/km;
```

7.1.3 Frequency domain analysis

We have already seen in the signal based sound modeling chapter how a sound can be represented in the frequency domain. Moreover we presented analysis methods which allow the estimation of the spectral representation. In the following sections other methods for sound analysis in the frequency domain will be presented.

7.1.3.1 Energy features

Sinusoid plus noise representation Let's suppose $x[n]$ to be a sound, which is constituted by two components $x_S[n]$ (sinusoidal), and $x_R[n]$ (residual): $x[n] = x_S[n] + x_R[n]$, where $x_S[n] = \sum_{i=1}^I a_i \cos(n2\pi f_i[n]/F_s + \phi_i[n])$.

In this form, a_i represents the amplitude (in linear scale) and f_i the frequency of i -th partial. These are time-changing parameters, which often are considered constant during a frame. The most used features, derived from this representation, are:

Total amplitude of the sinusoidal component, resulting by sum of the partials (dB expressed) within the frame:

$$AS_{tot} = 20 \log_{10} \left(\sum_{i=1}^I a_i \right)$$

where a_i is the amplitude of i -th partial;

Total amplitude of the residual component, resulting by sum of absolute values of the residual within the frame:

$$AR_{tot} = 20 \log_{10} \left(\sum_{n=0}^{M-1} |x_R[n]| \right) = 20 \log_{10} \left(\sum_{k=0}^{N-1} |X_R[k]| \right)$$

Total amplitude of the sound

$$\begin{aligned} A_{tot} &= 20 \log_{10} \left(\sum_{n=0}^{M-1} |x[n]| \right) = 20 \log_{10} \left(\sum_{k=0}^{N-1} |X[k]| \right) \\ &= 20 \log_{10} \left(\sum_{i=1}^I a_i + \sum_{k=0}^{N-1} |X_R[k]| \right) \end{aligned}$$

Total Energy [In Mpeg7 is AudioPower]" estimates the signal power at a given time. It is estimated directly from the signal frame.

Noise to Harmonic Ratio - NHR is defined as the ratio between the energy of noise and the energy of harmonic part. For sustained sounds, energy modulation and fundamental frequency modulation are taken into account.



7.1.3.2 Spectral shape features

Spectral centroid [In Mpeg7 is AudioSpectrumCentroid]” is the barycenter of the spectrum. It is computed considering the spectrum as a distribution which values are the frequencies and the probabilities to observe there are the normalized amplitude.

$$BR = \frac{\sum_{k=0}^{N-1} k|X[k]|}{\sum_{k=0}^{N-1} |X[k]|} \cdot \frac{F_S}{N}$$

In the case of harmonic sounds, brightness is related to the fundamental frequency F_0 :

$$BR_{F0} = \frac{\sum_{i=1}^I i a_i}{\sum_{i=1}^I a_i} = \sum_{i=1}^I i w_i$$

Bandwidth is the difference between the lowest and highest frequency components of a signal:

$$BW = \frac{\sum_{k=0}^{N-1} |X[k]| \cdot |f_k - BR|}{\sum_{k=0}^{N-1} |X[k]|}$$

Spectral spread [In Mpeg7 is AudioSpectrumSpread]” is the spread of the spectrum around its mean value, i.e. the variance of the spectral distribution.

Spectral skewness gives a measure of the asymmetry of a distribution around its mean value.

Spectral slope represents the amount of decreasing of the spectral amplitude. It is computed by linear regression of the spectral amplitude.

$$Stilt = \frac{1}{\sum_{i=1}^I t_i^2} \cdot \sum_{i=1}^I \frac{t_i a_i}{w_i}$$

where

$$t_i = \frac{1}{w_i} \left(f_i - \frac{\sum_{i=1}^I f_i / w_i^2}{\sum_{i=1}^I 1 / w_i^2} \right)$$



Spectral decrease still represents the decreasing of spectral amplitude, and it is supposed to be more correlated to human perception.

$$SD = \frac{1}{\sum_{i=2}^I a[i]} \sum_{i=2}^I \frac{a[i] - a[1]}{i - 1}$$

Spectral roll-off is the frequency R_s so that 85% of the amplitude distribution is below this frequency. It is correlated to harmonic/noise cutting frequency.

$$\sum_{k=1}^{R_s} |X[k]| = 0.85 \cdot \sum_{k=1}^{N-1} |X[k]|$$

Spectral Flux is defined as the Euclidean distance between two amplitude spectrums of two close frames.

$$SF = \sqrt{\sum_{k=1}^{N-1} [N_t[k] - N_{t-1}[k]]^2}$$

where $N_t[k]$ and $N_{t-1}[k]$ are respectively the spectral amplitude of frame FFT at instants t and $t - 1$.

7.1.3.3 Harmonic features

Fundamental frequency [In Mpeg7 is AudioFundamentalFrequency]" is the frequency that a periodic waveform repeats itself. There are many methods in the literature to detect the fundamental frequency F_0 . For the Mpeg7 descriptor it is computed using the maximum likelihood algorithm. The method used in our experiments will be described in section ??.

Noisiness [In Mpeg7 is AudioHarmonicity]" is the ratio between the energy of the noise an the total energy. It is close to zero for purely harmonic sounds.

$$Noisiness = \frac{\sum_{n=0}^{M-1} |x_R[n]|}{\sum_{n=0}^{M-1} |x[n]|}$$

Inharmonicity represents the divergence of the signal spectral components from a purely harmonic signal. It is computed as an energy weighted divergence of the spectral components. The range is [0,1].

$$HD = \sum_{i=1}^I |f_i - iF_o| \cdot w_i$$

Harmonic Spectral Deviation [In Mpeg7 is HarmonicSpectralDeviation]" is the deviation of the amplitude harmonic peaks from the global envelope.

$$HDEV = \frac{1}{I} \sum_{i=1}^I [a_i - spec_env(f_i)]$$

where $spec_env(f_i)$ is the smoothed spectral envelope computed at frequency f_i of i -th harmonic.



Even-odds energy is useful to distinguish sounds like clarinet sound, which has low energy on even harmonics, differing from the trumpet sound which has similar behavior for both kinds of harmonics.

$$OER = \frac{\sum_{i=even} a_i^2}{\sum_{i=odd} a_i^2}$$

7.1.3.4 Pitch detection from the spectrum

For an harmonic signal F0 is the frequency so that its integer multiple explain the content of the digital spectrum. In fact for a periodic sound, all the partials are multiple of the fundamental frequency, that is $f_i = iF0$. In real sounds this is not completely true, and F0 can be calculated as the weighted sum of frequencies, normalized over all the harmonics.

$$F0 = \sum_{i=1}^I \frac{f_i}{i} \cdot w_i \quad (7.18)$$

where

$$w_i = \frac{a_i}{\sum_{i=1}^l a_i} \quad (7.19)$$

is the weight of i-th harmonic, with reference to the total sinusoidal component, and a_i is its amplitude. For signals with a more distributed spectrum, cepstrum analysis (sect. 5.2.2) is the form more conventionally used to make the analysis of pitch.

7.1.4 Perceptual features

Specific Loudness is associated to each Bark band z , see Moore et al. [1997] for precise definitions. It can be simply defined as

$$N'(z) = E(z)^{0.23}$$

where $E(z)$ is the energy in the z -th bark-band.

Total Loudness is the sum of individual loudness.

$$N = \sum_{z=1}^{band} N'(z)$$

Sharpness is the perceptual equivalent to the spectral centroid, computed through the specific loudness of he Bark bands.

$$Sh = 0.11 \cdot \frac{\sum_{z=1}^{band} z \cdot g(z) \cdot N'(z)}{N}$$

where z is the index of the band and $g(z)$ is a function defined as follows:

$$g(z) = \begin{cases} 1 & \text{for } z < 15 \\ 0.066 \exp(0.171z) & \text{for } z \geq 15 \end{cases} \quad (7.20)$$



7.2 Spectral Envelope estimation

Families of musical instruments can often be described by typical spectral envelope. When processing sound, operations that preserves the spectral envelope are roughly expressed as *pith shifting* operations with timbre preservations. Various techniques are used to represent the shape of a stationary spectrum. In sect. 2.4.7.2 we already presented the Linear Predictive (LPC) analysis.

M-7.9

In LPC analysis, the position of formants (resonances) is related to the poles of the estimated transfer function. Factorize the denominator of the transfer function and estimate the frequency of the formants. Note that if θ_k is the argument of z_k complex conjugate zero of the denominator, then its corresponding resonant frequency f_k derives from $\theta_k = 2\pi f_k / F_s$; the formant bandwidth B_k is related to the zero modulus by $|z_k| = \exp(-\pi B / F_s)$.

7.2.1 Filterbank

Filter-bank is a classical spectral analysis technique which consists in representing the signal spectrum by the log-energies at the output of a filter-bank, where the filters are overlapping band-pass filters spread along the frequency axis. This representation gives a rough approximation of the signal spectral shape while smoothing out the harmonic structure if any. When using variable resolution analysis, the central frequencies of the filters are determined so as to be evenly spread on the warped axis and all filters share the same bandwidth on the warped axis.

M-7.10

Write a MATLAB function for the spectral envelope computing, with the filterbank approach. Try a filterbank of frequency linearly spaced filters and logarithmic spaced filters (e.g. third octave filters).

M-7.11

Write a MATLAB function for the spectral envelope computing, with the gamma tone filterbank approach. Look in the literature or on the web for gammatone filter definition. gamma tone filters simulate the behaviour of the cochlea.

7.2.2 Spectral Envelope and Pitch estimation via Cepstrum

Families of musical instruments can often be described by typical spectral envelope. When processing sound, operations that preserves the spectral envelope are roughly expressed as *pith shifting* operations with timbre preservations. Various techniques are used to represent the shape of a stationary spectrum. Cepstrum method allows the separation of a signal $y[n] = x[n] * h[n]$, (source-filter model), where the source $x[n]$ passes through a filtered described by impulse response $h[n]$. Signal spectrum $y[n]$ results $Y[k] = X[k] \cdot H[k]$, which is the product of two spectrums (k is the discrete-frequencies index). The former is related to the source spectrum, and the latter to the filter spectrum. It's pretty difficult to separate these two spectrums, thus what is usually done is to extract the envelope (real) of the filter, and making the phase related to the source only. Cespstrum idea is based on the properties of logarithms : $\log(a \cdot b) = \log(a) + \log(b)$. Taking into account the logarithm of the absolute value of spectrum $Y[k]$, we get:

$$\log |Y[k]| = \log(|X[k] \cdot H[k]|) = \log |X[k]| + \log |H[k]| \quad (7.21)$$

If we consider the diagram for $\log |Y[k]|$ as a time-domain signal, we can distinguish two components: a quick oscillation, due to harmonic structure (rows), and a slower behavior related to the filter resonances



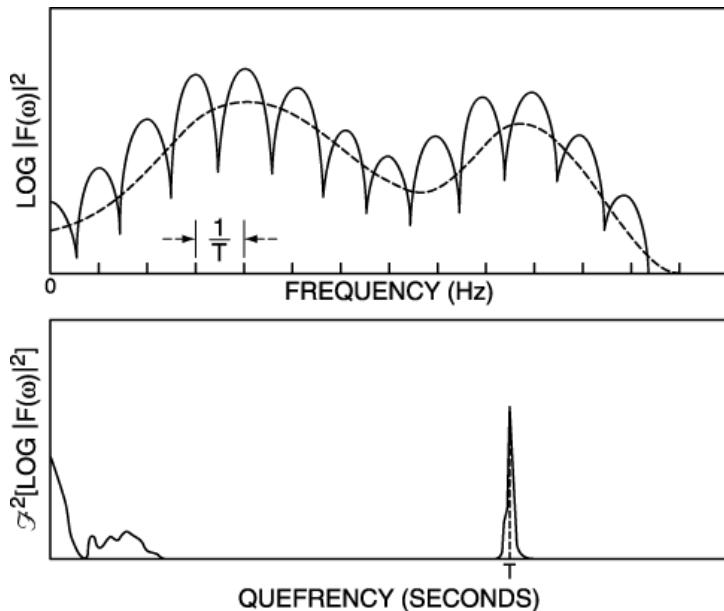


Figure 7.11: Example of cepstrum: on the top $\log |Y[k]|^2$; below, the related cepstrum $c[n] = \text{DFT}^{-1}(\log |Y[k]|)$

(spectral envelope). We can separate the components by low/high-pass filtering signal $\log |Y[k]|$ (see Fig. 5.11, top). For components separation, one method is based on IFFT:

$$\text{DFT}^{-1}(\log |Y[k]|) = \text{DFT}^{-1}(\log |X[k]|) + \text{DFT}^{-1}(\log |H[k]|) \quad (7.22)$$

The part of $\text{DFT}^{-1}(\log |Y[k]|)$ towards the origin describes the spectral envelope, far from excitation. There is a sort of line in correspondence with $\log |Y[k]|$ periodicity, and thus to sound periodicity (see Fig. 5.11, below). At this point the cepstrum name origin comes out. Cepstrum word corresponds to spectrum when backward reading the former (ceps) part. The pitch can be estimated by the following procedure:

