A Sparse Auditory Envelope Representation with Iterative Reconstruction for Audio Coding

Joachim Thiemann

Department of Electrical & Computer Engineering
McGill University
Montreal, Canada

April 2011

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of Doctor of Philosophy.

2011/04/04

In memory of Johannes Mathias Thiemann
1933–2009

Abstract

Modern audio coding exploits the properties of the human auditory system to efficiently code speech and music signals. Perceptual domain coding is a branch of audio coding in which the signal is stored and transmitted as a set of parameters derived directly from the modeling of the human auditory system. Often, the perceptual representation is designed such that reconstruction can be achieved with limited resources but this usually means that some perceptually irrelevant information is included. In this thesis, we investigate perceptual domain coding by using a representation designed to contain only the audible information regardless of whether reconstruction can be performed efficiently. The perceptual representation we use is based on a multichannel Basilar membrane model, where each channel is decomposed into envelope and carrier components. We assume that the information in the carrier is also present in the envelopes and therefore discard the carrier components. The envelope components are sparsified using a transmultiplexing masking model and form our basic sparse auditory envelope representation (SAER).

An iterative reconstruction algorithm for the SAER is presented that estimates carrier components to match the encoded envelopes. The algorithm is split into two stages. In the first, two sets of envelopes are generated, one of which expands the sparse envelope samples while the other provides limits for the iterative reconstruction. In the second stage, the carrier components are estimated using a synthesis-by-analysis iterative method adapted from methods designed for reconstruction from magnitude-only transform coefficients. The overall system is evaluated using subjective and objective testing on speech and audio signals. We find that some types of audio signals are reproduced very well using this method whereas others exhibit audible distortion. We conclude that, except for in some specific cases where part of the carrier information is required, most of the audible information is present in the SAER and can be reconstructed using iterative methods.

Sommaire

Le codage audio moderne exploite les propriétés du système auditif humain de manière à coder efficacement la parole et la musique. Le codage en domaine perceptuel est une branche du codage audio dans lequel le signal est enregistré et transmis sous forme d'un ensemble de paramètres provenant directement d'un modèle du système auditif humain. La représentation perceptuelle est souvent conçue pour que la reconstruction puisse être réalisée avec des ressources limitées, mais cela