```

compute cepstrum every 10-20 msec
search for periodicity peak in expected range of n_p
if found and above threshold
    sound is periodic
    pitch=location of cepstral peak
else sound is not periodic

```

A drawback of the cepstral coefficients is the linear frequency scale. Perceptually, the frequency ranges 100-200Hz and 10kHz -20kHz should be approximately equally important, and standard cepstral coefficients do not take this into account.

We can notice that the maxima of spectral envelope corresponds to resonances (formants) which are very important to differentiate the vowels. In Fig. 5.12 it is shown how to individuate the formants from the spectral envelope.

M-7.12

Estimate the formants of a voice in a song and plot their position on the spectrogram.



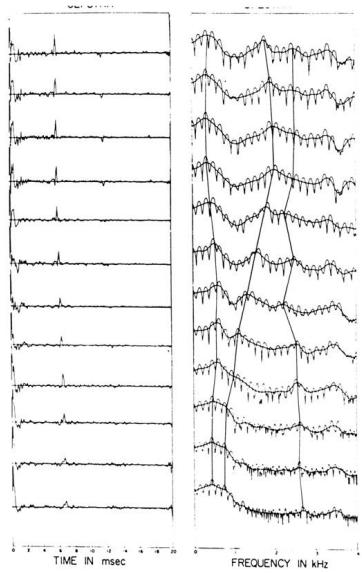


Fig. 7.15 Automatic formant estimation from cepstrally smoothed log Spectra. (After Schaefer and Rabiner [11].)

Figure 7.12: Automatically formant estimation from cepstrally smoothed log Spectra [from Schaefer Rabiner].

7.2.3 Analysis via mel-cepstrum

Psychoacoustic studies have shown that human perception of frequencies goes with a logarithmic-like scale. Each tone of f (Hz), corresponds to a subjective value of pitch measured on the *mel* scale. As reference on the mel scale, 1000 Hz match 1000 mel. To obtain a value in the mel scale, a non-linear transformation on frequency scale is applied (see Fig. 5.13(a)), computed as follows:

$$\text{mel}(f) = \begin{cases} f & \text{if } f \leq 1 \text{ kHz} \\ 2595 \log_{10} \left(1 + \frac{f}{700} \right) & \text{if } f > 1 \text{ kHz} \end{cases} \quad (7.23)$$

To apply the mel scale to cepstrum, triangular band-pass filterbanks are used, with central frequency in K mel values (see Fig. 5.13(b)). Each filter has bandwidth equal to the distance to previous filter central frequency, multiplied by two. First filter starts from 0. Thus, the bandwidth of filters below 1000 Hz is 200 Hz; then it will raise exponentially. Mel-cesptrum aim to estimate the spectral envelope of this filterbank output.

When Y_n is the logarithm of energy exiting from channel n , we can use the discrete time cosine transform DCT to obtain the mel-cepstral coefficients MFCC (mel frequency cepstral coefficients) by means of following equation:

$$c_k = \sum_{n=1}^N Y_n \cos \left[k \left(n - \frac{1}{2} \right) \frac{\pi}{N} \right] \quad k = 0, \dots, K \quad (7.24)$$

We can use the first K_m (with $K_m < K$) coefficients to draw a simplified spectral envelope $\tilde{C}(\text{mel})$, similar to what we have seen for cepstrum:

$$\tilde{C}(\text{mel}) = \sum_{k=1}^{K_m} c_k \cos(2\pi k \frac{\text{mel}}{B_m}) \quad (7.25)$$

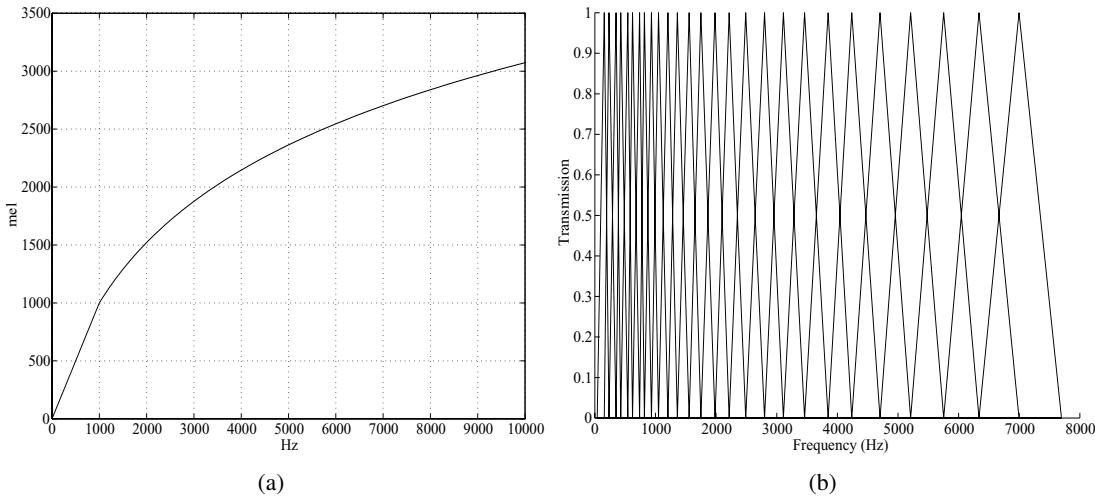


Figure 7.13: (a) Transformation from Hz to mel. (b) Mel-scale filterbank.

where B_m = is the bandwidth, expressed in mel. A typical value for K_m in music classification is $K_m = 20$. We can notice that the coefficient c_0 is the mean value (in dB) of the energy in the channel of the filterbank. Thus it is related to the signal energy and often is not considered when we want to compare two spectral envelopes.

M-7.13

Write a MATLAB function for the spectral envelope computing, with the mel-cepstral approach and experiment it for different kinds of sounds. Compare the results obtained with the different spectral envelope algorithms.

In fig. 5.14 an example of mel-cesptrum analysis of a clarinet tone is shown. Spectra in dB, represented on a logarithmic frequency scale are compared: tone spectrum (high left); spectral envelope reconstructed with first 6 mel cepstral coefficients (low right), spectral envelope rebuilt from LPC analysis (low left); spectral envelope estimated with all mel cepstrum coefficients (low right).

7.3 Mid-level features

7.3.1 Chromagram

The chromagram, also called Harmonic Pitch Class Profile, shows the distribution of energy along the pitches or pitch classes. It is computed in two steps:

- First, the spectrum is computed in the logarithmic scale, with selection of the 20 highest dB, and restriction to a certain frequency range that covers an integer number of octaves.
- Then, the spectrum energy is redistributed along the different pitches (i.e., chromas).

The chromagram can be represented along the full pitch scale (Figure 5.15) or can be wrapped along the 12 pitch classes (Figure 5.16).

Calculating the chromagram on a framed audio signal, it is possible to represent the evolution in time of the spectral energy along the pitch classes (Figure 5.17).



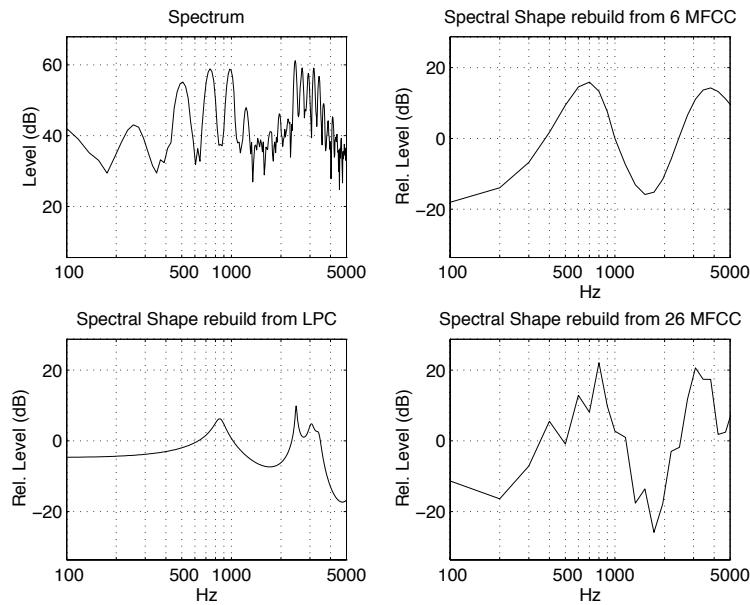


Figure 7.14: Example of mel-cepstrum analysis of a clarinet tone: tone spectrum (high left); spectral envelope reconstructed with first 6 mel cepstral coefficients (low right), spectral envelope rebuilt from LPC analysis (low left); spectral envelope estimated with all mel cepstrum coefficients (low right).

7.3.2 Keystrength

The Keystrength is a score between -1 and +1, associated with each possible key candidate. In other words, this feature allows an estimation of the prevalent key and mode of a musical excerpt. The Keystrength can be computed through a cross-correlation of the chromagram, wrapped and normalized, with similar profiles representing all the possible tonality candidates (Figure 5.18). These profiles have been estimated in previous studies, such as Krumhansl, 1990¹ and Gomez, 2006².

The resulting graph (Figure 5.19) indicate the cross-correlation score for each different tonality candidate.

¹Krumhansl, C. L., Cognitive foundations of musical pitch. Oxford UP, 1990.

²Gomez, E., Tonal description of music audio signal. Phd thesis, Universitat Pompeu Fabra, Barcelona, 2006.

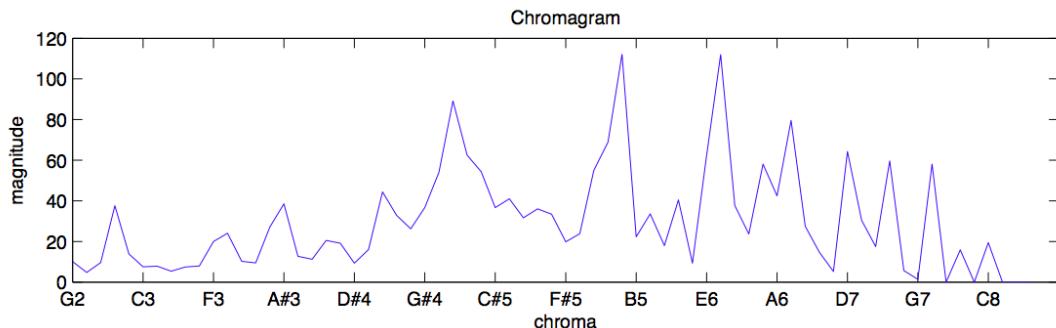


Figure 7.15: Chromagram in the pitch range G2 - C8.

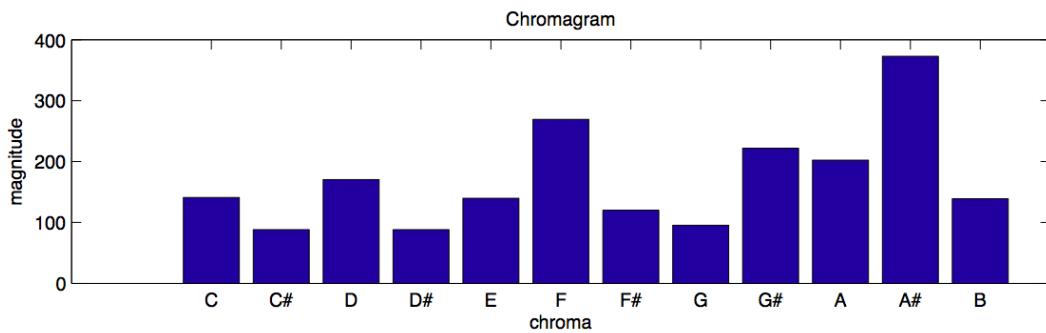


Figure 7.16: Wrapped chromagram.

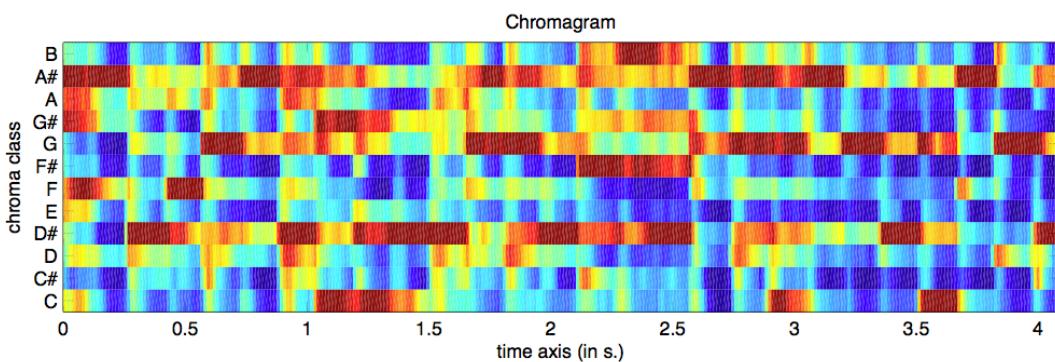


Figure 7.17: Chromagram of a framed signal.

7.3.3 Tempo

The tempo can be estimated by detecting periodicities from the onset detection curve. Various approaches can be followed:

- computing an autocorrelation function of the onset detection curve (Figure 5.20);
- computing a spectral decomposition of the onset detection curve;
- combining both strategies: the autocorrelation function is translated into the frequency domain in order to be compared to the spectrum curve, and the two curves are subsequently multiplied.

Then a peak picking is applied to the autocorrelation function or to the spectrum representation (Figure 5.21).

7.4 Onset Detection

This section is dedicated to the problem of the segmentation of the audio signal, through the onset detection. The concept of onset can be defined as the instant in which a new event starts. Event, in this context, can be defined as an auditory phenomenon that shows continuity inside the normal limits of perception. These auditory phenomena can be expressive features (legato, vibrato, etc), timbre or notes. Note onset detection and localization is useful in a number of analysis and indexing techniques



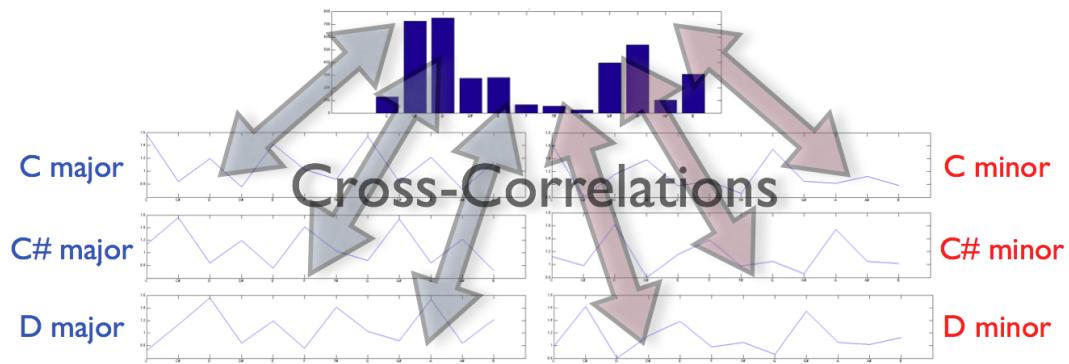


Figure 7.18: The Chromagram is compared with the profiles related to the different tonality candidate.

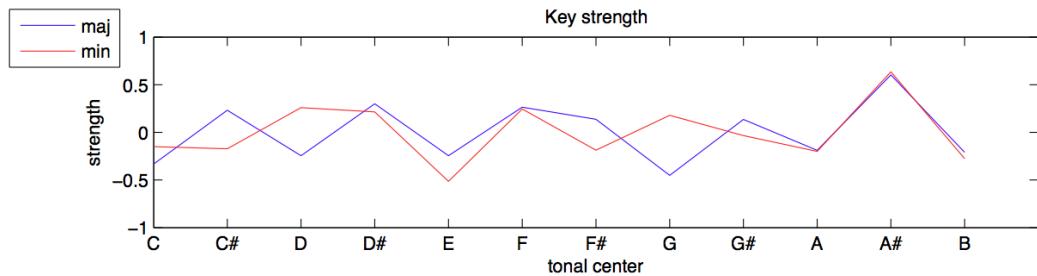


Figure 7.19: Keystrength computed on a musical excerpt.

for musical signals. In most cases, onset will coincide with the start of the transient of the note, or the earliest time at which the transient can be reliably detected.

First two methods widely used for the onset detection, based in frequency domain and local energy detection, will be presented. Then in section 5.4.3 a more complete method will be presented.

7.4.1 Onset detection in frequency domain

In the spectral domain, energy increases linked to transients tend to appear as a broadband event. Since the energy of the signal is usually concentrated at low frequencies, changes due to transients are more noticeable at high frequencies. This attack transient noise is particularly noticeable at high frequency locations, since at low frequencies, high concentrations of energy (in the bins corresponding to the first few harmonics of the played note) mask this effect. The High Frequency Content (HFC) function, is defined, for the j th frame, by:

$$D_H[j] = \sum_k k |X_j[k]|$$

where $|X_j(\cdot)|$ is the spectral magnitude of the j th frame. Aim of this function is to emphasize the high frequency content of the sound and it works well for identifying percussive sounds. If compared with energy, this HFC function has greater amplitude during the transient/attack time. The HFC function produces sharp peaks during attack transients and is notably successful when faced with percussive onsets, where transients are well modeled as bursts of white noise.



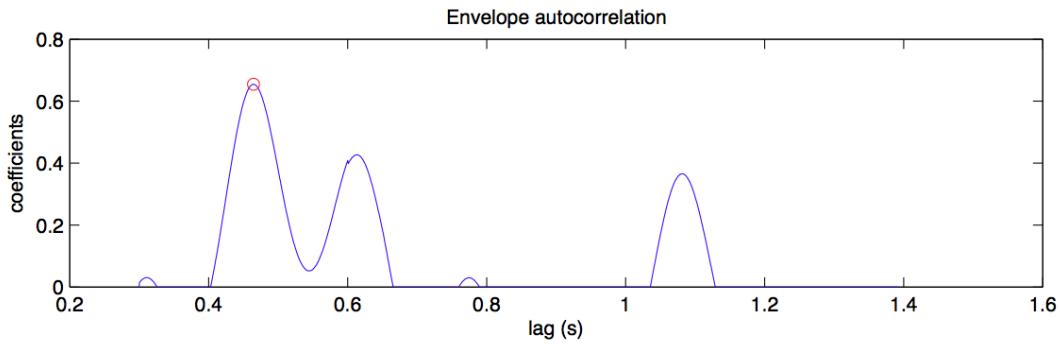


Figure 7.20: Autocorrelation of the onset detection curve.

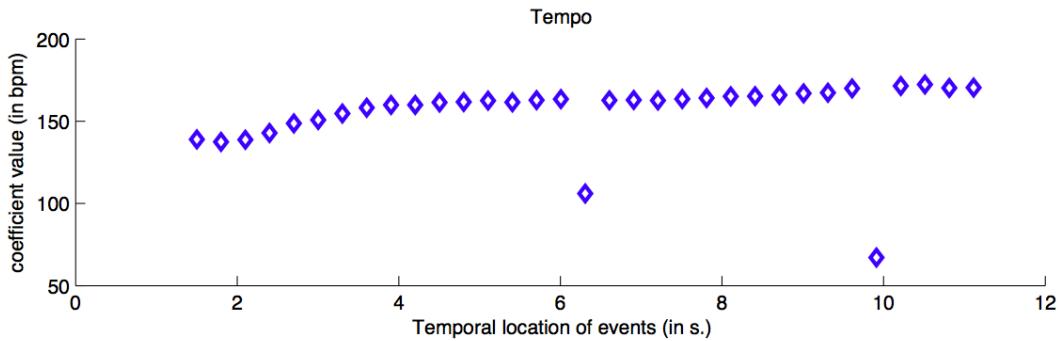


Figure 7.21: Tempo estimation of a framed audio signal.

7.4.2 Onset detection from Local Energy

In order to find onsets, a detection function is computed, i.e. an intermediate signal that reflects, in a simplified form, the local structure of the sound and manifests the occurrence of transients in the original signal.

Despite the number of variants, practically all time domain methods are based on the calculation of a first order difference function of the signal amplitude envelopes and taking the maximum rising slope as an onset or an onset component. The envelope of the signal is computed as explained in sect. 5.1.2.1.

A common approach is to use as detection function $D(t)$ the time derivative of the energy

$$D(t) = \frac{dE(t)}{dt}$$

(or rather the first difference for discrete-time signals) so that sudden rises in energy are transformed into narrow peaks in the derivative. An example is the algorithm based on the surfboard method of Schloss [1985], which involves smoothing the signal to produce an amplitude envelope and finding peaks in its slope using linear regression.

In Fig. 5.22 the effect of a simple onset detector based on Local energy is shown. In Fig. 5.22(a) the time-domain audio signal; in Fig. 5.22(b) its smoothed amplitude envelope drawn in bold over it, computed by a 40ms windowed RMS smoothing with 75% overlap and in Fig. 5.22(c) peaks in slope shown by dotted lines tangential to the envelope. This method is lossy, in that it fails to detect the onsets



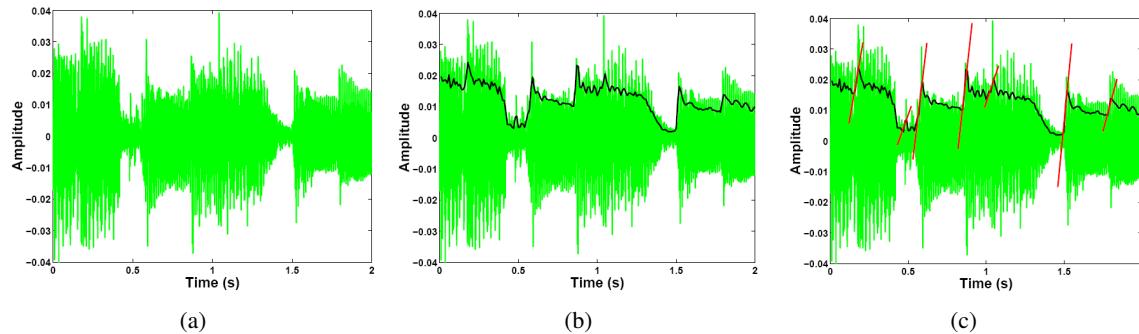


Figure 7.22: Example of onset detector based on local energy: time-domain audio signal (a), 40ms windowed RMS smoothing with 75% overlap (b), peaks in slope of envelope (c).

of many notes which are masked by simultaneously sounding notes. Occasional false onsets are detected, such as those caused by amplitude modulation in the signal.

When we take into account perceptual aspects, we may notice that psychoacoustics indicates that loudness is perceived logarithmically. This means that changes in loudness are judged relative to the overall loudness, since, for a continuous time signal,

$$D_r(t) = \frac{d(\log E(t))}{dt} = \frac{1}{E(t)} \frac{dE(t)}{dt}$$

Hence, computing the first-difference of $\log(E[n])$ roughly simulates the ears perception of loudness. The relative difference function $D_r(t)$ as detection function effectively solves the problems of low sound, where the amplitude grows slowly, by detecting the onset times earlier and, more importantly, by handling complicated onset tracks, since oscillations in the onset track of a sound do not matter in relative terms after its amplitude has started rising.

7.4.3 Combining pitch and local energy information

In this section a case study on how to combine pitch and energy information for onset detection is presented. The analysis of pitch variation can help in finding tone onsets. Figure 5.23 shows the progression of pitch along five seconds of a signal (Handel's Flute Sonata in E minor (Adagio) - “hard” performed): The red and the green circles show the considered spectral variation zones for the onsets detection. The red circles indicate the zones of effective occurrence of onsets, while that the circles the green indicate the zones where onsets had not occurred.

As we will see in the following section, this type of analysis can be complemented with an analysis of the signal envelope (that it equivalent to a study of the variation of the energy). It can be said that main objectives for that are:

- a) The elimination of false detections (circumscribed zones from the green circles), when the spectral disturbances are not follow for a minimum variation of energy (given by a minimum considered limit the note attack)
- b) Add great variations of energy, still that are not verified significant spectral disturbances. This election is also made with base in a threshold of variation of energy in the note attack.
- c) If there is a set of possible onsets in an interval of maximum distance, it is only considered the onset that corresponds to the lowest energy. All the others possibilities are discard.



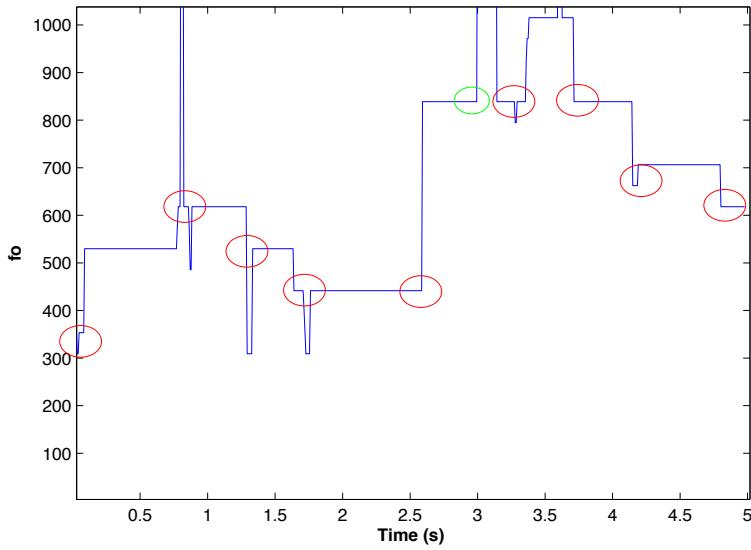


Figure 7.23: *F0 temporal variation (with a window and one step with the values above considered) calculated by standard spectrum for the 5 first seconds of a considered signal. In this type of study it is verified the variation of the spectral shape in respect to the first maximum of the FFT.*

Analysis of the temporal envelope In fig. 5.24 we see the temporal variation of the RMS envelope along the first 5 seconds of the considered signal. Instead of using a constant threshold that detected the brusque variations of the derivative of the *RMS* signal, we arrive at the conclusion that one adaptative threshold that follow the signal according to a more global dynamic, through the calculation of the average in a certain neighborhood of each point of the *RMS*, results much more efficient. The average was calculated between the 200 samples that surround each instant (1 second, for a step=0.005 s).The process consists in searching the minimum of *RMS* between two consecutive values detected by the threshold, since these values define a valley between two peaks.

Hybrid approach RMS-Frequency The analysis of the behavior of pitch and *RMS* envelope can be combined in the following steps.

- 1) We calculate all the onsets that result of the envelope analysis
- 2) In the case of the occurrence of onsets detected too much closed in the time, inside a given limit, is considered only in this interval the onset that has a lesser value of *RMS*. This limit consists of 0.25 of the mean of the measured distances between all detected onsets
- 3) We have considered that would be false detections (false positives) onsets that don't have a note attack in the energy domain greater than a given limit. We have eliminated to the list of onsets, that we have until this moment, the set of onsets that are below of this threshold. This threshold is calculated in relation to the average of the attacks of onsets considered until 2). Is considered the attack in the positive temporal direction and in the negative temporal direction. That is, the attack is defined as the energy jump that occurs between an onset and the next peak, and is defined also as the energy interval between an onset and the immediately previous peak. This threshold consists of 0.3 of the average of the considered attacks.
- 4) We calculate all onsets that result of spectral disturbances, that in our case can be seen as disturbances in the temporal progression of pitch.
- 5) As in 2), in the case of the occurrence of onsets too much closed in time, inside of a given

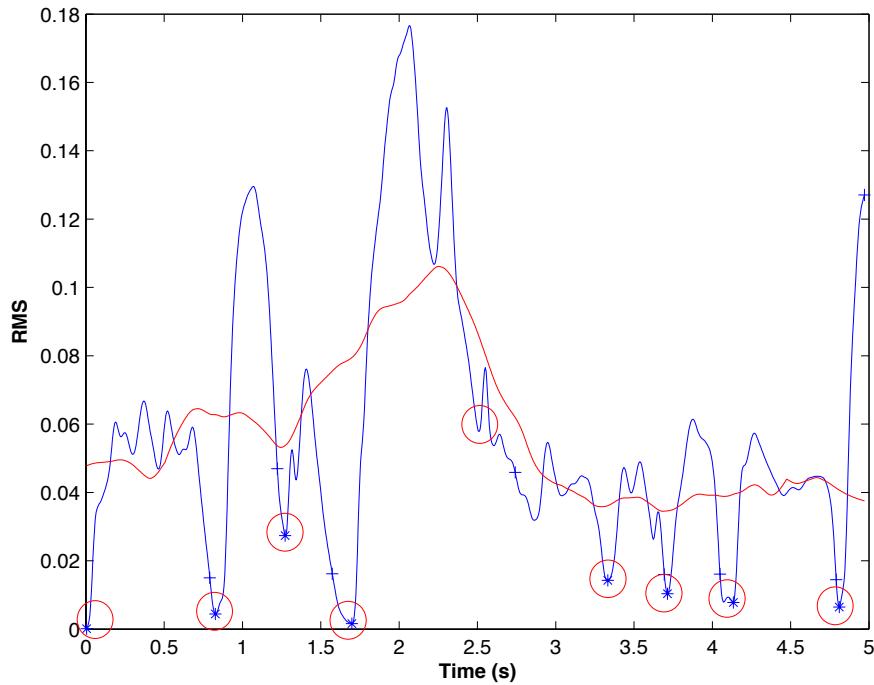


Figure 7.24: envelope of a 5 second window of signal. The blue line represents the RMS temporal variation, while that the red line represents the dynamic threshold that follows the RMS of the signal. The crosses on signal RMS represent the detected onsets. The circles represent the zones of effective onsets.

threshold, is considered only the onset that has a lesser value of *RMS*. This threshold is the same that was considered in 2).

6) Are considered here as valid onsets the onsets considered until 3) (valid onsets extracted by *RMS*) that are inside a given threshold of proximity in relation to the more closed onset in the set of onsets considered in 5) (onsets valid extracted by disturbances of pitch). This threshold of proximity corresponds to a maximum absolute distance of 0.1 seconds.

7) Of the onsets calculated by *RMS* analysis considered valid - before the onsets elimination according to the previous criterion (i.e. the set considered in 3)) we add to the list the onsets that had a note attack superior than a given threshold in the energy domain. The attack is here defined as the difference between the *RMS* value for the instant of the onset and the value of the next peak (that can be found between this onsets and the consecutive one). This limit is calculated relatively to the mean of all the considered attacks of onsets considered until 6). Numerically corresponds to 0,5 of the mean of these attacks.

7.5 Feature Selection

In many applications, reducing the dimensionality of the data by selecting a subset of the original variables may be advantageous for reasons including the expense of making, storing and processing measurements. The art of machine learning starts with the design of appropriate data representations. Better performance is often achieved using features derived from the original input. The reasons for reducing

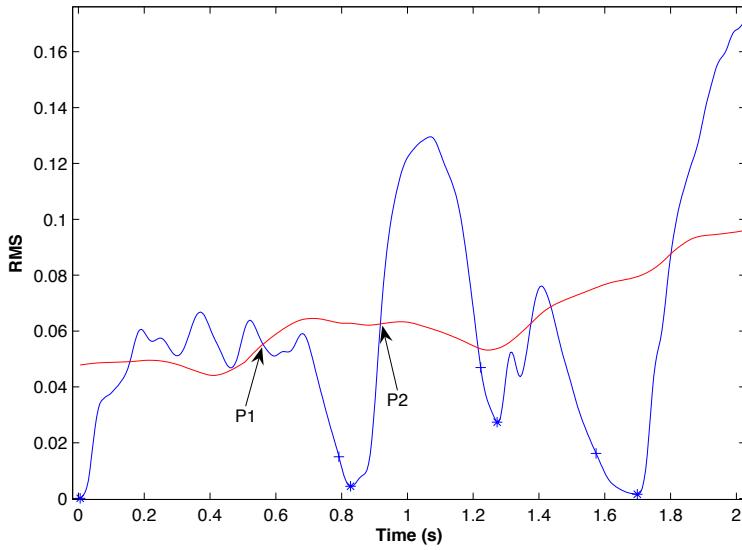


Figure 7.25: The points P_1 and P_2 are two values of the RMS signal detected by threshold. These values define a valley between two peaks. The value for the onset is the minimum that can be find between these two points.

the number of features to a sufficient minimum are that: (i) less features implies simpler models and then less training data needed; (ii) the higher the ratio of the number of training patterns N to the number of free classifier parameters, the better the generalisation properties of the resulting classifier; (iii) computational complexity, correlating (redundant) features.

Good features result in large between-class distance and small within-class variance; to perform this, the approaches divide according to: (i) examine features individually and discard those with little discrimination ability; (ii) to examine the features in combinations; (iii) linear (or nonlinear) transformation of the feature vector. Conventional statistical approaches such as comparing mean correlations within and between subgroups, principal components analysis (PCA), analysis of variance (ANOVA), and visualization techniques can be appropriately applied to determine which features to select. These techniques allow us to discover which data is redundant, which features have a relatively high variation between the subjects, and how the original feature set can be reduced without a big loss in the variance explained and recognition rate.

Variable subset selection procedure consists on measuring the relevance of a subset of input variables, and an optimization algorithm for searching for the optimal or a near-optimal subset with respect to the subset of variables. Procedures for variable subset selection can be classified into two groups: filter procedures and wrapper procedures. In case of *filter* procedures, the relevance measure is defined independently from the learning algorithm. The subset selection procedure in this case can be seen as a preprocessing step. In case of *wrapper* procedures, the relevance measure is directly defined from the learning algorithm, for example, in terms of the cost of the learning and the precision achieved by classification algorithm. On the other side, wrapper procedures need the number of possible parameters to be as low as possible, so that the algorithm should be highly computationally efficient.

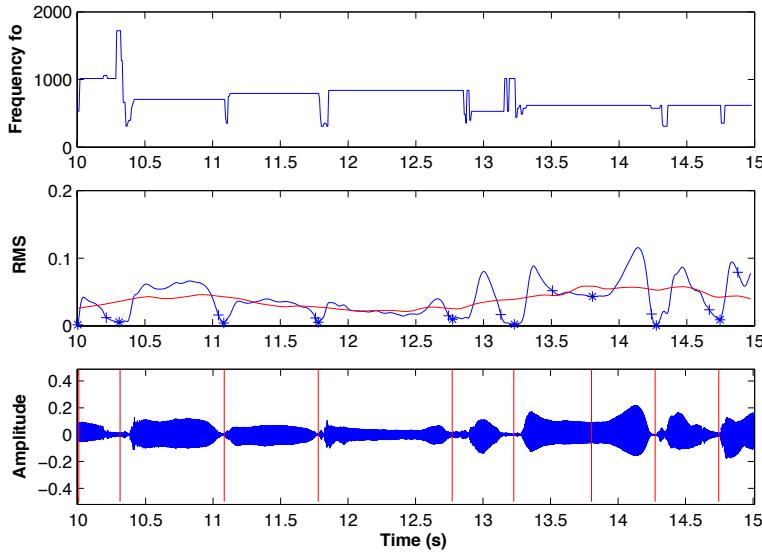


Figure 7.26: Detection of onsets for a five seconds window: the last figure represents the onset detection, where red lines indicate the instants for each detected onset.

7.5.1 One-way ANOVA

The analysis of variance (ANOVA) helps to identify the features which highlight differences between groups (subjects).

Let a database contain records of g subjects (number of groups) and n_l sessions for each subject. Let X_{lj} , where $l = 1, 2, \dots, g$ and $j = 1, 2, \dots, n_l$, be a random sample of size n_l from a population with mean μ_l , $l = 1, 2, \dots, g$. Anova is used to investigate whether the population mean vectors are the same, i.e. the null hypothesis of equality of means could be formulated as $H_0 : \mu_1 = \mu_2 = \dots = \mu_g$, and if not, which implies components differ significantly. The F -test rejects H_0 at level α if:

$$F = \frac{\frac{SSA/(g-1)}{g}}{\frac{SSW/(\sum_{l=1}^g n_l - g)}{}} > F_{g-1, \sum n_l - g}(\alpha) \quad (7.26)$$

where $F_{g-1, \sum n_l - g}(\alpha)$ is the upper (100α) th percentile of the F -distribution with $g-1$ and $\sum n_l - g$ degrees of freedom; $SSA = \sum_{l=1}^g n_l (\bar{x}_l - \bar{x})^2$ is the sum of squares among groups, where \bar{x}_l and \bar{x} are estimates of group and overall sample means respectively; $SSW = \sum_{l=1}^g \sum_{j=1}^{n_l} (x_{lj} - \bar{x}_l)^2$ is the total within-group sum of squares, where x_{lj} is an observation of the feature from subject l , session j .

7.5.2 Principal Component Analysis (PCA)

Principal component analysis (PCA) is concerned with explaining the variance-covariance structure of a set of variables through a few linear combinations of these variables. This means that the original feature space is transformed by applying e.g. a linear transformation via a PCA. PCA is a linear transformation that transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on. PCA can be used for dimensionality reduction in a dataset



while retaining those characteristics of the dataset that contribute most to its variance, by keeping lower-order principal components and ignoring higher-order ones. Such low-order components often contain the "most important" aspects of the data.

Given a set of n observations on p observed variables, the purpose of PCA is to determine r new variables, where r is small relative to p . The r new variables, called principal components, must together account for most of the variation in the p original variables. Principal component analysis operates by replacing the original data matrix \mathbf{X} by an estimate composed of the product of two matrices. In matrix notation the approximation $\widehat{\mathbf{X}}$ of the matrix \mathbf{X} is given by $\widehat{\mathbf{X}} = \mathbf{Z}\mathbf{V}'$, where \mathbf{Z} is the $(n \times r)$ matrix of observations on the first r principal components, and $\mathbf{V}(p \times r)$ is the matrix whose columns are the first r eigenvectors of $\mathbf{X}'\mathbf{X}$. The matrix $\mathbf{Z}'\mathbf{Z} = \Lambda$, where Λ is the diagonal matrix of r eigenvalues $\lambda_k, k = 1, 2, \dots, r$. The sum of squares and cross products matrix for the principal components is therefore a diagonal matrix with diagonal elements λ_k , that decline in magnitude. The eigenvalues λ_k are given by:

$$\sum_{i=1}^n z_{ik}^2 = \lambda_k, k = 1, 2, \dots, r$$

The sum of squares for each principal component is therefore given by the corresponding eigenvalue. The quantity to be minimised is given by:

$$tr(\mathbf{X} - \widehat{\mathbf{X}})'(\mathbf{X} - \widehat{\mathbf{X}}) = tr\mathbf{X}'\mathbf{X} - \sum_{k=1}^r \lambda_k$$

and hence if $r = p$, then this expression has the value zero. Each of the eigenvectors generates a portion of the total variation (or sum of squares) in \mathbf{X} as measured by $tr(\mathbf{X}'\mathbf{X})$. The contribution to $\sum_{k=1}^p \lambda_k$ provided by $\mathbf{z}_l = \mathbf{Z}\mathbf{v}_l'$ is $\mathbf{z}_l'\mathbf{z}_l = \lambda_l$. The proportion of the total variance measured by $tr(\mathbf{X}'\mathbf{X})$, accounted for by the component \mathbf{z}_l is given by $\lambda_l / \sum_{k=1}^p \lambda_k$. The number of components actually used for the approximation of \mathbf{X} can be guided by the measure $\sum_{k=1}^l \lambda_k / \sum_{k=1}^p \lambda_k$, where $l \leq p$.

In other words, the first principal component is the axis passing through the centroid of the feature vectors that has the maximum variance therefore explains a large part of the underlying feature structure. The next principal component tries to maximize the variance not explained by the first. In this manner, consecutive orthogonal components are extracted. The principal components depend solely on the covariance or correlation matrix of the data.

7.5.3 Further feature subset selection

To find the optimal subset of features, we should form all possible combinations of M features out of the D originally available; the best combination is then selected according to any desired class separability measure J . In practice, it is not possible to evaluate all the possible feature combinations. Thus the search is for a satisfactory set of features instead of an optimal set and greedy approaches are used.

Sequential backward selection (SBS) consists of choosing a class separability criterion J , and calculate its value for the feature vector which consists of all available features (length = D). Eliminate one feature, and for each possible resulting combinations (of length $D-1$) compute J . Select the best, and continue this for the remaining features, and stop when you have obtained the desired dimension M . This is a suboptimal search procedure, since nobody can guarantee that the optimal $r-1$ dimensional vector has to originate from the optimal r -dimensional one. This method is good for discarding a few worst features.

Sequential forward selection (SFS) consists of computing the criterion J for all individual features and select the best. Form all possible two-dimensional vectors that contain the winner from the previous



step, calculate the criterion for each vector and select the best; continue adding features one at time, taking always the one that results in the largest value of the criterion J , and stop when the desired vector dimension M is reached. This method is particularly suitable for finding a few good features.

Both SBS and SFS suffer from the nesting effect: once a feature is discarded in SBS (selected in SFS), it cannot be reconsidered again (discarded in SFS).

7.6 Music Information Retrieval: Issues, Problems, and Methodologies

7.6.1 Introduction

The core problem of Information Retrieval (IR) is to effectively retrieve documents which convey content being relevant to the user's information needs. Effective and efficient techniques have been developed to index, search, and retrieve documents from collections of hundreds of thousands, or millions of textual items.

The most consolidated results have been obtained for collection of documents and user's queries written in textual form and in English language. Statistical and probabilistic techniques have lead to the most effective results for basic system functions and are currently employed to provide advanced information access functions as well. The content description of media being different from text, and the development of different search functions are necessary steps for content-based access to Digital Libraries (DL). This statement mainly applies to cultural heritage domain, where different media and search functions live together.

In order to provide a content-based multimedia access, the development of new techniques for indexing, searching, and retrieving multimedia documents have recently been the focus of many researchers in IR. The research projects in DLs, and specifically those carried out in cultural heritage domain, have shown that the integrated management of diverse media - text, audio, image, video - is necessary.

The problem with content-based access to multimedia data is twofold.

- On the one hand, each media requires specific techniques that cannot be directly employed for other media.
- On the other hand, these specific techniques should be integrated whenever different media are present in a individual item.

The core IR techniques based on statistics and probability theory may be more generally employed outside the textual case and within specific non-textual application domains. This is because the underlying models, such as the vector-space and the probabilistic models, are likely to describe fundamental characteristics being shared by different media, languages, and application domains.

7.6.1.1 Digital Music and Digital Libraries

There is an increasing interest towards music stored in digital format, which is witnessed by the widespread diffusion on the Web of standards for audio like MP3. There are a number of reasons to explain such a diffusion of digital music.

- First of all, music is an art form that can be shared by people with different culture because it crosses the barriers of national languages and cultural backgrounds. For example, tonal Western music has passionate followers also in Japan and many persons in Europe are keen on classical Indian music: all of them can enjoy music without the need of a translation, which is normally required for accessing foreign textual works.



- Another reason is that technology for music recording, digitalization, and playback, allows for an access that is almost comparable to the listening of a live performance, at least at the level of audio quality, and the signal to noise ratio is better for digital formats than for many analog formats. This is not the case of other art forms, like painting, sculpture or even photography, for which the digital format is only an approximate representation of the artwork. The access to digitized paintings can be useful for studying the works of a given artist, but cannot substitute the direct interaction with the real world works.
- Moreover, music is an art form that can be both cultivated and popular, and sometimes it is impossible to draw a line between the two, as for jazz or for most of ethnic music.

These reasons, among others, may explain the increasing number of projects involving the creation of music DLs. A music DL allows for, and benefits from, the access by users from all over the world, it helps the preservation of cultural heritage, and it is not tailored only to scholars' or researchers' needs. More in general, as music is one of the most important means of expression, the organization, the integration with other media, and the access to the digitized version of music documents becomes an important multimedia DL component. Yet, music has some peculiarities that have to be taken into account when developing a music DL. In figure 5.27 the architecture of a music information retrieval system is shown.

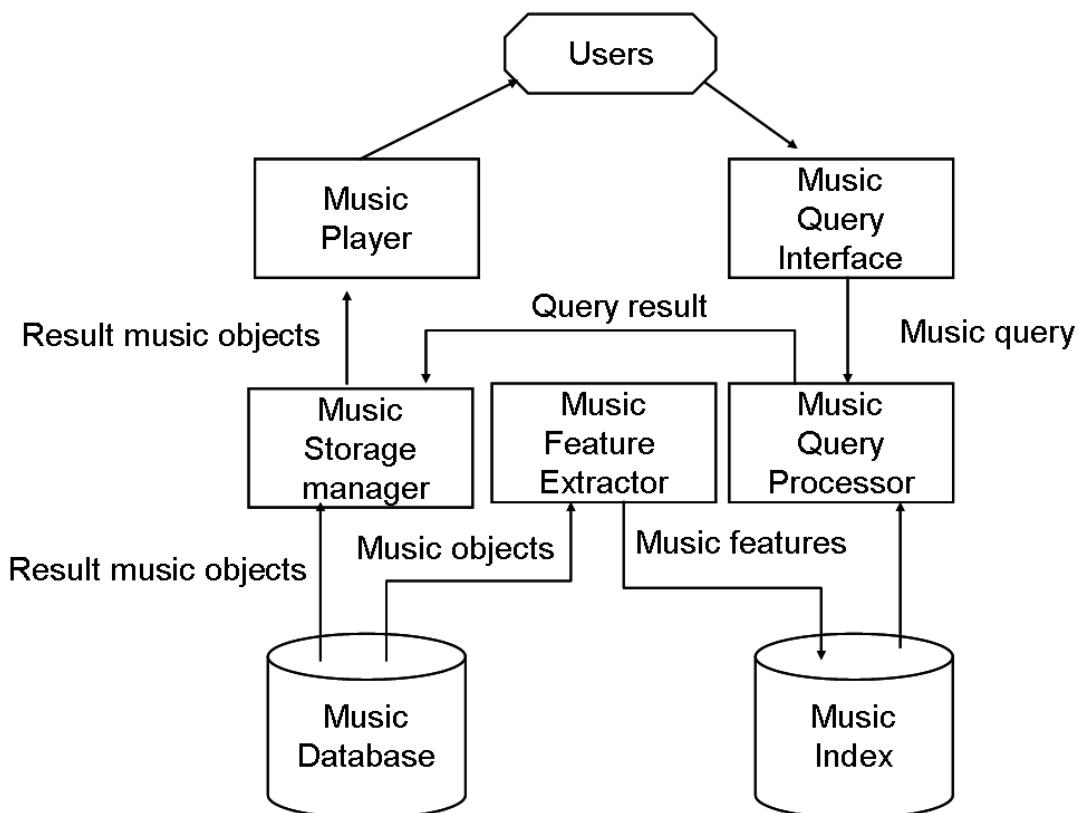


Figure 7.27: Architecture of a music information retrieval system

7.6.1.2 Music Information Retrieval

Specific and effective techniques being capable of indexing and retrieving such multimedia documents as the music ones need to be designed and implemented.

Current approaches to Music Information Retrieval (MIR) are based either on string matching algorithms or textual bibliographic catalogue.

- String matching approach makes content-based retrieval very difficult - indeed, retrieving textual files using Unix grep-like commands gives poor results.
- Textual bibliographic catalogue approach makes content-based retrieval impossible since the music content cannot be described by bibliographic catalogue.

The requirement for a content-based MIR has been stressed within the research area of music information systems as well. The developments in the representation of music suggest a need for an information retrieval philosophy directed toward non-text searching and eventual expansion to a system that encompasses the full range of information found in multimedia documents. As IR has dealt with the representation and the disclosure of content from its early days, it is natural to think that IR techniques should be investigated to evaluate their application to music retrieval. According to McLane “what has been left out of this discussion, and will no doubt be a topic for future study, is the potential for applying some of the standard principles of text information retrieval to music representations”.

- If we follow the hypothesis that the use of standard principles of text information retrieval to index and retrieve music documents is possible, then the design of ad-hoc segmentation algorithms to produce musical ‘lexical units’ like words in textual documents is required.

The concept of lexical unit may vary depending on the approach. A lexical unit can be: a fixed-length string, the incipit, a complete theme, a melodic phrase, and so on. Music is a continuous flow of events (e.g., notes, chords, and unpitched percussive sounds) without explicit separators, if not those perceived by listeners. Also music representation lacks of separators of lexical units, because it conveys information only about macro-events, like changes in tonality or the presence of repetitions. It is therefore necessary to automatically detect the perceived lexical units of a music document to be used like words in textual documents.

- Moreover, content-based MIR requires the design of normalization algorithms. Once detected, musical lexical units occur in documents with many variants like textual words do within textual documents. For example, a melodic pattern may occur in many music works, perhaps composed by different authors, with small deviations of note intervals or timing. Despite these deviations, different patterns may be perceptually similar, hence conveying the same music perception. It is therefore necessary to detect these variants and conflate all the similar musical lexical units into a common stem expressing the same music perception. This conflation process is analogous to the one performed in the textual case for detecting word stems through, for example, the Porter’s stemming algorithm.

To allow the integration of automatic music processing techniques with automatic IR techniques, segmentation and normalization algorithms are applied also on music queries.

In a content-based music IR system, users may be able to interact with the system by using the same language, that is the music language. This because content-based MIR requires users to be able of expressing the music document content. The most natural way of express music content is singing and playing music. This approach is often referred to as the query by example paradigm. Therefore, users should be provided with interfaces and search functions so that they can play music and send a music query to the system.



To make content-based music retrieval possible, query content and document content have to be matched: Describing query content is then necessary. If we regard music queries as music documents, segmentation and normalization can be performed also on music queries using the same algorithms used for disclosing document content.

7.6.2 Issues of Content-based Music Information Retrieval

Music, in its different representations, can be considered as another medium together with text, image, video, and speech. Nevertheless, there are some issues that make music different from other multimedia IR application domains. The issues we address are form, instantiation, dimension, content, perception, user profile, and formats. The most relevant issues are described in the following Sections.

7.6.2.1 Peculiarities of the Music Language

The same entity, i.e. a music work, can be represented in two different main forms: the notated and the acoustic form, respectively corresponding to score and performance. Hence the communication in music is performed at two levels:

- the composer translates his intentions in a music structure (music as a composing art),
- the musician translates the written score into sounds (music as a performing art).

Also users may have different needs, in particular the music scholar may look for a given composition, while the melomane may look for a particular performance.

Each music work may have different instantiations. As musicians can interpret scores, the resulting performances may differ and therefore more performances correspond to an individual score. Furthermore, the same music work may be transcribed into different scores, depending on the revisers' choices. As a consequence, different performances and scores may rely to the same music work.

Different dimensions characterize the information conveyed by music. Melody, harmony, rhythm, and structure are dimensions, carried by the written score, that may be all or in part of interest for the final user. In the case of a performance other dimensions should be added, for instance timbre, articulation, and timing. It is likely that the dimensions of interest vary with the level of user's expertise and the specific user's search task. As described in Section 5.6.2.3, different formats are able to capture only a reduced number of dimensions. Therefore, the choice of a representation format has a direct impact on the degree to which a music retrieval system can describe each dimension.

While text, image, video, or speech-based documents in general convey some information that form their content, it is still unclear what type of content, if any, music works do convey. Let us consider an example: the concept of tempest can be described with a textual document, such as the first chapter of Shakespeare's 'The Tempest', a painting, such as the landscape of Giorgione's 'The Tempest', a video or speech, such as broadcasting news about, for instance, a tornado. All these media are able to convey, among all the other information, the concept of tempest. There are up to forty music works of tonal Western music whose title is related to tempests, among those the most famous probably are Beethoven's Sixth Symphony IV Movement, Rossini's Overture of 'William Tell', and Vivaldi's Concerto 'La Tempesta di Mare'. These works differ in music style, form, key and time signature, and above all the user may be not able to recognize that the work is about a tempest and not just pure music.

In principle, music language does not convey information as, for instance, text or video do. Many composers wrote music to stir up emotions, and in general they aimed to communicate no specific information to the listener. The final user feels emotions on listening to the music, and he interprets some information independently from the composer's and performer's thought and differently from the other



users. There is a particular kind of music works, called *musica a programma*, in which the title (like Vivaldi's 'The Spring') or a lyric (like Debussy's 'Prélude l'aprÈs-midi d'un faune') suggests a meaning to the listener; this sort of textual data would be better managed using a database system rather than a IR system. Moreover in sung music, such as Cantatas, the accompanied text gives the work some meaning, yet that sort of text would require ad-hoc IR techniques to be effectively managed. In general the availability of textual material together with music documents is insufficient.

It is then important to consider how music is perceived and processed by listeners, to highlight which kind of content is carried by this medium. A number of different theories was proposed by musicologists, among which the most popular ones are the Generative Theory of Tonal Music (see Sect. 7.6.1) and the Implication-Realization Model (see Sect. 7.6.2). In both cases it is stated that listeners perceive music as structured and consisting of different basic elements. Therefore, even if music notation and performance lack of explicit separators (like blanks or commas in text) musicians and listeners perceive the presence of small elements which constitute the music work: we can consider these elements as the lexical units for a content-based approach to MIR. It is likely that all the dimensions of music language can be segmented in their lexical units and be used to extract a content from a music document.

7.6.2.2 The Role of the User

As always happens in IR, the effectiveness of techniques does strongly depend on the final user. DL systems does indeed interact with final users of very diverse types and with different levels of expertise in the use of the system itself. This is particularly true for music DLs, because there is a great difference in users' expertise depending on the practice of a musical instrument, the ability of reading a score, the knowledge of harmony rules, the familiarity with composition styles, and so on. Users may have different needs, for instance a music scholar may look on how a given cadenza is used by different authors, while a melomane may look for a particular performance of a well-known musician. This is a key aspect in the design of a methodology for content-based MIR, because it affects the choice of the dimension to be used for describing a music work, that is which kind of content has to be extracted from it.

Considering that access to DL is widely spread to users of any type, final users of a music DL may not have a deep knowledge of music language. Therefore, melody seems to be the most suitable dimension. In fact, almost everybody can recognize simple melodies and perform them at least by singing or humming. In this case, lexical units can be considered the musical phrases, which may be defined as short excerpts of the melody which constitute a single musical gesture. Moreover, melody carries also explicit information about rhythm and implicit information about harmony.

Melody can be the most suitable evidence for content-based music retrieval, it may however be the case that only a part of the melody can effectively be exploited as useful evidence for music document and query description. This implies that, if phrases can be detected by means of some segmentation algorithms, then it is likely that some of these phrases are 'good' descriptors of the music content from users' point of view, while others can be dropped since they give little contribution to the music content description and may negatively affect efficiency. This latter consideration leads us to contemplating the possibility of building lists of stop phrases, that may be dropped from the index of phrases similarly to the textual case. However, it is still unclear if stop phrases exist how users perceive them. While one can identify a word as stop word because it has no, little, or less meaning than keywords, one cannot identify a phrase as stop phrase because it is very difficult to say what 'phrase meaning' does mean, and frequency-based stop phrase list construction may be a difficult task because, for instance, users may recall melody excerpts just because they are very frequent in a musical genre.



7.6.2.3 Formats of Music Documents

As previously mentioned, the communication in music is achieved at two levels, corresponding to two forms: the composer translates his intentions into a musical structure, that is represented by a music score, and the musician translates the written score into a performance, that is represented by a flow of acoustic events. A number of different digital formats correspond to each form. It can be noted that, as musicians can interpret scores, the resulting performances differ and therefore more than one performance correspond to a single score. Even if the two forms can be considered as instantiations of the same object, they substantially differ in the information that can be manually or automatically extracted from their respective formats.

The first problem which arises in the automatic processing of music is then that a music work may be digitally stored in different formats. The same music piece can be represented, for example,

- by a reproduction of the manuscript,
- by a symbolic notation of the score,
- by a sequence of time-stamped events corresponding to pitched and unpitched sounds,
- or by a digital recording of an acoustic performance.

Each format carries different information on the content of the document. For instance, at the state-of-the-art it is impossible to recover informations about the written score from the digital sampling, e.g. stored in a compact disk, of a polyphonic audio signal, and the score carries no information about the timbre, expressive timing and other performing parameters. Hence, the documents format has to be chosen depending on the aims of the DL, which may encompass preservation, displaying, listening, indexing, and retrieval, and so on. As an example, preservation requires high quality audio coding and dissemination over the Internet requires lossy compression.

Formats for digital music documents can be divided in two classes.

- The score is a structured organization of symbols, which correspond to acoustic events; the score is a direct representation of all the dimensions of music (i.e., melody, harmony, and rhythm) and it usually contains all the information that is relevant for classifying and cataloguing: type of movement, time and key signatures, composer's notes, and so on. The symbolic nature of the score allows for an easy representation of its content, and many proposed formats represents score in the form of a textual markup language, for instance ABC and GUIDO.
- The performance is made of a sequence of gestures performed by musicians on their musical instruments; the result is a continuous flow of acoustic waves, which correspond to the vibration induced on musical instruments. Even if all the dimensions of music are embedded in a performance, it requires high-level information processing to recognize them. In particular, only experienced musicians can recognize all the dimensions of music from listening to a performance and, at the state of the art, there is no automatic system that can recognize them from an acoustic recording, apart from trivial cases. The nature of a performance does not allow for an easy representation of its content. The formats adopted to digitally represent performances, such as AIFF (Audio Interchange File Format, proposed by Apple Computers) or MP3 (MPEG1, Layer3), are a plain digital coding of the acoustic sound waves, with a possible data compression.

We present now an example of different representations of a melody with reference to fig. 5.28(a). we can represent as absolute or relative values.

- Absolute measure:





Figure 7.28: Example of a melody

- Absolute pitch: C5 C5 D5 A5 G5 G5 G5 F5 G5
- Absolute duration: 1 1 1 1 1 0.5 0.5 1 1
- Absolute pitch and duration:
(C5, 1) (C5, 1) (D5, 1) (A5, 1) (G5, 1) (G5, 0.5) (G5, 0.5) (F5, 1) (G5, 1)
- Relative measure:
 - Contour (in semitones): 0 +2 +7 -2 0 0 -2 +2
 - IOI (Inter onset interval) ratio: 1 1 1 1 0.5 1 2 1
 - Contour and IOI ratio:
(0, 1) (+2, 1) (+7, 1) (-2, 1) (0, 0.5) (0, 1) (-2, 2) (+2, 1)

In a polyphonic case (see fig. 5.28(b)) we can represent in different ways.

- Keep all information of absolute pitch and duration (start_time, pitch, duration)
(1, C5, 1) (2, C5, 1) (3, D5, 1) (3, A5, 1) (4, F5, 4) (5, C6, 1) (6, G5, 0.5) (6.5, G5, 0.5) ...
- Relative note representation: Record difference of start times and contour (ignore duration) (1, 0) (1, +2) (0, +7)
...
- Monophonic reduction, e.g. select one note at every time step (main melody selection)
(C5, 1) (C5, 1) (A5, 1) (F5, 1) (C6, 1) ...
- Homophonic reduction (chord reduction), e.g. select every note at every time step
(C5) (C5) (D5, A5) (F5) (C6) (G5) (G5) ...

With the aim of taking into account all the variety in which music information can be represented, it has been proposed the Standard Music Description Language (SMDL), as an application of the Standard ISO/IEC Hyper-media/Time-based Structuring Language. In SMDL, a music work is divided into different domains, each one dealing with different aspects, from visual to gestural, and analytical. SMDL provides a linking mechanism to external, pre-existing formats for visual representation or storage of performances. Hence SMDL may be a useful way for music representation standardization, but the solution is just to collect different formats rather than proposing a new one able to deal with all the aspects of the communication in music.

A Note on MIDI A format that can be considered as a compromise between the score and the performance forms is MIDI (Musical Instrument Digital Interface), which was proposed in 1982 for data exchange among digital instruments. MIDI carries both information about musical events, from which it is possible to reconstruct an approximate representation of the score, and information for driving a synthesizer, from which it is possible to listen to a simplified automatic performance. It seems then that MIDI draws a link between the two different forms for music representation. This characteristics,

together with the fortune of MIDI as an exchange format in the early times of the Internet, can explain why many music DLs and most projects regarding music indexing and retrieval refer to it. Some of the research work on music information retrieval take advantage of the availability of MIDI files of about all the different music genres and styles. MIDI files are parsed in order to extract a representation of the music score, and then indexed after different preprocessing.

Nevertheless, MIDI is becoming obsolete and users on the Internet increasingly prefer to exchange digital music stored in other formats such as MP3 or RealAudio, because they allow for a good audio-quality with a considerably small dimension of the documents size. Moreover, if the goal of a music DL is to preserve the cultural heritage, more complete formats for storing both scores and performances are required. Being a compromise between two different needs – i.e., to represent symbols and to be playable – MIDI turns out to fit neither the needs of users who want to access to a complete digital representation of the score, nor to users who want to listen to high-quality audio performances.

7.6.2.4 Dissemination of Music Documents

The effectiveness of a retrieval session depends also on the ability of users to judge whether retrieved documents are relevant to their information needs. The evaluation step, in a classical presentation-evaluation cycle, for an information retrieval session of textual documents usually benefits from tools for browsing the document (e.g., the ‘find’ function), in particular when the size of documents is large. Moreover, a general overview of the textual content may help users to judge the relevance of most of the retrieved documents.

Users of a music DL cannot take advantage of these shortcuts for the evaluation of documents relevance, when they are retrieving music performances. This is due to the central role played by time in the listening to music. A music performance is characterized by the organization of music events along the time axis, which concatenates the single sounds that form the whole performance. Changing playback speed of more than a small amount may result in a unrecognizable performance. In other words, it requires about 20 minutes to listen to a performance that lasts 20 minutes. It may be argued that many music works are characterized by their incipit, that is by their first notes, and hence a user could be required to listen only to the first seconds of a performance before judging its relevance to his information needs. Anyway, the relevant passage of a music document – e.g., a theme, the refrain – may be at any position in the time axis of the performance.

A tool that is often offered by playback devices is the ‘skip’ function, that allows for a fast access to a sequence of random excerpts of the audio files, to help listeners looking for given passages. Everyone who tried to find a particular passage in a long music performance, knows that the aid that the skip function gives when accessing to music documents is not even comparable with the find function for textual documents. This is partially due to the fact that auditory information does not allow a snapshot view of the documents as visual information does. The evaluation of relevance of retrieved music documents may then be highly time-consuming, if tools for a faster access to document content are not provided.

7.6.3 Approaches to Music Information Retrieval

There is a variety of approaches to MIR and there are many related disciplines involved. Because of such wide varieties, it is difficult to cite all the relevant work. Current approaches to MIR can broadly be classified into data-based and content-based approaches. For the aims of scientific research on multimedia IR, content-based approaches are more interesting, nevertheless the use of auxiliary textual data structures, or metadata, can frequently be observed in approaches to non-textual, e.g. image or video document indexing. Indeed, textual index terms are often manually assigned to multimedia documents to allow users retrieving documents through textual descriptions.



7.6.3.1 Data-based Music Information Retrieval

Data-based MIR systems allow users for searching databases by specifying exact values for predefined fields, such as composer name, title, date of publication, type of work, etc., in which cases we actually speak about exact match retrieval. Data-based approaches to MIR makes content-based retrieval almost impossible since the music content cannot easily be conveyed simply by bibliographic catalogue only.

Indeed, music works are usually described with generic terms like ‘Sonata’ or ‘Concerto’ which are related only to the music form and not the actual content. From an IR point of view, data-based approaches are quite effective if the user can exhaustively and precisely use the available search fields. However, bibliographic values are not always able to describe exhaustively and precisely the content of music works. For example, the term ‘Sonata’ as value of the type of work cannot sufficiently discriminate all the existing sonatas.

Moreover, many known work titles, such as the Tchaikovskij’s ‘Pathetic’, are insufficient to express a final user’s query whenever he would find the title not being a good description of the music work. The use of cataloging number, like K525 for Mozart’s ‘Eine Kleine Nachtmusik’, will be effective only if the user has a complete information on the music work, and in this case a database system will suffice.

Searching by composer name can be very effective. However, some less known composers and their works may not be retrieved if only because the authors are little known. Content-based MIR may allow for the retrieval of these pieces since querying by a known melodic pattern, such as a Mozart’s one, may retrieve previously not considered or unknown composers. On the other hand, for a prolific composer, just like Mozart, a simple query by composer’s name will retrieve an extremely high number of documents, unbearable for the final user.

7.6.3.2 Content-based Music Information Retrieval

Content-based approaches take into account the music document content, such as notation or performance, and automatically extract some features, such as incipites or other melody fragments, timing or rhythm, instrumentation, to be used as content descriptors. Typical content-based approaches are based on the extraction of note strings from the full-score music document. If arbitrarily extracted, note strings may be meaningless from a musical point of view because no music information is exploited to detect those strings, yet allows for a good coverage of all the possible features to be extracted.

Content-based approaches to MIR can sometimes be oriented to disclosing music document semantic content using some music information, under the hypothesis that music documents can convey some meaning and then some fragments can effectively convey such meaning. In the latter case, some music information is exploited to detect those strings so that the detected strings can musically make sense if, for instance, they were played.

The research work on this area of MIR can be roughly divided in two categories:

- on-line searching techniques, which compute a match between a representation of the query and a representation of the documents each time a new query is submitted to the system;
- indexing techniques, which extract off-line from music documents all the relevant information that is needed at retrieval time and perform the match between query and documents indexes.

Both approaches have positive and negative aspects.

- From the one hand, on-line search allows for a direct modelling of query errors by using, for instance, approximate pattern matching techniques that deal with possible sources of mismatch, e.g. insertion and/or deletion of notes. This high flexibility is balanced by high computational costs, because the complexity is at least proportional to the size of the document collection (and, depending on the technique, to the documents length).

- From the other hand, indexing techniques are more scalable to the document collection, because the index file can be efficiently accessed through hashing and the computational complexity depends only on query length. The high scalability is balanced by a more difficult extraction of document content, with non trivial problems arising in case of query errors that may cause a complete mismatch between query and document indexes.

Both approaches had given interesting and promising results. Yet, indexing approaches need to be investigated in more detail because of the intrinsic higher computational efficiency.

Previous work on on-line search has been carried out following different strategies. A first approach is based on the use of pattern discovery techniques, taken from computational biology, to compute occurrences of a simplified description of the pitch contour of the query inside the collection of documents. Another approach applies pattern matching techniques to documents and queries in GUIDO format, exploiting the advantages of this notation in structuring information. Approximate string matching has been used. Markov chains have been proposed to model a set of themes that has been extracted from music documents, while an extension to hidden Markov models has been presented as a tool to model possible errors in sung queries.

An example of research work on off-line document indexing has been presented in[8]. In that work melodies were indexed through the use of N-grams, each N-gram being a sequence of N pitch intervals. Experimental results on a collection of folk songs were presented, testing the effects of system parameters such as N-gram length, showing good results in terms of retrieval effectiveness, though the approach seemed not to be robust to decreases in query length. Another approach to document indexing has been presented in[24], where indexing has been carried out by automatically highlighting music lexical units, or musical phrases. Differently than the previous approach, the length of indexes was not fixed but depended on the musical context. That is musical phrases were computed exploiting knowledge on music perception, in order to highlight only phrases that had a musical meaning. Phrases could undergo a number of different normalization, from the complete information of pitch intervals and duration to the simple melodic profile.

Most of the approaches are based on melody, while other music dimensions, such as harmony, timbre, or structure, are not taken into account. This choice may become a limitation depending on the way the user is allowed to interact with the system and on his personal knowledge on music language. For instance, if the query-by-example paradigm is used, the effectiveness of a system depends on the way a query is matched with documents: If the user may express his information need through a query-by-humming interface, the melody is the most likely dimension that he will use. Moreover, for non expert users, melody and rhythm (and lyrics) are the more simple dimensions for describing their information needs.

Query processing can significantly differ within content-based approaches. After a query has been played, the system can represent it either as a single note string, or as a sequence of smaller note fragments. The latter can be either arbitrary note strings, such as n-grams, or fragments extracted using melody information. Regarding the query as a single note string makes content-based retrieval very difficult since it would be similar to retrieving textual files using Unix grep-like commands which provides very poor results. On the contrary, extracting fragments using melody information can result in a more effective query description. We then speak about partial match retrieval.

7.6.3.3 Music Digital Libraries

Digital library projects have been carried out for designing, implementing, and testing real MIR systems. Some of them implement data-based, content-based, or both approaches to MIR. We cite some of the projects being most relevant to our research aims. The reader can access to the cited papers to have a



complete description of methods and systems. The VARIATIONS digital library has been reported in [9], while the MELDEX project is reported in [4]. A project involved the University of Milan and the Teatro alla Scala, Milan [10] to implement a multimedia object-relational database storing the music contents of the archive, as well as catalogue data about the nights at the Teatro alla Scala. The access to the archive is basically based on fragment extraction and approximate string matching. A feasibility study was conducted for the ADMV (Digital Archive for the Venetian Music of the Eighteenth century) digital library project [3]. The feasibility study allowed for defining architecture, technology, and search functions for a data and content-based MIR and database management system. The system complexity is due to the number of inter-relationships of all the aspects being typical of a real effective DL: distributed databases, preservation, wide area networking, protection, data management, content-based access.

7.6.4 Techniques for Music Information Retrieval

Content-based MIR is a quite new research area, at least compared to classical textual IR. For this reason, most of the techniques applied to retrieve music documents derive from IR techniques. In this section, after introducing some terminology typical of content-based description of music documents, techniques for MIR and their relationship with IR techniques are described. A final example is given on how evaluation can be carried out.

7.6.4.1 Terminology

There is a number of terms that have a special meaning for the research community on MIR.

A **feature** is one of the characteristics that describe subsequent notes in a score. A note feature can be: the pitch, the pitch interval with the previous note (PIT), a quantized PIT, the duration, the interonset interval with the subsequent note (IOI), the ratio of IOI with the previous note, and so on. All the features can be normalized or quantized. In the example of sect. 5.6.5.4, features are related to pitch and rhythm that, though usually correlated, can be treated independently. For example, many songs can be guessed only by tapping the rhythm of the melody while other ones can be easily recognized even if played with no tempo or rubato.

A **string** is a sequence of features. Any sequence of notes in a melody can be considered a string. It can be noted that strings can be used as representative of a melody, which is the idea underlying many approaches to MIR, but the effectiveness by which each string represents a document may differ. For instance, it is normally accepted that the first notes of a melody play an important role in recognition, or that strings that are part of the main theme or motif are good descriptors as well. String length is an important issue: Long strings are likely to be effective descriptors, yet they may lead to problems when the user is request to remember long parts of a melody for querying a MIR system. Often, strings shorter than three notes can be discarded, because they can be considered not significant descriptors.

A **pattern** is a string that is repeated at least twice in the score. The repetition can be due to the presence of different choruses in the score or by the use of the same music material (e.g., motifs, rhythmical cells) along the composition. Each pattern is defined by the string of features, by its length n and by the number of times r it is repeated inside the score. All patterns that appear only inside longer patterns have been discarded in the example of sect. 5.6.5.4. The computation of patterns can be done automatically using well known algorithms for pattern discovery. Given a particular feature, patterns can be considered as effective content descriptors of a music document. Depending on the selected feature, patterns carry different information about document content.

It can be noted that a music documents may be directly indexed by its strings. In particular, it can be chosen to describe a document with all its strings of a given length, usually from 3 to 5 notes, that are called *n-grams*. The n-gram approach is a simple, but often effective, alternative to more complex



approaches that are based on melodic information. In the following sections, patterns are considered as possible content descriptors, yet the discussion may be generalized to n-grams, musical phrases, and so on. Moreover, in the following discussion, three kinds of features are considered for the pattern selection step – the interonset interval (IOI) normalized to the quarter note, the pitch interval (PIT) in semitones, and both (BTH).

7.6.5 Document Indexing

Document indexing is a mandatory step for textual information retrieval. Through indexing, the relevant information about a collection of documents is computed and stored in a format that allows easy and fast access at retrieval time. Document indexing is carried out only when the collection is created or updated, when users are not yet accessing the documents, and then the problems of computational time and efficiency are usually less restrictive. Indexing speeds up retrieval time because it is faster to search for a match inside the indexes than inside the complete documents.

Following the terminology introduced in the previous section, each document may be indexed by a number of patterns of different length and with different multiplicity. If it is assumed that patterns are effective descriptors for document indexing, the first step of document indexing consists in the automatic computation of the patterns of each document. As previously mentioned, relevant features which are usually taken into account are IOI, PIT, and BTH. Pattern computation can be carried out with ad-hoc algorithms that compute exhaustively all the possible patterns, and store them in a hash table.

An exhaustive pattern discovery approach highlights a high number of patterns that have little or no musical meaning; for instance, a pattern that is repeated only two or three times in a document is likely to be computed by chance just because the combination of features is repeated in some notes combinations. Moreover, some patterns related to scales, repeated notes, or similar musical gestures, are likely to appear in almost all documents and hence to be poor discriminants among documents. In general, the degree by which a pattern is a good index may vary depending on the pattern and on the document. This is a typical situation of textual information retrieval, where words may describe a document to a different extent. For this reason it is proposed to apply the classical $tf \cdot idf$ weighting scheme.

The extent by which a pattern describes a document is the result of the multiplication of two terms. The **term frequency** is the number of occurrences of a given pattern inside a document. Hence, the term frequency of pattern p for document d can be computed as

$$tf_p^d = \# \text{ occurrences of } p \in d$$

The **inverse document frequency** takes into account the number of different documents in which a pattern appears. The inverse document frequency of pattern p can be computed as

$$idf_p = -\log \frac{\# \text{ documents containing } p}{\# \text{ documents}}$$

Relevant patterns of a document may have a high tf – they are frequent inside the document – and/or a high idf – they are infrequent across the collection.

For the aims of indexing, a document is described by a sparse array, where each element is associated to a different pattern in the collection. The value of each element is given by the $tf \cdot idf$ value. The index is built as an inverted file, where each term of the vocabulary is a different pattern in a given notation (i.e., a text string). Each entry in the inverted file corresponds to a different pattern, and can efficiently be computed in an expected time $O(1)$ with a hashing function. Given the different sets of features, three inverted files are built, respectively for features IOI, PIT, and BTH. Inverted files can be efficiently stored in memory, eventually using compression, and fast accessed at retrieval time. The size of the inverted



file and the implementation of the hashing function depend on the number of different patterns of the complete collection.

It may be useful to fix the maximum allowable pattern length to improve indexing. In fact, it is likely that very long patterns are due to repetitions of complete themes in the score and taking into account also them will give a quite sparse inverted file. Moreover, it is unlikely that a user will query the system singing a complete theme. These considerations suggest that long patterns could be truncated when they are over a given threshold.

7.6.5.1 Query Processing

For the query processing step, it can be assumed that users interact with the system according to a query-by-example paradigm. In particular, users should be able to describe their information needs by singing (humming or whistling), playing, or editing with a simple interface a short excerpt of the melody that they have in mind. Pitch tracking can be applied to the user's query in order to obtain a transcription in a notation format, such as a string of notes. The string representing the translated query needs to undergo further processing, in order to extract a number of descriptors that can be used to match the query with potentially relevant documents. It is normally assumed that a query is likely to contain strings that characterize the searched document, either because they appear very often inside its theme or because they are peculiar of that particular melody. In other words, a query is likely to contain relevant patterns of the searched document, which may have a high *tf* and/or *idf*.

The automatic detection of relevant strings cannot be carried out through pattern analysis, because normally queries are too short to have repetitions and hence to contain patterns. A simple approach to extract relevant strings, or potential patterns, from a query consists in computing all its possible substrings. That is, from a query of length q notes are automatically extracted $q - 2$ strings of three notes, plus $q - 3$ strings of four notes, and so on until the maximum allowable length for a pattern is reached. This approach can be considered similar to query expansion in textual information retrieval, which is known to increase recall at the risk of lowering precision. On the other hand, it is expected that most of the arbitrary strings of a query will never form a relevant pattern inside the collection, and then the negative effects on precision could be bounded.

7.6.5.2 Ranking Relevant Documents

At retrieval time, the strings are automatically extracted from the query and matched with the patterns of each document. The computation of potentially relevant documents can be carried out computing the distance between the vector of strings representing the query and the vector of patterns representing each document. Hence, for each document a Retrieval Status Value (RSV) is calculated, the higher the RSV, the closer the document with the query. A rank list of potentially relevant documents is computed from RSVs, obtaining a different rank lists for each of features used.

In general the orderings of documents in the rank lists differ. Differences may be due to many factors, as the diverse importance of rhythm and melodic profile for a the document collection, the effect of errors in the query, the kind of melodic excerpt chosen by the user as a representative of his information needs. It is expected that BTH ranking will give high scoring to the relevant documents when the query is sufficiently long and correctly played, because BTH patterns are a closer representation of the original melody. On the other hand, IOI and PIT are robust to query errors in melodic profile and rhythm, respectively. Moreover, simple representations as IOI and PIT are expected to be less sensitive to query length because of the possible presence of subpatterns of relevant motifs.

It is possible to take advantage from the existence of different rank lists by fusing together the results, in order to give the user a single rank list which takes into account the results of the three parallel



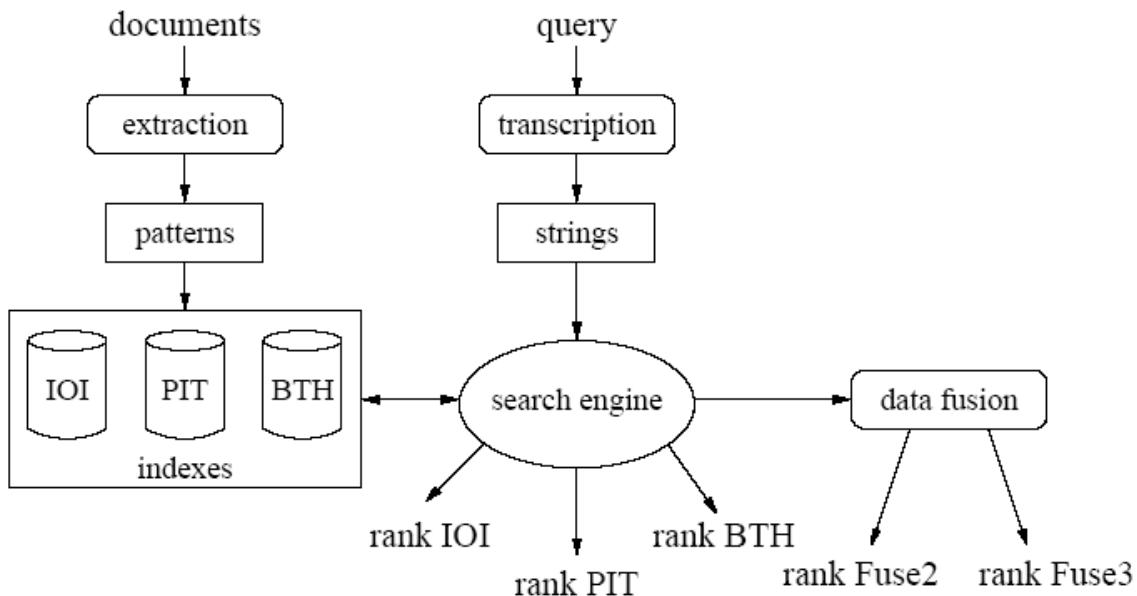


Figure 7.29: The phases of a methodology for MIR: Indexing, retrieval, and data fusion

approaches. This is a typical problem of **data fusion**, an approach that is usually carried out in the research area of Meta Search Engines, where the results obtained by different indexing and retrieval methodologies are combined – or fused – together according to a predefined weighting scheme. Since the RSVs of individual search engines are not known, or not comparable with others, the classical approach to data fusion is based on the information of rank only. In the case of MIR based on parallel features, the fusion can be carried out directly using the RSVs, because they are all based on the same $tf \cdot idf$ scheme. A new RSV can be computed as a weighted sum of RSVs of single features obtaining a new rank list.

A complete methodology for MIR shown in Figure 5.29, where steps undertaken at indexing time are shown on the left, while the operations that are performed at retrieval time are shown on the right. From Figure 5.29 and the above discussion, it is clear that the computational complexity depends on the query length – i.e., the number of strings that are computed from the query – while it is scalable on the number of documents. This is an important characteristic given by indexing techniques, because the time needed to reply to a query can be reasonably low also for large collections of documents.

7.6.5.3 Measures for Performances of MIR Systems

The output of almost any information retrieval system, and this applies also to MIR, is a ranked list of potentially relevant documents. It is clear that only the final user can judge if the retrieved documents are really relevant to his information needs. That is, the user should evaluate system performances in terms of retrieval effectiveness. There are two main reasons why the user may not be satisfied by the result of an information retrieval system.

- the system does not retrieve documents that are relevant for the user information needs – which is usually called **silence effect**;
- the system retrieves documents that are not relevant for the user information needs – which is usually called **noise effect**

All real systems for MIR try to balance these two negative effects. From the one hand, a high silence effect may result in not retrieving all the music documents that are similar to a given query sung by the user. From the other hand, a high noise effect may cause the user to spend great part of a retrieval session in listening to irrelevant documents.

Even if user satisfaction plays a central role in the evaluation of performances of a MIR system, and in general of any IR system, user studies are very expensive and time consuming. For this reason, the IR research community usually carries out automatic evaluation of the proposed systems using commonly accepted measures. In particular, there are two measures that are connected to the concepts of silence and noise effects. The first measure is **recall**, which is related to the ability of a system to retrieve the highest percentage of relevant documents (thus minimizing the silence effect). Recall is defined as

$$\text{recall} = \frac{\# \text{ relevant retrieved}}{\# \text{ total relevant}}$$

that is the number of relevant documents retrieved by the system divided by the total number of relevant documents in the complete database of documents. The second measure is **precision**, which is related to the ability of the system of retrieving the lowest percentage of irrelevant documents (thus minimizing the noise effect). Precision is defined as

$$\text{precision} = \frac{\# \text{ relevant retrieved}}{\# \text{ total retrieved}}$$

that is the number of relevant documents retrieved by the system divided by the total number of retrieved documents. An ideal system retrieved only relevant documents, and hence has 100% recall and precision. For real systems, high precision is usually achieved at the cost of low recall and viceversa.

Both precision and recall do not take into account that a MIR system may output a rank list of documents. For this reason it is a common practice to compute these measures also for the first N documents (for $N \in \{5, 10, 20, \dots\}$) and, in particular, to compute the precision at given levels of recall. Another approach is to summarize these measures, and the effect of the documents rank, in a single measure. For instance, the **average precision** is computed as the mean of the different precisions computed each time a new relevant document is observed in the rank list.

The evaluation of MIR systems is usually carried out on a test collection according to the Cranfield model for information retrieval, which is used at the Text REtrieval Conference (TREC). A test collection consists in a set of documents, a set of queries, and a set of relevance judgments that match documents to queries. The creation of a common background for evaluation is still an open issue in the MIR community, hence each research group created its own test collection from scratch. A “good” test collection should be representative of real documents and, in particular, of real user’s queries. The size of the document set, as well as the way queries are collected, may deeply influence the evaluation results. Relevance judgments should be normally given by a pool of experts in the music domain, which is an expensive task, but they can also be automatically constructed when queries are in the form of excerpts of a known tune. In this latter case, only the document from which the query derives is considered as relevant.

7.6.5.4 An Example of Experimental Evaluation

In the following paragraphs, the result of an experimental evaluation of a running MIR system are reported. The system is based on pattern analysis, based on three alternative features (IOI, PIT, and BTH) and data fusion techniques applied to the combination of IOI and PIT, called Fuse2, and the combination of all the three features, called Fuse3.



The Test Collection A small test collection of popular music has been created using 107 Beatles' song in MIDI format downloaded from the Web. As for any test collection, documents may contain errors. In a preprocessing step, the channels containing the melody have been extracted automatically and the note durations have been normalized; in case of polyphonic scores, the highest pitch has been chosen as part of the melody. After preprocessing, the collection contained 107 complete melodies with an average length of 244 notes, ranging from 89 of the shortest melody to 564 of the longest. Even if a number of approaches for performing automatic theme extraction has been already proposed in the literature, the methodology relies on indexing of complete melodies, because repetitions of choruses and verses can be taken into account by the $tf \cdot idf$ measure.

A set of 40 queries has been created by randomly selecting 20 themes in the dataset and using the first notes of the chorus and of the refrain. The initial note and the length of each query were chosen to have recognizable motifs that could be considered representative of real users' queries. The queries had an average length of 9.75 notes, ranging from 4 to 21 notes. Only the theme from which the query was taken was considered as relevant. Using this initial set of correct queries, an alternative set has been created by adding errors on pitch, duration, and both, obtaining a new set of 120 queries. A simple error model has been applied, because errors were uniformly distributed along the notes in the queries, with a probability of about 13.3%. As for many approaches to approximate string matching, an error can be considered the result of a deletion and an insertion, thus these alternative sources of errors have not been explicitly modelled. Tests on robustness to query length were carried out by automatically shortening the initial queries by an increasing percentage, disregarding the fact that query would not sound musical. In this way, 160 more queries with decreasing length have been automatically generated. For all the modified queries, only the theme of initial query was considered as relevant. In the following, we will refer to the only relevant document with the term *r-doc* for all the experiments.

Truncation of Patterns All the experimental analyses, whose results are shown in the following sections, have been carried out after truncating patterns longer than a given threshold t . When a pattern $[f_1 \dots f_n]$ had a length of $n > t$, it has been replaced (in the indexing step) by all its subpatterns of exact length t , that is the $n - t + 1$ subpatterns $[f_1 \dots f_t]$, $[f_2 \dots f_{t+1}]$, and so on until $[f_{n-t} \dots f_n]$, where some of the subpatterns may be already extracted, because they were part of other motifs.

With the aim of computing the optimal threshold for the test collection, five different thresholds have been tested, respectively 5, 7, 10, 15, and 20 notes. The retrieval effectiveness decreased with high values of the threshold, meaning that a compact representation of patterns can be more effective than longer ones. The average precision was approximately constant when thresholds higher than 15 – 20 notes were applied, probably because the number of different patterns longer than 20 notes is less than 8% and with a low value of r . The use of short patterns can be a useful way to control the increase of the index when new documents are added to the collection. Due to simple combinatorial reasons, the number of different patterns is bounded by the pattern length; on the other hand, the use of short patterns has the drawback of a higher number of patterns that are in common among documents, which may lower precision. It is interesting to note that data fusion approaches gave consistently better results than single approaches. This behaviour has been found in all our experiments, which are presented in the following sections, where results are shown only for $t = 5$.

Retrieval Effectiveness The first detailed analysis regarded the retrieval effectiveness with the set of 40 correct queries. Results are shown in Table 5.1, where the average precision (Av.Prec.), the percentage queries that gave the r-doc within the first k positions (with $k \in \{1, 3, 5, 10\}$), and the ones that did not give the r-doc at all ("not found"), are reported as representative measures. As it can be seen, IOI gave the poorest results, even if for 90% of the queries the r-doc were among the first three retrieved. The



highest average precision using a single feature was obtained by BTH, with the drawback of an on-off behaviour: either the r-doc is the first retrieved or it is not retrieved at all (2.5% of the queries). PIT gave good results, with all the queries that found the r-doc among the first three documents.

	IOI	PIT	BTH	Fuse2	Fuse3
Av.Prec.	0.74	0.93	0.98	0.96	0.98
= 1	57.5	87.5	97.5	92.5	95.0
≤ 3	90.0	100	97.5	100	100
≤ 5	95.0	100	97.5	100	100
≤ 10	97.5	100	97.5	100	100
not found	0	0	2.5	0	0

Table 7.1: Retrieval effectiveness for correct queries

The best results for Fuse2 and Fuse3 have been obtained assigning equal weights to the single ranks. When the $tf \cdot idf$ scores had different weights an improvement was still observed in respect to single rankings, though to a minor extent. For this reason, results for Fuse2 and Fuse3 are presented only when equal weights are assigned.

Robustness to Errors in the Queries Users are likely to express their information needs in an imprecise manner. The query-by-example paradigm is error prone because the example provided by the user is normally an approximation of the real information need. In particular, when the user is asked to sing an excerpt of the searched document, errors can be due to imprecise recall of the melody, problems in tuning, tempo fluctuations, and in general all the problems that untrained singers have. Moreover, transcription algorithms may introduce additional errors in pitch detection and in melody segmentation. The robustness to errors has been tested on an experimental setup. Since indexing is carried out on melodic contour and on rhythm patterns, the errors that may affect the retrieval effectiveness regard the presence of notes with a wrong pitch and a wrong duration. As previously mentioned, a set of queries with automatically added errors has been generated in order to test the robustness of the approach in a controlled environment.

As expected, the performances of IOI dropped for queries with errors in rhythm and the same applied to PIT for queries with errors in pitch. The same considerations apply to BTH in both cases, with an even bigger drop in the performances. It is interesting to note that data fusion allowed for compensating the decreases in performances of single ranks, giving for both Fuse2 and Fuse3 an average precision equal to the one obtained without errors. In the case of errors in both pitch and rhythm, also Fuse2 and Fuse3 had a decrease in performances, even if their average precision was consistently higher than the one of single features.

The experimental results showed that Fuse3 gave a considerable improvement in respect to the single rankings contribution. A query-by-query analysis showed that this behaviour is due to the fact that the sum of $tf \cdot idf$ scores of the single features gave always a new ranking where the r-doc was at the same level of the best of the three separate ranks; that is, if one of the three gave the r-doc as the most relevant document, also Fuse3 had the r-doc in first position. Moreover, for some queries, the fused rank gave the r-doc at first position even if none of the three single ranks had the r-doc as the most relevant document. These improvements can be explained by two factors: First, when the r-doc was retrieved at top position by one of the features, it had a very high $tf \cdot idf$ score that gave an important contribution to the final rank; Second, the r-doc was often retrieved with a high rank by two or three of the features, while in general other documents were not considered as relevant by more than one feature. Similar considerations apply,



though at a minor extent, also to Fuse2.

Dependency to Query Length A final analysis has been carried out on the effects of query length to the retrieval effectiveness. It is known that users of search engines do not express their information needs using much information. The community of information retrieval had to face the problems of finding relevant information also with vague or short queries. To some extent, a similar problem applies to MIR because users may not remember long excerpts of the music documents they are looking for. Moreover, untrained singers may not like to sing for a long time a song that they probably do not know very well. The effects of query length on a MIR system should then be investigated.

Tests on the dependency to query length have been carried out on a set of queries that were obtained from the original set of queries by shortening the number of notes from 90% to 60% of their original lengths. With this approach, queries may become very short, for instance a query of two notes cannot retrieve any document because patterns shorter than three notes are not taken into account.

Consistently with previous results, Fuse3 gave the best performances and showed a higher robustness to decrease in query length. Also in this case results showed that the data fusion approach was enough robust to changes in the initial queries. As previously mentioned, each initial query has been created selecting a number of notes that allowed to recognize the theme by a human listener. Moreover, each query was made by one or more musical phrases – or musical gestures or motifs – considering that a user would not stop singing his query at any note, but would end the query in a position that have a “musical sense”. For this reason, tests on query length can give only a general indication on possible changes in retrieval effectiveness.

7.6.6 Conclusions

This section present a short overview on some aspects of music IR. In particular, the issues typical of the music language have been discussed, taking into account the problems of formats and the role of the user. A number of approaches that have been proposed in the literature are presented, in particular the ones related to music Digital Libraries.

There are a number of aspects that are beyond the scope of this overview. In particular, all the research work related to audio processing that, even if not central to music IR, plays an important role in creating tools for classification of audio files and automatic extraction of low level features, that may be useful for expert users.

7.7 Commented bibliography

The reference book for Auditory scene analysis is Bregman [1990]. The Implication realization model is described in Narmour [1990]. The Local Boundary Detection algorithm is presented in Cambouropoulos [2001]. The Generative Theory of Tonal Music is described in Lerdahl and Jackendoff [1983].

Research on automatic metadata extraction for MIR can be classified in two main fields, depending on the two different classes of formats in which a music document can be represented: the automatic extraction of relevant information from a music score, which is typically achieved through melody segmentation and indexing; the automatic categorization of a music recording, which is typically achieved through audio classification. In this chapter we deal with the first field.

In the case of melody segmentation and indexing, the main assumption is that it is not possible to use textual descriptors for music documents, in particular for compositions and for melodies. Since it is not clear what kind of meaning is conveyed by a music document, the common approach is to describe a document using perceptually relevant elements, that may be in the same form of the document



itself (that is the only way to describe music is through music). Clearly, the alternative description of a music document should be more compact and summarize the most relevant information, at least from a perceptual point of view. The music language may be characterized by different dimensions, which may regard the score representation (e.g., melody, harmony, rhythm) the recording of performances (e.g., timbre, instrumentation) and high level information (e.g., structure, musical form). Among the different dimensions, melody seems to be the most suitable for describing music documents. First of all, users are likely to remember and use, in a query-by-example paradigm, parts of the melody of the song they are looking for. Moreover, most of the dimensions require a good knowledge of music theory to be effectively used, reducing the number of potential users to scholars, composers, and musicians. Finally, melody can benefit from tools for string analysis and processing to extract relevant metadata. For these reasons, most of the research work on metadata extraction focused on melody segmentation and processing. The need for automatic melody processing for extracting relevant information to be used as alternative descriptors, arises from the fact that the melody is a continuous flow of events. Even though listeners perceive the presence of elements in the melodic flow, which may be called lexical units, there is no explicit separator to highlight boundaries between them. Moreover, it is well known that there are parts of the melody (e.g., the incipit, the theme, the leit-motiv, and so on) that are more relevant descriptors of a music document than others. Yet, the automatic labelling of these relevant parts needs ad-hoc techniques.

One of the first works, probably the most cited in the early literature on MIR, is Ghias et al. [1995]. In this paper it is proposed the use of a query-by-example paradigm, with the aim of retrieving the documents that are more similar to the melody excerpts sung by the user: both documents and queries are transformed in a different notation that is related to the melodic profile. An alternative approach to MIR is proposed in Blackburn and DeRoure [1998], where metadata is automatically computed and stored in a parallel database. Metadata is in the form of hyperlinks between documents that are judged similar by the system.

Music language is quite different from other media, because it is not clear if music conveys a meaning and how a music document can be effectively described; this mostly because perception plays a crucial role in the way users can describe music. The important issue of perception is faced in Uitdenbogerd and Zobel [1998], where a user study is presented on users melody representation. The knowledge of music structure is exploited in Melucci and Orio [1999] for extracting relevant information, where music documents and queries are described by surrogates made of a textual description of musical lexical units. Experiments on normalization are also reported, in order to cope with variants in musical lexical units that may describe similar documents. In Bainbridge et al. [1999] is proposed a multimodal description of music documents, which encompasses the audio, a visual representation of the score, the eventual lyrics, and other metadata that are automatically extracted from files in MIDI format.

An alternative approach to automatically compute melodic descriptors of music documents is presented in Bainbridge et al. [1999], which is based on the use of N-grams as musical lexical units. Alternatively, musically relevant phrases are proposed in Melucci and Orio [2000], where an hypertextual structure is automatically created among documents and musical phrases. In this case a document is described by a set of links to similar documents and to its most relevant phrases. Musical structure is exploited in Hoos et al. [2001] for computing a set of relevant features from a music document in a complex notation format.

Alternatively to previous works, in Birmingham et al. [2001] it is proposed that a good descriptor of a music document is its set of main themes, which are units longer than N-grams or musical phrases. Themes are modelled through the use of Markov chains. An extension to hidden Markov models is presented in Shifrin et al. [2002], where possible mismatches between the representation of the query and of the documents are explicitly modelled by emission probabilities of Hidden Markov Models states. An evaluation of different approaches is presented in Hu and Dannenberg [2002], where the problem of



efficiency is raised and discussed.

References

- D. Bainbridge, C.G. Nevill-Manning, I.H. Witten, L.A. Smith, and McNab R.J. Musical information retrieval using melodic surface. In *Proc. International Symposium on Music Information Retrieval*, pages 161–169, 1999.
- W.P. Birmingham, R.B. Dannenberg, G.H. Wakefield, M. Bartsch, D. Bykowski, D. Mazzoni, C. Meek, M. Mellody, and W. Rand. Musart: Music retrieval via aural queries. In *Proc. International Symposium on Music Information Retrieval*, pages 73–82, 2001.
- S. Blackburn and D. DeRoure. A tool for content based navigation of music. In *Proc. ACM Multimedia Conference*, pages 361–368, 1998.
- A. S. Bregman. *Auditory Scene Analysis*. MIT Press, 1990.
- E. Cambouropoulos. The local boundary detection model (lbdm) and its application in the study of expressive timing. In *Proc. Int. Computer Music Conf.*, 2001.
- A. Ghias, J. Logan, D. Chamberlin, and B.C. Smith. Query by humming: Musical information retrieval in an audio database. In *Proceedings of ACM Digital Libraries (DL) Conference*, pages 231–236, 1995.
- H.H. Hoos, K. Renz, and M. Gorg. GUIDO/MIR - an experimental musical information retrieval system based on guido music notation. In *Proc. International Symposium on Music Information Retrieval*, pages 41–50, 2001.
- N. Hu and R.B. Dannenberg. A comparison of melodic database retrieval techniques using sung queries. In *Proc. ACM/IEEE Joint Conference on Digital Libraries*, pages 301–307, 2002.
- F. Lerdahl and R. Jackendoff. *A Generative Theory of Tonal Music*. The MIT Press, 1983.
- M. Melucci and N. Orio. Musical information retrieval using melodic surface. In *Proc. 4th ACM Conference on Digital Libraries*, pages 152–160, 1999.
- M. Melucci and N. Orio. Smile: a system for content-based musical s information retrieval environments. In *Proc. Intelligent Multimedia Information Retrieval Systems and Management (RIAO) Conference*, pages 1246–1260, 2000.
- B. C. J. Moore, B. R. Glasberg, and T. Baer. A model for the prediction of thresholds, loudness, and partial loudness. *Journal of the Audio Engineering Society*, 45(5):224–240, April 1997.
- Eugene Narmour. *The Analysis and cognition of basic melodic structures : the implication-realization model*. University of Chicago Press, 1990.
- W. Schloss. *On the Automatic Transcription of Percussive Music: From Acoustic Signal to High Level Analysis*. PhD thesis, Stanford University, 1985.
- J. Shifrin, B. Pardo, C. Meek, and W. Birmingham. Hmm-based musical query retrieval. In *Proc. ACM/IEEE Joint Conference on Digital Libraries*, pages 295–300, 2002.
- A. Uitdenbogerd and J. Zobel. Manipulation of music for melody matching. In *Proc. ACM Multimedia Conference*, pages 235–240, 1998.



Chapter 8

Recognizing and communicating expressive information

Giovanni De Poli

Copyright © 2005-2018 Giovanni De Poli

except for paragraphs labeled as *adapted from <reference>*

This book is licensed under the CreativeCommons Attribution-NonCommercial-ShareAlike 3.0 license. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/>, or send a letter to Creative Commons, 171 2nd Street, Suite 300, San Francisco, California, 94105, USA.

8.1 The quest for expressiveness

During the last decade, lot of research effort has been spent to connect two worlds that seemed to be very distant or even antithetic: machines and emotions. Mainly in the framework of human-computer interaction an increasing interest grew up in finding ways to allow machines communicating expressive, emotional content. Such interest has been justified with the objective of an enhanced interaction between humans and machines exploiting communication channels that are typical of human-human communication and that can therefore be easier and less frustrating for users, and in particular for non technically skilled users.

Starting from the findings from psychology and neurosciences, research has been aimed at developing computational models and algorithms for analysis and synthesis of emotional content. While from the one hand research on emotional communication found its way into more traditional fields of computer science like Artificial Intelligence, on the other hand novel fields developed explicitly focusing on such issues. Examples are researches on Affective Computing in the United States, KANSEI Information Processing in Japan and Expressive information processing in Europe.

In this section ¹ Affective Computing and KANSEI Information Processing are shortly described with reference to the work of the two researchers that in a certain way started the two fields: Rosalind Picard and her group at MIT Media Lab for Affective Computing, and Shuji Hashimoto and his group at Waseda University, Tokyo, for KANSEI Information Processing.

In the following sections, analysis and synthesis of expressive content in performing arts (a typical European research stream), with a particular reference to cultural application, is presented.

¹adapted from PhD dissertation of Gualtiero Volpe (2003)

8.1.1 Affective Computing: the American way to artificial emotions

The Affective Computing approach is mainly illustrated in the homonymous book (Picard, 1997). In her book Picard defines Affective Computing as *computing that relates to, arises from, or deliberately influences emotions*. Affective Computing addresses the design and implementation of machines that are able to

- recognize emotions,
- express emotions,
- have emotions.

These are human-centred machines that observe their users and sensitively interact with them by expressing emotions depending on what they observed and on the current "emotional state" of the machine.

- Computers that are able to *recognize* emotions are conceived as systems collecting a variety of input signals ranging from face expressions to voice, movement features (e.g., hand gestures, gait, posture), physiologic measures (e.g., respiration, electrocardiogram, blood pressure, temperature). They perform feature extraction and classification on these inputs (e.g., video analysis of movement, audio analysis of speech) and try to classify the emotion the user is communicating through a reasoning process taking into account information about context, situations, personal goals, social display rules, and other emotion related data. Learning techniques can be employed to adapt recognition to a specific user (e.g., a personal computer can learn the habits of its master to improve its performances in the recognition task). If the computer has an emotional state, this can influence the recognition process.
- Computer that are able to *express emotions* (either depending on instructions given by humans or as a result of an internal mechanism for generating emotions) are systems that modulate audio (e.g., synthetic voice, sound, music) and visual signals (e.g., face, posture, gait of animated creatures, colours) in a way suitable for the emotion that has to be communicated. The expressed emotion can be intentional (i.e., deliberated as a result of a reasoning process) or spontaneous (i.e., reactively triggered). It can directly express the affective state of the machine that can in turn be influenced by the expression of the emotion. Expression partially depends on social display rules.
- If computers *can have* emotions is perhaps one of the most controversial issues in Affective Computing. In her book, Picard proposes to consider five components of an emotional system: a computer can be said to have emotions if all five components are present in it. The five components are the following:
 1. *Emergent emotions and emotional behaviour* i.e. the machine is able to express an emotion through its behaviour even if it does not have any emotion. By observing the machine behaviour, humans naturally tend to attribute an emotional state to the machine.
 2. *Fast primary emotions* i.e. mechanisms to generate a kind of hard-wired, reactive responses (especially to potentially harmful events). Fast primary emotions are what Damasio calls primary emotions (Damasio, 1994). Studies about the mechanisms triggering such emotions can be found in neurosciences. They are associated with the inner regions of the brain.
 3. *Cognitively generated emotions* i.e. emotions that are generated as a result of explicit reasoning. Cognitively generated emotions are slower than fast primary emotions and are usually consequence of deliberate thoughts. They are located in the brain cortex. Several cognitive



models of emotion have been developed. One of the most famous is the model by Ortony, Clore, and Collins, usually referred as OCC model (Ortony, Clore, and Collins, 1988) that has been also employed in a number of concrete applications. Originally, the OCC model was not developed for building machines that could have emotions; rather it was conceived as a way for reasoning about emotions. The model develops a collection of rules associating emotions to cognitive evaluations about consequences of events, actions of agents, and aspects of objects.

4. *Emotional experience* i.e. the system is cognitively aware of its emotional state. Emotional experience consists of cognitive awareness, physiologic awareness and subjective feelings. If it is possible to have such an emotional experience in a machine and, if yes, how it can be implemented is still an open and quite tricky issue. It relates to consciousness and requires the machine to have sensors able to measure its own "emotional state".
5. *Body-mind interactions* i.e. the emotional state can influence other processes simulating similar human physical and cognitive functions like memory, perception, decision making, learning, goals, motivations, interest, planning, etc.

Research on Affective Computing has been applied in a number of application scenarios, ranging from entertainment, to edutainment, to detection of emotional responses (e.g., frustration) in particular relevant tasks (e.g., learning, driving), to the design and implementation of devices for analysis and synthesis of emotions.

With respect to the three issues mentioned above (i.e., machines recognizing, expressing, and having emotions), we will mainly address the first two aspects, i.e. the design and implementation of algorithms for recognizing and communicating expressive content, rather than with machines that *have* their own emotional state. In fact, if the goal is to open novel perspective to artistic performances by introducing new tools allowing an extension of the artistic languages by acting on the communicated expressive content through technology, what is mainly needed is

- the possibility to classify and encode in digital format the communicated expressive content in order to process it,
- the ability to produce suitable output to induce emotional reactions in spectators.

In other words, we believe that humans only have emotions. Machines do not need to have them, but they can give more and better support to human activities if they are able to process information not only related to the rational aspects of human behaviour, but also to the emotional ones.

8.1.2 The eastern approach: KANSEI Information Processing

In the same period the Affective Computing research started in the United States, another approach to understanding expressive content communication was developed in Japan: KANSEI Information Processing. According the Japanese view (Hashimoto, 1997) information processing has three phases (Tab. 6.1):

- *Physical information processing*. physical signals capturing data from the real world (e.g., sound, light, force) are identified as the first target of information processing. Signal processing is the technology field that is mainly responsible of processing such kind of information.
- *Semantic information processing*. The second phase is the semantic information processing to deal with knowledge and rule, that is the field of logic and symbolic knowledge. Artificial Intelligence is the discipline that mainly covers such aspects.



	Media	Regulation	Technology	Basis of reality
Physical phase	sound, light, force etc.	law of nature causality	signal processing virtual reality	physical explanation
Semantic	language knowledge	logic consistency	artificial intelligence data base	mathematical proof
KANSEI	music, art, poem	sympathy pleasure	KANSEI processing entertainment	emotional resonance

Table 8.1: Three phases of information processing. [from S. Hashimoto 1997]

- *KANSEI information processing.* The third target is KANSEI (a Japanese word) that refers to feelings, intuition, and sympathy and according to Hashimoto we are just entering in an historical period in which technology will start to deal with KANSEI, an issue that in the past was often left as a research field for only humanistic or humanistic related disciplines.

The exact meaning of the Japanese word KANSEI is something controversial for western people: it does not have a univocal correspondent in western languages and culture, but is rather associated to a collection of words related to the emotional sphere (e.g., emotion, sensibility, sensuality, sense, feeling). In his paper Hashimoto gives some examples of common uses of the word in Japanese language such as for example "Her KANSEI is excellent", "He is a man of rich KANSEI", "He has no KANSEI", "Her KANSEI seems well suited to me", etc. . It should be noticed that KANSEI refers to a dynamic process rather than to emotional labels or categories to be applied to expressive contents.

KANSEI Information Processing can be regarded as a coding and decoding process. In other words, KANSEI Information Processing supposes an underlying model in which expressive content is conceived as a kind of high-level information that, in the framework of a human-human communication process, *modulates* the physical signals carrying some usually symbolic message. That is, when a (human) sender sends a message to a (human) receiver he/she encodes in the message some expressive emotional information. Such information together with the symbolic content is embedded in the physical signal carrying the message. When the receiver receives the signal he/she decodes it and extracts both the symbolic message and the additional expressive information the sender encoded into it. Notice that it is not required that the sender deliberately add the expressive information to the message: such additional expressive information can be included unconsciously and can refer to aspects such as personality traits or personal dispositions toward objects, actions, and other people.

By making a comparison with the Affective Computing approach, it can be noticed that all the three aspects of recognizing, expressing, and having emotions are included in the KANSEI process: in fact,

- the sender expresses his/her emotions by encoding them in the physical signals carrying a message,
- the receiver recognizes the emotions expressed by the sender while decoding the message carried by the physical signals,
- sender and receiver have an emotional state that can both influence the encoding/decoding process and be itself the high-level additional expressive information encoded in a message.

KANSEI Information Processing seems therefore to adopt an holistic approach, broader with respect to the Affective Computing perspective because it includes in the same model of encoding/decoding process all the three aspect Affective Computing separately deals with, and because, while Affective Computing is more concerned with emotions, KANSEI rather refers to a wide collection of emotion related aspects (e.g., moods, feelings, personality traits etc.). This difference may reflect a cultural



difference between western and eastern approaches to problem solving: while western people usually tend to divide a problem in sub-problems following a top-down approach and sometimes losing the global perspective, eastern people often continue to keep an overall view of the problem even when they are focusing on a specific aspect of it.

8.1.3 Expressiveness in arts and culture

In general in human communication, two channels can be distinguished: one transmits explicit messages, which may be about anything or nothing; the other transmits implicit messages about the humans themselves. A lot of research is conducted in understanding the first, explicit channel, but less attention is paid to the second, which is not as well understood. Understanding the other party emotions is one of the key tasks associated with the second, implicit channel. Aim of the psychology study of emotion is understanding the mechanisms that intervene between message reaching a listener and an emotion being perceived, or experienced, by that person as a result of the message. Aim of the scientific and technological study of expressiveness, is to develop models able to describe such phenomena and systems for expression and emotion rendering and recognizing in multimodal communication.

In Europe the research on *expressive* communication via the implicit communication channel is often focused on artistic and cultural domain. This need of cross-fertilization opens novel frontiers to research in both the field of science and engineering and in the field of art and humanities: if from the one hand scientific and technological research can benefit of models and theories borrowed from psychology, social science, art and humanities, on the other hand these disciplines can take advantage of the tools technology provides for their own research, i.e., for investigating the hidden subtleties of human beings at a depth that was never reached before.

Art is often selected as main application scenario since it is a field where the analysis and synthesis of affect and expressiveness is of central importance. Making machines useful in artistic contexts that rely on different sensory modalities implies that the often subtle nuances of artistic expression should be dealt with in these different modalities. This, however, requires a technology that focuses focus on affect, emotion and expressiveness and cross-modality interactions.

8.2 Expressive systems and interfaces

Emotions are important in human intelligence, rational decision making, social interaction, perception, memory, learning, creativity, and more. They are necessary for intelligent day-to-day functioning. The negative connotation of *being emotional* or *acting emotionally* are not valid excuses for ignoring the study of emotions, or its application to computers. Instead, it is time to examine how emotions can be incorporated into models of intelligence and particularly, into computers and their interactions with humans.

To date, researchers trying to create intelligent computers have focused on problem solving, reasoning, learning, perception, language, and other cognitive tasks considered essential to intelligence. Most of them have been aware that emotion influences these functions in humans. Some have scoffed at the idea of giving computers emotions. However, now there is preponderance of evidence that emotion plays a pivotal role in functions considered essential to intelligence. This new understanding about the role of emotion in humans indicates a need to rethink the role of emotion computing.

Artists have a long history of communicating emotion in their work, but the process has been largely intuitive. Only recently have researchers tried to mechanize the dynamics of emotion, designing computational programs that can automatically change, for example, the state of an animated character from happy to sad. An emotional state change effects not just face, voice, and posture, but the entire spatio-temporal form of the character, physically and cognitively. The character will pick up an object differ-



ently when happy than when sad. The character will walk differently. Even the way that the character listens will be affected. Human emotion involves a constellation of interacting bodily signals, influencing thought and behaviour, modulating every action and movement. The next millennium will bring a balance to scientists understanding of cognition and emotion, and a more authentic model of human behaviour to graphics and animation. Researchers who try to synthesize mechanisms of human emotion in multimodal systems will ask questions that emotion researchers have yet to contemplate, and this will help advance knowledge. In the late nineteenth century, Freud and James argued for the importance of understanding emotion in human behaviour; but, the twentieth century has downplayed emotion, modelling man as an unimpassioned cognitive machine. Such a model describes only in part, as seeing in a mirror dimly. The new century will lead to the use of computing for more than making a perfect image; computing will be used to illuminate the nature of human emotion.

Terminology. Before proceeding, it is helpful to clarify a few pieces of terminology.

- *Emotional* and *affective* are often used interchangeably as adjectives describing either physical or cognitive components of emotion, although "affective" will sometimes be used in a broader sense than "emotional".
- An emotional *state* refers to your internal dynamics when you have an emotion. The state is multivariate, including aspects of both your mental state and physical state. It changes with time and with a variety of other activating and conditioning factors. Emotional state cannot be directly observed by another person, but may be inferred.
- An emotional *experience* refers to all you consciously perceive of your own emotional state. Some authors equate emotional experience with emotional *feelings*. Generally the term feelings refers to not just sensations as of pain and hunger, but also to subjective experience of affective phenomenon.
- The term emotional *expression* is used to describe what is revealed others, either voluntarily, such as by a deliberate smile, or involuntarily, such as by a nervous twitch. Emotional expression via the motor system or both bodily systems, is usually involuntary, and provides clues that others may observe to guess your emotional state.
- The term *mood*, although defined in many different ways in the literature, will be used to refer to a longer-term affective state. The precise duration is not well-defined, although moods can apparently last for hours, days, and maybe longer. In contrast, psychologists say that emotions are events that last at most a few minutes. A mood may arise when an emotion repeatedly activated, for example a bad mood may arise during a half hour reinforced negative thoughts or actions, or may be induced by taking drug or medication.
- The term *affective interfaces* indicates computer and software interfaces that can communicate emotions to users and recognize their emotional states (see Fig. 6.1).

8.2.1 Affective channel

The human voice is always changing. Even if the receptionist says the same "Good afternoon, welcome to Sirius Cybernetics Corporation," every time you call, his or her intonation is always slightly different: a cheerful hello, a brusque hello. The same person can speak the same words, but say them entirely differently. Sometimes the part of the message that communicate emotion is the most important part.



One problem the information age has brought is that of too much information, which tends to lead to cognitive fatigue and a reduced ability to accurately process new inputs. In contrast, information presented through the *affective channel* does not usually demand conscious attention. Affective information can be perceived in parallel with non-affective information, without increasing your workload. In speech, what is said can be considered the semantic information, and how it is said can be considered largely the affective information. The latter is communicated through the modulation of vocal parameters. Speaking a simple "Hello!" in a happier tone than usual is, for both the speaker and listener, less work than speaking the two separate messages: "Hello" and "I'm more happy than usual at this moment."

A machine can be used as a channel for transmitting human emotions. When two humans communicate via email or via teleconference, the machine and network act as a communication channel connecting people. Typically the channel is band-limited: all the information it receives at one end cannot be sent to the other end; some of it is lost. For example, it might convert a speech signal to text, throwing away the affective part of the signal. We might describe the affective bandwidth of a channel as how much affective information the channel lets through.

When sending the same words, different channels allow for more or for less affective channel capacity: email usually communicates the least affect, phone slightly more, video-teleconferencing more still, and "in person" communication the most. It is usually assumed that technology-mediated communication always has less affective bandwidth than person-to-person communication. Sometimes the limits on affective bandwidth are desirable. You might wish to choose a medium where your emotions are not as easily seen. However, rarely is it desirable to have these limits forced on you. Affective recognition and expression can be used to allow for more possibilities in communication, even with limited technology. For example, if there is not enough bandwidth to transmit each person's facial expression, the computer might recognize the expression, send just a few bits describing it, and represent these with an animated face on the other side of the channel.

Might technology increase affective bandwidth? Virtual environments and computer-mediated communication offer possibilities that we do not ordinarily have in person-to-person communication.

8.2.2 Affective interfaces

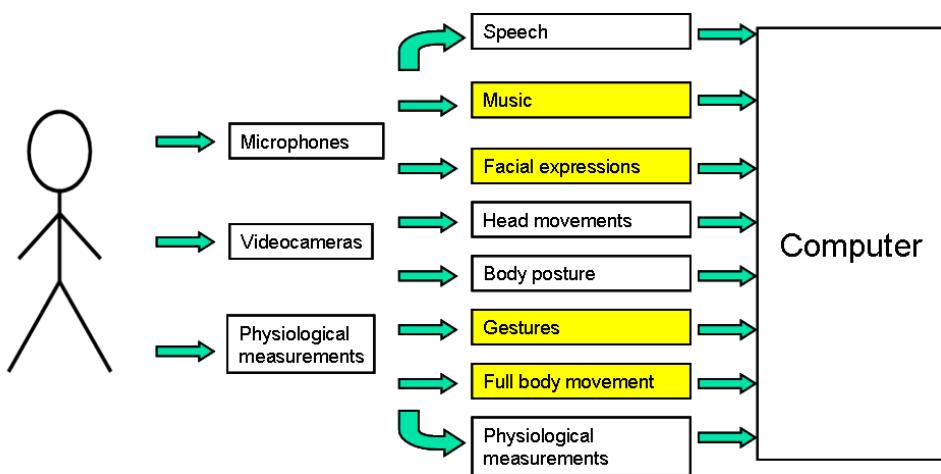


Figure 8.1: Affective interface.

One of the hallmarks of an intelligent computer will be its ability to recognize emotions, to infer an emotional state from observations of emotional expressions and through reasoning about an emotion-generating situation. The computer might try to recognize the emotions of its user, of other agents

with which it interacts, and of itself, if it has emotions. Recognition may require vision and hearing abilities for gathering facial expressions, gestures, and vocal intonation. Additionally, the computer may use other inputs that may or may not have analogies in human senses: reading infra-red temperature, measuring electrodermal response, and so forth. Once emotional expressions are sensed and recognized, the system can use its knowledge about the situation and its knowledge about emotion generation to infer the underlying emotional state which most likely gave rise to the expressions.

Criteria for affect recognition. Design criteria for a computer that can recognize emotion are summarized as follows:

- *Input.* Receives a variety of input signals, for example: face, voice, hand gestures, posture and gait, respiration, electrodermal response, temperature electrocardiogram, blood pressure, blood volume pulse, electromyogram, etc.
- *Pattern recognition.* Performs feature extraction and classification on these signals. For example: analyzes video motion features to discriminate a frown from a smile.
- *Reasoning.* Predicts underlying emotion based on knowledge about how emotions are generated and expressed. Ultimately, this ability requires perceiving and reasoning about context, situations, personal goals and preferences, social display rules, and other knowledge associated with generating emotions and expressing them.
- *Learning.* As the computer gets to know someone, it learns which of the above factors are most important for that individual, and gets quicker and better at recognizing his or her emotions.
- *Bias.* The emotional state of the computer, if it has emotions, influences its recognition of ambiguous emotions.
- *Output.* The computer names or describes the recognized expressions, and the emotions likely to be present.

Embedded in these criteria are numerous technical requirements, for example, receiving inputs requires accurate technology for gathering digital physiological, audio, and visual signals, as well as research to determine which signals are most important for the task at hand. In pattern recognition, informative features of the signals need to be identified (statistical structural, nonlinear, etc.) together with conditioning variables that influence the meanings of these features.

Criteria for affect communication. Design criteria for a computer that can express emotion are summarized as follows:

- *Input.* Computer receives instructions from a person, a machine, or from its own emotion-generation mechanisms if it has them, telling it what emotion(s) to express.
- *Intentional vs. spontaneous pathways.* The system may have at least two paths for activation of emotional expression: one that is intentional, and one that is spontaneous. The former is triggered by a deliberate decision, while the latter acts within a system that has emotion, automatically modulating some of the system's outputs with the current emotion.
- *Feedback.* Not only does affective state influence affective expression, but the expression can influence the state.



- *Bias-exclusion*. It is easiest to express the present affective state, and this state can make the expression of certain other states more difficult.
- *Social display rules*. When, where, and how one expresses emotions is determined in part by the relevant social norms.
- *Output*. System can modulate visible or vocal signals such as a synthetic voice, animated face, posture and gait of an animated creature, music, and background colours, in both overt ways such as changing a facial expression, and in subtle ways such as modifying discourse timing parameters.

8.2.2.1 Properties of affective signals.

Affective signals, which measures a persons emotional state, are characterized by the following properties of behaviour in a emotional system (Picard, 1997):

- *Response decay*: an emotional response is of relatively short duration, and will fall below a level of perceptibility unless re-activated.
- *Repeated strikes*: Rapid repeated activation of an emotion causes its perceived intensity to increase.
- *Temperament and personality influences*: A person's temperament and personality influence emotion activation and response.
- *Non-linearity* The human emotion system is non-linear, but may be approximated as a linear system for a certain range of inputs and outputs.
- *Time-invariance* The human emotional system can be modelled as independent of time for certain durations. For short durations, habituation effects occur. For long durations, factors such as a person's physiological circadian rhythms and hormonal cycles needed to be considered.
- *Activation*: Not all inputs can activate an emotion; they have to be of sufficient intensity. This intensity is not a fixed value, but depends of factors such as mood, temperament, cognitive expectation and context..
- *Saturation* No mater how frequently an emotion is activated, at some point the system will saturate and the response of the person will no longer increase. Similarly, the response cannot be reduced below a zero level
- *Cognitive and physical feedback*: Inputs to the system can be initiated by internal cognitive or physical processes. The physiological expression of an emotion can provide a feedback which acts as another input to the system, generating another emotional response.
- *Background mood*: All inputs contribute to a background mood, whether or not they are below the activation level for. emotions. The most recent inputs have the greatest influence on present mood.

The input of an emotional system is a complex function of cognitive and physical events. It is often approximated by a simple non-linear (sigmoid) function applied to the inputs to the emotional system:

$$y = \frac{g}{1 + e^{-\frac{x-x_0}{s}}} + y_0 \quad (8.1)$$

where x is the input stimuli (originating inside and/or outside the person). The output is y , the height of the curve. The parameter s controls the steepness of the slope, representing how fast the the output



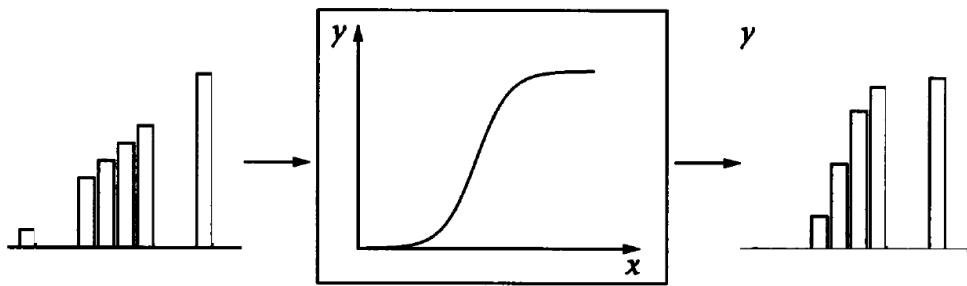


Figure 8.2: A sigmoid function is applied to the inputs to an emotional system (from Picard 1997).

y changes with the input x (depends on personality). Smaller values of s make the sigmoid steeper, and more responsive. The parameter x_0 shifts the sigmoid left or right. When it is far to the right, then a stronger input is required to activate an output. The sigmoid might be shifted left or right according to a person's mood. A good mood can allow smaller inputs to activate positive emotions, accomplished by shifting the sigmoid to the left. The parameter g controls the gain applied by the sigmoid. This value might be coupled to the arousal level of a person; someone highly aroused might be capable of experiencing a greater intensity of emotion. Finally, the parameter y_0 shifts the sigmoid up or down. This parameter might be controlled by cognitive expectation. The parameters of the sigmoid provide a rich set of controls for adjusting inputs before they proceed to activate emotions.

8.3 Mid-level representations of expressive information

It is convenient to distinguish different levels of representation of the expressive information. The traditional way is to have a low level representation as physical signals and a high level representation for semantic information, which can be both symbolic or affective (Fig. 6.3 left). In this perspective, recognizing means a direct mapping from low level features to semantic label or spaces. In this section the most important semantic representation of expressive information will be presented.

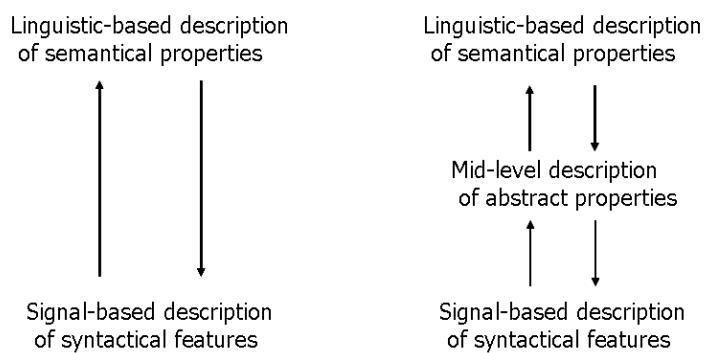


Figure 8.3: Representation levels of expressive information.

The process of trying to recognize an emotion is usually thought to involve a transformation from signal to symbol, from low-level physical phenomena to high-level abstract concepts. However, because reasoning about the situation can modify the kinds of observations that are made, information can be considered to flow not just from the low-level inputs to the high-level concepts, but also from the high level to the low level. Suppose that in reasoning about a situation you expect that somebody will be in a bad mood; in that case, your high-level expectation can cause your low-level perception to be biased in a

negative way, so that you are more likely to perceive a weak or ambiguous expression as being negative. The recognition of emotion therefore not merely bottom-up, from signals to symbols, but also top-down, in that higher level symbols can influence the way that signals are processed.

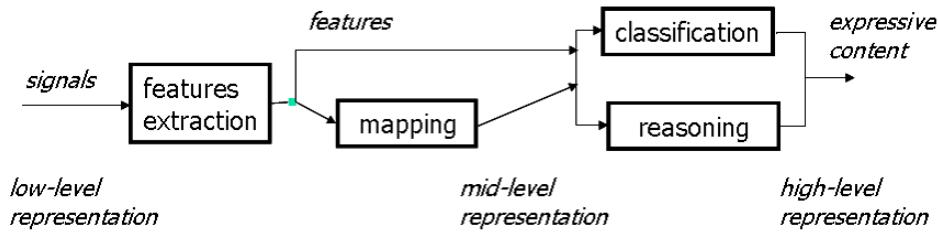


Figure 8.4: Expressive content recognition.

However for expressive information processing, it is convenient to define a mid level representation (Fig. 6.3 right), which can be different for diverse contexts or purposes. In Sect. 6.3.1 the main approaches to conceptualize emotions will be shortly overviewed. In Sect. 6.3.2 the theory of movement of Laban will be presented and a way to represent the expressive information conveyed by human gestures. In Sect. 6.3.3 a mid-level representation for expressiveness, derived from studies of music performance, will be introduced.

Consider what happens when you try to recognize somebody's emotion. First, your senses detect low-level signals motion around their mouth and eyes, perhaps a hand gesture, a pitch change in their voice and, of course, verbal cues such as the words they are using. Signals are any detectable changes that carry information or a message. Sounds, gestures, and facial expressions are signals that are observable by natural human senses, while blood pressure, hormone levels, and neurotransmitter levels require special sensing equipment. Second, patterns of signals can be combined to provide more reliable recognition. A combination of clenched hands and raised arm movements may be an angry gesture; a particular pattern of features extracted from an electromyogram, a skin conductivity sensor, and an acoustic pitch waveform, may indicate a state of distress. This medium-level representation of patterns can often be used to make a decision about what emotion is present. At no point, however, do you directly observe the underlying emotional state. All that can be observed is a complex pattern of voluntary and involuntary signals, in physical and behavioural forms.

The recognition process can be described as in Fig. 6.4. Physical signals, detected by the affective interfaces, are processed to extract low level features. Then a second processing stage maps these features into suitable trajectories or patterns in a mid-level representation. Finally by classification techniques and reasoning the semantic expressive information is obtained.

8.3.1 Approaches to conceptualizing emotions

Most people have an informal understanding, but there is a formal research tradition which has probed the nature of emotion systematically. The two theoretical traditions that have most strongly determined past research in this area are *categorical* (also called discrete) and *dimensional* emotion theories.

Categorical approach. The assumption of this approach is that people experience emotions as categories that are distinct from each other. Theorists in this tradition propose the existence of a small number, between 6 and 14, of basic or fundamental emotions that are characterized by very specific response patterns. From these basic emotions, all other emotional states can be derived. The focus is on characteristics that distinguish emotion from one another. There is a reasonable agreement

on four basic emotions: happiness, sadness, anger, fear. The next common two are disgust and surprise.

A problem with this approach is that different researchers propose different sets of basic emotions and the small number of primary basic emotions seems ill adapted to describe the extraordinary richness of the emotional effects of music reported in both fictional and scientific accounts. Moreover some emotions show up universally, and others seem to involve cultural specifics.

Dimensional approach. The use of two-dimensional valence-activation models has become very widespread in the affective sciences and is well represented in research on emotional effects of music. This approach has some obvious advantages. It is simple, easily understood by participants in experiments, and highly reliable.

The focus of this approach is on identifying emotions based on their placement on a space with a small number of dimensions. This space is derived from similarity judgements, analyzed using factor analysis or multidimensional scaling. Since the third dimension has been difficult to establish reliably in an empirical fashion via factor analyses, emotions are often defined in terms of a two-dimensional space.

Appraisal approach. One of the most influential emotion theories in modern psychology is the appraisal-based approach, which can be regarded as the extension of the dimensional approach described above. In this representation, an emotion is described through a set of stimulus evaluation checks, including the novelty, intrinsic pleasantness, goal-based significance, coping potential, and compatibility with standards. However, translating this scheme into one engineering framework for purposes of automatic emotion recognition remains challenging.

8.3.1.1 Valence-Arousal space

Valence-Arousal space is a dimensional representation that is both simple and capable of capturing a wide range of significant issues in emotion. The two major dimensions consist of the *valence* dimension (pleasant-unpleasant, agreeable-disagreeable) and an activity dimension (active-passive) sometimes also called *arousal* dimension. If a third dimension is used, it often represents either power or control. The most used representation is the Circumplex model of Russel (see Fig. 6.5). It presents a circular structure with activation and valence dimensions. It organizes emotions in terms of affect appraisal (pleasant - unpleasant) and physiological reaction (high - low arousal).

This approach provides a way of describing emotional states which is more tractable than using words, but which can be translated into and out of verbal descriptions. Translation is possible because emotion-related words can be understood, at least to a first approximation, as referring to positions in valence-activation space. Moreover this representation is useful for capturing the continuous change in emotional expression during a piece of music.

Identifying the centre as a natural origin has several implications. Emotional strength can be measured as the distance from the origin to a given point in activation-evaluation space. The concept of a full-blown emotion can then be translated roughly as a state where emotional strength has passed a certain limit. An interesting implication is that strong emotions are more sharply distinct from each other than weaker emotions with the same emotional orientation.

Valence-Arousal space is a surprisingly powerful device, and it has been increasingly used in computationally oriented research). It has to be emphasized, however, that representations of that kind depend on collapsing the structured, high-dimensional space of possible emotional states into a homogeneous space of two dimensions. There is inevitably loss of information and, worse still, different ways of making the collapse lead to substantially different results. A problem with this approach is that specifying the



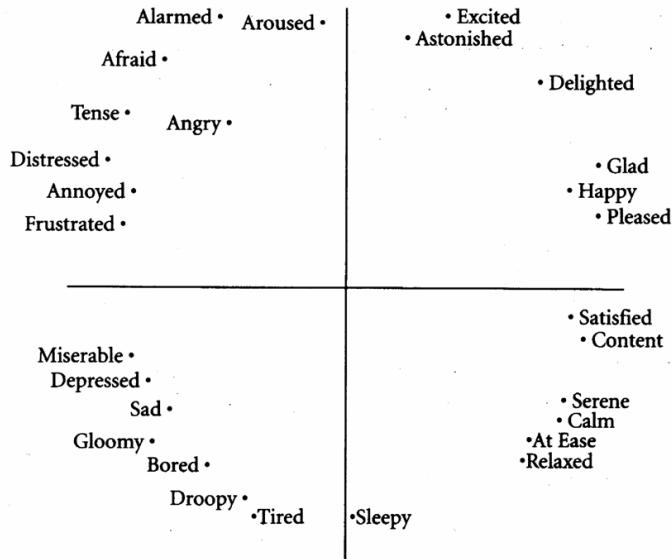


Figure 8.5: Dimensional representation of emotions: the circumplex model of Russel also called Valence-Arousal space. The horizontal axis represents the Valence dimension, while the vertical axis represents the Arousal dimension.

quality of a feeling only in terms of valence and activation does not allow a very high degree of differentiation - qualitatively rather different states can be close neighbours in valence-activation space (e.g., panic fear and hot anger). This is particularly important in research on music, where one may expect a somewhat reduced range of both the unpleasantness and the activation of the states produced by music. In consequence, adopting a valence by activation approach, asking listeners to rate their state on these two dimensions, may not allow a very fine-grained separation of the emotional effects of different pieces of music.

However, just because this representation is useful in affective computing does not imply that all emotions are continuously valenced. Neither does successful representation with a small set of discrete emotions imply that emotions are discrete, or that there is only a small set of them. Both representations have uses and limitations. The fact that both yield a concise representation is an advantage. Even if these are later found to be an oversimplification, they at least form a good starting point to begin modelling effort.

8.3.2 Laban Theory of Movement

Some of the key concepts we use in our exploration of human-motion intention are taken from Rudolf Laban (1963). In his theory of effort, he notes the dynamic nature of movement and the relationships between movement, space, and time. Laban's approach is an attempt to describe, in a formalized way, the characteristics of human movement without focusing on a particular kind of movement or dance expression. Effort-theory principles can be applied to dance and to everyday work practices.

Effort. At the center of Laban theory is the concept of effort, a property of movement. From an engineering point of view, we can consider it with a vector of parameters that identifies the quality of a movement performance. The most important concept is a description of the quality of movement.

Laban theory of effort is not concerned with degrees of joint rotation or moment directly, but it considers movement as a communication media and tries to extract parameters related to its expressive power.

The effort vector can be regarded as having four components generating a four-dimensional *Effort space* whose axes are Space, Time, Weight, and Flow. During a movement performance such effort vector moves in the effort space. Laban investigates the possible paths followed by the vector and the expressive intentions that may be associated with them. Therefore, variations of effort during the movement performance should be studied. Each effort component is measured on a bipolar scale, the extreme values of which represent opposite qualities along each axis (Fig. 6.6). The dimensions of Space and Time demonstrated to be particular suitable in the analysis of human full-body movement.

Axes	Indulging Effort	Fighting Effort
Space	Flexible	Direct
Time	Sustained	Quick
Weight	Light	Strong
Flow	Free	Bound

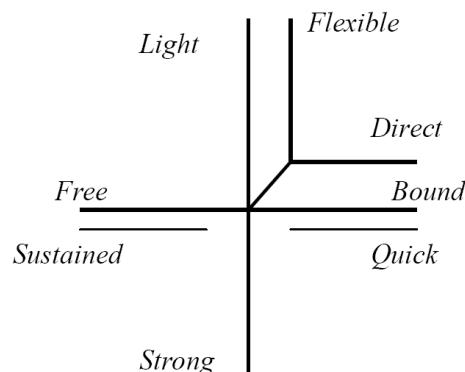


Figure 8.6: *Effort space with four components Space, Time, Weight, and Flow. Each effort component is measured on a bipolar scale, the extreme values of which represent opposite qualities along each axis.*

Space. Space refers to the direction of a motion stroke and to the path followed by a sequence of strokes (i.e., a sequence of directions). If the movement follows these directions smoothly the space component is considered to be *Flexible*, whereas if it follows them along a straight trajectory the space component is *Direct*.

Time. Time is related to impulsiveness and capacity of controlling a movement. Time is also considered with respect to a bipolar representation: an action can be *Sustained* or *Quick*, which allows the binary description of the time component of the effort space. Moreover, in a sequence of movements, each of them has a given duration in time: the ratio of the durations of subsequent movements gives the time-rhythm, as in a music score and performance.

Weight. Weight is a measure of how much strength and weight is exerted in a movement. It can be *Light* or *Strong*. For example, in pushing away a heavy object it is necessary to use a strong weight, whereas in handling a delicate and light object, the weight component has to be light.

Flow. Flow is a measure of how *Free* or *Bound* a movement, or a sequence of movements, appears. Movements performed with a high degree of bound flow reveal the readiness of the moving person to stop at any moment in order to readjust the effort if it proves to be wrong, or endangers success. In movements done with free (fluent) flow, a total lack of control or abandon become visible, in which the ability to stop is considered inessential.

The three components Space, Time and Weight define eight basic efforts (Fig. 6.7 left). In Fig. 6.7 (right), each corner represent a basic effort. Those connected by lines have two elements in common and differ, therefore, in one element only, which is indicated by the letters inserted in the connecting line.



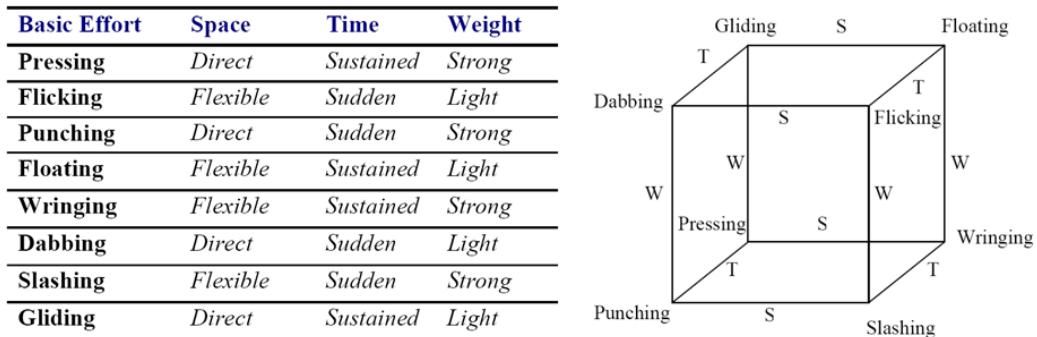


Figure 8.7: Basic efforts: (left) components; (right) relationship between basic efforts.

Laban did never explicitly associate any Effort component or quality to higher-level expressive intentions or emotional states, though he considered this plausible. Thus, investigation of Effort qualities should be considered as a first step toward a higher-level characterization of gesture. In other words, the Effort theory provides an intermediate-level description of gesture qualities that can be used as the basis for a further classification e.g., in terms of emotions or expressive intentions, grounded on results and models from psychological research. Nevertheless, such Effort qualities can already provide a characterization of expressiveness of 2D gesture e.g., in terms of hesitation, that can be fruitfully exploited in concrete applications (Fig. 6.7).

8.3.3 Sensory expressive intention representation

8.3.3.1 Musical expression

In music, the communication chain between the transmitter and the receiver contains a varying number of elements, depending on the musical repertory. In tonal western music it includes: the composer, the score, the performer, the acoustic signal, and the listener; in electro-acoustic music the composer can work directly with the sound, using an adequate support (typically magnetic tape), thus bypassing the performer. On the other hand, in improvised music, and this is particularly true for the jazz repertory, the role of the composer is merged with that of the performer, so a written score is not needed.

Few studies have investigated all the elements of musical communication empirically; in this section, the attention is focused on the communication between the performer's intentions and the listener's experience, with special regard to communication of expressive content like sensations or emotions. Music can express emotions in many different ways: emotions can be linked to a particular situation, can be generated by deviations from the listener's expectations or reflect the emotional status of the performer and of the listener. Obviously these categories are not mutually exclusive: a generic musical event can involve more than one of these possibilities.

Deviations in timing, in dynamics, in timbre, in tempo, which are not written in the score, are always introduced by the performer; they generally differ according to the musical genre, to the instrument used as well as according to the performer. There are implicit rules linked to different musical genres, which are usually transmitted orally and used in the instrumental practice; moreover, the notation in the score can sensibly vary in different musical genres and historical periods. However, even if the same score is used, different players can produce considerably differing performances. It was demonstrated that different performances of the same piece can communicate different expressive intentions. Most music performances would involve some intention (*expressive intention*) from the performer's side regarding what the music should express to the listeners. Consequently, interpretation involves assigning some

kind of meaning to the music.

In order to conceptualize expressive intentions, there are two different approaches: categorical and dimensional. The former assumes that people experience expressive intentions as categories that are distinct from each other. It is used typically in emotion research to categorize basic emotions from which all other emotional states can be derived. Dimensional approach focuses on the identification of expressive intentions based on their locations in a low-dimensional space. The categorical representations may apply to basic emotions well, but much less so elsewhere. In particular, expressive intention is a broad concept that comprehends emotions, but also sensory or metaphoric aspects. Labels as such are very poor descriptions. Moreover, in the music performance domain, there are different strategies to communicate the performance intention, depending on instruments, schools, music genre, and so on.

8.3.3.2 Kinetics-Energy space

In this section² a low-dimensional space for the understanding of musical expressive intentions, in particular inspired by sensory adjectives, is presented. The methodology, used in order to reach the understanding of expressive content, can be schematized as in Fig. 6.8.

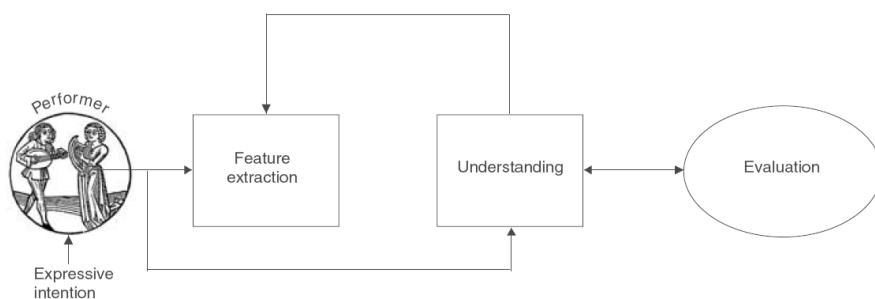


Figure 8.8: Method to understand the expressive content.

In studying music performance, we focused on expressive intentions described by sensorial adjectives, which are frequently used in music performance, i.e. light, heavy, soft, hard, bright and dark and as such, each had its opposite (soft vs. hard) so as to provoke contrasting performances by the musician. Some performances, played according to the different expressive intentions, were evaluated in listening experiments. Fig. 6.9 shows the resulting positions of the evaluation adjectives in the semantic space.

The step to understanding consisted in deriving a low dimensional structure by multivariate analysis of response data. Each performance was further analyzed to extract what acoustic features the performer used to achieve each expressive intentions. The acoustic features are related to the space dimensions in order to help the interpretation of the axes.

	Tempo	Legato	Intensity
Dim. 1 (<i>Kinetics</i>)	0.65	-0.28	-0.25
Dim. 2 (<i>Energy</i>)	0.33	-0.72	0.73

Table 8.2: Correlation between coordinate axes and acoustic parameters.

Factor analysis, using the performances as variables, found a two dimensional space (Fig. 6.10). The first factor is characterized by bright-light vs. heavy performances, the second one by soft vs. hard

²adapted from Canazza et al. JNMR 2003



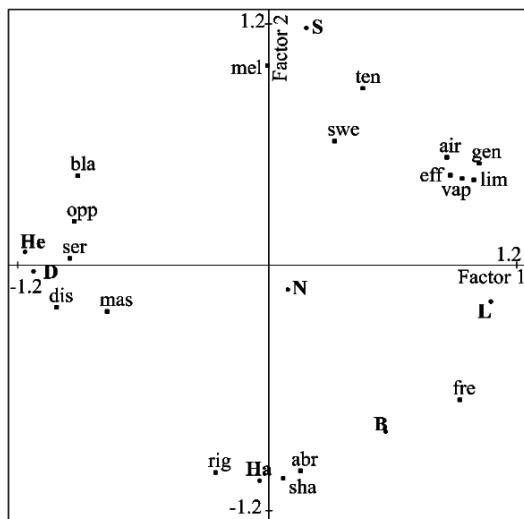


Figure 8.9: Position of the evaluation adjectives in the semantic space. Evaluation adjectives: black (nero), oppressive (greve), serious (grave), dismal (tetro), massive (massiccio), rigid (rigido), mellow (soffice), tender (tenero), sweet (dolce), limpid (limpido), airy (aereo), gentle (lieve), effervescent (spumeggiante), vaporous (vaporoso), fresh (fresco), abrupt (brusco), sharp (netto)

performances. Acoustical analysis of the performances showed that first factor is closely correlated with Tempo and can be interpreted as *Kinetics* (or Velocity) factor; while the second one is related to Attack Time, Legato/Staccato, Intensity and can be interpreted as *Energy* factor. By means of legato/staccato, we intend how much the notes are detached and distinctly separated, namely as the ratio between the duration of a given note (i.e., the measure of time between note-onset and note-offset) and the inter-onset interval (i.e., the measure of time between two consecutive note onsets) which occurs between its subsequent note. We can use this interpretation of Kinetics-Energy space as an indication of how listeners organized the performances in their own minds. The robustness of this mid-level representation, which we call *Kinetics-Energy space*, was confirmed by synthesizing different and varying expressive intentions in a musical performance.

Performance Worm. A representation based on similar concepts was proposed by Dixon (2002). The *PerformanceWorm* is a real time system for tracking and visualizing in a plane the tempo and dynamics of a performance and provides insight into the expressive patterns applied by skilled artists. Figure 6.11 shows a snapshot of the Worm as it tracks a performance of Rachmaninov's Prelude op.23 no.6 played by Vladimir Ashkenazy, bar 1-30. The horizontal axis represents tempo in beats per minute and vertical axis dynamics (loudness) in decibel. The darkest point represents the current instant, while instants further in the past appear fainter. In the example, we see an initial accelerando, followed by a simultaneous crescendo and ritardando followed by phases of accelerando-crescendo and rallentando-diminuendo. Observing trajectories such as in Figure 6.11, many interesting patterns emerge that reveal characteristics of performances. This representation also forms the basis for automatic recognition of performers' style. Indeed skilled musicians communicate high level information such as musical structure and emotion when they shape the music by the continuous modulation of aspects such as tempo and loudness.

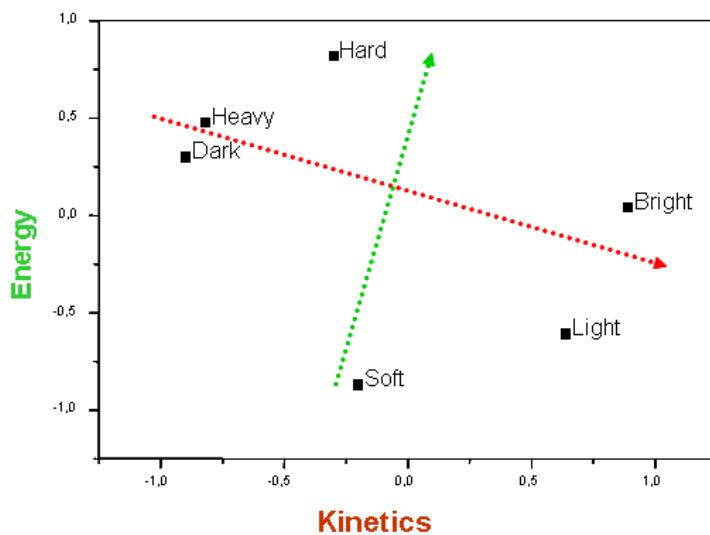


Figure 8.10: Kinetics-Energy space, as mid-level representation of expressive intentions.

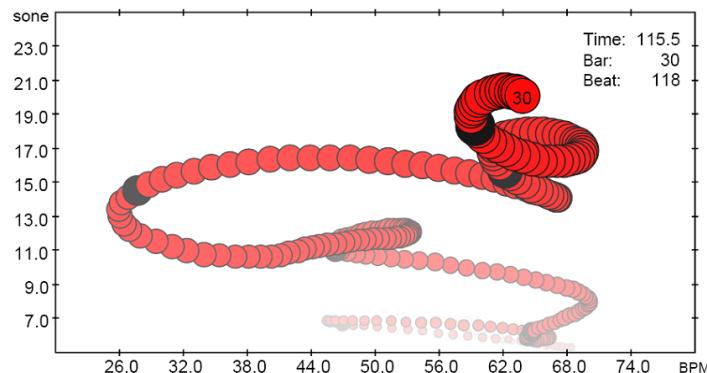


Figure 8.11: Screen shot of the PerformanceWorm, showing an expression trajectory with horizontal axis tempo in beats per minute and vertical axis dynamics (loudness) in decibel. The darkest point represents the current instant, while instants further in the past appear fainter.

8.3.4 An action based metaphor

The concept of expression is common to different modalities: one can speak of expression in speech, in music, in movement, in dance, in touch, and for each of these contexts the word expression can assume different meanings; this is the reason why expression is an ill-defined concept. In some contexts expression refers to gestures that sound natural (human-like), as opposed to mechanical gestures. In other contexts, expression refers to different qualities of natural actions, meaning that gestures can be performed following different expressive intentions which can be related to sensorial or affective characteristics. This level of expression has a strong impact on non verbal communication, and have led to interesting multimedia applications and to the development of new types of human-computer interfaces. Technology-mediated music access is more and more becoming an interactive process, involving non linguistic communication and action based modalities.

Almost all new mobile devices implement interfaces for music content access. While tools for music content on PC technology make mostly use of a traditional monitor and mouse interface, which allows a linguistic based interaction, the reduced dimension of the mobile devices requires novel interaction

modalities: recent mobile devices utilize touch sensitive screens, able to track different kinds of gestures. Can gestures be related to music experience, allowing for a more direct access to music content?

The last generation of video-game consoles are also powerful players for music content, with the peculiarity to put at the disposal of the users a set of devices that make possible a gestural interaction with music. Video-games such as Guitar Hero or Rockband allow the user to actively participate in the music production process by playing simplified models of musical instruments or moving their hand over an invisible guitar. This kind of applications opens new scenarios for music production and access, so that famous rock bands are working on versions of their music especially arranged for these devices. Relevant to our aims, is that music access is more and more becoming an interactive process, involving non linguistic communication and action based modalities. This fact places new challenges to the information technology field. In many cases, the technology is an obstacle that makes access to music difficult for many user. What is needed is an effective mediation technology based on content and experience, capable of sensing and responding appropriately to the user requests; a transparent mediation technology that relates musical involvement directly to sound.

Transparent technology should thereby give a feeling of non-mediation, a feeling that the mediation technology disappears when it is used. Such a technology would then act as a natural mediator for search-and-retrieval purposes as well as for interactive music-making [Leman, 2007]. Transparent technology requires a better understanding of the musical experience and how this experience can be described. The power of music to induce in the listeners different affective states or moods is a well known characteristic, but music experience is not evidently limited to emotions. The musical experience is a very complex issue, that can be described in manifold ways. Linguistic descriptions can capture only partial aspects of the musical experience and non linguistic metaphors should be used to represent other features, that can not be verbally conveyed. In particular, non linguistic modalities are probably more suited to represent non rational aspects of the human factors, which can be very important to determine the success of a technological product.

There are numerous ways to describe music expressiveness, which do not exclude each others, but rather they are complementary points of view of the same complex experience. Each way of description is a metaphor which allows to represent particular aspects of the musical experience, without totally representing that experience. Moreover, the different metaphors can be suitable to different applicative contexts: e.g., the emotional aspects of musical experience can be useful in affective human-computer interfaces; whereas gestural aspects can be useful in direct interaction and manipulation of contents. In this section we propose an *action based metaphor* as a way to describe some aspects of the musical experience, as other metaphors does, and we will put in evidence relationships between this metaphor and the emotional and the sensorial ones.

8.3.4.1 Similarities in the feature space

In [Mion and De Poli 2008] we addressed the question of whether expressive information can be communicated (and recognized) by means of features which are not strictly related to the score. Thus, relevant musical attributes for differentiating expressions (such as articulation) can be replaced by more physical features (e.g. the attack time). With the aid of machine learning techniques we found the audio features that are most relevant for the recognition of different expressive intentions (see sect. 6.4.1.1). Using these features as coordinates, we could place the expressions on a *feature space* and obtain an objective measure of physical similarity.

Normally affective and sensorial domains are studied separately. In order to study the expressive content conveyed by the performers from a more general point of view, we took into account both spaces by using two pairs of opposite labels to indicate the dimensions for each space. Regarding the affective space (see sect. 6.3.1), the categories Happy-Sad (High and Low Valence), Angry-Calm (High and



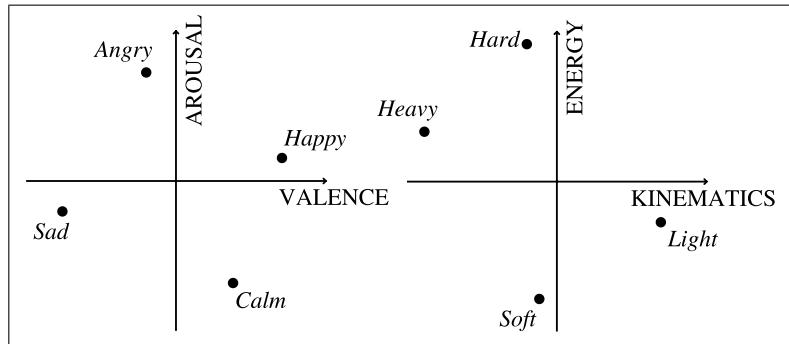


Figure 8.12: The Valence-Arousal space (left) and the Kinematics-Energy space (right), respectively, and placement of expressive intentions used in our experiments.

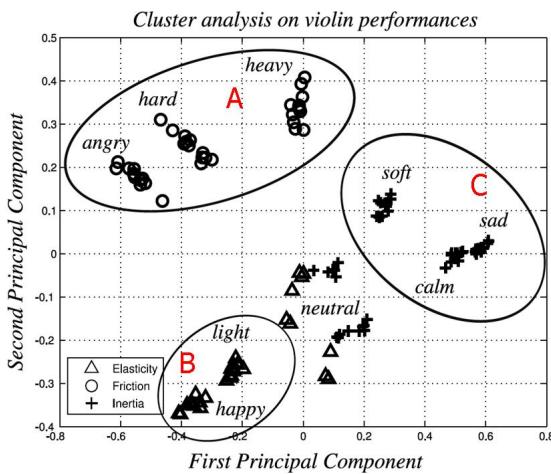


Figure 8.13: Feature space obtained by Principal Component Analysis of the features of expressive violin performances, according to adjectives from both affective and sensorial spaces (Fig. 6.12).

Low Arousal) represent the bipolarity induced by independent dimensions valence and arousal; for the sensorial space (see sect. 6.3.3), we have the correspondences Hard-Soft (High and Low Energy) and Light-Heavy (High and Low Kinetics). In this way each adjective has its opposite in order to deliberately induce contrasting performances by the musician. Fig. 6.12 shows the placement of expressive intentions within the used spaces. Beyond the pair of adjectives representing the bipolarities of the spaces, we considered a Neutral performance as well, which listeners placed between the pair of opposite adjectives. By neutral we mean a human performance without any specific expressive intention and stylistic choice.

In order to understand how expressive performances represented by the selected features are projected (and clustered) on a low-dimensional space, we applied Principal Component Analysis (PCA) and we used the k -means algorithm for unsupervised clustering of performances. Fig. 6.13 depicts the PCA projection on a joint 2D space and the cluster analysis of violin performances. The cluster analysis shows that three main clusters emerge in the feature space:

- A. Hard/Heavy/Angry
- B. Light/Happy
- C. Sad/Calm/Soft

A very similar behaviour characterized the projection of the performances from other instruments. These clusters derive from similarities of physical characteristics in the feature space.

8.3.4.2 Similarities in the perceptual space.

Since these expressive intentions are similar according to the features used for the recognition, and since recognition implies subjective evaluation, then we hypothesized that expressive intentions are similar also from a perceptual point of view. In order to verify if these similarities have also a perceptual basis (i.e., they are considered similar also by the listeners), we conducted an experiments to derive a perceptual description. We were interested in understanding which expressions are clustered together, in order to derive common evaluation criteria adopted by listeners. To this purpose, we conducted a listening experiment, in which subjects were allowed to group the expressive intentions according to their preference in order to avoid any influence on expected resulting clustering. A spatial representation (*perceptual*

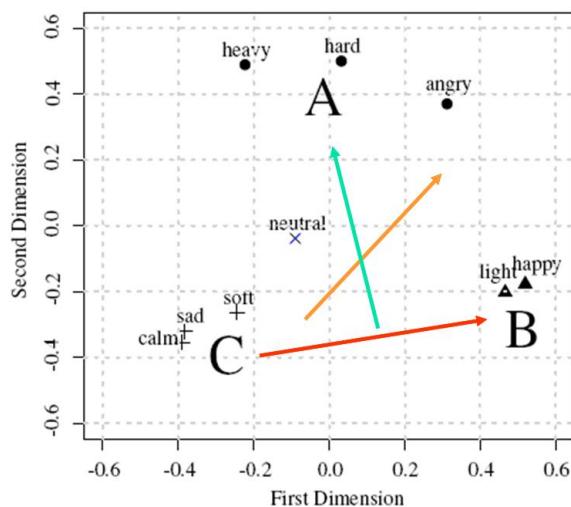


Figure 8.14: Perceptual space resulting from the listening experiment.

space) of the expressive intentions was obtained by multidimensional scaling (MDS) and cluster analysis (see fig. 6.14). The k -means algorithm was applied to the coordinate axes of each expressive intention and three stable groups were identified, the same that were found in the feature space:

- A. Hard/Heavy/Angry
- B. Light/Happy
- C. Sad/Calm/Soft

By observing the relations of acoustic parameters with each dimension, we can derive an interpretation of the meaning of the obtained dimensions. Then, we hypothesize an interpretation at an higher level of abstraction: we take into account both affective and sensorial expressions and we observe how expressions are clustered according to our analysis.

Correlation with acoustic parameters The coordinate axes of the nine expressive intentions along the two dimensions of the perceptual space were correlated to the relevant acoustical features described in Sec. 6.4.1.1. Correlation coefficients are reported in Table 6.3, showing that dimension 1 is strongly

	NPS	PSL	A	R	SRa	REh
Dim. 1	0.85*	0.66	-0.94**	0.8*	0.42	-0.41
Dim. 2	0.17	0.91**	-0.5	0.74	0.68	0.46

Table 8.3: Correlation between coordinate axes (INDSCAL - Group Space) and acoustic parameters (* $0.005 < p < 0.01$, ** $p < 0.005$).

correlated with Attack Time, Note per Second and Roughness ($r=0.94, 0.85, 0.8$), dimension 2 with Peak Sound Level ($r=0.91$).

By means of the correlation coefficients we can also derive additional considerations after observing the theoretical spaces that we used to derive the expressive intentions and the perceptual space (see Fig. 6.12 and Fig. 6.14). Dimension 1 (strongly correlated to *NPS*, *A* and *R*) separates expressive intentions Heavy to Light, and Happy to Sad. These expressions, in the theoretical spaces, were respectively opposed by Kinetics and Valence qualities. Features correlated to dimension 1 are related to qualitative properties of performance articulation such as number of events (*NPS*) and time of attack (*A*). On the other side, dimension 2 (correlated to Peak Sound Level) separates intentions related to Energy and Activity (e.g., Hard/Soft), thus dimension 2 can be considered as related to energetic quantities. Moreover, texture properties are expressed by Roughness, which is reasonably correlated to both dimensions since it has both energetic and qualitative properties, being related to the sensation of effort. Indeed, along the diagonal we can find expressions Angry/Calm in opposition as can be expected, since intuitively these adjectives are in opposition both according to energy and quality. In a way, we might argue that the resulting joint space can be expressed by valence-energy dimensions.

We can also exploit the analysis on selected features and the results on perceptual tests to see how expressive intentions are clustered and jointly organized in the semantic space. We also derive a semantic interpretation and possible association among affective and sensorial labels in music performances.

In Fig. 6.14 the clusters are indicated with letters A, B and C. According to the discussion reported above, we can see that two main oppositions clearly emerge in relation to the dimensions. Along dimension 2, cluster A is in opposition to clusters B and C. This opposition expresses a different behaviour for Hard/Heavy/Angry with respect to all other expressions in terms of Energy. This opposition is found both in the features (see Fig. 6.13) and perceptual spaces (see Fig. 6.14). Moreover, dimension 1 places clusters B and C at the opposite sides. According to the acoustic parameters correlated to this dimension, we can argue that Sad/Calm/Soft are in opposition to Light/Happy in terms of sound quality and performance articulation (e.g., in terms of kinematics, attack time, and roughness). Thus, we are induced to use two main criteria of interpretation: one based on energy (quantitative opposition) and another based on quality.

8.3.4.3 Toward an action based interpretation of expressive intentions

During listening tests we conducted interviews to participants, who were asked to verbally describe the performances using words at their will. People often used metaphors based on action or gesture (e.g., resistive or springing) to qualitatively describe the expression. Also, in the literature we can find examples of formalistic view of music where the content of music is tonally moving forms rather than feelings. Expressive intentions can be described by using adjectives as well as by the corresponding actions and gestures conveying such expression. The usage of affective and sensorial adjectives induce the listener to an evaluation strategy which is based on different perspective with respect to the original domains.

At this intermediate level, we found appealing an intuitive interpretation based on the *action metaphor*,



that is on the analogy of a human acting on a physical body. Linear mechanics tell us that we can distinguish three kinds of behaviour of an elementary object subjected to action from an external source, and we have three clusters determined by acoustical and perceptual similarities. Thus, we are induced to speculatively find a correspondences among the two groups, i.e. to conjecture a description of expression based on *action*, and an interpretation of the clusters based on physical analogy.

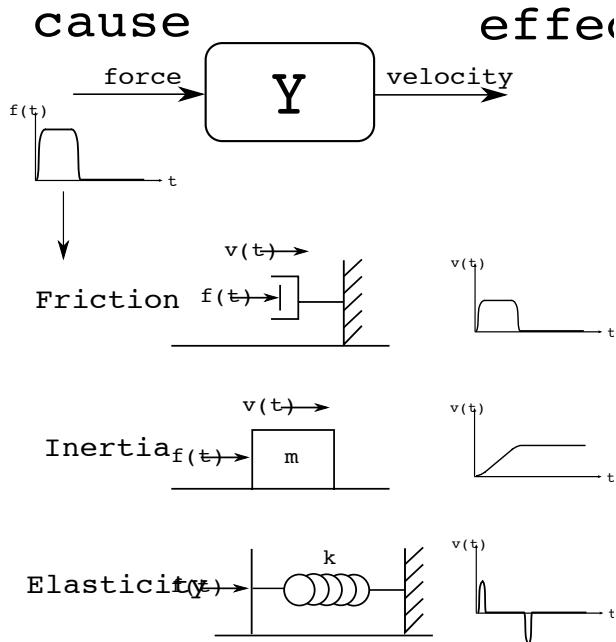


Figure 8.15: Behaviour of the basic linear mechanical systems: friction, inertia, elasticity.

When we act on a physical object, force is often subjectively considered as the cause and movement as the effect. The cause-effect relation is represented by the admittance Y which mathematically describes the dynamic mapping and the qualitative behaviour from force to velocity by an integral-differential equation. We can distinguish resistive admittance, which dissipates energy, from reactive impedance, which stores energy. In linear mechanical systems three elementary relations define the fundamental quantities friction, inertia and elasticity. Ideal friction is a pure resistive admittance, while ideal inertia and elasticity are pure reactive admittances: in particular inertia stores kinematics energy and it opposes changes in movement, while elasticity stores potential energy and opposes changes in force. In general the admittance is composed by a resistive part and a reactive part.

In order to have an intuitive idea of their behaviour, in Fig. 6.15 we represent the output (velocity $v(t)$) from the three basic elements when a smoothed large force pulse $f(t)$ is presented at their input. It can be seen that friction acts as a scaling factor of the input force and does not modify the shape of the input. The inertia (mass) tends to remain at its initial velocity, which is zero in the present example. Then it grows progressively and remains constant when the input stops; the mass progressively augments its kinetics energy. The elasticity (spring) instead reacts immediately to the input variations; it stores potential energy which is used to oppose to force changes.

Thus we propose an interpretative metaphor for expression based on the concept of action in the environment, and in particular based on the reactive behaviour of the three basic mechanical elements: friction, inertia and elasticity.

Associating expressiveness to action. From the qualitative description of the behaviour of the three basic elements we are induced to associate *friction* to the cluster Hard/Heavy/Angry, *inertia* to cluster Sad/Calm/Soft, and *elasticity* to cluster Light/Happy. In order to verify if the action based metaphor can appropriately describe some expressive characteristics of music content, we conducted two experiments which investigates subject's associations between two sets of musical stimuli and the actions on three haptic attractors, that we assumed to be representatives of the three components of the *KID* metaphor.

The set of attractors, representing different prototypes of actions, is composed by three haptic stimuli synthesized by means of a Phantom Omni haptic device³, which simulates the basic effect of a mechanical mass–spring–damper system. All the force feedback are omni-directional: the device reacts to the user's input in every points of the haptic sphere. Regarding the stimulus E (*elasticity*), the device generates a force feedback with intensity

$$f_E(t) = -K_{el} \cdot \|s(t) - s_0\| \quad (8.2)$$

where s_0 is the center of the haptic sphere, $s(t)$ is the position of the stylus at the instant t and K_{el} is the elasticity constant of the system. The stimulus F (*friction*) is characterized by a force feedback proportional to the velocity of the user's movement:

$$f_F(t) = -\eta_v \cdot v(t) \quad (8.3)$$

where $v(t)$ is the velocity of the stylus and η_v is the viscosity constant of the system. Finally, the stimulus I (*inertia*) simulates the interaction with an inertial mass m , moving in a field free of other (gravitas or magnetic) forces. The mass m is coupled to the stick, that we assume to have a negligible inertial mass. The intensity of the force follow the equation:

$$f_I(t) = -m \cdot a(t) \quad (8.4)$$

where $a(t)$ is the stylus acceleration. After several tests, we set $m = 0.5$ Kg, $K_{el} = 510$ N/m, and $\eta_v = 31.9$ Ns/m.

In the experiments, participants were asked to listen to each musical excerpt and to associate it to one of the three attractors. Participants were allowed to listen to the excerpts and to test the attractors as many time as wished, and to change their responses until they were satisfied by their choices. The attractors induced the listeners to organize the musical excerpts on the basis of similarity criteria which depend on the characteristics of the attractors. Therefore, we can expect that different set of attractors will induce a different mental organization of musical excerpts. On the other hand, the features of the set of music excerpts can influence the results as well.

In the **first experiment** as stimuli expressive performances of simple melodies, played according the affective and sensorial adjectives previously seen, were used. Subjects' responses were summarized into a two-way contingency table containing 9 rows (expressive intentions) and 3 columns (attractors Friction, Elasticity, and Inertia).

We conducted two analysis to investigate the association between expressive intentions and attractors: a Simple Correspondence Analysis and a k -means clustering. The contingency table was submitted to Simple Correspondence Analysis in order to graphically represent the degree of association between expressive intentions and attractors according to their χ^2 distances. We can see in Fig. 6.16 that the expressive intentions Angry-Hard-Heavy, Happy-Light, Calm-Sad-Soft are depicted close to attractors Friction, Elasticity and Inertia respectively. Then, we applied the k -means clustering (number of groups = 3) to the coordinates of the points in the Correspondence Plot in order to identify the stable groups. We did not take into account the Neutral intention. Three stable groups (see dashed lines in Fig. 6.16)

³<http://www.sensable.com/haptic-phantom-omni.htm>



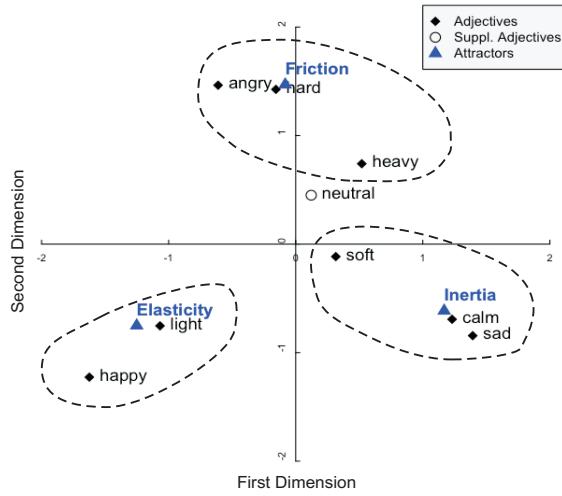


Figure 8.16: Correspondence analysis on experiment with expressive performances. Dashed lines represent the outcome of the cluster analysis.

were identified corresponding to the clusters: (A) Angry-Hard-Heavy, (B) Happy-Light and (C) Calm-Sad-Soft. By grouping these expressive intentions according their cluster membership we found strong relation among clusters and attractors. Moreover, significant relation has been found between A cluster and friction attractor, B cluster and elasticity attractor and C cluster and inertia attractor. It can be noticed that these clusters are consistent with the clusters found in the previous experiments.

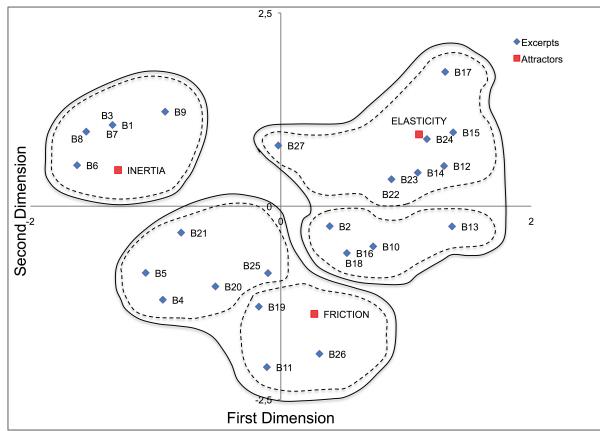


Figure 8.17: Correspondence analysis on experiment with Bigand musical excerpt. Dashed and continuous lines represent the outcome of the cluster analysis (with number of groups equal to 3 and 6, respectively).

In the **second experiment** we used as stimuli the music excerpts, extracted by recordings belonging to the Western music repertoire of the classic-romantic period, used by Bigand (2005) to study the emotion communication in music and derive the Valence/Arousal space.

The contingency table was submitted to Simple Correspondence Analysis in order to graphically represent the degree of association between musical stimuli and attractors (see Fig. 6.17). Then, we proceeded with a k -means analysis in order to identify clusters of stimuli. After several trials, we set the number of groups both to 3 and to 5 (respectively continuous and dashed lines in Fig. 6.17). Three of the

five clusters include those stimuli which were associated to one of the haptic attractors. The other two clusters are composed by stimuli that subjects associated equally to two attractors: Elasticity - Friction and Inertia - Friction.

Discussion. From these two experiments, we can conclude that in general the subjects were able to consistently recognize common characteristics between musical stimuli and haptic attractors. Concerning the single expressive intentions, Neutral intention was not recognized as related to one single attractor, but the contingency table shows a balanced contribution of all the three attractors, as we could expect due to its meaning. It is interesting to note that, in some cases, the scores in the contingency table suggest the idea the three attractors Friction, Elasticity, and Inertia constitute a sort of basic components; the various expressive nuances can be represented as a combination of these components. E.g., Heavy performance was perceived as related not only to Friction, but also to Inertia, as we could expect.

The results of the experiment with musical excerpts stimuli support a relation between the high Valence - high Arousal (Happy) cluster and the Elasticity attractor (confirmed by all the excerpts except for B10 and B11), and between the low Valence -low Arousal (Sad) cluster and the Inertia attractor. On other cases, subjects responses are divided between two attractors: e.g., excerpts B4, B5, B20, and B21 are associated both to Inertia and Friction. This observation is coherent with the Fig. 6.17, where two clusters are composed by stimuli that subjects associated to two attractors: Elasticity-Friction and Inertia-Friction.

Comparing results of the two experiments, we can note that, although subjects are able to recognize the different haptic feedback, the action based metaphor seems to be more suitable for representing expressive cues in expressive performances, where the expressive content is mainly related to performance cues, than in complex musical excerpts, where musical structure is more relevant. This result can be explained by the fact that music performance is more related to action-based aspects, whereas musical structure can involve aspects related to cognitive and/or cultural factors. In particular, the action metaphor seems to be not able to represent difference in the dimension of valence: for example between a sad and a calm excerpt. Results of experimental studies on music and emotions have already shown that valence is related to musical structure, in particular major-minor modality.

Friction, elasticity and inertia are the basic properties of ideal mechanical systems and the dynamic behaviour of each real system depends on a weighted combination of friction and elasticity or of friction and inertia, where friction represents the quantitative aspect of the dynamical behaviour and elasticity/inertia represents the qualitative aspect of dynamical behaviour. Likewise, the results of the experiments let us hypothesize that the expressive characteristics of musical excerpts can be associated with a weighted combination of quantitative and qualitative basic components.

8.4 Recognition of expression

This section presents how expression can be analyzed and detected by an affective interface. The process of extracting expression is illustrated in Fig. 6.18. First sensors measures some signals. Then this signal is processed and a number of features are extracted. These features can be seen as a compact representation of the input signal. Based on these features the expression in the input signal, if any, is recognised and converted into a symbol. When a system is running it outputs a string of symbols which can then be used to control music production or something completely different.



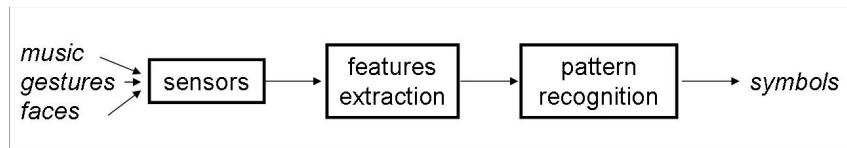


Figure 8.18: The process of extracting expression.

8.4.1 Recognition of expression in music performance

During a music performance, the musician adds and varies expressiveness to the musical message changing the timing, the dynamics, and the timbre of the musical events at his will to communicate an *expressive intention*. In particular, the same piece of music can be performed according to different interpretations by introducing deviations from the nominal value of musical parameters, as written on the score. The analysis of these systematic deviations led to the formulation of several models that try to describe the way performers convey the expressions. Up to now, many efforts have been spent for developing expressive rendering systems able to simulate such a human skill (e.g. SEE), but few steps have been done concerning the automatic analysis of expressiveness.

Understanding expression communication through the auditory modality aims to adapt the multi-media technologies to the basic forms of human communication. The spreading of systems for the musical document indexing used by search engines, as well as the development of standards for the fruition of multimedia contents such as MPEG-7 can benefit from the communication based on expressive paradigms. Musical data in particular suffers from the poor textual description traditionally retrieved by today's search engines. This lack implies the necessity of search engines that work at a higher level of abstraction. In particular, we believe that the understanding of the mechanisms of expressive communication by a music performer can help us comprehend how to retrieve expressive content on audio data, and then to design a next generation of search engines for Music Information Retrieval (MIR).

Beyond the potential applications to the MIR field, other application fields can be forecast. Analysis of music expression can lead to design synthesized expressive sounds which can be combined with real stimulus to experience augmented reality, and they can be used to provide navigational information supporting users exploration through virtual expressive environments. On the other side we can address issues related both for artistic and functional applications. In the case of artistic applications, the artist's aesthetic sensibility can drive the synthesis toward scenarios like sounding physical objects, controlling in real time the expressive information by tactile interaction, gestural controllers or methods for mapping and transforming audio data to create sound material. Functional applications can be derived: e.g. in the medical-therapeutic field the audio feedback can be reinforced according to the motor gestures, in order to be used for therapy monitoring. Expression can also be added to systems for generation and manipulation of auditory icons (non-speech sounds) like alarm enunciators, using the expression to alter upwards or downwards the perceived urgency as the situation demands.

Most models for analysis of expression are based on modelling the measured deviations in human performances. These works use similar approaches based on the knowledge of the score. However, the use of a score as reference has some drawbacks because an audio performance can communicate expression even if it is not based on a score: let us think for instance to a musical improvisation or to many non-western musical forms. In this section⁴ we deals with the analysis of expression of structured audio events, not measuring deviations from the score but by using machine learning techniques. We start from audio signal and we investigate the most relevant features for expression description at different levels of structural complexity (e.g. from simple sounds to music as structured organized events).

⁴adapted from Mion, DePoli 2008



8.4.1.1 Relevant features for expression recognition

With the aim of selecting features relevant for expression description, we recorded a set of expressive performances, played by professional musicians on various instruments. From these performances, we extracted a set of cues that were found to be important for discriminating different emotions in previous listening experiments. Then we applied Sequential Forward Selection (SFS) with reference to a Minimum Distance classifier to rank and select a set of relevant features. By Principal Component Analysis (PCA) on the performance data we removed correlated features and projected on a 2D space. As a result, we derived a set of features for a general description of the expressions and another one specific for each instrument. These features were tested and confirmed by the leave-one-out cross validation, and they can be grouped according to *local* audio features (using non overlapping frames of 46 ms length), and *event* features (using sliding windows with 4 s duration and 3.5 s overlap). The windows size allows to include a reasonable number of events and it corresponds roughly to the size of the echoic memory.

Among *local* features, the following were found to be relevant:

- Roughness R , which is computed by the auditory model of Leman and is considered to be a sensorial process highly related to sound texture perception (Fig. 6.19 left);
- Spectral Ratio SR_a ,

$$SR_a = \frac{\sum_{j \in LB} |X(j)|^2}{\sum_{k=1}^{N/2-1} |X(k)|^2} \quad (8.5)$$

which indicates the relative amount of energy in the low frequency band LB ($f < 1$ kHz), and is related to brightness;

- Residual Energy ratio RE_h , which describes the stochastic (noisy) energy in the high frequency band HB ($f > 1.8$ kHz), obtained by removing the sinusoidal components, and gives information on the quality of the perceived effort. RE_h can be computed by

$$RE_h = \frac{\sum_{j \in HB} |X_R[j]|^2}{\sum_{k=1}^{N/2-1} |X[k]|^2} \quad (8.6)$$

where X and X_R are the spectrum of the signal and of the stochastic component, respectively.

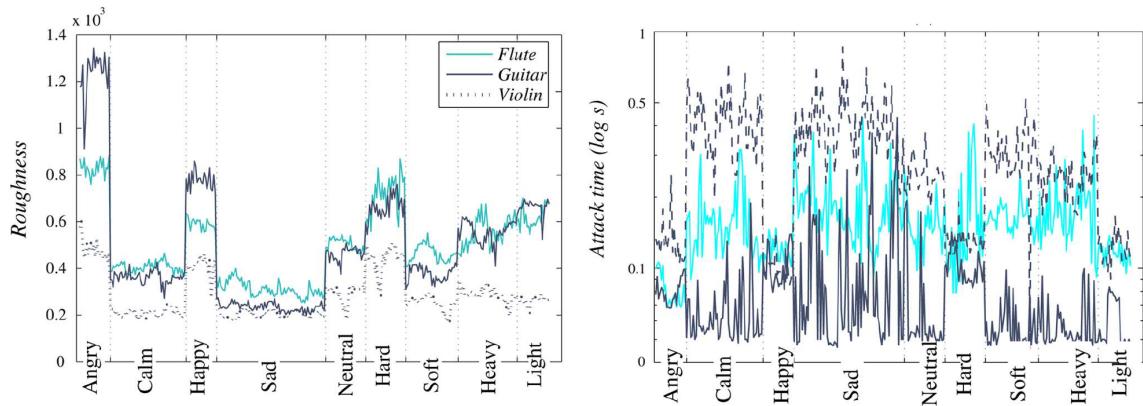


Figure 8.19: Sequence of profiles for roughness (left) and attack time (right) according to suggested adjectives performed by three instruments ("Twinkle Twinkle Little Star").

Among *event* features, the following were found to be relevant:



This book is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 license,

©2005-2018 by the authors except for paragraphs labeled as adapted from <reference>

- Peak Sound Level $PSL = \max [RMS(t)]$, where $RMS(t)$ is the temporal envelope;
- Attack time A as the time required to reach the $RMS(t)$ peak, starting from the onset instant (Fig. 6.19 right);
- Notes per Second NPS , which is computed by dividing the number of onsets by the window duration. For the computation of event features we segmented the signal by onset detection, based both on the derivative of the spectral magnitude and on pitch-tracking approach. The offset instant was detected when the temporal envelope $RMS(t)$ falls by the 60% from its previous maximal value.

8.4.1.2 From features to expressive intentions

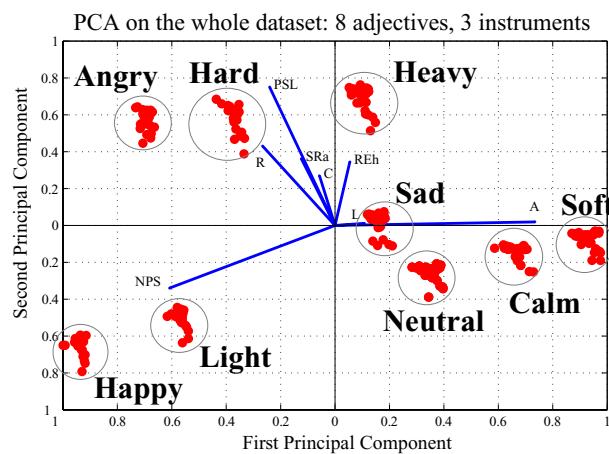


Figure 8.20: Principal Component Analysis on the whole recorded audio from three instruments, according to adjectives from both affective and sensorial spaces.

After the quantitative validation of the contribution of the features to the different expressions, we discuss the qualitative importance of our descriptors for the expressions that we are dealing with. In particular, a qualitative description of the union of the two domains was investigated, since the relation between sensorial and affective adjectives is not commonly explored. Fig. 6.20 shows how the expressions are projected into a 2D-space by the PCA analysis using the features that we accounted for the selection. Performances are projected in a way that allows us to relate the descriptors to the dimensions Energy/Kinetics and Valence/Arousal. In particular, we can see that the adjectives from the two spaces are placed along the projections of the General descriptors PSL , NPS , A , R : *Hard-Soft* and *Angry-Calm* are mainly differentiated along the descriptors PSL and R ; *Light-Heavy* and *Happy-Sad* mainly along the descriptors NPS and A .

Also, we can notice that the expressions splits into three clusters positioned in a way that induce us to associate Kinematics with Valence and Energy with Arousal. In particular, the position of the clusters reflects the intuitive correspondence of *Light* with *Happy*, *Sad* with *Calm* and *Soft*, *Hard* with *Heavy* and *Angry*.

A qualitative physical description of the expressive intentions by means of the selected features can be derived. Tab. 6.4 summarizes the qualitative contributions to the expression description. NPS , PSL , A are directly related to physical properties of the signal, and they can easily be mapped into physical description such as fast/slow, loud/weak, sudden/loose respectively. Moreover, roughness R is considered to be a sensory process highly related to texture perception, thus we can think of texture-related

Table 8.4: Qualitative description of adjectives

feature	<i>NPS</i>	<i>PSL</i>	<i>A</i>	<i>R</i>	$1 - R_a$	<i>REh</i>
	Tempo + fast - slow	Intensity + loud - weak	Attack + loose - sudden	Texture + rough - smooth	Brightness + bright - dark	Effort + strong - weak
Hard	+	+++	-	++	---	++
Soft	---	--	+++	--	++	+
Heavy	-	+++	+	++	---	+++
Light	++	-	--	-	+	--
Neutral	-	--	++	--	--	-
Happy	+++	-	---	-	+	---
Sad	-	-	+	-	+	-
Angry	++	+++	--	+++	---	++
Calm	--	--	+++	--	++	-

properties of expressions explained by the physical metaphor of rough/smooth. Feature *SRa* is related to the amount of energy in the frequency region below 1000 Hz. Thus, expressions characterized by low values of *SRa* reveal high energy in the higher bands, and this can be translated into the Brightness property, and then the adjectives can be described by means of bright/dark description. Finally, feature *REh* is related to the quality of the perceived effort, and it can be associated with physical metaphor of strong/weak.

8.4.2 Recognition of affect in speech

Research on vocal affect recognition is also largely influenced by a basic emotion theory. In turn, most of the existing efforts in this direction aim at the recognition of a subset of basic emotions from speech signals. Table 6.5 is a summary of relationships between emotion and speech parameters from a review by Murray and Arnott (1993).

	Anger	Happiness	Sadness	Fear	Disgust
Rate	Slightly faster	Faster or slower	Slightly slower	Much faster	Very much faster
Pitch Average	Very much higher	Much higher	Slightly lower	Very much higher	Very much lower
Pitch Range	Much wider	Much wider	Slightly narrower	Much wider	Slightly wider
Intensity	Higher	Higher	Lower	Normal	Lower
Voice Quality	Breathy, chest	Breathy, blaring tone	Resonant	Irregular voicing	Grumble chest tone
Pitch Changes	Abrupt on stressed	Smooth, upward inflections	Downward inflections	Normal	Wide, downward terminal inflects
Articulation	Tense	Normal	Slurring	Precise	Normal

Table 8.5: Emotions and Speech Parameters (from Murray and Arnott, 1993).

A recent trend in the research on automatic human affect recognition is the multimodal analysis of human affective behavior, including audiovisual analysis, combined linguistic and nonlinguistic analysis, and multicue visual analysis based on facial expressions, head movements, and/or body gestures. At the same time, several new challenging issues have been recognized, including the necessity of studying the temporal correlations between the different modalities (audio and visual) and between various behavioral cues (e.g., facial, head, and body gestures).

8.4.3 Expressive gestures

The study of gesture is a vast and complex field of research. Various communities explore gestures in different contexts and, not surprisingly, the definition of what a gesture is may greatly vary across these communities. *Gestures* may be considered as opposites to postures, i.e. gestures are dynamic and postures are static.

One simple way to distinguish between gestures is to organize them based upon whether or not they involve contact with a device. This approach yields two groups:

- Gestures for which no physical contact with a device or instrument is involved. These have been called *empty-handed*, free, semiotic or naked gestures.
- Gestures where some kind of physical contact takes place. These have been called *manipulative*, ergotic, haptic, or instrumental gestures.

Gestures can be analysed using three distinct approaches: descriptive, functional, and intrinsic.

Descriptive (or phenomenological) approach is based on three criteria:

- *Kinematic* criterion, consisting of an analysis of movement speed;
- *Spatial* criterion, involving the size of the space where the gesture takes place, for instance: large (arm movement) or small (finger movement);
- *Frequency range* criterion, taking into account movement decomposition regarding its frequency content, roughly between some tenths of a Hertz to 10 Hz.

Functional approach which refers to the possible functions a gesture may perform in a specific situation.

For instance, considering instrumental gestures, gestures can transmit energy to a device or modify certain of its characteristics: a performer may bow a violin in order to put the strings into vibration (an excitation gesture) or change the length of the string to change the note fundamental frequency (a parametric modification gesture).

Intrinsic approach, which focuses on the performer conditions of gesture production. For example, the hand is suitable for fine motor action and perception due to its dexterity and the density of nervous receptors at finger tips whereas the feet are more suited to the perform slower movements.

In music, empty-handed gestures are mostly associated with conductor technique (not always strictly empty-handed, since conductors usually hold a baton with the right hand). Conductor gestures are well defined and both hands have their own musical roles, primarily tempo keeping with the right hand and loudness control and other expressive cues with the left hand. Gestures related to instrument performance may be analyzed in at least three levels, from a purely functional to a purely symbolic one:

- *Effective* gestures, those that actually produce the sound;
- *Accompanying* gestures, body movements such as shoulder or head movements;

- *Figurative* gestures, which are perceived by a listener, but without a direct correspondence to a movement of the performer. Examples would be changes in note articulation, melodic variations, etc..

From functional point of view, musical gestures of the performer can be classified as

- sound producing gestures, which effectively produce sound, e.g. hitting, stroking, blowing, bowing;
- sound facilitating and accompanying gestures, which support sound producing gestures and follow the music;
- communicative gestures, which are used to communicate with other performers in an ensemble, or that have more theatrical functions for the benefit of the audience.

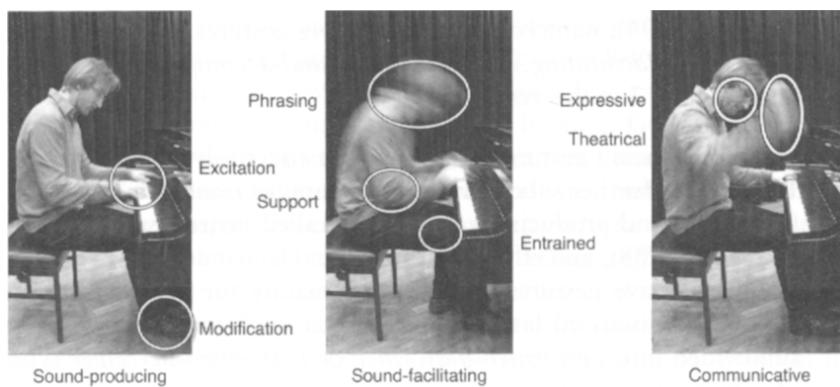


Figure 8.21: Examples of where different type of musical gestures (sound producing, sound facilitating and communicative) may be found in piano performance [from Leman-Godoy (2010)].

Fig. 6.21 shows different types of musical gestures involved in piano performance. Note that the different categories are not mutually exclusive, as several gestures have multiple functions. For example, hitting a final chord followed by a theatrical lift can be seen as having a sound producing and sound facilitating function, as well as a communicative function.

While the relevance of movement and gesture as a main channel of non-verbal communication becomes evident and increasing research efforts are devoted to them, the focus is here centered on the qualities that make a gesture expressive.

Gesture can be defined as "a movement of the body that contains information". The point is now what kind of information is contained in the movements of the body we are interested in. Especially in performing arts, gesture is not only intended to denote things or to support speech as in the traditional framework of natural gesture, but the information it contains and conveys is often related to the affective, emotional domain. That is, *expressive gesture* is the responsible of the communication of information that we call expressive content. Expressive content is different and in most cases independent from, even if often superimposed to, possible denotative meaning. Expressive content concerns aspects related to feelings, moods, affect, intensity of emotional experience. For example, the same action can be performed in several ways, by stressing different qualities of movement: it is possible to recognize a person from the way he/she walks, but it is also possible to get information about the emotional state of a person by looking at his/her gait, e.g., if he/she is angry, sad, happy. In the case of gait analysis, we can therefore distinguish among several objectives and layers of analysis: a first one aiming at describing the physical features of the movement, for example in order to classify it, a second one aiming at extracting

the expressive content gait conveys, e.g., in terms of information about the emotional state that the walker communicates through his/her way of walking. From this point of view, walking can be considered as an expressive gesture: even if no denotative meaning is associated with it, it still communicates information about the emotional state of the walker, i.e., it conveys a specific expressive content.

8.4.3.1 Recognition of expression in gestures

As shown in Fig. 6.4, we need to extract relevant features in order to recognize the expressive content of a gesture. In the human movement and dance analysis domain, we can distinguish among features calculated on different time scales:

- *local features*, calculated on a time interval of a few milliseconds (for example, one or a few frames coming from a video camera);
- *event features*, calculated on a movement stroke, or motion phase, on time durations of a few seconds; and
- *mid and high-level features* that relate to the conveyed expressive content (but also to cognitive aspects) and refer to sequences of movement strokes or motion (and pause) phases.

8.4.3.2 Local motion feature detection

Motion tracking In modeling human movement and gesture, a first stage is responsible of the processing of the incoming video frames in order to detect and obtain information about the *motion* that is actually occurring. It receives as input images from one or more videocameras and, possibly, information from other sensors (e.g., accelerometers). The system extracts the dancer's silhouette using background subtraction techniques (Fig. 6.22). Two types of output are generated: processed images (e.g., see Fig. 6.25) and trajectories of body parts. Feature extraction is accomplished by means of consolidated computer vision techniques usually employed for real-time analysis and recognition of human motion and activity. The techniques we use include feature tracking based on the Lucas-Kanade algorithm (Fig. 6.23), skin colour tracking to extract positions and trajectories of hands and head (Fig. 6.24), and Silhouette Motion Images.

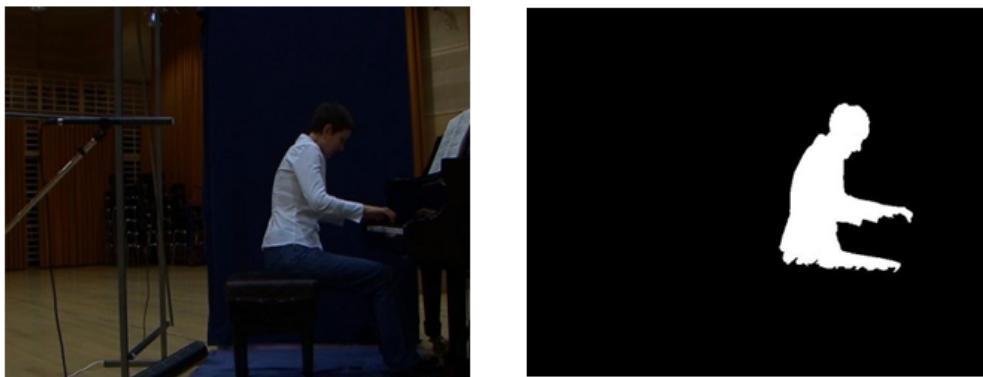


Figure 8.22: Silhouette extraction.

A Silhouette Motion Image (SMI) is an image carrying information about variations of the silhouette shape and position in the last few frames. SMIs are inspired to motion-energy images (MEI) and motion-history images (MHI). They differ from MEIs in the fact that the silhouette in the last (more recent)

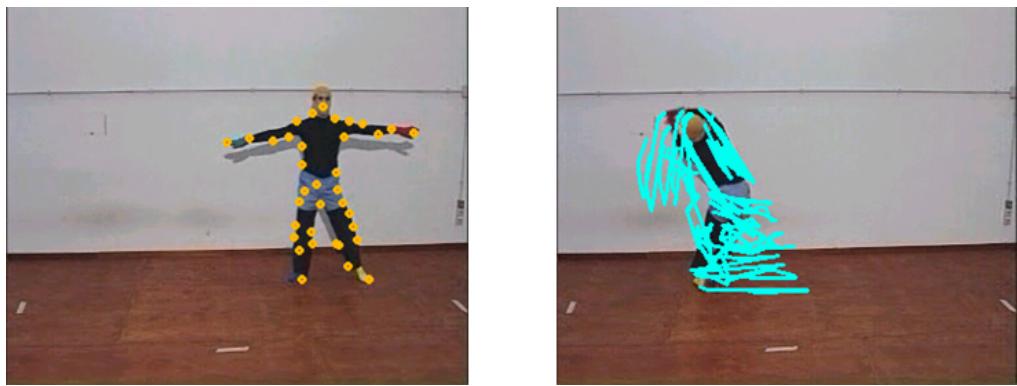


Figure 8.23: The Lucas - Kanade (LK) algorithm is used to track features in the input image.

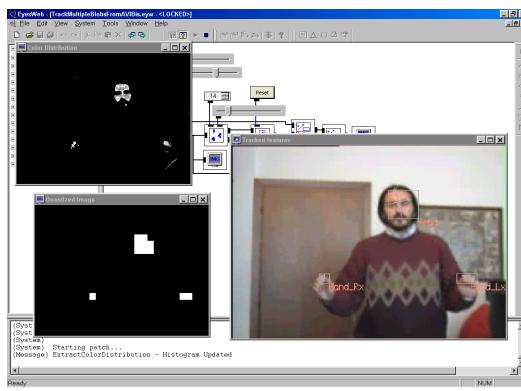


Figure 8.24: Example of skin colour tracking to extract positions and trajectories of hands and head.

frame is removed from the output image: in such a way only motion (including internal motion—that is, movement of overlapped parts of the body) is considered while the current posture is skipped.

$$SMI(t, N) = \left[\sum_{i=1}^N Silhouette(t-i) \right] - Silhouette(t) \quad (8.7)$$

Thus, SMIs can be considered as carrying information about the *amount of motion* occurred in the last N frames. Information about time is implicit in SMI and is not explicitly recorded. We also use an extension of SMIs, which takes into account the internal motion in silhouettes (see Fig. 6.25). In such a way, we are able to distinguish between global movements of the whole body in the General Space and internal movements of body limbs inside the Kinesphere.

Information motion detection and tracking provides to the upper levels is actually encoded in two different forms: positions and trajectories of points on the body (possibly related to specific body parts, e.g., hands, head, feet), and images directly resulting from the processing of the input frames (e.g., human silhouettes, SMIs).

Motion cues A second stage is responsible of the extraction of a set of *motion cues* from the data coming from low-level motion tracking. It computes a collection of motion cues describing movement and its qualities, by employing computer vision, statistical, and signal processing techniques. Important cues are Quantity of Motion and Contraction Index.

Quantity of Motion (QoM) is computed as the area (i.e., number of pixels) of a SMI (e.g., the number of pixels in the grey area in Fig. 6.25a). It can be considered as an overall measure of the amount of



Figure 8.25: (a) An example of SMI with time window of four frames. (b) Measure of internal motion in SMIs.

detected motion, involving velocity and force. QoM can be thought as a first rough approximation of the physical momentum, i.e., $q = mv$; where m is the mass of the moving body and v stands for its velocity. The shape of the QoM graph is close to the shape of the graphs of velocity of a marker put on a limb. QoM has two problems: (i) the measure depends on the distance from the camera; (ii) difficulties emerge when comparing measures from different dancers. These problems are solved by scaling the SMI area by the area of the most recent silhouette:

$$QoM(t) = \frac{\text{Area}[SMI(t, N)]}{\text{Area}[Silhouette(t)]} \quad (8.8)$$

In this way, the measure becomes relative, i.e., independent from the camera distance (in a range depending on the resolution of the videocamera), and it is expressed in terms of fractions of the body area that moved. For example, it is possible to say that at instant t a movement corresponding to the 2.5% of the total area covered by the silhouette happened.

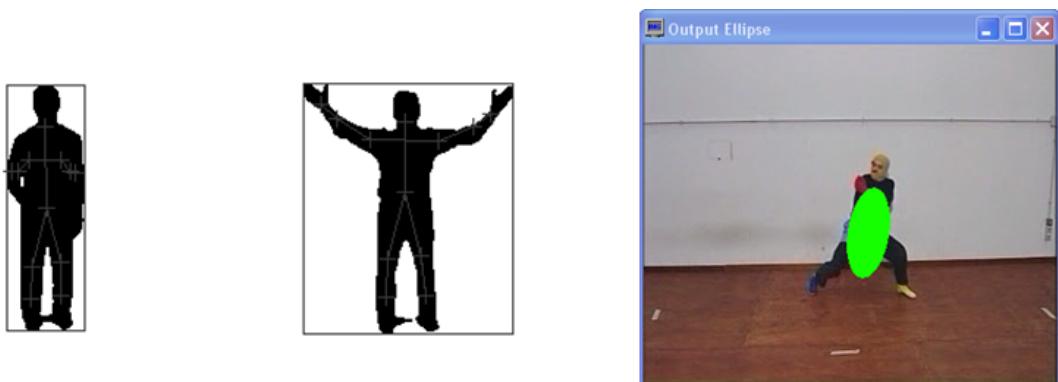


Figure 8.26: Computation of Contraction Index (CI).

The Contraction Index (CI) is a measure of how the dancer's body uses the space surrounding it. It is related to Laban's *personal space*. The algorithm to compute the CI combines two different techniques: the individuation of an ellipse approximating the body silhouette and computations based on the bounding region. The former is based on an analogy between the image moments and mechanical moments: in this perspective, the three central moments of second order build the components of the inertial tensor of the rotation of the silhouette around its centre of gravity: this allows to compute the axes (corresponding

to the main inertial axes of the silhouette) of an ellipse that can be considered as an approximation of the silhouette: eccentricity of such an ellipse is related to contraction/expansion; orientation of the axes is related to the orientation of the body. The second technique used to compute CI is related to the bounding region, i.e., the minimum rectangle surrounding the dancer's body. The algorithm computes the ratio between the area (i.e., the number of pixels) of the object's silhouette and the area of object's bounding box

$$CI(t) = \frac{\text{Area}[Silhouette(t)]}{\text{Area}[BoundingBox(t)]} \quad (8.9)$$

Intuitively, if the limbs are fully stretched and not lying along the body, this component of the CI will be low, while, if the limbs are kept tightly nearby the body, it will be high (near to 1). While the dancer is moving, the CI varies continuously. Even if it is used with data from only one camera, its information is still reliable, being almost independent from the distance of the dancer from the camera. A use of this cue consists of sampling its values at the end and the beginning of a stretch movement, in order to classify that movement as a contraction or expansion.

8.4.3.3 Event motion feature detection

Motion segmentation A third step of analysis consists in segmenting motion in order to individuate motion and non-motion (pause) phases. The temporal duration of such phases is then measured and compared with the total duration of the dance performance. The QoM measure has been used to perform the segmentation between pause and motion phases. QoM is related to the overall amount of motion and its evolution in time can be seen as a sequence of bell-shaped curves (motion bells). In order to segment motion, a list of these motion bells has been extracted and their features (e.g., peak value and duration) computed. An empirical threshold has been defined for these experiments: the dancer is considered to be moving if the area of the motion image (i.e., the QoM) is greater than 2.5% of the total area of the silhouette. Fig. 6.27 shows motion phases after automated segmentation: a motion bell characterizes each motion phase.

Motion segmentation can be considered as a first step toward the analysis of the rhythmic aspects of the dance. Analysis of the sequence of pause and motion phases and their relative time durations can lead to a first evaluation of dance tempo and its evolution in time, i.e., tempo changes, articulation (the analogous to music legato/staccato). Parameters from pause phases are also extracted to individuate real still standing positions from active pauses involving low-motion (hesitating or oscillation movements). For example, the Amount of Periodic Movement (PM) gives a preliminary information about the presence of rhythmic movements. Computation of PM starts from QoM. Movement is segmented in motion and pause phases using a threshold on QoM; inter-onset intervals are then computed as the time elapsing from the beginning of a motion phase and the beginning of the following motion phase. The variance of such inter-onset intervals is taken as an approximate measure of PM.

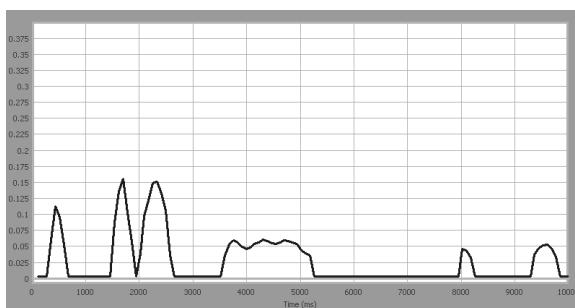


Figure 8.27: Motion segmentation.



Motion qualities With these data motion fluency and impulsiveness are evaluated. They are related to Laban's Flow and Time axes. *Fluency* can be estimated starting from an analysis of the temporal sequence of motion phases by a weighted contributions of several measures. The two most relevant measures are the percentage of acceleration and deceleration and the ratio between the durations of pause and motion phases in a given time window. A gesture performed with frequent stops and restarts (i.e., characterized by a high number of short pause and motion phases) will have less fluency than the same movement performed in a continuous, harmonic way (i.e., along a few long motion phases). The hesitating, bounded performance will be characterized by a higher percentage of acceleration and deceleration in the time unit (due to the frequent stops and restarts).

A first measure of *impulsiveness* can be obtained from the shape of a motion bell. In fact, since QoM is directly related to the amount of detected movement, a short motion bell having a high peak value will be the result of an impulsive movement (i.e., a movement in which speed rapidly moves from a value near or equal to zero, to a peak and back to zero). On the other hand, a sustained, continuous movement will show a motion bell characterized by a relatively long time period in which the QoM values have little fluctuations around the average value (i.e., the speed is more or less constant during the movement). Impulsiveness (IM) is measured as the variance of Quantity of Motion in a time window of 3 s, i.e., a user is considered to move in an impulsive way if the amount of movement the videocamera can detect on her changes considerably in the time window.

The Directness Index (DI) is a measure of how much a trajectory is direct or flexible. In the Laban's Theory of Effort it is related to the Space dimension. DI is computed as the ratio between the length of the straight line connecting the first and last point of a given trajectory and the sum of the lengths of each segment constituting the given trajectory (Fig. 6.29). Therefore, the more it is near to one, the more direct is the trajectory (i.e., the trajectory is "near" to the straight line).

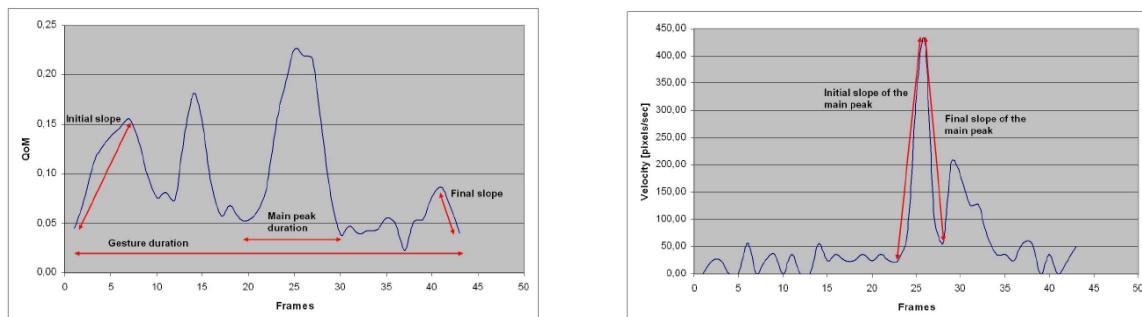


Figure 8.28: Temporal profile of motion features.

Motion phases features A main characteristic of modern dance is to explore how emotional experiences could be expressed in body movements. The basis of natural body expression is further developed in terms of dance movements. This close linkage between modern dance and human emotions, has led to the proposal that modern dance contains cues from the underlying principle of natural emotional movement expression. Therefore, it has been suggested that emotions expressed in dance movements are a unique way to extract cues for emotions in natural bodily expressions.

Figure 6.30 shows the mean values computed for each motion phase of QoM and CI. The four graphs refer to four performances by the same dancer in which the dancer tried to express the four basic emotions. In the figures line types are associated to emotions as follows: anger-solid line; fear-dashed line; joy-dash-dot line; grief-dotted line. It can be noticed, for example, that curves representing the average QoM for anger (solid line) and fear (dashed line) have a similar trend: i.e., they starts with low

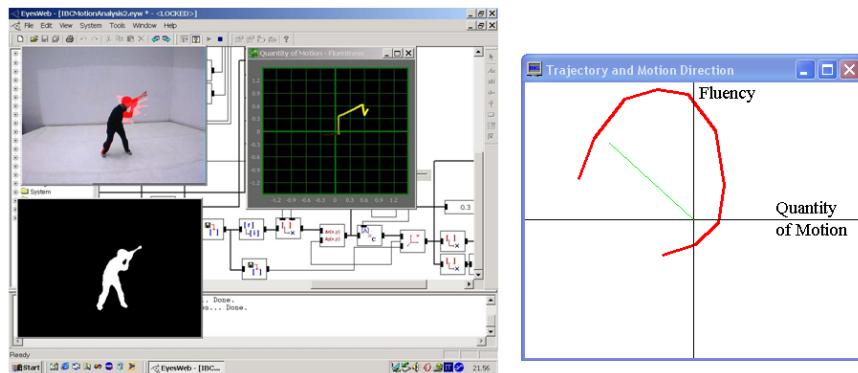


Figure 8.29: Directness index as a measure of how a trajectory is direct or flexible.

values and slow increase at the beginning, then they continuously increase with increasing steepness. Fear, however, have much more motion phases than anger indicating a less fluent motion. CI for joy (dash-dot line) has quite low values with respect to the other emotions, while fear (dashed line) has quite high values, meaning that the body is often contracted (i.e., limbs are often close to the centre of gravity). Grief (dotted line) always has a high number of motion phases and a high variance of the average values of QoM, meaning frequent transitions between motion and pause phases and very low fluency. Joy (dash-dot line), instead, has only four long motion phases indicating a very fluent motion.

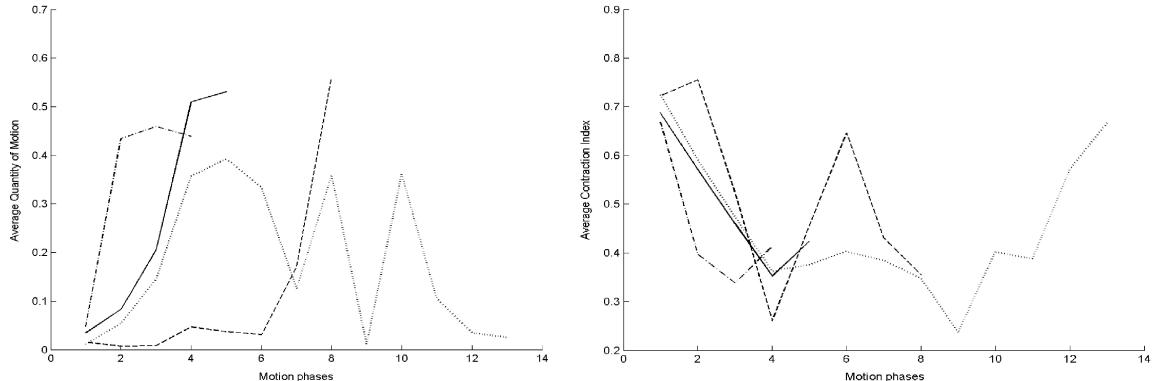


Figure 8.30: Mean values of the QoM (left) and CI (right) computed for each motion phase (the four graphs refer to four performances by the same dancer, each one expressing a different basic emotion: anger-solid line; fear-dashed line; joy-dash-dot line; grief-dotted line). The X-axis is the index of the motion phase in which the movement has been segmented (therefore, X is not the time axis).

8.4.3.4 Mid-level motion features

Local and event features can be mapped in the Laban space (see Sect. 6.3.2) by statistical methods like multiple regression, by classification methods like support vector machines (SVM), by methods based on fuzzy sets. For example a gesture can be classified as Direct or Flexible, Quick or Sustained. Analysis can be performed on the whole gesture, from its beginning to its end. Such analysis provides a global overview of the expressive qualities of the whole gesture but it might be less significant if the gesture is not characterized by a single over-standing quality, but rather its expressive qualities change during execution. A decisive step for high-level gesture interpretation and analysis is accomplished through the representation of gestures as trajectories in abstract multidimensional spaces whose dimensions are

relevant for expressive characterization. Thus for example, we can map expressive gestures in a 2D space whose dimensions are Laban's Space and Time dimensions (see Sect. 6.3.2).

A further step consists in analyzing the trajectories generated by classified gestures in the 2D Laban's space. For example, concentration of trajectories in the Quick and Direct quarter of the space may be interpreted as a preference of the user for fast, direct, targeted movements with a high degree of confidence and decision. At the opposite quarter of the space, a prevalence of sustained, smooth, and flexible movement may account for a calmer and more relaxed interaction style.

The role of Laban effort dimensions in dance gestures can be described as:

1. space dimension is related to Laban's notion of personal space, i.e. the extent in which limbs are contracted or expanded in relation to the body center;
2. time dimension is specified in terms of the overall duration of time and tempo changes, which can be elaborated as the underlying structure of rhythm and flow of the movement;
3. weight dimension is specified in terms of the amount of tension and dynamics in movements, e.g. the vertical component of acceleration;
4. flow dimension is specified in terms of an analysis of shapes of speed and energy curves, and features that relates to the frequency and rhythm of motion and pause phases.

Emotion	Movement cues
Anger	short duration of time frequent tempo changes, short stops between changes movements reaching out from body centre dynamic and high tension in the movement; tension builds up and then 'explodes'
Fear	frequent tempo changes, long stops between changes movements kept close to body centre sustained high tension in movements
Grief	long duration of time few tempo changes, smooth tempo continuously low tension in the movements
Joy	frequent tempo changes, longer stops between changes movements reaching out from body centre dynamic tension in movements; changes between high and low tension

Table 8.6: Movement cues associated to different emotions.

Motion cues can be associated in different combinations for each emotion category as shown in Table 6.6. Moreover expressive gestures in social interaction can be studied, focusing on aspects of synchronization, entrainment and empathy, using relations among expressive cues. Emotional communication can be studied focusing on expressive gestures as a way to communicate a particular emotion to the audience and expressive as a way to emotionally engage in the felt emotion, causing a strong emotional response.

8.4.4 Faces and emotional states

There is a long history of interest in the problem of recognizing emotion from facial expressions, influenced by Darwin's pioneering work. Beethoven, after he became deaf, wrote in his conversation

books that he could judge from the performer's facial expression whether or not the performer was interpreting his music in the right spirit. The face is where our eyes linger during conversation. In a video-teleconference, where the camera is free to point anywhere, the default is to have it point to the faces of the people in the room. Whether in person or over a video-telephone, we tend to communicate most affectively "face-to-face."

Facial expressions are subject to what Ekman has termed "social display rules" that limit the range of acceptable expression, such as in business or social settings. For example, it is inappropriate for a businessman to contort his face in extreme disgust or disappointment during a negotiation session. In serious meetings he knows to express only mild emotion, regardless of his feelings. However, at a sporting event the social display rules are different. There he is not only free to contort his face, but also to vociferate, to wave his arms and torso, and to jump up and down.

A long established tradition attempts to define the facial expression of emotion in terms of qualitative targets, i.e., static positions capable of being displayed in a still photograph. The still image usually captures the apex of the expression, i.e., the instant at which the indicators of emotion are most marked. More recently, emphasis has switched towards descriptions that emphasize gestures, i.e., significant movements of facial features.

In the context of faces, the task has almost always been to classify examples of archetypal emotions. That may well reflect the influence of Ekman and his colleagues, who have argued robustly that the facial expression of emotion is inherently categorical.

8.4.4.1 Targets and gestures associated with emotional expression

Analysis of the emotional expression of a human face requires a number of preprocessing steps which attempt to detect or track the face; locate characteristic facial regions such as eyes, mouth, and nose on it; extract and follow the movement of facial features, such as characteristic points in these regions; or model facial gestures using anatomic information about the face. Facial features can be viewed as either static (such as skin color), slowly varying (such as permanent wrinkles), or rapidly varying (such as raising of the eyebrows) with respect to time evolution. Detection of the position and shape of the mouth, eyes, particularly eyelids, wrinkles, and extraction of features related to them are the targets of techniques applied to still images of humans.

Techniques which attempt to identify facial gestures for emotional expression characterization face the problems of locating or extracting the facial regions or features, computing the spatio-temporal motion of the face through optical flow estimation, and introducing geometric or physical muscle models describing the facial structure or gestures.

Most of the above techniques are based on the work of Ekman and Friesen, who produced a system for describing all visually distinguishable facial movements, called the *facial action coding system* (FACS). FACS is an anatomically oriented coding system, based on the definition of *action units* (AUs) of a face that cause facial movements. Each AU may correspond to several muscles that together generate a certain facial action.

The FACS model has recently inspired the derivation of facial animation and definition parameters in the framework of the ISO MPEG-4 standard. In particular, the facial definition parameter set (FDP) (see Fig. 6.31a) and the facial animation parameter set (FAP) were designed in the MPEG-4 framework to allow the definition of a facial shape and texture, as well as the animation of faces reproducing expressions, emotions, and speech pronunciation. The FAPs are based on the study of minimal facial actions and are closely related to muscle actions. They represent a complete set of basic facial actions, such as squeeze or raise eyebrows, open or close eyelids, and therefore allow the representation of most natural facial expressions. Examples of FAPs are `raise_l_o_eyebrow` (rise vertical displacement of left outer eyebrow), `raise_r_i_eyebrow` (rise vertical displacement of left inner eyebrow), `open_jaw` (verti-



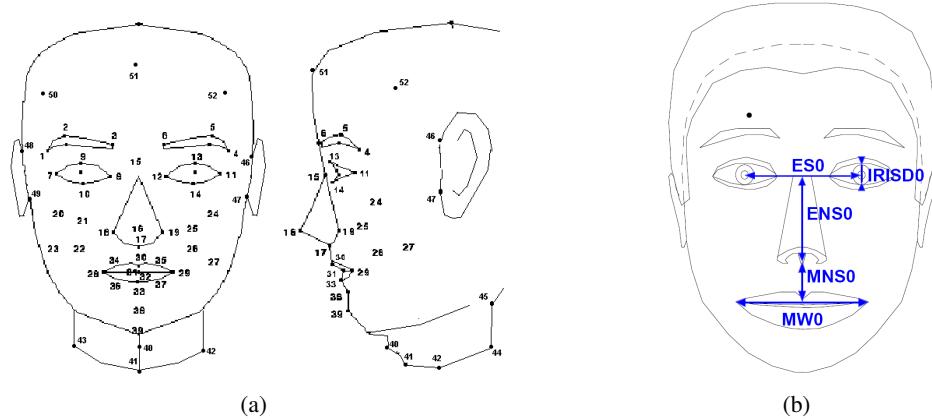


Figure 8.31: Facial Definition Parameters (a) and Facial Animation Parameter Units (b) in MPEG-4: IRIS Diameter (by definition it is equal to the distance between upper and lower eyelid) in neutral face, Eye Separation, Eye - Nose Separation, Mouth - Nose Separation, Mouth - Width Separation

cal jaw displacement - does not affect mouth opening). All FAPs involving translational movement are expressed in terms of the facial animation parameter units (FAPU). These units aim at allowing interpretation of the FAPs on any facial model in a consistent way, producing reasonable results in terms of expression and speech pronunciation. The FAPUs are illustrated in Fig. 6.31b and correspond to fractions of distances between some key facial features.

Mpeg-4 defines two high level FAPs, i.e. *visemes* for face animation and *expressions*, for reproduction of a facial expressions. Visemes (Tab. 6.7) are a mouth posture correlated to a phoneme, used for speech (visual counter part of phoneme), and are composed as a preset combination of FAPs. Expressions are used to show emotions and define seven archetypal expression (Fig. 6.32), also termed universal because they are recognized across cultures: i.e. Sadness, Anger, Joy Fear, Disgust, Surprise and Neutral. In order to render facial expressions, a weighted combination of 2 visemes and 2 facial expressions for each frame are coded. The decoder is free to interpret the effect of visemes and expressions after FAPs are applied. Definitions of visemes and expressions using FAPs can also be downloaded. The modifications to a neutral face are described in Table 6.8.

Viseme #	phonemes	example	Viseme #	phonemes	example
0	none	na	1	p, b, m	put, bed, mill
2	f, v	far, voice	3	T,D	think, that
4	t, d	tip, doll	5	k, g	call, gas
6	tS, dZ, S	chair, join, she	7	s, z	sir, zeal
8	n, l	lot, not	9	r	red
10	A:	car	11	e	bed
12	I	tip	13	Q	top
14	U	book			

Table 8.7: Viseme and Related Phonemes.

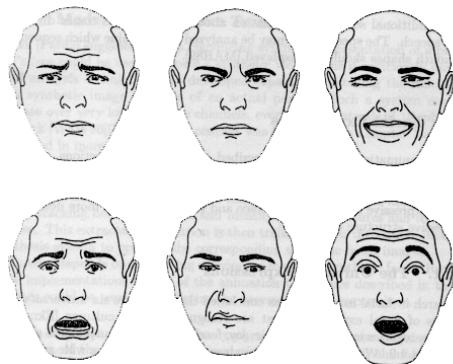


Figure 8.32: Archetypal face expressions in Mpeg-4: Sadness, Anger, Joy Fear, Disgust, Surprise.

expression name	textual description
joy	The eyebrows are relaxed The mouth is open and the mouth corners pulled back toward the ears.
sadness	The inner eyebrows are bent upward. The eyes are slightly closed. The mouth is relaxed.
anger	The inner eyebrows are pulled downward and together. The eyes are wide open. The lips are pressed against each other or opened to expose the teeth.
fear	The eyebrows are raised and pulled together. The inner eyebrows are bent upward. The eyes are tense and alert.
disgust	The eyebrows and eyelids are relaxed. The upper lip is raised and curled, often asymmetrically.
surprise	The eyebrows are raised. The upper eyelids are wide open, the lower relaxed. The jaw is opened.

Table 8.8: Facial archetypal expressions.

8.4.4.2 Facial expression recognition

The approaches, by which a computer can recover information about emotional state from facial actions-expressions, are divided into two main categories: target oriented and gesture oriented. In target-oriented approaches, recognition of a facial expression is performed using a single image of a face at the apex of the expression. Gesture-oriented approaches extract facial temporal information from a sequence of images in an episode where emotion is expressed, with facial expressions normally lasting between 0.5 and 4 s. Transitional approaches were also developed that use two images, representing a face in its neutral condition and at the apex of the expression.

Target-oriented approaches Most psychological research on facial expression analysis has been conducted on mug-shot pictures that capture the subject's expression at its apex (Fig. 6.33). These pictures allow one to detect the presence of static cues (such as wrinkles) as well as the positions and shapes of facial features. However, extracting the relevant cues from static images has proved difficult, and few

facial expression classification techniques based on static images have been successful.



Figure 8.33: Face expression at its apex (Neutral, Happiness, Surprise, Anger and Disgust.[from Essa and Pentland, PAMI 97])



Figure 8.34: Eight sample face images from the CMU dataset of 20 persons showing neutral, angry, happy, and sad facial expressions.

Transitional approaches Transitional approaches focus on computing motion of either facial muscles or facial features between neutral and apex instances of a face. The optical flow based approach uses dense motion fields computed in selected areas of the face, such as the mouth and eyes; it tries to map these motion vectors to facial emotions using motion templates which have been extracted by summing over a set of test motion fields

Gesture-Oriented Approaches Most approaches dealing with facial gestures are based on optical flow estimation. Image gradient, or image filtering, or image correlation are used for estimating optical flow. The extracted flow patterns can be used by conventional pattern classification techniques as well as neural networks to recognize the corresponding expressions.

Approaches to extracting facial emotions from image sequences fall into three classes which are described next.

Optical Flow-based Approach The optical flow based approach uses dense motion fields computed in selected areas of the face, such as the mouth and eyes; it tries to map these motion vectors to facial emotions using motion templates which have been extracted by summing over a set of test motion fields. A problem in these approaches, which in general are computationally intensive, is caused by the inherent noise of the local estimates of motion vectors, which may result in degradation of the recognition performance.

Feature Tracking Approach In the second approach, motion estimates are obtained only over a selected set of prominent features in the scene. Analysis is performed in two steps: first each image frame of a video sequence is processed to detect prominent features, such as edges, corner-like and high-level patterns like eyes, brows, nose, and mouth, followed by analysis of the image motion. In particular, the movement of features can be tracked between frames using Lucas-Kanade's optical flow algorithm, which has high tracking accuracy. The advantage of this approach is efficiency, due to the great reduction of image data prior to motion analysis; on the other hand, it is not certain that the feature points which can be extracted suffice for the emotion recognition task.

Model Alignment Approach The third approach aligns a 3-D model of the face and head to the image data to estimate both object motion and orientation (pose).

Flow is estimated taking into account consecutive frames; the computed motion field is accumulated over all time periods. Features from Motion flow and motion energies, corresponding to movement in eight different directions of important facial parts (Fig. 12), form a feature vector sequence, which can feed six HMMs for classification of the archetypal expressions.



Chapter 9

Music information processing

Giovanni De Poli

Copyright © 2005-2018 Giovanni De Poli

except for paragraphs labeled as *adapted from <reference>*

This book is licensed under the CreativeCommons Attribution-NonCommercial-ShareAlike 3.0 license. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/>, or send a letter to Creative Commons, 171 2nd Street, Suite 300, San Francisco, California, 94105, USA.

9.1 Elements of music theory and notation

Music as well as language was long cultivated by aural transmission before any kind of systematic method of writing it down was invented. But the desire to record laws, poetry and other permanent statements gave rise the problem of how to write down music. In western tradition the focus is on a symbolic system which can represent both the pitch and the rhythm of a melody. In the following section the general principles of western notation will be presented.

In music the word *note* can mean three things: (1) a single sound of fixed pitch; (2) the written symbol of a musical sound; (3) a key on the piano or other instrument. A note is often considered as the atomic element in the analysis and perception of the musical structure. The two main attributes of a note are pitch and duration. These are the two most important parameters in music notation and, probably not coincidentally, the first ones to evolve. A functional piece of music can be notated using just these two parameters. Most of the other ones, such as loudness, instrumentation, or tempo, are usually written in English or Italian somewhere outside of the main musical framework.

9.1.1 Pitch

In music, a scale is a set of musical notes that provides material for part or all of a musical work. Scales are typically ordered in pitch, with their ordering providing a measure of musical distance. Human pitch-perception is periodic: a note with a doubled frequency as another sounds very similar and is commonly given the same name, called *pitch class*. The interval (i.e. the span of notes) between these two notes is called *octave*. Thus the complete definition of a note consists of its pitch class and the octave it lies in. Scales in traditional Western music generally consist of seven notes (pitch classes) and repeat at the octave. The name of the notes of a scale is indicated by the first seven letters of the alphabet. For

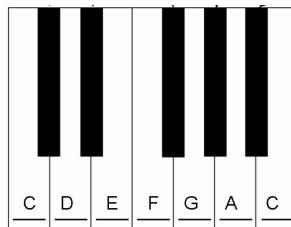


Figure 9.1: One octave in a piano keyboard.

historical reasons the musical alphabet starts from C and not from A, and it is arranged thus: C D E F G A B, closing again with C, so producing an interval from C to C of eight notes. These eight notes are represented by white keys on the piano keyboard (Figure 7.1). In Italian the pitch classes are called, respectively, do, re, mi, fa, sol, la, si. The octaves are indicated by numbers. In general the reference is the fourth octave containing the C4 (the middle C) and A4 (the diapason reference) with frequency $f = 440$ Hz. The lowest note on most pianos is A0, the highest C8.

9.1.1.1 Pitch classes, octaves and frequency

In most western music the frequencies of the notes are tuned according the twelve-tone equal temperament. In this system the octave is divided into a series of 12 equal steps (equal frequency ratio). On a piano keyboard the steps are represented by the 12 white and black keys forming an octave. The interval between two adjacent keys (white or black) is called *semitone* or half tone. The ratio s corresponding to a semitone can be determined considering that the octave ratio is composed by 12 semitones, i.e. $s^{12} = 2$, and thus the semitone frequency ratio is given by

$$s = \sqrt[12]{2} \approx 1.05946309 \quad (9.1)$$

i.e. about a six percent increase in frequency. The semitone is further divided in 100 (equal ratio) steps, called cents. I.e.

$$1 \text{ cent} = \sqrt[100]{s} \approx 1.000577$$

The distance between two notes whose frequency are f_1 and f_2 is $12 \log_2(f_1/f_2)$ semitones = $1200 \log_2(f_1/f_2)$ cents. The just noticeable difference in pitch is about five cents.

In the equal temperament system a note which is n steps or semitones apart the central A (A4) has frequency

$$f = 440 \times 2^{n/12} \text{ Hz} = 440 \times s^n \text{ Hz} \quad (9.2)$$

For example middle C (C4) is $n = -9$ semitones apart from A4 and has frequency $f = 440 \times 2^{-9/12} = 261.63$ Hz. A convenient logarithmic scale for pitch is simply to count the number of semitones from a reference pitch, allowing fractions to permit us to specify pitches which don't fall on a note of the Western scale. This creates a linear pitch space in which octaves have size 12 and semitones have size 1. Distance in this space corresponds to physical distance on keyboard instruments, orthographical distance in Western musical notation, and psychological distance as measured in psychological experiments and conceived by musicians. The most commonly used logarithmic pitch scale is *MIDI pitch*, in which the pitch 69 is assigned to a frequency of 440 Hz, i.e. A4, the A above middle C. A note with MIDI pitch p has frequency

$$f = 440 \times 2^{(p-69)/12} \text{ Hz} = 440 \times s^{p-69} \text{ Hz} \quad (9.3)$$

and a note with frequency f Hz has MIDI pitch

$$p = 69 + 12 \log_2(f/440) \quad (9.4)$$



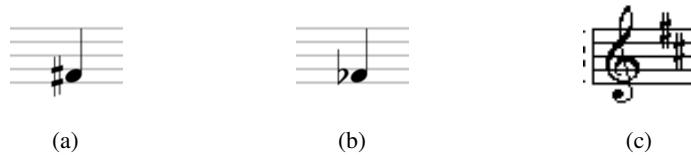


Figure 9.2: Example of a sharp (*a*) and a flat (*b*) note. Example of a key signature (*c*): D major.

Because there are actually 12 notes on the keyboard, the 7 note names can also be given a modifier, called *accidental*. The two main modifiers are sharps (Fig. 7.2(a)) and flats (7.2(b)) which respectively raise or lower the pitch of a note by a semitone, where a semitone is the interval between two adjacent keys (white or black).

If we ignore the difference between octave-related pitches, we obtain the pitch class space, which is a circular representation. Since pitch class space is a circle, we return to our starting point by taking a series of steps in the same direction: beginning with C, we can move "upward" in pitch class space, through the pitch classes C \sharp , D, D \sharp , E, F, F \sharp , G, G \sharp , A, A \sharp , and B, returning finally to C. We can assign numbers to pitch classes. These numbers provide numerical alternatives to the letter names of elementary music theory: 0 = C, 1 = C \sharp =D \flat , 2 = D, and so on. Thus given a Midi pitch p , its pitch class pc and octave number oct are given by

$$pc = p \pmod{12} \quad (9.5)$$

$$oct = \lfloor p/12 \rfloor - 1 \quad (9.6)$$

and viceversa

$$p = pc + 12(oct + 1) \quad (9.7)$$

For example middle C (C4) has $p = 60$, and $pc = 0$, $oct = 4$. Notice that some pitch classes, corresponding to black keys in the piano, can be spelled differently: e.g. $pc = 1$ can be spelled as C \sharp or as D \flat .

9.1.1.2 Musical scale.

All humans perceive a large continuum of pitch. However, the pitch systems of all cultures consist of a limited set of pitch categories that are collected into ordered subsets called scales. In music, a scale is a set of musical notes that provides material for part or all of a musical work. Scales in traditional Western music generally consist of seven notes (diatonic scale) derived from an alphabet of the 12 chromatic notes within an octave, and repeat at the octave. Notes are separated by whole and half step intervals of tones and semitones. In many musical circumstances, a specific note is chosen as the *tonic*: the central and most stable note of the scale. Relative to a choice of tonic, the notes of a scale are often labelled with roman numbers recording how many scale steps above the tonic they are. For example, the notes of the C diatonic scale (C, D, E, F, G, A, B) can be labelled I, II, III, IV, V, VI, VII, reflecting the choice of C as tonic. The term "scale degree" refers to these numerical labels: in the previous case, C is called the first degree of the scale, D is the second degree of the scale, and so on. In the C diatonic scale, with C chosen as tonic, C is the first scale degree, D is the second scale degree, and so on. In the major scale the pattern of intervals in semitones between subsequent notes is 2-2-1-2-2-2-1; these numbers stand for whole tones (2 semitones) and half tones (1 semitone). The interval pattern of minor scale is 2-1-2-2-1-2-2. The scale defines interval relations relative to the pitch of the first note, which can be any one of the keyboard.

In the western music, the scale define also a relative importance of the different degree. The first (I) degree (called tonic or keynote) is the most important. The degree next in importance is the fifth (V), called dominant because of its central position and dominating role in both melody and harmony. The fourth (IV) degree (subdominant) has a slightly dominating role than the dominant. The other degree are supertonic (II), mediant (III), submediant (VI), leading note (VII). The numerical classification depends also on the scale: for example in the major scale the (major) third has $2+2=4$ semitones interval, while in the minor scale the (minor) third has $2+1=3$ semitones interval. There are five adjectives to qualify the intervals: perfect intervals are the I, IV, V, and VIII. The remaining intervals (e.g. II, III, VI, VII) in the major scale are called major intervals. If a major interval is reduced by a semitone, we get a minor interval. If a major or perfect interval is increased by a semitone, we get a corresponding augmented interval. Any minor or perfect interval reduced by a semitone is called diminished interval.

The scale made by 12 tones per octave is called chromatic scale.

9.1.1.3 Musical staff

Notation of pitch is done by using a framework (or grid) of five lines called a staff. Both the lines and spaces are used for note placement. How high or low a pitch is played is determined by how high or low the note head is placed on the staff.

Notes outside the range covered by the lines and spaces of the staff are placed on, above or below shorter lines, called leger (or ledger) lines, which can be placed above or below the staff. Music is read from 'left' to 'right', thus it is a sort of two dimensional representation in a time-frequency plane.

A piano uses two staves, each one covering a different range of notes (commonly known as register). They are read simultaneously: two notes that are in vertical alignment are played together. An orchestral score will often have more than ten staves. To establish the pitch of any note on the staff we place a graphical symbol called a clef at the far left-hand side of the staff. The clef establishes the pitch of the note on one particular line of the staff and thereby fixes the pitch of all the other notes lying on, or related to, the same staff (see Fig. 7.3 and 7.4).

Sometimes (but not always) accidentals are placed to the immediate right of the clef sign and before the Time Signature. This indicates the tonality (or key) the song should be played in. The Key Signature consists of a small group of sharps or flats and tells you if any note (more precisely, pitch class) should be consistently sharped or flatted (Fig. 7.2(c)). For example, if there is a sharp on the F and on the C in a key signature (as in Fig. 7.2(c)), it tells a musician to play all notes "F" as "F♯" instead and all C notes as "C♯", regardless of whether or not they fall on that line. A flat on the B line tells a musician to play all notes "B" as Bb, and so on. The natural sign (♫) in front of a note will signal that the musician should play the white key version of the note. The absence of any sharp or flats at the beginning tells you the song is played in the key of C, i.e. without any pitch modification (as Fig. 7.3).

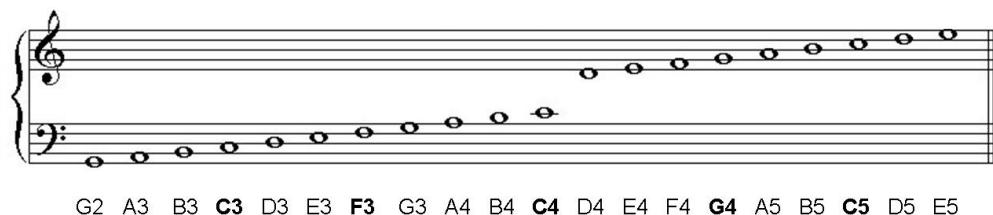


Figure 9.3: Staff and note names.



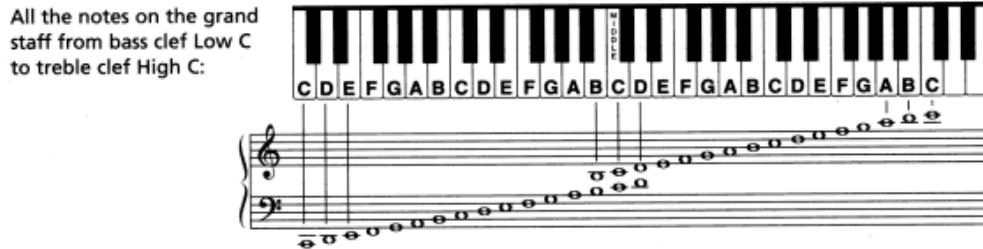


Figure 9.4: Correspondence of keys and notes on the staff.

9.1.2 Note duration

Music takes place in time, and so musicians have to organize it in terms not only of pitch but also of duration. They must choose whether the sounds they use shall be shorter or longer, according to the artistic purpose they wish to serve.

When we deal with symbolic representation, the symbolic duration (or note length) refers to the perceptual and cognitive organization of sounds, which prefer simple relations. Thus the symbolic duration is the time interval between the beginning of the event and the beginning of the next event, which can also be a rest. Notice that the actual sound duration (physical duration) can be quite different and normally is longer, due to the decay time of the instrument. In this chapter when not explicitly stated, we will deal with symbolic duration.

Duration symbols. In order to represent a sound, apart for naming it alphabetically, a symbol is used. Where the vertical position of a note on a staff or stave determines its pitch, its relative time value or length is denoted by the particular sign chosen to represent it. The symbols for note lengths are indicated in Table 7.1 and how sound lengths are divided is shown in Fig. 7.5. This is the essence of proportional time notation. The signs indicate only the proportions of time-lengths and do not give duration in units of time, minutes or seconds.

Note name: American Italian English	whole semibreve semibreve	half minima minim	quarter semiminima crotchet	eighth croma quaver	sixteenth semicroma semiquaver	thirty-second biscroma demisemiquaver
Length	1	1/2	1/4	1/6	1/16	1/32
Note symbol	o	o	o	o	o	o
Rest symbol	-	-	{}	γ	γ	γ

Table 9.1: Duration symbols for notes and rests.

At present the longest note in general use is the whole note or semibreve, which serves as the basic unit of length: i.e. the whole note has conventional length equal 1. This is divided (Fig. 7.5) into two half notes or minims (*minime*), 4 quarters or crotchets (*semiminime*), 8 eighths or quarvers (*crome*), 16 sixteenths or semiquavers (*semicrome*), 32 thirty-seconds or demisemiquavers (*biscrome*). The corresponding symbols for rests (period of silence) are shown in Figure 7.6.

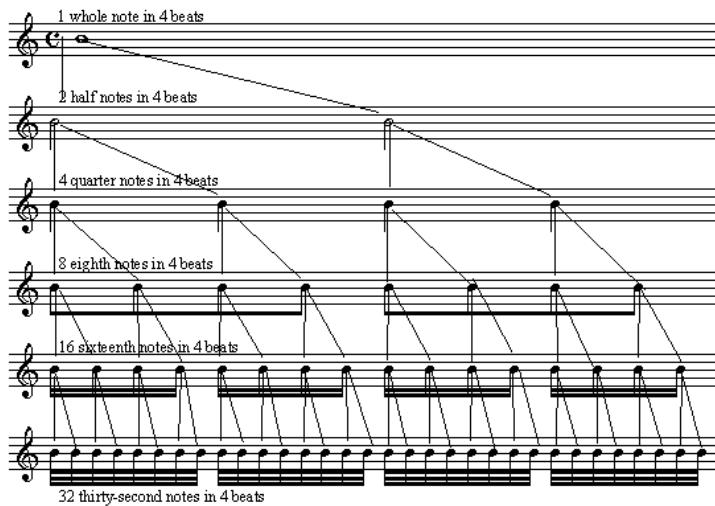


Figure 9.5: Symbols for note length.

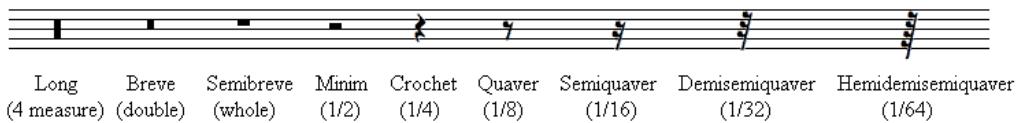


Figure 9.6: Symbols used to indicate rests of different length.

Notice that when we refer to symbolic music representation, as in scores, the note length is also called duration. However symbolic duration does not represent the actual duration of a sound; instead it refers to the difference from beginning of the next event to the beginning of the actual event. The real sound duration depends on the instrument type, how it is played, etc., and normally is not equivalent.



Figure 9.7: Tie example: crotchet (quarter note) tied to a quaver (eighth note) is equivalent to the dotted crotchet (dotted quarter note).

A dot, placed to the immediate right of the note-head, increases its time-value by half. A second dot, placed to the immediate right of the first dot, increases the original undotted time-value by a further quarter. Dots after rests increase their time-value in the same way as dots after notes. A tie (a curved line connecting the heads of two notes) serves to attach two notes of the same pitch. Thus the sound of the first note will be elongated according the value of the attached note. This is illustrated in the example given in Fig. 7.7 where a crotchet (quarter note) tied to a quaver (eighth note) is equivalent to the dotted crotchet (dotted quarter note) that follows. To divide a note value into three equal parts, or some other value than two, tuplets may be used. The most common triplet is the triplet: in this case the note length is reduced to 2/3 the original duration.



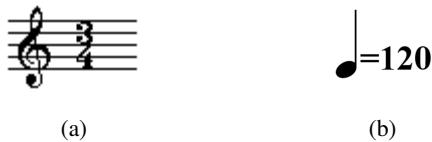


Figure 9.8: (a) Example of a time signature: 3/4 indicates three quarter note beats per measure. (b) Example of a metronome marking: 120 quarters to the minute.

9.1.3 Tempo

The signs of Table 7.1 do not give duration in units of time, minutes or seconds. The relationship between notes and rests is formalized but the duration or time value in seconds of any particular note is unquantified. It depends on the speed the musical piece is played. *Tempo* is the word used to cover all the variation of speed, from very slow to very fast.

Until the invention of a mechanical device called the metronome, the performance speed of a piece of music was indicated in three possible ways: through the use of tempo marks, most commonly in Italian; by reference to particular dance forms whose general tempi would have been part of the common experience of musicians of the time; by the way the music was written down, in particular, the choice of note for the beat and/or the time signature employed. Many composers give metronome marks to indicate exact tempo. The metronome measures the number of beats per minute (BPM) at any given speed. The allegro tempo may correspond to 120 BPM, i.e. beats per minute. This value corresponds to a frequency of $120/60 = 2$ beats per second. The beat duration is the inverse of the frequency, i.e. $d = 1/2 = 0.5$ sec. However most musicians would agree that it is not possible to give beats per minute (BPM) equivalents for these terms; the actual number of beats per minute in a piece marked allegro, for example, will depend on the music itself. A piece consisting mainly of minims (half notes) can be played very much quicker in terms of BPM than a piece consisting mainly of semi-quavers (sixteenth notes) but still be described with the same word.

9.1.4 Rhythm

Rhythm is the arrangement of events in time. In music, where rhythm has probably reached its highest conscious systematization, a regular pulse or *beat*, appears in groups of two, three and their compound combinations. The first beat of each group is accented. The metrical unit from one accent to the next is called a *bar* or *measure*. This unit is marked out in written scores by vertical lines (*bar lines*) through the staff in front of each accented beat.

Notice that tempo is often defined referring to rhythm and metre. The time signature (also known as "meter signature") is used to specify how many beats are in each bar and what note value constitutes one beat. Most time signatures comprise two numbers, one above the other. In text (as in this chapter), time signatures may be written in the manner of a fraction, e.g. 3/4. The first number indicates how many beats there are in a bar or measure; the second number indicates the note value which represents one beat (the "beat unit"). For example 3/4 indicates three quarter note beats per measure (Fig. 7.8(a)). In this case a metronome indication of 120 BPM (Fig. 7.8(b)) corresponds to $120/60$ beats per second: each quarter lasts $60/120 = 0.5$ sec and the measure lasts $3 \times 0.5 = 1.5$ sec. The duration of a musical unit, i.e. a semibreve, is $4 \times 0.5 = 2$ sec. In general given a time signature n_1/n_2 and a metronome marking m BPM, we have that the beat duration is $d_{beat} = 60/m$ sec, the bar duration $d_{bar} = n_1 \times 60/m$ sec, and the musical unit duration $d_{unit} = n_2 \times 60/m$ sec.

9.1.5 Dynamics

In music, dynamics refers to the volume or loudness of the sound or note. The full terms for dynamics are sometimes written out, but mostly are expressed in symbols and abbreviations (see Table 7.2). There are also traditionally in Italian and will be found between the staves in piano music. In an orchestral score, they will usually be found next to the part to which they apply.

SYMBOL	TERM	MEANING
<i>pp</i>	pianissimo	very soft
<i>p</i>	piano	soft
<i>mp</i>	mezzopiano	medium soft
<i>mf</i>	mezzoforte	medium loud
<i>f</i>	forte	loud
<i>ff</i>	fortissimo	very loud

Table 9.2: Symbols for dynamics notation.

In addition, there are words used to indicate gradual changes in volume. The two most common are *crescendo*, sometimes abbreviated to *cresc*, meaning "get gradually louder"; and *decrescendo* or *diminuendo*, sometimes abbreviated to *decrec* and *dim* respectively, meaning "get gradually softer". These transitions are also indicated by wedge-shaped marks. For example, the notation in Fig. 7.9 indicates music starting moderately loud, then becoming gradually louder and then gradually quieter:



Figure 9.9: Dynamics notation indicating music starting moderately loud (mezzo forte), then becoming gradually louder (crescendo) and then gradually quieter (diminuendo).

9.1.6 Harmony

In music theory, harmony is the use and study of the relationship of tones as they sound simultaneously and the way such relationships are organized in time. It is sometimes referred to as the "vertical" aspect of music, with melody being the "horizontal" aspect. Very often, harmony is a result of counterpoint or polyphony, several melodic lines or motifs being played at once, though harmony may control the counterpoint. The term "chord" refers to three or more different notes or pitches sounding simultaneously, or nearly simultaneously, over a period of time.

Within a given key, chords can be constructed on each note of the scale by superimposing intervals of a major or minor third (four and three semitones, respectively), such as C-E-G giving the C major triad, or A-C-E giving the A minor triad. A harmonic hierarchy similar to the tonal hierarchy has been demonstrated for chords and cadences. The harmonic hierarchy orders the function of chords within a given key according to a hierarchy of structural importance. This gives rise to one of the particularly rich aspects of Western tonal music: harmonic progression. In the harmonic hierarchy, the tonic chord (built on the first degree of the scale) is the most important, followed by the dominant (built on the fifth degree) and the sub-dominant (built on the fourth degree). These are followed by the chords built on the other scale degrees. Less stable chords, that is those that have a lesser structural importance, have a



tendency in music to resolve to chords that are more stable. These movements are the basis of harmonic progression in tonal music and also create patterns of musical tension and relaxation. Moving to a less stable chord creates tension, while resolving toward a more stable chord relaxes that tension. Krumhansl has shown that the harmonic hierarchy can be predicted by the position in the tonal hierarchies of the notes that compose the chords (see sect. 7.3.3).

9.2 Organization of musical events

9.2.1 Musical form

We can compare a single sound, chord, cadence to a letter, a word, or a punctuation mark in language. In this section we will see how all these materials take formal shape and are used within the framework of a musical structure.

9.2.1.1 Low level musical structure

The bricks of music are its *motives*, the smallest unit of a musical composition. To be intelligible, a motive has to consist of at least two notes, and have a clearly recognizable rhythmic pattern, which gives it life. Usually a motive consists of few notes as for example the four notes at the beginning of Beethoven's Fifth Symphony. If you recall the continuation of the symphony, you realize that this motive is the foundation of the whole musical building. It is by means of motive and its development (e.g. repetition, transposition, modification, contrapuntal use, etc.) that a composer states, and subsequently explains his idea.

A *figure* figure is a recurring fragment or succession of notes that may be used to construct the accompaniment. A figure is distinguished from a motif in that a figure is background while a motif is foreground.

9.2.1.2 Mid and high level musical structure

A musical phrase can consist of one or more motives. The end is marked by a punctuation, e.g. a cadence. Phrases can be combined to form a period or sentence: i.e. a section of music that is relatively self-contained and coherent over a medium time scale. In common practice phrases are often four and most often eight bars, or measures, long.

The mid-level of musical structure is made up of sections of music. Periods combine to form larger sections of musical structure. The length of a section may vary from sixteen to thirty-two measures in length - often, sections are much longer. At the macro-level of musical structure exists the complete work formed of motives, phrases and sections.

9.2.1.3 Basic patterns

Repetition, variation and contrast may be seen as basic patterns. These patterns have been found to be effective at all levels of music structure, whether it be shorter melodic motives or extended musical compositions. These basic patterns may be found not only in all world musics, but also in the other arts and in the basic patterns of nature.

Repetition of the material of music plays a very important role in the composing of music and somewhat more than in other artistic media. If one looks at the component motives of any melody, the successive repetition of the motives becomes apparent. A melody tends to "wander" without repetition of its rhythmic and pitch components and repetition gives "identity" to musical materials



and ideas. Whole phrases and sections of music often repeat. Musical repetition has the form A A A A A A A etc..

Variation means change of material and may be slight or extensive. Variation is used to extend melodic, harmonic, dynamic and timbral material. Complete musical phrases are often varied. Musical variation has the form A A1 A2 A3 A4 A5 A6 etc..

Contrast is the introduction of new material in the structure or pattern of a composition of music that contrasts with the original material. Contrast extends the listener's interest in the musical "ideas" in a phrase or section of music. It is most often used in the latter areas of phrases or sections and becomes ineffective if introduced earlier. Musical contrast has the form A B C D E F G etc..

The patterns of repetition, variation, and contrast form the basis for the structural design of melodic material, the accompaniment to melodic material, and the structural relationships of phrases and sections of music. When these basic patterns are reflected in the larger sectional structure of complete works of music, this level of musical structure defines the larger sectional patterns of music.

9.2.1.4 Basic musical forms

Form in music refers to large and small sectional patterns resulting from a basic model. There are basic approaches to form in music found in cultures around the world. In most cases, the form of a piece should produce a balance between statement and restatement, unity and variety, contrast and connection. Throughout a given composition a composer may:

1. Present a melody and continually repeat it (A-A-A-A-A-A etc.),
2. Present a melody and continually vary it (A A1 A2 A3 A4 A5 etc.),
3. Present a series of different melodies (A-B-C-D-E-F-G etc.),
4. Alternate a repeating melody with other melodies (A-B-A-C-A-D-A-E-A etc.),
5. Present a melody and expand and/or modify it.

Binary form is a way of structuring a piece of music into two related sections, both of which are usually repeated. Binary form is usually characterized as having the form AB. When both sections repeat, a more accurate description would be AABB. Ternary form is a three part structure. The first and third parts are identical, or very nearly so, while the second part is sharply contrasting. For this reason, ternary form is often represented as ABA. Arch form is a sectional way of structuring a piece of music based on the repetition, in reverse order, of all or most musical sections such that the overall form is symmetrical, most often around a central movement. The sections need not be repeated verbatim but at least must share thematic material. It creates interest through an interplay among memory, variation, and progression. An example is A-B-C-D-C-B-A.

9.2.2 Cognitive processing of music information

Adapted from: Mc Adams, Audition: Cognitive Psychology of Music 1996

When we consider the perception of large scale structures like music, we need to call into play all kinds of relationships over very large time scales on the order of tens of minutes or even hours. It is thus of great interest to try to understand how larger scale temporal structures, such as music, are represented and processed by human listeners. These psychological mechanisms are necessary for the sense of global form that gives rise to expectancies that in turn may be the basis for affective and emotional responses to musical works. One of the main goals of auditory cognitive psychology is to understand how humans can "think in sound" outside the verbal domain. The cognitive point of view postulates internal (or mental) representations of abstract and specific properties of the musical sound environment, as well as



processes that operate on these representations. For example, sensory information related to frequency is transformed into pitch, is then categorized into a note value in a musical scale and then ultimately is transformed into a musical function within a given context.

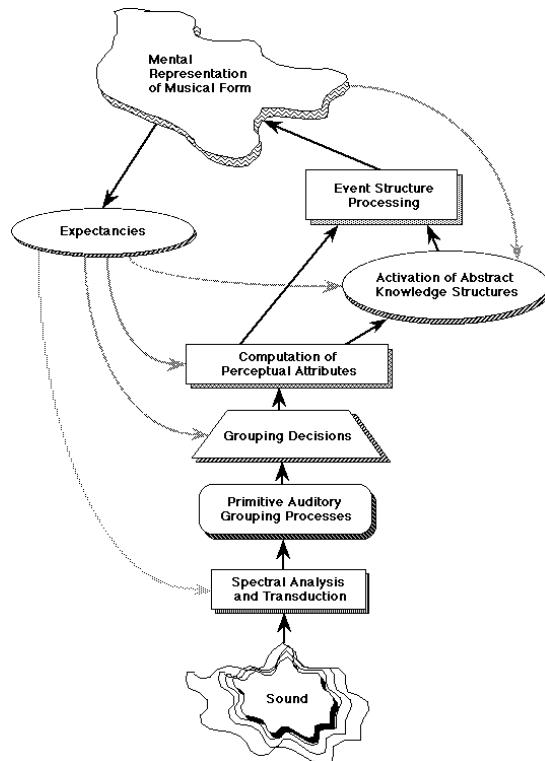


Figure 9.10: Schema illustrating the various aspects of musical information processing [from McAdams 1996].

The processing of musical information may be conceived globally as involving a number of different "stages" (Fig. 7.10). Following the spectral analysis and transduction of acoustic vibrations in the auditory nerve, the auditory system appears to employ a number of mechanisms (*primitive auditory grouping processes*) that organize the acoustic mixture arriving at the ears into mental "descriptions". These descriptions represent events produced by sound sources and their behaviour through time. Research has shown that the building of these descriptions is based on a limited number of acoustic cues that may reinforce one another or give conflicting evidence. This state of affairs suggests the existence of some kind of process (grouping decisions) that sorts out all of the available information and arrives at a representation of the events and sound sources that are present in the environment that is as unambiguous as possible. According to theory of auditory scene analysis, the *computation of perceptual attributes* of events and event sequences depends on how the acoustic information has been organized at an earlier stage. Attributes of individual musical events include pitch, loudness, and timbre, while those of musical event sequences include melodic contour, pitch intervals, and rhythmic pattern. Thus a composer's control of auditory organization by a judicious arrangement of notes can affect the perceptual result.

Once the information is organized into events and event streams, complete with their derived perceptual attributes, what is conventionally considered to be music perception begins.

- The auditory attributes activate *abstract knowledge structures* that represent in long-term memory the relations between events that have been encountered repeatedly through experience in a given cultural environment. That is, they encode various kinds of regularities experienced in the world.

Bregman (1993) has described regularities in the physical world and believes that their processing at the level of primitive auditory organization is probably to a large extent innate. There are, however, different kinds of relations that can be perceived among events: at the level of pitches, durations, timbres, and so on. These structures would therefore include knowledge of systems of pitch relations (such as scales and harmonies), temporal relations (such as rhythm and meter), and perhaps even timbre relations (derived from the kinds of instruments usually encountered, as well as their combinations). The sound structures to be found in various occidental cultures are not the same as those found in Korea, Central Africa or Indonesia, for example. Many of the relational systems have been shown to be hierarchical in nature.

- A further stage of processing (*event structure processing*) assembles the events into a structured mental representation of the musical form as understood up to that point by the listener. Particularly in Western tonal/metric music, hierarchical organization plays a strong role in the accumulation of a mental representation of musical form. At this point there is a strong convergence of rhythmic-metric and pitch structures in the elaboration of an event hierarchy in which certain events are perceived to be stronger, more important structurally, and more stable. The functional values that events and groups of events acquire within an event hierarchy generate perceptions of musical tension and relaxation or, in other words, musical movement. They also generate expectancies about where the music should be going in the near future based both on what has already happened and on abstract knowledge of habitual musical forms of the culture—even for pieces that one has never heard before. In a sense, we are oriented—by what has been heard and by what we “know” about the musical style—to expect a certain type of event to follow at certain pitches and at certain points in time.
- The *expectancies* drive and influence the activation of knowledge structures that affect the way we interpret subsequent sensory information. For example, we start to hear a certain number of pitches, a system of relations is evoked and we infer a certain key; we then expect that future information that comes in is going to conform to that key. A kind of loop of activity is set up, slowly building a mental representation that is limited in its detail by how much knowledge one actually has of the music being heard. It is also limited by one’s ability to represent things over the long term, which itself depends on the kind of acculturation and training one has had. It does not seem too extreme to imagine that a Western musician could build up a mental structure of much larger scale and greater detail when listening to a Mahler symphony that lasts one and half hours, than could a person who just walked out of the bush in Central Africa. The reverse would be true for the perception of complex Pygmy polyphonic forms. However, on the one hand we are capable of hearing and enjoying something new, suggesting that there may be inborn precursors to musical comprehension in all human beings that makes this possible. On the other hand, what we do hear and understand the first time we encounter a new musical culture is most likely not what a native of that culture experiences.

The expectancies generated by this accumulating representation can also affect the grouping decisions at the basic level of auditory information processing. This is very important because in music composition, by playing around with some of these processes, one can set up perceptual contexts that affect the way the listener will tend to organize new sensory information. This process involves what Bregman (1990) has called schema-driven processes of auditory organization.

While the nature and organization of these stages are probably similar across cultures in terms of the underlying perceptual and cognitive processing mechanisms involved, the “higher level” processes beyond computation of perceptual attributes depend quite strongly on experience and accumulated knowledge that is necessarily culture-specific.



9.2.3 Auditory grouping

Sounds and sound changes representing information must be capable of being detected by the listener. A particular configuration of sound parameters should convey consistent percept to the user. *Auditory grouping* studies the perceptual process by which the listener separates out the information from an acoustic signal into individual meaningful sounds (fig. 7.11).

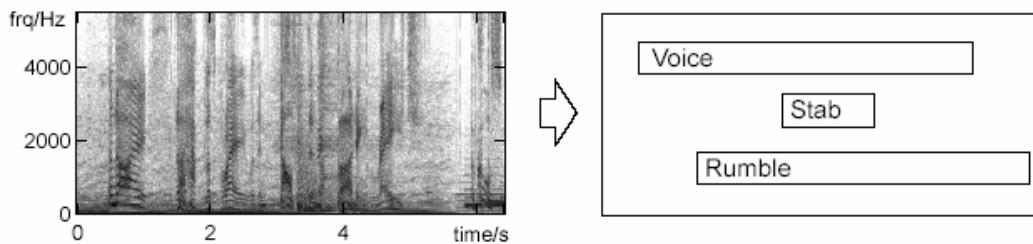


Figure 9.11: Auditory organization

The sounds entering our ears may come from a variety of sources. The auditory system is faced with the complex tasks of:

- Segregating those components of the combined sound that come from different sources.
- Grouping those components of the combined sound that come from the same source.

In hearing, we tend to organise sounds into auditory objects or streams. Bregman (1990) has termed this process Auditory Scene Analysis (fig. 7.12). It includes all the sequential and cross-spectral process which operate to assign relevant components of the signal to perceptual objects denoted *auditory streams*.

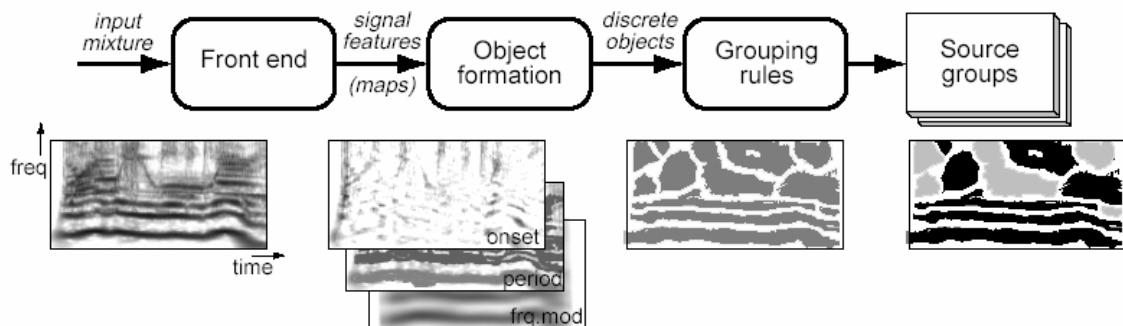


Figure 9.12: Auditory scene analysis

The brain needs to group simultaneously (separating out which frequency components that are present at a particular time have come from the same sound source) and also successively (deciding which group of components at one time is a continuation of a previous group). Some processes exclude part of the signal from a particular stream. Others help to bind each stream together.

A stream is

- a psychological organization with perceptual attributes that are not just the sum of the percept of its component but are dependent upon the configuration of the stream.

- a sequence of auditory events whose elements are related perceptually to one another, the stream being segregated from other co-occurring auditory events.
- A psychological organization whose function is to mentally represent the acoustic activity of a single source.

Auditory streaming is the formation of perceptually distinct apparent sound sources. Temporal order judgement is good within a stream but bad between streams. Examples include:

- implied polyphony,
- noise burst replacing a consonant in a sentence,
- click superimposed on a sentence or melody.

An auditory scene is the acoustic pressure wave carrying the combined evidence from all the sound sources present. Auditory scene analysis is the process of decoding the auditory scene, which occurs in auditory perception. Auditory Scene Analysis is a non-conscious process of guessing about "what's making the noise out there", but guessing in a way that fits consistently with the facts of the world. For example if a sound has a particular pitch, a listener will probably infer that any other sounds made by that sound source will be similar in pitch to the first sound, as well as similar in intensity, waveform, etc., and further infer that any sounds similar to the first are likely to come from the same location as the first sound. This fact can explain why we experience the sequence of pitches of a tune (Fig. 7.13) as a melody, pitch moving in time. Consecutive pitches in this melody are very close to each other in pitch-space, so on hearing the second pitch a listener will activate our Auditory Scene Analysis inference mechanisms, and assign it to the same source as the first pitch.



Figure 9.13: Score of *Frère Jacques*.

If the distance in pitch space had been large, they might have inferred that a second sound source existed, even although they knew that it's the same instrument that's making the sound - this inferred sound source would be a virtual rather than a real source. Hence a pattern such as shown in Figure 7.14(a), where successive notes are separated by large pitch jumps but alternate notes are close together in pitch, is probably heard as two separate and simultaneous melodies rather than one melody leaping around. This tendency to group together, to linearise, pitches that are close together in pitch-space and in time provides us with the basis for hearing a melody as a shape, as pitch moving in time, emanating from a single - real or virtual - source.

J. S. Bach used them frequently to conjure up the impression of compound, seemingly simultaneous, melodies even though only one single stream of notes is presented. For example, the pattern given in Figure 7.14(b) (from the Courante of Bach's First 'Cello Suite) can be performed on guitar on one string, yet at least two concurrent pitch patterns or streams will be heard - two auditory streams will be segregated (to use Bregman's terminology). We may distinguish analytic vs. synthetic listening. In synthetic perception the information is interpreted as generally as possible, e.g. hearing a room full of



Figure 9.14: (a) Pattern where successive notes are separated by large pitch jumps but alternate notes are close together in pitch, is probably heard as two separate and simultaneous melodies. (b) Excerpt from the Courante of Bach's First Cello Suite: two concurrent pitch patterns are heard.

voices. In analytic perception, the information is used to identify the components of the scene to finer levels, e.g. listening to a particular utterance in the crowded room. Interpretation of environmental sounds involves combining analytic and synthetic listening, e.g. hearing the message of a particular speaker.

Gestalt psychology theory offers an useful perspective for interpreting the auditory scene analysis behaviour.

9.2.4 Gestalt perception

Gestalt (pronounced G - e - sh - talt) psychology is a movement in experimental psychology that began just prior to World War I. It made important contributions to the study of visual perception and problem solving. The approach of Gestalt psychology has been extended to research in areas such as thinking, memory, and the nature of aesthetics. The word 'Gestalt' means 'form' or 'shape'.

The Gestalt approach emphasizes that we perceive objects as well-organized patterns rather than separate component parts. According to this approach, when we open our eyes we do not see fractional particles in disorder. Instead, we notice larger areas with defined shapes and patterns. The "whole" that we see is something that is more structured and cohesive than a group of separate particles. Gestalt theory states that perceptual elements are (in the process of perception) grouped together to form a single perceived whole (a gestalt).

The focal point of Gestalt theory is the idea of grouping, or how we tend to interpret a visual field or problem in a certain way. According to the Gestalt psychologists, the way that we perceive objects, both visual and auditory, is determined by certain principles (gestalt principles). These principles function so that our perceptual world is organised into the simplest pattern consistent with the sensory information and with our experience. The things that we see are organised into patterns or figures. In hearing, we tend to organise sounds into auditory objects or streams. Bregman (1990) has termed this process Auditory Scene Analysis.

The most important principles are

Proximity: components that are perceptually close to each other are more likely to be grouped together.

For example temporal proximity or frequency proximity. The principle of proximity refers to distances between auditory features with respect to their onsets, pitch, and loudness. Features that are grouped together have a small distance between each other, and a long distance to elements of another group. Tones close in frequency will group together, so as to minimize the extent of frequency jumps and the number of streams. Tones with similar timbre will tend to group together. Speech sounds of similar pitch will tend to be heard from the same speaker. Sounds from different locations are harder to group together across time than those from the same location.

The importance of pitch proximity in audition is reflected in the fact that melodies all over the

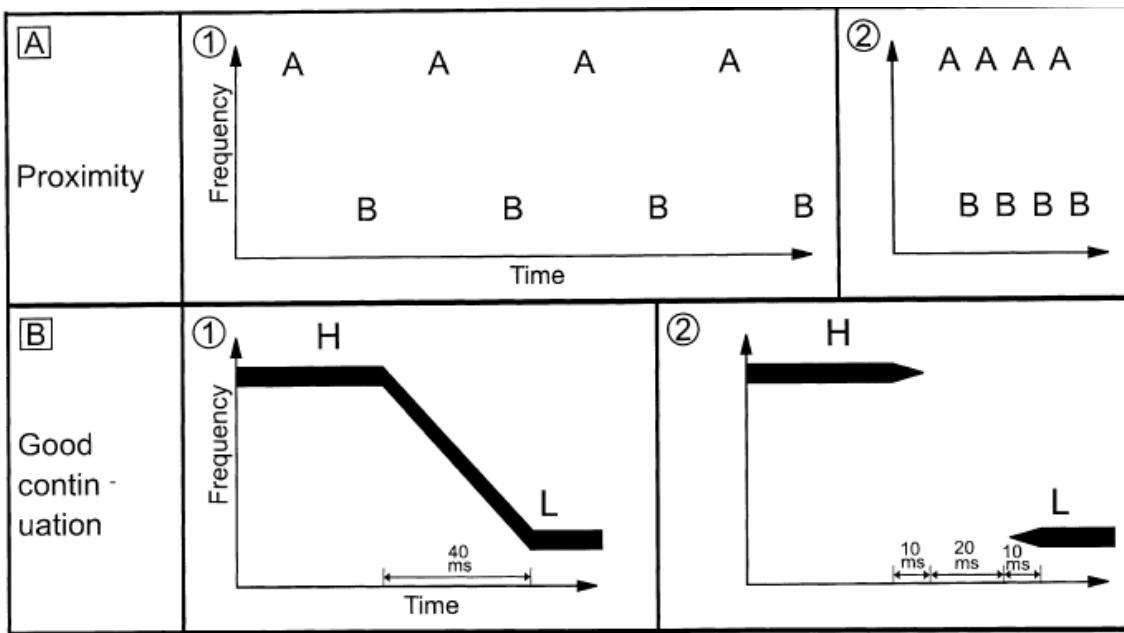


Figure 9.15: Experiments of Proximity and Good Continuation

world use small pitch intervals from note to note. Violations of proximity have been used in various periods and genres of both Western and non-Western music for a variety of effects. For example, fission based on pitch proximity was used to enrich the texture so that out of a single succession of notes, two melodic lines could be heard. Temporal and pitch proximity are competitive criteria, e.g. the slow sequence of notes A B A B . . . (figure 7.15, A1), which contains large pitch jumps, is perceived as one stream. The same sequence of notes played very fast (figure 7.15, A2) produces one perceptual stream consisting of As and another one consisting of Bs. A visual example is given in figure 7.17: the arrangement of points is not seen as a set of rows but rather a set of columns. We tend to perceive items that are near each other as groups.

Similarity: components which share the same attributes are perceived as related or as a whole. E.g. colour or form, in visual perception or common onset, common offset, common frequency, common frequency modulation, common amplitude modulation in auditory perception. For example one can follow the piano part in a group of instruments by following the sounds that have the timbre consistent with that of a piano. One can perceptually segregate one speaker's voice from those of others by following the pitch of the voice. Similarity is very similar to proximity, but refers to properties of a sound, which cannot be easily identified with a single physical dimension, like timbre.

A visual example is given in figure 7.18: things which share visual characteristics such as shape, size, color, texture, value or orientation will be seen as belonging together. In the example of 7.18(a), the two filled lines gives our eyes the impression of two horizontal lines, even though all the circles are equidistant from each other. In the example of 7.18(b), the larger circles appear to belong together because of the similarity in size.

Another visual example is given in figure 7.19: So in the graphic on the left you probably see an X of fir trees against a background of the others; in the graphic on the right you may see a square of the other trees, partly surrounded by fir trees. The fact that in one we see an X and in the other a square is, incidentally, an example of good form or pragnanz principle, stating that psychological

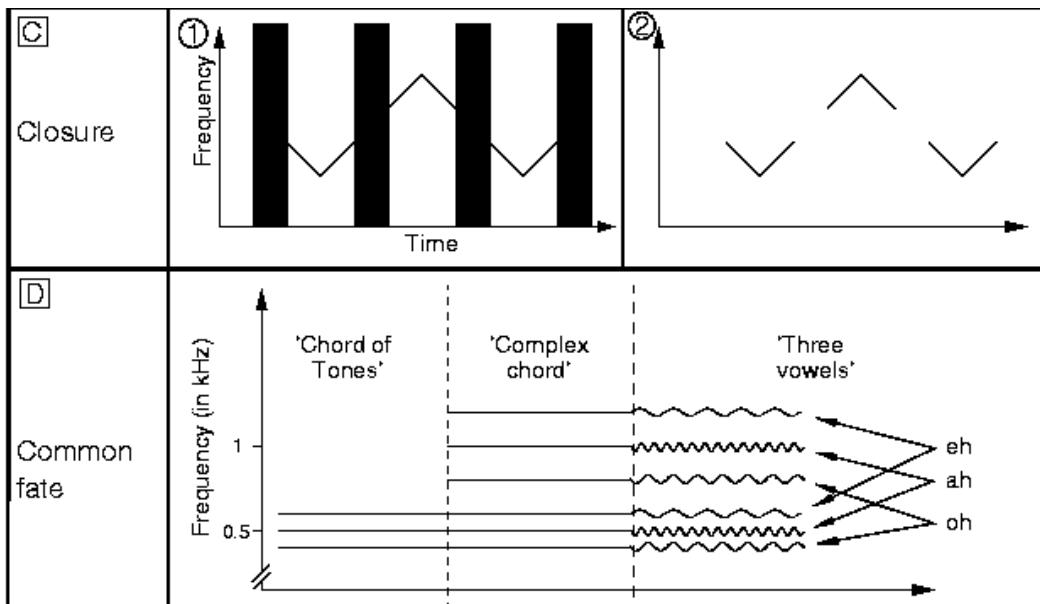


Figure 9.16: Experiments of Closure and Common Fate

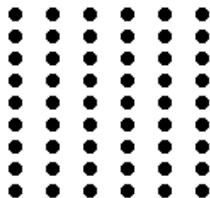


Figure 9.17: Example of proximity gestalt rule

organization will always be as 'good' as prevailing conditions allow. For Gestalt psychologists form is the primitive unit of perception. When we perceive, we will always pick out form.

Good continuation: Components that display smooth transitions from one state to another are perceived as related. Examples of smooth transitions are: proximity in time of offset of one component with onset of another; frequency proximity of consecutive components; constant glide trajectory of consecutive components; smooth transition from one state to another state for the same parameter. For example an abrupt change in the pitch of a voice produces the illusion that a different speaker has interrupted the original. The perception appears to depend on whether or not the intonation contour changes in a natural way. Sound that is interrupted by a noise that masks it, can appear to be continuous. Alternations of sound and mask can give the illusion of continuity with the auditory system interpolating across the mask.

In figure 7.15, B), high (H) and low (L) tones alternate. If the notes are connected by glissandi (figure 7.15, B1), both tones are grouped to a single stream. If high and low notes remain unconnected (figure 1, B2), Hs and Ls each group to a separate stream.

A visual example is given in figure 7.20. The law of good continuation states that objects arranged in either a straight line or a smooth curve tend to be seen as a unit. In figure 7.20(a) we distinguish two lines, one from **a** to **b** and another from **c** to **d**, even though this graphic could represent another set of lines, one from **a** to **d** and the other from **c** to **b**. Nevertheless, we are more likely to identify

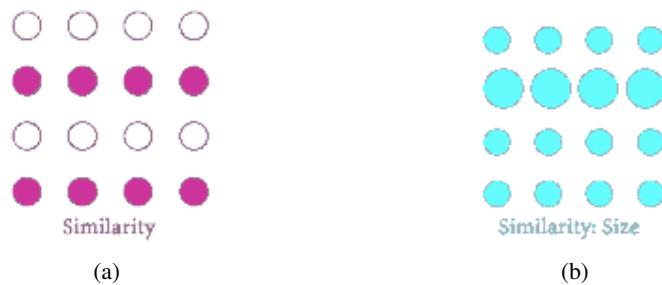


Figure 9.18: Example of similarity gestalt grouping principle.



Figure 9.19: Example of similarity gestalt grouping principle.

line **a** to **b**, which has better continuation than the line from **a** to **d**, which has an obvious turn. In figure 7.20(b) we perceive the figure as two crossed lines instead of 4 lines meeting at the centre.

Common Fate Sounds will tend to be grouped together if they vary together over time. Differences in onset and offset in particular are very strong grouping cues. Also, sounds that are modulated together (amplitude or frequency modulation) tend to be grouped together. The principle 'common fate' groups frequency components together, when similar changes occur synchronously, e.g. synchronous onsets, glides, or vibrato.

Chowning (Fig. 7.16, D) made the following experiment: First three pure tones are played. A chord is heard, containing the three pitches. Then the full set of harmonics for three vowels (/oh/, /ah/, and /eh/) is added, with the given frequencies as fundamental frequencies, but without frequency fluctuations. This is not heard as a mixture of voices but as a complex sound in which the three pitches are not clear. Finally, the three sets of harmonics are differentiated from one another by their patterns of fluctuation. We then hear three vocal sounds being sung at three different pitches.

Closure This principle is the tendency to perceive things as continuous even though they may be discontinuous. If the gaps in a sound are filled in with another more intense sound, the original sound may be perceived as being continuous. For example, if part of a sentence is replaced by the sound of a door slam, the speaker's voice may be perceived as being continuous (continuing through the door slam). The principle of closure completes fragmentary features, which already have a 'good Gestalt'. E.g. ascending and descending glissandi are interrupted by rests (Fig. 7.16, C2). Three temporally separated lines are heard one after the other. Then noise is added during the rests (Fig. 7.16 C1). This noise is so loud, that it would mask the glissando, unless it would be interrupted by rests. Amazingly the interrupted glissandi are perceived as being continuous. They have 'good Gestalt': They are proximate in frequency before and after the rests. So they can easily

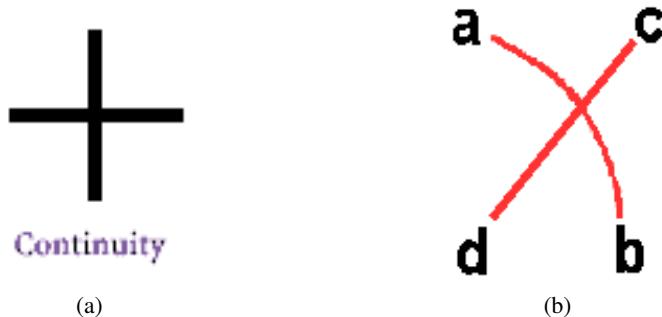


Figure 9.20: Examples of good continuation gestalt grouping principle.

be completed by a perceived good continuation. This completion can be understood as an auditory compensation for masking.



Figure 9.21: Example of closure.

Figure / Ground It is usual to perceive one sound source as the principal sound source to which one is attending, and relegate all other sounds to be background. We may switch our attention from one sound source to another quite easily. What was once figure (the sound to which we were attending) may now become ground (the background sound). An important topics in auditory perception are attention and learning. In a cocktail party environment, we can focus on one speaker. Our attention selects this stream. Also, whenever some aspect of a sound changes, while the rest remains relatively unchanging, then that aspect is drawn to the listener's attention ('figure ground phenomenon'). Let us give an example for learning: The perceived illusory continuity (see Fig. 7.16, C) of a tune through an interrupting noise is even stronger, when the tune is more familiar.

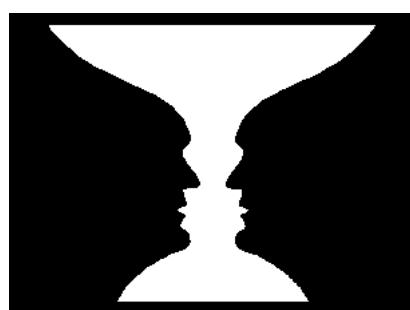


Figure 9.22: Rubin vase: example of figure/ground principle.

The Rubin vase shown in Fig. 7.22 is an example of this tendency to pick out form. We don't simply see black and white shapes - we see two faces and a vase. The problem here is that we see the two forms of equal importance. If the source of this message wants us to perceive a vase, then the vase is the intended figure and the black background is the ground. The problem here is a confusion of figure and ground. A similar everyday example is:

- an attractive presenter appears with a product; she is wearing a 'conservative' dress; eye-tracking studies show substantial attention to the product; three days later, brand-name recall is high;
- an attractive presenter appears with a product; she is wearing a 'revealing' dress; eye-tracking shows most attention on the presenter; brand-name recall is low.



Figure 9.23: Horses by M. Escher. An artistic example of figure and ground interchange.

Escher often designed art which played around with figure and ground in interesting ways. Look at how figure and ground interchange in fig. 7.23. Do you see the white horses and riders? Now look for the black horses and riders.

Gestalt grouping laws do not seem to act independently. Instead, they appear to influence each other, so that the final perception is a combination of all of the Gestalt grouping laws acting together. Gestalt theory applies to all aspects of human learning, although it applies most directly to perception and problem-solving.

9.3 Basic algorithms for melody processing

9.3.1 Melody

Melody may be defined as a series of individual musical events one occurring after another in order so that the composite order constitute a recognizable entity. The essential elements of any melody are duration, pitch, and sound quality (e.g. timbre, texture, and loudness). It represents the linear or horizontal aspect of music and should not be confused with harmony, which is the vertical aspect of music.

9.3.1.1 Melody representation: melodic contour

Contour may be defined as the general shape of an object, often, but not exclusively, associated with elevation or height, as a function of distance, length, or time. In music, contour can be a useful tool

for the study of the general shape of a musical passage. A number of theories have been developed that use the rise and fall of pitch level, changes in rhythmic patterns or changes in dynamics as a function of time (or temporal order) to compare or contrast musical passages within a single composition or between compositions of a single composer. One application of the melodic contour is finding out whether the sequence contains repeated melodic phrases. This can be done using computing the autocorrelation.

Parsons showed that encoding a melody by using only the direction of pitch intervals can still provide enough information for distinguishing between a large number of tunes. In Parsons code for melodic contours, each pair of consecutive notes is coded as "U" ("up") if the second note is higher than the first note, "R" ("repeat") if the pitches are equal, and "D" ("down") otherwise. Rhythm is completely ignored. Thus, the first theme from Beethoven's 8th symphony (Fig. 7.24) would be coded D U U D D D U R D R U U U U. Note that the first note of any tune is used only as a reference point and does not show up explicitly in the Parsons code. Often an asterisk (*) is used in the Parsons code field for the first note. A more precise and effective way of representing contours employs 5-level quantization (+++, +, 0, -, -) distinguishing between small intervals (steps), which are 1 or 2 semitones wide, from larger intervals (leaps), which are at least 3 semitones wide. The symbols (+++, +, 0, -, -) are used to code this representation. For example the Beethoven's theme of Fig. 7.24 will be coded as - + + - - + + 0 - 0 + + + +.



Figure 9.24: Melodic contour and Parson code.

In MPEG-7 the *Melody Contour DS* uses a 5-step contour (representing the interval difference between adjacent notes), in which intervals are quantized. The Melody Contour DS also represents basic rhythmic information by storing the number of the nearest whole beat of each note, which can dramatically increase the accuracy of matches to a query.

For applications requiring greater descriptive precision or reconstruction of a given melody, the *Melody DS* supports an expanded descriptor set and high precision of interval encoding. Rather than quantizing to one of five levels, the precise pitch interval (to cent or greater precision) between notes is kept. Precise rhythmic information is kept by encoding the logarithmic ratio of differences between the onsets of notes in a manner similar to the pitch interval.

9.3.1.2 Similarity measures

When we want to compare melodies, a computable similarity measure is need. The measures can roughly be classified in three categories: Vector measures, symbolic measures and musical (mixed) measures, according to the computational algorithm used.

- The vector measure treat the transformed melodies as vectors in a suitable real vector space, where methods like scalar products and other means of correlation can be applied to.
- On the contrary the symbolic measures treat the melodies as strings, i.e., sequences of symbols, where well-known measures like Edit Distance or n-gram-related measures can be used.
- The musical or mixed measures typically involve more or less specific musical knowledge and the computation can be from either the vector or the symbolical or even completely different ways like scoring models.

The distance can be computed on different representations of the melodies (e.g. the melody itself, its contour), or some statistic distributions (e.g. pitch classes, pitch class transitions, intervals, interval transitions, note durations, note duration transitions)

9.3.1.3 Edit distance

Approximate string pattern matching is based on the concept of *edit distance*. The edit distance $D(A, B)$ between string $A = a_1, \dots, a_m$ and $B = b_1, \dots, b_n$ is the minimum number of editing operations required to transform string A into string B , where an operation is an insertion, deletion, or substitution of a single character. The special case in which deletions and insertions are not allowed is called the Hamming distance.

We can define recursively the (edit) distance $d[i, j]$ for going from string $A[1..i]$ to string $B[1..j]$ as

$$d[i, j] = \min \begin{cases} d[i - 1, j] + w(a_i, 0), & //\text{deletion of } a_i \\ d[i, j - 1] + w(0, b_j), & //\text{insertion of } b_j \\ d[i - 1, j - 1] + w(a_i, b_j) & //\text{match or change} \end{cases} \quad (9.8)$$

where $w(a_i, 0)$ is the weight associated with the deletion of a_i , $w(0, b_j)$ is the weight for insertion of a_i , and $w(a_i, b_j)$ is the weight for replacement of element i of sequence A by element j of sequence B . The operation titled "match/change" sets $w(a_i, b_j) = 0$ if $a_i = b_j$ and a value greater than 0 if $a_i \neq b_j$. Often the weights used are 1 for insertion, deletion and substitution(change) and 0 for match. The initial conditions are given by $d[0, 0] = 0$, $d[i, 0] = d[i - 1, j] + w(a_i, 0)$ for $i \geq 1$ and $d[0, j] = d[0, j - 1] + w(0, b_j)$ for $j \geq 1$.

The edit distance $D(A, B) = d[n, m]$ can be computed by dynamic programming with running time $O(n \cdot m)$ with the algorithm given in Fig. 7.25.

```
Algorithm EditDistance ( $A[1..m], B[1..n,], w_{del}, w_{ins}, w_{sub}$ )
for i from 0 to m
    d[i, 0] := i ·  $w_{del}$ 
for j from 0 to n
    d[0, j] := j ·  $w_{ins}$ 

for i from 1 to m
    for j from 1 to n
        if A[i] = B[j]
            then cost := 0
            else cost :=  $w_{sub}$ 
        d[i, j] := min( d[i-1, j]+ $w_{del}$ , d[i, j-1]+ $w_{ins}$ , d[i-1, j-1]+cost )
return d[m, n]
```

Figure 9.25: Dynamic programming algorithm for computing EditDistance.

9.3.2 Melody segmentation

Generally a piece of music can be divided into section and segments at different level. The term grouping describe the general process of segmentation at all levels. Grouping in music is a complex matter. Most



computational approaches focused on low-level grouping structure. Grouping events together involves storing them in memory as a larger unit, which is encoded to aid further cognitive processing. Indeed grouping structure plays an important role in recognition of repeated patterns in music. Notice that also the metric structure organize the events on time. However meter involves a framework of level of beats and in itself implies no segmentation; grouping is merely a segmentation without accentural implications.

9.3.2.1 Gestalt based segmentation

Tenny and Polansky proposed a model for small-level grouping in monophonic melodies based on Gestalt rules of proximity (i.e. the preference for grouping boundaries at long intervals between onsets) and similarity (i.e. the preference for grouping boundaries at changes in pitch and dynamics). Moreover the boundary value depends on the context. Thus an interval value in some parameter tends to be a grouping boundary if it is a local maximum, i.e. if it is larger the values immediately preceding and following it. In order to combine the differences of all parameters in a single measure the L_1 norm is proposed, i.e. the absolute values are summed. The algorithm proceeds in this way:

Algorithm TenneyLLgrouping

1. Given a sequence of n tones with pitch $p[i]$ and IOI $ioi[i]$, for $i = 1, \dots, n$
2. **for** $i = 1$ **to** $n - 1$
Compute the distance $d[i]$ between event i and $i + 1$ as
$$d[i] = ioi[i] + |p[i + 1] - p[i]|$$
3. **for** $i = 2$ **to** $n - 2$
if $d[i - 1] < d[i] > d[i + 1]$ then i is a low-level boundary point, and $i + 1$ is the starting point of a new group.

For higher level grouping the changes perceived at the boundary are taken into account. In order to deal with this, a distinction is made between mean-intervals and boundary-intervals as follows:

- A mean-interval between two groups is the difference between their mean values in that parameter. For the time parameter, the difference of their starting time is considered.
- A boundary-interval is the difference the values of the final component of the first group and the initial component of the second group

The mean-distance between two groups is a weighted sum of the mean-intervals between them, and the boundary-distance is given by a weighted sum of the boundary-intervals between them. Finally the disjunction between two groups is a weighted sum of mean-distance and boundary-distance between them. As a conclusion a group at a higher level will be initiated whenever a group occurs whose disjunction is greater than those immediately preceding and following it.

The algorithm proceeds in the following way:

Algorithm TenneyHLgrouping

1. for every group k , the mean pitch is computed by weighting the pitches with the durations

$$mean_p[k] = \frac{\sum_j p[j] \cdot dur[j]}{\sum_j dur[j]}$$

where in the summations, j spans all the events in group k .



2. compute the mean-distance

$$\text{mean_dist}[k] = |\text{mean}_p[k+1] - \text{mean}_p[k]| + (\text{onset}[k+1] - \text{onset}[k])$$

3. compute the boundary-distance

$$\text{boundary_dist}[k] = |p[\text{first}[k+1]] - p[\text{last}[k]]| + (\text{onset}[k+1] - \text{onset}[k])$$

where $\text{first}[k]$ and $\text{last}[k]$ are the indexes of the first and last note of group k and $\text{onset}[k] = \text{onset}[\text{first}[k]]$.

4. compute the disjunction by

$$\text{disj}[k] = w_{md} \cdot \text{mean_dist}[k] + w_{bd} \cdot \text{boundary_dist}[k]$$

5. if $\text{disj}[k-1] < \text{disj}[k] > \text{disj}[k+1]$ then the k -th group is the starting point of a new higher-level segment.

9.3.2.2 Local Boundary Detection Model (LBDM)

In this section, a computational model (developed by Emilios Cambouropoulos 2001), that enables the detection of local melodic boundaries will be described. This model is simpler and more general than other models based on a limited set of rules (e.g. implication realization model seen in sect. 7.6.2) and can be applied both to quantised score and non-quantised performance data.

The Local Boundary Detection Model (LBDM) calculates boundary strength values for each interval of a melodic surface according to the strength of local discontinuities; peaks in the resulting sequence of boundary strengths are taken to be potential local boundaries.

The model is based on two rules: the Change rule and the Proximity rule. The Change rule is more elementary than any of the Gestalt principles as it can be applied to a minimum of two entities (i.e. two entities can be judged to be different by a certain degree) whereas the Proximity rule requires at least three entities (i.e. two entities are closer or more similar than two other entities).

- Change Rule (CR): Boundary strengths proportional to the degree of change between two consecutive intervals are introduced on either of the two intervals (if both intervals are identical no boundary is suggested).
- Proximity Rule (PR): If two consecutive intervals are different, the boundary introduced on the larger interval is proportionally stronger.

The Change Rule assigns boundaries to intervals with strength proportional to a degree of change function S_i (described below) between neighbouring consecutive interval pairs. Then a Proximity Rule scales the previous boundaries proportionally to the size of the interval and can be implemented simply by multiplying the degree-of-change value with the absolute value of each pitch/time/dynamic interval. This way, not only relatively greater neighbouring intervals get proportionally higher values but also greater intervals get higher values in absolute terms - i.e. if in two cases the degree of change is equal, such as sixteenth/eighth and quarter/half note durations, the boundary value on the (longer) half note will be overall greater than the corresponding eighth note.

The aim is to develop a formal theory that may suggest all the possible points for local grouping boundaries on a musical surface with various degrees of prominence attached to them rather than a theory that suggests some prominent boundaries based on a restricted set of heuristic rules. The discovered





Figure 9.26: Beginning of *Frère Jacques*. Higher-level grouping principles override some of the local detail grouping boundaries (note that LBDM gives local values at the boundaries suggested by parallelism - without taking in account articulation).

boundaries are only seen as potential boundaries as one has to bear in mind that musically interesting groups can be defined only in conjunction with higher-level grouping analysis (parallelism, symmetry, etc.). Low-level grouping boundaries may be coupled with higher-level theories so as to produce optimal segmentations (see fig. 7.26).

In the description of the algorithm only the pitch, *IOI* and rest parametric profiles of a melody are mentioned. It is possible, however, to construct profiles for dynamic intervals (e.g. velocity differences) or for harmonic intervals (distances between successive chords) and any other parameter relevant for the description of melodies. Such distances can also be asymmetric; for instance the dynamic interval between *p* and *f* should be greater than between *f* and *p*.

Local Boundary Detection algorithm description Given a melodic sequence of n tones, where the i -th tone is represented by pitch $p[i]$, onset $on[i]$, offset $off[i]$.

A melodic sequence is converted into a number of independent parametric interval profiles P_k for the parameters: pitch (pitch intervals), *ioi* (interonset intervals) and rest (rests - calculated as the interval between current onset with previous offset). Pitch intervals can be measured in semitones, and time intervals (for *IOIs* and rests) in milliseconds or quantised numerical duration values. Upper thresholds for the maximum allowed intervals should be set, such as the whole note duration for *IOIs* and rests and the octave for pitch intervals; intervals that exceed the threshold are truncated to the maximum value. Thus we have

Algorithm LBDM

1. Given: pitch $p[i]$, onset $on[i]$, offset $off[i]$ for $i = 1, \dots, n$.
2. Compute the pitch profile P_p as $P_p[i] = |p[i+1] - p[i]|$ with $i = 1, \dots, n-1$.
3. Compute the *IOI* profile P_{IOI} as $P_{IOI}[i] = |on[i+1] - on[i]|$ with $i = 1, \dots, n-1$.
4. Compute the rest profile P_r as $P_r[i] = \max(0; on[i+1] - off[i])$ with $i = 1, \dots, n-1$.
5. for each profile P_k ,
compute the strength sequence S_k with algorithm `ProfileStrength`
6. Compute the boundary strength sequence LB as a weighted average of the individual strength sequences S_k . I.e.

$$LB[i] = w_{pitch}S_p[i] + w_{ioi}S_{IOI}[i] + w_{rest}S_r[i].$$
7. Local peaks in this overall strength sequence LB indicate local boundaries.

The suggested weights for the three different parameters are $w_{pitch} = w_{rest} = 0.25$ and $w_{ioi} = 0.50$.

In order to compute the profile strength the following algorithm is used.

Algorithm ProfileStrength

1. Given the parametric profile $P_k = [x[1], \dots, x[n - 1]]$
2. Compute the degree of change $r[i]$ between two successive interval values x_i and $x[i + 1]$ by:

$$r[i] = \frac{|x[i] - x[i + 1]|}{x[i] + x[i + 1]}$$

if $x[i] + x[i + 1] \neq 0$ and $x[i], x[i + 1] \geq 0$; otherwise $r[i] = 0$.

3. Compute the strength of the boundary $s[i]$ for interval $x[i]$ which is affected by both the degree of change to the preceding and following intervals, and is given by the function:

$$s[i] = x[i] \cdot (r[i - 1] + r[i])$$

4. Normalise the strength sequence in the range $[0, 1]$, by computing $s[i] = s[i] / \max_j(s[j])$
5. Return the sequence $S = \{s[2], \dots, s[n - 1]\}$

9.3.3 Tonality: Key finding

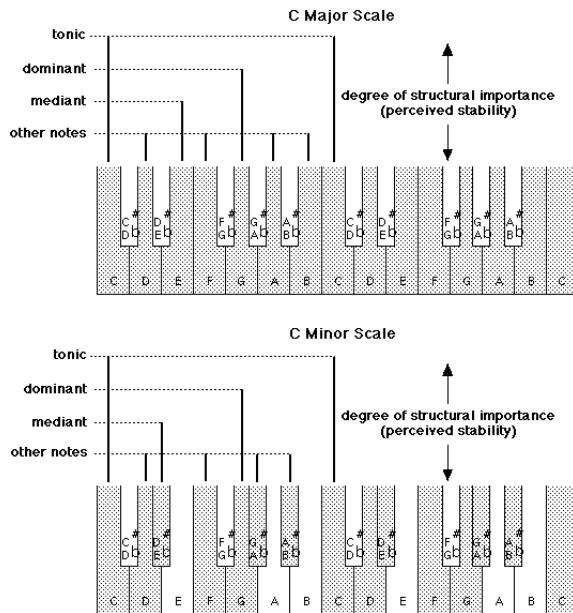


Figure 9.27: Piano keyboard representation of the scales of C major and C minor. Notes in each scale are shaded. The relative importance of the first (tonic - C), fifth (dominant - G) and third (mediant - E) degrees of the scale is illustrated by the length of the vertical bars. The other notes of the scale are more or less equally important followed by the chromatic notes that are not in the scale (unshaded) [from McAdams 1996].



This book is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 license,

©2005-2018 by the authors except for paragraphs labeled as adapted from <reference>

In the Western tonal pitch system, some pitches and chords, such as those related to the first and fifth degrees of the scale (C and G are the tonic and dominant notes of the key of C major, for example) are structurally more important than others (Fig. 7.27). This hierarchization gives rise to a sense of key. In fact when chords are generated by playing several pitches at once, the chord that is considered to be most stable within a key, and in a certain sense to "represent" the key, comprises the first, third and fifth degrees of the scale. In tonal music, one can establish a sense of key within a given major or minor scale and then move progressively to a new key (a process called modulation) by introducing notes from the new key and no longer playing those from the original key that are not present in the new key.

Factors other than the simple logarithmic distance between pitches affect the degree to which they are perceived as being related within a musical system. The probe tone technique developed by Krumhansl has been quite useful in establishing the psychological reality of the hierarchy of relations among pitches at the level of notes, chords, and keys. In this paradigm, some kind of musical context is established by a scale, chord, melody or chord progression, and then a probe stimulus is presented. Listeners are asked to rate numerically either the degree to which a single probe tone or chord fits with the preceding context or the degree to which two notes or chords seem related within the preceding context. This technique explores the listener's implicit comprehension of the function of the notes, chords, and keys in the context of Western tonal music without requiring them to explicate the nature of the relations.

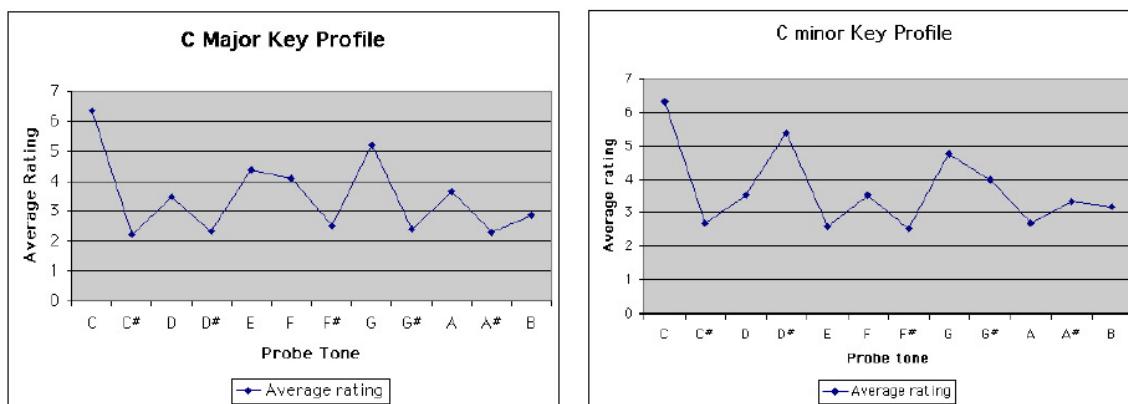


Figure 9.28: C Major and C minor profiles derived with the probe-tone technique from fittingness ratings by musician listeners.

If we present a context, such as a C major or C minor scale, followed by a single probe tone that is varied across the range of chromatic scale notes on a trial-to-trial basis, a rating profile of the degree to which each pitch fits within the context is obtained. This quantitative profile, when derived from ratings by musician listeners, fits very closely to what has been described intuitively and qualitatively by music theorists (Fig. 7.28). Note the importance of the tonic note that gives its name to the scale, followed by the dominant or fifth degree and then the mediant or third degree. These three notes form the principal triad or chord of the diatonic scale. The other notes of the scale are of lesser importance followed by the remaining chromatic notes that are not within the scale. These profiles differ for musicians and non-musicians. In the latter case the hierarchical structure is less rich and can even be reduced to a simple proximity relation between the probe tone and the last note of the context.

Krumhansl has shown (fig. 7.29) that the hierarchy of tonal importance revealed by these profiles is strongly correlated with the frequency of occurrence of notes within a given tonality (the tonic appears more often than the fifth than the third, and so on). It also correlates with various measures of tonal consonance of notes with the tonic, as well as with statistical measures such as the mean duration given these notes in a piece of music (the tonic often having the longest duration).

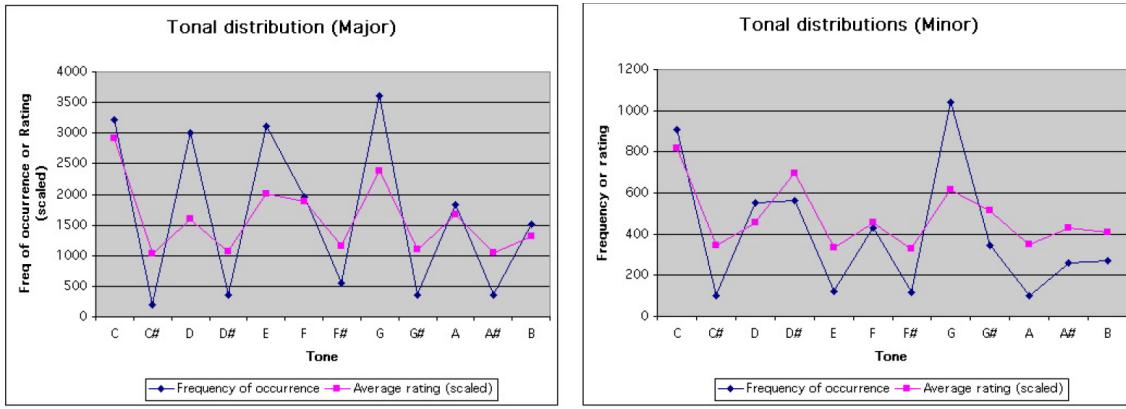


Figure 9.29: Comparison between tonal hierarchies and statistical distribution of tones in tonal works. It is shown the frequency of occurrence of each of the 12 chromatic scale tones in various songs and other vocal works by Schubert, Mendelssohn, Schumann, Mozart, Richard Strauss and J. A. Hasse. and the key profile (scaled).

9.3.3.1 Key finding algorithm

These correlations are the base of the classic key finding algorithm of Krumhansl-Schmuckler (as explained in Krumhansl's book Cognitive Foundations of Musical Pitch [Oxford University Press, 1990]). Each key has a key-profile: a vector representing the optimal distribution of pitch-classes for that key. The KSkeyFinding algorithm works as follows.

Algorithm KSkeyFinding

- Given a music segment of n tones, with pitch $p[i]$, duration $dur[i]$, for $i = 1, \dots, n$.
- Given the key profiles, 12 for major key and 12 for minor key
- Compute the pitch class distribution vector $pcd[0..11]$, taking into account the tone duration with:

```

for i from 1 to n
    pcd[i] = 0
for i from 1 to n
    pc = p[i] mod 12
    pcd[pc] = pcd[pc] + dur[i]
  
```

- Compute correlations of for all 24 major and minor pitch-class keys
- Assume that the estimated key for the passage is given by the largest positive correlation.

In this method, the input vector for a segment represents the total duration of each pitch-class in the segment. The match between the input vector and each key-profile is calculated using the standard correlation formula.

For example, if we take opening bar of Yankee Doodle, as shown in fig. 7.30, we find that: the sum of the durations of the G naturals gives .75 of a minim, the durations of the B naturals add up to half a minim, the durations of the A naturals add up to half a minim and there is one quaver D natural. We can then draw a graph showing the durations of the various pitch classes within the passage being analysed,





Figure 9.30: Example of Krumhansl-Schmuckler key finding algorithm: opening bar of Yankee Doodle.

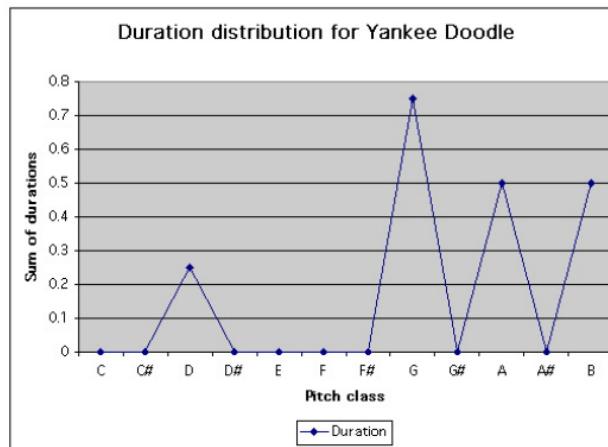


Figure 9.31: Example of Krumhansl-Schmuckler key finding algorithm: duration distribution of Yankee Doodle.

as shown in fig 7.31. The next step in the algorithm is to calculate the correlation between this graph and each of the 24 major and minor key profiles. This table (tab. 7.3) shows the correlation between this graph showing the durations of the various pitches in the Yankee Doodle excerpt and each of the major and minor key profiles. The algorithm then predicts that the perceived key will be the one whose profile best correlates with the graph showing the distribution of tone durations for the passage. So in this case, the algorithm correctly predicts that the key of Yankee Doodle is G major.

A variation of the key finding algorithm is proposed in Temperley 2001 (KSTkeyFinding algorithm). In this method, the input vector for a segment simply has 1 for a pitch-class if it is present at all in the segment (the duration and number of occurrences of the pitch-class are ignored) and 0 if it is not; the score for a key is given by the sum of the products of key-profile values and corresponding input vector values (which amounts to summing the key-profile values for all pitch class present in the segment). Moreover the key profiles were heuristically adjusted and are given in Table 7.4. Notice that given a C major key profile, the other major key profiles can be simply obtained by acyclical shift, and in a similar way all the minor key profiles can be obtained from the Cminor key profile.

The KSTkeyFinding algorithm works as follows.

Algorithm KSTkeyFinding

1. Given a music segment of n tones, with pitch $p[i]$, for $i = 1, \dots, n$.
2. Given the (modified) key profiles, 12 for major key and 12 for minor key
3. Compute the pitch class vector pv , where $pv[k] = 1$ if pitch class k is present in the music segment, else $pv[k] = 0$. I.e.

for k **from** 0 **to** 11

Key	Score	Key	Score
C major	0.274	C minor	-0.013
C sharp major	-0.559	C sharp minor	-0.332
D major	0.543	D minor	0.149
E flat major	-0.130	E flat minor	-0.398
E major	-0.001	E minor	0.447
F major	0.003	F minor	-0.431
F sharp major	-0.381	F sharp minor	0.012
G major	0.777	G minor	0.443
A flat major	-0.487	A flat minor	-0.106
A major	0.177	A minor	0.251
B flat major	-0.146	B flat minor	-0.513
B major	-0.069	B minor	0.491

Table 9.3: Correlation between the graph showing the durations of the various pitches in the Yankee Doodle excerpt and each of the major and minor key profiles.

C key note names	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
major key	5	2	3.5	2	4.5	4	2	4.5	2	3.5	1.5	4
minor key	5	2	3.5	4.5	2	4	2	4.5	3.5	2	1.5	4

Table 9.4: Temperley key profiles. The note names refer to C major and C minor key.

```

 $pv[k] = 0$ 
for  $i$  from 1 to  $n$ 
   $pv[p[i]] = 1$ 

```

4. **for all** 24 major and minor key profiles,
Compute the scalar product of pv with the key profile vector kp as

$$\sum_j pv[j] \cdot kp[j]$$

5. Assume that the estimated key for the passage is given by the largest positive scalar product.

9.3.3.2 Modulation

The key finding algorithms produce a single key judgement for a passage of music. However, a vital part of tonal music is the shift of keys from one section to another. In music, modulation is most commonly the act or process of changing from one key (tonic, or tonal center) to another.

The key finding algorithm could easily be run on individual sections of a piece, once these sections were determined. It is possible to handle modulation: in considering a key for a segment, a penalty is assigned if the key differs from the key of the previous segment. In this way, it will prefer to remain in the same key, other things being equal, but will change keys if there is sufficient reason to do so. This task can be dealt with an algorithm similar to Viterbi algorithm, which can be implemented by dynamic programming as the following KeyModulation algorithm.

Algorithm KeyModulation



```

Given  $m$  music segments
for every segment  $i = 1, \dots, m$ 
    compute  $q[i, \cdot]$  vector of key weights by a key finding algorithm
Let  $d[1, \cdot] = q[1, \cdot]$ 
for  $i = 2$  to  $m$ 
    for  $j = 0$  to 23
         $d[i, j] = q[i, j] + \max_k (d[i - i, k] - w(k, j))$ 
         $pr[i, j] = \arg \max_k (d[i - i, k] - w(k, j))$ 
     $key[m] = \arg \max_j d[m, j]$ 

for  $i = m-1$  downto 1
     $key[i] = pr[key[i + 1]]$ 

```

In this algorithm, the vector position $pr[i, j]$ contains the best previous key which conducted to the j -th key estimation of the segment i . The function $w(k, j)$ gives the penalty for passing from k to j key. The penalty value is zero if there is no key chance: i.e. $w(j, j) = 0$.

With this strategy, the choice does not depends only on the segment in isolation, but it takes into account also previous evaluations. At each segment each key receives a local score indicating how compatible that key is with the pitches of the segment. Then we compute the best so far analysis ending at that key. The best scoring analysis of the last segment can be traced back to yield the preferred analysis of the entire piece. Notice that some choices can be changes as we proceed in the analysis of the segments. In this way the dynamic programming model gives a nice account of an important phenomenon in music perception: the fact that we sometimes revise our initial analysis of a segment based on what happens later.

9.4 Algorithms for music composition

Composers have long been fascinated by mathematical concepts in relation to music. The concept of "music of the spheres," dating back to Pythagoras, held the notion that humans were governed by the perfect proportions of the natural universe. This mathematical order may be seen in the musical interval choice and system of organization that was used by the ancient Greek culture.

Procedures that entail rules or provisions to govern the act of musical composition have been used since the Medieval period; these same principles have been applied in very specific methods to many of the recent computer programs developed for algorithmic composition.

9.4.1 Algorithmic Composition

Algorithmic composition could be described as "a sequence (set) of rules (instructions, operations) for solving (accomplishing) a [particular] problem (task) [in a finite number of steps] of combining musical parts (things, elements) into a whole (composition)". From this definition we can see that it is not necessary to use computers for algorithmic composition as we often infer; Mozart did not when he described the Musical Dice Game.

The concept of algorithmic composition is not something new. Pythagoras (around 500 B.C.) believed that music and mathematics were not separate studies. Hiller and Isaacson (1959) were probably the first who used a computational model using random number generators and Markov chains for algorithmic composition. Since then many researchers have tried to address the problem of algorithmic composition from different points of view.



Some of the algorithmic programs and compositions specify score information only. Score information includes pitch, duration, and dynamic material, whether written for acoustic and/or electronic instruments. That is, there are instances in which a composer makes use of a computer program to generate the score while the instrumental selection has been predetermined as either an electronic orchestra or a realization for acoustic instruments. Other algorithmic programs specify both score and electronic sound synthesis. In this instance, the program is used not only to generate the score, but also the electronic timbres to be used in performance. This distinction has its roots in the traditional differentiation between score and instrument, but a computer-generated continuum between two different sounds, however, is both score and sound synthesis. In both types of synthesis, the appearance of events in time is structured, both globally (form) as well as locally (sound, timbre).

The selection or construction of algorithms for musical applications can be divided into three categories:

- **Modeling traditional, non-algorithmic compositional procedures.** This category refers to algorithms that model traditional composition techniques (tonal harmony, tonal rhythm, counterpoint rules, specific formal devices, serial parametrisation, etc.). This approach has been scarcely used in music composition, but it has become an essential element of musicological Research.
- **Modeling new, original compositional procedures, different from those known before.** This category refers to algorithms that create new constructs which sport some inherently musical qualities. These algorithms range from Markov chains to stochastic and probabilistic functions. Such algorithms have been pioneered by the composer Iannis Xenakis in the "50s-'60s and widely used by a consistent group of composers since then.
- **Selecting algorithms from extra-musical disciplines.** This category refers to algorithms invented to model other, non-musical, processes and forms. Some of these algorithms have been used very proficiently by composers to create specific works. These algorithms are generally related to self-similarity (which is a characteristic that is closely related to that of "thematic development" which seems to belong universally to all musics) and they range from genetic algorithms to fractal systems, from cellular automata, to swarming models and coevolution. In this same category, a persistent trend of using biological data to generate compositional structures has developed since the 60's. Using brain activity (through EEG measurements), hormonal activity, human body dynamics, there has been a constant attempt to render biological data with musical structures.

9.4.2 Computer Assisted Composition

Another use of computers for music generation has been that of *Computer-Assisted Composition*. In this case, computers do not generate complete scores. Rather, they provide mediation tools to help composers to manage and control some aspects of musical creation. Such aspects may range from extremely relevant decision-making processes to minuscule details according to the composers' wishes.

Two main approaches can be observed in Computer-Assisted Composition:

- Integrated tools and languages that will cover all possible composing desiderata;
- Personalised micro-programs written in small languages like awk, lisp, perl, prolog, python, ruby, etc. (written by the composer herself and possibly interconnected together via pipes, sockets and common databases).

While computer assistance may be a more practical and less generative use of computers in Musical Composition, it is currently enjoying a much wider diffusion among composers.



9.4.3 Categories of algorithmic processes

A review can not be exhaustive because there have been so many attempts. In the following subsections¹ we give some representative examples of systems which employ different methods which we categorise, based on their most prominent feature, as follows:

9.4.3.1 Mathematical models

Stochastic processes and especially Markov chains have been used extensively in the past for algorithmic composition (e.g., Xenakis, 1971).

The basic algorithm is

Algorithm GenerateAndTest

```
while composition is not terminated
    generate raw materials
    modify according to various functions
    select the best results according to rules
```

The simplest way to generate music from a history-based model is to sample, at each stage, a random event from the distribution of events at that stage. After an event is sampled, it is added to the piece, and the process continues until a specified piece duration is exceeded.

Algorithm RandomWalk

```
Get events distribution by analysing a music repertoire
while composition is not terminated
    sample a random event from the distribution of events
    add to the piece
```

One manner of statistical analysis that has been frequently used in musical composition is Markov Analysis or Markov Chains. Named for the mathematician Andre Andreevich Markov (1856-1922), Markov Chains were conceived as a means by which decisions could be made based on probability. Information is linked together in a series of states based on the probability that state A will be followed by state B. The process is continually in transition because state A is then replaced by state B which continues to look at the probability of being followed by yet another state B. The so-called orders of Markov Analysis indicate the relationship between states. For instance, zeroth-order analysis assumes that there are no relationships between states; that is, the relationship between any two states is random. First-order analysis simply counts the frequency with which specific states occur within the given data. Second-order analysis examines the relationships between any two consecutive states (e.g., what is the probability that the state B would follow state A?). Third-order analysis determines the probability of three consecutive states occurring in a row (e.g., what is the probability that state A would be followed by state B, would be followed by state C?). Fourth-order analyzes the chance of four states following each other. Composer/scientist Lejaren Hiller made use of Markov Chains, statistical analysis, and stochastic procedures in algorithmic composition beginning in the late 1950s.

¹adapted from Papadopoulos, Wiggins 1993



Probably the most important reason for stochastic processes is their low complexity which makes them good candidates for real-time applications.

We also see computational models based on chaotic nonlinear systems or iterated functions but it is difficult to judge the quality of their output, because, unlike all the other approaches, their "knowledge" about music is not derived from humans or human works. Since the 1970s basic principles of the irregularities in nature have been studied by the scientific community, and by the 1980s chaos was the focus of a great deal of attention. The new science has spawned its own language, an elegant shop talk of fractals and bifurcations, intermittencies and periodicities, folded-towel diffeomorphisms and smooth noodle maps. . . To some physicists chaos is a science of process rather than state, or becoming rather than being. One subcategory of chaotic structures that has come to the forefront since ca. 1975 is fractals. Fractals are recursive and produce 'parent-child' relationships in which the offspring replicate the initial structure. Seen in visual art as smaller and smaller offshoots from the original stem, fractals were categorized by Benoit Mandelbrot in his book, *The Fractal Geometry of Nature*. The underlying principles of chaos may best be thought of in terms of natural, seemingly disorderly designs.

"Nature forms patterns. Some are orderly in space but disorderly in time, others orderly in time but disorderly in space. Some patterns are fractal, exhibiting structures self-similar in scale. Others give rise to steady states or oscillating ones. Pattern formation has become a branch of physics and of materials science, allowing scientists to model the aggregation of particles into clusters, the fractured spread of electrical discharges, and the growth of crystals in ice and metal alloys."

The main disadvantages of stochastic processes are:

- Someone needs to find the probabilities by analysing many pieces. Something necessary if we want to simulate one style. The resulting models will only generate music of similar style to the input.
- For higher order Markov models, transition tables become unmanageably large for the average computer. While many techniques exist for a more compact representation of sparse matrices (which usually result for higher order models), these require extra computational effort that can hinder real time performance.
- The deviations from the norm and how they are incorporated in the music is an important aspect. They also provide little support for structure at higher levels (unless multiple layered models are used where each event represents an entire Markov model in itself).

9.4.3.2 Knowledge based systems

Many early systems focused on taking existing musicological 'rules' and embedding them in computational procedures. In one sense, most AI systems are knowledge based systems (KBS). Here, we mean systems which are symbolic and use rules or constraints. The use of KBS in music seems to be a natural choice especially when we try to model well defined domains or we want to introduce explicit structures or rules. Their main advantage is that they have explicit reasoning; they can explain their choice of actions.

Even though KBS seem to be the most suitable choice, as a stand alone method, for algorithmic composition they still exhibit some important problems:

- Knowledge elicitation is difficult and time consuming, especially in subjective domains such as music.



- Since they do what we program them to do they depend on the ability of the "expert", who in many cases is not the same as the programmer, to clarify concepts, or even find a flexible representation.
- They become too complicated if we try to add all the "exceptions to the rule" and their preconditions, something necessary in this domain.

9.4.3.3 Grammars

The idea that there is a grammar of music is probably as old as the idea of grammar itself.

Linguistics is an attempt to identify how language functions: what are the components, how do the components function as a single unit, and how do the components function as single entities within the context of the larger unit. Linguistic theory models this unconscious knowledge [of speech] by a formal system of principles or rules called a generative grammar, which describes (or 'generates') the possible sentences of the language. Curtis Roads has made a distinction between the specific use of generative grammars and the more open-ended field of algorithmic composition in that "Generative modeling of music can be distinguished from algorithmic composition on the basis of different goals. While algorithmic composition aims at an aesthetically satisfying new composition, generative modeling of music is a means of proposing and verifying a theory of an extant corpus of compositions or the competence that generated them."

Experiments in Musical Intelligence (EMI) is a project focused on the understanding of musical style and stylistic replication of various composers (Cope, 1991, 1992). EMI needs as an input a minimum of two works from which extracts "signatures" using pattern matching. The meaningful arrangement of these signatures in replicated works is accomplished through the use of an augmented transition network (ATN).

Some basic problems of the grammars are:

- They are hierarchical structures while much music is not (i.e. improvisation). Therefore ambiguity might be necessary since it "can add to the representational power of a grammar".
- Most, if not all, musical grammar implementations do not make any strong claims about the semantics of the pieces.
- Usually a grammar can generate a large number musical strings of questionable quality.
- Parsing is, in many cases, computationally expensive especially if we try to cope with ambiguity.

9.4.3.4 Evolutionary methods

Genetic algorithms (GAs) have proven to be very efficient search methods, especially when dealing with problems with very large search spaces. This, coupled with their ability to provide multiple solutions, which is often what is needed in creative domains, makes them a good candidate for a search engine in a musical application. Taking inspiration from natural evolution to guide search of problem space, the idea is that 'good' compositions, or composition systems can be evolved from an initial (often random) starting point.

Algorithm GeneticAlg

Initialise population

while not finished evolving

 Calculate fitness of each individual



Select preferred individuals to be parents

for N := populationSize

Breed new individuals

(cross over + mutation)

Build next generation

Render output

We can divide these attempts into two categories based on the implementation of the fitness function.

Use of an objective fitness function. In this case the chromosomes are evaluated based on formally stated, computable functions. The efficacy of the GA approach depends heavily on the amount of knowledge the system possesses; even so GAs are not ideal for the simulation of human musical thought because their operation in no way simulates human behaviour.

Use of a human as a fitness function. Usually we refer to this type of GA as interactive-GA (IGA). In this case a human replaces the fitness function in order to evaluate the chromosomes.

These attempts exhibit two main drawbacks associated with all IGAs:

- Subjectivity
- Efficiency, the "fitness bottleneck". The user must hear all the potential solutions in order to evaluate a population.

Moreover, this approach tells us little about the mental processes involved in music composition since all the reasoning is encoded inaccessibly in the user's mind. Most of these approaches exhibit very simple representations in an attempt to decrease the search space, which in some cases compromises their output quality.

9.4.3.5 Systems which learn

In the category of learning systems are systems which, in general, do not have a priori knowledge (e.g. production rules, constraints) of the domain, but instead learn its features by examples. We can further classify these systems, based on the way they store the information, to subsymbolic/distributive (Artificial Neural Networks, ANN) and symbolic (Machine Learning, ML).

ANNs offer an alternative for algorithmic composition to the traditional symbolic AI methods, one which loosely resembles the activities in the human brain, but at the moment they do not seem to be as efficient or as practical, at least as a stand-alone approach. Some of their disadvantages are:

- Composition as compared with cognition is a much more highly intellectual process (more "symbolic"). The output from a ANN matches the probability distribution of the sequence set to which it is exposed, something which is desirable, but on the other hand shows us its limit: "While ANNs are capable of successfully capturing the surface structure of a melodic passage and produce new melodies on the basis of the thus acquired knowledge, they mostly fail to pick up the higher-level features of music, such as those related to phrasing or tonal functions".
- The representation of time can not be dealt efficiently even with ANNs which have feedback. Usually they solve toy problems, with many simplifications, when compared with the knowledge based approaches.



- They can not even reproduce fully the training set and when they do this it might mean that they did not generalise.
- Even though it seems exciting that a system learns by examples this is not always the whole truth since the human in many cases needs to do the "filtering" in order not to have in the training set examples which conflict.
- Usually, the researchers using ANNs say that their advantage over knowledge based approaches is that they can learn from examples things which can't be represented symbolically using rules (i.e. the "exceptions").

9.4.3.6 Hybrid systems

Hybrid systems are ones which use a combination of AI techniques. In this section we discuss systems which combine evolutionary and connectionist methods, or symbolic and subsymbolic ones. The reason behind using hybrid systems, not only for musical applications, is very simple and logical. Since each AI method exhibits different strengths then we should adopt a "postmodern" attitude by combining them.

The main disadvantage of hybrid systems is that they are usually complicated, especially in the case of tightly-coupled or fully integrated models. The implementation, verification and validation is also time consuming.

9.4.4 Discussion

First there is usually no evaluation of the output by real experts (e.g., professional musicians) in most of the systems and second, the evaluation of the system (algorithm) is given relatively small consideration

Knowledge representation Two almost ubiquitous issues in AI are representation of knowledge and search method. From one point of view, our categorisation above, reflects the search method, which however, constrains the possible representations of knowledge. For example structures which are easily represented symbolically are often difficult to represent with a ANN.

In many AI systems, especially symbolic, the choice of the knowledge representation is an important factor in reducing the search space. For example Biles (1994) and Papadopoulos and Wiggins (1998) used a more abstract representation, representing the degrees of the scale rather than the absolute pitches. This reduced immensely the search space since the representation did not allow the generation of non-scale notes (they are considered dissonant) and the inter-key equivalence was abstracted out.

Most of the systems reviewed exhibit a single, fixed representation of the musical structures. Some, on the other hand, use multiple viewpoints which we believe simulate more closely human musical thinking.

Computational Creativity Probably the most difficult task is to incorporate in our systems the concept of creativity. This is difficult since we do not have a clear idea of what creativity is (Boden, 1996).

Some characteristics of computational creativity, which were proposed by Rowe and Partridge (1993) are:

- Knowledge representation is organised in such a way that the number of possible associations is maximised. A flexible knowledge representation scheme. Similarly Boden (1996) says that representation should allow to explore and transform the conceptual space.
- Tolerate ambiguity in representations.



- Allow multiple representations in order to avoid the problem of "functional fixity".
- The usefulness of new combinations should be assessable.

New combinations need to be elaboratable to find out their consequences. One question that AI researchers should aim to answer is: do we want to simulate human creativity itself or the results of it? (Is DEEP BLUE creative, even if it does not simulate the human mind?) This is more or less similar to the, subtle in many cases, distinction between cognitive modeling and knowledge engineering.

9.4.5 Emerging Trends

There are trends that, while being foreign to the Music Generation Modelling domain, propose issues that need to be taken in due consideration because they may condition the musical creation in the very near future².

Internet as a Participation Architecture The Internet is developing as an architecture of participation. There is a fast development of support to the creation of musical subcultures. In fact new musical styles develop with a speed never seen before. The spreading of new works of art happens through peer recommendation. In this way the Internet contributes to social innovation and the creation of social interaction and integration without much of geographic boundaries. Even language boundaries are less important in the musical domain, which stimulates the emergence of World Musics. In this way music and Internet have functions that create mutual synergy. In this way music can become an antidote for individualism. Technology could help in bringing people together through musical communication and interaction. However, many of these new systems depend on information gathering technologies that cannot stand the test of acceptable user privacy and on the other hand social participation and effects of entrainment are not well understood. What kind of participative technologies are needed in this domain?

Music as a Multi-Modal phenomenon Up to recent, most music technology researchers have associated music research with audio. Yet the above trend shows that music is in fact grounded on multi-modal perception and action. The way music is experienced in non-Western cultures and in the modern Western popular culture is a good example of the multi-modal basis of music, e.g. its association with dance, costumes, decor etc... The multi-modal aspect of musical interaction draws on the idea that the sensory systems, auditory, visual, haptic, and tactile, as well as movement perception, form a fully integrated part of the way the human subject is involved with music during interactive musical communication. However, the multi-modal basis of the musical experience is very badly understood, as is the coupling between perception and action. A more thorough scientific understanding of the multi-modal basis of music, as well as of the close interaction between perception and action, is needed in view of the new trends towards multimedia.

Active Listener Participation Looking at the consumption pattern of people, there is also a trend which shows that people become more active consumers. For example, children nowadays like more their computer environment than television because they can be active with it. We don't consume what is presented to us, but we perform actions and we choose. (Digital television is likely to focus on this new type of consumers in the near future in that it will offer programs with active participation.) In music creation and performance, active participation of the audience is likely to become a new trend provided that there is a technology which processes the actions of the consumer and feed them back into the performance. More research is needed in exploring technology as an extension of the human body,

²adapted from Sound and Music Computing roadmap, S2S² project (in preparation)



capture responses of the human body, as individual and as a group, and allow active participation of the participant. This involves massive wireless networking of many people gathered in indoor or outdoor theatres, which goes much beyond any present day mobile technology capacity density, and high quality portable music equipment.

9.5 Markov Models and Hidden Markov Models

Andrei Andreevich Markov first introduced his mathematical model of dependence, now known as Markov chains, in 1907. A Markov model (or Markov chain) is a mathematical model used to represent the tendency of one event to follow another event, or even to follow an entire sequence of events. Markov chains are matrices comprised of probabilities that reflect the dependence of one or more events on previous events. Markov first applied his modeling technique to determine tendencies found in Russian spelling. Since then, Markov chains have been used as a modeling technique for a wide variety of applications ranging from weather systems to baseball games.

Statistical methods of Markov source or hidden Markov modeling (HMM) have become increasingly popular in the last several years. The models are very rich in mathematical structure and hence can form a basis for use in a wide range of applications. Moreover the models, when applied properly, work very well in practice for several important applications.

9.5.1 Markov Models or Markov chains

Markov models are very useful to represent families of sequences with certain specific statistical properties. To explain the idea consider a simple 3 state model of the weather. We assume that once a day, the weather is observed as being one of the following: rain (state 1); cloudy (state 2); sunny (state 3).

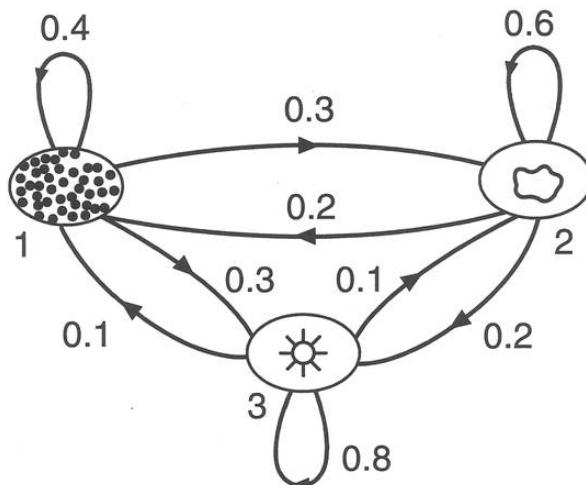


Figure 9.32: State transition of the weather Markov model (from Rabiner 1999).

If we examine a sequence of observation during a month, the state rain appears a few times, and it can be followed by rain, cloud or sun. Given a long sequence of observations, we can count the number of times the state rain is followed by, say, a cloudy state. From this we can estimate the probability that a rain is followed by a cloudy state. If this probability is 0.3 for example, we indicate it as shown in Figure 7.32. The figure also shows examples of probabilities for every state to transition to other states,

including itself. The first row of the matrix A

$$A = \{a_{i,j}\} = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix} \quad (9.9)$$

shows the three probabilities more compactly (notice that their sum is unity). Similarly the probabilities that the cloudy state would transition into the three states can be estimated, and is shown in the second row of the matrix. This 3×3 matrix is called a state transition matrix, and is denoted as \mathbf{A} and the coefficients have the properties $a_{i,j} \geq 0$ and $\sum_j a_{i,j} = 1$ since they obey standard stochastic constraints. Figure 7.32 is called a Markov model.

Formally a Markov model (MM) models a process that goes through a sequence of discrete states, such as notes in a melody. At regular spaced, discrete times, the system undergoes a change of state (possibly back to same state) according to a set of probabilities associated with the state. The time instances for a state change is denoted t and the actual state at time t as $x(t)$. The model is a weighted automaton that consists of:

- A set of N states, $S = \{s_1, s_2, s_3, \dots, s_N\}$.
- A set of transition probabilities, \mathbf{A} , where each $a_{i,j}$ in \mathbf{A} represents the probability of a transition from s_i to s_j . I.e $a_{i,j} = P[x(t) = j | x(t-1) = i]$.
- A probability distribution, π , where π_i is the probability the automaton will begin in state s_i , i.e $\pi_i = P(x_1 = i)$, $1 \leq i \leq N$. Notice that the stochastic property for the initial state distribution vector is $\sum_i \pi_i = 1$.
- E , a subset of S containing the legal ending states.

In this model, the probability of transitioning from a given state to another state is assumed to depend only on the current state. This is known as the Markov property.

Given a sequence or a set of sequences of "similar kind" (e.g., a long list of melodies from a composer) the parameters of the model (the transition probabilities) can readily be estimated. The process of identifying the model parameters is called training the model. In all discussions it is implicitly assumed that the probabilities of transitions are fixed and do not depend on past transitions.

Suppose we are given a Markov model (i.e., \mathbf{A} given). Given an arbitrary state sequence $\mathbf{x} = [x(1), x(2), \dots, x(L)]$ we can calculate the probability that \mathbf{x} has been generated by our model. This is given by the product

$$P(\mathbf{x}) = P(x(1)) \times P(x(1) \rightarrow x(2)) \times P(x(2) \rightarrow x(3)) \times \cdots \times P(x(L-1) \rightarrow x(L))$$

where $P(x(1)) = \pi(x(1))$ is the probability that $x(1)$ is the initial state, $P(x(k) \rightarrow x(m))$ is the transition probability for going from $x(k)$ to $x(m)$, and can be found from the matrix \mathbf{A} . For example with reference to the weather Markov model of equation 7.9, given that the weather on day 1 is sunny (state 3), we can ask the question: What is the probability that the weather for next 7 days will be "sun-sun-rain-rain-sun-cloudy-sun . . ."? This probability can be evaluated as

$$\begin{aligned} P &= \pi_3 \cdot a_{33} \cdot a_{33} \cdot a_{31} \cdot a_{11} \cdot a_{13} \cdot a_{32} \cdot a_{13} \\ &= 1(0.8)(0.8)(0.1)(0.4)(0.3)(0.1)(0.2) \\ &= 1.536 \times 10^{-4} \end{aligned}$$

The usefulness of such computation is as follows: given a number of Markov models (\mathbf{A}_1 for a composer, \mathbf{A}_2 for a second composer, and so forth) and given a melody \mathbf{x} , we can calculate the probabilities that this melody is generated by any of these models. The model which gives the highest probability is most likely the model which generated the sequence.



9.5.2 Hidden Markov Models

A hidden Markov model (HMM) is obtained by a slight modification of the Markov model. Thus consider the state diagram shown in Figure 7.32 which shows three states numbered 1, 2, and 3. The probabilities of transitions from the states are also indicated, resulting in the state transition matrix \mathbf{A} shown in equation 7.9. Now we can suppose that we can not observe directly the state, but only a symbol that is associated in a probabilistic way to the state. For example when the weather system is in a particular state, it can output one of four possible symbols L, M, H, VH (corresponding to temperature classes low, medium, high, very high), and there is a probability associated with each of these. This is summarized in the so-called output matrix \mathbf{B}

$$\mathbf{B} = \{b_{i,j}\} = \begin{bmatrix} 0.4 & 0.3 & 0.2 & 0.1 \\ 0.2 & 0.5 & 0.2 & 0.1 \\ 0.1 & 0.1 & 0.4 & 0.4 \end{bmatrix} \quad (9.10)$$

The element $b_{i,j}$ represents the probability of observing the temperature class j when the weather is in the (non observable) state i , i.e. $b_{i,j} = P(x(t) = s_i | x(t) = j)$. For example when the weather is rainy (state $i = 1$), the probability of measuring medium temperature (output symbol $j = 2$) is $b_{1,2} = 0.3$.

More formally, an HMM requires two things in addition to that required for a standard Markov model:

- A set of possible observations, $O = \{o_1, o_2, o_3, \dots, o_n\}$.
- A probability distribution \mathbf{B} over the set of observations for each state in S .

Basic HMM problems In order to apply the hidden Markov model theory successfully there are three problems that need to be solved in practice. These are listed below along with names of standard algorithms which have been developed for these.

1. *Learn structure problem.* Given an HMM (i.e., given the matrices \mathbf{A} and \mathbf{B}) and an output sequence $o(1), o(2), \dots$, compute the state sequence $x(k)$ which most likely generated it. This is solved by the famous Viterbi's algorithm (see 7.5.4.2).
2. *Evaluation or scoring problem.* Given the HMM and an output sequence $o(1), o(2), \dots$ compute the probability that the HMM generates this. We can also view the problem as one of scoring how well a given model matches a given output sequence. If we are trying to choose among several competing models, this ranking allow us to choose the model that best matches the observations. The forward-backward algorithm solves this (see 7.5.4.1).
3. *Training problem.* How should one design the model parameters \mathbf{A} and \mathbf{B} such that they are optimal for an application, e.g., to represent a melody? The most popular algorithm for this is the expectation maximization algorithm commonly known as the EM algorithm or the Baum-Welch algorithm (see Rabiner [1989] for more details).

For example let us consider a simple isolated word recognizer (see Figure 7.33). For each word we want to design a separate N -state HMM. We represent the speech signal as a time sequence of coded spectral vectors. Hence each observation is the index of the spectral vector closest to the original speech signal. Thus for each word, we have a training sequence consisting of repetitions of codebook indices of the word.

The first task is to build individual word models. This task is done by using the solution to Problem 3 to estimate model parameters for each word model. To develop an understanding of physical meaning of



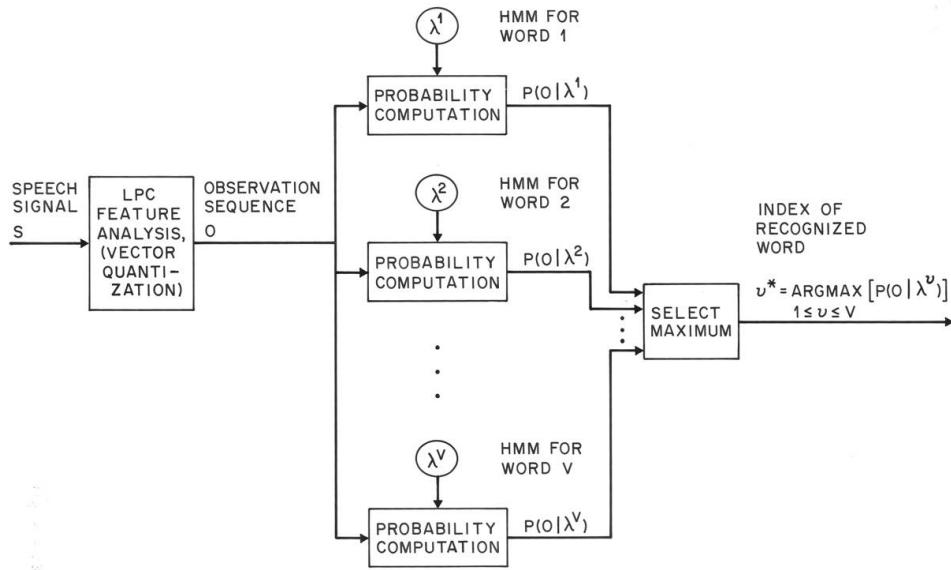


Figure 9.33: Block diagram of an isolated word recognizer (from Rabiner 1999).

the model state, we use the solution to Problem 1 to segment each of the word state sequence into states, and then study the properties of the spectral vectors that lead to the observations occurring in each state. Finally, once the set of HMMs has been designed, recognition of an unknown word is performed using the solution to Problem 2 to score each word model based on the observation sequence, and select the word whose model score is highest.

We should remark that the HMM is a stochastic approach which models the given problem as a “doubly stochastic process” in which the observed data are thought to be the result of having passed the “true” (hidden) process through a second process. Both processes are to be characterized using only the one that could be observed. The problem with this approach, is that one do not know anything about the Markov chains that generate the speech. The number of states in the model is unknown, there probabilistic functions are unknown and one can not tell from which state an observation was produced. These properties are hidden, and thereby the name hidden Markov model.

9.5.3 Markov Models Applied to Music

Hiller and Isaacson (1957) were the first to implement Markov chains in a musical application. They developed a computer program that used Markov chains to compose a string quartet comprised of four movements entitled the Illiac Suite. Around the same time period, Meyer and Xenakis (1971) realized that Markov chains could reasonably represent musical events. In his book Formalized Music Xenakis [1971], Xenakis described musical events in terms of three components: frequency, duration, and intensity. These three components were combined in the form of a vector and then were used as the states in Markov chains. In congruence with Xenakis, Jones (1981) suggested the use of vectors to describe notes (e.g., note = pitch, duration, amplitude, instrument) for the purposes of eliciting more complex musical behavior from a Markov chain. In addition, Polansky, Rosenboom, and Burk (1987) proposed the use of hierarchical Markov chains to generate different levels of musical organization (e.g., a high level chain to define the key or tempo, an intermediate level chain to select a phrase of notes, and a low level chain to determine the specific pitches). All of the aforementioned research deals with the compositional aspects and uses of Markov chains. That is, all of this research was focused on creating musical output using Markov chains.



9.5.3.1 HMM models for music search: MuseArt

In the MuseArt system for music search and retrieval, developed at Michigan University by Jonah Shifrin, Bryan Pardo, Colin Meek, William Birmingham, musical themes are represented using a hidden Markov model (HMM).

Representation of a query. The query is treated as an observation sequence and a theme is judged similar to the query if the associated HMM has a high likelihood of generating the query. A piece of music is deemed a good match if at least one theme from that piece is similar to the query. The pieces are returned to the user in order, ranked by similarity.

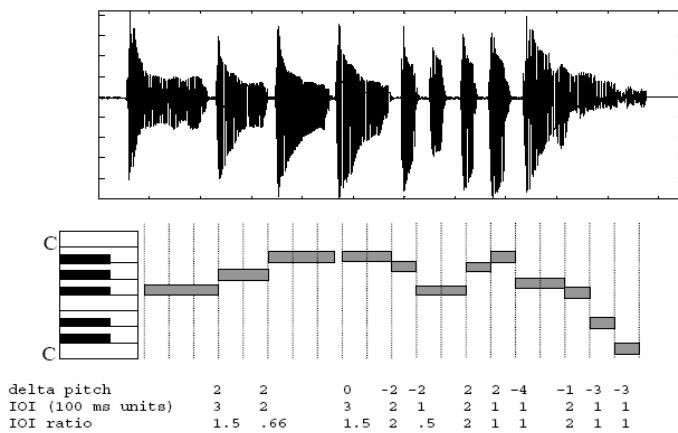


Figure 9.34: A sung query (from Shifrin 2002)

A query is a melodic fragment sung by a single individual. The singer is asked to select one syllable, such as "ta" or "la," and use it consistently during the query. The consistent use of a single consonant-vowel pairing lessens pitch-tracker error by providing a clear onset point for each note, as well as reducing error caused by vocalic variation. A query is recorded as a .wav file and is transcribed into a MIDI based representation using a pitch-tracking system. Figure 7.34 shows a time-amplitude representation of a sung query, along with example pitch-tracker output (shown as piano roll) and a sequence of values derived from the MIDI representation (the *deltaPitch*, *IOI* and *IOIratio* values). Time values in the figure are rounded to the nearest 100 milliseconds. We define the following.

- A note transition between note n and note $n + 1$ is described by the tuple $\langle \text{deltaPitch}_n, \text{IOIratio}_n \rangle$.
- deltaPitch_n is the musical interval, i.e. the pitch difference in semitones between note n and note $n + 1$.
- IOIratio_n is $\text{IOI}_n / \text{IOI}_{n+1}$, where the inter onset interval (IOI_n) is the difference between the onset of notes n and $n + 1$. For the final transition, $\text{IOI}_n = \text{IOI}_n / \text{duration}_{n+1}$.

A query is represented as a sequence of note transitions. Note transitions are useful because they are robust in the face of transposition and tempo changes. The *deltaPitch* component of a note transition captures pitch-change information. Two versions of a piece played in two different keys have the same *deltaPitch* values. The *IOIratio* represents the rhythmic component of a piece. This remains constant even when two performances are played at very different speeds, as long as relative durations within



each performance remain the same. In order to reduce The number of possible IOI ratios is reduced by quantizing them to one of 27 values, spaced evenly on a logarithmic scale. A logarithmic scale was selected because data from a pilot study indicated that sung *IOIratio* values fall naturally into evenly spaced bins in the log domain.

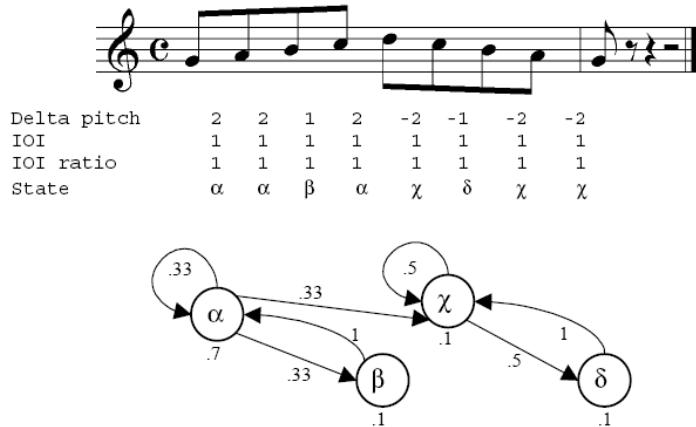


Figure 9.35: Markov model for a scalar passage (from Shifrin 2002)

The directed graph in Figure 7.35 represents a Markov model of a scalar passage of music. States are note transitions. Nodes represent states. The numerical value below each state indicates the probability a traversal of the graph will begin in that state. As a default, all states are assumed to be legal ending states. Directed edges represent transitions. Numerical values by edges indicate transition probabilities. Only transitions with non-zero probabilities are shown.

In Markov model, it is implicitly assumed that whenever state s is reached, it is directly observable, with no chance for error. This is often not a realistic assumption. There are multiple possible sources of error in generating a query. The singer may have incorrect recall of the melody he or she is attempting to sing. There may be production errors (e.g., cracked notes, poor pitch control). The transcription system may introduce pitch errors, such as octave displacement, or timing errors due to the quantization of time. Such errors can be handled gracefully if a probability distribution over the set of possible observations (such as note transitions in a query) given a state (the intended note transition of the singer) is maintained. Thus, to take into account these various types of errors, the Markov model should be extended to a hidden Markov Model, or HMM. The HMM allows us a probabilistic map of observed states to states internal to the model (hidden states). In the system, a query is a sequence of observations. Each observation is a note-transition duple, $\langle \text{deltaPitch}, \text{IOIratio} \rangle$. Musical themes are represented as hidden Markov models whose states also corresponds to note-transition duples. To make use of the strengths of a hidden Markov model, it is important to model the probability of each observation o_i in the set of possible observations, O , given a hidden state, s .

Making Markov Models from MIDI. Our system represents musical themes in a database as HMMs. Each HMM is built automatically from a MIDI file encoding the theme. The unique duples characterizing the note transitions found in the MIDI file form the states in the model. Figure 7.35 shows a passage with eight note transitions characterized by four unique duples. Each unique duple is represented as a state. Once the states are determined for the model, transition probabilities between states are computed by calculating what proportion of the time state a follows state b in the theme. Often, this results in a large number of deterministic transitions. Figure 7.36 is an example of this, where only a single state has two possible transitions, one back to itself and the other on to the next state. Note that



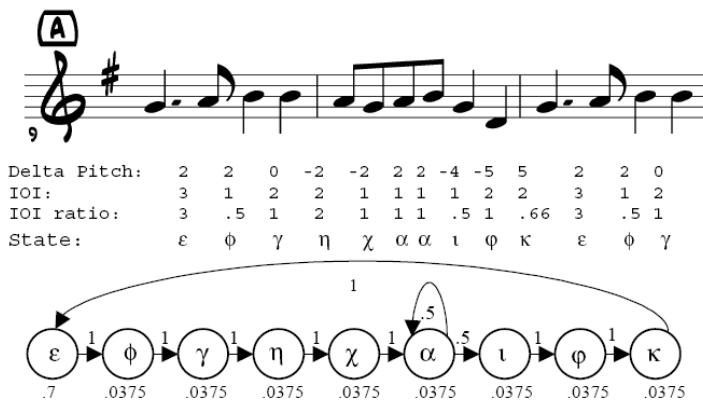


Figure 9.36: Markov model for Alouette fragment (from Shifrin 2002)

there is not a one-to-one correspondence between model and observation sequence. A single model may create a variety of observation sequences, and an observation sequence may be generated by more than one model. Recall that our approach defines an observation as a tuple, $\langle \text{deltaPitch}, \text{IOIratio} \rangle$. Given this, the observation sequence $q = \{(2, 1), (2, 1), (2, 1)\}$ may be generated by the HMM in Figure 7.35 or the HMM in Figure 7.36.

Finding the best target. The themes in the database are coded as HMMs and the query is treated as an observation sequence. Given this, we are interested in finding the HMM most likely to generate the observation sequence. This can be done using the Forward algorithm. The Forward algorithm, given an HMM and an observation sequence, returns a value between 0 and 1, indicating the probability the HMM generated the observation sequence. Given a maximum path length, L , the algorithm takes all paths through the model of up to L steps. The probability each path has of generating the observation sequence is calculated and the sum of these probabilities gives the probability that the model generated the observation sequence. This algorithm takes on the order of $|S|^2 L$ steps to compute the probability, where $|S|$ is the number of states in the model.

Let there be an observation sequence (query), O , and a set of models (themes), M . An order may be imposed on M by performing the Forward algorithm on each model m in M and then ordering the set by the value returned, placing higher values before lower. The i -th model in the ordered set is then the i -th most likely to have generated the observation sequence. We take this rank order to be a direct measure of the relative similarity between a theme and a query. Thus, the first theme is the one most similar to the query.

9.5.3.2 Markov sequence generator

Markov models can be thought of as generative models. A generative model describes an underlying structure able to generate the sequence of observed events, called an observation sequence. Note that there is not a one-to-one correspondence between model and observation sequence. A single model may create a variety of observation sequences, and an observation sequence may be generated by more than one model.

A HMM can be used as generator to give an observation sequence O as follow.

Algorithm `MarkovSequenceGenerator`

1. Choose initial state $x(1) = S_1$ according the initial state distribution π .



2. Set $t = 1$
3. Choose $o(t)$ according the symbol probability distribution in state $x(t)$ described in matrix \mathbf{B}
4. Transit to new state $x(t + 1) = S_j$ according to the state transition probability for state $x(t) = i$, i.e. $a_{i,j}$
5. Set $t = t + 1$ and return to step 2

If a simple Markov model is used as generator, step 3 is skipped, and the state $x(t)$ is used in output.

The "hymn tunes" of Figure 7.37 were generated by computer from an analysis of the probabilities of notes occurring in various hymns. A set of hymn melodies were encoded (all in C major). Only hymn melodies in 4/4 meter and containing two four-bar phrases were used. The first "tune" was generated by simply randomly selecting notes from each of the corresponding points in the analyzed melodies. Since the most common note at the end of each phrase was 'C' there is a strong likelihood that the randomly selected pitch ending each phrase is C.



Figure 9.37: "Hymn tunes" generated by computer from an analysis of the probabilities of notes occurring in various hymns. From Brooks, Hopkins, Neumann, Wright. "An experiment in musical composition." IRE Transactions on Electronic Computers, Vol. 6, No. 1 (1957).

9.5.4 Algorithms

9.5.4.1 Forward algorithm

The Forward algorithm is used to solve the evaluation or scoring problem. Given the HMM $\lambda = (A, B, \Pi)$ and an observation sequence $O = o(1)o(2)\dots o(L)$ compute the probability $P(O|\lambda)$ that the HMM generates this. We can also view the problem as one of scoring how well a given model matches a given output sequence. If we are trying to choose among several competing models, this ranking allows us



to choose the model that best matches the observations. The most straightforward procedure is through enumerating every possible state sequence of length L (the number of observations), computing the joint probability of the state sequence and O and finally summing the joint probability over all possible state sequence. But if there are N possible states that can be reached, there are N^L possible state sequences and thus such direct approach have exponential computational complexity.

However we can notice that there are only N states and we can apply a dynamic programming strategy. To this purpose let us define the forward variable $\alpha_t(i)$ as

$$\alpha_t(i) = P(o(1)o(2)\dots o(t), x(t) = s_i | \lambda)$$

i.e. the probability of the partial observation $o(1)o(2)\dots o(t)$ and state s_i at time t , given the model λ . The Forward algorithm solves the problem with a dynamic programming strategy, using an iteration on the sequence length (time t), as follows:

Algorithm Forward

1. Initialization

$$\alpha_1(i) = \pi(i)b_i(o_1), 1 \leq i \leq N$$

2. Induction

$$\alpha_{t+1}(i) = \left[\sum_{j=1}^N \alpha_t(j)a_{ij} \right] b_j(o_{t+1}) \quad 1 \leq t \leq L-1 \\ 1 \leq i \leq N$$

3. Termination

$$P(O|\lambda) = \sum_{i=1}^N \alpha_L(i)$$

Step 1) initializes the forward probabilities as the joint probability of state i and initial observation $o(1)$. The induction step is illustrated in Figure 7.38(a). This figure shows that state s_j can be reached at time $t+1$ from the N possible states at time t . Since $\alpha_t(i)$ is the probability that $o(1)o(2)\dots o(t)$ is observed and $x(t) = s_i$, the product $\alpha_t(i)a_{ij}$ is the probability that $o(1)o(2)\dots o(t)$ is observed and state s_j is reached at time $t+1$ via state s_i at time t . Summing this product over all the possible states results in the probability of s_j with all the previous observations. Finally $\alpha_{t+1}(i)$ is obtained by accounting for observation o_{t+1} in state s_j , i.e. by multiplying by the probability $b_j(o_{t+1})$. Finally step 3) gives the desired $P(O|\lambda)$ as the sum of the terminal forward variables $\alpha_L(i)$. In fact $\alpha_L(i)$ is the probability of the observed sequence and that the system at time $t=L$ is in the state s_i . Hence $P(O|\lambda)$ is just the sum of the $\alpha_L(i)$'s. The time computational complexity of this algorithm is $O(N^2L)$. The forward probability calculation is based upon the lattice structure shown in figure 7.38(b). The key is that since there are only N states, all the possible state sequences will remerge into these N nodes, no matter how long the observation sequence. Notice that the calculation of $\alpha_t(i)$ involves multiplication with probabilities. All these probabilities have a value less than 1 (generally significantly less than 1), and as t starts to grow large, each term of $\alpha_t(i)$ starts to head exponentially to zero, exceed the precision range of the machine. To avoid this problem, a version of the Forward algorithm with scaling should be used. See Rabiner [1989] for more details.



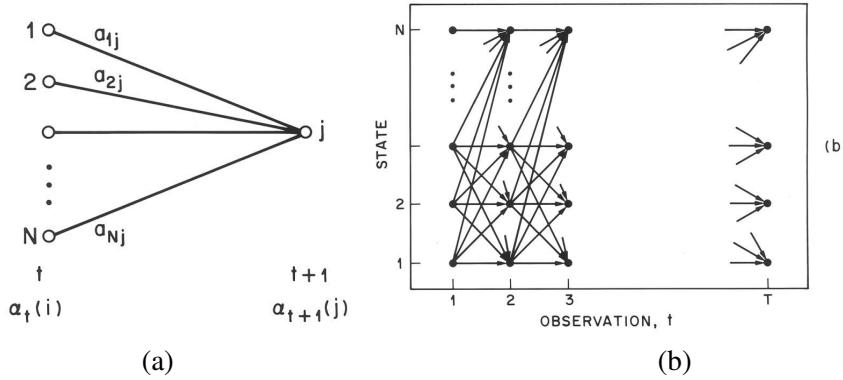


Figure 9.38: (a) Illustration of the sequence of operations required for the computation of the forward variable $\alpha_{t+1}(i)$. (b) Implementation of the computation of $\alpha_{t+1}(i)$ in terms of a lattice of observation t and states i .

9.5.4.2 Viterbi algorithm

The Viterbi algorithm, based on dynamic programming, is used to solve the structure learning problem. Given an HMM λ (i.e., given the matrices \mathbf{A} and \mathbf{B}) and an output sequence $O = \{o(1)o(2)\dots o(L)\}$, find the *single* best state sequence $X = \{x(1)x(2)\dots x(L)\}$ which most likely generated it. To this purpose we define the quantity

$$\delta_t(i) = P[x(1)x(2)\dots x(t) = s_i, o(1)o(2)\dots o(t) | \lambda]$$

i.e. $\delta_t(i)$ is the best score (highest probability) along a single path at time t , which accounts for the first t observations and ends in state s_i . By induction we have

$$\delta_{t+1}(i) = \max_i [\delta_t(i)a_{ij}] b_j(o_{t+1})$$

To actually retrieve the state sequence, we need to keep track of the argument which maximized the previous expression, for each t and j using a predecessor array $\psi_t(j)$. The complete procedure of Viterbi algorithm is

Algorithm Viterbi

1. Initialization
for $1 \leq i \leq N$
 $\delta_1(i) = \pi(i)b_i(o_1)$
 $\psi_1(i) = 0$
2. Induction
for $1 \leq t \leq L - 1$
for $1 \leq j \leq N$
 $\delta_{t+1}(j) = \max_i [\delta_t(i)a_{ij}] b_j(o_{t+1})$
 $\psi_{t+1}(j) = \text{argmax}_i [\delta_t(i)a_{ij}]$
3. Termination
 $P^* = \max_i [\delta_T(i)]$
 $x^*(T) = \text{argmax}_i [\delta_T(i)]$



4. Path backtracking

for $t = L - 1$ **downto** 1

$$x^*(t) = \psi_{t+1}(x^*_{t+1})$$

Notice that the structure of Viterbi algorithm is similar in implementation to forward algorithm. The major difference is the maximization over the previous states which is used in place of the summing procedure in forward algorithm. Both algorithms used the lattice computational structure of figure 7.38(b) and have computational complexity N^2L . Also Viterbi algorithm presents the problem of multiplication of probabilities. One way to avoid this is to take the logarithm of the model parameters, giving that the multiplications become additions. The induction thus becomes

$$\log[\delta_{t+1}(i)] = \max_i (\log [\delta_t(i)] + \log [a_{ij}] + \log [b_j(o_{t+1})])$$

Obviously will this logarithm become a problem when some model parameters has zeros present. This is often the case for A and π and can be avoided by adding a small number to the matrixes. See Rabiner [1989] for more details.

To get a better insight of how the Viterbi (and the alternative Viterbi) works, consider a model with $N = 3$ states and an observation of length $L = 8$. In the initialization ($t = 1$) is $\delta_1(1)$, $\delta_1(2)$ and $\delta_1(3)$ found. Lets assume that $\delta_1(2)$ is the maximum. Next time ($t = 2$) three variables will be used namely $\delta_2(1)$, $\delta_2(2)$ and $\delta_2(3)$. Lets assume that $\delta_2(1)$ is now the maximum. In the same manner will the following variables $\delta_3(3)$, $\delta_4(2)$, $\delta_5(2)$, $\delta_6(1)$, $\delta_7(3)$ and $\delta_8(3)$ be the maximum at their time, see Fig.7.39. This algorithm is an example of what is called the Breadth First Search (Viterbi employs this essentially). In fact it follows the principle: "Do not go to the next time instant $t + 1$ until the nodes at at time T are all expanded".

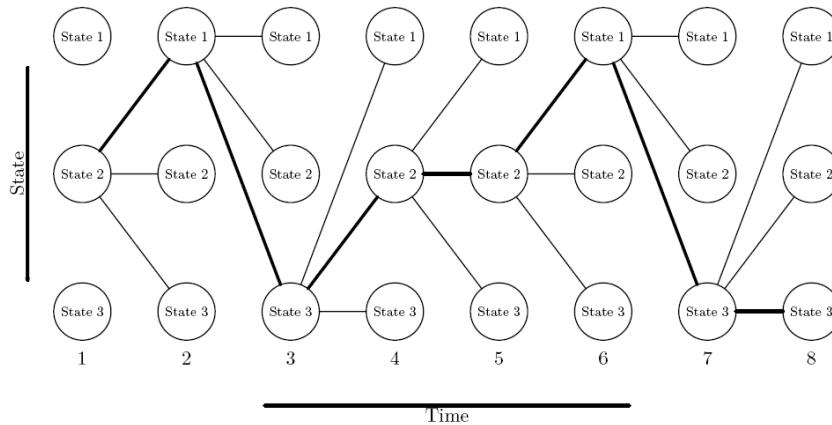


Figure 9.39: Example of Viterbi search.

9.6 Appendix

9.6.1 Generative Theory of Tonal Music of Lerdahl and Jackendorff

Lerdahl and Jackendoff (1983) developed a model called Generative Theory of Tonal Music (GTTM). This model offers a complementary approach to understanding melodies, based on a hierarchical structure of musical cognition. According to this theory music is built from an inventory of notes and a set of rules. The rules assemble notes into a sequence and organize them into hierarchical structures of music



cognition. To understand a piece of music means to assemble these mental structures as we listen to the piece.

It seeks to elucidate a number of perceptual characteristics of tonal music - segmentation, periodicity, differential degrees of importance being accorded to the components of a musical passage or work, the flow of tension and relaxation as a work unfolds - by employing four distinct analytical levels, each with its own more-or-less formal analytical principles, or production rules. These production rules, or Well-Formedness rules, specify which analytical structures may be formed - which analytical structures are possible - in each of the four analytical domains on the basis of a given musical score. Each domain also has a set of Preference Rules, which select between the possible analytical structures so as to achieve a single "preferred" analysis within each domain.

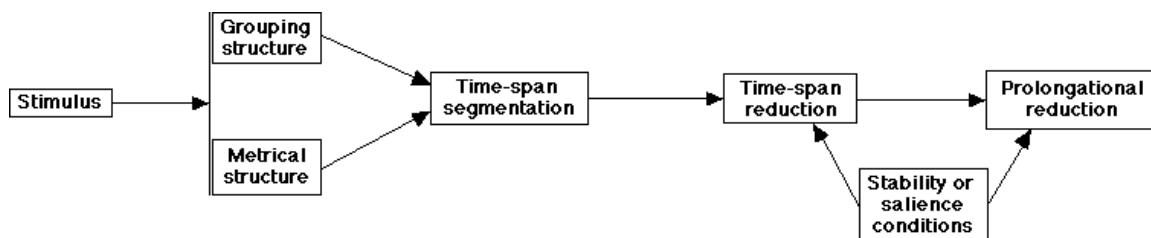


Figure 9.40: Main components of Lerdahl and Jackendoff's generative theory of tonal music.

GTTM proposes four types of hierarchical structures associated with a piece: the grouping structure, the metrical structure, the time-span reduction structure, and the prolongational reduction structure (fig. 7.40).

The grouping structure describes the segmentation units that listeners can establish when hearing a musical surface: motives, phrases, and sections.

The metrical structure describes the rhythm hierarchy of the piece. It assigns a weight to each note depending on the beat in which it is played. In this way notes played on strong (down) beats have higher weight than notes played on weak (up) beats.

The time-span reduction structure is a hierarchical structure describing the relative structural importance of notes within the audible rhythmic units of a phrase (see Fig. 7.41). It differentiates the essential parts of the melody from the ornaments. The essential parts are further dissected into even more essential parts and ornaments on them. The reduction continues until the melody is reduced to a skeleton of the few most prominent notes.

The prolongational reduction structure is a hierarchical structure describing tension-relaxation relationships among groups of notes. This structure captures the sense of musical flow across phrases, i.e. the build-up and release of tension within longer and longer passages of the piece, until a feeling of maximum repose at the end of the piece. Tension builds up as the melody departs from more stable notes to less stable ones and is discharged when the melody returns to stable notes. Tension and release are also felt as a result of moving from dissonant chords to consonant ones, from non accented notes to accented ones and from higher to lower notes.

The four domains - Metrical, Grouping, Time-Span and Prolongational - are conceived of as partially interdependent and at the same time as modelling different aspects of a listener's musical intuitions.

Each of these four components consists of three sets of rules:

Well-formedness Rules which state what sort of structural descriptions are possible. These rules define a class of possible structural descriptions.



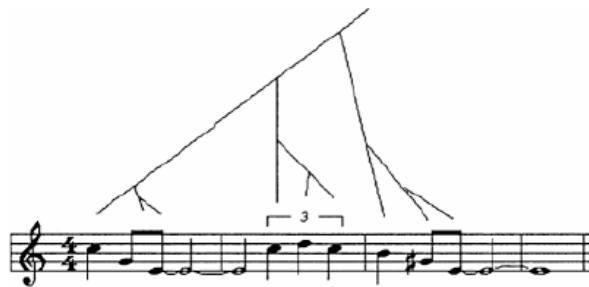


Figure 9.41: Example of a time-span tree for the beginning of the All of me ballad [from Arcos 1997].

Preference Rules which try to select from the possible structures the ones that correspond to what an experienced listener would hear. They are designed to work together to isolate those structural descriptions in the set defined by the well-formedness rules that best describe how an expert listener interprets the passage given to the theory as input.

Transformational Rules that allow certain distortions of the strict structures prescribed by the well-formedness rules.

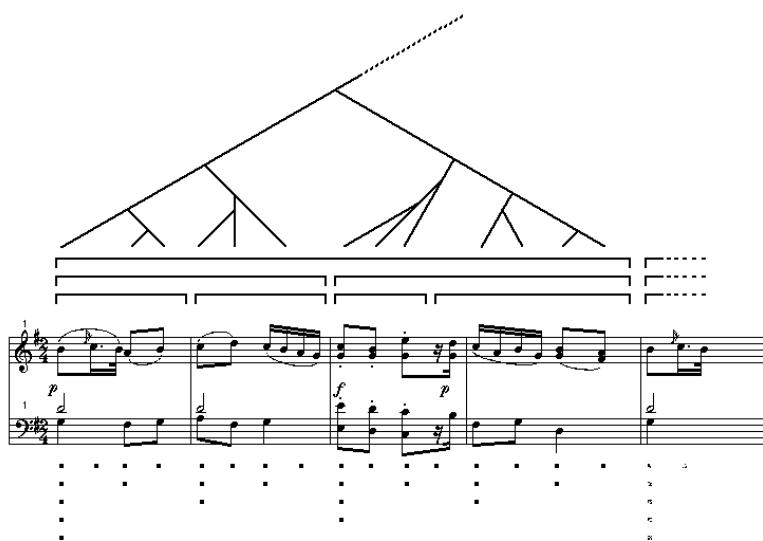


Figure 1(a)

Figure 9.42: Example of GTTM analysis of the first four bars of the second movement of Mozart's K.311: Metrical analysis (dots below the piece) and Time-Span analysis (tree-structure above the piece) [from Cross 1998].

The application of their theory to the first four bars of the second movement of Mozart's K.311 is shown in fig. 7.42 and 7.43. The Metrical analysis (shown in the dots below the piece in Figure 7.42) appears self-evident, deriving from Well-Formedness Rules such as those stating that "Every attack point must be associated with a beat at the smallest metrical level present at that point in the piece" (although the lowest, semiquaver, level is not shown in the figure), "At each metrical level, strong beats are spaced either two or three beats apart", etc. These Well-Formedness rules are supplemented by Preference rules, that suggest preference should be given to e.g., "metrical structures in which the strongest beat in a

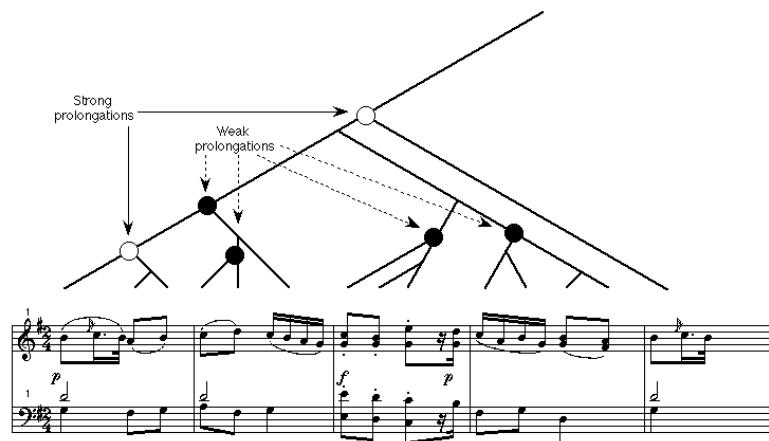


Figure 1(b)

Figure 9.43: Example of GTTM analysis of the first four bars of the second movement of Mozart's K.311: Prolongational analysis [from Cross 1998].

group appears relatively early in the group”, ”metrical structures in which strong beats coincide with pitch events”, etc..

The Grouping structure (shown in the brackets above the piece in Figure 7.42) appears similarly self-evident, being based on seemingly truistic Well-Formedness rules such as ”A piece constitutes a group”, ”If a group contains a smaller group it must contain all of that smaller group” (thus ensuring a strictly nested hierarchy), etc. Preference rules here specify such matters as the criteria for determining group boundaries (which should occur at points of disjunction in the domains of pitch and time), conditions for inferring repetition in the grouping structure, etc. Thus a group boundary is formed between the end of bar two and the beginning of bar three both in order to ensure the symmetrical subdivision of the first four bars (themselves specifiable as a group in part because of the repetition of the opening of bar one in bar five) and because the pitch disjunction occurring between the G and the C is the largest pitch interval that has occurred in the upper voice of the piece up to that moment. Perhaps the only point of interest in the Grouping analysis is the boundary between the third quaver of bar three and the last semiquaver of that bar, brought about by the temporal interval between the two events (again, the largest that has occurred in the piece up to that moment). Here, the Grouping structure and the Metrical structure are not congruent, pointing-up a moment of tension at the level of the musical surface that is only resolved by the start of the next group at bar five.

The Time-Span analysis (tree-structure above the piece in Figure 7.42) is intended to depict the relative salience or importance of events within and across groups. The Grouping structure serves as the substrate for the Time-Span analysis, the Well-Formedness rules in this domain being largely concerned with formalising the relations between Groups and Time-Spans. The Preference rules suggest that metrical and harmonically stable events should be selected as the ”heads” of Time-Spans, employment of these criteria resulting in the straightforward structure shown in the Figure. This shows clearly the shift in metrical position of the most significant event in each Group or Time-Span, from downbeat in bar one to upbeat crotchet in bars two and three to upbeat quaver in bar four.

A similar structure is evident in the Prolongational analysis (Figure 7.43), which illustrates the building-up and release of tension as a tonal piece unfolds. The Prolongational analysis derives in



part from the Time-Span analysis, but is primarily predicated on harmonic relations, which the Well-Formedness and Preference rules specify as either prolongations (tension-producing or maintaining) or progressions (tension-releasing).

Lerdahl and Jackendoff's theory however lack of a detailed, formal account of tonal-harmonic relations and tend to neglect of the temporality of musical experience. Moreover it let the analyst to make different choices that are quite difficult to formalize and implement on a computational model. Although the authors attempt to be thorough and formal throughout the theory, they do not resolve much of the ambiguity that exists through the application of the preference rules. There is little or no ranking of these rules to say which should be preferred over others and this detracts from what was presented as a formal theory.

9.6.2 Narmour's implication realization model

An intuition shared by many people is that appreciating music has to do with expectation. That is, what we have already heard builds expectations on what is to come. These expectations can be fulfilled or not by what is to come. If fulfilled, the listener feels satisfied. If not, the listener is surprised or even disappointed. Based on this observation, Narmour proposed a theory of perception and cognition of melodies based on a set of basic grouping structures, the Implication/Realization model, or I/R model³.

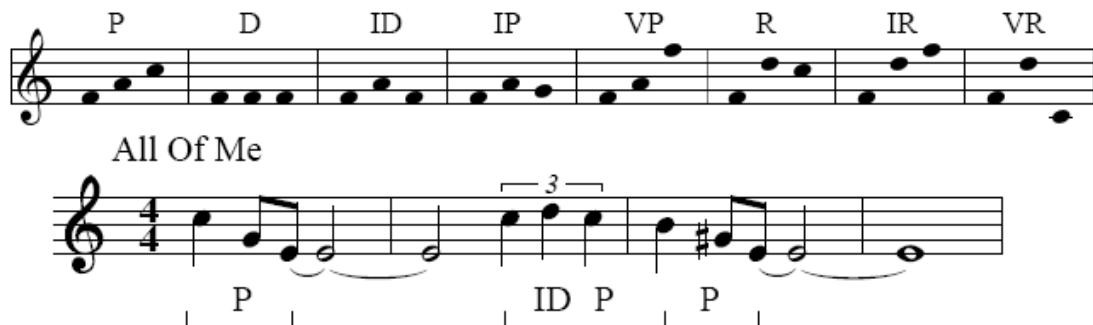


Figure 9.44: Top: Eight of the basic structures of the I/R model. Bottom: First measures of All of Me, annotated with I/R structures.

According to this theory, the perception of a melody continuously causes listeners to generate expectations of how the melody will continue. The sources of those expectations are two-fold: both innate and learned. The innate sources are hard-wired into our brain and peripheral nervous system, according to Narmour, whereas learned factors are due to exposure to music as a cultural phenomenon, and familiarity with musical styles and pieces in particular.

The innate expectation mechanism is closely related to the gestalt theory for visual perception. Narmour claims that similar principles hold for the perception of melodic sequences. In his theory, these principles take the form of implications: Any two consecutively perceived notes constitute a melodic interval, and if this interval is not conceived as complete, or closed, it is an implicative interval, an interval that implies a subsequent interval with certain characteristics. In other words, some notes are more likely to follow the two heard notes than others. Two main principles concern registral direction and intervallic difference.

- The principle of registral direction states that small intervals imply an interval in the same registral direction (a small upward interval implies another upward interval, and analogous for downward

³adapted from Mantaras AI Magazine 2001

intervals), and large intervals imply a change in registral direction (a large upward interval implies another upward interval and analogous for downward intervals).

- The principle of intervallic difference states that a small (five semitones or less) interval implies a similarly-sized interval (plus or minus 2 semitones), and a large intervals (seven semitones or more) implies a smaller interval.

Based on these two principles, melodic patterns can be identified that either satisfy or violate the implication as predicted by the principles. Such patterns are called structures and labelled to denote characteristics in terms of registral direction and intervallic difference. Eight such structures are shown in figure 7.44(top). For example, the P structure (Process) is a small interval followed by another small interval (of similar size), thus satisfying both the registral direction principle and the intervallic difference principle. Similarly the IP (Intervallic Process) structure satisfies intervallic difference, but violates registral direction.

Additional principles are assumed to hold, one of which concerns closure, which states that the implication of an interval is inhibited when a melody changes in direction, or when a small interval is followed by a large interval. Other factors also determine closure, like metrical position (strong metrical positions contribute to closure, rhythm (notes with a long duration contribute to closure), and harmony (resolution of dissonance into consonance contributes to closure).

These structures characterize patterns of melodic implications (or expectation) that constitute the basic units of the listener perception. Other resources such as duration and rhythmic patterns emphasize or inhibit the perception of these melodic implications. The use of the implication-realization model provides a musical analysis of the melodic surface of the piece.

The basic grouping structure are shown in fig. 7.44:

P (process) structure a pattern composed of a sequence of at least three notes with similar intervallic distances and the same registral direction;

ID (intervallic duplication) structure a pattern composed of a sequence of three notes with the same intervallic distances and different registral direction;

D (duplication) structure a repetition of at least three notes;

IP (intervallic process) structure a pattern composed of a sequence of three notes with similar intervallic distances and different registral direction;

R (reversal) structure a pattern composed of a sequence of three notes with different registral direction; the first interval is a leap, and the second is a step;

IR (intervallic reversal) structure a pattern composed of a sequence of three notes with the same registral direction; the first interval is a leap, and the second is a step;

VR (registral reversal) structure a pattern composed of a sequence of three notes with different registral direction; both intervals are leaps.

In fig. 7.44 (bottom) the first three notes form a P structure, the next three notes an ID, and the last three notes another P. The two P structures in the figure have a descending registral direction, and in both cases, there is a duration cumulation (the last note is significantly longer).

Looking at melodic grouping in this way, we can see how each pith interval implies the next. Thus, an interval can be continued with a similar one (such as P or ID or IP or VR) or reversed with a dissimilar one. That is, a step (small interval) is followed by a leap (large interval) between notes in the same



direction would be a reversal of the implied interval (another step was expected, but instead, a leap is heard) but not a reversal of direction. Pitch motion can also be continued by moving in the same direction (up or down) or reversed by moving in the opposite direction. The strongest kind of reversal involves both a reversal of intervals and of direction. When several small intervals (steps) move consistently in the same direction, they strongly imply continuation in the same direction with similar intervals. If a leap occurs instead of a step, it creates a continuity gap, which triggers the expectation that the gap should be filled in. To fill it, the next step intervals should move in the opposite direction from the leap, which also tends to limit pitch range and keeps melodies moving back toward a centre.

Basically, continuity (satisfying the expectation) is nonclosural and progressive, whereas reversal of implication (not satisfying the expectation) is closural and segmentative. A long note duration after reversal of implication usually confirm phrase closure.



Figure 9.45: Example of Narmour analysis of the first four bars of the second movement of Mozart's K.311 [from Cross 1998].

Any given melody can be described by a sequence of Narmour structures. Fig. 7.45 Narmour's analysis of the first four bars of the second movement of K.311 is shows. Letters (IP, P, etc.) within the "grouping" brackets identify the patterns involved, while the b's and d's in parentheses above the top system indicate the influence of, respectively, metre and duration. The three systems show the progressive "transformation" of pitches to higher hierarchical levels, and it should be noted that the steps involved do not produce a neatly nested hierarchy of the sort that Lerdahl and Jackendoff's theory provides.

9.7 Commented bibliography

A good tutorial on Hidden Markov Models is Rabiner [1989]. Hiller and Isaacson [1959] were the first to implement Markov chains in a musical application. The application of HMM rep-

resentation of musical theme for search, described in Sect. 7.5.3.1, is presented in Shifrin et al. [2002].

References

- Lejaren A. Hiller and L. M. Isaacson. *Experimental Music-Composition with an Electronic Computer*. McGraw-Hill, 1959.
- L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77 (2):257–286, 1989.
- J. Shifrin, B. Pardo, C. Meek, and W. Birmingham. Hmm-based musical query retrieval. In *Proc. ACM/IEEE Joint Conference on Digital Libraries*, pages 295–300, 2002.
- Iannis Xenakis. *Formalized Music*. Indiana University Press, 1971.



Chapter 10

Standards for audio and music representation

Giovanni De Poli

Copyright © 2005-2018 Giovanni De Poli

except for paragraphs labeled as *adapted from <reference>*

This book is licensed under the CreativeCommons Attribution-NonCommercial-ShareAlike 3.0 license. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/>, or send a letter to Creative Commons, 171 2nd Street, Suite 300, San Francisco, California, 94105, USA.

10.1 Digital audio compression

adapted from Noll (2000)

Audio compression or audio coding algorithms are used to obtain compact digital representations of high-fidelity (wideband) audio signals for the purpose of efficient transmission or storage. The central objective in audio coding is to represent the signal with a minimum number of bits while achieving transparent signal reproduction, i.e., generating output audio that cannot be distinguished from the original input, even by a sensitive listener ("golden ears").

10.1.1 Bit Rate Reduction

Typical audio signal classes are telephone speech, wideband speech, and wideband audio, all of which differ in bandwidth, dynamic range, and in listener expectation of offered quality. The quality of telephone-bandwidth speech is acceptable for telephony and for some video telephony and video-conferencing services. Higher bandwidths (7 kHz for wideband speech) may be necessary to improve the intelligibility and naturalness of speech. Wideband (high fidelity) audio representation including multi-channel audio needs bandwidths of at least 20 kHz.

The conventional digital format for these signals is PCM, with sampling rates and amplitude resolutions (PCM bits per sample) as given in Table 8.1. The compact disc (CD) is today's de facto standard of digital audio representation. On a CD with its 44.1kHz sampling rate the resulting stereo net bit rate is $2 \times 44.1 \times 16 \times 1000 = 1.41$ Mb/s. However, the CD needs a significant overhead for a run length-limited line code, which maps 8 information bits into 14 bits, for synchronization and for error

correction, resulting in a 49-bit representation of each 16-bit audio sample. Hence, the total stereo bit rate is $1.41 \times 49/16 = 4.32$ Mb/s. For archiving and processing of audio signals, sampling rates of at least 96 kHz and amplitude resolutions of up to 24 bit per sample are used. The Digital Versatile Disk (DVD) with its capacity of 4.7 GB (single layer) or 8.5 GB (double layer) is the appropriate storage medium for such applications.

Audio signal	Frequency range in Hz	Sampling rate in kHz	PCM (bit/sample)	PCM bit rate (kbit/sec)
Telephone speech	300 -3,400	8	8	64
Wideband speech	50-7,000	16	8	128
Wideband audio (stereo)	10-20,000	48	2×16	2×768
CD	10-20,000	44.1	2×16	2×705.6

Table 10.1: Basic parameters for three classes of acoustic signals.

Although high bit rate channels and networks become more easily accessible, low bit rate coding of audio signals has retained its importance. The main motivations for low bit rate coding are the need to minimize transmission costs or to provide cost-efficient storage, the demand to transmit over channels of limited capacity such as mobile radio channels, and to support variable-rate coding in packet-oriented networks. For these reasons many researcher worked toward formulation of compression schemes that can satisfy simultaneously the conflicting demands of high compression ratios and transparent reproduction quality for high-fidelity audio signals. A *lossless* or noiseless coding system is able to reconstruct perfectly the samples of the original signal from the coded (compressed) representation. In contrast, a coding scheme incapable of perfect reconstruction from the coded representation is denoted *lossy*.

Basic requirements of low bit rate audio coders are first, to retain a high quality of the reconstructed signal with robustness to variations in spectra and levels. In the case of stereophonic and multi-channel signals spatial integrity is an additional dimension of quality. Second, robustness against random and bursty channel bit errors and packet losses is required. Third, low complexity and power consumption of the codecs are of high relevance. For example, in broadcast and playback applications, the complexity and power consumption of audio decoders used must be low, whereas constraints on encoder complexity are more relaxed. Additional network-related requirements are low encoder/decoder delays, robustness against errors introduced by cascading codecs, and a graceful degradation of quality with increasing bit error rates in mobile radio and broadcast applications. Finally, in professional applications, the coded bit streams must allow editing, fading, mixing, and dynamic range compression.

First proposals to reduce wideband audio coding rates have followed those for speech coding. Differences between audio and speech signals are manifold; however, audio coding implies higher sampling rates, better amplitude resolution, higher dynamic range, larger variations in power density spectra, stereophonic and multi-channel audio signal presentations, and, finally, higher listener expectation of quality. Indeed, the high quality of the CD with its 16-bit per sample PCM format has made digital audio popular. Speech and audio coding are similar in that in both cases quality is based on the properties of human auditory perception. On the other hand, speech can be coded very efficiently because a speech production model is available, whereas nothing similar exists for audio signals.



10.1.2 Auditory Masking and Perceptual Coding

10.1.2.1 Auditory Masking

The inner ear performs short-term critical band analyses where frequency-to-place transformations occur along the basilar membrane. The power spectra are not represented on a linear frequency scale but on limited frequency bands called critical bands. The auditory system can roughly be described as a band-pass filter-bank, consisting of strongly overlapping bandpass filters with bandwidths in the order of 50 to 100 Hz for signals below 500 Hz and up to 5000 Hz for signals at high frequencies. Twenty-five critical bands covering frequencies of up to 20 kHz have to be taken into account.

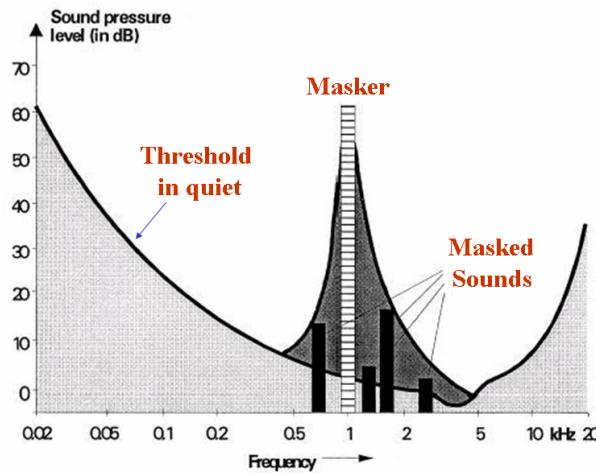


Figure 10.1: Threshold in quiet and masking threshold. Acoustical events in the shaded areas will not be audible.

Simultaneous masking is a frequency domain phenomenon where a low-level signal (the maskee) can be made inaudible (masked) by a simultaneously occurring stronger signal (the masker), if masker and maskee are close enough to each other in frequency. Such masking is greatest in the critical band in which the masker is located, and it is effective to a lesser degree in neighboring bands. A *masking threshold* can be measured below which the low-level signal will not be audible. This masked signal can consist of low-level signal contributions, quantization noise, aliasing distortion, or transmission errors. The masking threshold, in the context of source coding also known as threshold of just noticeable distortion (JND), varies with time. It depends on the sound pressure level (SPL), the frequency of the masker, and on characteristics of masker and maskee. Take the example of the masking threshold for the SPL = 60 dB narrowband masker in Fig. 8.1: around 1 kHz the four maskees will be masked as long as their individual sound pressure levels are below the masking threshold. The slope of the masking threshold is steeper towards lower frequencies, i.e., higher frequencies are more easily masked. It should be noted that the distance between masker and masking threshold is smaller in noise-masking-tone experiments than in tone-masking-noise experiments, i.e., noise is a better masker than a tone. In MPEG coders both thresholds play a role in computing the masking threshold.

Without a masker, a signal is inaudible if its sound pressure level is below the *threshold in quiet* which depends on frequency and covers a dynamic range of more than 60 dB as shown in the lower curve of Figure 8.1. The quiet (absolute) threshold is well approximated by the nonlinear function (see Fig. 8.3)

$$T_q(f) = 3.64(f/1000)^{-0.8} - 6.5^{-0.6}(f/1000-3.3)^2 + 10^{-3}(f/1000)^4 \text{ dB SPL}$$



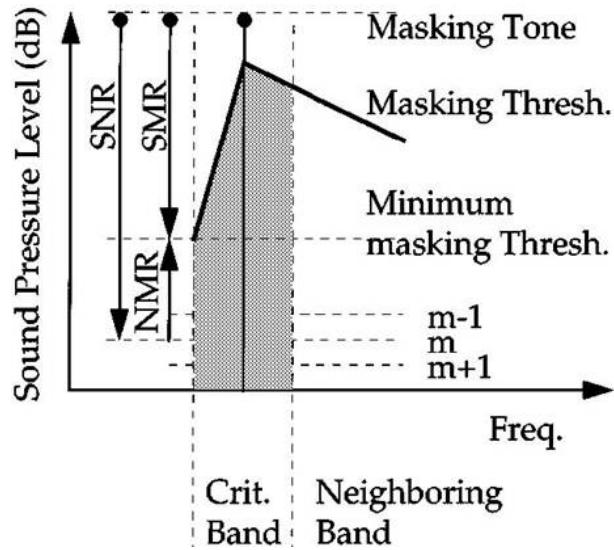


Figure 10.2: Masking threshold and signal-to-mask ratio (SMR). Acoustical events in the shaded areas will not be audible.

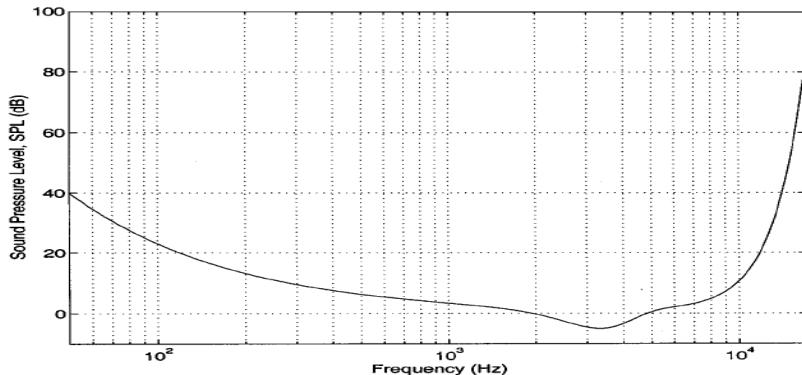


Figure 10.3: The absolute threshold of hearing in quiet. Across the audio spectrum, it quantifies the SPL required at each frequency such that an average listener will detect a pure tone stimulus in a noiseless environment. From (Painter-Spanias 2000).

The qualitative sketch of Fig. 8.2 gives a few more details about the masking threshold: a critical band, tones below this threshold (darker area) are masked. The distance between the level of the masker and the masking threshold is called Signal-to-Mask Ratio (*SMR*). Its maximum value is at the left border of the critical band (point A in Fig. 8.2), its minimum value occurs in the frequency range of the masker and is around 6dB in noise-masks-tone experiments. Assume a m -bit quantization of an audio signal. Within a critical band the quantization noise will not be audible as long as its signal to-noise ratio *SNR* is higher than its *SMR*. Noise and signal contributions outside the particular critical band will also be masked, although to a lesser degree, if their SPL is below the masking threshold.

Defining $SNR(m)$ as the signal-to-noise ratio resulting from an m -bit quantization, the perceivable distortion in a given subband is measured by the *Noise-to-Mask Ratio*:

$$NMR(m) = SMR - SNR(m) \quad (\text{in dB}).$$

The noise-to-mask ratio $NMR(m)$ describes the difference in dB between the signal-to-mask ratio and



the signal-to-noise ratio to be expected from an m -bit quantization. The NMR value is also the difference (in dB) between the level of quantization noise and the level where a distortion may just become audible in a given subband. Within a critical band, coding noise will not be audible as long as $NMR(m)$ is negative.

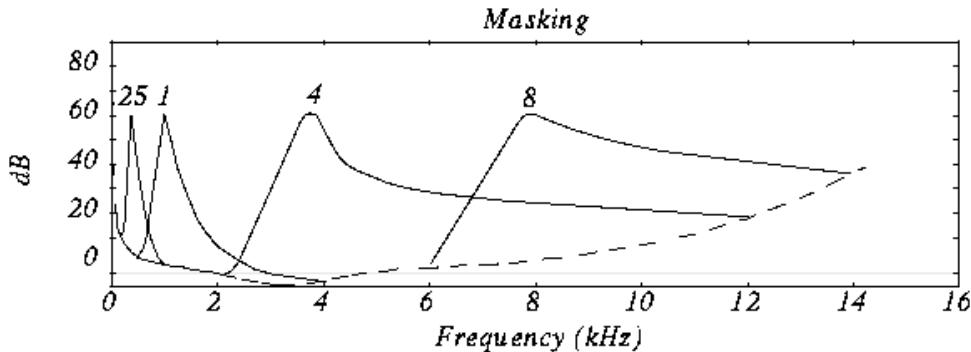


Figure 10.4: When many simultaneous maskers, each has its own masking threshold, and a global masking threshold can be computed.

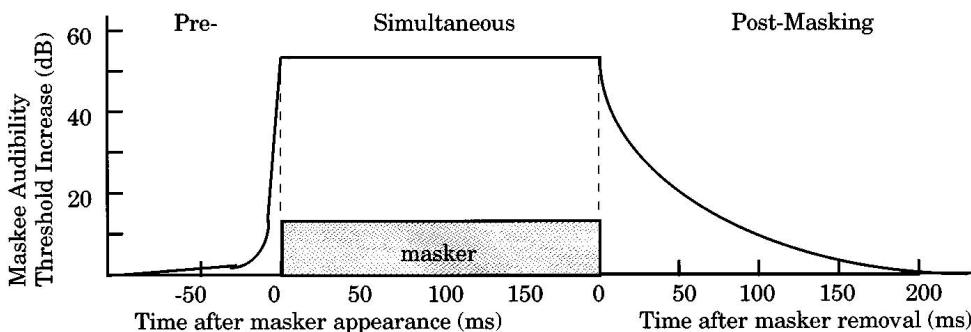


Figure 10.5: Temporal masking properties of the human ear. Backward (pre) masking occurs prior to masker onset and lasts only a few milliseconds; forward (post) masking may persist for more than 100 ms after masker removal.

We have just described masking by only one masker. If the source signal consists of many simultaneous maskers, each has its own masking threshold, and a *global masking threshold* can be computed that describes the threshold of just noticeable distortions as a function of frequency (see for example Fig. 8.4).

In addition to simultaneous masking, the time domain phenomenon of *temporal masking* plays an important role in human auditory perception. As shown in Fig. 8.5, masking phenomena extend in time beyond the window of simultaneous stimulus presentation. In other words, for a masker of finite duration, temporal masking occurs both prior to masker onset as well as after masker removal. The skirts on both regions are schematically represented in Fig. 8.5. Essentially, absolute audibility thresholds for masked sounds are artificially increased prior to, during, and following the occurrence of a masking signal. It may occur when two sounds appear within a small interval of time. Depending on the individual sound pressure levels, the stronger sound may mask the weaker one, even if the maskee precedes the masker (Fig. 8.5)! Temporal masking can help to mask pre-echoes caused by the spreading of a sudden large quantization error over the actual coding block. The duration within which pre-masking applies is significantly less than one tenth of that of the post-masking which is in the order of 50 to 200 ms. Both

pre-and post masking are being exploited in MPEG/Audio coding algorithms. In Fig. 8.6 the net effect of simultaneous and temporal masking is shown.

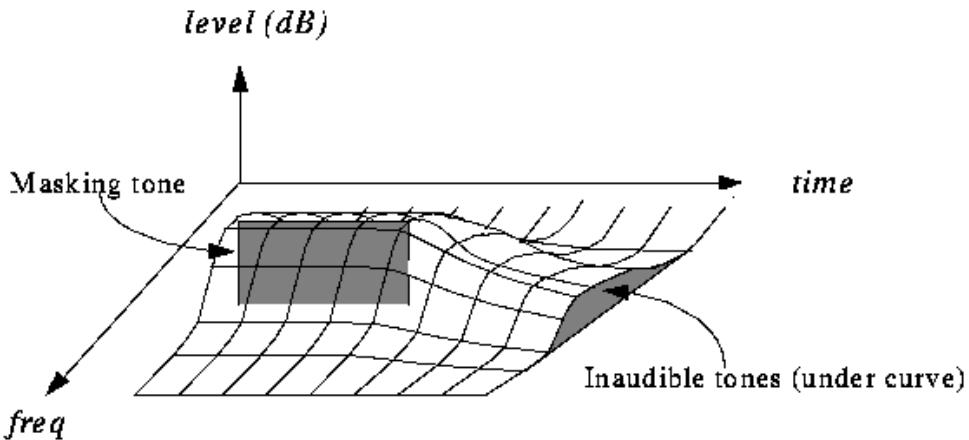


Figure 10.6: Net effect of simultaneous and temporal masking. Acoustical events under the surface will not be audible.

10.1.2.2 Perceptual Coding

Digital coding at high bit rates is dominantly waveform-preserving, i.e., the amplitude vs. time waveform of the decoded signal approximates that of the input signal. The difference signal between input and output waveform is then the basic error criterion of coder design. At lower bit rates, facts about the production and perception of audio signals have to be included in coder design, and the error criterion has to be in favour of an output signal that is useful to the human receiver rather than favouring an output signal that follows and preserves the input waveform. Basically, an efficient source coding algorithm will (1) remove *redundant* components of the source signal by exploiting correlations between its samples and (2) remove components that are *irrelevant* to the ear. Irrelevancy manifests itself as unnecessary amplitude or frequency resolution; portions of the source signal that are masked do not need to be transmitted.

The dependence of human auditory perception on frequency and the accompanying perceptual tolerance of errors can (and should) directly influence encoder designs; noise-shaping techniques can emphasize coding noise in frequency bands where that noise perceptually is not important. To this end, the noise shifting must be dynamically adapted to the actual short-term input spectrum in accordance with the signal-to-mask ratio which can be done in different ways. However, frequency weightings based on linear filtering, as typical in speech coding, cannot make full use of results from psychoacoustics. Therefore, in wideband audio coding, noise-shaping parameters are dynamically controlled in a more efficient way to exploit simultaneous masking and temporal masking.

Figure 8.7 depicts the structure of a perception-based coder that exploits auditory masking. The encoding process is controlled by the *signal to mask ratio* (SMR), the ratio of short term signal power within each frequency band and the masking threshold. From the SMR the needed amplitude resolution (and hence the bit allocation and rate) in each frequency band is derived. Moreover the number of bits assigned to each band is adapted to short-term spectrum of the audio coding block on a block-by-block basis. Such a *dynamic bit allocation* is used in all recent audio coding algorithms, which assign bits in order to maximize the overall perceptual quality. Algorithm 8.1 presents the pseudocode for the *PerceptualCoding* algorithm. Fig. 8.7 show the basic block diagram of a perceptual coding system. It consists of the following blocks:



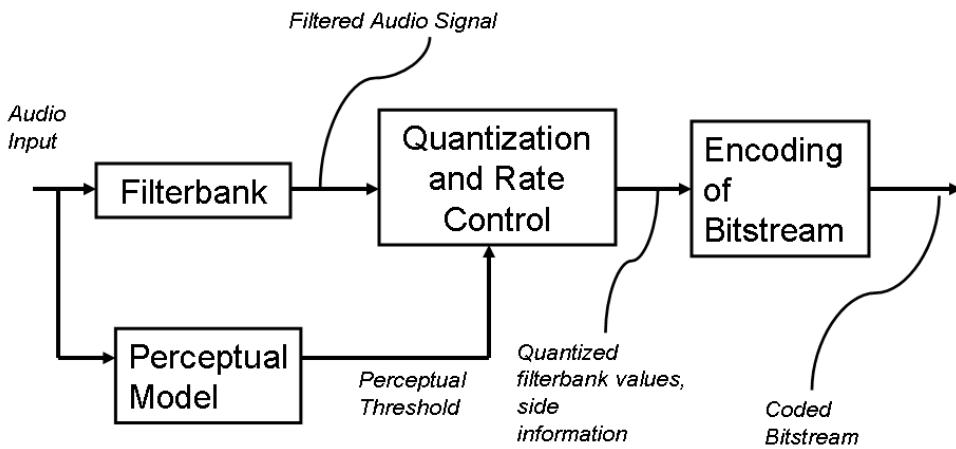


Figure 10.7: Block diagram of a generic perceptual coder.

Algorithm 10.1: PerceptualCoding

Data:

Result:

- 1 Use filters to divide the audio signal into frequency subbands;
 - 2 Determine amount of masking for each band caused by nearby band using the psycho-acoustic model;
 - 3 **if** the power in a band is below masking threshold **then**
 - 4 don't encode it
 - 5 **else**
 - 6 determine no. of bits needed to represent the coefficient such that noise introduced by quantization is below the masking effect (one fewer bit of quantization introduces about 6 dB of noise)
 - 7 Format bitstream;
-

- **Filter bank.** A filter bank is used to decompose the input signal into subsampled spectral components in time frequency domain. Together with the corresponding filter in the decoder, it forms as analysis/synthesis system.
- **Perceptual model** Using the time domain input signal and/or the output of the analysis filter bank, an estimate of the actual (time frequency dependent) masking threshold is computed using a perceptual model which implements known rules from psychoacoustics.
- **Quantization and coding.** The spectral components are quantized and coded with the aim of keeping the noise, which is introduced by quantization, below the masking threshold.
- **Encoding of beat stream.** A beat stream formatter is used to assemble the bitstream, which typically consists of the quantized and coded spectral components and some side information, e.g. bit allocation information.

The decoder (Fig. 8.8) simply reverses the formatting, then reconstructs the quantized subband values, and finally transforms the set of subband values into a time-domain audio signal.



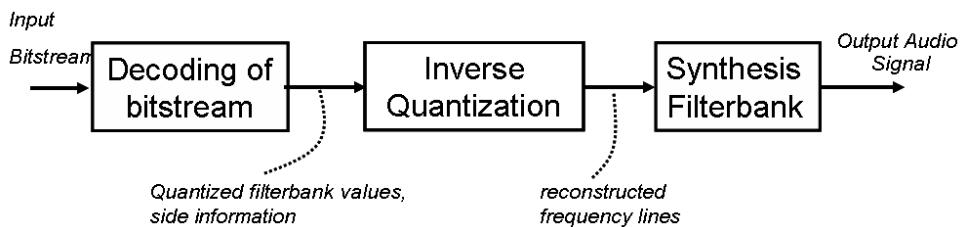


Figure 10.8: Block diagram of a generic perceptual decoder.

10.1.3 MPEG 1 Layer-3 coding

MPEG, a working group formally named as ISO/IEC JTC1/SC29/ WG11, but mostly known by its nickname, Moving Pictures Experts Group (MPEG), was set up by the ISO/IEC standardization body in 1988 to develop generic (i.e. useful for different applications) standards for the coded representation of moving pictures, associated audio and their combination. Since then, MPEG has undertaken the standardization of compression techniques for video and audio.

MPEG-1 Audio is an International Organization for Standardization (ISO) standard for high-fidelity audio compression. It is one part of a three-part compression standard. With the other two parts, video and systems, the composite standard addresses the compression of synchronized video and audio at a total bit rate of roughly 1.5 megabits per second. MPEG/audio compression is lossy; however, the MPEG algorithm can achieve transparent, perceptually lossless compression.

MPEG-1 Audio consists of three operating modes called *Layers*, with increasing complexity and performance, named Layer-1, Layer-2 and Layer-3. Layer-3, with the highest complexity, was designed to provide the highest sound quality at low bit-rates (around 128 kbit/s for a typical stereo signal). In general MP3 is appropriate for applications involving storage or transmission of mono or stereo music or other audio signals. Since it is implemented on virtually all digital audio devices playback is always ensured. Layer-3, or as it is mostly called nowadays mp3, is the most pervasive audio coding format for storage of music on PC platforms, and transmission of music over the Internet. Mp3 has created a new class of consumer electronics devices named after it, the mp3 player. It is found on almost all CD and DVD players and in an increasing number of car stereo systems and home stereo devices like networked home music servers. Additionally, Layer-3 finds wide application in satellite digital audio broadcast and on cellular phones.

10.1.3.1 The Psychoacoustic Model

The psychoacoustic model is the key component of the MPEG encoder that enables its high performance. The job of the psychoacoustic model is to analyze the input audio signal and determine where in the spectrum quantization noise will be masked and to what extent. The encoder uses this information to decide how best to represent the input audio signal with its limited number of code bits. The MPEG/audio standard provides two example implementations of the psychoacoustic model. Algorithm 8.2 (Determine_Masking_Threshold) outlines of the basic steps involved in the psychoacoustic



calculations for either model.

Algorithm 10.2: Determine_Masking_Threshold

Data: a frame (1152 samples) of the input signal

Result: Compute signal-to-mask ratio (SMR) for each subband

- 1 Time align audio data. ;
 - 2 Convert audio to spectral domain;
 - 3 Partition spectral values into critical bands;
 - 4 Separate into tonal and non-tonal components;
 - 5 Apply spreading function;
 - 6 Find the minimum masking threshold for each subband;
 - 7 Calculate signal-to-mask ratio
-

1. *Time align audio data* - The psychoacoustic model must account for both the delay of the audio data through the filter bank and a data offset so that the relevant data is centered within its analysis window. For example, when using psychoacoustic model two for Layer-1, the delay through the filter bank is 256 samples, and the offset required to center the 384 samples of a Layer-1 frame in the 512-point psychoacoustic analysis window is $(512 - 384)/2 = 64$ points. The net offset is 320 points to time align the psychoacoustic model data with the filter bank outputs.
2. *Convert audio to spectral domain* - The psychoacoustic model uses a time-to-frequency mapping such as a 512-or 1,024-point Fourier transform. A standard Hann weighting, applied to audio data before Fourier transformation, conditions the data to reduce the edge effects of the transform window. The model uses this separate and independent mapping instead of the filter bank outputs because it needs finer frequency resolution to calculate the masking thresholds.
3. *Partition spectral values into critical bands* - To simplify the psychoacoustic calculations, the model groups the frequency values into perceptual quanta (Fig 8.9).
4. *Incorporate threshold in quiet* - The model includes an empirically determined absolute masking threshold. This threshold is the lower bound for noise masking and is determined in the absence of masking signals.
5. *Separate into tonal and non-tonal components* - The model must identify and separate the tonal and noise-like components of the audio signal because the noise-masking characteristics of the two types of signal are different.
6. *Apply spreading function* - The model determines the noise-masking thresholds by applying an empirically determined masking or spreading function to the signal components.
7. *Find the minimum masking threshold for each subband* - The psychoacoustic model calculates the masking thresholds with a higher-frequency resolution than provided by the filter banks. Where the filter band is wide relative to the critical band (at the lower end of the spectrum), the model selects the minimum of the masking thresholds covered by the filter band. Where the filter band is narrow relative to the critical band, the model uses the average of the masking thresholds covered by the filter band.
8. *Calculate signal-to-mask ratio* - The psychoacoustic model takes the minimum masking threshold and computes the signal-to-mask ratio; it then passes this value to the bit (or noise) allocation section of the encoder.

A block scheme of the MPEG 1 Layer-3 encoder is shown in Fig. 8.10. Notice that a hybrid filter bank is introduced to increase frequency resolution and thereby better approximate critical band behavior. The hybrid filter bank is constructed by following each subband filter with an adaptive Modified Digital Cosine Transform (MDCT). This practice allows for higher frequency resolution and pre-echo control.



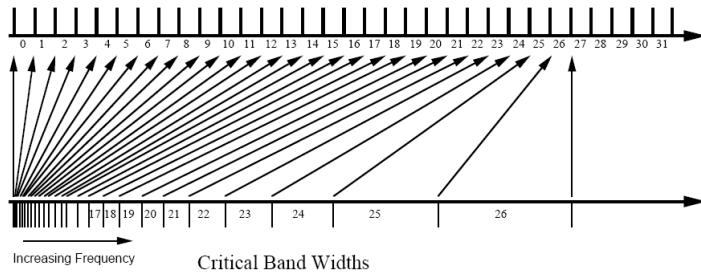


Figure 10.9: MPEG/Audio filter bandwidths versus critical bandwidths.

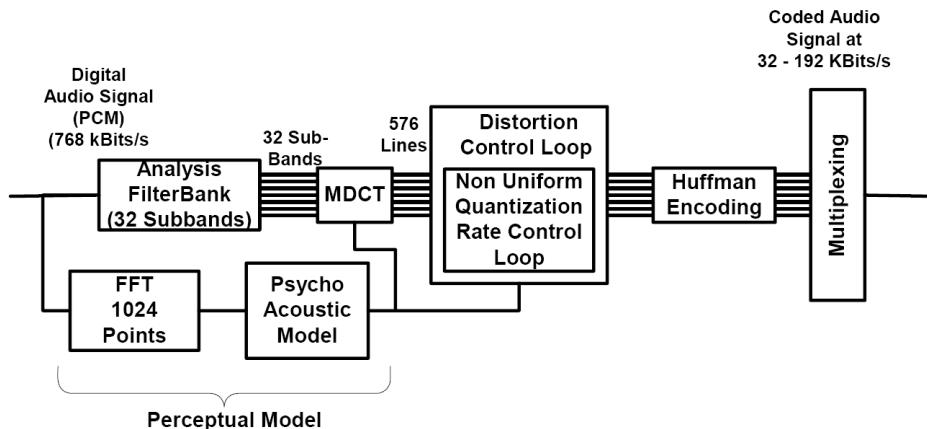


Figure 10.10: Block diagram of the MPEG 1 Layer-3 encoder.

Use of an 18-point MDCT, for example, improves frequency resolution to 41.67 Hz per spectral line. Bit allocation and quantization of the spectral lines are realized in a nested loop procedure that uses both nonuniform quantization and Huffman coding. The inner loop adjusts the nonuniform quantizer step sizes for each block until the number of bits required to encode the transform components falls within the bit budget. The outer loop evaluates the quality of the coded signal (analysis-by-synthesis) in terms of quantization noise relative to the JND thresholds.

10.1.3.2 The Problem of Stereo Coding

There are several new issues introduced when the issue of stereophonic reproduction is introduced:

- *The problem of Binaural Masking Level Depression (BLMD).* At lower frequencies, $f < 3000$ Hz, the Human Auditory System is able to take the phase of interaural signals into account. This can lead to the case where, for instance, a noise image and a tone image can be in different places. This can reduce the masking threshold by up to 20dB in extreme cases. BLMD can create a situation whereby a signal that was the same as the original in a monophonic setting sounds substantially distorted in a stereophonic setting. Two good, efficient monophonic coders do NOT make one good efficient stereo coder.
- *The problem of image distortion or elimination.* In addition to BLMD issues, a signal with a distorted high-frequency envelope may sound transparent in the monophonic case, but will not in general provide the same imaging effects in the stereophonic case.

Both the low-frequency BLMD and the high-frequency envelope effects behave quite similarly in terms of stereo image impairment or noise unmasking, when we consider signal envelope at high frequencies or waveforms themselves at low frequencies. The effect is not as strong between 500Hz and 2 kHz.

In order to control the imaging problems in stereo signals, several methods must be used. The MPEG/audio compression algorithm supports two types of stereo redundancy coding: intensity stereo coding and middle/side (M/S) stereo coding. Both forms of redundancy coding exploit the above seen perceptual weakness of the ear. Psychoacoustic results show that, within the critical bands covering frequencies above approximately 2 kHz, the ear bases its perception of stereo imaging more on the temporal envelope of the audio signal than its temporal fine structure. All layers support intensity stereo coding. Layer-3 also supports M/S stereo coding.

In *intensity stereo mode* or MPEG-1 Layer-1,2 *joint stereo mode*, the encoder codes some upper-frequency filter bank outputs with a single summed signal rather than send independent codes for left (L) and right (R) channels for each of the 32 filter bank outputs; i.e. the relative intensities of the L and R channels are used to provide high-frequency imaging information. Usually, one signal ($L + R$, typically) is sent, with two gains, one for L and one for R . The intensity stereo decoder reconstructs the left and right channels based only on independent left-and right-channel scale factors. With intensity stereo coding, the spectral shape of the left and right channels is the same within each intensity-coded filter bank signal, but the magnitude is different. Intensity stereo methods do not guarantee the preservation of the envelope of the signal for high frequencies. For lower quality coding, intensity stereo is a useful alternative to M/S stereo coding. We may think of intensity stereo as the coder equivalent of a pan-pot.

A psychoacoustic model that takes account of BMLD and envelope issues must be included. BMLD is best calculated and handled in the M/S paradigm. The *M/S stereo mode* is mid/side, or mono/stereo coding. It encodes the signals for left (L) and right (R) channels in certain frequency ranges as middle (M) and side (S) channels. where M and S are defined as:

$$\begin{aligned} M &= L + R \\ S &= L - R \end{aligned}$$

The normalization of 1/2 is usually done on the encoding side. In this mode, the encoder uses specially tuned techniques to further compress side-channel signal. A good stereo coder uses both M/S and L/R coding methods, as appropriate. M/S, while very good for some signals, creates either a false noise image or a substantial overcoding requirement for other signals. An M/S coder provides a great deal of redundancy elimination when signals with strong central images are present, or when signals with a strong "surround" component are present. Finally, an M/S coder provides better signal recovery for signals that have "matrixed" information present, by preserving the M and S channels preferentially to the L and R channels when one of M or S has the predominant energy.

10.1.3.3 Discussion on MPEG Layer-3

MPEG Layer-3 emerged as the main tool for Internet audio delivery. Considering the reasons, the following factors were definitely helpful.

- *Open standard* MPEG is defined as an open standard. The specification is available (for a fee) to everybody. While there are a number of patents covering MPEG Audio encoding and decoding, all patent-holders have declared that they will license the patents on fair and reasonable terms to everybody. Public example source code is available to help implementers to understand the text of the specification. As the format is well defined, no problems with interoperability of equipment or software from different vendors have been reported - except from some rare incomplete implementations.



- *Availability of encoders and decoders* DSP-based hardware and software encoders and decoders have been available for a number of years - driven at first by the demand for professional use in broadcasting.
- *Supporting technologies* While audio compression is viewed as a main enabling technology, other evolving technologies contributed to the MP3 boom, such as: the widespread use of computer sound cards; computers becoming powerful enough to run software audio decoders and even encoders in real-time; fast Internet access for universities and businesses; the availability of CD-ROM and CD-Audio writers.
- *Normative versus Informative* A very important property of the MPEG standards is the principle of minimizing the amount of normative elements in the standard. In the case of MPEG Audio, this led to the fact that only the data representation, i.e. the format of the compressed audio, and the decoder are normative. Even the decoder is not specified in a bit-exact fashion. Instead, formulae are given for most parts of the algorithm, and compliance is defined by a maximum deviation of the decoded signal from a reference decoder, implementing the formulae with double-precision arithmetic accuracy. This enables decoders running both on floating-point and fixed-point architectures. Depending on the skills of the implementers, fully-compliant high-accuracy Layer-3 decoders can be constructed with down to 20-bit arithmetic wordlength, without using double-precision calculations.

In short, MPEG Layer-3 had the luck to be the right technology available at the right time. In the meantime, research on perceptual audio coding progressed, and codecs with better compression efficiency became available. Of these, MPEG-2 Advanced Audio Coding (AAC) was developed as the successor of MPEG-1 Audio. Other - proprietary - audio coding schemes were also introduced, claiming a higher performance than MP3.

Quality Considerations: However, the pure compliance of an encoder with an MPEG audio standard does not guarantee any quality of the compressed music. Audio quality differs between different items, depending on basic parameters including, of course, the bit-rate of the compressed audio and the sophistication of different encoders, even if they work with the same set of basic parameters. To gain more insight into the level of quality possible with MP3, let us first have a look at typical artefacts associated with perceptual audio coders.

- *Common types of artefacts* Unlike analogue hi-fi equipment or FM broadcasting, perceptual encoders exhibit sound deficiencies when run at too low bit-rates or with the wrong parameters. These so-called "artefacts" are in most cases different from usual noise or distortion signals. Perceptual audio coding schemes such as MPEG Layer-3 introduce an error signal that can be described as a time-varying error at certain frequencies, which is not constrained to the harmonics of the music signal. The resulting music signal may sound: distorted, but not like harmonic distortions; noisy, but with the noise introduced only in a certain frequency range; rough, with the roughness often being very objectionable as the error may change its characteristics about every 24 ms.
- *Loss of bandwidth* If an encoder does not find a way to encode a block of music data with the required fidelity within the limits of the available bit-rate, it "runs out of bits". This may lead to the deletion of some frequency lines, typically affecting the high-frequency content. Compared to a constant bandwidth reduction, such an effect becomes more objectionable if the effective bandwidth changes frame-by-frame (e.g. every 24 ms).



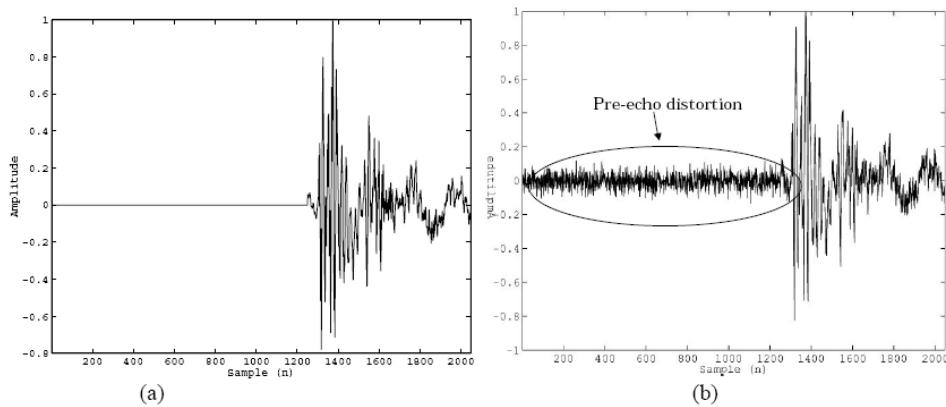


Figure 10.11: Pre-Echo Example: (a) Uncoded Castanets. (b) Transform Coded Castanets, 2048-Point Block Size.

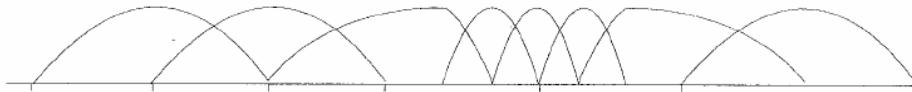


Figure 10.12: Changing the window length to match the signal properties of the input helps in avoiding pre-echoes. A long window is used to maximize coding gain and achieve good channel separation during segments identified as stationary, and a short window is used to localize time-domain artifacts when pre-echoes are likely.

- **Pre-echoes** Pre-echoes are very common artefacts, in the case of perceptual audio coding schemes using high-frequency resolution (see Fig. 8.11). The name "pre-echo", although somewhat misleading, nicely describes the artefact, which is a noise signal occurring even before the music event that causes such noise.

To understand the origin of Bitstream in pre-echoes, let us consider the decoder of a perceptual coding system (see Fig. 8.8). The reconstructed frequency lines are combined by the synthesis filterbank, consisting of a modulation matrix and a synthesis window. The quantization error introduced by the encoder can be seen as a signal added to the original frequency lines, with a length in time that is equal to the length of the synthesis window. Thus, reconstruction errors are spread over the full window length. If the music signal contains a sudden increase in signal energy (e.g. a castanet attack), the quantization error is increased as well. If such an attack occurs well within the analysis window, its error signal will be spread within the full synthesis window, preceding the actual cause for its existence in time (see Fig. 8.11). If such a pre-noise signal extends beyond the pre-masking period of the human ear, it becomes audible and is called *pre-echo*. There are a number of techniques to avoid audible pre-echoes, including changing the window length (Fig. 8.12), variable bit-rate coding or a local increase in the bit-rate to reduce the amplitude of the pre-echo. In general, these artefacts belong to the most difficult to avoid category.

- **Roughness, double-speak** Especially at low bit-rates and low sampling frequencies, there is a mismatch between time resolution of the coder and the time structure of some signals. This effect is most noticeable on speech signals and when listening via headphones. As a single voice tends to sound like it has been recorded twice and then overlaid, this effect is sometimes called *double-*



speak.

Using Perceptual Audio Coding: Perceptual audio coding is intended for final delivery applications. It is not advisable for principle recording of signals, or in cases where the signal will be processed heavily *after* the coding is applied. Perceptual audio coding is applicable where the signal will not be reprocessed, equalized, or otherwise modified before the final delivery to the consumer. If we are in a situation where we must do multiple encodings, we should avoid it to the extent possible and use a high bit rate for all but the final delivery bitstream. Finally, perceptual coding of audio is a very powerful technique for the final delivery of audio signals in situations where the delivery bit rate is limited, and/or when the storage space is small.

10.2 MIDI representation of music

Music is widely used in multimedia applications, so we require a media type for music to focus on the computers musical capabilities. We can distinguish two kinds of music representation, i.e. Operational vs. Symbolic. *Operational representations* specify exact timings for music and physical descriptions of the sounds to be produced, while *symbolic representations* use descriptive symbolism to describe the form of the music and allow great freedom in the interpretation. Both types are described as structural representations, since instead of representing music by audio samples there is information about the internal structure of the music

To illustrate the structural representations, we present MIDI, a widely used protocol allowing the connection of computers and musical equipment, an operational representation. The original Musical Instrument Digital Interface (MIDI) specification defined a physical connector and message format for connecting devices and controlling them in "real time". A few years later Standard MIDI Files were developed as a storage format so performance information could be recalled at a later date. The three parts of MIDI are often just referred to as "MIDI", even though they are distinctly different parts with different characteristics. Almost all recordings today uses MIDI as a key enabling technology for recording music. MIDI is also used in live performances to control stage lights and effects pedals.

The MIDI Message specification (or "MIDI Protocol") is probably the most important part of MIDI. Though originally intended just for use with the MIDI DIN transport as a means to connect two keyboards, MIDI messages are now used inside computers and cell phones to generate music, and transported over any number of professional and consumer interfaces (USB, FireWire, etc.) to a wide variety of MIDI-equipped devices. There are different message groups for different applications, only some of which will be explained here.

The final part of MIDI is the Standard MIDI File (and variants), which is used to distribute music playable on MIDI players of both the hardware and software variety. All popular computer platforms can play MIDI files (*.mid) and there are thousands of web sites offering files for sale or even for free. Anyone can make a MIDI file using commercial (or free) software that is readily available, and many people do, with a wide variety of results. The quality of a specific MIDI file can depend on how well it was created, and how accurately the synthesizer plays the file: not all synthesizers are the same, and unless the one used for listening at is similar to that of the file composer, what is heard may not be at all what he or she intended.

10.2.1 MIDI Messages

MIDI started out as a music description language in digital (binary) form. It was designed for use with keyboard-based musical instruments, so the message structure is oriented to performance events, such as picking a note and then striking it, or setting typical parameters available on electronic keyboards.



MIDI messages are used by MIDI devices to communicate with each other (Fig. 8.13). Every MIDI connection is a one-way connection from the MIDI Out connector of the sending device to the MIDI In connector of the receiving device. Each such connection can carry a stream of MIDI messages, with most messages representing a common musical performance event or gesture. All of those messages include *channel number*. There are 16 possible channels in the protocol. The channels are used to separate "voices" or "instruments", somewhat like tracks in a multi-track mixer. The ability to multiplex 16 channels onto a single wire makes it possible to control several instruments at once using a single MIDI connection. When a MIDI instrument is capable of producing several independent sounds simultaneously (a multitimbral instrument), MIDI channels are used to address these sections independently. This should not be confused with "polyphonic"; the ability to play several notes simultaneously in the same "voice".

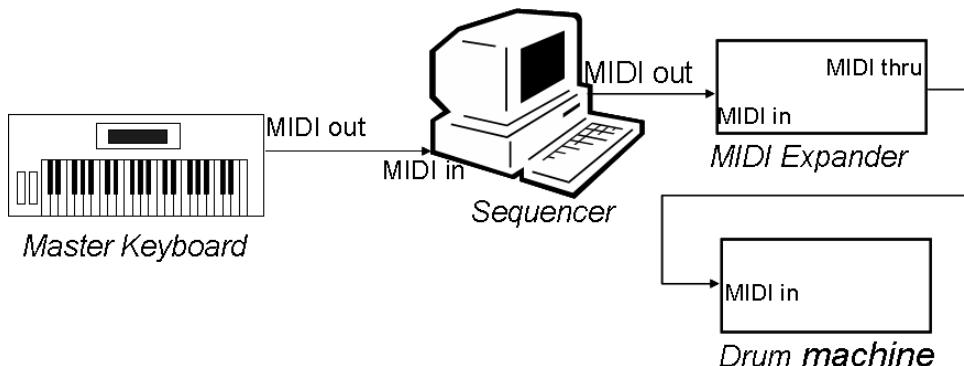


Figure 10.13: Example of a connection of MIDI devices.

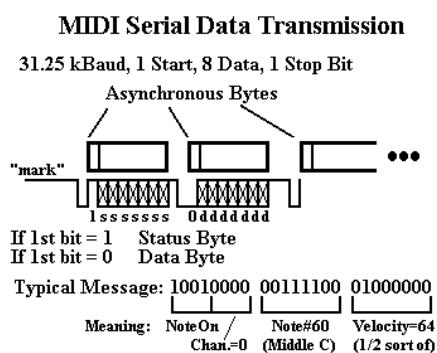


Figure 10.14: Midi messages.

MIDI specifies asynchronous serial communication using current-loop signalling at 31.25 Kbit/sec. There is 1 start bit, 8 data bits, and 1 stop bit; thus, each data byte takes 10 serial bits, and is sent in 320 microseconds (Fig. 8.14). Since the majority of MIDI messages need three bytes, only about a thousand messages per second can be transmitted.

A MIDI message can consist of from one to several thousand bytes of data. The receiving instrument knows how many bytes to expect from the value of the first byte of the message. This byte is known as the status byte, the others are data bytes. Status bytes always have the most significant bit equal to 1 and it is used to inform the receiver as to what to do with incoming data. Many of the commands include the channel number (0-15) as the 4 low-order bits of the status byte. The 3 remaining bits identify the message. The most significant bit of data byte is set to 0 and thus actual data values are limited to



numbers less than 128. This restricts many things in the MIDI universe, such as the number of presets.

There are a number of different types of MIDI messages. At the highest level, MIDI messages are classified as being either Channel Messages or System Messages. *Channel messages* are those which apply to a specific Channel, and the Channel number is included in the status byte for these messages. *System messages* are not Channel specific, and no Channel number is indicated in their status bytes.

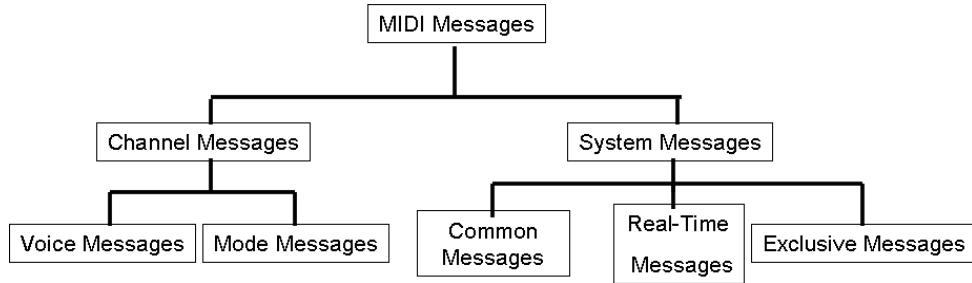


Figure 10.15: Organization of MIDI messages.

10.2.1.1 Channel messages

Channel Messages may be further classified as being either Channel Voice Messages, or Mode Messages. Channel Voice Messages carry musical performance data, and these messages comprise most of the traffic in a typical MIDI data stream. The messages in this category are the Note On, Note Off, Polyphonic Key Pressure, Channel Pressure, Pitch Bend Change, Program Change, and the Control Change messages. Channel Mode messages affect the way a receiving instrument will respond to the Channel Voice messages.

Voice Message	Status Byte	Mess. size	Data Byte1	Data Byte2
Note off	0x8c	2	Key number	Note Off velocity
Note on	0x9c	2	Key number	Note On velocity
Polyphonic Key Press.	0xAc	2	Key number	Amount of pressure
Control Change	0xBc	2	Controller	Controller value
Program Change	0xCc	1	Program num.	—
Channel Pressure	0xDc	1	Pressure val.	—
Pitch Bend	0xEc	2	bend LSB	bend MSB

Table 10.2: MIDI channel voice messages: c indicates the channel number.

Note On and Note Off In MIDI systems, the activation of a particular note and the release of the same note are considered as two separate events. When a key is pressed on a MIDI keyboard instrument or MIDI keyboard controller, the keyboard sends a *Note On* message on the MIDI OUT port. The keyboard may be set to transmit on any one of the sixteen logical MIDI channels, and the status byte (0x9c) for the Note On message will indicate the selected Channel number *c*. The Note On status byte is followed by two data bytes, which specify *note number* (indicating which note was pressed) and *key velocity* (how hard the key was pressed). The note number is used in the receiving synthesizer to select which note should be played, and the velocity is normally used to control the amplitude of the note. The note numbers start with 0 representing the lowest C. Middle C is note number 60. So, a MIDI note number

of 69 is used for A440 tuning (i.e. the A note above middle C). For example to play note number 80 (= 0x50) with maximum velocity 127 (= 0x7F) on channel 14 (= 0xD), the MIDI device would send these three hexadecimal byte values: 0x9D 0x50 0x7F.

A tone with note number (MIDI pitch) p has frequency

$$f = 440 \times 2^{(p-69)/12} \text{ Hz} = 440 \times s^{p-69} \text{ Hz}$$

and a note with frequency f Hz has MIDI pitch

$$p = 69 + 12 \log_2(f/440)$$

where $s = \sqrt[12]{2} \approx 1.059$ is the semitone frequency ratio. Notice that frequencies, that are not in the tempered musical scale, can not be represented directly in MIDI.

When the key is released, the keyboard instrument or controller will send a *Note Off* message (status byte 0x8c). The Note Off message also includes data bytes for the note number and for the velocity with which the key was released. The Note Off velocity information is normally ignored. A velocity of zero in a Note On event is a Note Off event. This manner of thinking, requiring separate actions to start and stop a note, greatly simplifies the design of receiving instruments (the synthesizer does not have to keep time). Note Off is actually not used very much. Instead, MIDI allows for a shorthand, known as running status: a MIDI message can be sent without its Status byte (i.e., just its data bytes are sent) as long as the previous, transmitted message had the same Status.

Controllers with continuous values

Aftertouch. Some MIDI keyboard instruments have the ability to sense the amount of pressure which is being applied to the keys while they are depressed. This pressure information, commonly called "aftertouch", may be used to control some aspects of the sound produced by the synthesizer (vibrato, for example). If the keyboard has a pressure sensor for each key, then the resulting "polyphonic aftertouch" information would be sent in the form of *Polyphonic Key Pressure* messages (0xAc). These messages include separate data bytes for key number and pressure amount.

It is currently more common for keyboard instruments to sense only a single pressure level for the entire keyboard. This "Channel aftertouch" information is sent using the *Channel Pressure* message (0xDc), which needs only one data byte to specify the pressure value. Since the musician can be continually varying his pressure, devices that generate Polyphonic Key Pressure and Channel Pressure typically send out many such messages while the musician is varying his pressure.

Pitch Bend. The *Pitch Bend Change* message (0xEc) is normally sent from a keyboard instrument in response to changes in position of the pitch bend wheel. The pitch bend information is used to modify the pitch of sounds being played on a given Channel. The Pitch Bend message includes two data bytes to specify the pitch bend value. Two bytes are required to allow fine enough resolution to make pitch changes resulting from movement of the pitch bend wheel seem to occur in a continuous manner rather than in steps.

Control Change There is a group of commands called *Control Changes* (0xBc), that set a particular controller's value. A controller is any switch, slider, knob, etc, that implements some function (usually) other than sounding or stopping notes. Each command has two parts, defining which control to change and what to change it to. Controllers are numbered from 0 to 121, and some of them have defined purposes; e.g. control n. 1 = Modulation wheel; 2 = Breath controller; 4 = Foot controller, etc. The



values 122-127 are reserved for special mode messages which affect the way a synthesizer responds to MIDI data. A controller usually has a single data byte, giving a range of 0-127 as the value. This is rather coarse, so the controllers from 32 to 63 are reserved to give extra precision to those assigned from 0 to 31.

Program Change The sound of a synthesizer is determined by the connections between the modules and settings of the module controls. Very few current models allow repatching of the digital subroutines that substitute for modules, but they have hundreds of controls to set. The settings are just numbers, and are stored in the synthesizer memory. A particular group of settings is called a Patch, Preset, Voice, or Tone for different brands, but the official word is program. The *Program Change* message (0xBc) is used to specify the program (type of instrument) which should be used to play sounds on a given Channel. This message needs only one data byte which specifies the new program number.

10.2.1.2 System Messages

The preceding messages are Channel Voice Messages which apply only to instruments set to the specified channel. *System Messages* apply to all machines and carry information that is not channel specific, such as timing signal for synchronization, positioning information in pre-recorded MIDI sequences, and detailed setup information for the destination device.

System Real-Time Message	Status Byte
Timing Clock	0xF8
Start Sequence	0xFA
Continue Sequence	0xFB
Stop Sequence	0xFC
Active Sensing	0xFE
System Reset	0xFF

Table 10.3: *MIDI System Real-Time Messages.*

System Real Time messages are used to synchronize all of the MIDI clock-based equipment within a system, such as sequencers and drum machines. To help ensure accurate timing, System Real Time messages are given priority over other messages. The System Real Time messages are the Timing Clock, Start, Continue, Stop, Active Sensing, and the System Reset message. The Timing Clock message is the master clock which sets the tempo for playback of a sequence. The Timing Clock message is sent 24 times per quarter note. The Start, Continue, and Stop messages are used to control playback of the sequence. The Active Sensing signal is used to help eliminate "stuck notes" which may occur if a MIDI cable is disconnected during playback of a MIDI sequence.

System Common Message	Status Byte	Number of Data Bytes
MIDI Timing Code	0XF1	1
Song Position Pointer	0XF2	2
Song Select	0XF3	1
Tune Request	0XF6	None

Table 10.4: *MIDI System Common Messages.*



System common messages which are currently defined include: the MIDI Timing Code (0xF1), used for synchronization of MIDI equipment and other equipment such as audio or video tape machines; The Song Select message (0xF3), used with MIDI equipment, such as sequencers or drum machines, which can store and recall a number of different songs; the Song Position Pointer (0xF3), used to set a sequencer to start playback of a song at some point other than at the beginning; the Song Position Pointer (0xF2) followed by 2 data bytes containing a value related to the number of MIDI clocks which would have elapsed between the beginning of the song and the desired point in the song.

System exclusive messages are related to things that cannot be standardized and addition to the original MIDI specification. It is just a stream of bytes, all with their high bits set to 0, bracketed by a pair of system exclusive start and end messages (0xF0 and 0xF7). System Exclusive messages may be used to send data such as patch parameters or sample data between MIDI devices. Manufacturers of MIDI equipment may define their own formats for System Exclusive data. Manufacturers are granted unique identification (ID) which is included as part of the System Exclusive message.

A Simple MIDI Stream The hexadecimal bytes in Table 8.5 are a simple, typical stream of MIDI data. Each message must be sent at the real-time that the synthesizer is supposed to perform the action. Sophisticated sequencers will send the message slightly ahead of time to account for the duration of the message – one third of a millisecond per byte – and the response time of the particular synthesizer – perhaps several milliseconds. The perceptual improvement of accuracy under 5 or 10 milliseconds, to the untrained listener, is questionable.

MIDI bytes	Description
90 3C 40	Play middle C (note number 60, hexadecimal 3C) on channel 1 (the zero nibble of 90), at half the full velocity (velocity 64, hexadecimal 40).
43 40	Play note G above middle C (note number 67, hexadecimal 43) at velocity 64. Note that the status byte of 90 from the first message is still in effect, and did not have to be resent.
B9 07 33	Change the volume of MIDI channel 10 to 51 (hexadecimal 33). The MIDI volume controller number is 7.
B3 07 10	Change the volume of MIDI channel 8 to 16 (hexadecimal 10). Since the channel number is different than the previous message, running status could not be used.
90 3C 00	Stop playing middle C on channel 1. The last byte of zero indicates a "key down velocity of 0" which indicates a "note-off" event.
80 43 64	Stop playing the G above middle C (key number 67, hexadecimal 43). The key is released with a velocity of 64. Since very few, if any, synthesizers implement an interpretation of the note-off volume, this message is generally equivalent to 90 43 00.

Table 10.5: Example of a simple MIDI stream.

10.2.2 MIDI files

10.2.2.1 Standard MIDI Files

When MIDI messages are stored on disks, they are commonly saved in the Standard MIDI file (SMF) format, which is slightly different from native MIDI protocol, because the events are also time-stamped



for playback in the proper sequence. Music delivered by MIDI files is the most common use of MIDI today. Just about every personal computer is now equipped to play Standard MIDI files. One reason for the popularity of MIDI files is that, unlike digital audio files (.wav, .aiff, etc.) or even compact discs or cassettes, a MIDI file does not need to capture and store actual sounds. Instead, the MIDI file can be just a list of events which describe the specific steps that a soundcard or other playback device must take to generate certain sounds. This way, MIDI files are very much smaller than digital audio files, and the events are also editable, allowing the music to be rearranged, edited, even composed interactively, if desired.

MIDI files are typically created using desktop/laptop computer-based sequencing software (or sometimes a hardware-based MIDI instrument or workstation) that organizes MIDI messages into one or more parallel "tracks" for independent recording and editing. In most but not all sequencers, each track is assigned to a specific MIDI channel and/or a specific General MIDI instrument patch. An SMF consists of one header chunk and one or more track chunks. There are three SMF formats; the format is encoded in the file header. Format 0 contains a single track and represents a single song performance. Format 1 may contain any number of tracks, enabling preservation of the sequencer track structure, and also represents a single song performance. Format 2 may have any number of tracks, each representing a separate song performance.

Many values are stored in a variable-length format which may use one or more bytes per value. Variable-length values use the lower 7 bits of a byte for data and the top bit to signal a following data byte. If the top bit is set to 1, then another value byte follows. Table 8.6 presents some examples to help demonstrate how variable length values are used.

Value		Variable length	
Hex	Bin	Hex	Bin
00	0000 0000	00	0000 0000
48	0100 1000	48	0100 1000
7F	0111 1111	7F	0111 1111
80	1000 0000	81 00	1000 0001 0000 0000
C8	1100 1000	81 48	1000 0001 0100 1000

Table 10.6: Examples of values and their variable-length equivalents.

Track events are used to describe all of the musical content of a MIDI file, from tempo changes to sequence and track titles to individual music events. Each event includes a delta time, event type and usually some event type specific data. The *event delta time* is defined by a variable-length value. It determines when an event should be played relative to the track's last event. A delta time of 0 means that it should play simultaneously with the last event. A track's first event delta time defines the amount of time to wait before playing this first event. Events unaffected by time are still preceded by a delta time, but should always use a value of 0 and come first in the stream of track events. Examples of this type of event include track titles and copyright information. The most important thing to remember about delta times is that they are relative values, not absolute times. The actual time they represent is determined by a couple factors: the time division (defined in the MIDI header chunk) and the tempo (defined with a track event). If no tempo is defined, 120 beats per minute is assumed.

There are three types of events: MIDI Channel Events, System Exclusive Events and Meta Events. The first two types are represented as explained in the previous section. Meta Events include specifications for tempo, time and key signature, lyrics, sequence and track numbers, score markers, copyright notice.



As an example, MIDI Files for the following excerpt are shown in Table 8.7. A format 0 file is used, with all information intermingled. A resolution of 96 ticks per quarter note is used. A time signature of 4/4 and a tempo of 120, though implied, are explicitly stated.

Delta Time (decimal)	Event Code (hex)	Other Bytes (decimal)	Comment
0	FF 58	04 04 02 24 08	4 bytes: 4/4 time, 24 MIDI clocks/click, 8 32nd notes/24 MIDI clocks
0	FF 51	03 500000	3 bytes: 500,000 5sec per quarter-note
0	C0	5	Ch. 1, Program Change 5
0	C0	5	Ch. 1, Program Change 5
0	C1	46	Ch. 2, Program Change 46
0	C2	70	Ch. 3, Program Change 70
0	92	48 96	Ch. 3 Note On C2, forte
0	92	60 96	Ch. 3 Note On C3, forte
96	91	67 64	Ch. 2 Note On G3, mezzo-forte
96	90	76 32	Ch. 1 Note On E4, piano
192	82	48 64	Ch. 3 Note Off C2, standard
0	82	60 64	Ch. 3 Note Off C3, standard
0	81	67 64	Ch. 2 Note Off G3, standard
0	80	76 64	Ch. 1 Note Off E4, standard
0	FF 2F	00	Track End

Table 10.7: Examples of midi file events.

10.2.2.2 General MIDI

General MIDI (GM) is a response to a problem that arose with the popularity of the Standard MIDI file. As composers began exchanging compositions (and selling them) in SMF format, they discovered that pieces would change when played on different synthesizers. That's because the MIDI program commands simply provide a number for a preset. What sound you get on preset four is anybody's guess.

General MIDI is a specification for synthesizers which imposes several requirements beyond the more abstract MIDI standard. While MIDI itself provides a protocol which ensures that different instruments can interoperate at a fundamental level (e.g. that pressing keys on a MIDI keyboard will cause an attached MIDI sound module to play musical notes), GM goes further in two ways: it requires that all GM-compatible instruments meet a certain minimal set of features, such as being able to play at least 24 notes simultaneously (polyphony), and it attaches certain interpretations to many parameters and control messages which were left unspecified in MIDI.

The *Instrument patch map* is a standard program list consisting of 128 patch types. The patches are arranged into 16 "families" of instruments, with each family containing 8 instruments. Program numbers 1-8 are piano sounds, 9-16 are chromatic percussion sounds, 17-24 are organ sounds, 25-32 are guitar sounds, etc.. For example among the 8 instruments within the Reed family, you will find Saxophone, Oboe, and Clarinet. Channel 10 is reserved for percussion under GM; this channel always sounds as percussion regardless of whatever program change numbers it may be sent, and different note numbers are interpreted as different instruments.



The *Percussion map* specifies 47 percussion sounds, e.g. A note-on with note number 402 will trigger a Electric Snare. Note that GM specifies which instrument or sound corresponds with each program/patch number, but it does not specify how these sounds are produced. Thus for example a Clarinet sound on two GM synthesizers which use different synthesis techniques may sound quite different. GM also specifies which operations should be performed by several controllers.

GM Standard makes it easy for musicians to put Program Change messages in their MIDI (sequencer) song files, confident that those messages will select the correct instruments on all GM sound modules, and the song file would therefore play all of the correct instrumentation automatically. Finally, musicians didn't have to worry that a snare drum part, for example, would be played back on a Cymbal. All of these standards help to ensure that MIDI Files play back properly upon setups of various equipment. GM is most important in the soundcards that plug into PCs. These allow game programmers to create MIDI based scores instead of including recorded sounds for the music cuts. On the other hand it greatly limits the possible freedom of the composer.

10.2.2.3 MIDI Timing

Any MIDI device which records data (e.g. a sequencer) needs timing information so that the data can be replayed at the correct rate. Those MIDI devices which do not record data (e.g. synthesizers and effects units) do not use timing information. Timing information can be transferred between MIDI devices via the MIDI link. There is no need for separate synchronization connections. MIDI provides two ways to count time: via MIDI Clock messages or via MIDI Timecode.

MIDI Clock signals are not simple pulses. Each clock message is a single byte. Clock messages are sent at the rate of 6 per semiquaver (a semiquaver = "sixteenth note"), i.e. a rate of 24 per quarter note. So, a MIDI sequencer refers to a position in a sequence of events in terms of musical divisions such as beats and bars. The MIDI beat does not therefore occupy a fixed amount of time. Its duration depends on the speed ("tempo") of the music.

This is in contrast with SMPTE timecode where the timing information relates to *absolute time* (measured in hours, minutes, seconds, frames). Both methods of determining position are useful in their respective applications but in the modern studio it is necessary to reconcile the two. Nowadays recording studios integrate MIDI systems and audio recording systems. This integration is achieved by relating MIDI timing data to timecode recorded onto tape.

MIDI timecode (MTC) were introduced as a way of converting SMPTE timecode into MIDI messages. MTC generates absolute time signals that synchronize SMPTE with MIDI devices. There are two kinds of MTC message: *quarter-frame messages*, which update the receiving device regularly when the transmitting device is running at normal speed, and *full-frame messages* are used during spooling, when updating at the quarter-frame rate would result in an excessive rate of data transmission. Timing information received from a tape recorder using conventional timecode (SMPTE or EBU) is thus distributed as MIDI timecode (MTC) around a MIDI system.

In addition to the synchronization of tape recorders and sequencers, MTC is increasingly being used in a variety of automated equipment. Mixing console automation is currently resulting in interesting developments in the integration of mixer control and MIDI sequencing. MTC can be used to control channel routing and muting, leaving the engineer free to experiment with levels, EQ, panning etc. A number of mixer settings can be prepared in advance away from the mixing console (prepared "off-line").



10.2.3 Discussion on MIDI representation

10.2.3.1 MIDI limitations

While MIDI performance capabilities has been a great boom for computer music software applications since 1984, it has limitations when compared to replicating authentic sounding real-time live musical performances.

Bandwidth limitations. MIDI messages are extremely compact, due to the low bandwidth of the connection, and the need for real-time accuracy. However, the serial nature of MIDI messages means that long strings of MIDI messages take an appreciable time to send, at times even causing audible delays, especially when dealing with dense musical information or when many channels are particularly active. Notice that every Note On messages takes about 1 ms to transmit. The protocol was designed to record the keyboard performances of one to four human keyboard players without much continuous controller manipulation. The bandwidth can be overwhelmed by a single virtuoso.

Performance nuances limitation. MIDI measures many variables between 0-127 (not note placement.). So the velocity of a kick drum is always between 0-127 which might not seem like a big deal but if you emulating real instruments something closer to infinity is necessary. Interpretative ingredients of music performances can not be easily included or controlled in MIDI files unless done with individual adjustments made to every note. Such a process is not practical because it is so time consuming with the specific attention directed to single notes with regards to attack, delay, sustaining, vibrato, pulse and other creative performance elements. Consequently, MIDI file performances can easily sound contrived, predictable musical expressiveness or mechanical rather than like live performances where the individual musical nuances are created for particular interpretations.

Music representation limitations. A more fundamental constraint of MIDI is its concept of music embodied in its design. It is biased toward popular songs (metered, in equal temperament) as played on a musical keyboard. Moreover MIDI lacks of representation of timbre. General MIDI mode table was designed for home entertainment and not for professional musicians: it includes only a tiny fractions of the timbres possible in computer music. MIDI concept of pitch is based on equal temperament. It is possible to detune a pitch with a Pitch Bend message, but it is a global operation applying to all notes on a channel. One of the justification of computer music is the ability to go beyond the pitch, timing and timbre limitations of traditional instruments.

10.2.3.2 Operation on music

In considering music representation, we can recognize several advantages over audio: music representation will be more compact than audio; it is portable and can be synthesized with the fidelity and complexity appropriate to the output devices used; while digital audio suffers from inherent noise, musical representations are noise free; many operations can be performed on music that would be infeasible or require extensive processing on audio.

- Playback and Synthesis. During audio playback, the listener has limited influence over the musical aspects of the performance, beyond changing the volume or processing the audio in some way. If music is produced by synthesis from a structural representation the listener can independently change pitch and tempo, increase or decrease individual instruments volumes or change the sounds they produce. Musical representations offer greater potential for interactivity than audio
- Timing. Structural representation makes timing of musical events explicit. The ability to modify tempo makes it possible to alter the timing of groups of musical events and adjust the synchronization of those events with other events (film, video, etc.)



- **Editing and Composition.** Basic editing allows the user to modify primitive events and notes; more complex editing operations operate on musical aggregates (chords, bars, etc.) to permit phrase-repetition, melody replacement and other such functions. Composition software simplifies the task of generating and combining or rearranging tracks, and prints the score

MIDI sequence editing A sequencer is a multitrack recorder for MIDI information. It allows you to record, overdub, and edit MIDI performances on different channels and play them back simultaneously, triggering sounds from your MIDI instrument. Sequencers are often included in many of today's keyboards, but the most popular way to record MIDI is by using computer-based sequencing software.

In addition to channels, MIDI sequencers use tracks and patterns as organizing concept. *Tracks* are arbitrary units of musical organization in a sequencer. It may refer to separate recording versions of a musical line, with the idea of selecting one track for the final mix at a later stage. Another way is refers to different layers of a composition, such as a track for each synthesizer voice, separate tracks for each controller etc..

Pattern orientation means that the data can be broken into subsequences within a sequence. A pattern can be looped, played back in combination with other patterns. Moreover a composer can add, delete, transpose or transform a pattern The same as note events.

For editing purposes, MIDI information can be displayed in several representations. An *event list* represent MIDI data in an alphanumeric listing, sorted in time order. It is used to fine tuning a sequence. In *Piano roll* representation, individual notes are laid down vertically, while the start time and duration of events are encoded as a horizontal line (Fig. 8.16). Some sequencers can display MIDI data into a form of common music notation. To this purpose metronome information is useful and time quantization is necessary to take into account performer's deviations from exact metrical time.

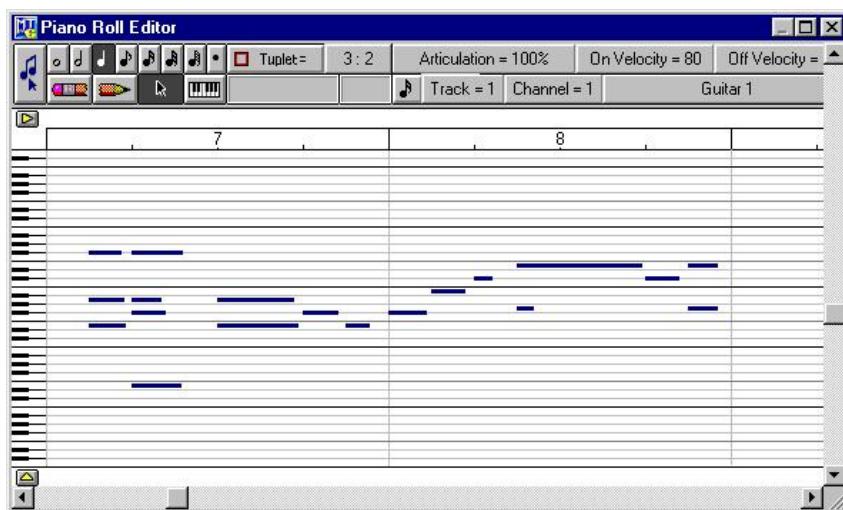


Figure 10.16: Piano roll representation of Midi data.

10.3 MusicXML: a music interchange file format

MusicXML is a music interchange format designed for notation, analysis, retrieval, and performance applications. The MusicXML format has been developed by Recordare. It is open for use by anyone under a royalty-free license. It actually constitutes the most promising solution for music score interchange. MusicXML was designed from the ground up for sharing sheet music files between applications, and for

archiving sheet music files for use in the future. You can count on MusicXML files being readable and usable by a wide range of music notation applications, now and in the future. MusicXML complements the native file formats used by Finale and other programs, which are designed for rapid, interactive use. Just as MP3 files have become synonymous with sharing recorded music, MusicXML files have become the standard for sharing interactive sheet music.

The goals of MusicXML approach are:

- A universal translator for common Western musical notation
- Supports notation, analysis, information retrieval, and performance applications
- Augments, but does not replace, specialized proprietary formats
- Adequate, not optimal, for diverse music applications

MusicXML is based on XML. Using XML frees users and developers from worrying about the basic syntax of the language, instead letting us worry about the semantics - what to represent, versus the syntax of how to represent it. Similarly, there is no need to write a low-level parser to read the language: XML parsers exist everywhere. Basing a music interchange language on XML lets music software, a relatively small community, leverage the investment in tools made by much larger markets such as electronic commerce.

10.3.1 MusicXML structure

Say we have a piece of music for two or more people to play. It has multiple parts, one per player, and multiple measures. XML represents data in a hierarchy, but musical scores are more like a lattice. How do we reconcile this? Should the horizontal organization of musical parts be primary, or should the vertical organization of musical measures (Fig. 8.17)? The answer is different for every music application.

MusicXML allows both views and the possibility of switching between them easily. This is why MusicXML has two different top-level DTDs, each with its own root element. If you use the partwise DTD, the root element is `<score-partwise>`. The musical part is primary, and measures are contained within each part. If you use the timewise DTD, the root element is `<score-timewise>`. The measure is primary, and musical parts are contained within each measure. The MusicXML XSD includes both of the top-level document elements in a single XSD file.

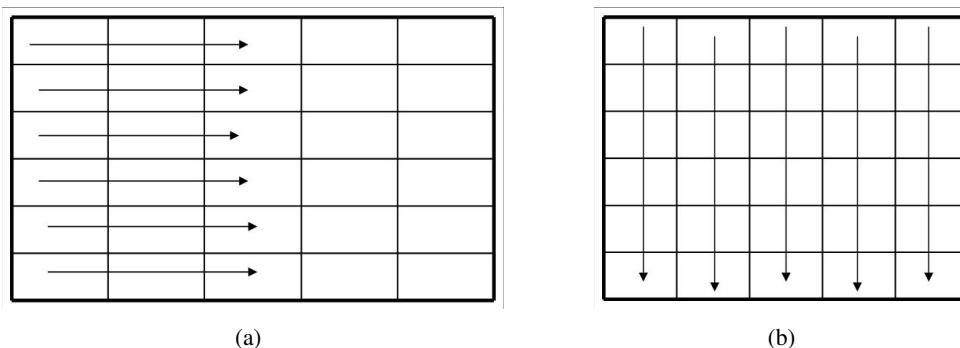


Figure 10.17: Partwise (a) and timewise (b) score organization.

The figure shows a side-by-side comparison. On the left is a snippet of MusicXML XML code illustrating the part-wise organization of a score. On the right is a corresponding musical notation in 2/4 time with a key signature of one sharp, featuring two measures of music.

```

<score-partwise>
  <identification>...</identification>
  <part-list>...</part-list>
  <part id="P1">
    <measure number="1">
      <attributes>...</attributes>
      <note>...</note>
      <note>...</note>
      <note>...</note>
      <note>...</note>
    </measure>
    <measure number="2">
      <note>...</note>
      <note>...</note>
    </measure>
  </part>
  <part id="P2">...</part>
</score-partwise>

```

Figure 10.18: The partwise organoization of a score in MusicXML. Each part listed serially. A part consists of measures and the measures contain notes and attributes.

The MusicXML score structure of part-wise organization is shown if Fig. 8.18. This is the most common organization. Each part listed serially. A part consists of measures and the measures contain

- <note>s (items with duration) and
- <attribute>s (items without duration such as clef, time signature, key signature, etc.)
- <direction>s (dynamics)
- <sound/> (tempo)

The figure shows the XML structure of a note element next to its corresponding musical notation. The XML code includes attributes for pitch (step=C, octave=5), duration (1/16th note), and graphical duration (indicated by a vertical stem). Annotations like articulations and staccato placement are also shown. Red arrows point from the XML labels to the corresponding musical elements.

```

<note>
  <pitch>
    <step>C</step>
    <octave>5</octave>
  </pitch>
  <duration>1</duration>           logical duration
  <voice>1</voice>
  <type>16th</type>              graphical duration
  <stem>down</stem>
  <notations>
    <articulations>
      <staccato placement="above" />
    </articulations>
  </notations>
</note>

```

Figure 10.19: Example of note element.

An example of note element is shown in Fig. 8.19

10.3.2 MusicXML: a simple example

Brian Kernighan and Dennis Ritchie popularized the practice of writing a program that prints the words "hello, world" as the first program to write when learning a new programming language. It is the minimal program that tests how to build a program and display its results.

In MusicXML¹, a song with the lyrics "hello, world" is actually more complicated than we need for a simple MusicXML file. Let us keep things even simpler: a one-measure piece of music that contains a whole note on middle C, based in 4/4 time:

Here it is in MusicXML:

¹from <http://www.makemusic.com/musicxml/tutorial/hello-world>



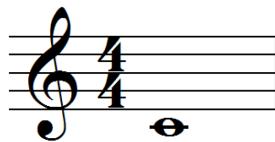


Figure 10.20: A one-measure piece of music that contains a whole note on middle C, based in 4/4 time.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE score-partwise PUBLIC
  "-//Recordare//DTD MusicXML 3.0 Partwise//EN"
  "http://www.musicxml.org/dtds/partwise.dtd">
<score-partwise version="3.0">
  <part-list>
    <score-part id="P1">
      <part-name>Music</part-name>
    </score-part>
  </part-list>
  <part id="P1">
    <measure number="1">
      <attributes>
        <divisions>1</divisions>
        <key>
          <fifths>0</fifths>
        </key>
        <time>
          <beats>4</beats>
          <beat-type>4</beat-type>
        </time>
        <clef>
          <sign>G</sign>
          <line>2</line>
        </clef>
      </attributes>
      <note>
        <pitch>
          <step>C</step>
          <octave>4</octave>
        </pitch>
        <duration>4</duration>
        <type>whole</type>
      </note>
    </measure>
  </part>
</score-partwise>
```

Let's look at each part in turn:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

This is the XML declaration required of all XML documents. We have specified that the characters are written in the Unicode encoding "UTF-8". This is the version of Unicode that has ASCII as a subset. Setting the value of standalone to "no" means that we are defining the document with an external definition in another file.

```
<!DOCTYPE score-partwise PUBLIC
  "-//Recordare//DTD MusicXML 3.0 Partwise//EN"
  "http://www.musicxml.org/dtds/partwise.dtd">
```

This is where we say that we are using MusicXML, specifically a partwise score where measures are contained within parts. We use a PUBLIC declaration including an Internet location for the DTD. The URL in this declaration is just for reference. Most applications that read MusicXML files will want to install a local copy of the MusicXML DTDs on the users machine. Use the entity resolver in your XML parser to validate against the local copy, rather than reading the DTDs slowly over the network.

If your application wants to validate against the MusicXML XSD rather than a DTD, you can use an entity resolver in your XML parser to do this. When writing MusicXML files, writing the DOCTYPE makes it easier for all applications - DTD or XSD based - to validate MusicXML files.

```
<score-partwise version="3.0">
```

This is the root document type. The `<score-partwise>` element is made up of parts, where each part is made up of measures. There is also a `<score-timewise>` option which is made up of measures, where each measure is made up of parts. The version attribute lets programs distinguish what version of MusicXML is being used more easily. Leave it out if you are writing MusicXML 1.0 files.

```
<part-list>
  <score-part id="P1">
    <part-name>Part 1</part-name>
  </score-part>
</part-list>
```

Whether you have a partwise or timewise score, a MusicXML file starts off with a header that lists the different musical parts in the score. The above example is the minimal part-list possible: it contains one score-part, the required id attribute for the score-part, and the required part-name element.

```
<part id="P1">
```

We are now beginning the first (and only, in this case) part within the document. The id attribute here must refer to an id attribute for a score-part in the header.

```
<measure number="1">
```

We are starting the first measure in the first part.

```
<attributes>
```

The attributes element contains key information needed to interpret the notes and musical data that follow in this part.

```
<divisions>1</divisions>
```



Each note in MusicXML has a duration element. The divisions element provided the unit of measure for the duration element in terms of divisions per quarter note. Since all we have in this file is one whole note, we never have to divide a quarter note, so we set the divisions value to 1.

Musical durations are typically expressed as fractions, such as "quarter" and "eighth" notes. MusicXML durations are fractions, too. Since the denominator rarely needs to change, it is represented separately in the divisions element, so that only the numerator needs to be associated with each individual note. This is similar to the scheme used in MIDI to represent note durations.

```
<key>
  <fifths>0</fifths>
</key>
```

The key element is used to represent a key signature. Here we are in the key of C major, with no flats or sharps, so the fifths element is 0. If we were in the key of D major with 2 sharps, fifths would be set to 2. If we were in the key of F major with 1 flat, fifths would be set to -1. The name "fifths" comes from the representation of a key signature along the circle of fifths. It lets us represent standard key signatures with one element, instead of separate elements for sharps and flats.

```
<time>
  <beats>4</beats>
  <beat-type>4</beat-type>
</time>
```

The time element represents a time signature. Its two component elements, beat and beat-type, are the numerator and denominator of the time signature, respectively.

```
<clef>
  <sign>G</sign>
  <line>2</line>
</clef>
```

MusicXML allows for many different clefs, including many no longer used today. Here, the standard treble clef is represented by a G clef on the second line of the staff (e.g., the second line from the bottom of the staff is a G).

```
</attributes>
<note>
```

We are done with the attributes, and are ready to begin the first note.

```
<pitch>
  <step>C</step>
  <octave>4</octave>
</pitch>
```

The pitch element must have a step and an octave element. Optionally it can have an alter element, if there is a flat or sharp involved. These elements represent sound, so the alter element must always be included if used, even if the alteration is in the key signature. In this case, we have no alteration. The pitch step is C. The octave of 4 indicates the octave the starts with middle C. Thus this note is a middle C.

```
<duration>4</duration>
```



Our divisions value is 1 division per quarter note, so the duration of 4 is the length of 4 quarter notes.

```
<type>whole</type>
```

The `type` element tells us that this is notated as a whole note. You could probably derive this from the duration in this case, but it is much easier to work with both notation and performance applications if the notation and performance data is represented separately.

In any event, the performance and notation data do not always match in practice. For example, if you want to better approximate a swing feel than the equal eighth notes notated in a jazz chart, you might use different duration values while the type remains an eighth note. Bach's music contains examples of shorthand notation where the actual note durations do not match the standard interpretation of the notes on the page, due to his use of a notational shorthand for certain rhythms.

The duration element should reflect the intended duration, not a longer or shorter duration specific to a certain performance. The note element has attack and release attributes that suggest ways to alter a note's start and stop times from the nominal duration indicated directly or indirectly by the score.

```
</note>
```

We are done with the note.

```
</measure>
```

We are done with the measure.

```
</part>
```

We are done with the part.

```
</score-partwise>
```

And we are done with the score.

One limitation of XML's document type definitions is that if you want to limit the number of elements within another element, you generally must also restrict how they are ordered as well. In the attributes, for instance, we want no more than one divisions element; for the note's pitch, we want one and only one step element and octave element. In order to do this, the order in which these elements appear must be constrained as well.

Thus the order in which elements appear in these examples does matter. The DTD definitions should make it clear what ordering is required; we will not spell that out in detail during the tutorial.

10.4 Object description: MPEG-4

Adapted from MPEG-4 specifications

10.4.1 Scope and features of the MPEG-4 standard

The MPEG-4 standard provides a set of technologies to satisfy the needs of authors, service providers and end users alike.

- For authors, MPEG-4 enables the production of content that has far greater reusability, has greater flexibility than is possible today with individual technologies such as digital television, animated graphics, World Wide Web (WWW) pages and their extensions. Also, it is now possible to better manage and protect content owner rights.



- For network service providers MPEG-4 offers transparent information, which can be interpreted and translated into the appropriate native signaling messages of each network with the help of relevant standards bodies.
- For end users, MPEG-4 brings higher levels of interaction with content, within the limits set by the author. It also brings multimedia to new networks, including those employing relatively low bitrate, and mobile ones.

For all parties involved, MPEG seeks to avoid a multitude of proprietary, non-interworking formats and players.

MPEG-4 achieves these goals by providing standardized ways to support:

Coding representing units of aural, visual or audiovisual content, called *media objects*. These media objects can be of natural or synthetic origin; this means they could be recorded with a camera or microphone, or generated with a computer;

Composition describing the composition of these objects to create compound media objects that form audiovisual scenes;

Multiplex multiplexing and synchronizing the data associated with media objects, so that they can be transported over network channels providing a QoS appropriate for the nature of the specific media objects;

Interaction interacting with the audiovisual scene generated at the receiver end or, via a back channel, at the transmitter's end.

The standard explores every possibility of the digital environment. Recorded images and sounds co-exist with their computer-generated counterparts; a new language for sound promises compact-disk quality at extremely low data rates; and the multimedia content could even adjust itself to suit the transmission rate and quality.

Possibly the greatest of the advances made by MPEG-4 is that viewers and listeners need no longer be passive. The height of “interactivity” in audiovisual systems today is the user’s ability merely to stop or start a video in progress. MPEG-4 is completely different: it allows the user to interact with objects within the scene, whether they derive from so-called real sources, such as moving video, or from synthetic sources, such as computer-aided design output or computer-generated cartoons. Authors of content can give users the power to modify scenes by deleting, adding, or repositioning objects, or to alter the behavior of the objects; for example, a click on a box could set it spinning.

10.4.2 The utility of objects

At the atomic level, to use a chemical analogy, the audio and video components of MPEG-4 are known as objects. These can exist independently, or multiple ones can be grouped together to form higher-level audiovisual bonds, to coin a phrase. The grouping is called composition, and the result is an MPEG-4 scene [Fig. 1]. The strength of this so-called object-oriented approach is that the audio and video can be easily manipulated.

Visual objects in a scene are described mathematically and given a position in a two- or three-dimensional space. Similarly, audio objects are placed in a sound space. When placed in 3-D space, the video or audio object need only be defined once; the viewer can change his vantage point, and the calculations to update the screen and sound are done locally, at the user’s terminal. This is a critical feature if the response is to be fast and the available bit-rate is limited, or when no return channel is available, as in broadcast situations.

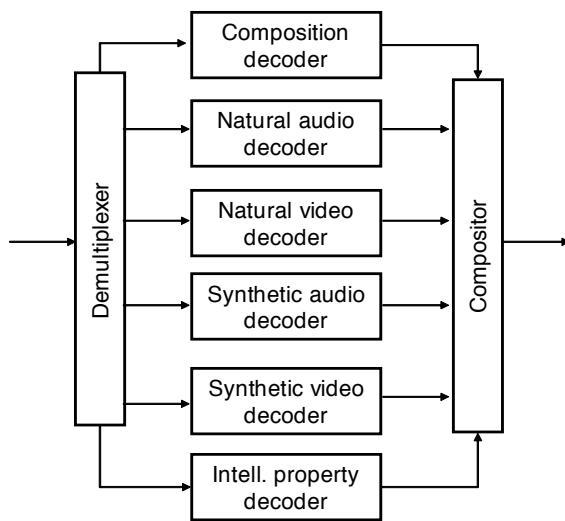


Figure 10.21: Mpeg-4 high level system architecture.

Figure 8.21 shows a high-level diagram of an MPEG-4 system's components. It serves as a reference for the terminology used in the system's design and specification: the demultiplexer, the elementary media decoders (natural audio, natural video, synthetic audio, and synthetic video), the specialized decoders for the composition information, and the specialized decoders for the protection information,

The following sections illustrate the MPEG-4 functionalities described above, using the audiovisual scene depicted in Figure 8.22.

10.4.3 Coded representation of media objects

MPEG-4 audiovisual scenes are composed of several media objects, organized in a hierarchical fashion. At the leaves of the hierarchy, we find primitive media objects, such as:

- Still images (e.g. as a fixed background);
- Video objects (e.g. a talking person - without the background);
- Audio objects (e.g. the voice associated with that person, background music).

MPEG-4 standardizes a number of such primitive media objects, capable of representing both natural and synthetic content types, which can be either 2- or 3-dimensional. In addition to the media objects mentioned above and shown in Figure 8.22, MPEG-4 defines the coded representation of objects such as:

- Text and graphics;
- Talking synthetic heads and associated text used to synthesize the speech and animate the head; animated bodies to go with the faces;
- Synthetic sound.

A media object in its coded form consists of descriptive elements that allow handling the object in an audiovisual scene as well as of associated streaming data, if needed. It is important to note that in its

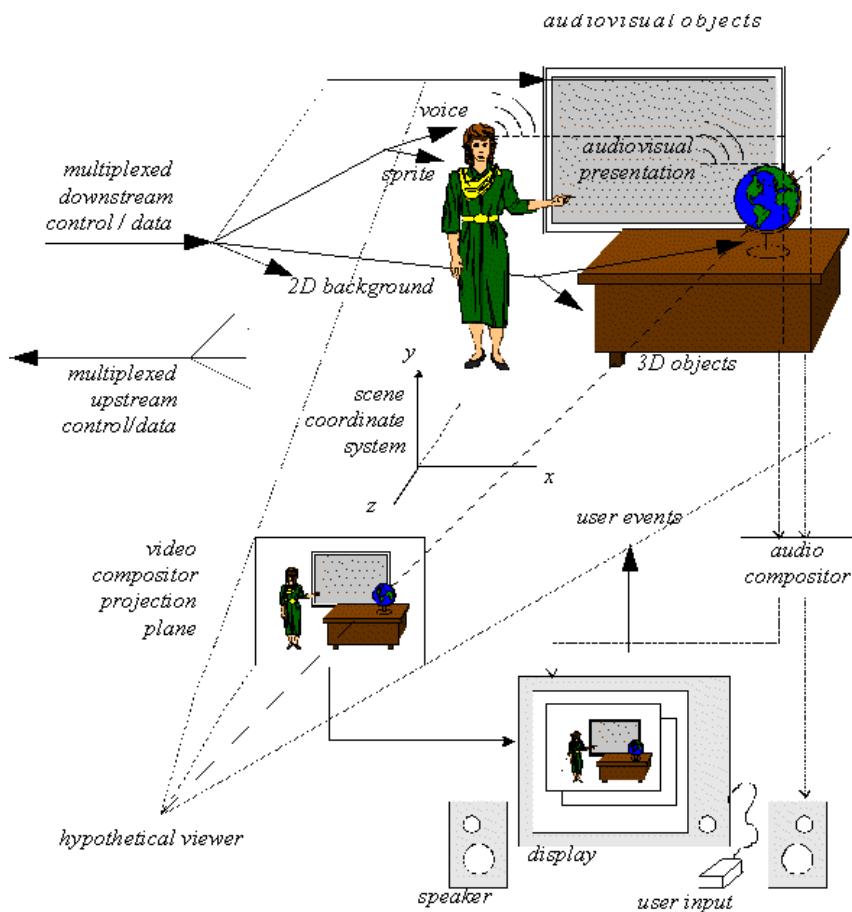


Figure 10.22: An example of an MPEG-4 scene.

coded form, each media object can be represented independent of its surroundings or background. The coded representation of media objects is as efficient as possible while taking into account the desired functionalities. Examples of such functionalities are error robustness, easy extraction and editing of an object, or having an object available in a scalable form.

10.4.3.1 Composition of media objects

The MPEG-4 standard deals with frames of audio and video (vectors of samples and matrices of pixels). Further, it deals with the objects that make up the audiovisual scene. Thus, a given scene has a number of video objects, of possibly differing shapes, plus a number of audio objects, possibly associated to video objects, to be combined before presentation to the user. Composition encompasses the task of combining all of the separate entities that make up the scene.

Figure 8.22 explains the way in which an audiovisual scene in MPEG-4 is described as composed of individual objects. The figure contains compound media objects that group primitive media objects together. Primitive media objects correspond to leaves in the descriptive tree while compound media objects encompass entire sub-trees. As an example: the visual object corresponding to the talking person and the corresponding voice are tied together to form a new compound media object, containing both the aural and visual components of that talking person.

Such grouping allows authors to construct complex scenes, and enables consumers to manipulate meaningful (sets of) objects.

More generally, MPEG-4 provides a standardized way to describe a scene, allowing for example to:

- Place media objects anywhere in a given coordinate system;
- Apply transforms to change the geometrical or acoustical appearance of a media object;
- Group primitive media objects in order to form compound media objects;
- Apply streamed data to media objects, in order to modify their attributes (e.g. a sound, a moving texture belonging to an object; animation parameters driving a synthetic face);
- Change, interactively, the user is viewing and listening points anywhere in the scene.

Composition information consists of the representation of the hierarchical structure of the scene. A graph describes the relationship among elementary media objects comprising the scene. The scene description builds on several concepts from the Virtual Reality Modeling language (VRML) in terms of both its structure and the functionality of object composition nodes and extends it to fully enable the aforementioned features.

The resulting specification addresses issues specific to an MPEG-4 system:

- description of objects representing natural audio and video with streams attached,
- description of objects representing synthetic audio and video (2D) and 3D material) with streams attached (such as streaming text or streaming parameters for animation of a facial model).

The scene description represents complex scenes populated by synthetic and natural audiovisual objects with their associated spatiotemporal transformations.

MPEG-4's language for describing and dynamically changing the scene is named the Binary Format for Scenes (BIFS). BIFS commands are available not only to add objects to or delete them from the scene, but also to change visual or acoustic properties of an object without changing the object in itself; thus the color alone of a 3-D sphere might be varied.

BIFS can be used to animate objects just by sending a BIFS command and to define their behavior in response to user input at the decoder. Again, this is a nice way to build interactive applications. In principle, BIFS could even be used to put an application screen (such as a Web browser's) as a "texture" in the scene.

BIFS borrows many concepts from the Virtual Reality Modeling Language (VRML), which is the method used most widely on the Internet to describe 3-D objects and users' interaction with them. BIFS and VRML can be seen as different representations of the same data. In VRML, the objects and their actions are described in text, as in any other high-level language. But BIFS code is binary, and thus is shorter for the same content—typically 10 to 15 times. More important, unlike VRML, MPEG-4 uses BIFS for real-time streaming, that is, a scene does not need to be downloaded in full before it can be played, but can be built up on the fly.

The author can generate this description in textual format, possibly through an authoring tool. The scene description then conforms to the VRML syntax with extensions. For efficiency, the standard defines a way to encode the scene description in a binary representation—Binary Format for Scene Description (BIFS). Multimedia scenes are conceived as hierarchical structures represented as a graph. Each leaf of the graph represents a media object (audio; video; synthetic audio like a Musical Instrument Digital Interface, or MIDI, stream; synthetic video like a face model). The graph structure isn't necessarily static, as the relationships can evolve over time as nodes or subgraphs are added or deleted. All the parameters describing these relationships are part of the scene description sent to the decoder.



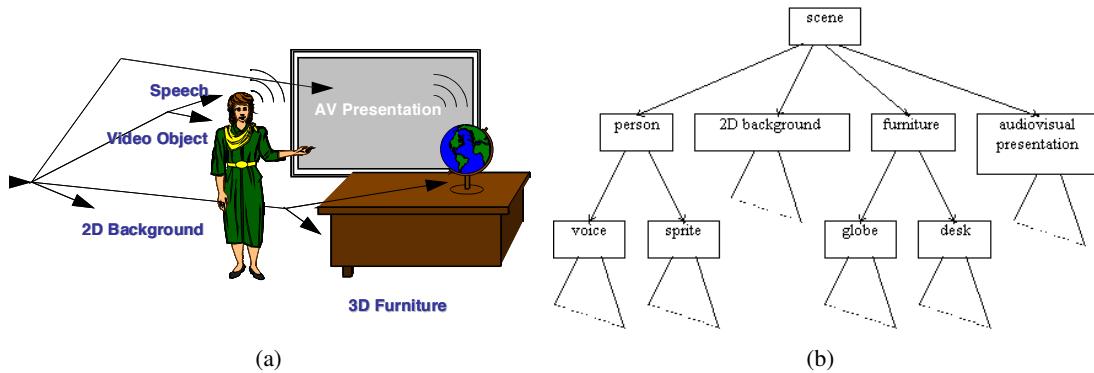


Figure 10.23: Objects in a scene (a) and the corresponding BIInary Format for Scene (BIFS) representation (b).

The initial snapshot of the scene is sent or retrieved on a dedicated stream. It is then parsed, and the whole scene structure is reconstructed (in an internal representation) at the receiver terminal. All the nodes and graph leaves that require streaming support to retrieve media contents or ancillary data (video stream, audio stream, facial animation parameters) are logically connected to the decoding pipelines.

An update of the scene structure may be sent at any time. These updates can access any field of any updatable node in the scene. An updatable node is one that received a unique node identifier in the scene structure. The user can also interact locally with the scenes, which may change the scene structure or the value of any field of any updatable node.

Composition information (information about the initial scene composition and the scene updates during the sequence evolution) is, like other streaming data, delivered in one elementary stream. The composition stream is treated differently from others because it provides the information required by the terminal to set up the scene structure and map all other elementary streams to the respective media objects.

How objects are grouped together An MPEG-4 scene follows a hierarchical structure, which can be represented as a directed acyclic graph. Each node of the graph is a media object, as illustrated in Figure 8.24(b) (note that this tree refers back to Figure 8.22 and Figure 8.24(a)). The tree structure is not necessarily static; node attributes (e.g., positioning parameters) can be changed while nodes can be added, replaced, or removed.

How objects are positioned in space and time: In the MPEG-4 model, audiovisual objects have both a spatial and a temporal extent. Each media object has a local coordinate system. A local coordinate system for an object is one in which the object has a fixed spatio-temporal location and scale. The local coordinate system serves as a handle for manipulating the media object in space and time. Media objects are positioned in a scene by specifying a coordinate transformation from the object local coordinate system into a global coordinate system defined by one more parent scene description nodes in the tree.

Wrapping the data Just as MPEG-4's representation of multimedia content is new and versatile, so is its scheme for preparing that content for transportation or storage (and, logically, for decoding) [Fig 2]. Here, objects are placed in so-called elementary streams (ESs). Some objects, such a sound track or a video, will have a single such stream. Others objects may have two or more. For instance, a scalable



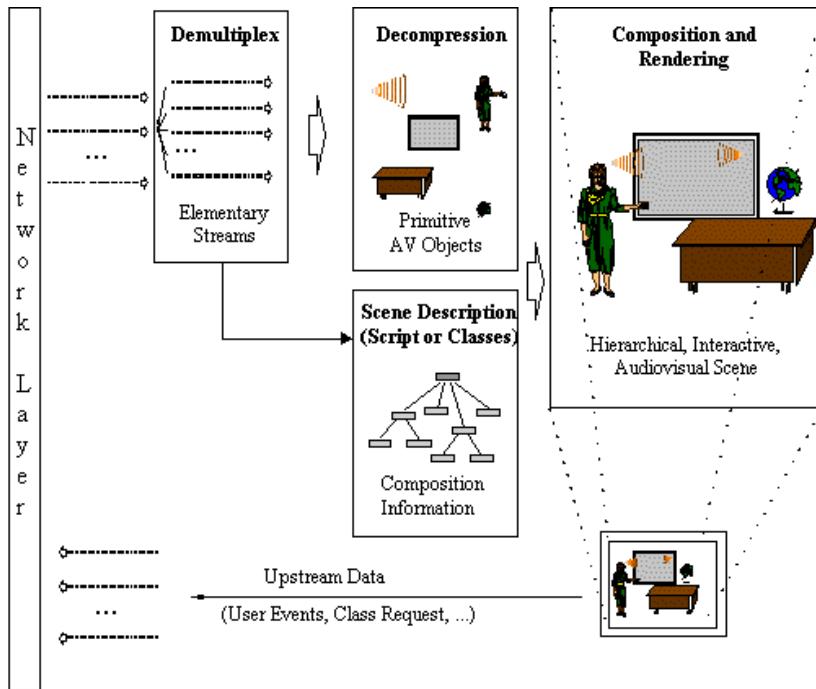


Figure 10.24: Major components of an MPEG-4 terminal (receiver side).

object would have an ES for basic-quality information plus one or more enhancement layers, each of which would have its own ES for improved quality, such as video with finer detail or faster motion.

Higher-level data describing the scene—the BIFS data defining, updating and positioning the media objects—is conveyed in its own ES. Here again the virtue of the hierarchical, object-based conceptions in MPEG-4 can be seen: it is easier to reuse objects in the production of new multimedia content, or (to say it another way) the new production is easier to modify without changing an encoded object itself. If parts of a scene are to be delivered only under certain conditions, say, when it is determined that enough bandwidth is available, multiple scene description ESs for the different circumstances may be used to describe the same scene.

To inform the system which elementary streams belong to a certain object, MPEG-4 uses the novel, critical concept of an object descriptor (OD). Object descriptors in their turn contain elementary stream descriptors (ESDs) to tell the system what decoders are needed to decode a stream. With another field, optional textual information about the object can be supplied. Object descriptors are sent in their own, special elementary stream, which allows them to be added or deleted dynamically as the scene changes.

The play-out of the multiple MPEG-4 objects is coordinated at a layer devoted solely to synchronization. Here, elementary streams are split into packets, and timing information is added to the payload of these packets. These packets are then ready to be passed on to the transport layer.

Streams Timing information for the decoder consists of the speed of the encoder clock and the time stamps of the incoming streams, which are relative to that clock. Two kinds of time stamps exist: one says when a piece of information must be decoded, the other says when the information must be ready for presentation.

The distinction between the types of stamp is important. In many video compression schemes, some frames are calculated as an interpolation between previous and following frames. Thus, before such a frame can be decoded and presented, the one after it must be decoded (and held in a buffer). For

predictable decoder behavior, a buffer model in the standard augments the timing specification.

In terms of the ISO seven-layer communications model, no specific transport mechanism is defined in MPEG-4. Existing transport formats and their multiplex formats suffice, including the MPEG-2 transport stream, asynchronous transfer mode (ATM), and real-time transport protocol (RTP) on the Internet. Incidentally, the fact that the MPEG-2 transport stream is used by digital TV has the important consequence of allowing co-broadcast modes.

A separate transport channel could be set up for each data stream, but there can be many of these for a single MPEG-4 scene, and as a result the process could be unwieldy and waste bits. To remedy matters, a small tool in MPEG-4, FlexMux, was designed to act as an intermediate step to any suitable form of transport. In addition, another interface defined in MPEG-4 lets the application ask for connections with a certain quality of service, in terms of parameters like bandwidth, error rate, or delay.

From the application's point of view, this interface is the same for broadcast channels, interactive sessions, and local storage media. Application designers can therefore write their code without having to worry about the underlying delivery mechanisms. Further, the next release of the standard will allow differing channels to be used at either end of a transmission/receive network, say, an Internet protocol channel on one end and an ATM one on the other.

Another important addition in Version 2 is a file format known as mp4, which can be used for exchange of content and which is easily converted. MPEG-1 and MPEG-2 did not include such a specification, but the intended use of MPEG-4 in Internet and personal computer environments makes it a necessity. It will be the only reliable way for users to exchange complete files of MPEG-4 content.

10.4.3.2 Description and synchronization of streaming data for media objects

Media objects may need streaming data, which is conveyed in one or more elementary streams. An object descriptor identifies all streams associated to one media object. This allows handling hierarchically encoded data as well as the association of meta-information about the content (called object content information) and the intellectual property rights associated with it.

Each stream itself is characterized by a set of descriptors for configuration information, e.g., to determine the required decoder resources and the precision of encoded timing information. Furthermore the descriptors may carry hints to the Quality of Service (QoS) it requests for transmission (e.g., maximum bit rate, bit error rate, priority, etc.)

Synchronization of elementary streams is achieved through time stamping of individual access units within elementary streams. The synchronization layer manages the identification of such access units and the time stamping. Independent of the media type, this layer allows identification of the type of access unit (e.g., video or audio frames, scene description commands) in elementary streams, recovery of the media object scene description time base, and it enables synchronization among them. The syntax of this layer is configurable in a large number of ways, allowing use in a broad spectrum of systems.

10.4.4 MPEG-4 visual objects

Classical, "rectangular" video, as the type that comes from a camera may be called, is of course one of the visual objects defined in the standard. In addition, objects with arbitrary shapes can be encoded apart from their background and then placed before other video types.

In fact, MPEG-4 includes two ways of describing arbitrary shapes, each appropriate to a different environment. In the first, known as binary shape, an encoded pixel (of a certain color, brightness, and so on) either is or is not part of the object in question. A simple but fairly crude technique, it is useful in low bit-rate environments, but can be annoying—the edges of pixels are sometimes visible, and curves have little jagged steps, known as aliasing or "the jaggies."



For higher-quality content, a shape description known as gray scale, or alpha shape, is used. Here, each pixel belonging to a shape is not merely on or off, but is assigned a value for its transparency. With this additional feature, transparency can differ from pixel to pixel of an object, and objects can be smoothly blended, either into a background or with other visual objects.

One instance of smooth blending can be seen in most television weather reports. The weatherman's image seems to be standing in front of a map, which in fact is generated elsewhere. Not surprisingly, then, manufacturers of television studio equipment have expressed an interest in the capabilities for arbitrary shape objects and scene description since, conceptually, they closely match the way things are already done in a studio. In fact, MPEG video has started working on bit-rates and quality levels well beyond the high-definition television (HDTV) level that can already be achieved.

Note that MPEG does not specify how shapes are to be extracted. Doing this automatically—video segmentation, as it is known—is still a matter of intensive research. Current methods still have limitations but there are ways to get the job done: the best way of obtaining shaped objects such as the weatherman is recording them with a blue or green background, colors that can easily be filtered out.

Actually, MPEG-4, like its predecessors, specifies only the decoding process. Encoding processes, including any improvements, are left to the marketplace. Even today, improvements in MPEG-2 compression quality sometimes show up, even though that standard was cast in stone several years ago.

On the other end of the bit-rate spectrum, a great deal of effort has gone into making moving video possible at very low bit-rates, notably for mobile devices. MPEG-4 has been found usable for streaming wireless video transmission (making use of GSM/Global System for Mobile Communications) at 10 kb/s—the data rate in GSM currently used for voice communications.

10.4.5 MPEG-4 audio

MPEG-4 coding of audio objects provides tools for both representing natural sounds (such as speech and music) and for synthesizing sounds based on structured descriptions. The representation for synthesized sound can be derived from text data or so-called instrument descriptions and by coding parameters to provide effects, such as reverberation and spatialization. The representations provide compression and other functionalities, such as scalability and effects processing.

To achieve the highest audio quality within the full range of hit rates and at the same time provide extra functionality's, the MPEG-4 Audio standard includes six types of coding techniques:

- Parametric coding modules
- Linear predictive coding (LPC) modules
- Time/frequency (T/F) coding modules
- Synthetic/natural hybrid coding (SNHC) integration modules
- Text-to-speech (TTS) integration modules
- Main integration modules, which combine the first three modules to a scalable encoder

While the first three parts describe real coding schemes for the low-bit-rate representation of natural audio sources, the SNHC and TTS parts only standardize the interfaces to general SNHC and TTS systems. Because of this, already established synthetic coding standards, such as MIDI, can be integrated into the MPEG-4 Audio system. The TTS interfaces permit plugging TTS modules optimized for a special language into the general framework. The main module contains global modules, such as the speed change functionality of MPEG-4 and the scalability add-ons necessary to implement large-step



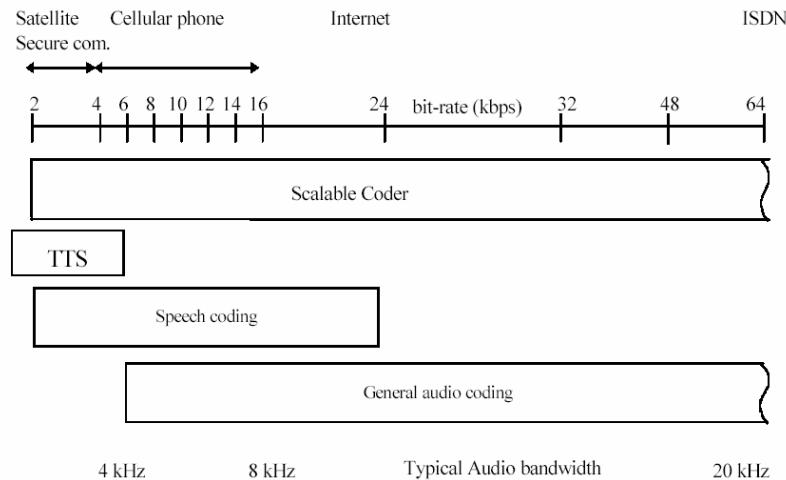


Figure 10.25: Major components of an MPEG-4 terminal (receiver side).

scalability by combining different coding schemes. To allow optimum coverage of the bitrates and to allow for bitrate and bandwidth scalability, a general framework has been defined. This is illustrated in Figure 8.25.

10.4.5.1 Natural audio

Starting with a coder operating at a low bitrate, by adding enhancements to a general audio coder, both the coding quality as well as the audio bandwidth can be improved.

Bitrate scalability, often also referred to as embedded coding, allows a bitstream to be parsed into a bitstream of lower bitrate that can still be decoded into a meaningful signal. The bitstream parsing can occur either during transmission or in the decoder. Bandwidth scalability is a particular case of bitrate scalability whereby part of a bitstream representing a part of the frequency spectrum can be discarded during transmission or decoding.

Encoder complexity scalability allows encoders of different complexity to generate valid and meaningful bitstreams. The decoder complexity scalability allows a given bitstream to be decoded by decoders of different levels of complexity. The audio quality, in general, is related to the complexity of the encoder and decoder used.

The MPEG-4 systems layer allows codecs according to existing (MPEG) standards, e.g. MPEG-2 AAC, to be used. Each of the MPEG-4 coders is designed to operate in a stand-alone mode with its own bitstream syntax. Additional functionalities are realized both within individual coders, and by means of additional tools around the coders. An example of such a functionality within an individual coder is speed or pitch change within HVXC.

10.4.5.2 Synthesized audio

MPEG-4 defines decoders for generating sound based on several kinds of ?structured? inputs. Text input is converted to speech in the Text-To-Speech (TTS) decoder, while more general sounds including music may be normatively synthesized. Synthetic music may be delivered at extremely low bitrates while still describing an exact sound signal.

Text To Speech. TTS coders bitrates range from 200 bit/s to 1.2 Kbit/s, which allows a text or a text with prosodic parameters (pitch contour, phoneme duration, and so on) as its inputs to generate intelligible speech.



ble synthetic speech. It supports the generation of parameters that can be used to allow synchronization to associated face animation, international languages for text and international symbols for phonemes. Additional markups are used to convey control information within texts, which is forwarded to other components in synchronization with the synthesized text. Note that MPEG-4 provides a standardized interface for the operation of a Text To Speech coder (TTSI = Text To Speech Interface), but not a normative TTS synthesizer itself.

An itemized overview:

- Speech synthesis using the prosody of the original speech
- Lip synchronization control with phoneme information.
- Trick mode functionality: pause, resume, jump forward/backward.
- International language and dialect support for text. (i.e. it can be signaled in the bitstream which language and dialect should be used)
- International symbol support for phonemes.
- support for specifying age, gender, speech rate of the speaker support for conveying facial animation parameter(FAP) bookmarks.

Score Driven Synthesis: Structured Audio The Structured Audio tools decode input data and produce output sounds. This decoding is driven by a special synthesis language called SAOL (Structured Audio Orchestra Language) standardized as a part of MPEG-4. This language is used to define an orchestra made up of instruments (downloaded in the bitstream, not fixed in the terminal) which create and process control data. An instrument is a small network of signal processing primitives that might emulate some specific sounds such as those of a natural acoustic instrument. The signal-processing network may be implemented in hardware or software and include both generation and processing of sounds and manipulation of pre-stored sounds.

MPEG-4 does not standardize a single method of synthesis, but rather a way to describe methods of synthesis. Any current or future sound-synthesis method can be described in SAOL, including wavetable, FM, additive, physical-modeling, and granular synthesis, as well as non-parametric hybrids of these methods.

Control of the synthesis is accomplished by downloading scores or scripts in the bitstream. A score is a time-sequenced set of commands that invokes various instruments at specific times to contribute their output to an overall music performance or generation of sound effects. The score description, downloaded in a language called SASL (Structured Audio Score Language), can be used to create new sounds, and also include additional control information for modifying existing sound. This allows the composer finer control over the final synthesized sound. For synthesis processes that do not require such fine control, the established MIDI protocol may also be used to control the orchestra.

Careful control in conjunction with customized instrument definition, allows the generation of sounds ranging from simple audio effects, such as footsteps or door closures, to the simulation of natural sounds such as rainfall or music played on conventional instruments to fully synthetic sounds for complex audio effects or futuristic music.

An important result of the description framework is that the synthetic audio fed to each terminal is identical. Thus, barring the vagaries of physical equipment that one user might have compared to another, the output is guaranteed to sound the same from terminal to terminal.

For terminals with less functionality, and for applications which do not require such sophisticated synthesis, a wavetable bank format is also standardized. Using this format, sound samples for use in



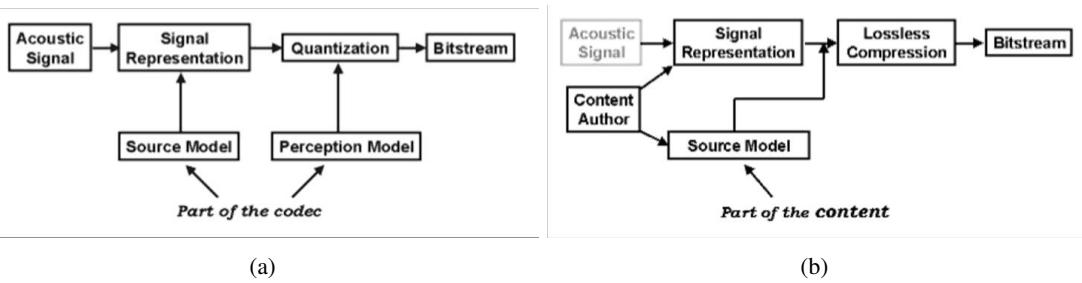


Figure 10.26: In a traditional audio coder, the source model and perception model are defined outside of the transmission (for example, in a standards document). The codec designers do the best job they can at designing these models, but then they are fixed for all content (a). In Structured Audio, the source model is part of the content. It is transmitted in the bitstream and used to give different semantics to the signal representation for each piece of content. There can be a different source model, or multiple source models, for each different piece of content (b).

wavetable synthesis may be downloaded, as well as simple processing, such as filters, reverbs, and chorus effects. In this case, the computational complexity of the required decoding process may be exactly determined from inspection of the bitstream, which is not possible when using SAOL.

Structured audio and traditional coding Structured audio coding differs from traditional audio coding in that the sound model is not fixed in the protocol (Fig. 8.26a), but dynamically described as part of the transmission stream, where it may vary from signal to signal (Fig. 8.26b). That is, where a traditional audio coder makes use of a fixed model such as a vocal-tract approximation (for LPC coding) or a psychoacoustic masking model (for wideband techniques such as MPEG-AAC or Dolby AC-3), a structured audio coder transmits sound in two parts: a description of a model and a set of parameters making use of that model.

The fact that we have great flexibility to encode different sound models in the bitstream means that, in theory, SA coding can subsume all other audio coding techniques. For example, if we wish to transmit speech in CELP-coded format, but only have the SA decoding system available, we can still use CELP: we write the CELP-decoder in SAOL, transmit it in the bitstream header, and then send frames of data optimized for that CELP model as the bitstream data. This bitstream will be nearly the same size as the CELP bitstream; it only requires a fixed constant-size data block to transmit the orchestra containing the decoder, and then the rest of the bitstream is the same size.

SAOL example SAOL is a “C-like” language. The syntactic framework of SAOL is familiar to anyone who programs in C, although the fundamental elements of the language are still signal variables, unit generators, instruments, and so forth, as in other synthesis languages. The program below shows a simple SAOL instrument that creates a simple beep by applying an envelope to the output of a single sinusoidal oscillator.

```
// This is a simple SAOL instrument that makes a short tone,
// using an oscillator over a stored function table.
```

```
instr tone(pitch,amp) {
    table wave(harm,2048,1); // sinusoidal wave function
    asig sound;           // 'asig' denotes audio signal
    ksig env;            // 'ksig' denotes control signal
```

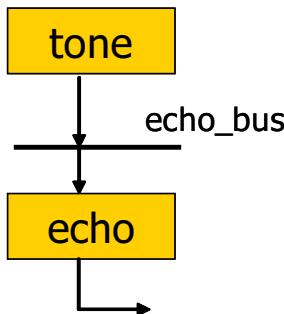


Figure 10.27: In SAOL, a metaphor of bus routing is employed that allows the concise description of complex networks. The output of the `tone` instrument is placed on the bus called `echo_bus`; this bus is sent to the instrument called `echo` for further processing.

```

env = kline(0,0.1,1,dur-0.1,0); //make envelope
sound = oscil(wave, pitch) * amp * env;
    // create sound by enveloping an oscillator
output(sound); // play that sound
}
  
```

A number of features are immediately apparent in this instrument. The instrument name (`tone`), parameters (or “p-fields”: `pitch` and `amp`), stored-function table (`wave`), and table generator (`harm`) all have names rather than numbers. All of the signal variables (`sound` and `env`) are explicitly declared with their rates (`asig` for audio rate and `ksig` for control rate), rather than being automatically assigned rates based on their names. There is a fully recursive expression grammar, so that unit generators like `kline` and `oscil` may be freely combined with arithmetic operators. The stored-function tables may be encapsulated in instruments or in the orchestra when this is desirable; they may also be provided in the score, in the manner of Music V (Csound also allows both options). The unit generators `kline` and `oscil` are built into the language; so is the wavetable generator `harm`.

The control signal `dur` is a standard name, which is a variable automatically declared in every instrument, with semantics given in the standard. There is a set of about 20 standard names defined in SAOL; `dur` always contains the duration of the note that invoked the instrument.

This SASL file plays a melody on `tone`:

```

0.5  tone 0.75 164.6
1.5  tone 0.75 329.6
2.5  tone 0.5   311.1
3    tone 0.25 246.9
3.25 tone 0.25 277.2
3.5  tone 0.5   311.1
4    tone 0.5   329.6
5    end
  
```

An aspect of modularity in SAOL involves its flow-of-control processing model. In SAOL, a metaphor of bus routing is employed that allows the concise description of complex networks. Its use is shown in the following example:

```

// This is a complete SAOL orchestra that demonstrates the
// use of buses and routing in order to do effects processing.
// The output of the 'tone' instrument is placed on the bus
  
```



```

// called 'echo_bus'; this bus is sent to the instrument called
// 'echo' for further processing.

global {
    srate 32000; krate 500;

    send(echo; 0.2; echo_bus);
    // use 'echo' to process the bus 'echo_bus'
    route(echo_bus, beep);
    // put the output of 'beep' on 'echo_bus'
}

instr tone(pitch, amp) {
    // as above
}

instr echo(dtime) {
    // a simple digital-delay echo. 'dtime' is the
    // cycle time.
    asig x;

    x = delay(x/2 + input[0], dtime);
    output(x);
}

```

In this orchestra, a global block is used to describe global parameters and control. The `srate` and `krate` tags specify the sampling rate and control (LFO) rate of the orchestra. The `send` instruction creates a new bus called `echo_bus`, and specifies that this bus is sent to the effects processing instrument called `echo`. The `route` instruction specifies that the samples produced by the instrument `beep` are not turned directly into sound output, but instead are “routed onto” the bus `echo_bus` for further processing (see Fig. 8.27).

The instrument `echo` implements a simple exponentially decaying digital-echo sound using the `delay` core opcode. The `dtime` p-field specifies the cycle time of the digital delay. Like `dur` in Figure 1, `input` is a standard name; `input` always contains the values of the input to the instrument, which in this case is the contents of the bus `echo_bus`. Note that `echo` is not a user-defined opcode that implements a new unit generator, but an effects-processing instrument.

This bus-routing model is modular with regard to the instruments `tone` and `echo`. The `tone` sound-generation instrument does not “know” that its sound will be modified, and the instrument itself does not have to be modified to enable this. Similarly, the `echo` instrument does not “know” that its input is coming from the `beep` instrument; it is easy to add other sounds to this bus without modification to `echo`. The bus-routing mechanism in SAOL allows easy reusability of effects-processing algorithms. There are also facilities that allow instruments to manipulate busses directly, if such modularity is not desirable in a particular composition.

10.4.5.3 Sound spatialization

Although less self-evident than with images, audio is also represented in the form of objects. An audio object can be a monaural speech channel or a multichannel, high-quality sound object. The composition process is in fact far more strictly prescribed for audio than for video. With the audio available as objects in the scene graph, different mixes from input channels (objects) to output channels (speakers) can be defined for different listening situations.

Another advantage of having audio as objects is that they then can have effects selectively applied to them. For example, if a soundtrack includes one object for speech and one for background audio, an artificial reverberation can be applied to the speech as distinct from the background music. If a user moves a video object in the scene, the audio can move along with it, and the user could also change how audio objects are mixed and combined.

Like video objects, audio objects may be given a location in a 3-D sound space, by instructing the terminal to spatially position sounds at certain spots. This is useful in an audio conference with many people, or in interactive applications where images as well as audio are manipulated.

A related feature known as environmental spatialization will be included in MPEG-4 Version 2. This feature can make how a sound object is heard depend on the room definition sent to the decoder, while the sound object itself need not be touched. In other words, the spatializations work locally, at the terminal, so again virtually no bit-transmission overhead is incurred.

Imagine spatialization when a person walks through a virtual house: when a new room of different shape and size is entered, the sound of the voice object changes accordingly without the object itself having to be changed.

10.4.5.4 Audio BIFS

The AudioBIFS system [part of the MPEG-4 Binary Format for Scene Description (BIFS)] allows multiple sounds to be transmitted using different coders and then mixed, equalized, and post-processed once they are decoded. This format is structurally based on the Virtual Reality Modeling Language (VRML) 2.0 syntax for scene description, but contains more powerful sound-description features.

Using MPEG-4 AudioBIFS, each part of a soundtrack may be coded in the format that best suits it. For example, suppose that the transmission of voice over with background music is desired in the style of a radio advertisement. It is difficult to describe high-quality spoken voice with sound-synthesis techniques, and so Structured Audio alone cannot be used; but speech coders are not adequate to code speech with background music, and so MPEG-4 CELP alone cannot be used. In MPEG-4 with AudioBIFS, the speech is coded using the CELP coder, and the background music is coded in Structured Audio. Then, at the decoding terminal, the two “streams” of sound are decoded individually and mixed together. The AudioBIFS part of the MPEG-4 standard describes the synchronization provided by this mixing process.

AudioBIFS is built from a set of nodes that link together into a tree structure, or scene graph. Each of these nodes represents a signal-processing manipulation on one or more audio streams. In this, AudioBIFS is somewhat itself like a sound-processing language, but it is much simpler (there are only seven types of nodes, and only “filters,” no “generators”). The scene-graph structure is used because it is a familiar and tested mechanism for computer-graphics description, and AudioBIFS is only a small part of the overall BIFS framework. The functions performed by AudioBIFS nodes allow sounds to be switched (as in a multiple-language soundtrack), mixed, delayed, “clipped” for interactive presentation, and gain-controlled.

More-advanced effects are possible by embedding SAOL code into the AudioBIFS scene graph with the AudioFX node. Using AudioFX, any audio effect may be described in SAOL and applied to the output of a natural or synthetic audio decoding process (see Figure 8.28).

For example, if we want to transmit speech with reverberated background music, we code the speech with MPEG-4 CELP. As above, we code the background music using MPEG-4 Structured Audio and provide an AudioFX node containing SAOL code that implements the desired reverberator. When the transmission is received, the synthetic music stream is decoded and the reverberation processing is performed; the result is added to the decoded speech stream. Only the resulting sound is played back to the listener. Just as MPEG-4 Structured Audio allows exact, terminal-independent control of sound synthesis for the composer, MPEG-4 AudioBIFS allows exact, terminal-independent control of audio-effects



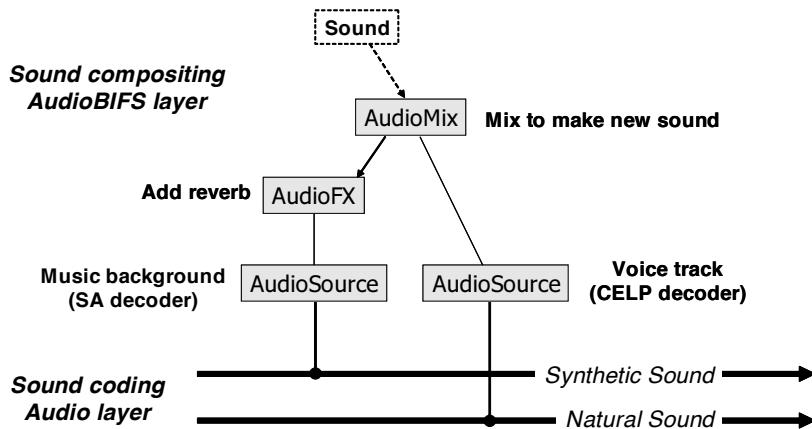


Figure 10.28: Two sound streams are processed and mixed using the AudioBIFS scene graph. A musical background is transmitted with the MPEG-4 Structured Audio system and reverb is added with an AudioFX node. Then the result is mixed with a speech sound transmitted with MPEG-4 CELP.

processing for the sound designer and producer.

The combination of synthetic and natural sound in the same sound scene with downloaded mixing and effects processing is termed synthetic/natural hybrid coding, or SNHC audio coding, in MPEG-4. Other features of AudioBIFS include 3-D audio spatialization for sound presentation in virtual-reality applications, and the creation of sound scenes that render differently on different terminals, depending (for example) on sampling rate, speaker configuration, or listening conditions.

Finally, the AudioBIFS component of MPEG-4 is part of the overall BIFS system, which provides sophisticated functionality for visual presentation of streaming video, 3-D graphics, virtual-reality scenes, and the handling of interactive events. The audio objects described with AudioBIFS and the various audio decoders may be synchronized with video or computer-graphics objects, and altered in response to user interaction.

10.5 Multimedia Content Description: Mpeg-7

Adapted from MPEG-7 specifications

10.5.1 Introduction

The MPEG-7 standard also known as “Multimedia Content Description Interface” aims at providing standardized core technologies allowing description of audiovisual data content in multimedia environments. It supports some degree of interpretation of the information’s meaning, which can be passed onto, or accessed by, a device or a computer code. MPEG-7 is not aimed at any one application in particular; rather, the elements that MPEG-7 standardizes shall support as broad a range of applications as possible.

Accessing audio and video used to be a simple matter - simple because of the simplicity of the access mechanisms and because of the poverty of the sources. An incommensurable amount of audiovisual information is becoming available in digital form, in digital archives, on the World Wide Web, in broadcast data streams and in personal and professional databases, and this amount is only growing. The value of information often depends on how easy it can be found, retrieved, accessed and filtered and managed.

The transition between two millennia abounds with new ways to produce, offer, filter, search, and



manage digitized multimedia information. Broadband is being offered with increasing audio and video quality and speed of access. The trend is clear. In the next few years, users will be confronted with such a large number of contents provided by multiple sources that efficient and accurate access to this almost infinite amount of content will seem to be unimaginable. In spite of the fact that users have increasing access to these resources, identifying and managing them efficiently is becoming more difficult, because of the sheer volume. This applies to professional as well as end users. The question of identifying and managing content is not just restricted to database retrieval applications such as digital libraries, but extends to areas like broadcast channel selection, multimedia editing, and multimedia directory services.

This challenging situation demands a timely solution to the problem. MPEG-7 is the answer to this need.

10.5.1.1 Context of MPEG-7

Audiovisual information plays an important role in our society, be it recorded in such media as film or magnetic tape or originating, in real time, from some audio or visual sensors and be it analogue or, increasingly, digital. Everyday, more and more audiovisual information is available from many sources around the world and represented in various forms (modalities) of media, such as still pictures, graphics, 3D models, audio, speech, video, and various formats. While audio and visual information used to be consumed directly by the human being, there is an increasing number of cases where the audiovisual information is created, exchanged, retrieved, and re-used by computational systems. This may be the case for such scenarios as image understanding (surveillance, intelligent vision, smart cameras, etc.) and media conversion (speech to text, picture to speech, speech to picture, etc.). Other scenarios are information retrieval (quickly and efficiently searching for various types of multimedia documents of interest to the user) and filtering in a stream of audiovisual content description (to receive only those multimedia data items which satisfy the user preferences). For example, a code in a television program triggers a suitably programmed PVR (Personal Video Recorder) to record that program, or an image sensor triggers an alarm when a certain visual event happens. Automatic transcoding may be performed from a string of characters to audible information or a search may be performed in a stream of audio or video data. In all these examples, the audiovisual information has been suitably “encoded” to enable a device or a computer code to take some action.

Audiovisual sources will play an increasingly pervasive role in our lives, and there will be a growing need to have these sources processed further. This makes it necessary to develop forms of audiovisual information representation that go beyond the simple waveform or sample-based, compression-based (such as MPEG-1 and MPEG-2) or even objects-based (such as MPEG-4) representations. Forms of representation that allow some degree of interpretation of the informations meaning are necessary. These forms can be passed onto, or accessed by, a device or a computer code. In the examples given above an image sensor may produce visual data not in the form of PCM samples (pixels values) but in the form of objects with associated physical measures and time information. These could then be stored and processed to verify if certain programmed conditions are met. A video recording device could receive descriptions of the audiovisual information associated to a program that would enable it to record, for example, only news with the exclusion of sport. Products from a company could be described in such a way that a machine could respond to unstructured queries from customers making inquiries.

MPEG-7 is a standard for describing the multimedia content data that support these operational requirements. The requirements apply, in principle, to both real-time and non real-time as well as push and pull applications. MPEG-7 does not standardize or evaluate applications, although in the development of the MPEG-7 standard applications have been used for understanding the requirements and evaluation of technology. It must be made clear that the requirements are derived from analyzing a wide range of potential applications that could use MPEG-7 tools. MPEG-7 is not aimed at any one application in



particular; rather, the elements that MPEG-7 standardizes support as broad a range of applications as possible.

10.5.1.2 MPEG-7 objectives

In October 1996, MPEG started a new work item to provide a solution to the questions described above. The new member of the MPEG family, named “Multimedia Content Description Interface” (in short MPEG-7), provides standardized core technologies allowing the description of audiovisual data content in multimedia environments. It extends the limited capabilities of proprietary solutions in identifying content that exist today, notably by including more data types.

Audiovisual data content that has MPEG-7 descriptions associated with it, may include: still pictures, graphics, 3D models, audio, speech, video, and composition information about how these elements are combined in a multimedia presentation (scenarios). A special case of these general data types is facial characteristics. MPEG-7 descriptions do, however, not depend on the ways the described content is coded or stored. It is possible to create an MPEG-7 description of an analogue movie or of a picture that is printed on paper, in the same way as of digitized content.

MPEG-7 allows different granularity in its descriptions, offering the possibility to have different levels of discrimination. Even though the MPEG-7 description does not depend on the (coded) representation of the material, MPEG-7 can exploit the advantages provided by MPEG-4 coded content. If the material is encoded using MPEG-4, which provides the means to encode audio-visual material as objects having certain relations in time (synchronization) and space (on the screen for video, or in the room for audio), it will be possible to attach descriptions to elements (objects) within the scene, such as audio and visual objects.

Because the descriptive features must be meaningful in the context of the application, they will be different for different user domains and different applications. This implies that the same material can be described using different types of features, tuned to the area of application. To take the example of visual material: a lower abstraction level would be a description of e.g. shape, size, texture, color, movement (trajectory) and position (where in the scene can the object be found); and for audio: key, mood, tempo, tempo changes, position in sound space. The highest level would give semantic information: “This is a scene with a barking brown dog on the left and a blue ball that falls down on the right, with the sound of passing cars in the background”. Intermediate levels of abstraction may also exist.

The level of abstraction is related to the way the features can be extracted: many low-level features can be extracted in fully automatic ways, whereas high level features need (much) more human interaction.

Next to having a description of what is depicted in the content, it is also required to include other types of information about the multimedia data:

The form - An example of the form is the coding format used (e.g. JPEG, MPEG-2), or the overall data size. This information helps determining whether the material can be read by the user terminal;

Conditions for accessing the material - This includes links to a registry with intellectual property rights information, and price;

Classification - This includes parental rating, and content classification into a number of pre-defined categories;

Links to other relevant material - The information may help the user speeding up the search;

The context - In the case of recorded non-fiction content, it is very important to know the occasion of the recording (e.g. Olympic Games 1996, final of 200 meter hurdles, men).



The main elements of the MPEG-7 standard are:

Descriptions Tools comprising

Descriptors (D), that define the syntax and the semantics of each feature (metadata element);

Description Schemes (DS), that specify the structure and semantics of the relationships between their components, that may be both Descriptors and Description Schemes;

A Description Definition Language (DDL) to define the syntax of the MPEG-7 Description Tools and to allow the creation of new Description Schemes and, possibly, Descriptors and to allow the extension and modification of existing Description Schemes;

System tools to support binary coded representation for efficient storage and transmission, transmission mechanisms (both for textual and binary formats), multiplexing of descriptions, synchronization of descriptions with content, management and protection of intellectual property in MPEG-7 descriptions, etc.

Therefore, MPEG-7 Description Tools allows to create descriptions (i.e., a set of instantiated Description Schemes and their corresponding Descriptors at the users will), to incorporate application specific extensions using the DDL and to deploy the descriptions using System tools

The MPEG-7 descriptions of content that may include:

- Information describing the creation and production processes of the content (director, title, short feature movie).
- Information related to the usage of the content (copyright pointers, usage history, broadcast schedule).
- Information of the storage features of the content (storage format, encoding).
- Structural information on spatial, temporal or spatio-temporal components of the content (scene cuts, segmentation in regions, region motion tracking).
- Information about low level features in the content (colors, textures, sound timbres, melody description).
- Conceptual information of the reality captured by the content (objects and events, interactions among objects).
- Information about how to browse the content in an efficient way (summaries, variations, spatial and frequency subbands,).
- Information about collections of objects.
- Information about the interaction of the user with the content (user preferences, usage history).

All these descriptions are of course coded in an efficient way for searching, filtering, etc.

To accommodate this variety of complementary content descriptions, MPEG-7 approaches the description of content from several viewpoints. The sets of Description Tools developed on those viewpoints are presented here as separate entities. However, they are interrelated and can be combined in many ways. Depending on the application, some will present and others can be absent or only partly present.



A description generated using MPEG-7 Description Tools will be associated with the content itself, to allow fast and efficient searching for, and filtering of material that is of interest to the user.

MPEG-7 data may be physically located with the associated AV material, in the same data stream or on the same storage system, but the descriptions could also live somewhere else on the globe. When the content and its descriptions are not co-located, mechanisms that link the multimedia material and their MPEG-7 descriptions are needed; these links will have to work in both directions.

MPEG-7 addresses many different applications in many different environments, which means that it needs to provide a flexible and extensible framework for describing audiovisual data. Therefore, MPEG-7 does not define a monolithic system for content description but rather a set of methods and tools for the different viewpoints of the description of audiovisual content.

10.5.2 MPEG-7 terminology

This section presents the terminology used by MPEG-7. This terminology plays a major role in the understanding of the MPEG-7 process.

Data. Data is audio-visual information that will be described using MPEG-7, regardless of storage, coding, display, transmission, medium, or technology. This definition is intended to be sufficiently broad to encompass graphics, still images, video, film, music, speech, sounds, text and any other relevant AV medium. Examples for MPEG-7 data are an MPEG-4 stream, a video tape, a CD containing music, sound or speech, a picture printed on paper, and an interactive multimedia installation on the web.

Feature. A Feature is a distinctive characteristic of the data which signifies something to somebody. Features themselves cannot be compared without a meaningful feature representation (descriptor) and its instantiation (descriptor value) for a given data set. Some examples are: color of an image, pitch of a speech segment, rhythm of an audio segment, camera motion in a video, style of a video, the title of a movie, the actors in a movie, etc.

Descriptor. A Descriptor (D) is a representation of a Feature. A Descriptor defines the syntax and the semantics of the Feature representation. A Descriptor allows an evaluation of the corresponding feature via the descriptor value. It is possible to have several descriptors representing a single feature, i.e. to address different relevant requirements. Possible descriptors are: the color histogram, the average of the frequency components, the motion field, the text of the title, etc.

Descriptor Value. A Descriptor Value is an instantiation of a Descriptor for a given data set (or subset thereof).

Description Scheme. A Description Scheme (DS) specifies the structure and semantics of the relationships between its components, which may be both Descriptors and Description Schemes. A Description Scheme corresponds to an entity or relationship at the level of the MPEG-7 Audio-visual Conceptual Model. A Description Scheme shall have descriptive information and may participate in many-to-one relationships with other description elements. Examples: A movie, temporally structured as scenes and shots, including some textual descriptors at the scene level, and color, motion and some audio descriptors at the shot level. The distinction between a Description scheme and a Descriptor is that a Descriptor is concerned with the representation of a Feature, whereas the Description Scheme deals with the structure of a Description.

Description. A Description consists of a DS (structure) and the set of Descriptor Values (instantiations) that describe the Data. Depending on the completeness of the set of Descriptor Values, the DS may be fully or partially instantiated.



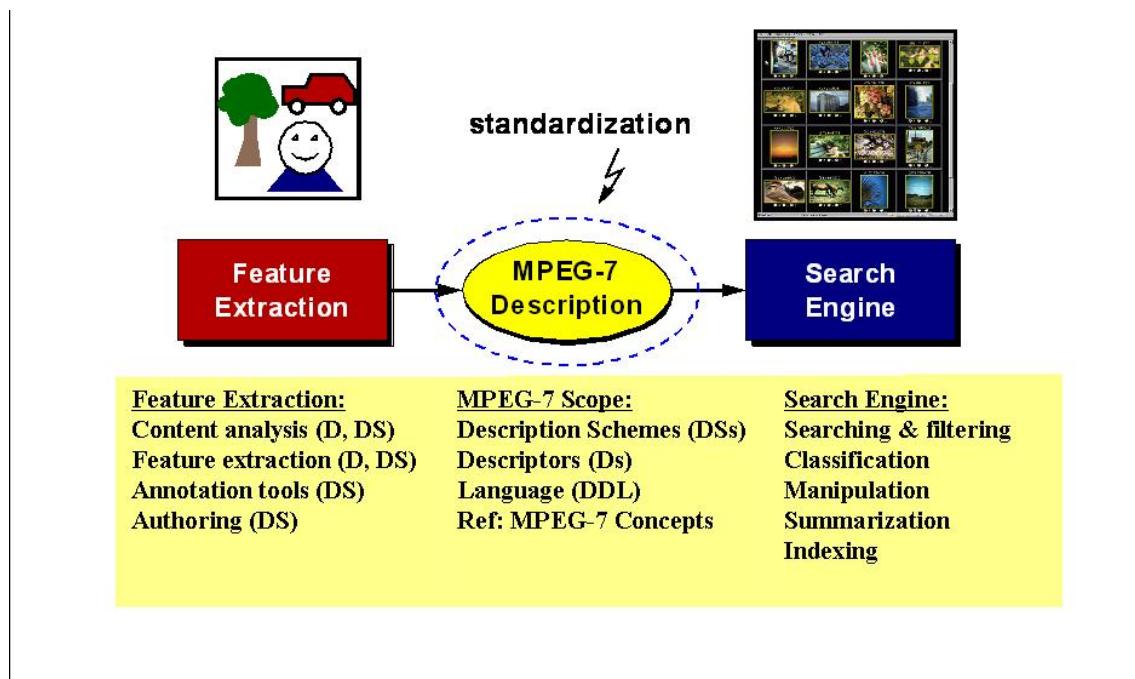


Figure 10.29: Scope of MPEG-7.

Coded Description. A Coded Description is a Description that has been encoded to fulfil relevant requirements such as compression efficiency, error resilience, random access, etc.

10.5.3 Scope of the Standard

MPEG-7 addresses applications that can be stored (on-line or off-line) or streamed (e.g. broadcast, push models on the Internet), and can operate in both real-time and non real-time environments. A real-time environment in this context means that the description is generated while the content is being captured

Figure 8.29 shows a highly abstract block diagram of a possible MPEG 7 processing chain, included here to explain the scope of the MPEG-7 standard. This chain includes feature extraction (analysis), the description itself, and the search engine (application). To fully exploit the possibilities of MPEG-7 descriptions, automatic extraction of features (or descriptors) are extremely useful. It is also clear that automatic extraction is not always possible, however. As was noted above, the higher the level of abstraction, the more difficult automatic extraction is, and interactive extraction tools will be of good use. However useful they are, neither automatic nor semi-automatic feature extraction algorithms is inside the scope of the standard. The main reason is that their standardisation is not required to allow interoperability, while leaving space for industry competition. Another reason not to standardise analysis is to allow making good use of the expected improvements in these technical areas.

Also the search engines, filter agents, or any other program that can make use of the description, are not be specified within the scope of MPEG-7; again this is not necessary, and here too, competition will produce the best results.

Figure 8.30 shows the relationship among the different MPEG-7 elements introduced above. The DDL allows the definition of the MPEG-7 description tools, both Descriptors and Description Schemes, providing the means for structuring the Ds into DSs. The DDL also allows the extension for specific applications of particular DSs. The description tools are instantiated as descriptions in textual format (XML) thanks to the DDL (based on XML Schema). Binary format of descriptions is obtained by means

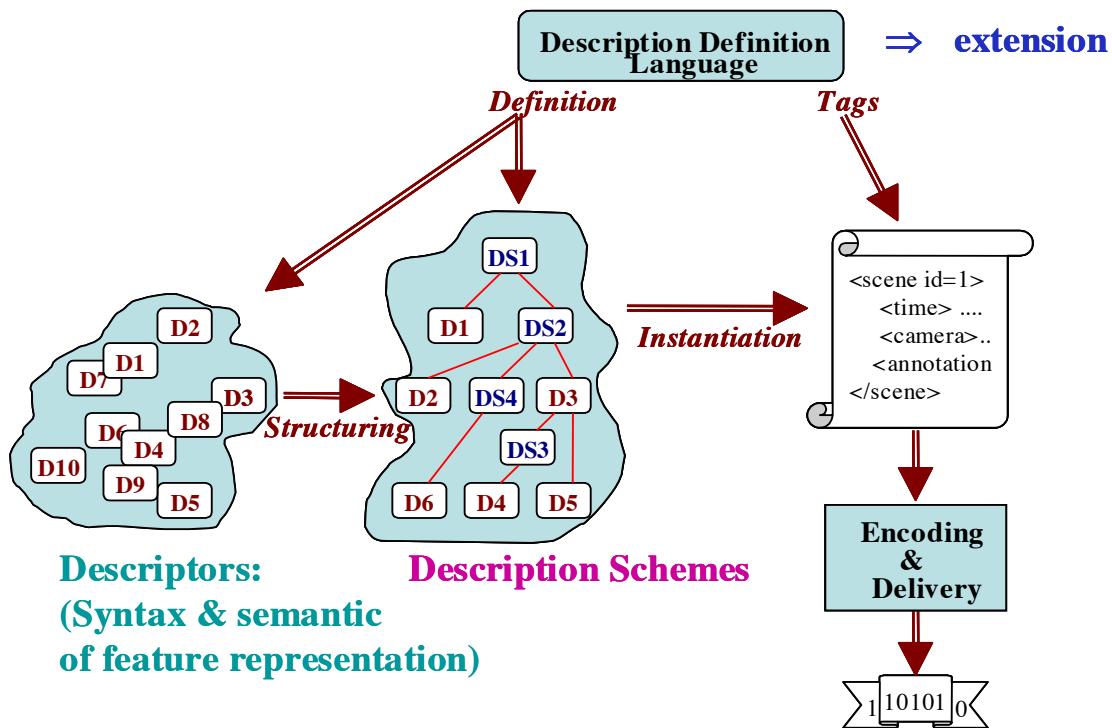


Figure 10.30: MPEG-7 main elements.

of the BiM defined in the Systems part.

Figure 8.31 explains a hypothetical MPEG-7 chain in practice [There can be other streams from content to user; these are not depicted here. Furthermore, it is understood that the MPEG-7 Coded Description may be textual or binary, as there might be cases where a binary efficient representation of the description is not needed, and a textual representation would suffice.] . From the multimedia content an Audiovisual description is obtained via manual or semi-automatic extraction. The AV description may be stored (as depicted in the figure) or streamed directly. If we consider a pull scenario, client applications will submit queries to the descriptions repository and will receive a set of descriptions matching the query for browsing (just for inspecting the description, for manipulating it, for retrieving the described content, etc.). In a push scenario a filter (e.g., an intelligent agent) will select descriptions from the available ones and perform the programmed actions afterwards (e.g., switching a broadcast channel or recording the described stream). In both scenarios, all the modules may handle descriptions coded in MPEG-7 formats (either textual or binary), but only at the indicated conformance points it is required to be MPEG-7 conformant (as they show the interfaces between an application acting as information server and information consumer). The emphasis of MPEG-7 is the provision of novel solutions for audio-visual content description. Thus, addressing text-only documents was not among the goals of MPEG-7. However, audio-visual content may include or refer to text in addition to its audio-visual information. MPEG-7 therefore has standardized different Description Tools for textual annotation and controlled vocabularies, taking into account existing standards and practices.

To provide a better understanding of the terminology introduced above (i.e. Descriptor, Description Scheme, and DDL), please refer to Figure 8.32 and Figure 8.32.

Figure 8.32 shows possible relation between Descriptors and Description Schemes. The arrows from DDL to Description Schemes are generated using the DDL. Furthermore, it indicates that the DDL



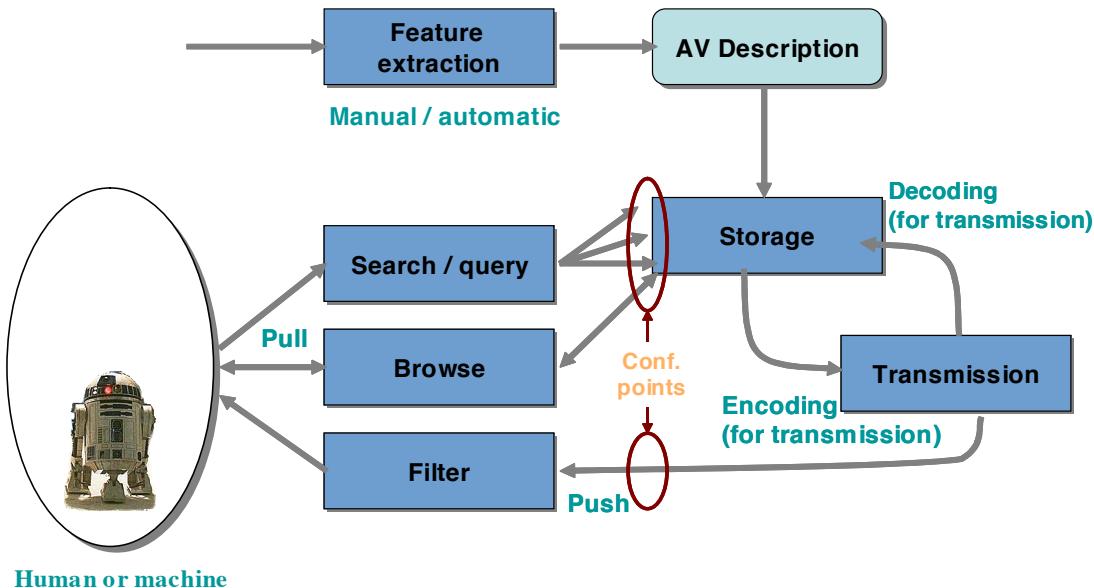


Figure 10.31: Abstract representation of possible applications using MPEG-7.

provides the mechanism to build a Description Scheme which in turn forms the basis for the generation of a Description (see also Figure 8.33).

Figure 8.33 explains a hypothetical MPEG-7 chain in practice. The circular boxes depict tools that are doing things, such as encoding or decoding, whereas the square boxes represent static elements, such as a description. The dotted boxes in the figure encompass the normative elements of the MPEG-7 standard.

The emphasis of MPEG-7 is the provision of novel solutions for audio-visual content description. Thus, addressing text-only documents is be among the goals of MPEG-7. However, audio-visual content may include or refer to text in addition to its audio-visual information. MPEG-7 therefore considers existing solutions developed by other standardisation organisations for text only documents and support them as appropriate.

Besides the descriptors themselves, the database structure plays a crucial role in the final retrievals performance. To allow the desired fast judgement about whether the material is of interest, the indexing information will have to be structured, e.g. in a hierarchical or associative way.

10.5.4 MPEG-7 Applications Areas

The elements that MPEG-7 standardizes supports a broad range of applications (for example, multimedia digital libraries, broadcast media selection, multimedia editing, home entertainment devices, etc.). MPEG-7 will also make the web as searchable for multimedia content as it is searchable for text today. This would apply especially to large content archives, which are being made accessible to the public, as well as to multimedia catalogues enabling people to identify content for purchase. The information used for content retrieval may also be used by agents, for the selection and filtering of broadcasted “push” material or for personalized advertising. Additionally, MPEG-7 descriptions allows fast and cost-effective usage of the underlying data, by enabling semi-automatic multimedia presentation and editing.

All applications domains making use of multimedia benefits from MPEG-7. Considering that at

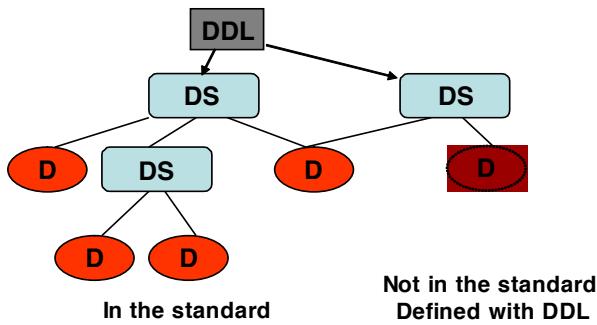


Figure 10.32: Abstract representation of possible relation between Descriptors and Description Schemes.

present day it is hard to find one not using multimedia, please extend the list of the examples below using your imagination:

- Architecture, real estate, and interior design (e.g., searching for ideas)
- Broadcast media selection (e.g., radio channel, TV channel)
- Cultural services (history museums, art galleries, etc.)
- Digital libraries (e.g., image catalogue, musical dictionary, bio-medical imaging catalogues, film, video and radio archives)
- E-Commerce (e.g., personalised advertising, on-line catalogues, directories of e-shops)
- Education (e.g., repositories of multimedia courses, multimedia search for support material)
- Home Entertainment (e.g., systems for the management of personal multimedia collections, including manipulation of content, e.g. home video editing, searching a game, karaoke)
- Investigation services (e.g., human characteristics recognition, forensics)
- Journalism (e.g. searching speeches of a certain politician using his name, his voice or his face)
- Multimedia directory services (e.g. yellow pages, Tourist information, Geographical information systems)
- Multimedia editing (e.g., personalised electronic news service, media authoring)
- Remote sensing (e.g., cartography, ecology, natural resources management)
- Shopping (e.g., searching for clothes that you like)
- Social (e.g. dating services)
- Surveillance (e.g., traffic control, surface transportation, non-destructive testing in hostile environments)

The way MPEG-7 data is used to answer user queries is outside the scope of the standard. In principle, any type of AV material may be retrieved by means of any type of query material. This means, for example, that video material may be queried using video, music, speech, etc. It is to the search engine to match the query data and the MPEG-7 AV description. A few query examples are:

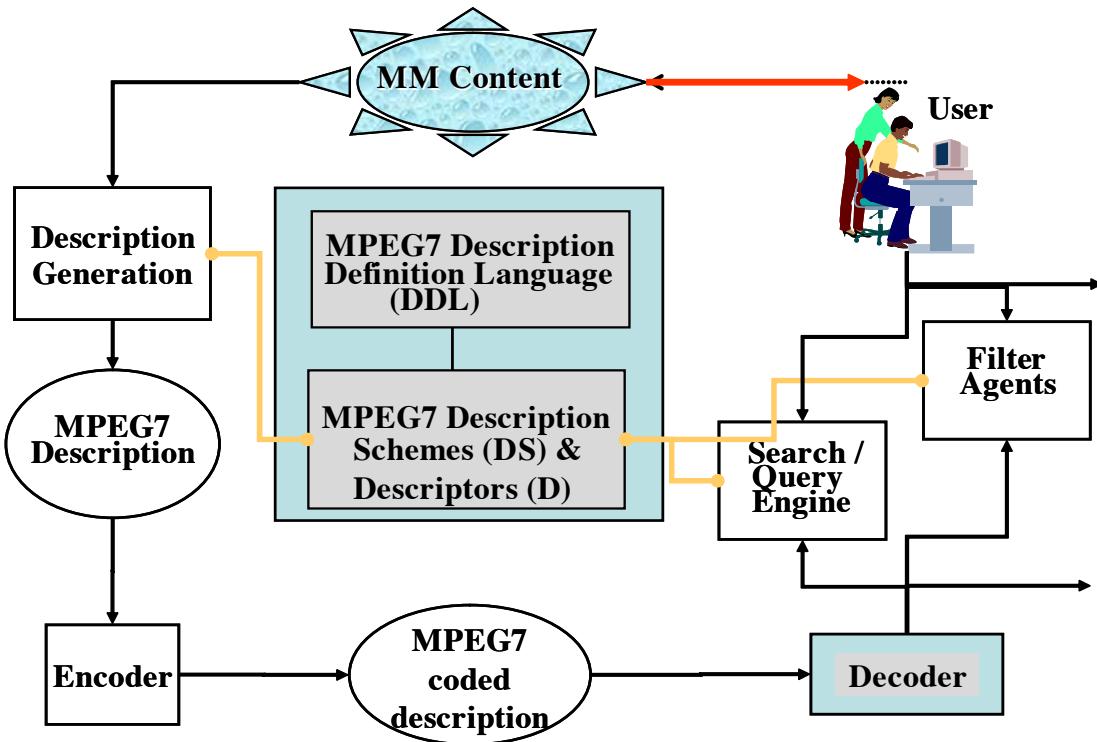


Figure 10.33: Abstract representation of possible applications using MPEG-7.

- Play a few notes on a keyboard and retrieve a list of musical pieces similar to the required tune, or images matching the notes in a certain way, e.g. in terms of emotions.
- Draw a few lines on a screen and find a set of images containing similar graphics, logos, ideograms,...
- Define objects, including colour patches or textures and retrieve examples among which you select the interesting objects to compose your design.
- On a given set of multimedia objects, describe movements and relations between objects and so search for animations fulfilling the described temporal and spatial relations.
- Describe actions and get a list of scenarios containing such actions.
- Using an excerpt of Pavarotti's voice, obtaining a list of Pavarotti's records, video clips where Pavarotti is singing and photographic material portraying Pavarotti.

10.5.4.1 Making audio-visual material as searchable as text

MPEG-7 began as a scheme for making audiovisual material as searchable as text is today. Indeed, it is conceivable that the structure and discipline to even minimally describe multimedia may exceed the current state of textual information retrieval. Although the proposed multimedia content descriptions now serve as much more than search applications, they remain the primary applications for MPEG-7. These retrieval applications involve databases, audio-visual archives, and the Web-based Internet paradigm (a client requests material from a server). TV and film archives represent a typical application in this domain. They store vast amounts of multimedia material in several different formats (digital or

analog tapes, film, CD-ROM, and so on) along with precise descriptive information (metadata) that may or may not be precisely timecoded. This metadata is stored in databases with proprietary formats. An enormous potential interest exists in an international standard format for the storage and exchange of descriptions that could ensure

- interoperability between video archive operators,
- perennial relevance of the metadata, and
- a wider diffusion of the data to the professional and general public.

To support these goals, MPEG-7 must accommodate visual and other searches of such existing multi-media databases. In addition, a vast amount of the older, analog audio-visual material will be digitized in years to come. This creates a tremendous opportunity to include content-based indexing features (extractable during the digitization/compression process⁵) into those existing databases.

For new audio-visual material, the ability to associate descriptive information within video streams at various stages of video production can dramatically improve the quality and productivity of manual, controlled vocabulary annotation of video data in a video archive. For example, preproduction and postproduction scripts, information captured or annotated during shooting, and postproduction edit lists would be very useful in the retrieval and reuse of archival material.

MPEG-7 specific requirements for such applications include

- Full-text descriptions as well as structured fields (database descriptions).
- A mechanism by which different MPEG-7 descriptions can support the ability to interoperate between different content-description semantics (such as different database schemas, different thesauri, and so on).
- A robust linking mechanism that allows referencing audio-visual objects or object instances and time references (including descriptions with incomplete or missing time references) even in an analog format.
- A structure to handle multiple versions of the same document at several stages in the production process and descriptions that apply to multiple copies of the same material.

For audio databases we face a similar situation. The consumer music industry is currently struggling with how to reach consumers with increasingly fragmented tastes. Music, as with all broadcast media artifacts, is undergoing the same Internet-flavored transformation as cable TV. An ideal way of presenting consumers with available music is to let them search effortlessly for what they want. Searchers may hum approximate renditions of the song they seek from a kiosk or from the comfort of their own home.⁷ Alternately, they may seek out music with features (musicians, style, tempo, year of creation) similar to those they already know. From there, they can listen to an appropriate sample (and perhaps view associated information such as lyrics or a video) and buy the music on the spot. The requirements for such types of audio-oriented applications on MPEG-7 include

- A mechanism that supports melody and other musical features that allow for reasonable errors by the indexer to accommodate queryby- humming.
- A mechanism that supports descriptors based on information associated with the data (such as textual data).
- Support description schemes that contain descriptors of visual, audio, and/or other features, and support links between the different media (cross-modal).



Other interesting applications related to audio include sound effects libraries, historical speech databases, and movie scene retrieval by memorable auditory events.

10.5.4.2 Supporting push and pull information acquisition methods

Filtering is essentially the converse of search. Search involves the “pull” of information, while filtering implies information “push.” Search requests the inclusion of information, while filtering excludes data. Both pursuits benefit strongly from the same sort of meta-information. Typical domains for such applications include broadcasting and the emerging Webcasting. These domains have very distinct requirements, generally dealing with streamed descriptions rather than static descriptions stored on databases.

User-agent-driven media selection and filtering in a broadcasting environment has particular interest for MPEG-7. This approach lets users select information more appropriate to their uses and desires from a broadcast stream of 500 channels, using the same meta-information as that used in search. Moreover, this application gives rise to several subtypes, primarily divided among types of users. A consumer-oriented selection leads to personalized audio-visual programs, for example. This can go much farther than typical video-on-demand in collecting personally relevant news programs, for example. A content-producer-oriented selection made on the segment or shot level is a way of collecting raw material from archives. The requirements for such types of applications on MPEG-7 include

- Support for descriptors and description schemes that allow multiple languages.
- A mechanism by which a media object may be represented by a set of concepts that may depend on locality or language.
- Support efficient interactive response times.

However, new ways of automating and streamlining the presentation of that data also requires selecting and filtering. A system that combines knowledge about the context, user, application, and design principles with knowledge about the information to be displayed can accomplish this.⁸ Through clever application of that knowledge, you can have an intelligent multimedia presentation system. For MPEG, this requires a mechanism by which to

- encode contextual information and
- represent temporal relationships.

Finally, selecting and filtering facilitates accessibility to information for all users, especially those who suffer from one or several disabilities such as visual, auditory, motor, or cognitive disabilities. Providing active information representations might help overcome such problems. The key issue is to allow multimodal communication to present information optimized for individual users’ abilities. Consider, for example, a search agent that does not exclude images as an information resource for the blind, but rather makes the MPEG-7 meta-data available. Aided by that metadata, sonification (auditory display) or haptic display becomes possible. Similarity of metadata helps provide a set of information in different modalities, in case the user can’t access the particular information. Thus, MPEG-7 must support descriptions that contain descriptors of visual, audio, and/or other features.

10.5.4.3 Enabling nontraditional control of information

The following potential MPEG-7 applications don’t limit themselves to traditional, media-oriented, multimedia content, but are functional within the metacontent representation in development under MPEG-7. Interestingly, they are neither push nor pull, but reject a certain amount of control over information



through metadata. These applications reach into such diverse, but data-intensive domains as medicine and remote sensing. Such applications can only increase the usefulness and reach of this proposed international standard.

One of the specific applications is semi-automated video editing. Assuming that sufficient information exists about the content and structure of a multimedia object (see the previous section), a smart multimedia clip could start to edit itself in a manner appropriate to its neighboring multimedia. For example, a piece of music and a video clip from different sources could combine in such a way that the music stretches and contracts to synchronize with specific hit points in the video, creating an appropriate customized soundtrack.

This could be a new paradigm for multimedia, adding a method layer on top of MPEG-7 representation layer. (We by no means suggest that such methods for interaction be standardized in MPEG-7. As with many other advanced capabilities building on the standard, it is an issue for implementers to address.) Making multimedia aware to an extent opens access to novice users and increases productivity for experts. Such hidden intelligence on the part of the data itself shifts multimedia editing from direct manipulation to loose management of data.

Semi-automated multimedia editing encompasses a broad category of applications. It can aid home users as well as experts in studios through varying amounts of guidance or assistance through the process. In its simpler version, assisted editing can consist of an MPEG-7-enabled browser for selecting video shots, using a suitable shot description language. In an intermediate version, assisted editing could include planning, proposing shot selections and edit points, thereby satisfying a scenario expressed in a sequence description language.

The education domain relates closely to semi-automated editing. The challenge of using multimedia in educational software lies in exploiting the intrinsic information as much as possible to support different pedagogical approaches such as summarization, question answering, or detection of and reaction to misunderstanding or nonunderstanding. By providing direct access to short video sequences within a large database, MPEG-7 can promote the use of audio, video, and film archive material in higher education in many areas, including history, performing art, film music.

10.5.5 Mpeg-7 description tools

In this section, more details on the MPEG-7 description tools, which comprise all of MPEG-7 predefined descriptors and description schemes, will be presented. We can also define additional description tools for specific applications using the MPEG-7 Description Definition Language (DDL), an extension of the XML Schema.

We can group these description tools in different classes according to their functionality (see Figure 8.34). These description tools are standardized by three parts: Visual for descriptors related to visual features that apply to images and/or videos; Audio for the description tools related to audio features, covering areas from speech to music; and Multimedia Description Schemes for description tools related to features applying to audio, visual, and audio-visual content.

Figure 8.34 provides an overview of the organization of MPEG-7 Multimedia DSs into the following areas: Basic Elements, Content Description, Content Management, Content Description, Content Organization, Navigation and Access, and User Interaction.

Basic elements MPEG-7 provides a number of Schema Tools that assist in the formation, packaging, and annotation of MPEG-7 descriptions. A number of basic elements are used throughout the MDS specification as fundamental constructs in defining the MPEG-7 DSs. The basic data types provide a set of extended data types and mathematical structures such as vectors and matrices, which are needed by the DSs for describing AV content. The basic elements include also constructs for



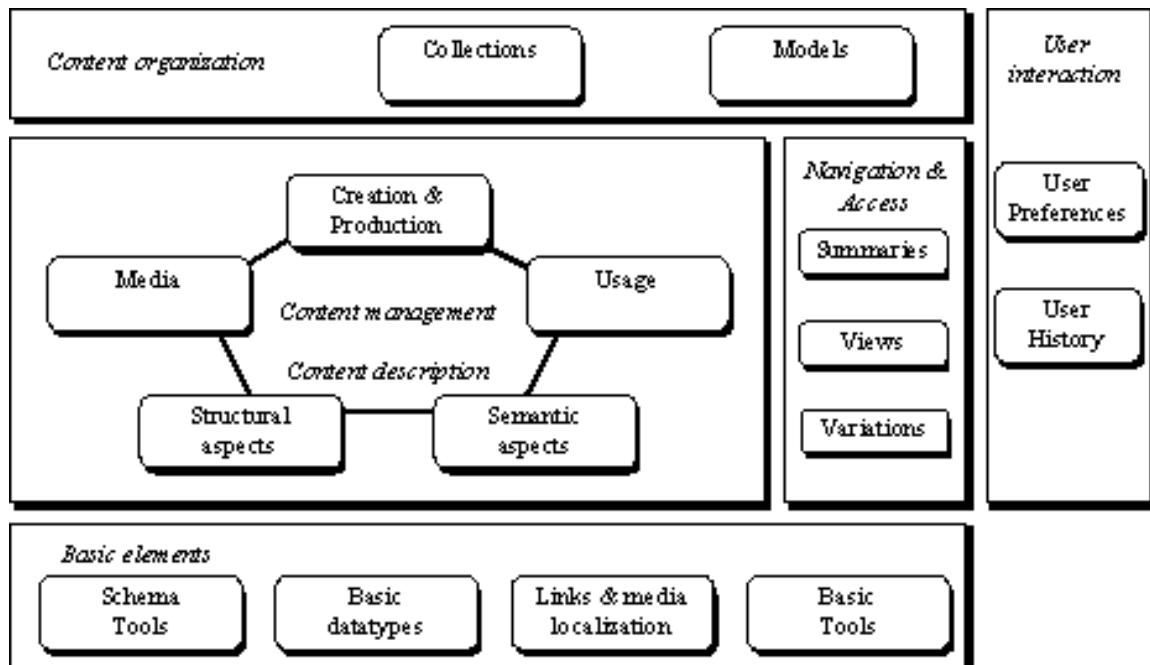


Figure 10.34: Overview of the MPEG-7 Multimedia DSS.

linking media files, localizing pieces of content, and describing time, places, persons, individuals, groups, organizations, and other textual annotation.

Content Description Content Description Tools represent perceptible information, including structural aspects (structure description tools), audio and visual features, and conceptual aspects (semantic description tools).

Structure description tools let us describe content in terms of spatio-temporal segments organized in a hierarchical structure, for example, letting us define a table of contents or an index. We can attach audio, visual, annotation, and content management description tools to the segments to describe them in detail.

Visual description tools include the visual basic structures (such as description tools for grid layout, time series, and spatial coordinates) and visual description tools that let us describe color, texture, shape, motion, localization, and faces.

Audio description tools comprise the audio description framework (including the scale tree for creating a scalable series of audio samples, low-level audio descriptors, and the silence descriptor) and high-level audio description tools that let us describe musical instrument timbre, sound recognition, spoken content, and melody.

Semantic description tools let us describe the content with real-world semantics and conceptual notions: objects, events, abstract concepts, and relationships. We can cross-link the semantic and structure description tools with a set of links.

Content Management The Content Management Tools let us specify information about media features, creation, and usage of multimedia content.

Media description tools let us describe the storage media, coding format, quality, and transcoding hints for adapting content to different networks and terminals.

Creation description tools let us describe the creation process (for example, title, agents, materials, places, and dates), classification (for example, genre, subject, parental rating, and languages), and related materials. This information may very likely be subject to change during the lifetime of the audio-visual (AV) content.

Usage description tools let us describe the conditions for use (for example, rights and availability) and the history of use (for example, financial results and audience).

Content Organization MPEG-7 provides also DSs for organizing and modeling collections of audio-visual content and of descriptions. We can describe each collection as a whole by their attribute values characterized by models and statistics.

Navigation and Access Navigation and Access Tools let us specify summaries, partitions and decompositions, and variations of multimedia content for facilitating browsing and retrieval.

Summary description tools provide both hierarchical and sequential navigation modes to provide efficient preview access to the multimedia material.

Partitions and decompositions description tools allow multiresolution and progressive access in time, space, and frequency.

Variations description tools let us describe preexisting views of multimedia content: summaries, different media modalities, (for example, image and text), scaled versions, and so on.

User Interaction The User Interaction tools describe user preferences and usage history pertaining to the consumption of the multimedia material. This allows, for example, matching between user preferences and MPEG-7 content descriptions in order to facilitate personalization of audio-visual content access, presentation and consumption.

10.5.6 MPEG-7 Audio

MPEG-7 Audio comprises five technologies: the audio description framework (which includes scalable series, low-level descriptors, and the uniform silence segment), musical instrument timbre description tools, sound recognition tools, spoken content description tools, and melody description tools.

10.5.6.1 MPEG-7 Audio Description Framework

The Audio Framework contains low-level tools designed to provide a basis for the construction of higher level audio applications. By providing a common platform for the structure of descriptions and the basic semantics for commonly regarded audio features, MPEG-7 Audio establishes a platform for interoperability across all applications that might be built on the framework.

There are essentially two ways of describing low-level audio features. One may sample values at regular intervals or one may use Segments (see the MDS description) to demarcate regions of similarity and dissimilarity within the sound. Both of these possibilities are embodied in two low-level descriptor types (one for scalar values, such as power or fundamental frequency, and one for vector types, such as spectra), which create a consistent interface. Any descriptor inheriting from these types can be instantiated, describing a segment with a single summary value or a series of sampled values, as the application requires.

The sampled values themselves may be further manipulated through another unified interface: they can form a Scalable Series. The Scalable Series allows one to progressively down-sample the data contained in a series, as the application, bandwidth, or storage required. The scale tree may also store



various summary values along the way, such as minimum, maximum, and variance of the descriptor values.

The low-level audio descriptors are of general importance in describing audio. There are seventeen temporal and spectral descriptors that may be used in a variety of applications. They can be roughly divided into the following groups:

Basic: The two basic audio Descriptors are temporally sampled scalar values for general use, applicable to all kinds of signals. The AudioWaveform Descriptor describes the audio waveform envelope (minimum and maximum), typically for display purposes. The AudioPower Descriptor describes the temporally-smoothed instantaneous power, which is useful as a quick summary of a signal, and in conjunction with the power spectrum, below.

Basic Spectral. The four basic spectral audio Descriptors all share a common basis, all deriving from a single time-frequency analysis of an audio signal. They are all informed by the first Descriptor, the AudioSpectrumEnvelope Descriptor.

AudioSpectrumEnvelope descriptor which is a logarithmic-frequency spectrum, spaced by a power-of-two divisor or multiple of an octave. This AudioSpectrumEnvelope is a vector that describes the short-term power spectrum of an audio signal. It may be used to display a spectrogram, to synthesize a crude “auralization” of the data, or as a general-purpose descriptor for search and comparison.

AudioSpectrumCentroid descriptor describes the center of gravity of the log-frequency power spectrum. This Descriptor is an economical description of the shape of the power spectrum, indicating whether the spectral content of a signal is dominated by high or low frequencies.

AudioSpectrumSpread descriptor complements the previous Descriptor by describing the second moment of the log-frequency power spectrum, indicating whether the power spectrum is centered near the spectral centroid, or spread out over the spectrum. This may help distinguish between pure-tone and noise-like sounds.

AudioSpectrumFlatness descriptor describes the flatness properties of the spectrum of an audio signal for each of a number of frequency bands. When this vector indicates a high deviation from a flat spectral shape for a given band, it may signal the presence of tonal components.

Signal parameters The two signal parameter Descriptors apply chiefly to periodic or quasi-periodic signals.

AudioFundamentalFrequency descriptor describes the fundamental frequency of an audio signal. The representation of this descriptor allows for a confidence measure in recognition of the fact that the various extraction methods, commonly called “pitch-tracking,” are not perfectly accurate, and in recognition of the fact that there may be sections of a signal (e.g., noise) for which no fundamental frequency may be extracted.

AudioHarmonicity descriptor represents the harmonicity of a signal, allowing distinction between sounds with a harmonic spectrum (e.g., musical tones or voiced speech [e.g., vowels]), sounds with an inharmonic spectrum (e.g., metallic or bell-like sounds) and sounds with a non-harmonic spectrum (e.g., noise, unvoiced speech [e.g., fricatives like /f/], or dense mixtures of instruments).

Timbral Temporal The two timbral temporal descriptors describe temporal characteristics of segments of sounds, and are especially useful for the description of musical timbre (characteristic tone quality independent of pitch and loudness). Because a single scalar value is used to represent the



evolution of a sound or an audio segment in time, these Descriptors are not applicable for use with the Scalable Series.

LogAttackTime descriptor characterizes the “attack” of a sound, the time it takes for the signal to rise from silence to the maximum amplitude. This feature signifies the difference between a sudden and a smooth sound.

TemporalCentroid descriptor also characterizes the signal envelope, representing where in time the energy of a signal is focused. This Descriptor may, for example, distinguish between a decaying piano note and a sustained organ note, when the lengths and the attacks of the two notes are identical.

Timbral Spectral The five timbral spectral Descriptors are spectral features in a linear-frequency space especially applicable to the perception of musical timbre.

SpectralCentroid descriptor is the power-weighted average of the frequency of the bins in the linear power spectrum. As such, it is very similar to the AudioSpectrumCentroid Descriptor, but specialized for use in distinguishing musical instrument timbres. It has a high correlation with the perceptual feature of the “sharpness” of a sound.

The four remaining timbral spectral Descriptors operate on the harmonic regularly-spaced components of signals. For this reason, the descriptors are computed in linear-frequency space.

HarmonicSpectralCentroid is the amplitude-weighted mean of the harmonic peaks of the spectrum. It has a similar semantic to the other centroid Descriptors, but applies only to the harmonic (non-noise) parts of the musical tone. T

HarmonicSpectralDeviation descriptor indicates the spectral deviation of log-amplitude components from a global spectral envelope.

HarmonicSpectralSpread describes the amplitude-weighted standard deviation of the harmonic peaks of the spectrum, normalized by the instantaneous HarmonicSpectralCentroid.

HarmonicSpectralVariation descriptor is the normalized correlation between the amplitude of the harmonic peaks between two subsequent time-slices of the signal.

Spectral Basis The two spectral basis Descriptors represent low-dimensional projections of a high-dimensional spectral space to aid compactness and recognition. These descriptors are used primarily with the Sound Classification and Indexing Description Tools, but may be of use with other types of applications as well.

AudioSpectrumBasis descriptor is a series of (potentially time-varying and/or statistically independent) basis functions that are derived from the singular value decomposition of a normalized power spectrum.

AudioSpectrumProjection descriptor is used together with the AudioSpectrumBasis Descriptor, and represents low-dimensional features of a spectrum after projection upon a reduced rank basis.

Together, the descriptors may be used to view and to represent compactly the independent subspaces of a spectrogram. Often these independent subspaces (or groups thereof) correlate strongly with different sound sources. Thus one gets more salience and structure out of a spectrogram while using less space. For example, in Figure 8.35, a pop song is represented by an AudioSpectrumEnvelope Descriptor, and visualized using a spectrogram. The same song has been data-reduced in Figure 8.36, and yet the individual instruments become more salient in this representation.



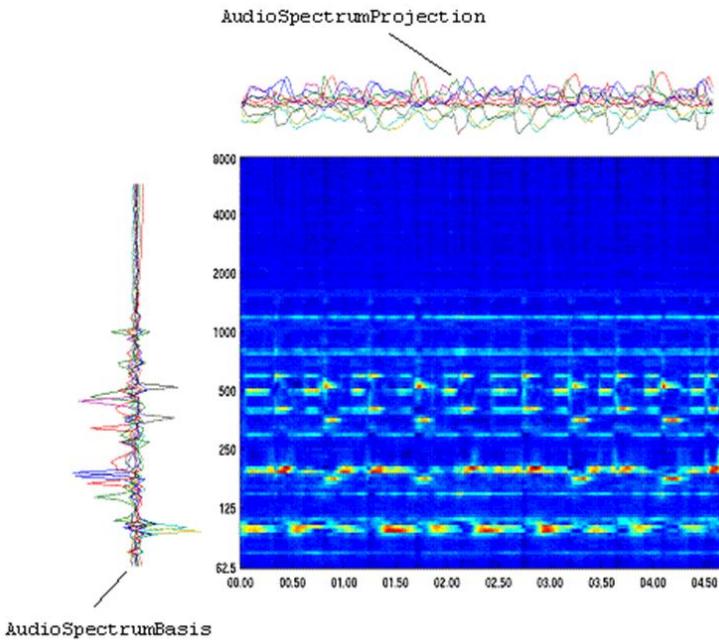


Figure 10.35: *AudioSpectrumEnvelope* description of a pop song. The required data storage is NM values where N is the number of spectrum bins and M is the number of time points.

Each of these can be used to describe a segment with a summary value that applies to the entire segment or with a series of sampled values. The Timbral Temporal group is an exception, as their values only apply to segments as a whole.

While low-level audio descriptors in general can serve many conceivable applications, the spectral flatness descriptor specifically supports the functionality of robust matching of audio signals. Applications include audio fingerprinting, identification of audio based on a database of known works and, thus, locating metadata for legacy audio content without metadata annotation.

Additionally, a very simple but useful tool is the

Silence descriptor. It attaches the simple semantic of “silence” (i.e. no significant sound) to an Audio Segment. It may be used to aid further segmentation of the audio stream, or as a hint not to process a segment.

10.5.6.2 High-level audio description tools (Ds and DSS)

Because there is a smaller set of audio features (as compared to visual features) that may canonically represent a sound without domain-specific knowledge, MPEG-7 Audio includes a set of specialized high-level tools that exchange some degree of generality for descriptive richness. The five sets of audio Description Tools that roughly correspond to application areas are integrated in the standard: audio signature, musical instrument timbre, melody description, general sound recognition and indexing, and spoken content. The latter two are excellent examples of how the Audio Framework and MDS Description Tools may be integrated to support other applications.

Audio Signature Description Scheme While low-level audio Descriptors in general can serve many conceivable applications, the spectral flatness Descriptor specifically supports the functionality of robust matching of audio signals. The Descriptor is statistically summarized in the AudioSignature

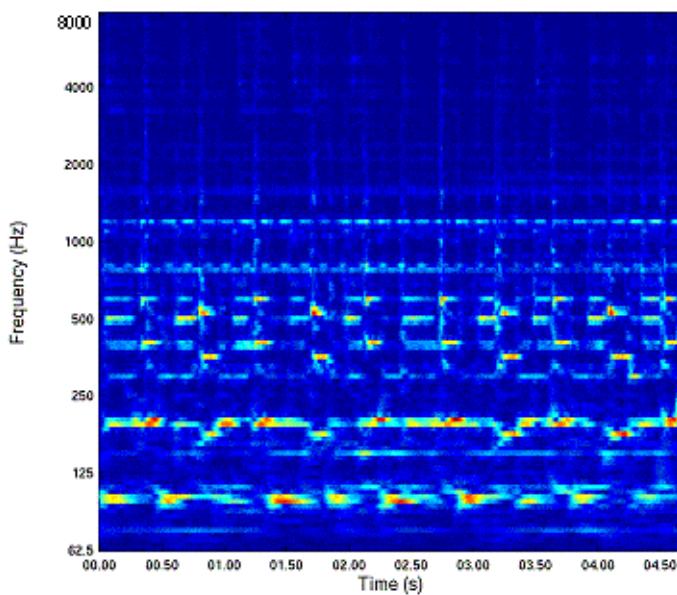


Figure 10.36: A 10-basis component reconstruction showing most of the detail of the original spectrogram including guitar, bass guitar, hi-hat and organ notes. The left vectors are an *AudioSpectrumBasis Descriptor* and the top vectors are the corresponding *AudioSpectrumProjection Descriptor*. The required data storage is $10(M + N)$ values.

Description Scheme as a condensed representation of an audio signal designed to provide a unique content identifier for the purpose of robust automatic identification of audio signals. Applications include audio fingerprinting, identification of audio based on a database of known works and, thus, locating metadata for legacy audio content without metadata annotation

Musical Instrument Timbre description tools Timbre descriptors aim at describing perceptual features of instrument sounds. Timbre is currently defined in the literature as the perceptual features that make two sounds having the same pitch and loudness sound different. The aim of the Timbre description tools is to describe these perceptual features with a reduced set of descriptors. The descriptors relate to notions such as “attack”, “brightness” or “richness” of a sound. Within four detailed possible classes of musical instrument sounds, two classes are well detailed, and had been the subject of core experiment development. At this point, harmonic, coherent sustained sounds, and non-sustained, percussive sounds are represented in the standard. The timbre descriptor for sustained harmonic sounds combines the above Timbral Spectral low-level descriptors with a log attack time descriptor. The percussive instrument descriptor combines the Timbral Temporal low-level descriptors with a spectral centroid descriptor. Comparisons between descriptions using either set of descriptors are done with an experimentally-derived scaled distance metric.

Melody Description Tools The melody Description Tools include a rich representation for monophonic melodic information to facilitate efficient, robust, and expressive melodic similarity matching. The Melody Description Scheme includes a MelodyContour Description Scheme for extremely terse, efficient melody contour representation, and a MelodySequence Description Scheme for a more verbose, complete, expressive melody representation. Both tools support matching between melodies, and can support optional supporting information about the melody that may further aid content-based search, including query-by-humming.

MelodyContour Description Scheme uses a 5-step contour (representing the interval difference between adjacent notes), in which intervals are quantized into large or small intervals, up, down, or the same. The Melody Contour DS also represents basic rhythmic information by storing the number of the nearest whole beat of each note, which can dramatically increase the accuracy of matches to a query.

MelodySequence. For applications requiring greater descriptive precision or reconstruction of a given melody, the MelodySequence Description Scheme supports an expanded descriptor set and high precision of interval encoding. Rather than quantizing to one of five levels, the precise pitch interval (to cent or greater precision) between notes is kept. Precise rhythmic information is kept by encoding the logarithmic ratio of differences between the onsets of notes in a manner similar to the pitch interval. Arrayed about these core Descriptors are a series of optional support Descriptors such as lyrics, key, meter, and starting note, to be used as desired by an application.

Sound recognition tools The sound recognition descriptors and description schemes are a collection of tools for indexing and categorization of general sounds, with immediate application to sound effects. Support for automatic sound identification and indexing is included as well as tools for specifying a taxonomy of sound classes and tools for specifying an ontology of sound recognizers. Such recognizers may be used to automatically index and segment sound tracks.

The recognition tools use the low-level Spectral Basis descriptors as their foundation. These basis functions are then further segmented into a series of states that comprise a statistical model, such as a hidden Markov or Gaussian mixture model, which is then trained on the likely transitions between these states. This model may stand on its own as a representation, have a label associated with it according to the semantics of the original sound, and/or associated with other models in order to categorize novel sounds input into a recognition system.

Spoken Content description tools The Spoken Content description tools allow detailed description of words spoken within an audio stream. In recognition of the fact that current Automatic Speech Recognition (ASR) technologies have their limits, and that one will always encounter out-of-vocabulary utterances, the Spoken Content description tools sacrifice some compactness for robustness of search. To accomplish this, the tools represent the output and what might normally be seen as intermediate results of Automatic Speech Recognition (ASR). The tools can be used for two broad classes of retrieval scenario: indexing into and retrieval of an audio stream, and indexing of multimedia objects annotated with speech.

The Spoken Content description tools are divided into two broad functional units: the lattice, which represents the actual decoding produced by an ASR engine, and the header, which contains information about the speakers being recognized and the recognizer itself. The lattice consists of combined word and phone lattices for each speaker in an audio stream. By combining these lattices, the problem of out-of-vocabulary words is greatly alleviated and retrieval may still be carried out when the original word recognition was in error.



Chapter 11

Multimodal interaction

Giovanni De Poli and Federico Avanzini

Copyright © 2005-2018 Giovanni De Poli and Federico Avanzini
except for paragraphs labeled as *adapted from <reference>*

This book is licensed under the CreativeCommons Attribution-NonCommercial-ShareAlike 3.0 license. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/>, or send a letter to Creative Commons, 171 2nd Street, Suite 300, San Francisco, California, 94105, USA.

In human-human communication, interpreting the mix of audio-visual signals is essential in communicating. Researchers in many fields recognize this, and thanks to advances in the development of unimodal techniques (in speech and audio processing, computer vision, etc.), and in hardware technologies (inexpensive cameras and other types of sensors), there has been a significant growth in multimodal interaction research.

As in human-human communication, however, effective communication is likely to take place when different input devices are used in combination. Multimodal interfaces have been shown to have many advantages: they prevent errors, bring robustness to the interface, help the user to correct errors or recover from them more easily, bring more bandwidth to the communication, and add alternative communication methods to different situations and environments. Disambiguation of error-prone modalities using multimodal interfaces is one important motivation for the use of multiple modalities in many systems.

A multimodal system is simply one that responds to inputs in more than one modality or communication channel (e.g., speech, gesture, writing, and others). We use a human-centred approach and by modality we mean mode of communication according to human senses and computer input devices activated by humans or measuring human qualities. The human senses are sight, touch, hearing, smell, and taste. The input modalities of many computer input devices can be considered to correspond to human senses: cameras (sight), haptic sensors (touch), microphones (hearing), olfactory(smell), and even taste. Many other computer input devices activated by humans, however, can be considered to correspond to a combination of human senses, or to none at all: keyboard, mouse, writing tablet, motion input(e.g., the device itself is moved for interaction), galvanic skin response, and other biometric sensors.

In the context of HCI, multimodal techniques can be used to construct many different types of interfaces. Of particular interest are perceptual, attentive, and enactive interfaces. Perceptual interfaces are highly interactive, multimodal interfaces that enable rich, natural, and efficient interaction with computers. Perceptual interfaces seek to leverage sensing (input)and rendering (output) technologies in order to provide interactions not feasible with standard interfaces and common I/O devices such as the keyboard,

the mouse, and the monitor, making computer vision a central component in many cases. Attentive interfaces are context-aware interfaces that rely on a person's attention as the primary input - that is, attentive interfaces use gathered information to estimate the best time and approach for communicating with the user. Since attention is epitomized by eye contact and gestures (although other measures such as mouse movement can be indicative), computer vision plays a major role in attentive interfaces. Enactive interfaces are those that help users communicate a form of knowledge based on the active use of the hands or body for apprehension tasks. Enactive knowledge is not simply multisensory mediated knowledge, but knowledge stored in the form of motor responses and acquired by the act of *doing*. Typical examples are the competence required by tasks such as typing, driving a car, dancing, playing a musical instrument, and modelling objects from clay. All of these tasks would be difficult to describe in an iconic or symbolic form.

In this chapter different perspectives on multimodal interaction, with special emphasis on sound, music and creativity, will be presented.

11.1 Research paradigms on sound and sense

In this section¹ it is shown that the modern scientific approach to sound and music computing has historical roots in research traditions that aimed at understanding the relationship between sound and sense, physics and meaning. This chapter gives a historical-philosophical overview of the different approaches that led to the current computational and empirical approaches to music cognition. It is shown that music cognition research has evolved from Cartesian dualism with a rather strict separation between sound and sense, to an approach in which sense is seen as embodied and strongly connected to sound. Along this development, music research has always been a driver for new developments that aim at bridging the gap between sound and sense. This culminates in recent studies on gesture and artistic applications of music mediation technologies.

In all human musical activities, musical sound and sense are tightly related with each other. Sound appeals to the physical environment and its moving objects, whereas sense is about private feelings and the meanings it generates. Sound can be described in an objective way, using instruments and machines. In contrast, sense is a subjective experience which may require a personal interpretation in order to explicit this experience. How are the two related with each other? And what is the aim of understanding their relationship?

11.1.1 From music philosophy to music science

Ancient Greek philosophers such as Pythagoras, Aristoxenos, Plato and Aristotle had quite different opinions about the relationship between sound and sense. Pythagoras mainly addressed the physical aspects by considering a mathematical order underlying harmonic pitch relationships. In contrast, Aristoxenos addressed perception and musical experience. Plato comes into the picture mainly because he attributed strong powers to music, thus, a strong effect from sound to sense. However, for him, it was a reason to abandon certain types of music because of the weakening effect music could have on the virtue of young people. Aristotle understood the relationship between sound and sense in terms of a mimesis theory (e.g. Politics, Part V). In this theory, he stated that rhythms and melodies contain similarities with the true nature of qualities in human character, such as anger, gentleness, courage, temperance and the contrary qualities. By imitating the qualities that these characters exhibit in music, our souls² are moved

¹adapted from Marc Leman, Frederik Styns and Nicola Bernardini S2S2book Polotti and Rocchesso [2008]

²The concept of soul is an old philosophical concept while modern philosophers tend to relate to the more modern concepts of "self", "ego", or even "mind".



in a similar way, so that we become in tune with the affects we experience when confronted with the original. For example, when we hear imitations of men in action in music, then our feelings tend to move in sympathy with the original. When listening to music, our soul thus undergoes changes in tune with the affective character being imitated.

Understood in modern terms, Aristotle thus observed a close connection between sound and sense in that the soul would tend to move along or resonate with sound features in music that mimic dynamic processes (gestures, expressions) in humans. Anyhow, with these views on acoustics (Pythagoras' approach to music as ratios of numbers), musical experience (Aristoxenos' approach to music as perceived structure), and musical expressiveness (Aristotle's approach to music as imitation of reality), there was sufficient material for a few centuries of philosophical discussion about sound to sense relationships.

All this started again from scratch in the 17th century with the introduction of the so-called Cartesian dualism, which states that sound and sense are two entirely different things. He divided music into three basic components, namely, (i) the mathematical-physical aspect (Pythagoras), (ii) the sensory perception (Aristoxenos), and (iii) the ultimate effect of music perception on the individual listener's soul (or mind) (Aristotle). To Descartes, it is sound, and to some degree also sensory perception, that can be the subject of a scientific study. The reason is that sound, as well as the human ear, deal with physical objects. Since objects have extension and can be put into motion, we can apply our mathematical methods to them. In contrast, the effect of perception and the meaning that ensues from it resides in the soul. There it can be a subject of introspection. In that respect, sense is less suitable to scientific study because sense has no extension.

In more recent times, this concept will reappear as "body image" and "body schema". So far, Descartes' approach thus clearly distinguished sound and sense. His focus on moving objects opened the way for scientific investigations in acoustics and psychoacoustics, and it pushed matters related to sense and meaning a bit further away towards a disembodied mental phenomenon.

Like Descartes, many scientists working on music often stressed the mathematical and physical aspects, whereas the link with musical meaning was more a practical consequence. For example, the calculation of pitch tunings for clavichord instruments was often (and sometimes still is) considered to be a purely mathematical problem, yet it had consequences for the development of the harmonic and tonal system and the way it touches our sense of tone relationships and, ultimately, our mood and emotional involvement with music. Emotions and expressive gestures were not considered to be a genuine topic of scientific study. Emotions and expressive gestures were too much influenced by sound, and therefore, since they were not based on pure thinking, they were prone to error and not reliable as a basis for scientific study and knowledge. Thus, while Plato and Aristotle saw a connection between sound and sense through mimesis, Descartes claimed that sound and sense had a different ontology.

Parallel with this approach, the traditions of Aristoxenos and Aristotle also culminated in rule-based accounts of musical practices such as Zarlino's, and later Rameau's and Mattheson's. Somehow, there was the feeling that aspects of perception which closely adhere to the perception of syntax and structure had a foundation in acoustics. Yet not all aspects could be explained by it. The real experience of its existence, the associated feeling, mood and pleasure were believed to belong to the subject's private life, which was inaccessible to scientific investigation.

Descartes' dualism had a tremendous impact on scientific thinking and in particular also on music research. The science of sound and the practice of musical experience and sense were no longer connected by a common concept. Sound was the subject of a scientific theory, while sense was still considered to be the by-product of something subjective that is done with sound. Apart from sensory perception (e.g. roughness in view of tuning systems), there was no real scientific theory of sense, and so, the gap between mind and matter, sense and sound, remained large.



11.1.2 The cognitive approach

Empirical studies of subjective involvement with music started to take place in the 19th century. Through the disciplines of psychophysics and psychology, the idea was launched that between sound and sense there is the human brain, whose principles could also be understood in terms of psychic principles and later on, as principles of information processing. With this development, the processes that underlay musical sense come into the picture.

11.1.2.1 Psychoacoustics

With the introduction of psychoacoustics by von Helmholtz (1863), the foundations were laid for an information processing approach to the sound/sense relationship. Helmholtz assumed that musical sensing, and ultimately, its experience and sense, was based on physiological mechanisms in the human ear. This idea became very influential in music research because it provided an explanation of why some very fundamental structural aspects of musical sense, such as consonance and dissonance, harmony and tonality, had an impact on our sense. This impact was no longer purely a matter of acoustics, but also of the working of our sensing system. Through scientific experiments, the causality of the mechanisms (still seen as a moving object) could be understood and mathematical functions could capture the main input/output relationships. This approach provided the foundation for experimental psychology and later for Gestalt psychology in the first half of the 20th century, and the cognitive sciences approach of the second half of the 20th century.

11.1.2.2 Gestalt psychology

The Gestalt movement gained prominence by about 1920. It had a major focus on sense as the perception and representation of musical structure, including the perception of tone distances and intervals, melodies, timbre, as well as rhythmic structures. The Gestalt approach influenced music research in that it promoted a thorough structural and cognitive account of music perception based on the idea that sense emerges as a global pattern from the information processing of patterns contained in musical sound.

11.1.2.3 Information theory

It also gradually became clear that technology would become an important methodological pillar of music research, next to experimentation. Soon after 1945, with the introduction of electronics and the collaboration between engineers and composers, electronic equipment was used for music production activities, and there was a need for tools that would connect musical thinking with sound energies. This was a major step in the development of technologies which extend the human mind to the electronic domain in which music is stored and processed. Notions such as entropy and channel capacity provided objective measures of the amount of information contained in music and the amount of information that could possibly be captured by the devices that process music. The link from information to sense was easily made. Music, after all, was traditionally conceived of in terms of structural parameters such as pitch and duration. Information theory thus provided a measurement, and thus a higher-level description, for the formal aspects of musical sense. Owing to the fact that media technology allowed the realisation of these parameters into sonic forms, information theory could be seen as an approach to an objective and relevant description of musical sense.



11.1.2.4 Symbol-based modelling of cognition

The advent of computers marked a new area in music research. Computers could replace analogue equipment, but, apart from that, it also became possible to model the mind according to the Cartesian distinction between mind and matter. Based on information processing psychology and formal linguistics it was believed that the *res cogitans* (related to that aspect of the mind which also Descartes had separated from matter) could be captured in terms of symbolic reasoning. Computers now made it possible to mimic human "intelligence" and develop an "artificial intelligence".

Cognitive science, as the new trend was called, conceived the human mind in terms of a machine that manipulates representations of content on a formal basis. The application of the symbol-based paradigm to music was very appealing. However, the major feature of this approach is that it works with a conceptualisation of the world which is cast in symbols, while in general it is difficult to pre-define the algorithms that should extract the conceptualised features from the environment.

11.1.2.5 Subsymbol-based modelling of cognition

In the 1980s, based on the results of the so-called connectionist computation, a shift of paradigm from symbol-based modelling to subsymbol-based modelling was initiated. Connectionism (re)introduced statistics as the main modelling technique for making connections between sound and sense. This approach was rather appealing for music research because it could take into account the natural constraints of sound properties better than the symbol-based approach could. By including representations of sound properties (rather than focusing on symbolic descriptions which are devoid of these sound properties), the subsymbol-based approach was more in line with the naturalistic epistemology of traditional musical thinking. It held the promise of an ecological theory of music in which sound and sense could be considered as a unity. The method promised an integrated approach to psychoacoustics, auditory physiology, Gestalt perception, self-organisation and cognition, but its major limitation, however, was that it still focused exclusively on perception.

11.1.3 Beyond cognition

The cognitive tradition was criticised for several reasons. One reason was the fact that it neglected the subjective component in the subject's involvement with the environment. Another reason was that it neglected action components in perception and therefore remained too much focused on structure and form.

11.1.3.1 Embodied music cognition

The action-based viewpoint has generated a lot of interest and a new perspective on how to approach the sound/sense relationship. In this approach, the link between sound and sense is based on the role of action as mediator between physical energy and meaning. In the cognitive approach the sound/sense relationship was mainly conceived from the point of view of mental processing. The approach was effective in acoustics and structural understanding of music, but it was less concerned with action, gestures and emotional involvement. In that respect, one could say that the Aristotelian component, with its focus on mimesis as binding component between sound and sense, was not part of the cognitive programme, nor was multi-modal information processing, or the issue of action-relevant perception (as reflected in the ecological psychology of Gibson).

To sum up, the embodied cognition approach states that the sound/sense relationship is mediated by the human body, and this is put as an alternative to the disembodied cognition approach where the mind



is considered to be functioning on its own. The embodied cognition approach of the early 20th century is largely in agreement with recent thinking about the connections between perception and action.

11.1.3.2 Music and emotions

The study of subjective involvement with music draws upon a long tradition of experimental psychological research in which descriptions of emotion and affect are related to descriptions of musical structure. These studies take into account a subjective experience with music. Few authors, however, have been able to relate descriptions of musical affect and emotions with descriptions of the physical structure that makes up the stimulus. Most studies, indeed, interpret the description of structure as a description of perceived structure, and not as a description of physical structure. In other words, description of musical sense proceeds in terms of perceptual categories related to pitch, duration, timbre, tempo, rhythms, and so on.

11.1.3.3 Gesture modelling

During the last decade, research has been strongly motivated by a demand for new tools in view of the interactive possibilities offered by digital media technology. This stimulated the interest in gestural foundations of musical involvement. This gestural approach has been rather influential in that it puts more emphasis on sensorimotor feedback and integration, as well as on the coupling of perception and action. It is likely that more attention to the coupling of perception and action will result in more attention to the role of corporeal involvement in music, which in turn will require more attention to multi-sensory perception, perception of movement (kinaesthesia), affective involvement, and expressiveness of music.

11.1.3.4 Physical modelling

Much of the recent interest in gesture modelling has been stimulated by advances in physical modelling. A physical model of a musical instrument generates sound on the basis of the movements of physical components that make up the musical instrument. In contrast with spectral modelling, where the sound of a musical instrument is modelled using spectral characteristics of the signal that is produced by the instrument, physical modelling focuses on the parameters that describe the instrument physically, that is, in terms of moving material object components. Sound generation is then a matter of controlling the articulatory parameters of the moving components.

Physical models, so far, are good at synthesising individual sounds of the modelled instrument. And although it is still far from evident how these models may synthesise a score in a musically interesting way - including phrasing and performance nuances - it is certain that a gesture-based account of physical modelling is the way to proceed. Humans would typically add expressiveness to their interpretation, and this expressiveness would be based on the constraints of body movements that take particular forms and shapes, sometimes perhaps learned movement sequences and gestures depending on cultural traditions. One of the goals of gesture research related to music, therefore, aims at understanding the biomechanical and psychomotor laws that characterise human movement in the context of music production and perception.

11.1.3.5 Motor theory of perception

Physical models suggest a reconsideration of the nature of perception in view of stimulus-source relationships and gestural foundations of musical engagement.

Purves and Lotto (2003), for example, argue that invariance in perception is based on statistics of proper relationships between the stimulus and the source that produces the stimulus. Their viewpoint is



largely influenced by recent studies in visual perception. Instead of dealing with feature extraction and object reconstruction on the basis of properties of single stimuli, they argue that the brain is a statistical processor which constructs its perceptions by relating the stimulus to previous knowledge about stimulus-source relationships. Such a statistics, however, assumes that aspects related to human action should be taken into account because the source cannot be known unless through action. In that respect, this approach differs from previous studies in empirical modelling, which addressed perception irrespective of action related issues. Therefore, the emphasis of empirical modelling on properties of the stimulus should be extended with studies that focus on the relationship between stimulus and source, and between perception and action.

The extension of empirical modelling with a motor theory of perception is currently a hot topic of research. It has some very important consequences for the way we conceive of music research, and in particular also for the way we look at music perception and empirical modelling.

11.1.4 Embodiment and mediation technology

The embodiment hypothesis entails that meaningful activities of humans proceed in terms of goals, values, intentions and interpretations, while the physical world in which these activities are embedded can be described from the point of view of physical energy, signal processing, features and descriptors. In normal life, where people use simple tools, this difference between the subject's experiences and the physical environment is bridged by the perceptive and active capabilities of the human body. In that perspective, the human body can be seen as the natural mediator between the subject and the physical world. The subject perceives the physical world on the basis of its subjective and action-oriented ontology, and acts accordingly using the body to realise its imagined goals. Tools are used to extend the limited capacities of natural body. This idea can be extended to the notion of mediation technology.

For example, to hit a nail into a piece of wood, I will use a hammer as an extension of my body. And by doing this, I'll focus on the nail rather than on the hammer. The hammer can easily become part of my own body image, that is, become part of the mental representation of my (extended) body. My extended body then allows my mental capacities to cross the borders of my natural human body, and by doing this, I can realise things that otherwise would not be possible. Apart from hitting nails, I can ride a bike to go to the library, I can make music by playing an instrument, or I can use my computer to access digital music. For that reason, technologies that bridge the gap between our mind and the surrounding physical environment are called mediation technologies. The hammer, the bike, the musical instrument and the computer are mediation technologies. They influence the way in which connections between human experience (sense) and the physical environment (e.g. sound) can take place.

Mediation concerns the intermediary processes that bridge the semantic gap between the human approach (subject-centered) and the physical approach (object or sound-centered), but which properties should be taken into account in order to make this translation effective? The hammer is just a straightforward case, but what about music that is digitally encoded in an mp3-player? How can we access it in a natural way, so that our mind can easily manipulate the digital environment in which music is encoded? What properties of the mediation technology would facilitate access to digitally encoded energy? What mediation tools are needed to make this access feasible and natural, and what are their properties? The answer to this question is highly dependent on our understanding of the sound/sense relationship as a natural relationship. This topic is at the core of current research in music and sound computing.

11.1.4.1 An object-centered approach to sound and sense

State-of-the-art engineering solutions are far from being sufficiently robust for use in practical sense/sound applications. For example, (Paivo, 2007) demonstrates that the classical bottom-up approach (he took

the melody extraction from polyphonic audio as a case study, using state-of-the-art techniques in auditory modelling, pitch detection and frame-concatenation into music notes) has reached its performance platform. Similar observations have been made in rhythm and timbre recognition. The use of powerful stochastic and probabilistic modelling techniques (Hidden Markov Chains, Bayesian modelling, Support Vector Machines, Neural Networks) (see also <http://www.ismir.net/> for publications) do not really close this gap between sense and sound much further (De Mulder et al., 2006). The link between sound and sense turns out to be a hard problem. There is a growing awareness that the engineering techniques are excellent, but that the current approaches may be too narrow. The methodological problems relate to:

- *Unimodality*: the focus has been on musical audio exclusively, whereas humans process music in a multi-modal way, involving multiple senses (modalities) such as visual information and movement.
- *Structuralism*: the focus has been on the extraction of structure from musical audio files (such as pitch, melody, harmony, tonality, rhythm) whereas humans tend to access music using subjective experiences (movement, imitation, expression, mood, affect, emotion).
- *Bottom-up*: the focus has been on bottom-up (deterministic and learning) techniques whereas humans use a lot of top-down knowledge in signification practices.
- *Perception oriented*: the focus has been on the modelling of perception and cognition whereas human perception is based on action-relevant values.
- *Object/Product-centered*: research has focused on the features of the musical object (waveform), whereas the subjective factors and the social/cultural functional context in musical activities (e.g. gender, age, education, preferences, professional, amateur) have been largely ignored.

11.1.4.2 A subject-centered approach to sound and sense

Research on gesture and subjective factors such as affects and emotions show that more input should come from a better analysis of the subjective human being and its social/cultural context. That would imply: Multi-modality: the power of integrating and combining several senses that play a role in music such as auditory, visual, haptic and kinaesthetic sensing. Integration offers more than the sum of the contributing parts as it offers a reduction in variance of the final perceptual estimate. Context-based: the study of the broader social, cultural and professional context and its effect on information processing. Indeed, the context is of great value for the disambiguation of our perception. Similarly, the context may largely determine the goals and intended musical actions. Top-down: knowledge of the music idiom to better extract higher-level descriptors from music so that users can have easier access to these descriptors. Traditionally, top-down knowledge has been conceived as a language model. However, language models may be extended with gesture models as a way to handle stimulus disambiguation. Action: the action-oriented bias of humans, rather than the perception of structural form (or Gestalt). In other words, one could say that people do not move just in response to the music they perceive, rather they move to disambiguate their perception of music, and by doing this, they signify music. User-oriented: research should involve the user in every phase of the research. It is very important to better understand the subjective factors that determine the behavior of the user.

The subject-centered approach is complementary to the object-centered approach. Its grounding in an empirical and evidence-based methodology fits rather well with the more traditional engineering approaches. The main difference relates to its social and cultural orientation and the awareness that aspects of this orientation have a large impact on the development of mediation technology. After all, the relationship between sense and sound is not just a matter of one single individual person in relation to its musical environment. Rather, this single individual person lives in contact with other people, and



in a cultural environment. Both the social and cultural environment will largely determine what music means and how it can be experienced.

11.1.5 Music as innovator

The above historical and partly philosophical overview gives but a brief account of the different approaches to the sound and sense relationship. This account is certainly incomplete and open to further refinement. Yet a striking fact in this overview is that music, in spanning a broad range of domains from sound to sense and social interaction, appears to be a major driver for innovation. This innovation appears both in the theoretical domain where the relationship between body, mind, and matter is a major issue, and in the practical domain, where music mediation technology is a major issue.

The historical overview shows that major philosophical ideas, as well as technical innovations, have come from inside music thinking and engagement. Descartes' very influential dualist philosophy of mind was first developed in a compendium on music. Gestalt theory was heavily based on music research. Later on, the embodied cognition approach was first explored by people having strong roots in music playing (e.g. Truslit was a music teacher). In a similar way, the first explorations in electronic music mediation technologies were driven by composers who wanted to have better access to the electronic tools for music creation. Many of these ideas come out of the fact that music is fully embedded in sound and that the human body tends to behave in resonance with sound, whereas the "mind's I" builds up experiences on top of this. Music nowadays challenges what is possible in terms of object-centered science and technology and it tends to push these approaches more in the direction of the human subject and its interaction with other subjects. The human way in which we deal with music is a major driver for innovation in science and technology, which often approaches music from the viewpoint of sound and derived sound-features. The innovative force coming from music is related to the subject-centered issues that are strongly associated with creativity and social-cultural factors.

The idea that music drives innovation rather than vice versa should not come as completely unexpected. Music is solidly anchored to scientific foundations and as such it is an epistemological domain which may be studied with the required scientific rigour. However, music is also an art and therefore certain ways of dealing with music do not require scientific justification per se because they justify themselves directly in signification practices. The requirements of musical expression can indeed provide a formidable thrust to scientific and technological innovation in a much more efficient way than the usual R&D cycles may ever dream of. In short, the musical research carried out in our time by a highly specialised category of professionals (the composers) may be thought as a sort of fundamental think tank from where science and technology have extracted (and indeed, may continue to extract in the future) essential, revolutionary ideas. In short, musical expression requirements depend, in general, on large scale societal changes whose essence is captured by the sensible and attuned composers. These requirements translate quickly into specific technical requirements and needs. Thus, music acts in fact as an opaque but direct knowledge transfer channel from the subliminal requirements of emerging societies to concrete developments in science and technology.

Conclusion

This section aims at tracing the historical and philosophical antecedents of sense/sound studies in view of a modern action-oriented and social-cultural oriented music epistemology. Indeed, recent developments seem to indicate that the current interest in embodied music cognition may be expanded to social aspects of music making. In order to cross the semantic gap between sense and sound, sound and music computing research tends to expand the object-centered approach engineering with a subject-centered approach from the human sciences. The subject-centered character of music, that is, its sense, has always been



a major incentive for innovation in science and technology. The modern epistemology for sound and music computing is based on the idea that sound and sense are mediated by the human body, and that technology may form an extension of this natural mediator. The chapter aims at providing a perspective from which projections into the future can be made.

The section shows that the relationship between sound and sense is one of the main themes of the history and philosophy of music research. In this overview, attention has been drawn to the fact that three components of ancient Greek thinking already provided a basis for this discussion, namely, acoustics, perception, and feeling ("movement of the soul"). Scientific experiments and technological developments were first (17th - 18th century) based on an understanding of the physical principles and then (starting from the late 19th century) based on an understanding of the subjective principles, starting with principles of perception of structure, towards a better understanding of principles that underlay emotional understanding.

During the course of history, the problem of music mediation was a main motivating factor for progress in scientific thinking about the sound/sense relationship. This problem was first explored as an extension of acoustic theory to the design of music instruments, in particular, the design of scale tuning. In modern times this problem is explored as an extension of the human body as mediator between sound and sense. In the 19th century, the main contribution was the introduction of an experimental methodology and the idea that the human brain is the actual mediator between sound and sense.

In the last decades, the scientific approach to the sound/sense relationship has been strongly driven by experiments and computer modelling. Technology has played an increasingly important role, first as measuring instrument, later as modelling tool, and more recently as music mediation tools which allow access to the digital domain. The approach started from a cognitive science (which adopted Cartesian dualism) and symbolic modelling, and evolved to sub-symbolic modelling and empirical modelling in the late 1980ies. In the recent decades, more attention has been drawn to the idea that the actual mediator between sound and sense is the human body.

With regards to new trends in embodied cognition, it turns out that the idea of the human body as a natural mediator between sound and sense is not entirely a recent phenomenon, because these ideas have been explored by researchers such as Lipps, Truslit, Becking, and many others. What it offers is a possible solution to the sound/sense dichotomy by saying that the mind is fully embodied, that is, connected to body. Scientific study of this relationship, based on novel insights of the close relationship between perception and action, is now possible thanks to modern technologies that former generations of thinkers did not have at their disposal.

A general conclusion to be drawn from this overview is that the scientific methodology has been expanding from purely physical issues (music as sound) to more subjective issues (music as sense). Scientists conceived these transition processes often in relation to philosophical issues such as the mind-body problem, the problem of intentionality and how perception relates to action. While the sound/sense relationship was first predominantly considered from a cognitive/structural point of view, this viewpoint has gradually been broadened and more attention has been devoted to the human body as the natural mediator between sound and sense. Perception is no longer conceived in terms of stimulus and extraction of structures. Instead, perception is conceived within the context of stimulus disambiguation and simulated action, with the possibility of having loops of action-driven perception. This change in approach has important consequences for the future research. Music has thereby been identified as an important driver for innovation in science and technology. The forces behind that achievement are rooted in the fact that music has a strong appeal to multi-modality, top-down knowledge, context-based influences and other subject-centered issues which strongly challenge the old disembodied Cartesian approaches to scientific thinking and technology development.



11.2 Enaction, Arts and Creativity

Enaction and Creativity are two concepts difficult to define precisely. Enaction is here understood in a large sense of considering the role of action (and further of interaction, action could not exist without interaction) at the center of the human activities whatever they are, for human biological survival as well for human cultural creation of new objects or symbols. Creativity will be considered in this section³ according two simple meanings:

- mainly as synonymous of Artistic Creation Process
- and as creative processes in design and modelling activities.

Since the XIX century industrial revolution, creation activity in arts was mainly considered as an abstract activity starting the clearly cut separation between composer and instrumentalists in music and designers and producers in fine arts; choreographers and dancers in choreographic arts. The apogee of such period was at the middle of the XX century with the primacy of formal approaches in arts, as the serialism in music (synonymous of contemporary music) or the conceptualism in fine arts (synonymous of "contemporary arts").

At the beginning of the use of the computer in arts (at the middle of sixties), the main stream of theories and uses focuses on the conquest of "immateriality" allowed by computer. Keywords were "overcome the limit of the matter", "reach a pure thinking of musical cues", "Music for mind", "Abstraction for visual cues", "breaking the real", etc.

Recently, about ten of twenty years ago, after the relative failure of such extreme theories and points of view, and under the recent technological propositions of interactivity allowing the computers to be more and more adapted to the human senses and action, arts became more and more interactive. The "instrumentality" is progressively re-introduced as a design locus through its sub-instance of interactivity. The role of "gestures" has been rehabilitated, not only to produce sensorial predefined events but to properly create artistic properties. In music, performance, previously considered as the end of the musical production process, as a kind of "sonification" of the musical pre-written score, was rehabilitated as a creative process in itself, not only in musical improvisations as in specific musical styles (jazz, free music, etc.) but as a creative process in itself, as in open or interactive composition. Such approaches shift the creation process from the formal organization of musical or visual events to the production process itself. Simultaneously, the role of the "instrument" as a tangible object able to feed and steer the creative process by imposing constraints and of the "instrument trade" was rehabilitated against the "free constraint approaches".

Such theoretical shift is totally in adequacy with the concept of enaction. More, Arts is probably the realm (beside biology), in which high level media of communication and of cultural data are produced by means of closed-loop sensori-motor interaction.

This historical movement is particularly clear in Music, that is an "*allographic art*", needed another way of representation and of design - the musical graphical notation -different than itself. It is less evident in other arts, that are "*autographic arts*", as visual arts or choreographic arts, meaning that they are in themselves their own tools of representation and design. However, it traverses all the Arts that we called "Dynamic Instrumental Arts". "Instrumental arts" refer to arts that need physical medium (object, body) to exist. Dynamic refers to the fact that at any stage of its production process, sensorial artistic events are evolving events. Basic Dynamic Instrumental Arts are Music, Visual Arts as animation of fine arts, Choreographic arts. Each of them addresses the question of the role of the instrument and of the interaction between artists and their instruments in specific ways according to their own historical positioning and their own particularisms.

³adapted from Luciani Enactive Interfaces NoE WP13 report (2005)



Indeed, the word "*instrument*" is usually reserved to the musical realm. However, if we dare to use it in a more general meaning, as a physical mediator able to produce exteroceptive stimuli, visual and auditory, by an action of human body on it, we understand immediately that all the arts that need such mediator are necessarily temporally-based and interaction-based. Physical interaction, sensory-motor coupling, gesture, instrument, movement, etc. are complementary components that are always present and that are always cooperating in all sensory-based (conversely than language based) arts. Conversely, the question of the link with the interaction performance activity and the conceptual processes is raise as one of their core question. One main property of such "instrumental concept" is to reveal the implicit familiarity of artistic creation process with the "*enaction concept*".

In Musical arts, the concept of instrumentality is an ancestral concept that exists from the origin of the music and of the sound production. The pair instrument -instrumentalist is always present in music, even in computer music with the field of "Digital Musical Instruments". The main fundamental question in music is the link between the inevitable instrumental process and the musical notation and composition, and the link of the musical composition with musical perception and cognition.

In Visual Arts, two sub-domains have to be distinguished: arts that produce "static objects and events" (sculpture, paintings, etc.) and arts that produce "movements" or "moving objects and visual events" (movies, automata, animation). In the first case, as in Music, instrumentality is a native and ancient practice. The role of their matter and of the interaction with it is widely recognized and respected in such artistic style as well as in all craft practices. Differently than Music, as autographic arts, they don't need external and foreign way of notation and for their design and their composition. There is no so dramatic problem of notation and composition as it exists Music and no dramatic crack between compositional activity and other musical activities. In the second case, except in some minor cases, as shadows' theater or puppet theater, instrumentality is less difficult to define before the arrival of the computer. We cannot play with objects producing visual events as we are able to play with a violin. Movies and animation using conventional media (cinema or video) do not implement explicitly the instrumental concept. Similarly than in Music, the question of the motion notation is of a crucial and dramatic importance.

From the point of view of the novelty brought by computers, the two types of visual arts (static and moving) have been differently fed:

- computers trend to devalue and to minorate the type of craft manual and interactional process;
- computers triggered really a revolution in the visual art of motion by allowing the designing of "objects" that can be manipulated as "violin" to produce visual evolving events.

In both cases, Enactive Interfaces and Enactive Knowledge are means to experiment and to rehabilitate the prominent role of the interaction and of the matter in the visual artistic process.

In Choreographic arts, as in theater arts, "instrumentality" is not an explicit usual concept, the human body being its own instrument. The concept of instrument has been introduced recently with the introduction of the notion of "augmented body" by external devices and equipments able at least to capture the motion of the body. Such motion, transformed in a signal, becomes an "object" that can be processed and applied to control other objects and others instruments. Computers trends to bring together musical arts, visual dynamic arts and choreographic arts, the common concept becoming the instrumental concept with all its derivatives: interaction and interactive control. As in music and in visual motion, the core difficult non-solved question is this of the notation and of the composition of such evolving events.

Summarizing in a differentiate way the major questions risen by each of the main Dynamic Instrumental Arts, we can say:

- In *Music*, the haunting question being the relation between instrumentality and composition, are new computers tools and new ways of interaction with computerized instruments, able to overcome this frontier or not? Are Enactive Interfaces able to reconcile the opposites, the enemy brothers?



- In *Visual static arts*, is the generalization of interactivity concept able to instil in the production process, as in craft process, the minimum of instrumentality required to support craft know-how?
- In *Visual Dynamic Arts*, is the notion of virtual manipulable objects able to produce visual dynamic arts with the same level of quality for the visual shapes and for the expressivity of the motion? and is the motion processing able to overcome the duality between space (autographic representations) and time (allographic representations)?
- In *choreographic arts*, is computers a step in the motion representation without the creation of a break between choreographic performance and choreographic design, that is nowadays a core and passionate question?

From such contemporary questions asked by such arts, near to the enactive concept, to computers, some relevant but non-exhaustive issues can be listed:

1. What common issue? Is the motion and the gesture as a specific motion which represents action - its processing, its rendering, its production, its notation - a common feature shared by all such instrumental dynamic arts? Could the motion and the gesture the common mean to bring them near or to merge them in a very novel and genuine way?
2. What types of computer models and computer representations and interfaces should be the best candidates to receive gestures and to produce genuine movements;
3. What type of links between the primary evolving event (gesture, movement, action) and the sensory outputs visual and auditory? Trivial links only as those proposed now in computer graphics and animation? Arbitrary links as those proposed in the mapping process in computer music? Others links? Can we speak of gestures composition independently of the 3D object that is receiving or producing such motions? Can we apply every kind of gestures and action on every type of production process?
4. What types of design processes and link between the design process and the performance processes
5. What should be the relation between the enactive concept, well revealed by the necessity of the gestural interaction, and the artistic emotion? What is the role of the instrument and of the interaction in the shift from the production process to the aesthetic process? From the history of our artistic tools and theories, nowadays, we only know only that such mediator, such interaction cannot be totally avoided.

11.3 Some core questions about creativity: a philosophical and linguistic point of view

In this short section⁴, Roberto Casati (Institut Nicod, Paris) addresses some basic issues about creativity in a question-and-answer manner.

11.3.1 Creativity: eight basic questions

What is creativity?

This question is hard to answer, as is any "what is" question - and this difficulty is crucial. But much depends on it. First of all, we talk both about creative ideas and methods, and about creative people.

⁴adapted from R. Casati Enactive Interfaces NoE, WP13 report (2005).



Obviously the sense in which we talk about creative individuals completely depends on the sense that we give to the notion of creative ideas. We say that someone is creative when it has a creative idea. We do not say, on the other hand, that an idea or a method is creative because it comes from a creative individual, as if there was a magic creativity touch. A creative individual is one with creative ideas. First comes the idea and the process, then the label. It is important to see this because otherwise one is led to think that there is a sort of magic touch, a creativity faculty, or a creativity gift, that some people have and others do not. There may be people who produce more creative ideas than others, of course, but this depends on many factors, most of which are contextual, and does not depend on a mysterious creativity system of the brain.

So, what is a creative idea, a creative thought or method?

We all seem to be able, intuitively, to distinguish between a creative idea and an idea that is not so creative. But how do we actually draw the line between creative and non-creative ideas, and is there a fact of the matter that can justify the way we draw the line?

From a cognitive point of view most of what we know about creativity comes from our understanding of how language works. Think of the sentence I just uttered: "From a cognitive point of view most of what we know about creativity comes from our understanding of how language works". It is very likely to be the first time this sentence has been uttered in the whole history of mankind. I never heard it before anyway. And I never pronounced those words before. As least as it concerns me, I created it. But so is with most sentences, for purely combinatorial reasons. Think of a simple language in which sentences are made by simple juxtaposition of words (a very simple grammar). If there were only two words in this language, and you had a rule that sentences have exactly 8 words, then you would have 256 sentences. In a real language there are so many words, and no limit to the lengths of sentences. There is, indeed, an infinity of sentences to choose from. So creativity in language is almost mandatory -you cannot help, you are almost automatically creative.

We overlook this fact because it is so familiar for us, and it is taken for granted. But think of it. Sometimes indeed we catch ourselves in the act of repeating ourselves. We immediately recognize this fact. We would be surprised. If on the other hand someone repeats himself, we are annoyed. Either way, we seem very good at detecting non-creative behavior.

So, we are linguistically creative. Each of us is creative when it comes to speaking. When speaking, no one ever repeats, with some very clear exceptions, things that one has heard. If I tried to repeat word by word the sentence I mentioned before, I would not be able to; I would probably get more or less the same thought, but not the exact wording for it.

The crucial point is that this type of creativity in turn depends on there being rules that allow us to produce certain combinations of words and not others. I stress this point because there seems to be a romantic idea around in informal discussions about creativity: that creativity is a "breaking of the rules". We can agree that in certain contexts creativity is partly that, but the situation is far more complex. Creativity in language only exists because there are rules that allow us to form sentences out of words. Sentences (this is an important point) that you can understand. I am not creative at all if I say, "Grumpziso nemosenn ximadou", or "Subtly or John Cage", and I am creative if I say "the Moon is, actually, a giant dark stone". I cannot claim creativity if no one can understand what I say because I made up an invented language or made up a forbidden (ungrammatical) sentence.

But isn't there a difference between linguistic creativity and creativity in thought?

Indeed. I said that I may not be able to repeat the same words, but I would be able maybe to express the same thought. This means that language-creativity should be distinct from thought-creativity.



So what is it to create new thoughts, new ideas?

My suggestion is that the process is in principle not different from the process at work when we produce new sentences.

Let me state some conditions for creativity, that is, for recognition of an idea as creative.

- A. When we recognize that an idea is creative, we recognize that it is new. But "new" presupposes that we also recognize how the situation would have been without that idea.
- B. There is a (limited) tolerance for simultaneous creativity and novelty: two people can come up with a new idea at the same time. It is not uniqueness that makes an idea creative.
- C. Not only that. We tend to consider as creative those ideas that are solutions to problems. Someone just coming up with a novel scream is not particularly creative. "Creative" is reserved for ideas that come in and help making a progress on a background of an existing problem space. Here I am using "solutions" and "problems" in a very wide sense. There may be problems in jazz, in the arts, as there are in rationalizing the work-flow and in the engineering of an apparatus. There are mathematical problems and translation problems. All these may require creative solutions - as opposed to routine solutions. This means the above-mentioned romantic notion of creativity should give way to a less ambitious, but more true to the facts, description of how people come up with new ideas. It is tricky and hard to find out how this happens and I do not think that there is enough research available to make a final statement (More about this later).
- D. The existence of a set of rules is a precondition for creativity. There are for many reasons. For an idea to be recognized as creative, people must be able to see that it is new. But they can only see its novelty if they understand the alternatives (the non-creative alternatives). Think of language again. In order for you to come up with a new sentence in English, you must know (implicitly) the rules of English.

Analogy, metaphors, and perceptual problem solving have been presented in the literature as the nuts and bolts of creative processes. For instance, problem solving has been associated to a kind of perception. It is a way to see things in a new perspective. It is a "aha!" experience, similar to the one we undergo when we recognize a face in a set of lines, or a dalmatian dog in a set of patches.

Some philosophers and cognitive scientists have tried to develop "artificial creativity". Why is this interesting?

By reflecting on how to produce machines that are recognized as creative, one may on the way find out some so far overlooked features of creativity. Among others, Douglas Hofstadter (1995) and Paul Thagard (1995) - the latter has studied how analogy enters scientific thought. In both case it is understood that some creative ideas come from the use of analogy. You look at an object, it reminds you of something else you know, then you use some of the properties of the old thing to improve on the new thing (as an example, think of the desktop analogy for running your computer.)

Margared Boden (1990, 1994), a philosopher of artificial intelligence, has distinguished a "combinational creativity" which mainly consists in a new arrangement of existing ideas; and a different creativity, in which not only there is a solution to an existing problem, but there is the creation of a new problem. This later type of creativity is "a structured, disciplined, sometimes even systematic search for the meanings promised by the new idea."

Boden thinks that creativity is basically an exploration of a conceptual space. A conceptual space can be a set of constraints. One must first accept the constraints, then explore the space. Imagine a railroad



example: If you set constraints to movement (railroads force you not to make narrow turns, you have no steps), thereby allowing for lesser degrees of freedom, you will be forced to find a solution to the path between two points that must be creative. So you may end up inventing tunnels and bridges, to keep your railroad even and to minimize bending. Or, you can put constraints on colors, and decide to take pictures in black and white. Then you have to act creatively in order to enhance contrasts for colors that have the same brightness, such as red and greens. Or again, you can decide to compose in jazz. The range of what you can compose is small as compared to the range of sounds that you can concatenate, and this forces you to find interesting concatenations.

Sometimes constraints depend on social acceptance. Sometimes they depend on perception. Not everything that is possible is perceptually valid. It is no good to loosen too much the time constraints on music. You cannot sensibly listen to a piece of music whose beginning is now, whose second note is in two years, etc.

It appears then that one type of creativity consists in exploring a given, preset space, a given a set of constraints. Another type consists in inventing new constraints, thereby creating a new space. (A new constraint space is like a new style.) Of course it is of no avail to invent a new style and then completely rigidify it -the constraints should allow for a certain degree of freedom. A good new style is one with productive constraints, constraints that allow us to create within the style.

Hence I would add another principle to the conditions for creativity:

- E. Creativity within a given space is fast; creating a new space is slow. In order to be creative, an idea must be recognized as creative, but if you change the language, the rules of the game, it will take time for people to see this. So you have to put great care in making sure that the transition to the new language, to the new game, is understood.)

Is creativity necessary for humans? If so, why?

Socially, creativity is clearly more than encouraged: it is taken for granted. Think, again, of language, of an ordinary conversation, such as the ones we all have and enjoy with friends. Social pressure is merciless. Apart from some very clearly defined contexts, people are required to be creative all the time. Imagine I repeat exactly, word by word, what a friend just said. Except for ironical usages, or for a request for clarification, this behavior would not be permissible, or it would be considered boring. Suppose, further, that I repeat word by word what the dominant personality of the group said. I would be stigmatized as servile. Suppose, again, that I repeated word by word something I just said, three or four times. I would be considered weird. Suppose, finally, that I say something quite articulate, and the next day some friend finds out that I reported, word by word, an opinion expressed in a newspaper. Then I would be considered a cheater. Boring, servile, weird, cheater - all these are verbal punishment for not having been creative. Again, creativity is taken for granted. It is the natural condition of us all in many social contexts.

So if there is all this pressure on creativity, it may well be that it depends on some evolutionary advantage that creative societies have had upon noncreative societies?

It may well be, but I hesitate in suggesting evolutionary explanations. Still, the explanation is suggestive. Cooperative behavior is another case for which an evolutionary explanation has been suggested. Cooperative societies may fare better than non-cooperative societies, and this is why we tend to cooperate so much, even in cases in which we would be, on a local scale, better off if we defaulted on cooperation. So if creativity was necessary, because it led to improved social settings, its adaptive advantage may have made it happen that evolution "discovered" it -and reinforced it socially. But, again, these are very rough speculations.



Is there a creative " type "? Are some people predisposed to be more creative than others?

This is an important point. The problem is that it may well be the case that "creativity" does not delineate a category at all. As I suggested before, there seem not there be any "creative" system in the brain. I would like to make a comparison with other so-called "personality" traits. Common sense treats people as courageous, as cowards, as intelligent or stupid. Some scientists have tried to come up with methods for measuring these alleged "traits" of people. The results are not very encouraging: It is hard to pinpoint a stupid behavior out of very specific contexts. There appears to be no general stupidity that manifests itself in many repeated occasions. What is worse, the occasions dump systematically the alleged trait. Two very famous experiments, dating back from the '50, and oftentimes replicated, have shown this point quite dramatically. In the Millgram experiment, it turned out that most people, even those who thought of themselves as provided with a strong personality, would inflict painful experiences on fellows just because someone told them to do so. In the Good Samaritan experiment, people were shown to be disposed to help someone in distress only if they had time to do it. So, unless you are prepared to say that most people are evil, you should give up personality traits as useful explanatory categories. People just tend to maximize the coherence of their actions and beliefs on a very narrow temporal scale: this is why it is easy to convince people to do even horrible things -on pain of losing face, say. If personality traits do not exist, or are trumped by contingencies of the situation, then there is no creativity trait - or if there is, we can always find a way to trump it. And then the question arises of how is it possible to stimulate creativity. The answer is that one should try and engineer situations in which people would naturally come up with new solutions.

11.4 Auditory displays and sound design

The goal of this section⁵ is to provide an overview of research in Sound Design and Auditory Display, from warning design and computer auditory display to Architecture and Media.

The section organization into six main subsection reflects the topics that have been most extensively studied in the literature, i.e., warnings, earcons, auditory icons, mapping, sonification, and sound design. Indeed, there are wide overlaps between these areas (e.g., sound design and mapping can be considered as part of a task of sonification).

11.4.1 Warnings, Alerts and Audio Feedback

Auditory warnings are perhaps the only kind of auditory displays that have been thoroughly studied and for whom solid guidelines and best design practices have been formulated. We can identify five areas of applications for auditory warnings: personal devices, transport, military, control rooms and geographic-scale alerts.

The scientific approach to auditory warnings is usually divided into the two phases of hearing and understanding, the latter being influenced by training, design, and number of signals in the set. Studies in hearing triggered classic guidelines: for example alarms should be set between 15 and 25 dB above the masked threshold of environment. Moreover they faced also the issue of design for understanding, by suggesting a sound coding system that would allow mapping different levels of urgency. The problem of the legacy with traditional warnings is important: e.g. sirens are usually associated with danger, and horns with mechanical failures. The retention of auditory signals is usually limited to 4 to 7 items that can be acquired quickly, while going beyond is hard. In order to ease the recalls, it is important to design the temporal pattern accurately. Moreover, there is a substantial difference in discriminating signals

⁵ Amalia de Götzen, Pietro Polotti, Davide Rocchesso



in absolute or relative terms. Alarm-related behaviors can be classified as Alarm-Initiated Activities (AIA) in routine events (where ready-made responses are adequate) and critical events (where deductive reasoning is needed). Designing good warnings means balancing between attention-getting quality of sound and impact on routine performance of operators.

“Auditory warning affordances” investigates on the use of “ecological” stimuli as auditory warnings. The expectation is that sounds that are representative of the event to which they are alarming would be more easily learnt and retained. By using evocative sounds, auditory warnings should express a potential for action: for instance, sound from a syringe pump should confer the notion of replacing the drug.

Some results are that:

- Learned mappings are not easy to override;
- There is a general resistance to radical departures in alarm design practice;
- Suitability of a sound is easily outweighed by lack of identifiability of an alarm function;
- Need for participatory design practice.

However, for affordances that are learnt through long-time practice, performance may still be poor if an abstract sound is chosen.

Case study: “acqua alta” in Venice Special cases of warnings are found where it is necessary to alert many people simultaneously. Sometimes, these people are geographically spread, and new criteria for designing auditory displays come into play. Avanzini and co-workers (2005) face the problem of a system alert for the town of Venice, periodically flooded by the so-called “acqua alta”, i.e. the high tide that covers most of the town with 10-40 cm of water. Nowadays, a system of 8 electromechanical and omnidirectional sirens provide an alert system for the whole historic town.

A study of the distribution of the signal levels throughout the town was first performed. A noise map of the current alert system used in Venice was realized by means of a technique that extracts building and terrain data from digital city maps in ArcView format with reasonable confidence and limited user intervention. Then a sound pressure level map was obtained by importing the ArcView data into SoundPLAN, an integrated software package for noise pollution simulations. This software is mainly based on a ray tracing approach. The result of the analysis was a significantly non-uniform distribution of the SPL throughout the town. One of the goals of this work is, thus, the redefinition and optimization of the distribution of the loudspeakers. The authors considered a Constraint Logic Programming (CLP) approach to the problem. CLP is particularly effective for solving combinatorial minimization problems. Various criteria were considered in proposing new emission points. For instance, the aforementioned Patterson’s recommendations require that the acoustic stimulus must be about 15 dB above the background noise to be clearly perceived. Also, installation and maintenance costs make it impractical to install more than 8 to 12 loudspeakers in the city area. By taking into account all of these factors, a much more effective distribution of the SPL of the alert signals was achieved. The second main issue of this work is the sound design of the alert signals. In this sense the key questions here considered are:

- How to provide information not only about the arrival of the tide but also about the magnitude of the phenomenon;
- How to design an alert sound system that would not need any listening-training, but only verbal/textual instructions.

Being Venice a tourist town, this latter point is particularly important. It would mean that any person should intuitively understand what is going on, not only local people. The choices of the authors went



towards abstract signals, i.e. earcons, structured as a couple of signals, according to the concept of “attenson” (attention-getting sounds). The two sound stages specify the rising of the tide and the tide level, respectively. Also, the stimulus must be noticeable without being threatening. The criteria for designing sounds providing different urgency levels were the variation of the fundamental frequency, the sound inharmonicity and the temporal patterns.

The validation of the model concludes the paper. The subjects did not receive any training but only verbal instructions. The alert signal was proved to be effective, and no difference between Venetians and not-Venetians was detected. In conclusion, a rich alert model for a very specific situation and for a particular purpose was successfully designed and validated. The model takes into account a number of factors ranging from the topography and architecture of Venice, to the need of culturally non-biased alert signal definition, as well as to the definition of articulated signals able to convey the gravity of the event in an intuitive way.

11.4.2 Earcons

Blattner, Sumikawa and Greenberg introduced the concept of *earcons*, defining them as “non-verbal audio messages that are used in computer/user interfaces to provide information to the user about some computer object, operation or interaction”. These messages are called *motives*, “brief succession of pitches arranged in such a way as to produce a tonal pattern sufficiently distinct to allow it to function as an individual recognizable entity”. Earcons must be learned, since there is no intuitive link between the sound and what it represents: the earcons are abstract/musical signals as opposed to auditory icons, where natural/everyday sounds are used in order to build auditory interfaces (see Section 9.4.3).

Brewster (1998) presents a new structured approach to auditory display defining composing rules and a hierarchical organization of musical parameters (timbre, rhythm, register, etc.), in order to represent hierarchical organizations of computer files and folders. Typical applications of this work are telephone-based interfaces (TBIs), where navigation is a problem due to visual display dimensions. As already mentioned, the main idea is to define a set of sound-design/composing rules for very simple “musical atoms”, the earcons, with the characteristics of being easily distinguishable one from the other.

11.4.3 Auditory Icons

Another concept has been introduced in the nineties by Gaver as an earcon counterpart: *auditory icons*. The basic idea is to use natural and everyday sounds to represent actions and sounds within an interface. He individuates a fundamental aspect of our way of perceiving the surrounding environment by means of our auditory system. Trying to reply to the question “what do we hear in the world?”, a first and most relevant consideration emerges: a lot of research efforts were and are devoted to the study of musical perception, while our auditory system is first of all a tool for interacting with the outer world in everyday life.

When we consciously listen to or hear more or less unconsciously “something” in our daily experience, we do not really perceive and recognize sounds but rather events and sound sources. This “natural” listening behavior is denoted by Gaver as “everyday listening” as opposed to “musical listening”, where the perceptual attributes are those considered in the traditional research in audition. As an example, Gaver writes: “while listening to a string quartet we might be concerned with the patterns of sensation the sounds evoke (musical listening), or we might listen to the characteristics and identities of the instruments themselves (everyday listening). Conversely, while walking down a city street we are likely to listen to the sources of sounds - the size of an approaching car, how close it is and how quickly it is approaching.” Despite the importance of non-musical and non-speech sounds, the research in this field



is scarce. What Gaver writes is true: we do not really know how our senses manage to gather so much information from a situation like the one of the approaching car described above.

11.4.4 Mapping

Auditory Display in general, and Sonification in particular, are about giving an audible representation to information, events, and processes. These entities may take a variety of forms and can be reduced to space- or time-varying data. In any case, the main task of the sound designer is to find an effective mapping between the data and the auditory objects that are supposed to represent them in a way that is perceptually and cognitively meaningful.

Kramer (1994) gave a first important contribution describing the role of mediating structures between the data and the listener or, in other words, of mapping. It is important to investigate the role of mediating structures between the data and the listener or, in other words, of mapping. The term audification was proposed to indicate a “direct translation of a data waveform to the audible domain for purposes of monitoring and comprehension”. Examples are found in electroencephalography, seismology and in sonar signal analysis. In sonification, instead, data are used to control a sound generation, and the generation technique is not necessarily in direct relationship to the data. For instance, we may associate pitch, loudness, and rhythm of a percussive sound source with the physical variables being read from sensors in an engine.

A major problem is how to recall the mappings. This can be done via metaphors (e.g., high pitch = up) or feelings (e.g., harsh = bad situation), and the interactions between the two. These aspects are still very hot and open for further research nowadays.

Direct mapping (Audification) The most straightforward kind of mapping is the one that takes the data to feed the digital-to-analog converters directly, thus playing back the data at an audio sampling rate. This can be of some effectiveness only if the data are temporal series, as it is the case in seismology.

The idea of listening to the data produced by seismograms to seek relevant phenomena and improve understanding is quite old. If the seismic signals are properly conditioned and transposed in frequency, they sound pretty natural to our ears, and we can use our abilities in interpreting noises in everyday conditions.

One of the main motivations for using auditory display is that there are important events that are difficult to detect in visual time-series displays of noisy data, unless using complex spectral analyzes. Conversely, these events are easily detected by ear. There are several problems that have to be faced when trying to sonify seismic data, especially related with the huge dynamic range (> 100 dB) and with the frequency bandwidth which, albeit restricted below 40 Hz, spans more than 17 octaves. Many of the mentioned problems cause headaches to visual analysts as well. In order to let relevant events audible, the recorded signals have to be subject to a certain amount of processing, like gain control, time compression, frequency shift or transposition, annotation, looping, stereo placement.

Naturalistic mapping In some cases, it is possible to use natural or mechanical sounds to convey information of various kinds. This is especially effective when the information is physically related to the reference sound sources, so that our everyday physical experience can be exploited in interpreting the sounds.

Abstract mapping Sonification often implies an abstract mapping of nonacoustic events onto acoustic events. An example of abstract mapping is Geiger counters which are used to detect and sonify ionizing radiation. An inert gas-filled tube (usually helium, neon or argon with halogens added) briefly conducts



electricity when a particle or photon of radiation makes the gas conductive. The tube amplifies this conduction by a cascade effect and outputs a current pulse, which is then often displayed as audible clicks.

A good mapping can be the key to demonstrate the superiority of auditory over other forms of display for certain applications. Indeed, researchers in Sonification and Auditory Display have long been looking for the killer application for their findings and intuitions. This is especially difficult if the data are not immediately associable with sound objects, and abstract mappings have to be devised, as for example stock market data.

Musical mapping Music has its own laws and organizing principles, but sometimes these can be bent to follow flows of data.

11.5 Sonification

Sonification can be considered as the auditory equivalent of graphic representation in the visual domain. The main goal of sonification is to define a way for representing reality by means of sound. Scaletti (1994) proposed a working definition of sonification as “a mapping of numerically represented relations in some domain under study to relations in an acoustic domain for the purpose of interpreting, understanding, or communicating relations in the domain under study.”

Another less restrictive interpretation of sonification is found by transposition of what people currently intend with the word visualization.

11.5.1 Information Sound Spaces (ISS)

In his thesis work, Stephen Barras aims at defining a methodology for representing information by means of sonification processes. The initial motivation of Barrass' work could be summarized in the following quotation: “The computer-based workplace is unnaturally quiet...and disquietingly unnatural...”. In other words, the starting point of his work was the problem of the development of auditory displays for the computer. The first goal becomes, thus, to solve the contrast between the informative soundscape of the everyday world and the silence of the computer-based workplace. On the other side the danger is that a “noisy/musical” computer could easily become an annoying element. This concern, according to Barrass, highlights the need to design useful but not intruding/obsessive sounds.

More into detail, his thesis addresses the problems pointed out by previous researchers in the field of auditory display, as:

- The definition of a method for evaluating the usefulness of the sounds for a specific activity;
- The definition of methods for an effective representation of data relations by means of sounds;
- The achievement of a psychoacoustic control of auditory displays;
- The development of computer aided tools for auditory information design.

Barrass illustrates a set of already existing approaches to auditory display design. A possible classification of these approaches is:

- Syntactic and grammar-based (eg. Morse code, Earcons);
- Pragmatic: materials, lexicon and/or palette;

- Semantic: the sound is semantically related to what is meant to represent. In particular, the semantic relationships can be subdivided in:
 - Symbolic: the signifier does not resemble the signified;
 - Indexical: the signified is causally related to the signifier (e.g. the sound of a tennis ball);
 - Iconical: the signifier resembles the signified that is the case of a picture/a photograph.

Barrass proposes different approaches for auditory display design. Among the others, a Pragmatic approach and a Task-oriented approach are discussed. The Pragmatic approach concerns design principles of warnings and alarms (see also Section 9.4.1). A set of rules are asserted as:

1. Use two stages signals a) attention demanding b) designation signal;
2. Use interrupted or variable signals;
3. Use modulated signals;
4. Do not provoke startling;
5. Do not overload the auditory channel.

A Task-oriented approach takes a particular role in the following developments of the Thesis, in terms of Sound Design for Information display. Task analysis is a method developed in Human-Computer Interaction (HCI) design to analyze and characterize the information required in order to manipulate events, modes, objects and other aspects of user interfaces. The methodology is based on Task analysis and Data characterization (TaDa). According to this analysis, the information requirements necessary for an information representation on a certain display addressing a specific kind of user are defined. One of the possible strategies to take into consideration the user from the very first step of the design process is to use a story to describe a problem. The tools become storyboards, scenarios, interviews, and case studies.

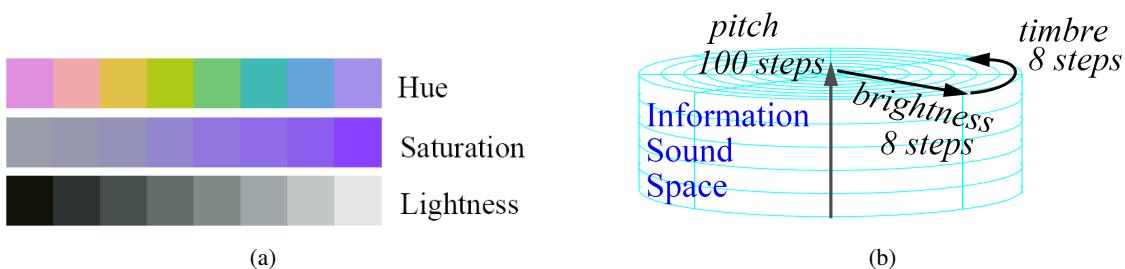


Figure 11.1: left: 8 steps in hue, saturation and lightness; right: The TBP prototype of an Information-Sound Space (ISS).

Another important part of Barrass' thesis is the definition of an Information-Sound Space, what he calls a cognitive artefact for auditory design. Barrass starts from the example of the Hue, Saturation, Brightness (HSB) model for the representation of the “color space” and from its representation of a color choosing tool by means of a circle with the hues corresponding to different sectors, the saturation levels mapped along the rays and the brightness controlled by means of a separated slider.

The ISS (Information Sound –Space) representation is a cylinder with the following dimensions:



- categorical (not ordered) organization of the information (the sectors of the circle)
- perceptual metric (ordered) along the radial spokes
- perceptual metric along vertical axle

The three dimensions of the ISS are related to timbre, brightness and pitch, the brightness corresponding to the radial dimension and the pitch to the height dimension.

11.5.2 Interactive Sonification

Interactive Sonification acknowledges the importance of human interaction for understanding and using auditory feedback. Interaction allows users to continuously query different "auditory views" of objects in the case of sonification the data under analysis) which help to give a more complete overview.

For example Rath and Rocchesso (2005) explain how continuous sonic feedback made by physical models can be used in HCI. The control metaphor which is used to demonstrate this statement is balancing a ball along a tiltable track. The idea of using a continuous feedback comes out by simply analyzing the natural behavior: trigger sounds are less usual, while we often refer to continuous sounds to get information about what is happening around us. Sonic feedback has the advantage that it can help us without changing our focus of attention: the audio channel can improve the effectiveness and naturalness of the interaction.

The physical model of a rolling ball is used: this sound is particularly informative, conveying information about direction, velocity, shape and surface textures of the contacting objects. The main characteristic of this model is its reactivity and dynamic behavior: "the impact model used produces complex transients that depend on the parameters of the interaction and the instantaneous states of the contacting objects". The physics based algorithm developed in this work involves a degree of simplification and abstraction that implies efficient implementation and control. The approach is the cartoonification of sounds: a simplification of the models aimed at retaining perceptual invariants, useful for an efficient interaction, instead of producing the exact replica of the original sounds.

This investigation suggests that continuous feedback of a carefully designed sound can be used for sensory substitution of haptic or visual feedback in embodied interfaces. Many multimodal contexts can benefit from cartoon sound models to improve the effectiveness of the interaction: video games and virtual environments are the more obvious ones.

11.6 Interactive sounds

Most of Virtual Reality (VR) applications built to date make use of visual displays, haptic devices, and spatialized sound displays. Multisensory information is essential for designing immersive virtual worlds, as an individual's perceptual experience is influenced by interactions among sensory modalities. As an example, in real environments visual information can alter the haptic perception of object size, orientation, and shape. Similarly, being able to hear sounds of objects in an environment, while touching and manipulating them, provides a sense of immersion in the environment not obtainable otherwise. Properly designed and synchronized haptic and auditory displays are likely to provide much greater immersion in a virtual environment than a high-fidelity visual display alone. Moreover, by skewing the relationship between the haptic and visual and/or auditory displays, the range of object properties that can be effectively conveyed to the user can be significantly enhanced.

The importance of multimodal feedback in computer graphics and interaction has been recognized for a long time and is motivated by our daily interaction with the world. Streams of information coming from different channels complement and integrate each other, with some modality possibly dominating



over the remaining ones, depending on the task. Research in ecological acoustics demonstrates that auditory feedback in particular can effectively convey information about a number of attributes of vibrating objects, such as material, shape, size, and so on (see also Chapter 9.4).

11.6.1 Ecological acoustics

Perception refers to how animals, including humans, can be aware of their surroundings. Perception involves motions of receptor systems (often including the whole body), and action involves motion of effectors (often including the whole body). Thus, the perception and control of behavior is largely equivalent to the perception and control of motion. Movements are controlled and stabilized relative to some referents. To watch tennis, the eyes must be stabilized relative to the moving ball. To stand, the body must be stabilized relative to the gravito-inertial force environment. Action and perception can be controlled relative to a myriad of referents. We must select referents for the control of action. The selection of referents should have a functional basis, that is, it should depend on the goals of action (e.g., a pilot who controls orientation relative to the ground may lose aerodynamic control, and a pilot who controls navigation relative to gravito-inertial force will get lost). One aspect of learning to perform new tasks will be the determination of which referents are relevant.

The ecological approach to perception, originated in the work of Gibson, refers to a particular idea of how perception works and how it should be studied. The label “ecological” reflects two main themes that distinguish this approach from more established views. First, perception is an achievement of animal-environment systems, not simply animals (or their brains). What makes up the environment of a particular animal is part of this theory of perception. Second, the main purpose of perception is to guide action, so a theory of perception cannot ignore what animals do. The kinds of activities that a particular animal does, e.g. how it eats and moves, are part of this theory of perception.

11.6.1.1 The ecological approach to perception

Direct versus indirect perception

The ecological approach is considered controversial because of one central claim: perception is direct. To understand the claim we can contrast it with the more traditional view.

Roughly speaking, the classical theory of perception states that perception and motor control depend upon internal referents, such as the retina for vision and cochlea for audition. These internal, psychological referents for the description and control of motion are known as sensory reference frames. Sensory reference frames are necessary if sensory stimulation is ambiguous (i.e., impoverished) with respect to external reality; in this case, our position and motion relative to the physical world cannot be perceived *directly*, but can only be derived *indirectly* from motion relative to sensory reference frames. Motion relative to sensory reference frames often differs from motion relative to physical reference frames (e.g., if the eye is moving relative to the external environment). For this reason, sensory reference frames provide only an indirect relation to physical reference frames. For example, when objects in the world reflect light, the pattern of light that reaches the back of the eye (the retina) has lost and distorted a lot of detail. The role of perception is then fixing the input and adding meaningful interpretations to it so that the brain can make an inference about what caused that input in the first place. This means that accuracy depends on the perceiver’s ability to “fill in the gaps” between motion defined relative to sensory reference frames and motion defined relative to physical reference frames, and this process requires inferential cognitive processing.

A theory of *direct* perception, in contrast, argues that sensory stimulation is determined in such a way that there exists a 1:1 correspondence between patterns of sensory stimulation and the underlying aspects of physical reality. This is a very strong assumption, since it basically says that reality is fully specified



in the available sensory stimulation. Gibson provides the following example in the domain of visual perception, which supports, in his opinion, the direct perception theory. If one assumes that objects are isolated points in otherwise empty space, then their distances on a line projecting to the eye cannot be discriminated, as they stimulate the same retinal location. Under this assumption it is correct to state that distance is not perceivable by eye alone. However Gibson argues that this formulation is inappropriate for describing how we see. Instead he emphasizes that the presence of a continuous background surface provides rich visual structure.

Including the environment and activity into the theory of perception allows a better description of the input, a description that shows the input to be richly structured by the environment and the animal's own activities. According to Gibson, this realization opens up the new possibility that perception might be *veridical*. A relevant consequence of the direct perception approach is that sensory reference frames are unnecessary: if perception is direct, then anything that can be perceived can also be measured in the physical world.

Energy flows and invariants

Consider the following problem in visual perception: how can a perceiver distinguish object motion from his or her own motion? Gibson provides an ecological solution to this problem, from which some general concepts can be introduced. The solution goes as follows: since the retinal input is ambiguous, it must be compared with other input. A first example of additional input is the information on whether any muscle commands had been issued to move the eyes or the head or the legs. If no counter-acting motor commands are detected, then object motion can be concluded; on the contrary, if such motor commands are present then this will allow the alternative conclusion of self-motion. When the observer is moved passively (e.g. in a train), other input must be taken into account: an overall (global) change in the pattern of light indicates self-motion, while a local change against a stationary background indicates object motion.

This argument opened a new field of research devoted to the study of the structure in changing patterns of light at a given point of observation: the *optic flow*. The goal of this research is to discover particular patterns, called *invariants*, which are relevant to perception and hence to action of an animal immersed in an environment. Perceivers exploit invariants in the optic flow, in order to effectively guide their activities. For example: a waiter, who rushes towards the swinging door of the restaurant kitchen, adjusts his motion in order to control the collision with the door: he maintains enough speed to push through the door, and the same time he is slow enough not to hurt himself. In order for his motion to be effective he must know when a collision will happen and how hard the collision will be. One can identify structures in the optic flow that are relevant to these facts: these are examples of quantitative invariants.

The above considerations apply not only to visual perception but also to other senses, including audition (see Section 9.6.2 next). Moreover, recent research has introduced the concept of *global array*. According to this concept, individual forms of energy (such as optic or acoustic flows) are subordinate components of a higher-order entity, the global array, which consists of spatio-temporal structure that extends across many dimensions of energy. The general claim underlying this concept is that observers are not separately sensitive to structures in the optic and acoustic flows but, rather, observers are directly sensitive to patterns that extend across these flows, that is, to patterns in the global array.

Affordances

The most radical contribution of Gibson's theory is probably the notion of *affordance*. Gibson uses the term affordance as the noun form of the verb "to afford". The environment of a given animal affords things for that animal. What kinds of things are afforded? The answer is that behaviors are afforded. A



stair with a certain proportion of a person's leg length affords climbing (is climbable); a surface which is rigid relative to the weight of an animal affords stance and traversal (is traversable); a ball which is falling with a certain velocity, relative to the speed that a person can generate in running toward it, affords catching (is catchable), and so on. Therefore, affordances are the possibilities for action of a particular animal-environment setting; they are usually described as “-ables”, as in the examples above. What is important is that affordances are not determined by absolute properties of objects and environment, but depend on how these relate to the characteristics of a particular animal, e.g. size, agility, style of locomotion, and so on.

The variety of affordances constitute ecological reformulations of the traditional problems of size, distance, and shape perception. Note that affordances and events are not identical and, moreover, that they differ from one another in a qualitative manner. Events are defined without respect to the animal, and they do not refer to behavior. Instead, affordances are defined relative to the animal and refer to behavior (i.e., they are animal-environment relations that afford some behavior). The concept of affordance thus emphasizes the relevance of activity to defining the environment to be perceived.

11.6.2 Everyday sounds and the acoustic array

Ecological psychology has traditionally concentrated on visual perception. There is now interest in auditory perception and in the study of the *acoustic array*, the auditory equivalent of the optic array.

The majority of the studies in this field deal with the perception of properties of environment, objects, surfaces, and their changing relations, which is a major thread in the development of ecological psychology in general. In all of this research, there is an assumption that properties of objects, surfaces, and events are perceived as such. Therefore studies in audition investigate the identification of sound source properties, such as material, size, shape, and so on.

Musical listening versus everyday listening

Gaver introduces the concept of *everyday listening*, as opposed to *musical listening*. When a listener hears a sound, he might concentrate on attributes like pitch, loudness, and timbre, and their variations over time. Or he might notice that its masking effect on other sounds. Gaver refers to these as examples of *musical listening*, meaning that the considered perceptual dimensions and attributes have to do with the sound itself, and are those used in the creation of music.

On the other hand, the listener might concentrate on the characteristics of the sound source. As an example, if the sound is emitted by a car engine the listener might notice that the engine is powerful, that the car is approaching quickly from behind, or even that the road is a narrow alley with echoing walls on each side. Gaver refers to this as an example of *everyday listening*, the experience of listening to events rather than sounds. In this case the perceptual dimensions and attributes have to do with the sound-producing event and its environment, rather than the sound itself.

Everyday listening is not well understood by traditional approaches to audition, although it forms most of our experience of hearing the day-to-day world. Descriptions of sound in traditional psychoacoustics are typically based on Fourier analysis and include frequency, amplitude, phase, and duration. Traditional psychoacoustics takes these “primitive” parameters as the main dimensions of sound and tries to map them into corresponding “elemental” sensations (e.g., the correspondence between sound amplitude and perceived loudness, or between frequency and perceived pitch). This kind of approach does not consider higher-level structures that are informative about events.

Everyday listening needs a different theoretical framework, in order to understand listening and manipulate sounds along source-related dimensions instead of sound-related dimensions. Such a framework must answer two fundamental questions. First, it has to develop an account of ecologically relevant per-



ceptual attributes, i.e. the features of events that are conveyed through listening. Thus the first question asked by Gaver is: "What do we hear?". Second, it has to develop an ecological acoustics, that describes which acoustic properties of sounds are related to information about the sound sources. Thus the second question asked by Gaver is: "How do we hear it?"

Acoustic flow and acoustic invariants

Any source of sound involves an interaction of materials. Let us go back to the above example of hearing an approaching car: part of the energy produced in the engine produces vibrations in the car, instead of contributing to its motion. Mechanical vibrations, in turn, produce waves of acoustic pressure in the air surrounding the car, where the waveforms follows the movement of the car's surfaces (within limits determined by the frequency-dependent coupling of the surface's vibrations to the medium). These pressure waves then contain information about the vibrations that caused them, and result in a sound signal from which a listener might obtain such information. More in general, the patterns of vibration produced by contacting materials depend both on contact forces, duration of contact, and time-variations of the interaction, as well as sizes, shapes, materials, and textures of the objects.

Sound also conveys information about the environment in which the event have occurred. In everyday conditions, a listener's ear is reached not only by the direct sound but also by the reflections of sound over various other objects in the environment, resulting in a coloration of the spectrum. In addition, the transmitting medium also has an influence on sound signals: dissipation of energy, especially at high-frequency, increases with the path travelled by the sound waves and thus carries information about the distance of the source. Another example is Doppler effect, which is produced when sound sources and listeners in relative motion, and results in a shift of the frequencies. Changes in loudness caused by changes in distance from a moving sound source may provide information about time-to-contact in a fashion analogous to changes in visual texture. The result is an *acoustic array*, analogous to the optical array described previously.

Several *acoustic invariants* can be associated to sound events: for instance, several attributes of a vibrating solid, including its size, shape, and density, determines the frequencies of sound it produces. It is quite obvious that a single physical parameters can influence simultaneously many different sound parameters. As an example, changing the size of an object will scale the sound spectrum, i.e. will change the frequencies of the sound but not their pattern. On the other hand, changing the object shape results in a change of both the frequencies and their relationships. Gaver argues that these complex patterns of change may serve as information distinguishing the physical parameters responsible: ecological acoustics focuses of discovering this kind of acoustic invariants.

Maps of everyday sounds

As already mentioned, Gaver has proposed an ecological categorization of everyday sounds.

A first category includes sounds generated by solid objects. The pattern of vibrations of a given solid is structured by a number of its physical attributes. Properties can be grouped in terms of attributes of the *interaction* that has produced the vibration, those of the *material* of the vibrating objects, and those of the *geometry* and configuration of the objects.

Aerodynamic sounds are caused by the direct introduction and modification of atmospheric pressure differences from some source. The simplest aerodynamic sound is exemplified by an exploding balloon. Other aerodynamic sounds, e.g. the noise of a fan, are caused by more continuous events. Another sort of aerodynamic event involves situations in which changes in pressure themselves transmit energy to objects and set them into vibration (for example, when wind passes through a wire).



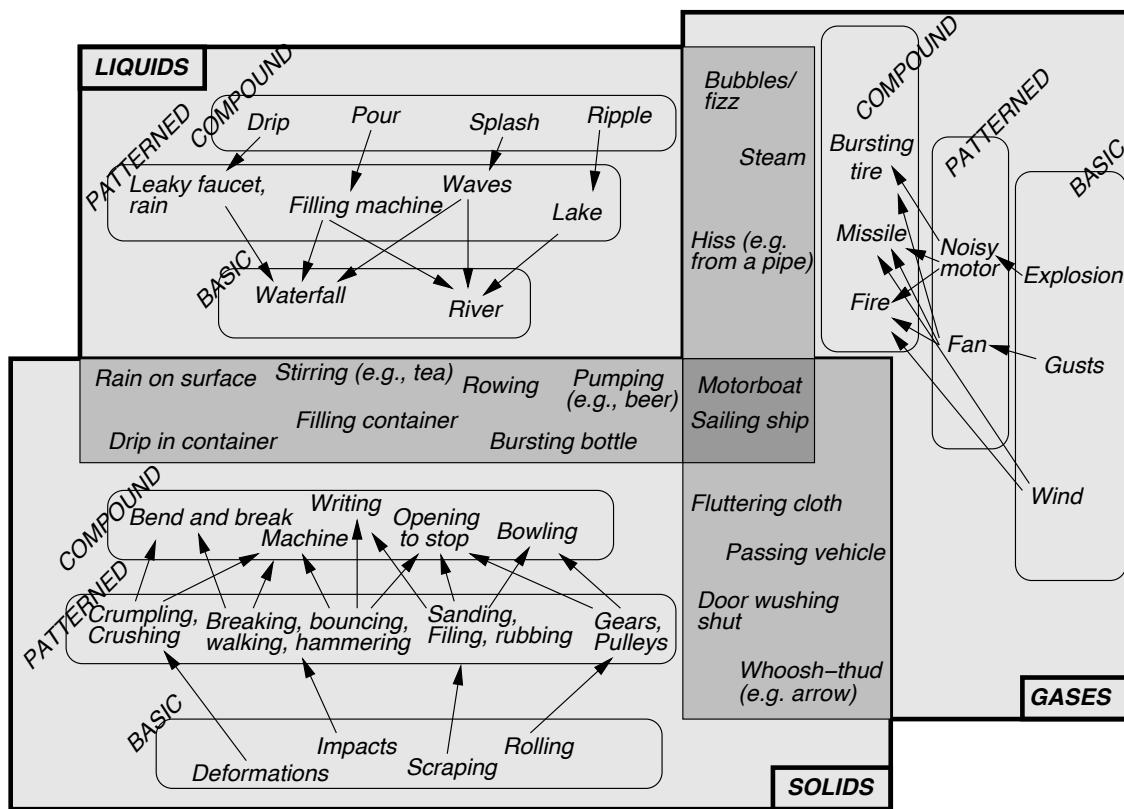


Figure 11.2: A map of everyday sounds. Complexity increases towards the center. Figure based on (Gaver, 1993).

Sound-producing events involving liquids (e.g., dripping and splashing) are similar to those of vibrating solids: they depend on an initial deformation that is counter-acted by restoring forces in the material. The difference is that no audible sound is produced by the vibrations of the liquid. Instead, the resulting sounds are created by the resonant cavities (bubbles) that form and oscillate and in the surface of the liquid. As an example, a solid object that hits a liquid pushes it aside and form a cavity that resonates to a characteristic frequency, amplifying and modifying the pressure wave formed by the impact itself.

Although all sound-producing events involve any of the above categories (vibrating solids, aerodynamic, or liquid interactions), many also depend on complex patterns of simpler events. As an example, footsteps are temporal patterns of impact sounds. The perception of these *patterned* sounds is also related to the timing of successive events, (e.g. successive footprint sounds must occur within a range of rates and regularities in order to be perceived as walking). A slightly more complex example is a door slam, which involves the squeak of scraping hinges and the impact of the door on its frame. This kind of *compound* sounds involve mutual constraints on the objects that participate in related events: concatenating the creak of a heavy door closing slowly with the slap of a light door slammed shut would probably not sound natural.

Starting from these considerations, Gaver derived a tentative map of everyday sounds, which is shown in figure 9.2 and discussed in the following.

- Basic Level Sources: consider, for example, the region describing sounds made by vibrating solids. Four different sources of vibration in solids are indicated as basic level events: deformation, impacts, scraping and rolling.

- Patterned Sources involve temporal patterning of basic events. For instance walking, as described above, but also breaking, spilling, and so on, are all complex events involving patterns of simpler impacts. Similarly, crumpling or crushing are examples of patterned deformation sounds. In addition, other sorts of information are made available by their temporal complexity. For example, the regularity of a bouncing sound provides information about the symmetry of the bouncing object.
- Compound events involve more than one type of basic level event. An example is the slamming door discussed above. Other examples are the sounds made by writing, which involve a complex series of impacts and scrapes over time, while those made by bowling involve rolling followed by impact sounds.
- Hybrid events involve yet another level of complexity in which more than one basic types of material is involved. As an example, the sounds resulting from water dripping on a reverberant surface are caused both by the surface's vibrations and the quickly-changing reverberant cavities, and thus involve attributes both of liquid and vibrating solid sounds.

11.7 Multimodal perception and interaction

11.7.1 Combining and integrating auditory information

Humans achieve robust perception through the combination and integration of information from multiple sensory modalities. According to some authors, multisensory perception emerges gradually during the first months of life, and experience significantly shapes multisensory functions. By contrast, a different line of thinking assumes that sensory systems are fused at birth, and the single senses differentiate later. Empirical findings in newborns and young children have provided evidence for both views. In general experience seems to be necessary to fully develop multisensory functions.

Sensory combination and integration

Looking at how multisensory information is combined, two general strategies can be identified: the first is to maximize information delivered from the different sensory modalities (*sensory combination*). The second strategy is to reduce the variance in the sensory estimate to increase its reliability (*sensory integration*).

Sensory combination describes interactions between sensory signals that are not redundant: they may be in different units, coordinate systems, or about complementary aspects of the same environmental property. Disambiguation and cooperation are examples for this kind of interactions: if a single modality is not enough to provide a robust estimate, information from several modalities can be combined. As an example, object recognition is achieved through different modalities that complement each other and increase the information content.

By contrast, *sensory integration* describes interactions between redundant signals. For example, when knocking on wood at least three sensory estimates about the location of the knocking event can be derived: visual, auditory and proprioceptive. In order for these three location signals to be integrated, they first have to be transformed into the same coordinates and units. For this, the visual and auditory signals have to be combined with the proprioceptive neck-muscle signals to be transformed into body coordinates. The process of sensory combination might be non-linear. At a later stage the three signals are then integrated to form a coherent percept of the location of the knocking event.

There are a number of studies that show that vision dominates the integrated percept in many tasks, while other modalities (in particular audition and touch) have a less marked influence. This phenomenon of visual dominance is often termed *visual capture*. As an example, it is known that in the spatial domain



vision can bias the perceived location of sounds whereas sounds rarely influence visual localization. One key reason for this asymmetry seems to be that vision provides more accurate location information. In general, however, the amount of cross-modal integration depends on the features to be evaluated or the tasks to be accomplished.

Auditory capture and illusions

Psychology has a long history of studying intermodal conflict and illusions in order to understand mechanisms of multisensory integration. Much of the literature on multisensory perception has focused on spatial interactions: an example is the ventriloquist effect, in which the perceived location of a sound shifts towards a visual stimulus presented at a different position. Identity interactions are also studied: an example is the already mentioned McGurk effect, in which what is being heard is influenced by what is being seen (for example, when hearing /ba/ but seeing the speaker say /ga/ the final perception may be /da/).

As already noted, the visual modality does not always win in such crossmodal tasks. In particular, the senses can interact in time, i.e they interact in determining not what is being perceived or where it is being perceived, but *when* it is being perceived. The temporal relationships between inputs from the different senses play an important role in multisensory integration. Indeed, a window of synchrony between auditory and visual events is crucial even in the spatial ventriloquist effect, which disappears when the audio-visual asynchrony exceeds approximately 300 ms. This is also the case in the McGurk effect, which fails to occur when the audio-visual asynchrony exceeds 200 – 300 ms.

There is a variety of crossmodal effects that demonstrate that, outside the spatial domain, audition can bias vision.

11.7.2 Perception is action

Embodiment and enactment

According to traditional mainstream views of perception and action, perception is a process in the brain where the perceptual system constructs an internal representation of the world, and eventually action follows as a subordinate function. This view of the relation between perception and action makes then two assumptions. First, the causal flow between perception and action is primarily one-way: perception is input from world to mind, action is output from mind to world, and thought (cognition) is the mediating process. Second, perception and action are merely instrumentally related to each other, so that each is a means to the other. If this kind of “input-output” picture is right, then it must be possible, at least in principle, to disassociate capacities for perception, action, and thought.

Although everyone agrees that perception depends on processes taking place in the brain, and that internal representations are very likely produced in the brain, more recent theories have questioned such a modular decomposition in which cognition interfaces between perception and action. The ecological approach discussed in section 9.6.1 reject the one-way assumption, but not the instrumental aspect of the traditional view, so that perception and action are seen as instrumentally interdependent. Others argue that a better alternative is to reject both assumptions: the main claim of these theories is that it is not possible to disassociate perception and action schematically, and that every kind of perception is intrinsically active and thoughtful: perception is not a process in the brain, but a kind of skilful activity on the part of the animal as a whole. Only a creature with certain kinds of bodily skills (e.g. a basic familiarity with the sensory effects of eye or hand movements, etc.) could be a perceiver.

One of the most influential contributions in this direction is due to Varela et al. (1991). They presented an “enactive conception” of experience, which is not regarded as something that occurs inside the animal, but rather as something that the animal *enacts* as it explores the environment in which it is



situated. In this view, the subject of mental states is the *embodied*, environmentally situated animal. The animal and the environment form a pair in which the two parts are coupled and reciprocally determining. Perception is thought of in terms of activity on the part of the animal. The term “embodied” is used by the authors as a mean to highlight two points: first, cognition depends upon the kinds of experience that are generated from specific sensorimotor capacities. Second, these individual sensorimotor capacities are themselves embedded in a biological, psychological, and cultural context. Sensory and motor processes, perception and action, are fundamentally inseparable in cognition.

11.8 Multimodal and Cross-Modal Approaches to Control of Interactive Systems

11.8.1 Introduction

This section⁶ briefly surveys some relevant aspects of current research into control of interactive (music) systems, putting into evidence research issues, achieved results, and problems that are still open for the future. A particular focus is on multimodal and cross-modal techniques for expressive control of sound and music processing and synthesis. The section will discuss a conceptual framework, the methodological aspects, the research perspectives.

The problem of effectively controlling sound generation and processing has always been relevant for music research in general and for Sound and Music Computing in particular. Research into control concerns perceptual, cognitive, affective aspects. It ranges from the study of the mechanisms involved in playing traditional acoustic instruments to the novel opportunities offered by modern digital music instruments. More recently, the problem of defining effective strategies for real-time control of multimodal interactive systems, with particular reference to music but not limited to it, is attracting growing interest from the scientific community because of its relevance also for future research and applications in broader fields of human-computer interaction.

In this framework, research into control extends its scope to include for example analysis of human movement and gesture (not only gestures of musicians playing an instrument but also gestures of subjects interacting with computer systems), analysis of the perceptual and cognitive mechanisms of gesture interpretation, analysis of the communication of non-verbal expressive and emotional content through gesture, multimodality and cross-modality, identification of strategies for mapping the information obtained from gesture analysis onto real-time control of sound and music output including high-level information (e.g. real-time control of expressive sound and music output).

A key issue in this research is its cross-disciplinary nature. Research can highly benefit from cross-fertilisation between scientific and technical knowledge on the one side, and art and humanities on the other side. Such need of cross-fertilisation opens new perspectives to research in both fields: if from the one side scientific and technological research can benefit from models and theories borrowed from psychology, social science, art and humanities, on the other side these disciplines can take advantage of the tools that technology can provide for their own research, i.e. for investigating the hidden subtleties of human beings at a depth that was hard to reach before. The convergence of different research communities such as musicology, computer science, computer engineering, mathematics, psychology, neuroscience, arts and humanities as well as of theoretical and empirical approaches bears witness to the need and the importance of such cross-fertilisation.

⁶adapted from Camurri, De Poli, Leman and Volpe / IEEE Multimedia 2005 and from Antonio Camurri, Carlo Drioli, Barbara Mazzarino, Gualtiero Volpe, Polotti and Rocchesso [2008], chapt. 6



11.8.2 Multisensory Integrated Expressive Environments

Multisensory Integrated Expressive Environments (MIEEs) are a framework for mixed reality applications in the performing arts, culture-oriented applications, and future applications such as home entertainment, therapy, and rehabilitation. Paradigmatic contexts for applications of MIEEs are multimedia concerts, interactive dance/music/video installations, interactive museum exhibitions, and distributed cooperative environments for theatre and artistic expression.

Imagine a home high-fidelity (hi-fi) music system that not only has the standard controls for volume, treble, bass, balance, and so forth, but also features *expressive knobs* possibly controlled by your movement, such as dancing, in your living room. The system lets you actively listen to, say, a Chopin piece, by changing the agogics, that is, the music interpretation. For example, light and smooth movements might influence a more intimate legato phrasing in the music performance, while jumpy and joyful movements might change the phrasing to a faster tempo and staccato.

MIEEs address the expressive aspects of non-verbal human communication. In the above example, we mentioned a few terms from music performance theory (such as staccato and legato, as well as adjectives such as light, heavy, joyful, and jumpy) which are often used in humanistic theories of the performing arts. In MIEEs, real and virtual subjects interact with each other through the exchange of information that represents the communicative expressiveness in different sensory modalities (auditory, visual, tactile, and so on). The main goal of MIEEs is to establish a framework that accounts for the relationship between the current state-of-the-art in audio-visual technology on the one hand and humanistic theories on expressive actions and aesthetic experience on the other hand.

For the artistic aspect, MIEEs provide a deeper and enhanced experience of the artistic content. In the music research community, for example, MIEEs are sometimes conceived of as a new generation of musical instruments based on real-time and intelligent human-machine interaction. These new musical instruments are a holistic human-machine concept based on an assembly of modular input/output devices and musical software components arranged according to essential human musical content processing capabilities. The focus on technology and multisensory expressive content processing adds a new dimension to the mediation of art. MIEEs introduce a level of cross-modality that interconnects microscopic scales of information processing with human capabilities in synesthesia (multimodal perception of features of objects) and kinesthesia (perception of movement).

The main motivation in introducing MIEEs is to adapt the new mediating technology to basic human forms of communication. Many typical communication modes are nonlinguistic and based on movement, action, gestures, and mimetic activities. Unfortunately, the current state of the art in MIEEs suffers from a serious lack of advanced content processing capabilities in the cognitive, affective/emotive, and motoric domains. For example, although advances have been made in the processing of musical pitch, timbre, texture, and rhythm, the results are mainly restricted to low-level features. We can say the same about movement gestures. It is difficult to characterize a musical object, or to specify a movement gesture characteristics. If MIEEs have to interact intelligently and spontaneously with users, then their communication capabilities should rely on a set of advanced musical and gestural content processing tools. In many situations, users might want to interact in a spontaneous and expressive way with these systems, using descriptions of perceived qualities or making expressive movements. Moreover, making machines useful in artistic contexts that rely on different sensory modalities implies that the often subtle nuances of artistic expression should be dealt with in these different modalities. A technology focusing on affect, emotion, expressiveness, and cross-modality interactions is thus required.



11.8.3 Cross-modal expressiveness

This section briefly surveys some relevant aspects of current research on control, putting into evidence research issues, achieved results, and problems that are still open for the future. A particular focus is on multimodal and cross-modal techniques for expressive control. Multimodal analysis enables the integrated analysis of information coming from different multimedia streams (audio, video) and affecting different sensorial modalities (auditory, visual). Cross-modal analysis enables exploiting potential similarities in the approach for analyzing different multimedia streams: so, for example techniques developed for analysis in a given modality (e.g. audio) can also be used for analysis in another modality (e.g. video); further, commonalities at mid-and high-level in representations of different sensory channels are an important perspective for developing models for control and mapping based on a-modal, converging representations.

The physical stimuli that make up an artistic environment contain information about expressiveness. That information can, to some extent, be extracted and then communicated among a MIEE's virtual and real subjects. With multiple sensory modalities (auditory, visual, motoric, gestural), this allows the transmission of expressiveness parameters from one domain to another - for example, from music (auditory) to computer animation (visual), or from dance (motoric) to music (auditory). That is, expressive parameters are an example of parameters emerging from modalities and independent from them. In other words, expressive parameters define a cross-modal control space that is at a higher level with respect to the single modalities.

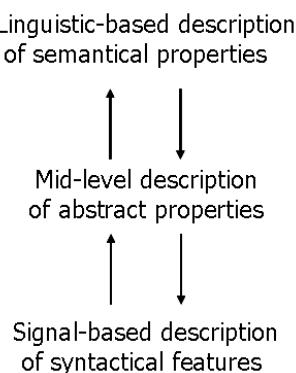


Figure 11.3: The layered conceptual framework distinguishes between syntax and semantics, and in between, a connection layer that consists of affect/emotion expressiveness spaces and mappings.

Figure 9.3 shows a way of conceiving the transmission of cross-modal expressiveness. This layered framework will be analysed more in detail in Section 9.8.4. The signal-based level represents the analysis and synthesis of physical properties (bottom). The symbolic level represents the descriptions of meanings, affects, emotions, and expressiveness in terms of linguistic or visual symbolic entities (top). The connection layer (gesture-based level) represents spaces in which trajectories allow the connection from signal-based descriptions to symbolic-based descriptions (middle). In most cases, the signal-based descriptions pertain to the signal syntactical properties, while the symbol-based descriptions pertain to its semantic properties. The latter may include cognitive, emotive, affective, and expressive evaluations.

The flow of expressiveness may go in two directions (upward and downward). Taking the expressive hi-fi music system as an example, physical properties of human movement (of the system user) may be extracted and gestures mapped as a trajectory on a space. That trajectory describes the expressive content in terms of linguistic/semantic descriptors such as how much the movement is fluent, smooth, heavy, rigid, and so on. Starting from this linguistic-semantic description (in the downward direction),



a particular gesture-based trajectory may be used to synthesize physical properties of that expressive content in another modality, in our case music (in terms of legato/staccato, amplitude, shapes of notes, and so on). The gesture-based mappings describe properties of expressiveness that are independent from any particular sensory modality. This level of mapping introduces flexibility to the multimodal representational model.

11.8.4 A conceptual framework

A relevant foundational aspect for research in sound and music control concerns the definition of a conceptual framework envisaging control at different levels under a multimodal and cross-modal perspective: from low-level analysis of audio signals, toward high-level semantic descriptions including affective, emotional content. This Section presents a conceptual framework worked out in the EU-IST Project MEGA (2000-2003)⁷ that can be considered as a starting point for research on this direction.

High-level expressive information: e.g., recognized emotions (e.g., anger, fear, grief, joy); prediction of spectators' intensity of emotional experience.



Semantic and narrative descriptions

Modelling techniques (for example, classification in terms of basic emotions, or prediction of intense emotional experience in spectators): e.g., based on multiple regression, neural networks, support vector machines, decision trees, Bayesian networks.



Segmented gestures and related parameters (e.g., absolute and relative durations), trajectories representing gestures in semantic spaces.



Gesture-based representations

Techniques for gesture segmentation: motion segmentation (e.g., in pause and motion phases), segmentation of musical excerpts in musical phrases. Representation of gestures as trajectories in semantic spaces (e.g., Laban's Effort space, energy-articulation space)



Motion and audio descriptors: e.g., amount of energy - loudness, amount of contraction/expansion - spectral width and melodic contour, low fluency - roughness etc.



Signal level representation

Analysis of video and audio signals: techniques for background subtraction, motion detection, motion tracking (e.g., techniques for colour tracking, optical flow based feature tracking), techniques for audio pre-processing and filtering, signal conditioning.



Data from several kinds of sensors, e.g., images from videocameras, positions from localization systems, data from accelerometers, sampled audio, MIDI messages.

Figure 11.4: The layered conceptual framework makes a distinction between syntax and semantics, and in between, a connection layer that consists of affect / emotion / expressiveness (AEE) spaces and mappings.

A main question thus relates to the nature of the physical cues that carry expressiveness, and a sec-

⁷www.megaproject.org



ond question is how to set up cross-modal interchanges (as well as person/machine interchanges) of expressiveness. These questions necessitated the development of a layered conceptual framework for affect processing that splits up the problem into different sub-problems. The conceptual framework aims at clarifying the possible links between physical properties of a particular modality, and the affective/emotive/expressive (AEE) meaning that is typically associated with these properties. Figure 9.4 sketches the conceptual framework in terms of

- a syntactical layer that stands for the analysis and synthesis of physical properties (bottom),
- a semantic layer that contains descriptions of affects, emotions, and expressiveness (top),
- a layer of AEE mappings and spaces that link the syntactical layer with the semantic layer (middle).

The syntactical layer contains different modalities, in particular audio, movement, and animation and arrows point to flows of information. Communication of expressiveness in the cross-modal sense could work in the following way. First, (in the upward direction) physical properties of the musical audio are extracted and the mapping onto an AEE-space allows the description of the affective content in the semantic layer. Starting from this description (in the downward direction), a particular AEE-mapping may be selected that is then used to synthesise physical properties of that affect in another modality, such as animation. This path is followed, for example, when sadness is expressed in a piece of music, and correspondingly an avatar is displaying this sadness in his posture.

Structure		Concept level		Musical content features				
Contextual	Global beyond 3 seconds	High II	Expressive	Cognition	Emotion	Affect = Syntactic + semantic concepts		
				Melody	Harmony	Rhythm	Source	Dynamics
		High I	Formal	Key Profile	Tonality Cadence	Rhythmic patterns Tempo	Instrument Voice	Trajectory Articulation
	Global < 3 seconds	Mid	Perceptual	Successive intervallic pattern	Simultaneous intervallic pattern	Beat Inter-onset interval	Spectral envelope	Dynamic range Sound level
				Pattern		Time	Timbre	Loudness
	Local + spatial	Low II	Sensorial	Periodicity pitch Pitch deviations Fundamental frequency		Note duration Onset Offset	Roughness Spectral flux Spectral centroid	Neural energy Peak
Non contextual	Local + temporal	Low I	Physical	Frequency		Duration	Spectrum	Intensity

Figure 11.5: Taxonomy of musical syntactical cues.

11.8.4.1 Syntactic layer

The syntactic layer is about the extraction of the physical features that are relevant for affect, emotion and expressiveness processing. In the domain of musical audio processing, Lesaffre and colleagues



worked out a useful taxonomy of concepts that gives a structured understanding of this layer in terms of a number of justified distinctions (Figure 9.5). A distinction is made between low-level, mid-level, and high-level descriptors of musical signals. In this viewpoint, the low-level features are related to very local temporal and spatial characteristics of sound. They deal with the physical categories of frequency, duration, spectrum, intensity, and with the perceptual categories of pitch, time, timbre, and perceived loudness. Low-level features are extracted and processed (in the statistical sense) in order to carry out a subsequent analysis related to expression. For example, in the audio domain, these low-level features are related to tempo (i.e. number of beats per minute), tempo variability, sound level, sound level variability, spectral shape (which is related to the timbre characteristics of the sound), articulation (features such as legato, staccato), articulation variability, attack velocity (which is related to the onset characteristics which can be fast or slow), pitch, pitch density, degree of accent on structural important notes, periodicity, dynamics (intensity), roughness (or sensory dissonance), tonal tension (or the correlation between local pitch patterns and global or contextual pitch patterns), and so on.

When more context information is involved (typically in musical sequences that are longer than 3 seconds), then other categories emerge, in particular, categories related to melody, harmony, rhythm, source, and dynamics. Each of these categories has several distinct specifications, related to an increasing complexity, increasing use of contextual information, and increasing use of top-down knowledge. The highest category is called the expressive category. This layer can in fact be developed into a separate layer because it involves affective, emotive and expressive meanings that cannot be directly extracted from audio structures. Figure 9.4 introduced this layer as a separate layer that is connected with the syntactical cues using a middle layer of mappings and spaces. Examples of mappings and spaces will be given below.

In the domain of movement (dance) analysis, a similar approach can be envisaged that leans on a distinction between features calculated on different time scales. In this context also, it makes sense to distinguish between (i) low-level features, calculated on a time interval of a few milliseconds (e.g. one or a few frames coming from a video camera), (ii) mid-level features, calculated on a movement stroke (in the following also referred as "motion phase"), i.e. on time durations of a few seconds, and (iii) high-level features that are related to the conveyed expressive content (but also to cognitive aspects) and referring to sequences of movement strokes or motion (and pause) phases. An example of a low-level feature is the amount of contraction/expansion that can be calculated on just one frame, i.e. on 40 ms with the common sample rate of 25 fps. Other examples of low-level features are the detected amount of movement, kinematic measures (e.g. velocity and acceleration of body parts), measures related to the occupation of the space surrounding the body. Examples of mid-level descriptors are the overall direction of the movement in the stroke (e.g. upward or downward) or its directness (i.e. how much the movement followed direct paths), motion impulsiveness, and fluency. At this level it is possible to obtain a first segmentation of movement in strokes that can be employed for developing an event-based representation of movement. In fact, strokes or motion phases can be characterised by a beginning, an end, and a collection of descriptors including both mid-level features calculated on the stroke and statistical summaries (e.g. average, standard deviation), performed on the stroke, of low-level features (e.g. average body contraction/expansion during the stroke).

11.8.4.2 Semantic layer

The semantic layer is about the experienced meaning of affective, emotive, expressive processing. Apart from aesthetic theories of affect processing in music and in dance, experimental studies were set up that aim at depicting the underlying structure of affect attribution in performing arts (see next sections). Affect semantics in music has been studied by allowing a large number of listeners to use adjectives (either on a completely free basis, or taken from an elaborate list) to specify the affective content of



musical excerpts. Afterwards, the data are analysed and clustered into categories. There seems to be a considerable agreement about two fundamental dimensions of musical affect processing, namely Valence and Activity. Valence is about positively or negatively valued affects, while Activity is about the force of these affects. A third dimension is often noticed, but its meaning is less clearly specified. These results provided the basis for the experiments performed along the project.

11.8.4.3 Connecting syntax and semantics: Maps and spaces

Different types of maps and spaces can be considered for connecting syntax and semantics. One type is called the semantic map because it relates the meaning of affective/emotive/expressive concepts with physical cues of a certain modality. In the domain of music, for example, several cues have been identified and related to affect processing. For example, tempo is considered to be the most important factor affecting emotional expression in music. Fast tempo is associated with various expressions of activity/excitement, happiness, potency, anger and fear while slow tempo with various expressions of sadness, calmness, solemnity, dignity. Loud music may be determinant for the perception of expressions of intensity, power, anger and joy whereas soft music may be associated with tenderness, sadness, solemnity, and fear. High pitch may be associated with expressions such as happy, graceful, exciting, angry, fearful and active, and low pitch may suggest sadness, dignity, excitement as well as boredom and pleasantness, and so on. Kinematics spaces or energy-velocity spaces are another important type of space. They have been successfully used for the analysis and synthesis of the musical performance. This space is derived from factor analysis of perceptual evaluation of different expressive music performances. Listeners tend to use these coordinates as mid level evaluation criteria. The most evident correlation of energy-velocity dimensions with syntactical features is legato-staccato versus tempo. The robustness of this space is confirmed in the synthesis of different and varying expressive intentions in a musical performance. The MIDI parameters typically control tempo and key velocity. The audio-parameters control legato, loudness, brightness, attack time, vibrato, and envelope shape.

11.8.5 Methodologies of gesture analysis

The definition of suitable scientific methodologies for investigating - within the conceptual framework and under a multimodal perspective - the subtleties involved in sound and music control is a key issue. An important topic for control research is gesture analysis of both performers and interacting subjects. Gestures are an easy and natural way for controlling sound generation and processing. For these reasons, this section discusses methodologies and approaches focusing on full-body movement and gesture. Nevertheless, the concepts here discussed can be easily generalised to include other modalities. Discovering the key factors that characterise gesture, and in particular expressive gesture, in a general framework is a challenging task. When considering such an unstructured scenario one often has to face the problem of the poor or noisy characterisation of most movements in terms of expressive content. Thus, a common approach consists in starting research from a constrained framework where expressiveness in movement can be exploited to its maximum extent.

11.8.5.1 Bottom-up approach

Let us consider the dance scenario (consider, however, that what we are going to say also applies to music performance). A possible methodology for designing repeatable experiments is to have a dancer performing a series of dance movements (choreographies) that are distinguished by their expressive content. We use the term "micro-dance" for a short fragment of choreography having a typical duration in the range of 15-90 s. A microdance is conceived as a potential carrier of expressive information, and it is not strongly related to a given emotion (i.e. the choreography has no explicit gestures denoting emotional



states). Therefore, different performances of the same micro-dance can convey different expressive or emotional content to spectators: e.g. light/heavy, fluent/rigid, happy/sad, emotional engagement, or evoked emotional strength. Human testers/spectators judge each micro-dance performance. Spectators' ratings are used for evaluation and compared with the output of developed computational models (e.g. for the analysis of expressiveness). Moreover, micro-dances can also be used for testing feature extraction algorithms by comparing the outputs of the algorithms with spectators' ratings of the same micro-dance performance (see for example the work by Camurri et al. (2004b) on spectators' expectation with respect to the motion of the body center of gravity). In case of music performances, we have musical phrases (corresponding to micro-dances above) and the same approach can be applied.

11.8.5.2 Subtractive approach

Micro-dances can be useful to isolate factors related to expressiveness and to help in providing experimental evidence with respect to the cues that choreographers and psychologists identified. This is obtained by the analysis of differences and invariants in the same micro-dance performed with different expressive intentions. With the same goal, another approach is based on the live observation of genuinely artistic performances, and their corresponding audiovisual recordings. A reference archive of artistic performances has to be carefully defined for this method, chosen after a strict intensive interaction with composers and performers. Image (audio) processing techniques are utilised to gradually subtract information from the recordings. For example, parts of the dancer's body could be progressively hidden until only a set of moving points remain, deforming filters could be applied (e.g. blur), the frame rate could be slowed down, etc. Each time information is reduced, spectators are asked to rate the intensity of their emotional engagement in a scale ranging from negative to positive values (a negative value meaning that the video fragment would rise some negative feeling in the spectator). The transitions between positive and negatives ratings and a zero-rating (i.e. no expressiveness was found by the spectator in the analysed video sequence) would help to identify what are the movement features carrying expressive information. An intensive interaction is needed between the image processing phase (i.e. the decisions on which information has to be subtracted) and the rating phase.

11.8.6 Examples of multimodal and cross-modal analysis

Here we provide some concrete examples of multimodal and cross-modal analysis with reference to the above mentioned conceptual framework. Multi-modal and cross-modal analysis can be applied both in a bottom-up approach and in a subtractive approach. In the latter, they are used for extracting and comparing features among subsequent subtraction steps.

11.8.6.1 Analysis of human full-body movement

A major step in multimodal analysis of human full-body movement is the extraction of a collection of motion descriptors. With respect to the approaches discussed above, such descriptors can be used in the bottom-up approach for characterizing motion (e.g. micro-dances). The top-down approach can be used for validating the descriptors with respect to their role and contribute in conveying expressive content.

With respect to the conceptual framework, at Layer 1 consolidated computer vision techniques (e.g. background subtraction, motion detection, motion tracking) are applied to the incoming video frames. Two kinds of outputs are usually generated: trajectories of points on the dancers' bodies (motion trajectories) and processed images. As an example a Silhouette Motion Image (SMI) is an image carrying information about variations of the shape and position of the dancer's silhouette in the last few frames. We also use an extension of SMIs taking into account the internal motion in silhouettes.

From such outputs a collection of motion descriptors are extracted including:



- Cues related to the amount of movement (energy) and in particular what we call Quantity of Motion (QoM). QoM is computed as the area (i.e. number of pixel) of SMI. It can be considered as an overall measure of the amount of detected motion, involving velocity and force.
- Cues related to body contraction/expansion and in particular the Contraction Index (CI), conceived as a measure, ranging from 0 to 1, of how the dancer's body uses the space surrounding it. The algorithm to compute the CI combines two different techniques: the individuation of an ellipse approximating the body silhouette and computations based on the bounding region.
- Cues derived from psychological studies such as amount of upward movement, dynamics of the Contraction Index (i.e. how much CI was over a given threshold along a time unit);
- Cues related to the use of space, such as length and overall direction of motion trajectories.
- Kinematical cues, such as velocity and acceleration of motion trajectories.

A relevant task for Layer 2 is motion segmentation. A possible technique for motion segmentation is based on the measured QoM. The evolution in time of the QoM resembles the evolution of velocity of biological motion, which can be roughly described as a sequence of bell-shaped curves (motion bells, see Figure 6.3). In order to segment motion by identifying the component gestures, a list of these motion bells and their features (e.g. peak value and duration) is extracted. An empirical threshold is defined to perform segmentation: the dancer is considered to be moving if the QoM is greater than 2.5% of the total area of the silhouette. It is interesting to notice that the motion bell approach can also be applied to sound signal analysis.

Segmentation allows extracting further higher-level cues at Level 2. A concrete example is the Directness Index (DI), calculated as the ratio between the length of the straight trajectory connecting the first and the last point of a motion trajectory and the sum of the lengths of each segment constituting the trajectory. Furthermore, motion fluency and impulsiveness can be evaluated. Fluency can be estimated from an analysis of the temporal sequence of motion bells. A dance fragment performed with frequent stops and restarts will result less fluent than the same movement performed in a continuous, "harmonic" way. The hesitating, bounded performance will be characterised by a higher percentage of acceleration and deceleration in the time unit (due to the frequent stops and restarts). A first measure of impulsiveness can be obtained from the shape of a motion bell. In fact, since QoM is directly related to the amount of detected movement, a short motion bell having a high pick value will be the result of an impulsive movement (i.e. a movement in which speed rapidly moves from a value near or equal to zero, to a peak and back to zero). On the other hand, a sustained, continuous movement will show a motion bell characterised by a relatively long time period in which the QoM values have little fluctuations around the average value (i.e. the speed is more or less constant during the movement).

One of the tasks of Layer 4 is to classify dances with respect to their emotional/expressive content. For example, in a study carried in the framework of the EU-IST Project MEGA results were obtained on the classification of expressive gestures with respect to their four basic emotions (anger, fear, grief, joy) by an automatic system based on decision trees.

11.8.7 Examples of Multisensory Integrated Expressive Environments

Here we present a few examples of MIEEs developed with the EyesWeb open software platform in the European Union Information Society Technologies (IST) Multisensory Expressive Gesture Applications (MEGA) project (<http://www.megaproject.org>).

Interactive concert We developed a MIEE to model and implement the interaction between an actress and her vocal clone. This piece, named *Allegoria dell'opinione verbale* (Allegory of the spoken opinion) by the composer Roberto Doati based on poetry by Gianni Revello, was conceived in the Department of Communication, Computer, and System Sciences, or DIST, InfoMus Lab. It was performed on stage during the 2001-2002 season of Gran Teatro La Fenice (Opera House of Venice) at the Teatro Malibran in Venice, within a musical theatre production that performed works from Aperghis, Cage, Casale, Kagel, Pachini, and Schnebel. The piece was also performed at the Opera House of Genova Teatro Carlo Felice in Genova, Italy.

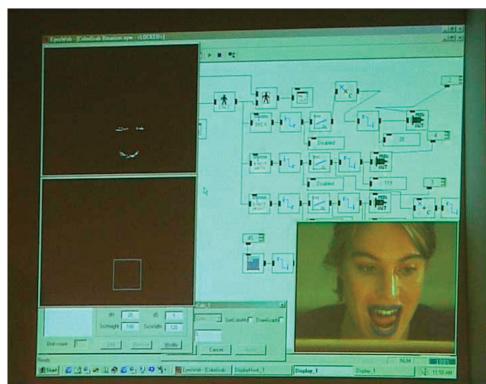


Figure 11.6: The EyesWeb application for Allegoria dell'opinione verbale by R. Doati.

During the concert the actress (Francesca Faiella) sits on a stool placed in the front of the stage near the left side. She is turned toward the left backstage so that the audience sees her profile. A large screen projects a frontal view of her face. A video camera is placed (hidden) in the left part of the backstage. The camera captures images of the actress' face to be projected on the large screen and acquires her lip and facial movements. The actress enacts the poetry in front of the camera. EyesWeb extracts and processes the movements of her lips and face. The system uses expressive cues to process her voice in real time and diffuse spatialized electroacoustic music on eight loudspeakers placed within the auditorium. The signals reproduced by the loudspeakers are derived only from the actress' voice. Previous recordings of her voice reciting the Revello poetry are resynthesized and processed in real time with parameters controlled by lips movements. The audience can observe the movements of the actress' face on the large screen while listening to the piece and thus perceiving the interaction of her movements with sound changes coming from the loudspeakers. Figure 9.6 shows the EyesWeb patch employed in the concert;

Medea: Exploring sound-movement expressiveness. Since 1959, when electronic music was established as a new way of music composition, the rules of traditional music performance and enjoyment have changed to include space, motion, and gesture as musical parameters. For example, musicians are often located somewhere other than the stage, sometimes even in the audience, and where the music will be performed often influences compositional thinking. Loudspeakers move sound through the space at varying speeds (based on other musical parameters). In addition, the development of live electronics—that is, computers applied to real-time processing of instrumental sounds—has allowed space as a musical instrumental practice to flourish. Electro-acoustic technologies let composers explore new listening dimensions and consider the sounds coming from loudspeakers as possessing different logical meanings from the sounds produced by traditional instruments.

Medea, Adriano Guarnieri's "video opera," is an innovative work stemming from research in multi-

media that demonstrates the importance and amount of research dedicated to sound movement in space⁸. Among Medea's intentions, derived from artistic and musical suggestions and needs, is a desire to establish an explicit connection between sound movement and expressiveness and to show how engagement can be enhanced acoustically in multimodality environments, for example, through the motion of sound through virtual spaces. Whereas sound positioning and movement have seldom been used in concert settings, the ear has great detection capabilities connected to its primary role (a signalling device for invisible or unseen cues); music is now trying to put these capabilities to creative use.

Sound motion through space is an established tradition in much of contemporary music, much of which exploits multimodality to enhance performance. Music-specifically sound motion in space-conveys expressive content related to performance gestures. Although composers have investigated the connection between music and emotion for traditional parameters, such as intensity, timbre, and pitch, spatialization is still a new research path. The use of space as a musical parameter in an expressive dimension requires new paradigms for interaction, mapping strategies, and multimedia interfaces based on real-time analysis and synthesis of expressive content in music and gesture. Researchers have developed and applied models and algorithms for extracting high-level, qualitative information about expressive content to real-time music and multimedia applications. Analysis of expressive gestures has sought to extract expressive information from human movements and gestures and to control the generation of audio content depending on the analysis. Figure 9.7 diagrams the link between physical and spatial movement. Medea offers real-world examples of such multimodal extensions.

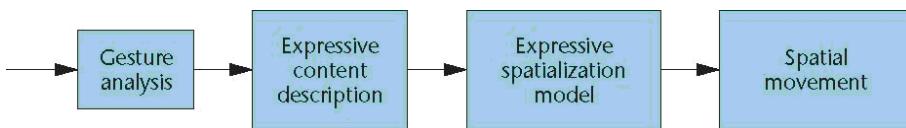


Figure 11.7: Connection between physical and spatial movements.

Medea's score cites sound spatialization as a fundamental feature of the opera. The musicians in the hall should be considered a sonic body living among the audience to create a sort of gravitational centre for the trumpets located on either side of the audience. The presence of trombones with their gestural posture becomes a central expressive feature. The live-electronics performers executed all movements and transformations following the conductor and the score (each instrument has its own spatialization modes, and the score marks each transformation and movement precisely), with all sound movements except for those coming from the trombones having been predetermined. Guarnieri defined 11 modalities for the trombone movements. An EyesWeb patch controlled the trombones' random space movements, a webcam captured movements derived from trombone players' gestures, and the EyesWeb program digitally processed them to provide each movement's speed parameter using a gesture-speed mapping. This method's functionality derives from a translation of the image bitmap in terms of speed: intense instrumental gestural activity (rocking off) leads to a large bitmap variation and therefore to high speed, while reduced gestural activity corresponds to a moderate movement speed.

Figure 9.8 shows one of the four trombone players during Medea's premiere performance. In the context of Medea as a video opera, the expressive matching between physical movement (by the instrumentalist) and sound movement through space clearly plays the metaphorical role of a "camera car," where the public enters the physical movement through the movement of sound itself. It should be noted that all of these considerations are subtle and subliminal as is most of the experience of listening to contemporary music.

⁸adapted from De Gotzen, IEEE Multimedia 2004.



Figure 11.8: Trombone during the premiere performance of Adriano Guarnieri's *Medea*.

11.8.8 Perspectives

Multimodal and cross-modal approaches for integrated analysis of multimedia streams offers an interesting challenge and opens novel perspectives for control of interactive music systems. Moreover, they can be exploited in the broader fields of multimedia content analysis, multimodal interactive systems, innovative natural and expressive interfaces.

This section presented a conceptual framework, research methodologies and concrete examples of cross-modal and multimodal techniques for control of interactive music systems. Preliminary results indicate the potential of such approach: cross-modal techniques enable to adapt to the analysis in a given modality approaches originally conceived for another modality, allowing in this way the development of novel and original techniques. Multi-modality allows integration of features and use of complementary information, e.g. use of information in a given modality for supplementing lack of information in another modality or for reinforcing the results obtained by analysis in another modality.

While these preliminary results are encouraging, further research is needed for fully exploiting cross-modality and multimodality. For example, an open problem which is currently under investigation at DIST - InfoMus Lab concerns the development of high-level models allowing the definition of cross-modal features. That is, while the examples in this chapter concern cross-modal algorithms, a research challenge consists of identifying a collection of features that, being at a higher-level of abstraction with respect to modal features, are in fact independent of modalities and can be considered cross-modal since they can be extracted from and applied to data coming from different modalities. Such cross-modal features are abstracted from the currently available modal features and define higher-level feature spaces allowing for multimodal mapping of data from one modality to another.

Another, more general, open research issue is how to exploit the information obtained from multimodal and cross-modal techniques for effective control of future interactive music systems. That is, how to define suitable strategies for mapping the information obtained from the analysis of users' behavior (e.g. performer's expressive gestures) onto real-time generation of expressive outputs (e.g. expressive sound and music output). This issue includes the development of mapping strategies integrating both fast adaptive and reactive behavior and more high-level decision-making processes. Current state-of-the-art control strategies often consist of direct associations, without any dynamics, of features of analyzed (expressive) gestures with parameters of synthesised (expressive) gestures (e.g. the actual position of a dancer on the stage may be mapped onto the reproduction of a given sound). Such direct associations are usually employed for implementing statically reactive behavior. The objective is to develop high-level indirect strategies, including reasoning and decision-making processes, and related to rational and cognitive processes. Indirect strategies implement adaptive and dynamic behavior and are usually charac-

terised by a state evolving over time and decisional processes. Production systems and decision-making algorithms may be employed to implement this kind of strategies. Multimodal interactive systems based on a dialogical paradigm may employ indirect strategies only or a suitable mix of direct and indirect strategies.

As a final remark, it should be noticed that control issues in the Sound and Music Computing field are often related to aesthetic, artistic choices. To which extent can a multimodal interactive (music) system make autonomous decisions? That is, does the system have to follow the instructions given by the director, the choreographer, the composer, (in general the creator of a performance or of an installation) or is it allowed to have some degree of freedom in its behavior? The expressive autonomy of a multimodal interactive system is defined as the amount of degrees of freedom that a director, a choreographer, a composer (or in general the designer of an application involving communication of expressive content) leaves to the system in order to make decisions about the most suitable expressive content to convey in a given moment and about the way to convey it. In general, a multimodal interactive system can have different degrees of expressive autonomy and the required degree of expressive autonomy is crucial for the development of its multimodal and cross-modal control strategies.

References

- P. Polotti and D. Rocchesso. *Sound to Sense - Sense to Sound: A state of the art in Sound and Music Computing*. Logos Verlag, Berlin, Germany, 2008.



Chapter 12

Musical cultural heritage: From preservation to restoration

Sergio Canazza

Copyright © 2005-2018 Sergio Canazza

except for paragraphs labeled as *adapted from <reference>*

This book is licensed under the CreativeCommons Attribution-NonCommercial-ShareAlike 3.0 license. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/>, or send a letter to Creative Commons, 171 2nd Street, Suite 300, San Francisco, California, 94105, USA.

12.1 Introduction

The availability of digital archives and libraries on the Web represents a fundamental impulse for cultural and didactic development. Guaranteeing an easy and ample dissemination of some of the fundamental moments of the music culture of our times is an act of democracy that must be assured to future generations, even through the creation of new tools for the acquisition, preservation, and transmission of information. This is a crucial point, which is nowadays one of the core reflections of the international archive community. If, on the one hand, scholars and the general public have begun paying greater attention to the recordings of artistic events, on the other hand, the systematic preservation and consultation of these documents is complicated by their diversified nature, because the data contained in the recordings offer a multitude of information on their artistic and cultural life, that goes beyond the audio signal itself.

In this sense, a complete access to the audio content cannot be carried out without accessing to the contextual information, that is to all the content-independent information available from the cover, the signs on the carrier, and so on. In addition, a preservative re-recording and cataloguing of audio document collections cannot leave out a consideration of the history of the institutions or collections in which they are held. In fact, this information helps defining the strategy to adopt during the preservative interventions.

It is well-known that the recording of an event can never be a neutral operation, because the timbre quality and the plastic value of the recorded sound, which are of great importance in contemporary music, are already influenced by the positioning of the microphones used during the recording. In ad-

dition, the audio processing carried out by the Tonmeister¹ is a real interpretative element added to the recording of the event. Thus, musicological and historic-critical competence becomes essential for the individuation and correct cataloguing of the information contained in audio documents. Being made of unstable base materials, sound carriers are more subject to damage caused by inadequate handling. The commingling of a technical and scientific formation with historic-philological knowledge becomes essential for preservative re-recording operations, going beyond mere analog-to-digital (A/D) transfer.

Since the first recording² on paper made in 1860 (by Edouard-Léon Scott de Martinville "Au Clair de la Lune" using his phonautograph) to the modern Blu-ray Disc, what we have in the audio carriers field today is a Tower of Babel: a bunch of incompatible analogue and digital approaches and carriers – paper, wire, wax cylinder, shellac disc, film, magnetic tape, vinyl record, magnetic and optical disc, to mention only the principal ones – without standard players able to read all of them. As far as audio memories are concerned, preservation is divided into a passive³ preservation, that is the defence of the carrier from external agents without altering the structure, and an *active preservation*, which involves data transfer on new media. The commingling of a technical and scientific formation with historic-philological and philosophical knowledge also becomes essential for preservative re-recording operations, which do not completely coincide with pure A/D transfer, as is, unfortunately, often thought. Three examples will be made, in different music genres.

1. *Stria* (John Chowning, 1977): in the CCRMA version (four-channels), there was a signal discontinuity in the D/A conversion of the data at 6' 29" (389") from the original computation that was unintended. This caused a sudden change of timbre and a consequent click: "[T]he PDP-10 burped!". This imperfection in the computation emerges slightly, but very clearly indeed, in the audio source. John Chowning did not re-compute the section to eliminate this problem. He rather learned to accept it "as one does a birth mark or beauty mark on ones skin... noticeable but of no substantive consequence". The faulty, imperfect, and therefore fascinating four-channel version is the version that John Chowning now uses to play during the concerts. Conversely, in the commercial version (CD Wergo, WER 2012-50) this "burp" is missing. The audio is truncated exactly at that point (6' 29") with a fade out to the following section. This is an example of a lack of the philological attention during the re-recording process.
2. *Y entonces comprendió* (Luigi Nono, 1970): it is a four-channel music work. Luigi Nono produced also a stereophonic version in a four-channel tape (A, B, C, D), mixing the original four-channel tape (1, 2, 3, 4): A = 1 + 3; B = 1 + 3; C = 2 + 4; D = 2 + 4. The Stereo Long Playing Deutsche Grammophon DGG 2530436 (stereo: X, Y) was mixed: X = A + C; Y = B + D. In this way, a stereo version is reduced to a monophonic version, because of a transmission error.
3. The commercial audio discs, dating from 1894, have been recorded following a large set of different carriers and encodings. As for their physical composition, audio discs can range from fragile forms such as rubber (the earliest disc recordings), acetate or lacquer (sometimes with glass, aluminum, or cardboard backings), to more-durable shellac and vinyl discs and the metal masters used to stamp commercial discs. The distinct characteristics of each disc type require different techniques, often highly specialized, to coax the sound from the carrier. So, in this case several choices (in relation with the phonographic disc history) are necessary to optimize the extraction of

¹The term Tonmeister describes a person who has a detailed theoretical and practical knowledge of all aspects of sound recording. But, unlike a sound engineer, he/she must be also deeply musically trained. Both competencies have equal importance in a Tonmeister's work.

²Unlike Edison's similar 1877 invention, the phonograph, the phonautograph only created visual images of the sound playback capabilities. Scott de Martinville's device was used only for scientific investigations of sound waves.

³*Passive preservation* is divided into indirect, which does not physically involve the carrier, and direct, in which the carrier is treated without altering its structure and composition.



the audio signal from the original carrier: the pick-up arm, the cartridge, the stylus, the speed, and the replay equalization are all factors that influence the result of the re-recording process.

It is worth noting that, in the Seventies/Eighties of 20th Century, expert associations (Audio Engineering Society: AES; National Archives and Records Administration: NARA; Association for Recorded Sound Collections: ARSC) were still concerned about the use of digital recording technology and digital storage media for long-term preservation. They recommended re-recording of endangered materials on analogue magnetic tapes, because of: a) rapid change and improvement of the technology, and thus rapid obsolescence of hardware, digital format and storage media; b) lack of consensus regarding sample rate, bit depth and record format for sound archiving; c) questionable stability and durability of the storage media. The digitization was considered primarily a method of providing access to rare, endangered, or distant materials – not a permanent solution for preservation. Abby Smith (director of programs at the Council on Library and Information Resources (CLIR), USA, <http://www.clir.org>), still in 1999, suggested that digitization should be considered a means for access, not preservation – “at least not yet”.

Nowadays, it is well-known that preserving the carriers and maintaining the dedicated equipment for their reproduction is hopeless. The audio information stored in obsolete formats and carriers is in risk of disappearing. To this end, the audio preservation community introduced the concept “preserve the content, not the carrier”. Audio (and video) preservation must therefore be based on digital copying of contents. Consequently, analogue holdings must be digitized. At the end of the 20th century, the traditional “preserve the original” paradigm shifted to the “distribution is preservation” idea of digitizing the audio content and making it available using digital libraries technology. Now the importance of transferring into the digital domain (active preservation) is clear, namely for carriers in risk of disappearing, respecting the indications of the international archive community (e.g., Audio Engineering Society, AES; International Association of Sound and Audiovisual Archives, IASA; International Federation of Library Associations, IFLA).

This chapter, after a detailed overview of the debate evolved since the Seventies inside the archivist community on audio documents preservation (Sect. 10.2), describes the protocols defined, the processes undertaken, the results ascertained from several international audio documents preservation projects and the techniques used. In particular, in Sect. 10.3 and Sect. 10.4, some guidelines are given, including recommendations to the A/D process directed to minimize the information loss and to automatically measure the unintentional alterations introduced by the A/D equipment, focusing on the high quality/high cost/low throughput cases. The author believes that the increased dimensionality of the data contained within an audio digital library should be dealt with by means of automatic annotations. Therefore, this chapter presents in Sect. 10.5 a set of tools able to extract, in a semi-automatic way, metadata from photos and video shootings of audio carriers. These tools are useful, in particular, in settings where it is necessary to put attention to the cost-benefit tradeoffs. Sect. 10.6 presents an original system for reconstructing the audio signal from a still image of a disc surface and an alignment technique aimed at comparing the effectiveness and the robustness of different re-recording techniques.

12.2 Audio Documents Preservation

A reconnaissance on the most significant positions of the debate evolved since the Seventies inside the archivist community on the audio documents active conservation highlights at least three different points of view, described below.



12.2.1 “Two Legitimate Directions”

It was William Storm, at that time Assistant Director of the Thomas A. Edison Re-recording Laboratory Syracuse University Libraries, who focussed on the problem of standardizing the procedures of audio restoration in an article which became famous for the numerous controversies it arose. Storm individuated two legitimate directions, two types of re-recording which are suitable from the archival point of view: 1) the sound preservation of audio history, and 2) the sound preservation of an artist.

The first type of re-recording (Type I) represents a level of reproduction defined as the perpetuation of the sound of an original recording as it was initially reproduced and heard by the people of the era. Storm’s contribution aimed at shifting the archivist’s interest from the simple collecting of audio carriers to the information contained in the recording, and at highlighting the double documentary value of re-recording by proposing an audio-history sound preservation: on the one hand, he wanted to offer a historically faithful reproduction of the original audio recording by extracting the sound content according to the historical conditions and technology of the era in which it was produced; on the other hand, he wanted to document the quality of sound reception offered by the recording and reproducing systems of the time. These two instances, conceptually joined in a single type of re-recording, had induced Storm to prescribe the use of original playback equipment. The aim of history preservation is to first hear how records originally sounded to the general public.

The second type of re-recording (Type II) was presented by Storm as a further stage of audio restoration, as a more ambitious research objective, conceived as a coherent development of Type I: The knowledge acquired through audio-history preservation provides the sound engineer with a logical place to begin the next step – the search for the true sound of an artist. Type II is then characterized by the use of playback equipment other than that originally intended so long as the researcher proves that the process is objective, valid, and verifiable, with the intent of obtaining the live sound of original performers, transcending the limits of a historically faithful reproduction of the recording.

12.2.2 “To Save History, Not Rewrite It”

The “Safeguarding the Documentary Heritage. A Guide to Standards, Recommended Practices and Reference Literature Related to the Preservation of Documents of All Kinds” commissioned by UNESCO reports the philosophical approach *save history, not rewrite it*. The audio section is clearly influenced by the new formulations made by Dietrich Schüller. Schüller’s works move from a different methodological point of view, which is to analyse what the original carrier represents, technically and artistically, and to start from that analysis in defining what the various aims of re-recording may be. Regarding the reconstruction of the history of music perception Schüller states: “The only case where the use of original equipment is justified is in the exotic aim to reconstruct the sound of a historical recording as it was heard originally”. Instead he points directly towards defining a procedure which guarantees the re-recording of the signal’s best quality by limiting the audio processing to the minimum. Having set aside the general philosophical themes, Schüller goes on to an accurate investigation of signal alterations which he classifies in two categories: intentional and unintentional. The former include recording, equalization, and noise reduction systems, while the latter are further divided into two groups: the ones caused by the imperfection of the recording technique of the time, resulting in various distortions and the ones caused by misalignment of the recording equipment, for example, wrong speed, deviation from the vertical cutting angle in cylinders or misalignment of the recording in magnetic tape.

The choice whether or not to compensate for these alterations reveals different re-recording strategies: historical faithfulness can refer to various levels: Type A the recording as it was heard in its time, which is equivalent to Storm’s Type I presented in the previous section; Type B the recording as it has been produced, precisely equalized for intentional recording equalizations, compensated for eventual er-



rors caused by misaligned recording equipment and replayed on modern equipment to minimize replay distortions.

Type B re-recording defines a historically faithful level of reproduction that, from a strictly preservative point of view, is preliminary to any further possible processing of the signal. These compensations use knowledge which is external to the audio signal; therefore, even in the operations provided for by Type B, there is a certain margin of interpretation because a historical acquaintance with the document is called into question alongside with technical-scientific knowledge. For instance, to individuate the equalization curves of magnetic tapes or to determine the rotation speed of a record. Most of the information provided by Type B is retrievable from the history of audio technology, while other information is instead experimentally inferable with a certain degree of precision. The re-recording work can thus be carried out with a good degree of objectivity and represents an optimal level within which the standard for a preservation copy can be defined.

After having established an operational criterion for preservative re-recordings, based on stable procedures and derived from an objective knowledge of the degradations, Schüller individuated a third level of historically faithful reproduction, type C: The recording as produced, but with additional compensation for recording imperfections caused by the recording technique of the time. While the compensations of type B are commonly accepted and must – as Schüller writes – be carried out, in type C they have to do with the area of equalizations used to compensate for non-linear frequency response, caused by imperfect historical recording equipment and to eliminate rumble, needle noise, or tape hiss. These are operations which elude standard operational criteria and must therefore be rigorously documented by the restorer, who must write out accurate reports in which he specifies both the equipment and systems used as well as all the restoration phases.

12.2.3 “Secondary Information”: the History of the Audio Document Transmission

The studies of George Brock-Nannestad are in line with the modeling of the degradations through reverse engineering. In these studies he focused on the A/D conversion of acoustic recordings (thus recordings made before 1925) and, in particular, the strong line spectrum in the recording transfer function and unknown recording speed. Brock-Nannestad goes back to the first studies in the acoustics of sound reproduction and to the scientific works of Dayton C. Miller, whom we must recall as the first to attempt to retrieve the true sound once it had been recorded. In order to be consistent and have scientific value, the re-recording work requires a complete integration between the historical-critical knowledge which is external to the signal and the objective knowledge which can be inferred by examining the carrier and the degradations highlighted by the analysis of the signal.

12.2.4 The Audio Preservation Protocol

Starting from these positions, I define a *preservation copy* a digital data set that groups the information carried by the audio document, considered as an artifact. It aims to preserve the documentary unity, and its bibliographic equivalent is the facsimile or the diplomatic copy. Signal processing techniques are allowed only when they are finalized to the carrier restoration. The audio format identification and the choice of the playing equipment are crucial because only the intentional alterations have to be compensated. The A/D transfer process should represent the original document characteristics, from either information and material points of view, as it arrived to us.

Fig. 10.1 summarizes the different points of view inside the debate evolved inside the archivist community on the audio documents re-recording.

According to the indications of the international archive community: 1) the re-recording is transferred from the original carrier; 2) if necessary, the carrier is cleaned and restored so as to repair any



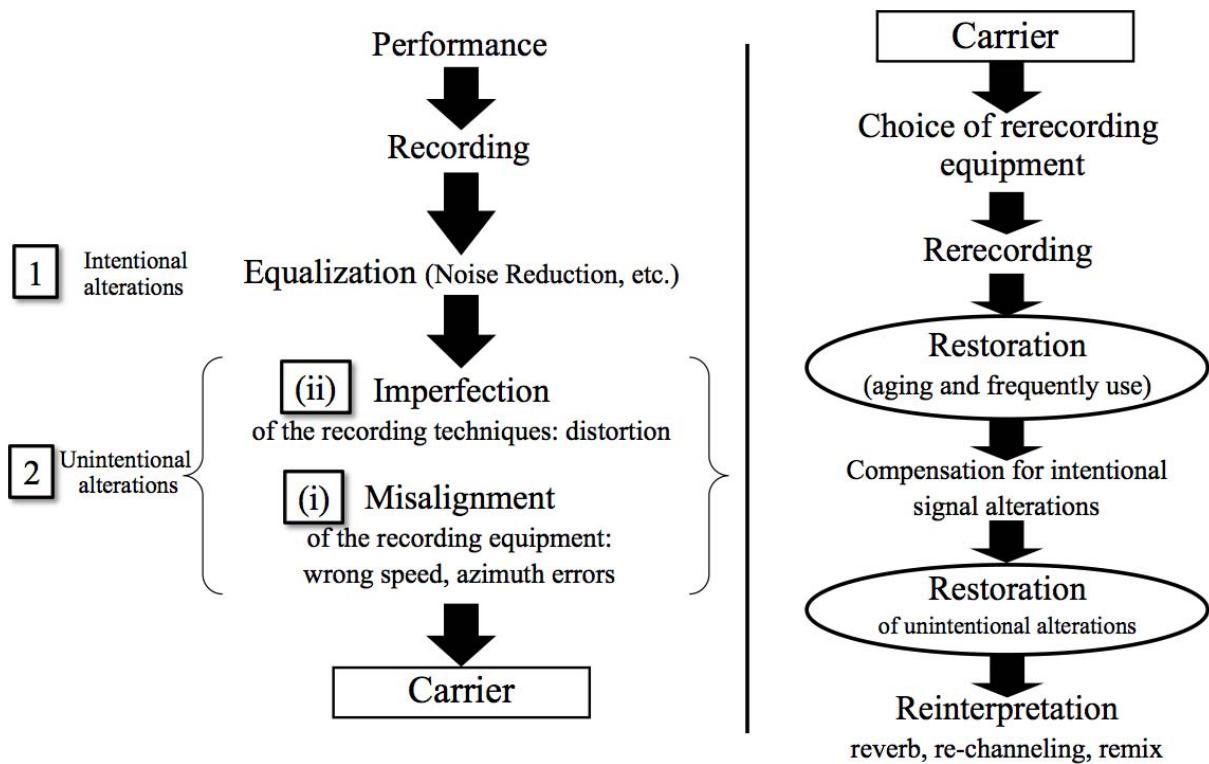


Figure 12.1: The schema of the most significant positions of the debate evolved since the Seventies inside the archivist community on the audio documents active conservation.

climatic degradations which may compromise the quality of the signal; 3) re-recording equipment is chosen among the current professional equipment available in order not to introduce further distortions; 4) sampling frequency and bit rate must be chosen with respect to the archival sound record standard (see Sect. 10.4.3.1); 5) the digital audio file format should support high resolution, it should be transparent with simple coding schemes, without data reduction. Moreover, differently by Schüller position, it is our belief that - in a preservation copy - only the intentional alterations must be compensated (correct equalization of the re-recording system and decoding of any possible intentional signal processing interventions). All the unintentional alterations (also the ones caused by misalignments of the recording equipment) could be compensated only at the access copy level: these imperfections/distortions must be preserved because they witness the history of the audio document transmission.

Because these guidelines should be customized for each carrier, the archivists have to know all their implications, from physic and chemical points of view, and should posses a deep knowledge about the technology for re-recording and of the digital formats in which the digital preservation copy is to be stored.

12.3 Passive Preservation

The direct passive preservation can be carried only if the main causes of the physical Carriers deterioration are known and consequently avoided. We summarize the main risks for the two most common categories of carriers: mechanical carriers and magnetic tapes.

Carrier	Period	Composition	Stocks
cylinder – recordable	1886-1950s	Wax	300,000
cylinder – replicated	1902-1929	Wax and Nitrocellulose with plaster (“Blue Amberol”)	1,500,000
coarse groove disc – replicated	1887-1960	Mineral powders bound by organic binder (“shellac”)	10,000,000
coarse and microgroove discs – recordable (“instantaneous discs”)	1930-1950s	Acetate or nitrate cellulose coating on aluminum (or glass, steel, card)	3,000,000
microgroove disc (“vinyl”) - replicated	1948-	Polyvinyl chloride - polyacetate co-polymer	30,000,000

Table 12.1: Typologies of analogue mechanical carriers

12.3.1 Mechanical Carriers

The common factor with this group of documents is the method of recording the information, which is obtained by means of a groove cut into the surface by a stylus modulated by the sound, either directly in the case of acoustic recordings or by electronic amplifiers. Mechanical carriers include: phonograph cylinders; coarse groove gramophone, instantaneous and vinyl discs. Tab. 10.1 summarizes the typologies of these carriers.

The main causes of deterioration are related to the instability of mechanical carriers and can be summarized as:

1. **Humidity.** Humidity, as with all other data carriers, is a most dangerous factor. While shellac and vinyl discs are less prone to hydrolytic instability, most kinds of instantaneous discs are extremely endangered by hydrolysis. Additionally, all mechanical carriers may be affected by fungus growth which occurs at humidity levels above 65% RH.
2. **Temperature** Elevated temperatures beyond 40 C are dangerous, especially for vinyl discs and wax cylinders. Otherwise the temperature determines the speed of chemical reactions like hydrolysis and should therefore be kept reasonably low and, most importantly, stable to avoid unnecessary dimensional changes.
3. **Mechanical Deformation.** Mechanical integrity is of the greatest importance for this kind of carriers. It is imperative that scratches and other deformation caused by careless operation of replay equipment are avoided. The groove that carries the recorded information must be kept in an undistorted condition. While shellac discs are very fragile, instantaneous and vinyl discs are more likely to be bent by improper storage. Generally, all mechanical discs should be shelved vertically. The only exceptions are some soft variants of instantaneous discs.
4. **Dust and Dirt.** Dust and dirt of all kinds will deviate the pick-up stylus from its proper path causing audible cracks and clicks. Fingerprints are an ideal adhesive for foreign matter. A dust-free environment and cleanliness is, therefore, essential.

12.3.2 Magnetic Tape

The basic principles for recording signals on a magnetic medium were set out in a paper by Oberlin Smith in 1880. The idea was not taken any further until Valdemar Poulsen developed his wire recording



system in 1898. Magnetic tape was developed in Germany in the mid 1930's to record and store sounds. The use of tape for sound recording did not become widespread, however, until the 1950's. Magnetic tape can be either reel to reel or in cassettes. Tab. 10.2 summarizes the typology of these supports:

Period	Type of recording	Composition
1935-1960	Analogue	base: cellulose acetate magnetic pigment: Fe_2O_3 formats: open reel
1944-1960	Analogue	base: PVC magnetic pigment: Fe_2O_3 formats: open reel
1959-	Analogue	base: polyester magnetic pigment: Fe_2O_3 formats: open reel
1969-	Analogue/Digital	base: polyester magnetic pigment: CrO_2 formats: compact cassette IEC II, DCC
1979	Analogue/Digital	base: polyester magnetic pigment: metal particle formats: compact cassette IEC IV, R-DAT

Table 12.2: Typology of magnetic tape carriers

The main causes of deterioration are related to the instability of magnetic tape carriers and can be summarized as:

1. **Humidity.** Humidity is the most dangerous environmental factor. Water is the agent of the main chemical deterioration process of polymers: hydrolysis. Additionally, high humidity values (above 65% RH) encourage fungus growth, which literally eats up the pigment layer of magnetic tapes and floppy disks⁴ and also disturbs, if not prevents, proper reading of information.
2. **Temperature.** Temperature is responsible for dimensional changes of carriers, which is a particular problem for high density tape formats. Temperature also determines the speed of chemical processes: the higher the temperature, the faster a chemical reaction (e.g., hydrolysis) takes place.
3. **Mechanical Integrity.** Mechanical integrity is a much underrated factor in the accessibility of data recorded on magnetic media: even slight deformations may cause severe deficiencies in the playback process. Most careful handling has to be exercised, along with regular professional maintenance of replay equipment, which, in case of malfunctioning, can destroy delicate carriers such as R-DAT very quickly. With all tape formats, it is most important to obtain an absolutely flat surface of the tape pack to prevent damage to the tape edges which serve as mechanical references in the replay of many high density formats. All forms of tape should be stored upright.

⁴Floppy disks are one of the most used supports to store audio documents in the field of electronic music in the 80s and 90s of the last century. The composers usually saved in floppy disks some short sound objects, synthesized at low sampling Hertz (8 – 15kHz). The study of this musical excerpt is very important from a musicologist point of view. For instance, the Archive of the Centro di Sonologia Computazionale (CSC, University of Padova, Italy: <http://csc.dei.unipd.it/>) has hundreds of floppy disks: it is unquestionably an outstanding testimony of the musical history in the 80' and 90' years of XX Century.



4. **Dust and Dirt.** Dust and dirt prevents the intimate contact of replay heads to the medium which is essential for the correct access to the information especially with high density carriers. The higher the data density, the more cleanliness has to be observed. Even particles of cigarette smoke are big enough to hide information on modern magnetic formats. Also pollution caused by industrial smog can accelerate chemical deterioration. The effective prevention of dust is an indispensable measure for the proper preservation of magnetic media.
5. **Magnetic Stray Fields.** Magnetic stray fields are the natural enemy of magnetically recorded information. Sources of dangerous fields include dynamic microphones, loudspeakers and headsets. Also the simple magnets used for magnetic notice boards possess magnetic fields of dangerous magnitudes. By their nature, analogue audio recordings, including audio tracks on video tapes, are the most sensitive to magnetic stray fields. It should be noted that normally a distance of 10-15 cm is enough to diminish the field strength of even strong magnets to acceptably low values.

Among the others, some effects can be:

- “drop out” (i.e. the magnetic material fall off the tape);
- “bleed through” (i.e. the signal from one section of tape imprinting on another when the tape has been stored for a long time: this is a big issue in several magnetic recordings and is really noticeable in the excerpts with a low SNR);
- “stretch” (i.e. the actual permanent stretching of the polyester cause by too tightly spooling the tape with noticeable pitch dropping).

Tab. 10.3 shows the correct parameters for the passive preservation of mechanical and tape carriers.

	temp.	$\pm/24\text{h}$	\pm/year	RH	$\pm/24\text{h}$	\pm/year
preservation storage	$5^\circ\text{C} < t < 10^\circ\text{C}$	$\pm 1^\circ\text{C}$	$\pm 2^\circ\text{C}$	30%	$\pm 5\%$	$\pm 5\%$
access storage	about 20°C	$\pm 1^\circ\text{C}$	$\pm 2^\circ\text{C}$	40%	$\pm 5\%$	± 5

Table 12.3: Recommended climatic storage parameters for mechanical and tape carriers

12.4 Active Preservation

This section details a protocol for the task of audio documents active preservation, which is summarized in Fig. 10.2. The protocol has been defined by the author and put it into practice in several European audio archives projects (see Sect. 10.8).

12.4.1 Carrier Analysis and Restorative Actions

During this phase (steps 1 and 2 shown in Fig. 10.2) the state of the document must be evaluated and the physical characteristics of the carrier and its format assessed, also on the basis of historical research carried out on the technologies in use at the time of the recording. The preservative re-recording operation should be monitored so to memorize every phase of the process and to testify the accuracy of the protocol used. In particular, a video recording, synchronized with the audio signal, should document the presence



of splices, corruptions and graphical signs. The documentation of this meaningful editing traces is very important for the signal alteration classification and for the philological work of genesis reconstruction.

The information on the format of the carrier has to be inferred from the direct analysis of the carrier and then compared with the technical data contained on the case/cover/label, even if it is often wrong or missing. The data inferred from the history of audio technology are a source of knowledge which cannot be ignored when defining methods and procedures for the survey of the formats and replay parameters adopted during the original recording, because they allow us to solve specific problems caused by the technical defects of the equipment used for the creation of the document. Clearly, all the results of this recognition have to be stored as additional information.

12.4.2 Re-recording

This phase details steps 3 and 4 shown in Fig. 10.2. On the basis of the information gathered in the first phase, the playback analogue equipment is chosen to avoid introducing further distortions and to collect more information than the one offered by the equipment of the time. The technical-functional analysis confirms the importance of this choice. For instance, tape recorders built before the 80s present: a) low signal-to-noise-ratio (SNR); b) fixed and non-modifiable equalizations; c) unreliability of the tape transport system in guaranteeing the physical integrity of the original document. According to the considerations given in Sect. 10.2, the transfer from the old to the new format has to be carried out without subjective alterations or “improvements”, such as de-noising, because the unintended and undesirable artifacts are also part of the sound document, even if they have been subsequently added to the original signal by mishandling, poor storage or as a consequence of aging. Both have to be preserved with the utmost accuracy, because they provide information about the persons and the corporate bodies that were involved in the creation and in the transmission of the document. Alteration removal or attenuation on the signal need subjective choices of the restorer.

The A/D transfer is a delicate aspect of the re-recording procedure. Because original carriers may contain secondary information (i.e., bias frequency⁵, broadband impulsive noise) which falls outside the frequency range of the primary information (signal), the transfer must be carried out to the highest among the available standards.

Every audio document presents original technical aspects. It is precisely because of this instability inherent in the document that it is impossible to carry out automatic re-recordings with the simultaneous use of several systems. The process should be constantly monitored, and a number of signal alterations need to be catalogued and described:

- local noise: clicks, pops, signal dropout due to joints or tape degradation;
- global noise: hums, background noise, distortion (periodical or non-periodical);
- alterations produced during the sound recording phase: electrical noises (clicks, ripples), microphone distortions, blows on the microphone, induction noise;
- signal degradation due to malfunctions of the recording system (i.e., partial tracks deletion).

12.4.3 Preservation Copy

This section describes steps from 5 to 8 shown in Fig. 10.2.

⁵bias is the addition of an inaudible high-frequency signal to the audio signal. Bias increases the signal quality of audio recordings pushing the signal into the linear zone of the tape's transfer function.



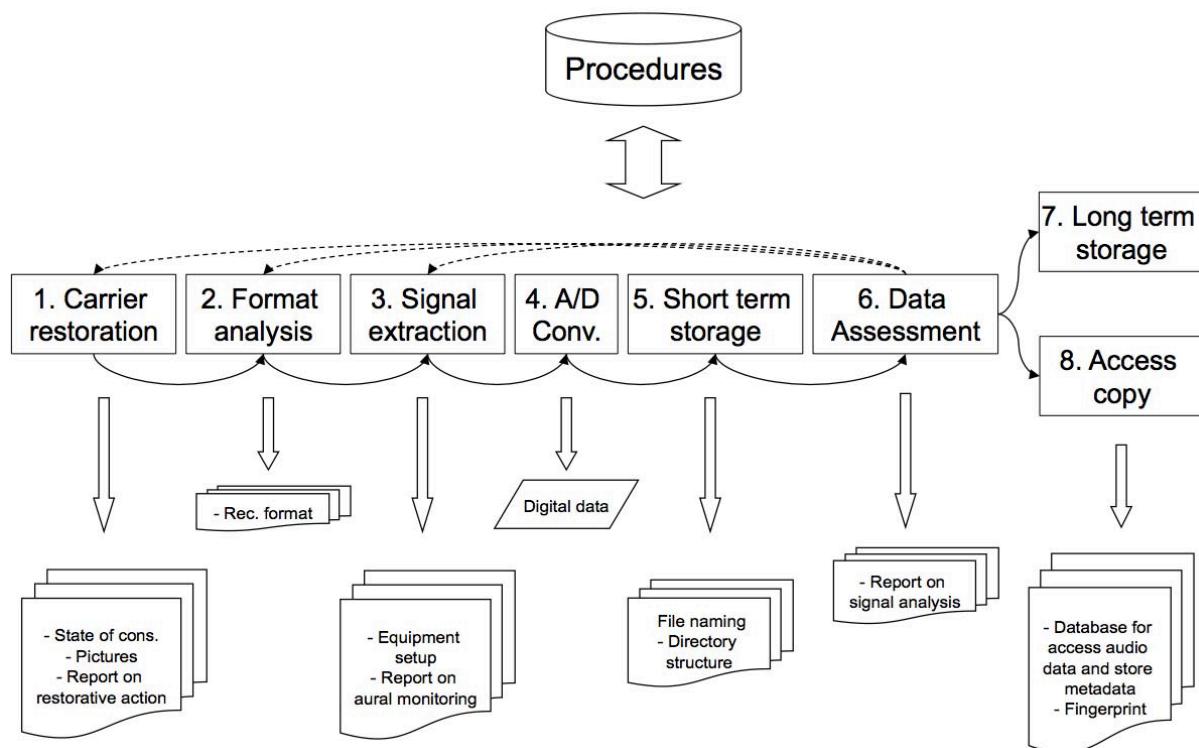


Figure 12.2: Representation of the A/D transfer protocol

A *preservation copy* (or archive copy) is the artifact designated to be stored and maintained as the preservation master. Such a designation may be given either to the earliest generation of the artifact held in the collection, to a preservation transfer copy of such an artifact, and/or to both such items in the possession of the archive. Such a designation means that the item is used only under exceptional circumstances⁶. During the process of active preservation, the original document – multimedia in itself, because it is made up of the audio signal, static images (label, case, carrier corruptions, etc.), text (attachments), smell (mould, vinegary, etc.) – is converted into a digital document, which could be defined as an unimedia document, because it is a fusion of different media in a single bit flow.

This projection of a multidimensional object into a one-dimensional space produces a particularly large and varied set of digital documents, which are made up of the audio signal, the metadata and the contextual information. It is important to note that in this context, as it is common practice in the audio processing community, we use the term metadata to indicate content-dependent information that can be automatically extracted by the audio signal; as already mentioned we indicate as contextual information the additional content-independent information. The goal of active preservation is to minimize the information loss during the A/D transfer of the document. In order to preserve the documentary unity it is therefore necessary to digitize contextual information, which is included in the original document and the

⁶ Audio carriers, especially modern high density formats, are, by their very nature, vulnerable. Additionally, there is always the risk of accidental damage through improper handling, malfunctioning equipment or disaster. One strategy, for the long term storage, that is widely used is the creation of access copies of documents. A poor quality copy can act as an adjunct to the catalogue to aid researchers to decide what documents they wish to study. A good quality copy may be acceptable for study in place of the original. The (online or local) use of copies to reduce the frequency of access to the original document will reduce the stress on the original and help to preserve it. A clear policy about the classes of researchers allowed access to original documents – particularly fragile ones – will also help documents survive. It is clearly impossible to totally restrict access to originals but many users can perform their research using good quality access copies.

metadata which comes out from the transfer process: the information written on the edition containers (envelopes, cases and boxes), on the label, on the flange, on the carrier and on possible attachments (text, images, physical conditions, intentional alterations, corruptions) and the information related to the process of audio signal transfer (schemes of the A/D system) must be arranged and so they become a complete part of the conservative copy.

As for all types of digital documents, also in this case digital preservation methods and techniques have to be exploited, to maintain the accessibility of the preservation copy, its metadata and contextual information.

12.4.3.1 Format for the Audio Files

According to the rule *the worse the signal, the higher the resolution*, the audio signal should be stored in the preservation copy using the Broadcast Wave Format, sampled at least at 96 kHz with a 24 bit resolution. It is advisable to use the monophonic format, where each recording track is equivalent to a different file with Pulse Code Modulation representation.

In order to preserve sound documents in a philologically correct way during the re-recording procedures, it is essential to rely on operational protocols aimed at avoiding the overlapping of modern phonic aspects that alter the original sound content. In particular, the criteria for the preservation of documents should not be influenced by the market-induced tendency to use lossy compression formats. The low quality of lossy compression, especially if considered in relation to the phonic richness of much contemporary music, imposes the rigorous avoidance of any mixture between the acquisition of documents for conservative aims (preservation copies) and the archiving for common use (access copies).

12.4.3.2 Video Shooting and Photographic Documents

The information written on edition containers, labels and other attachments should be stored with the preservation copy as static images (two examples are given in Fig. 10.3 (a) and (b)), as well as the photos of clearly visible carrier corruptions. A video of the carrier playing – synchronized with the audio signal – ensures the preservation of the information on the carrier (physical conditions, presence of intentional alterations, corruptions, graphical signs). The video recording offers:

1. Information related to magnetic tape assembly operations and corruptions of the carrier (disc, cylinder or tape), which are indispensable to distinguish the intentional from the unintentional alterations during the restoration process.
2. A description of the irregularities in the playback speed of analogue recordings (wow and flutter⁷): in discs, a spindle hole not precisely centered and/or the warping of the disc cause a pitch variation; in tape recorders, an irregular tape motion during playback (a change in the angular velocity of the capstan, or dragging of the tape within an audio cassette shell) cause changes in frequency. From the video, it is possible to locate automatically the imperfections occurred during the A/D transfer (see Sect. 10.5 for some examples): in this way, in the restoration process we will be able to distinguish among the alterations occurred at the recording step or at playback level.
3. Instructions for the performance of the piece (in particular in the electro-acoustic music for tape): from the video analysis, some prints of the tape can be displayed; they represent either the synchronization of the score or the indication of particular sound events (Fig. 10.4).

⁷Wow and flutter are audio distortions perceived as an undesired frequency modulation in the range of: i) wow from 0.5 Hz to 6 Hz, ii) flutter from 6 Hz to 100 Hz. The distortions are introduced to a signal by an irregular velocity of the analogue medium. As the irregularities can originate from various mechanisms, the resulting parasitic frequency modulations can range from periodic to accidental, having different instantaneous values.





Figure 12.3: (a) a sound postcard: it looked like a standard postcard on the back, but on the front an analogue recording was engraved in a thin layer of laminate. Sound postcards were usually made by small firms, and the recording quality was extremely low; in this case the importance of storing the picture in with the preservation copy is particularly evident. (b) displays a label of His Master's Voice disc: DK 119 (on the label, right) is the catalogue number; 2-054042 (on the label, left, and at the top of the mirror) is a second catalogue number (as its minor typographic importance, probably it is the first issue catalogue number: therefore here we have a reprint); A12804 (in the mirror, down) is the matrix number. It is possible to decode this information: DK = 30 cm diameter; Yellow label = "International Celebrità" series, printed in Hayes; 2-054 prefix in catalogue number corresponds to a second issue (2), 30 cm diameter (0), Italian catalogue (5) and duet or trio as sound content (4); by means of a comparison between matrix number and published repertoires we can deduce the recording date (17th, January, 1913). (c) and (d) show two typical corruptions in a tape and in a disc respectively: this information should be stored with the preservation copy also, in order to have a deep insight the artifacts of the audio signal.

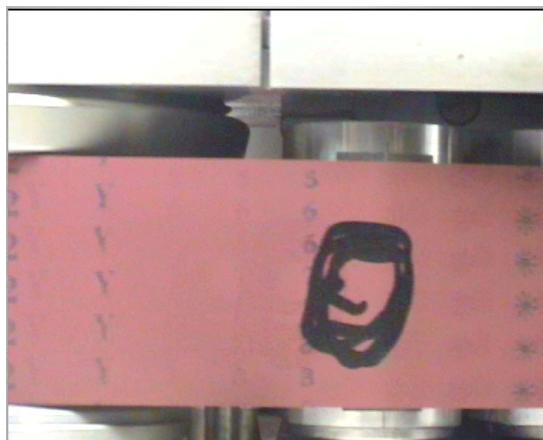


Figure 12.4: Frame of a video recording of an open reel tape: the circle drawn in black marks a specific sound event. Often, in the electro-acoustic music field (in the works for tape and acoustic musical instruments) the marks on the tape are used as a synchronization means between live-electronics performer and the recorded tape music. If this information was not preserved, it would not be possible to perform the piece.

The video file should be stored with the preservation copy. The selected resolution and the compression factor must at least allow to locate the signs and corruptions of the support. In our experience, a 320x240 pixels resolution video with medium quality DivX compression yielded satisfactory results.

12.4.3.3 Audio Fingerprinting

The deterioration of the digital carrier used for storing the preservation copy could cause some errors in the audio files. If the errors are restricted to the bits assigned to the audio signal codification, however the file is proved to be readable, but it is no longer capable of returning exactly an audio signal equal to the one which was digitized. A control device of the integrity of the audio files, thus, should be introduced in the preservation copy.

A common approach to face this problem is the use of error detection codes, for instance hashing techniques such as MD5 that are computed over the complete file and help identifying changes in the bit flow. In order to highlight the actual temporal positioning of these changes, we propose to enrich the metadata extracted from images and videos of the carrier with an *audio fingerprint* of the audio signal. A fingerprint is a unique set of features automatically extracted from the audio signal that aims at the identification of digital copies, even in presence of noise, distortion, and compression. To this end, a fingerprint can be considered as a content-based signature that summarizes an audio recording. It is important to note that, although robust to noise, typical audio fingerprinting techniques can measure the difference between the original signal and the distorted copies.

Although usually aimed at digital rights management, being a compact representation of the audio signal, fingerprinting can find useful applications also in the development of music digital libraries other than tracking the diffusion of illegal copies of protected material. In particular, it can be useful to align different audio files of the same re-recording procedure, for instance the high quality audio which is the main goal of the A/D conversion and the low quality audio embedded in the video capture. Moreover, periodic extraction and comparison of the fingerprints can detect the exact time positioning of errors in the preservation copy due to aging of the digital carrier. Finally, we propose that fingerprinting can be used to measure the difference between the preservation and the access copies, because they are both originated from the same audio file.

Another technique that is worth mentioning, and which is often considered an alternative to audio fingerprinting, is *audio watermarking*. In this case, research on psychoacoustics is exploited to embed in a digital recording an arbitrary message, the watermark, without altering the human perception of the sound. The message can provide contextual information about the recording (such as title, author, performers), the copyright owner, and the user that purchases the digital item. Also in this case, this latter information can be useful to track the responsible of an illegal distribution of digital material. Similarly to fingerprints, audio watermarks should be robust to distortions, additional noise, A/D and D/A conversions, and compressions. Yet, the message that can be inserted through non-audible watermarking is still limited, and thus this technique cannot be used for embed complex information into the signal. Surely, audio watermarking should be used to add a unique identifier at least to any access copy.

12.4.3.4 Descriptive card

The data stored in the preservation copy can be easily copied onto new digital carriers. As digital carriers have a incremented data storage capacity (optical carriers: HD-DVD and Blue-ray Disc up to 50 GB of storage; cartridge digital magnetic tapes up to 800 GB of storage; HDD with some TB), the modification of the data organization in the preservation copies is to be expected, by introducing more documents into the same carrier and by adopting a different logical structure. For this purpose, it is necessary to provide the preservation copy with a list of all the documents belonging to the preservation copy, some metadata of the audio signal, and a description of the analogue original document. In our experience, it is necessary a descriptive card composed, at least, by five elements:

1. Heading;
2. Description of the preservation copy;
3. List of the documents stored in the preservation copy;
4. Description of the original document;
5. Description of the video recording.

12.5 Automatic Metadata Extraction

The increased dimensionality of the data contained within an audio digital library, which has been explained in the previous section, should be dealt with by means of automatic annotation. The auditory information contained in the audio medium can be augmented with cross-modal cues. For instance, the visual and textual information carried by the cover, the label and other attachments should be acquired through photos and/or videos. The extraction of this valuable information can be performed through well-known techniques for image and video processing, such as OCR, video segmentation and so on. We believe that it is interesting as well, even if not studied yet, to deal with other visual information regarding the carrier corruption and imperfection occurred during the A/D transfer.

Computer vision algorithms and techniques can be applied to the automatic extraction of relevant metadata. This section presents a set of tools able to extract, automatically, metadata from photos and video recordings of magnetic tape and phonographic disc.

12.5.1 Reel to Reel Magnetic Tape

The auditory information contained in the audio medium can be augmented with cross-modal cues. For example, a video of a winding tape can document its state of preservation and record precious information



such as the presence of splices and marks. Regarding video, well-known techniques such as change detection by background subtraction can be applied to detect discontinuities as seen in Fig. 10.5. In this case, I have employed background subtraction with automatic thresholding and a voting step to detect major changes in the image due to the presence of different materials (i.e., magnetic vs. header tapes).

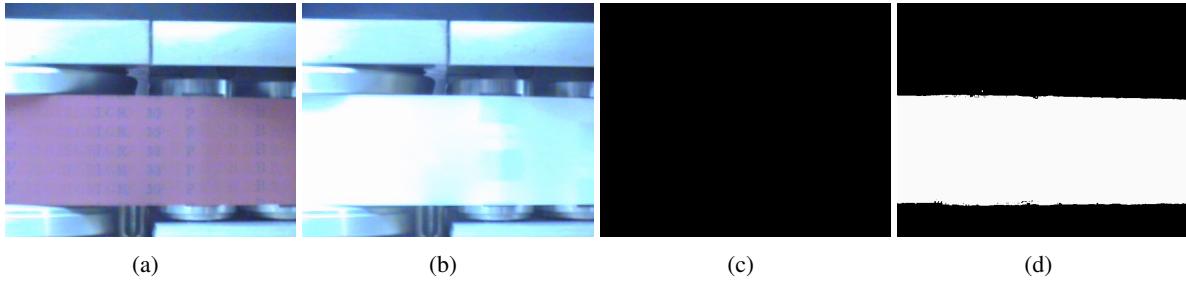


Figure 12.5: (a) and (b) show source frames from the video of a winding tape, while (c) and (d) show the corresponding processed images.

Fig. 10.5(c) is completely black as no significant changes have been detected between the current frame of Fig. 10.5(a) and the background image. In Fig. 10.5(d) a major change has occurred (white pixels) in the source frame shown in Fig. 10.5(b) (tape without magnetic layer). Therefore, the automatic detection of the start of a magnetic tape can be performed in a very simple and effective way via the processing steps mentioned above and by setting a threshold on the percentage of changed pixels with respect to the Region Of Interest (ROI). The ROI could be set in order to focus the algorithms only on a subregion of the image. As it can be seen in the source frames of Fig. 10.5, the tape occupies roughly 50% of the image, while other details such as the player's heads are not relevant for the processing and should be discarded by setting a ROI on the tape region. The approach described above is very similar to the techniques used for scene cut detection for automatic annotation of video sequences.

Fig. 10.6 shows how other information can be extracted by processing the videos of a winding tape. The basic processing steps are the same employed in the previous experiment, additional steps are required to detect splices or specific marks. In Fig. 10.6(b) no significant changes are detected, the image is not completely black but detected changes do not form a connected component large enough to pass the threshold.

Fig. 10.6(d) shows how a tape splice can be detected. The Hough transform is applied to detect lines in the subregions where changes have been detected. As it can be seen, the transform detects a line corresponding to the tape splice. In Fig. 10.6(f) a connected component corresponding to the dot in Fig. 10.6 e) is detected. The system can therefore annotate the corresponding frame linking it to the specific sound event marked by the felt-tip pen sign.

12.5.2 Warped Phonographic Discs

The characteristics of the arm's oscillations can be related to pitch variation of the audio signal. As such, they constitute valuable metadata for audio signal restoration processes. Also in this case, computer vision techniques can be applied to the automatic analysis of rotating discs. We have employed a feature tracking algorithm known as the Lucas-Kanade tracker. The algorithm locates feature points on the image to be tracked between consecutive frames. The technique, initially conceived for image registration, is here employed as a feature tracker to keep track of the position of the features from a frame to the following one. Fig. 10.7 shows some frames from one of the sequences used in the experiments: (b) shows the lowest position of the arm's head in one oscillation and (c) the highest position, where the

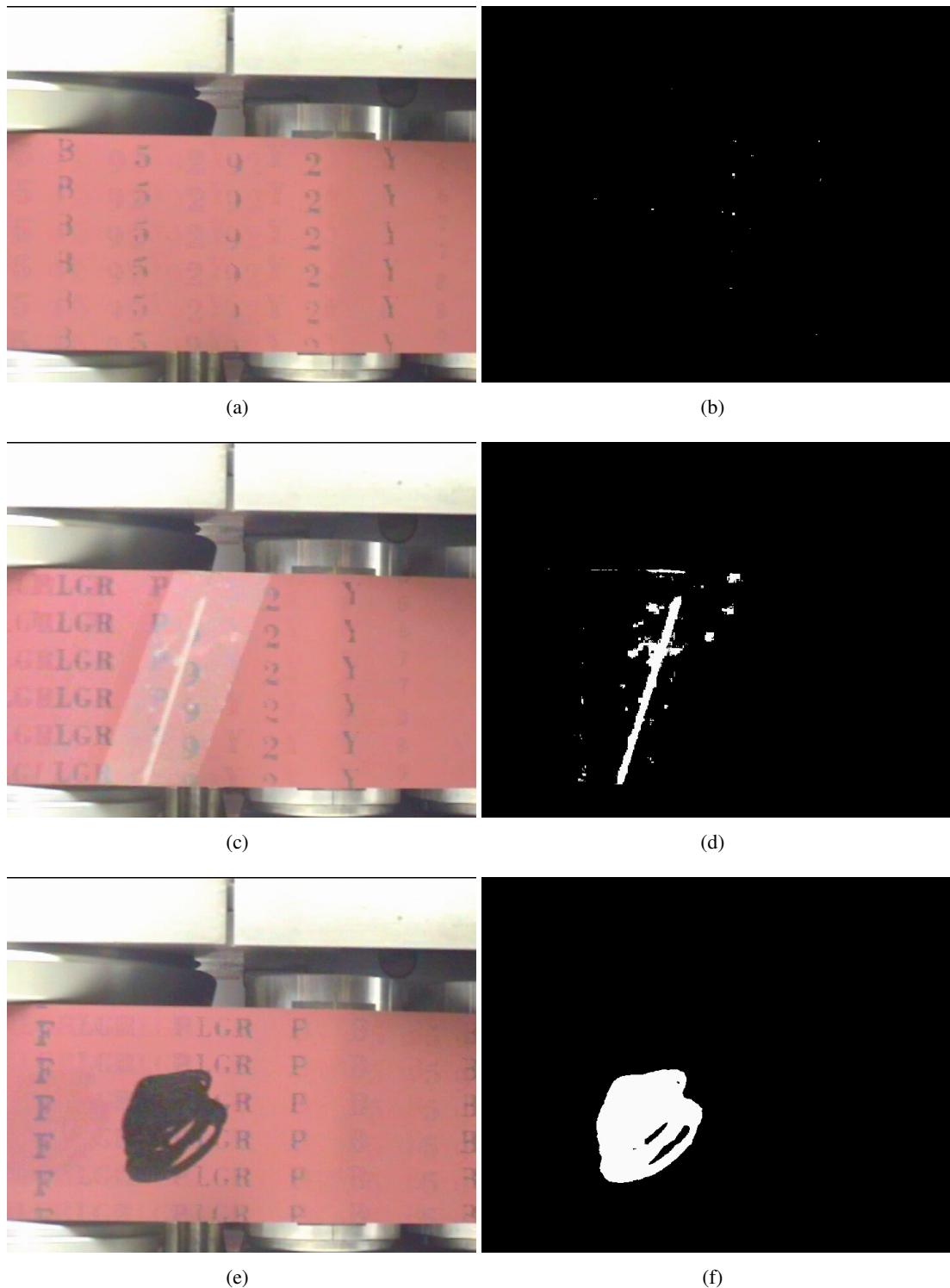


Figure 12.6: Automatic discontinuities extraction from a winding tape (splices, marks).



Figure 12.7: Processed frames from a video of an oscillating record player's arm. (a) Photo of the turntable arm; (b) Lowest position of the arm in an oscillation, (c) its highest position. (b) and (c) show Lucas-Kanade features detected on the arm's head and tracked through the oscillation. (d) shows the differences between lowest and highest positions.

Lucas-Kanade features can be seen on the arm's head while being tracked through the oscillation. Even if from the Fig. 10.7 the differences between the highest and lowest positions are almost unnoticeable (see the differences between them in (d)), our approach is able to track them clearly, as shown in Fig. 10.8.

Fig. 10.8 shows the temporal evolution of the y coordinate of a feature located on the arm's head. The x-axis shows the number of frames and the y-axis reports the position in pixels on the image plane. The oscillatory evolution is clearly visible. There is a 29 frames gap between Fig. 10.7(b) and Fig. 10.7(c), which is consistent to the period of the oscillations shown in Fig. 10.8.

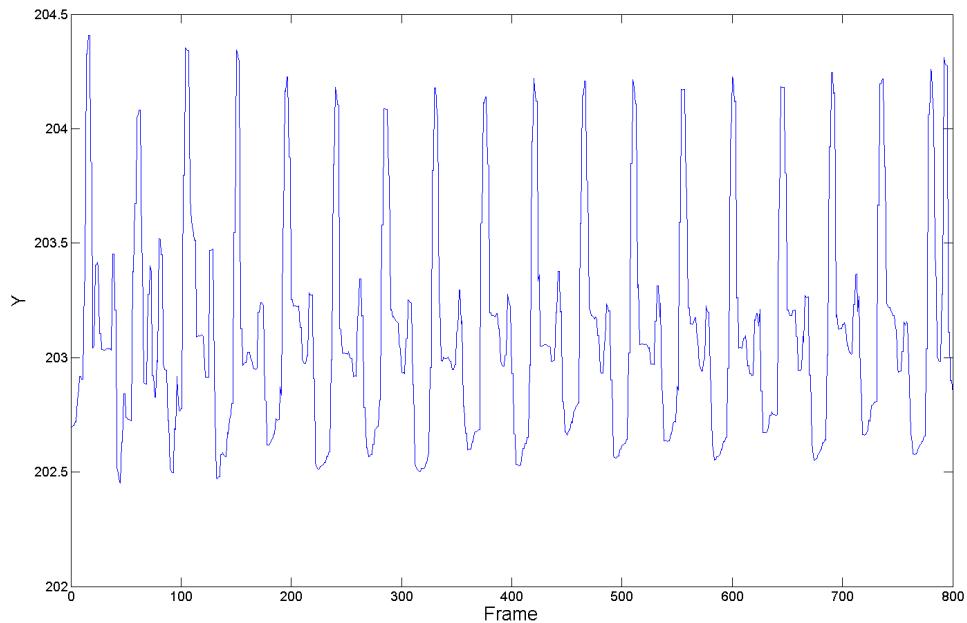


Figure 12.8: Temporal evolution of the y coordinate of a Lucas-Kanade feature located on the arm's head. It can be seen clearly how the oscillations indicate a deformed disc.

12.5.3 Off-centered Phonographic Disc

Interesting properties of a phonograph record can be automatically extracted by analyzing a picture of it. For example, we wanted to calculate the eccentricity of the disc, that is, the offset between the spindle hole axis and the exact central rotation axis. This production flaw, which could affect individual copies or entire stocks of records, is responsible for the well-known warp effect that introduces a pitch variation

in the audio signal. To accomplish this automatically I have exploited the consolidated literature on iris detection. Since our problem shares the same lucky circular properties of the problem of iris detection, we have employed the integro-differential operator which was developed for detecting the pupillary boundary and the outer boundary of the iris. The integro-differential operator has the following form:

$$\max_{(r,x_0,y_0)} \left| G_\sigma(r) * \frac{\partial}{\partial r} \oint_{r,x_0,y_0} \frac{I(x,y)}{2\pi r} ds \right| \quad (12.1)$$

The operator is computed over the image $I(x, y)$ where it searches for the maximum of the blurred partial derivative, with respect to the radius r , of the normalized circular integral of radius r and center coordinates (x_0, y_0) calculated on $I(x, y)$. The blur is obtained through convolution with a Gaussian smoothing function of scale σ . In other words, the operator works as circular edge detector and provides the centre coordinates and the radius of the strongest circular edge in the image. The outer contour of the disc is extracted and then the operator on the image for detecting the spindle hole contour is rerun. The second pass can be computed very fast as it takes advantage of the known geometrical properties of vinyl discs. That is, once the outer boundary has been detected the spindle hole contour can be searched in a subregion of the image inside the outer contour. The disc was laying on a plane parallel to the image and the spindle hole was on-axis with the camera's optical axis. Although this constraint is not particularly restrictive for a dedicated set-up in an audio laboratory, a step further can be taken by removing this assumption and considering perspective deformations given by out-of-axis images.

Having detected the outer boundary of the disc and the spindle hole contour, the calculation of the offset between their centers is trivial. In the author experience, the estimated offset can be greater than 1 cm. The processing described in this subsection can be performed on-line in real-time. The experiments shown in Fig. 10.5, Fig. 10.6 and Fig. 10.7, have been carried out on off-line 320x240 resolution video sequences with an above real-time frame rate processing performance of 50 frames/sec on a 3 GHz single processor machine. The application has been coded in C++. In addition, no particular setup was required for this experiment. Video sequences have been acquired with a consumer digital camcorder at PAL resolution and subsequently rescaled and compressed into DivX video files at medium-high quality setting. As can be seen comparing Fig. 10.5, Fig. 10.6 and Fig. 10.7, the algorithms are robust to different lighting conditions. The achieved results hint the possibility to perform tape marks detection in real-time, as the tape is winding. This would be a practical set-up for audio laboratories and audio digital libraries.

12.5.4 Representing Metadata

Once all this content-dependent information has been extracted, a suitable metadata schema for its representation has to be chosen for its representation. Among the existing metadata standards, probably the Metadata Encoding and Transmission Standard (METS) is particularly suitable for representing the information about the carriers and the A/D transfer. It can be noted that METS has already been used to encode music documents with profiles for both scores and sound recordings, for instance in the Digital Library of the Brown University. The, METS documents have two sections that are particularly significant for the aims of this study: the *File Section* allows us to keep information about additional files, which is particularly significant since also the extracted metadata is in the form of additional multimedia documents, and the *Structural Map* that can represent the hierarchy between different metadata, for instance ranging from the the video capture of the A/D transfer of a warped phonographic disc, to the tracking of feature points on the pickup, to the representation of the movement of the pickup along the vertical axis, as explained in Sect. 10.5.2.

As it is well-known, another suitable schema for music documents is MPEG. In particular, MPEG-7 can easily represent the description, the definition and the content of extracted metadata as accompanying features of the audio digital object. The application of MPEG-7 seems particularly appealing because of



its ability to describe low-level characteristics, as the ones extracted automatically from the images of the carrier and the video of the A/D transfer. The XML-based structure of MPEG-7 allows a straightforward extension to include the multimedia material and the results of the analysis techniques presented in this and in the following sections. Yet, a discussion of the metadata schema is beyond of the scope of this paper.

12.6 Audio Data Extraction and Alignment from Phonographic Disc

This section introduces: a) a system for reconstructing the audio signal from a still image of a phonographic disc surface; b) alignment techniques useful in the comparison of alternative digital acquisitions. A case study where the alignment tool is used to annotate disc corruptions is described in the following section.

12.6.1 Photos of GHOSTS (PoG)

Nowadays, automatic text scanning and optical character recognition are in wide use at major libraries. Yet, unlike text scanning, A/D transfer of historical sound recordings is often an invasive process.

As it is well-known, several phonographs exist that are able to play gramophone records using a laser beam as pickup (laser turntable). This playback system has the advantage of never physically touch the record during playback: the laser beam traces the signal undulations in the record, without friction. Unfortunately, the laser turntables are constrained to the reflected laser spot only and are susceptible to damage and debris and very sensitive to surface reflectivity.

Digital image processing techniques can be applied to the problem of extracting audio data from recorded grooves, acquired using a digital camera or other imaging system. The images can then be processed to extract audio data. Such an approach offers a way to provide non-contact reconstruction and may in principle sample any region of the groove, also in the case of a broken disc. These scanning methods have several advantages: a) delicate samples can be played without further damage; b) broken samples can be re-assembled virtually; c) the re-recording approach is independent from record material and format (wax, metal, shellac, acetates, etc.); d) effects of damage and debris (noise sources) can be reduced through image processing; e) scratched regions can be interpolated; f) discrete noise sources are resolved in the “spatial domain” where they originate rather than being an effect in the audio playback; g) dynamic effects of damage (skips, ringing) are absent; h) classic distortions (wow, flutter, tracking errors, etc) are absent or removed as geometrical corrections; i) no mechanical method is needed to follow the groove; l) they can be used for mass digitization.

In the literature, there are several approaches to this problem, based on: Digital Cameras (2D or horizontal only view, frame based); Confocal Scanning (3D or vertical+horizontal view, point based); Chromatic sensors (3D, point based); White Light Interferometry (3D, frame based). The authors have developed the Photos of GHOSTS (PoG) system that: a) is able to recognize different rpm and to perform track separation automatically; b) does not require human intervention; c) works with low-cost hardware; d) is robust with respect to dust and scratches; e) outputs de-noised and de-wowed audio, by means of novel restoration algorithms. The user can choose to apply an equalization curve among the hundreds stored in the system, each one with appropriated references (date, company, roll-off, turnover). Moreover, PoG allows the user to process the signal by means of several audio restoration algorithms. The software automatically finds the record centre and radius from the scanned data, for groove rectification and for track separation. Starting from the light intensity curve of the pixels in the scanned image, the groove is modeled and the audio samples are obtained. The complete process is depicted in Fig. 10.9.

The system enhancements include:



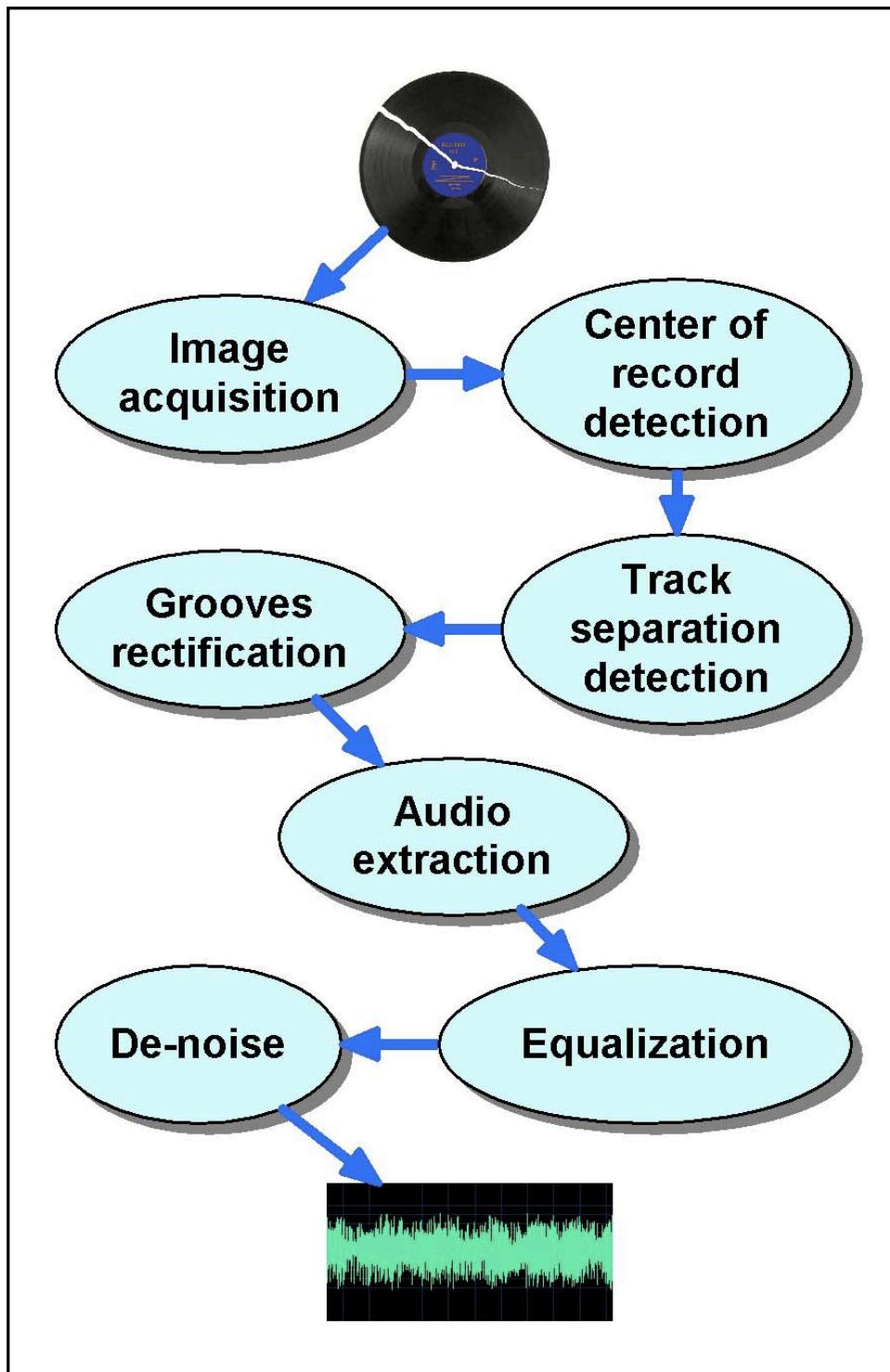


Figure 12.9: Photos of GHOSTS schema.

1. the user can select the correct equalization in a list including 225 different curves, able to cover all the electric recordings, since 1925.
2. A de-noise algorithm in a frequency domain⁸ based on the use of a suppression rule, which considers the psychoacoustics masking effect. The spreading thresholds which present the original signal $x(n)$ are not known *a priori* and are to be calculated. This estimation can be obtained by applying a noise reduction STSA standard technique leading to an estimate in the frequency domain of $x(n)$, for which the masking thresholds m_k , defined as the non negative threshold under which the listener does not perceive an additional noise, can be calculated by using an appropriate psychoacoustic model. The masking effect obtained is incorporated into one of the EMSR technique, taking into consideration the masking thresholds m_k for each k frequency of the STFT transform. A cost function depending on m_k , which minimization gives the suppression rule for the noise reduction, is created. This cost function can be a particularization of the mean square deviation to include the masking thresholds, under which the cost of an error is equal to zero.
3. The design and the realization of ad-hoc prototype of a customized scanner device with a rotating lamp carriage in order to position every sector with the optimal alignment relative to the lamp (coaxially incident light). In this way we improved (from experimental results: more than 20%) the accuracy of the groove tracking step.

PoG may form the basis of a strategy for: a) larger scale A/D transfer of mechanical recordings which retains maximal information (2D or 3D model of the grooves) about the native carrier; b) small scale A/D transfer processes, where there are not sufficient resources (trained personnel and/or high-end equipments) for a traditional transfer by means of turntables and converters; c) the active preservation of carriers with heavy degradation (breakage, flaking, exudation).

12.7 Audio restoration

The audio restoration algorithms can be divided into three categories:

1. frequency-domain methods, such as various forms of non-causal Wiener filtering or spectral subtraction schemes and recent algorithms that attempt to incorporate knowledge of the human auditory system; these methods use little *a priori* information;
2. time-domain restoration by signal models such as Extended Kalman Filtering (EKF): in these methods a lot of *a priori* information is required in order to estimate the statistical description of the audio events;
3. restoration by source models: only *a priori* information is used.

The advantage of frequency-domain methods is that they are straightforward and easy to implement. However, the limitations are as follows: musical noise (short sinusoids randomly distributed over time

⁸ Audio restoration algorithms can be divided in three categories:

- (a) *frequency-domain* methods, such as various forms of noncasual Wiener filtering or spectral subtraction schemes and recent algorithms that attempt to incorporate knowledge of the human auditory system; these methods use little *a priori* information (only the Power Spectral Density noise estimation);
- (b) *time-domain* restoration by signal models such as Extended Kalman filtering: in these methods it is necessary a lot of *a priori* information in order to estimate the statistical description of the audio events;
- (c) restoration by *source models*: only *a priori* information is used.



and frequency) is unavoidable; the results depend on a good noise estimation. Restoration by source model is limited to very few cases (e.g., only monophonic recordings) and it is not generalizable. The EKF is able, in principle, to simultaneously solve the problems of filtering, parameter tracking and elimination of the outliers, but it is very sensitive to parameter setting (i.e., the order p of the AR model; the length q of the signal vector, the length of the initial training segment in the bootstrap procedure, the adaption speed λ , the forgetting factor γ , the threshold μ for detection of impulsive noise).

This section presents algorithms, developed at the Centro di Sonologia Computazione (Dept. Information Engineering) using the VST plug-in architecture, able to offer satisfying examples of the above mentioned categories. The algorithms are detailed in the next subsections:

- CREAK (Canazza REstoration Audio - extended Kalman filter): A de-noise and de-click system based on Extended Kalman Filter, dedicated to the restoration of audio signal re-recorded from shellac discs: low Signal to Noise Ratio (SNR), clicks, pops, crackle.
- CMSR (Canazza-Mian Suppression Rule): A de-noise algorithm based on STSA (Short Time Spectral Attenuation), dedicated to the restoration of audio signal re-recorded from wax and amberol cylinders and shellac discs: low SNR.
- PAR (Perceptual Audio Restoration): A de-hiss based on perceptual algorithm for reel-to-reel tapes and cassettes: high SNR.

Of course, regardless their dedications, in a real restoration work it is opportune to combine these tools in order to obtain the better results.

12.7.1 CREAK: A de-noise and de-click system dedicated to shellac discs

In this tool we employ an algorithm whose objective is to simultaneously solve the problems of filtering/parameter tracking/elimination of the outliers (“clicks”) by using the Extended Kalman Filter theory (EKF), as proposed by M. Niedzwiecki and K. Cisowski. In particular the algorithm can be interpreted as the nonlinear combination of two Kalman filters: the first is used to follow the slow variations of the signal time-varying AR model parameters, while the second takes part in the reduction of background and impulsive noise. Because the old analogue discs (in particular: shellac discs) are corrupted by a broadband noise and by a large amount of impulsive disturbances (pops, clicks and crackle), this algorithm is suitable for these carriers. In order to achieve maximum performance from the EKF, it is essential to optimize its implementation. For this purpose, to cope with the non-stationary nature of the audio signal, we used two properly combined EKF filters (forward and backward), and introduced a bootstrapping procedure for model tracking. The careful combination of the proposed techniques and an accurate choice of some critical parameters, allows to improve the performance of the EKF algorithm.

12.7.1.1 Bootstrap procedure

The first problem we deal with is the choice of the filter initial conditions. To this purpose, let us notice first that starting the algorithm from scratch implies an initial transient of the parameter tracker during which the EKF noise reduction capabilities are greatly reduced. To solve the problem, I’ve found useful to introduce a bootstrap procedure: the first 100 ms of the signal are time-reversed and fed to the filter. This way, parameters for a proper initialization of the model are estimated and restoration of the “true” signal will use these values as initial conditions.



12.7.1.2 Forward/backward filtering

The non-stationarity of the audio signal has an important consequence: the results of the forward and backward (reversing the time axes) filtering can be different. The algorithm is directional for its nature, that is, it uses the whole past history plus a finite number of future samples, depending on the model order. A provision that improves the algorithm performance is given by the use of two properly combined EKFs operating forward and backward on the signal.

It is clear that, with broadband noise, sharp changes in dynamics of the music signal are treated in a more effective way if they are “covered downhill” (i.e., passing from loud to soft intensity), independently from the direction of the filter. This is due to the fact that the estimate of the EKF benefits from having a signal segment with a better local Signal to Noise Ratio, before the transition loud/soft.

The comparison between the residuals of the forward and backward filtering shows that the former works better than the latter at the end of the restored segment, and that the opposite situation holds in the initial zone.

Furthermore the forward/backward strategy improves the detection of impulsive disturbances: indeed, it can happen that the clicks are identified (and removed) in a more effective way in one direction than in the other.

Since the two filters give different signal estimates, $\hat{s}_+(t)$ (forward) and $\hat{s}_-(t)$ (backward), we found it effective to combine them according to:

$$\hat{s}_w(t) = \frac{\hat{\sigma}_{\varepsilon-}^2(t)}{\hat{\sigma}_{\varepsilon+}^2(t) + \hat{\sigma}_{\varepsilon-}^2(t)} \hat{s}_+(t) + \frac{\hat{\sigma}_{\varepsilon+}^2(t)}{\hat{\sigma}_{\varepsilon+}^2(t) + \hat{\sigma}_{\varepsilon-}^2(t)} \hat{s}_-(t) \quad (12.2)$$

The basic idea is to weigh $\hat{s}_+(t)$ and $\hat{s}_-(t)$ in a way that is inversely proportional to signal variance $\hat{\sigma}_{\varepsilon}^2$.

With such a provision it is possible, in the author’s experience, to effectively remove broadband noise in audio signal with low SNR and, since we have two different click detectors, the effectiveness in removing impulsive disturbances is also improved. In this sense, it is particularly well-suited for the restoration of analogue discs.

12.7.2 CMSR: A de-noise algorithm dedicated to wax and Amberol cylinders and shellac discs

The most widespread techniques (Short Time Spectral Attenuation, STSA) employ a signal analysis through the Short-Time Fourier Transform (which is calculated on small partially overlapped portions of the signal) and can be considered as a non-stationary adaptation of the Wiener filter in the frequency domain. The time-varying attenuation applied to each channel is calculated through a determined *suppression rule*, which has the purpose of producing an estimate (for each channel) of the noise power. A typical suppression rule is based on the Wiener filter: usually the mistake made by this procedure in retrieving the original sound spectrum has an audible effect, since the difference between the spectral densities can give a negative result at some frequencies. Should we decide to arbitrarily force the negative results to zero, in the final signal there will be a disturbance, constituted of numerous random frequency pseudo-sinusoids, which start and finish in a rapid succession, generating what in literature is known as *musical noise*.

More elaborated suppression rules depend on both the relative signal and on *a priori* knowledge of the corrupted signal, that is to say, on *a priori* knowledge of the probability distribution of the under-band signals. A substantial progress was made with the solution carried out in Ephraim and Malah, that aims at minimizing the mean square error (MSE) in the estimation of the spectral components (Fourier coefficients) of the musical signal. The gain applied by the filter to each spectral component does not



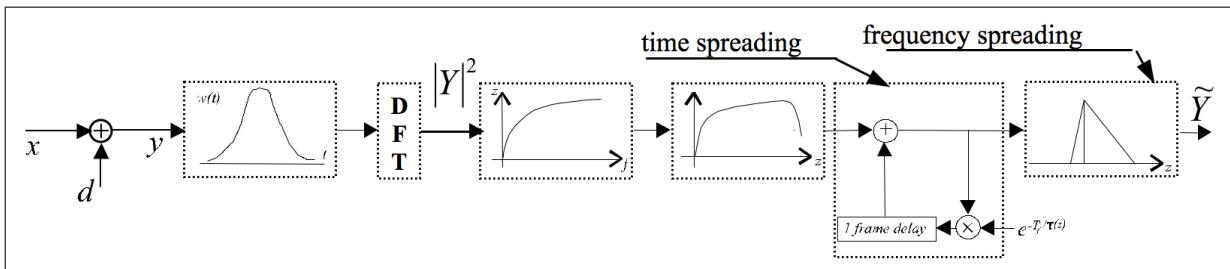


Figure 12.10: The audio signal transformation from “outer” to “inner representation. The signal $x(n)$ is first windowed by the $w(n)$ window and transformed in the frequency domain. The short time spectral power is transformed from Hertz (f) to Bark (z) scale, band-limited and spread both in time and frequency.

depend on the simple Signal to Noise Ratio (Wiener Filter), but it is in relation with the two parameters Y_{prior} (SNR calculated taking into account the information of the preceding frame) and Y_{post} (SNR calculated taking into account the information of the current frame). A parameter (α) controls the balance between the current frame information and that of the preceding one. By varying this parameter, the filter smoothing effect can be regulated. Y_{prior} has less variance than Y_{post} : this way, musical noise is less likely to occur.

Unfortunately, in the case of cylinders or shellac discs an optimal value of α does not exist, as it should be time-varying (because of the cycle-stationary characteristics of the cylinder/disc surface corruptions). Considering this, the author has developed a new suppression rule (Canazza-Mian⁹ Suppression Rule, CMSR), based on the idea of using a *punctual* suppression without memory (Wiener like) in the case of a null estimate of Y_{post} , according to:

$$\alpha = \begin{cases} 0.98, & \text{if } Y_{post}(k, p) > 0 \\ 0, & \text{otherwise.} \end{cases} \quad (12.3)$$

The experiments carried out confirm that the filter performs very well, with a noise removal decidedly better than other suppression rules (e.g., classic EMSR) and with the advantage of not introducing musical noise, at least for $SNR \in [0 \div 20]$ dB (a typical value in the audio signal re-recorded from the cylinder and shellac discs). Furthermore, the behavior in the transients is similar of the EMSR filter, without having the perceptual impression of a processing “low-pass filter” like.

12.7.3 PAR: A de-hiss perceptual algorithm dedicated to reel-to-reel tapes and cassettes

This tool considers the perceptually relevant characteristics of the signal. This way, within model fidelity, only the audible noise components are removed in order to preserve the signal from possible distortions caused by the restoration process. In this sense, this method is particularly suitable for the restoration of signals with a high SNR ($SNR > 20$ dB).

To filter the noise in a perceptually meaningful way, it is necessary to transform the audio signal from an “outer” to “inner” representation, i.e., into a representation that takes into account how the sound waves are perceived by the auditory system. The device used is the Beerends and Stemerdink model, sketched in Figure 10.10. The signal $x(n)$ is first windowed by the $w(n)$ window and transformed in

⁹Gian Antonio Mian (1942-2006) was a professor of Digital Signal Processing at the Dept. of Information Engineering, University of Padua, a leading researcher in our department, and an outstanding teacher whose brightness and kindness I will always remember. These results are affectionately dedicated to his memory.

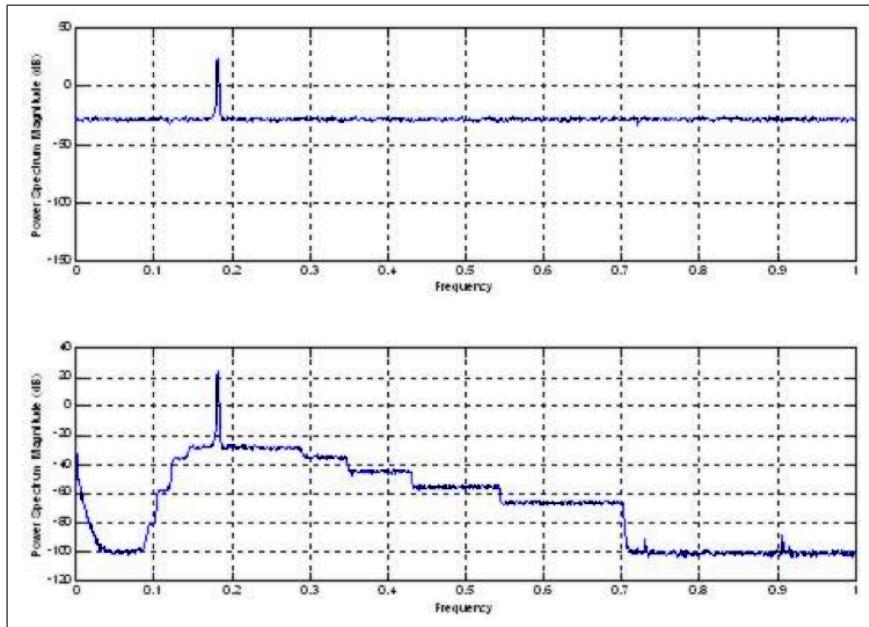


Figure 12.11: A sinusoid with broadband noise (top) and after the perceptual de-noise (bottom). Only the audible noise components are removed. X-axis: frequency normalized to the Nyquist frequency; Y-axis: Power Spectrum Magnitude (dB).

the frequency domain. The short time spectral power is transformed from Hertz (f) to Bark (z) scale, band-limited and spread both in time and frequency.

As a result, the outer frequency domain representation $Y(p, f) = X(p, f) + D(p, f)$, with X and D signal and noise spectrum estimates, is transformed into the internal representation $\tilde{Y}(p, z) \approx \tilde{X}(p, z) + \tilde{D}(p, z)$, defined in the Bark domain, band-limited and processed taking into account the spreading both in time and frequency. Finally, the \tilde{Y}_{prio} and \tilde{Y}_{post} terms (see Sec. 10.7.2) are calculated according to the inner representation and the gain $\tilde{G}(p, z)$ is derived. Figure 10.11 shows a representative example: a sinusoid with broadband noise (top) and after the perceptual de-noise (bottom), in which it can be noticed that only the audible noise components are removed.

12.7.4 Experimental results

A series of experiments with real usage data from different international audio archives were conducted. In this section experimental results of applying the above described techniques related to audio restoration are presented. As first case study, Figure 10.12 shows a restoration of a wax cylinder by means of CMSR (see Sec. 10.7.2). The song is *My Mariuccia take-a steamboat*, performed by Billy Murray (vocal tenor) in 1906. Edison Gold Moulded Record: 9430; cylinder length: 2' 13". It is a comic song in Italian dialect with orchestra accompaniment. In Figure 10.12: at the top there is the waveform of the original (corrupted) audio extract, at bottom, the restored data by means of CMSR. Only a de-noise is performed. An increase of SNR can be noticed.

Considering impulsive disturbances, the Figure 10.13 shows a de-click of a shellac disc by means of CREAK (see Sec. 10.7.1). The song is *La signorina sfinciosa* (The funny girl), performed by Leonardo Dia. Shellac 78 rpm 10", Victor V-12067-A (BVE 53944-2); disc length: 3' 19". The lyrics are in an Italian dialect, with the musical accompaniment of a mandolin (Alfredo Cibelli) and two guitars (unknown players). Recorded in New York, July, 24th, 1929. In Figure 10.13 is pointed out a click,



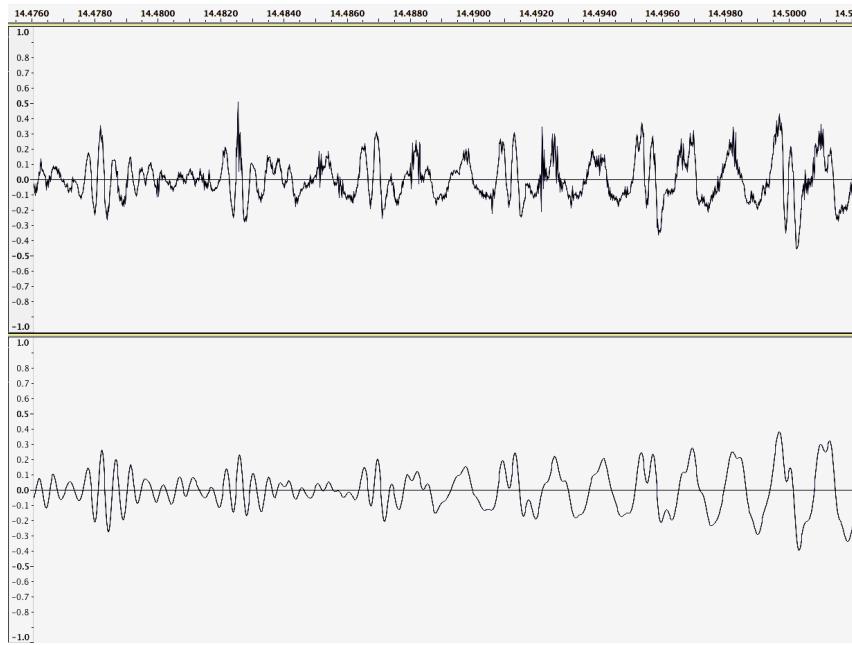


Figure 12.12: Top: the waveform of the original (corrupted) audio extract. Bottom: the reconstructed data by CMSR. The increase of SNR can be noticed. X-axis: time (s). Y-axis: amplitude (normalized).

before (top) and after (bottom) the audio restoration performed by CREAK.

In a third case study, a tape recording (unpublished) of Portuguese fado music (from the audio archive of the Universidade Nova de Lisboa – Faculdade de Ciencias Sociais e Humanas, Portugal¹⁰) is considered. In this case we performed only a de-noise by means of PAR tool. Figures 10.14 and 10.15 show the corrupted (top) and restored (bottom) signals respectively in time and frequency domains of two different (representative) excerpts of the musical piece.

Finally, an example of a combined methods is presented. We consider the shellac disc *Nofrio e la finta americana*, performed by Giovanni De Rosalia and Francesca Gaudio (vocals). Shellac 78 rpm 10", Victor 72404 B (B 22911-2); disc length 2' 40". Recorded in New York, June, 11th, 1919. In this case, we carried out de-click and de-noise by means of CREAK. Because of the low SNR (SNR \sim 5 dB), we processed the signal also with CMSR. In this way, we obtained a SNR = 40 dB¹¹, without introducing particularly audible distortions (musical noise). Figures 10.16 and 10.17 show the corrupted (top) and restored (middle and bottom) signals respectively in the time and frequency domains.

12.7.4.1 Comparison

Figure 10.18 shows the gain trend introduced by the filters described above in comparison with some *standard* filters (Wiener filter, Power Subtraction, EMSR) at the varying of the noisy signal SNR, considering 35 carriers of ethnic music (20 shellac discs recorded from 1910 to 1930, 6 wax cylinders recorded from 1900 and 1914, and 9 open-reel music tapes recorded from 1960 to 1975). The term gain indicates the difference between the de-noised signal SNR and the input signal SNR. As can be noticed, all the three filters have a good performance, in particular CMSR for signal with low or medium SNR, CREAK for SNR [15 \div 30] dB and PAR seems adapt to reduce the noise in audio signal with a good SNR.

¹⁰The author would like to thank Salwa Castelo Branco for sharing the audio documents of the archive.

¹¹The measuring of noise power is made by taking the noise print in an interval where there is only background noise.



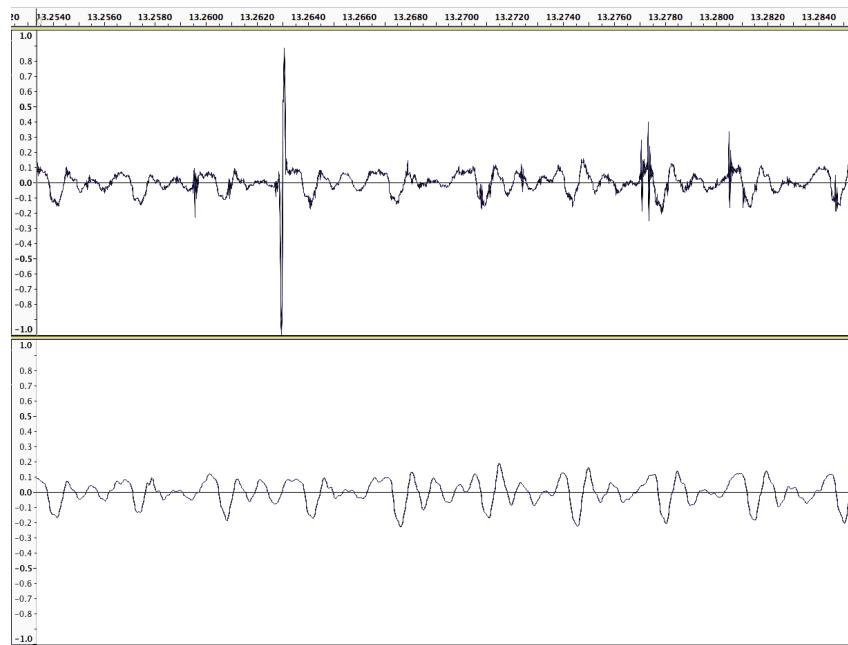


Figure 12.13: Top: the waveform of the original (corrupted) audio extract. Bottom: the reconstructed data by CREAK. The click removal can be noticed. X-axis: time (s). Y-axis: amplitude (normalized).

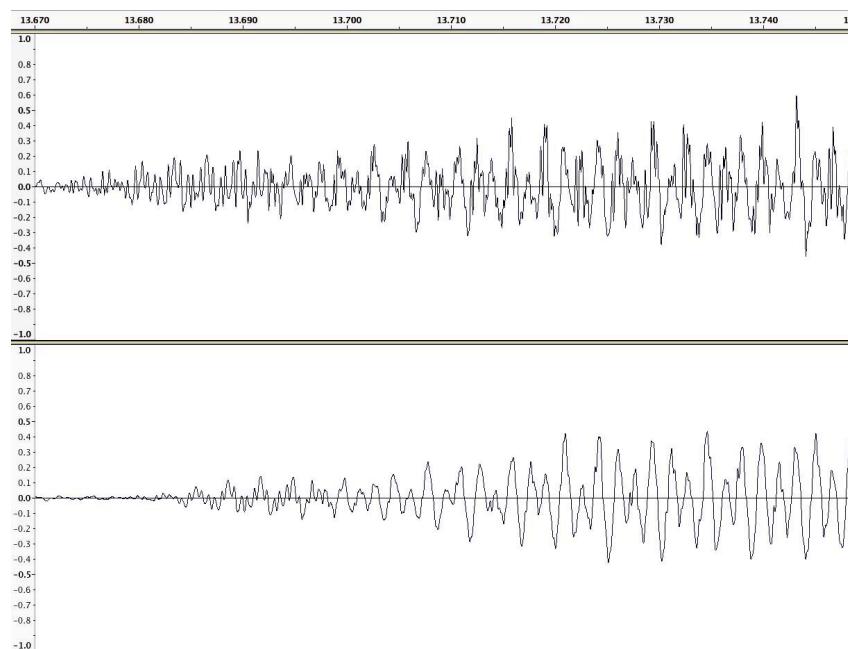


Figure 12.14: Top: the waveform of the original (corrupted) audio extract. Bottom: the restored data by PAR. X-axis: time (s). Y-axis: amplitude (normalized).



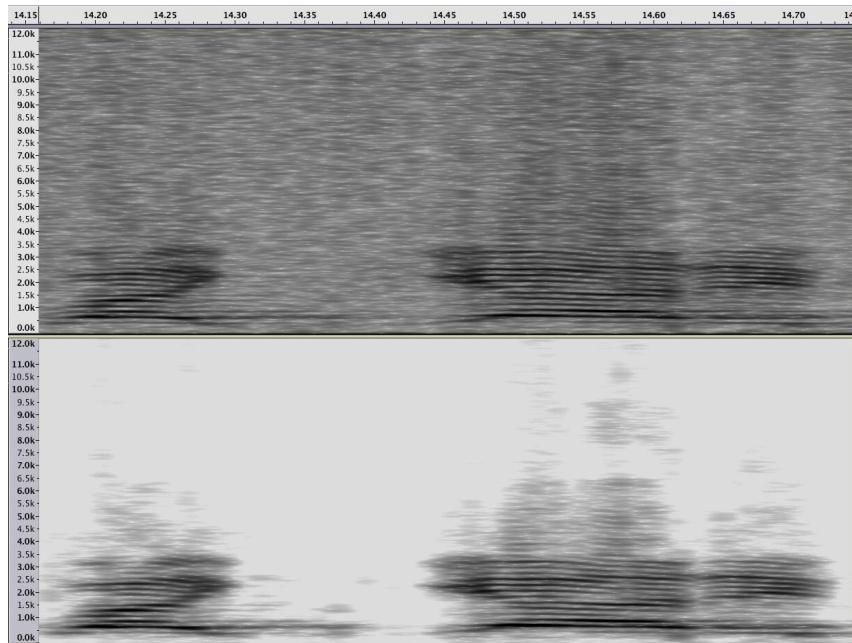


Figure 12.15: Top: the spectrum of the original (corrupted) audio extract. Bottom: the restored data by PAR. X-axis: time (s). Y-axis: frequency (Hertz).

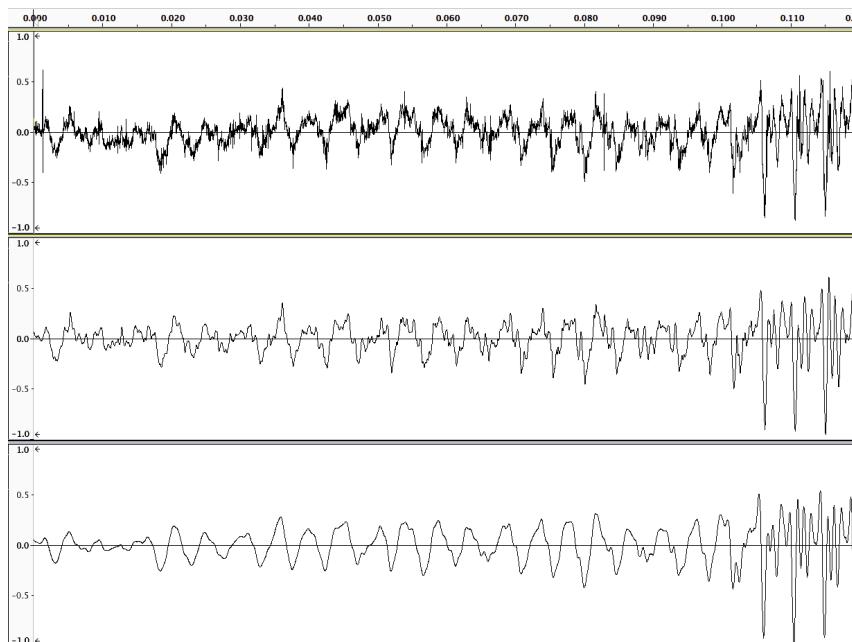


Figure 12.16: Top: the waveform of the original (corrupted) audio extract. Middle: de-clicked and de-noised by CREAK. Bottom: de-noised by CMSR. X-axis: time (s). Y-axis: amplitude (normalized).

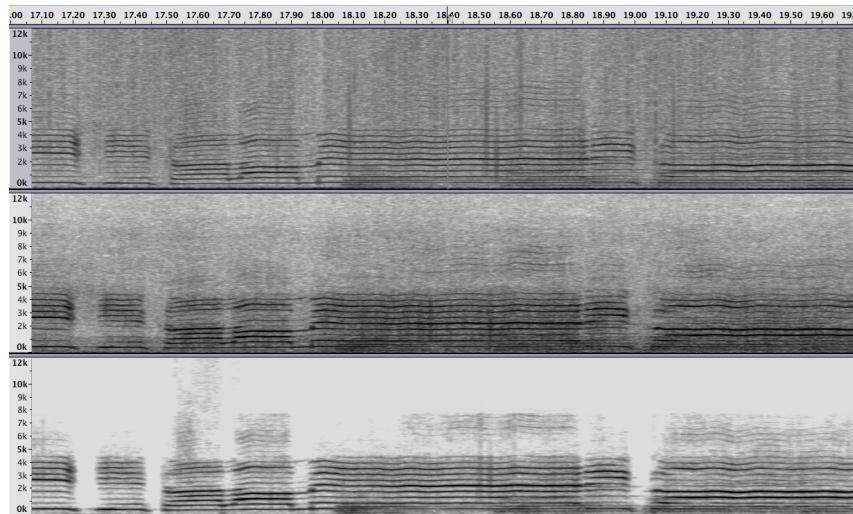


Figure 12.17: Top: the spectrum of the original (corrupted) audio extract. Middle: de-clicked and de-noised by CREAK. Bottom: de-noised by CMSR. X-axis: time (s). Y-axis: frequency (Hertz).

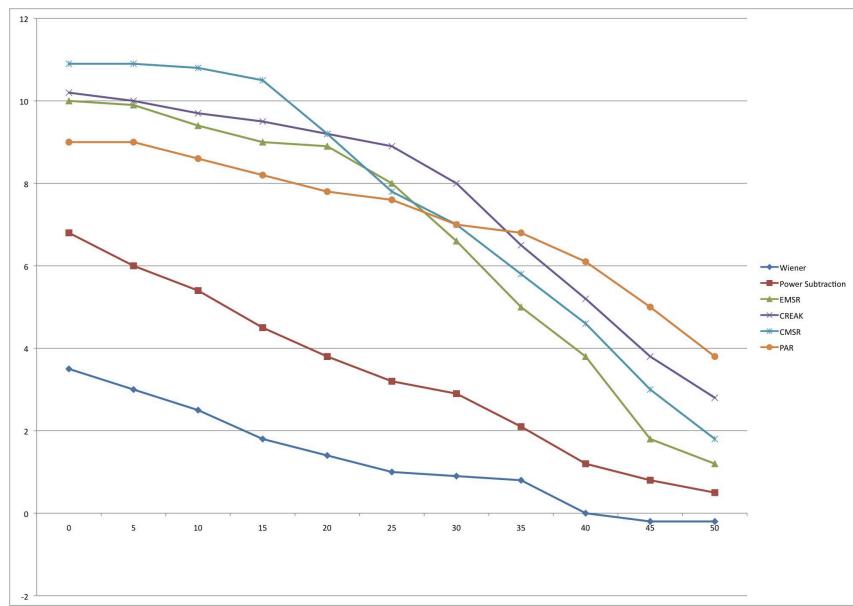


Figure 12.18: Gain trend introduced by the filters in the frequency domain at the varying of the input SNR ($SNR_{out}-SNR_{in}$ vs. SNR_{in} in dB). The three filters have a good performance: CMSR for signal with low or medium SNR; CREAK for SNR [15 ÷ 30] dB; PAR for high SNR.

12.7.5 Assessment

In order to show the methodology usually adopted in the field of audio enhancement and compression (e.g., by the MPEG group) to validate audio processing systems, a listening test carried out to validate the audio restoration algorithms detailed above is described.

Material. Three music pieces recorded in clearly different supports were used (see Sec. 10.7.4 for the details of these audio documents):

1. Cylinder: *My Mariuccia take-a steamboat*.
2. Shellac disc: *La signorina sfinciosa*.
3. Tape: unpublished recording of Portuguese fado music

In order to minimize the fatigue and maximize the attention of the participants, the first 20 seconds of each track were selected. Since the task was more a comparison than an individual analysis, those short extracts seemed to be sufficient.

Restoration of the noisy stimuli was performed by CREAK in de-click mode, then using CREAK, CMSR and PAR as well as through the following commercial products:

1. X-Click and X-Noise of Waves Restoration bundle (Waves V6 Update 2);
2. De-click and De-noise of Sonic NoNOISE plug-in suite (a software tool able to offer the same algorithms found in the SonicStudio version);
3. Declicker and Denoiser (enable its *Musical noise suppression* filter) of iZotope RX v1.06;
4. Auto Declick and Auto Dehiss of CEDAR Tools;
5. Adobe Audition 3.0;
6. Audacity 1.3.6 (an open source software for recording and editing audio signals).

The plug-ins by Sonic NoNOISE and by CEDAR are used in a Pro Tools HD 2 system. The parameters used to control the different systems were subjectively set to obtain the best tradeoff between noise removal and music signal preservation. This way, 27 restored stimuli were produced.

Test method. The tests were conducted using the EBU MUSHRA test method, which is a recommended evaluation method adopted by ITU. This protocol is based on “double-blind triple-stimulus with hidden reference” method, which is stable and permits accurate detection of small impairments. An important feature of this method is the inclusion of the hidden reference and two bandwidth-limited anchor signals (7 kHz and 3.5 kHz).

Training phase. The purpose of the training phase, according to the MUSHRA specification, was to allow each listener: i) to become familiar with all the sound excerpts under test and their quality-level ranges; ii) to learn how to use the test equipment and the grading scale.

Listeners. Two subject groups were selected: 12 researchers of Universities of Padua and Udine and 14 students of the Multimedia Communication Degree (University of Udine). All the subjects were musically trained.

Equipment. The audio signals were recorded at 96 kHz/24 bit (uncompressed sound files) and played through Apple iMac Intel Core 2 Duo with 2 GB 800 MHz DDR2 SDRAM (D/A converter: RME Fireface 400), and headphones (AKG K 501). The listeners could play all the stimuli under test any order they liked, including the hidden reference and the two bandwidth-limited anchor signals.

Test duration. The training session for each listener took approximately 1 hour, including an explanation about the tests and equipment, and a practice grading session. The grading phase consisted of 3



test sessions (one for each music piece), each one containing 12 test signals (1 noisy signal, 9 restored signals, 2 anchors). Each session took, on average, about 10 minutes. Subjects were allowed a rest period between each session, but not during a session.

Main results. The statistical analysis method described in the MUSHRA specification was used to process the test data. The results are presented in Tab. 10.4 as mean grades. The results from two listeners were removed because the mean of their rates (in absolute value) on hidden references was greater than $+/- 0.5$.

In *My Mariuccia take-a steamboat* CEDAR and CMSR are the only restoration system with a score > 3 ; the other software produced lower scores, with fair quality assessments. The anchor at 7 kHz obtains a score greater than 0 (similar to those of a few commercial products). The quality range between the best and the worst restoration system is 2.60. CMSR produces scores greater than NoNOISE; CREAK and noNOISE achieve the same score.

The best restoration systems for *La signorina sfinciosa* are CEDAR, CREAK and CMSR. The anchor at 7 kHz produces scores equal to 0. The quality range between the best and worst restoration system is only 3.45.

The best restoration systems for the stimuli recorded on tape are PAR and CEDAR; all the other software produced similar scores, with low quality assessments. The anchors produce scores below 0. The quality range between the best and worst restoration system is only 2.2.

Discussion. It is possible to make some important comments:

- Observing the quality range between the best and worst restoration system, it seems reasonable to conclude that all the restoration algorithms work quite well (i.e., the user's evaluation is good enough) with high SNR signals ($\text{SNR} > 30 \text{ dB}$) as well as with very low SNR stimuli ($\text{SNR} < 10 \text{ dB}$): see the scores achieved with the *My Mariuccia take-a steamboat* and the tape stimuli.
- the behavior is in connection with the results of the objective comparison carried out (see the results showed in Sec 10.7.4.1). Summarizing: CMSR for the low and medium SNR, CREAK for medium SNR and PAR for high SNR.
- The best single tool is CEDAR, with a Grand Average equal to 3.72 (see Tab. 10.4). However, let us consider the three tools CREAK, CMSR and PAR as a single restoration environment (it could be called *CARE tool*: CAnazza REstoration or Csc Audio REstoration¹²): in this sense, the grand average of CARE is 3.83. This result explains the expedience to develop (and to use, of course) different tools, in relation to the supports considered.

12.8 Concluding Remarks

The opening up of archives and libraries to a large telecoms community represents a fundamental impulse for cultural and didactic development. Guaranteeing an easy and ample dissemination of some of the fundamental moments of the musical culture of our times is an act of democracy which cannot be renounced and which must be assured to future generations, even through the creation of new instruments for the acquisition, preservation and transmission of information. This is a crucial point, which is nowadays the core of the reflection of the international archival community. If, on the one hand, scholars and the general public started paying greater attention to the recordings of artistic events, on the other hand, the systematic preservation and access to these documents is complicated by their diversified nature and amount.

¹²CSC stands for Centro Sonologia Computazionale, the international laboratory of the University of Padua, leader in the Sound and Music Computing field since 1979.



Table 12.4: Mean for restored stimuli and anchors, 24 subjects. Stimuli: S1 = My Mariuccia take-a steamboat; S2 = La signorina sfinciosa; S3 = Unpublished tape recording

Restoration system	S1	S2	S3	Grand Average
CMSR	+3.10	+3.55	+2.80	+3.15
CREAK	+2.90	+3.95	+3.00	+3.28
PAR	+2.00	+1.20	+4.45	+2.55
CEDAR Tools	+3.20	+3.70	+4.25	+3.72
NoNOISE	+2.90	+1.78	+4.00	+2.89
iZotope RX	+2.40	+1.70	+2.40	+2.17
Waves	+2.30	+1.55	+3.45	+2.43
Audacity	+0.60	+1.25	+3.20	+1.68
Adobe Audition	+0.60	+0.55	+2.25	+1.13
Anchor 7 kHz	+0.50	+0.00	-2.50	-0.67
Anchor 3.5 kHz	-1.00	-4.00	-5.00	-3.33

Musicologist, Ethno-musicologists, scholars, audio archives personnel usually need to use a large – hardly manageable – number of sources stored in different media: outlines and annotations, scores, theater programmes and critical reviews, setting photos, audio signal and video footages. Although over the past few years the European Union has provided fundings for many projects focused on text codification and the creation of editions in electronic format, in most cases, modeling the traditional ecdotics methods, these studies apply technology without a real sharing of models and methodologies already in use by the information science.

The goal of this chapter is to stress that the archiving process of digitized audio documents is complete only when it includes all the ancillary information, in particular metadata of the original carriers. In this sense guidelines to the A/D transfer are detailed, in order to minimize the information loss and to automatically measure the unintentional alterations introduced by the A/D equipment. In addition, this chapter has presented:

1. A novel system able to synthesize the audio signal from a still image of a phonographic disc surface.
2. A software to extract metadata from photos and video shootings of audio carriers.

In relation to audio restoration, this chapter has presented:

- The CARE tool (Sec. 10.7). Sometimes, music audio documents are usually recorded in non-professional carriers by means of amateur recording system (e.g., ethnic music field). Thus, for their appropriate fruition and/or for a suitable use of MIR techniques is necessary to process the audio signals by means of audio restoration techniques.
- Four different case studies (Sec. 10.7.4), carried out using different carriers.
- An objective comparison, in order to validate the system, of the CARE tool with some *standard* filters at the varying of the SNR, considering 35 ethnic music documents (Sec. 10.7.4.1).
- A perceptual assessment, using the EBU MUSHRA test protocol (Sec. 10.7.5).



The extensive tests, carried out to investigate the effectiveness of the suggested algorithms when applied to a variety of audio data, show that the tool presented outperforms standard approaches to restoration. Listening tests (Sec. 10.7.5) confirm practical usefulness of the proposed solutions. The algorithm are implemented as a plug-in based software tool, which can be used as an added module to the most commonly used audio editors.

This chapter summarizes a number of experiences in several research/applied project on Digital Audio Archives and Audio Access, carried out by the author, including: “Electronic Storage and Preservation of Artistic and Documentary Audio Heritage (speech and music)” funded by the National Research Council of Italy (CNR); “Preservation and Online Access of Contemporary Music Italian Archive” funded by the Italian Ministry for Scientific Research; “Preservation and Online Fruition of the Audio Documents from the European Archives of Ethnic Music” funded by the EU under the Program Culture2000; “Search in Audio-Visual Content Using Peer-to-Peer Information Retrieval” funded by the EU under the Sixth Framework Programme; “Restoration of the Vicentini Archive in Verona and its Accessibility as an Audio e-Library”, joint project between the University of Verona and Arena Foundation. Equally important for defining the protocols described in this paper, has been the collaboration with important European audio archives, including: “Speech and Music Archives” of the National Research Council of Italy “Archive of the Studio di Fonologia Musicale”, owned by the Italian National Broadcaster Television; “Luigi Nono Archive”; “Bruno Maderna Archive”; “Historic Archive of Contemporary Arts” of the LaBiennale of Venice.

12.9 Commented bibliography

IASA-TC 03 [2005] presents guideline for the passive/active preservation of audio documents. A good tutorial on Audio Restoration methods is Godsill et al. [1998]. Ephraim and Malah [1985] details STSA algorithms, in particular the Ephraim-Malah suppression rule. In Canazza et al. [2010] the EKF applications in audio restoration field is presented. Canazza and Vidolin [2001] discusses the audio restoration in the field of electronic music.

References

- S. Canazza and A. Vidolin. Preserving electroacoustic music. *Journal of New Music Research*, 30(4):351–363, 2001.
- Sergio Canazza, Giovanni De Poli, and Gian Antonio Mian. Restoration of audio documents by means of extended kalman filter. *IEEE Trans on Audio Speech and Language Processing*, In press, 2010.
- Y. Ephraim and D. Malah. Speech enhancement using a minimum mean-square error log-spectral amplitude estimator. *IEEE Trans. Acoust., Speech, and Signal Process*, 33(2):443–445, 1985.
- S. Godsill, P. Rayner, and O. Cappé. *Digital audio restoration*. Kluwer, Boston, MA, 1998.
- IASA-TC 03. *The Safeguarding of the Audio Heritage: Ethics, Principles and Preservation Strategy*. IASA Technical Committee, 2005.

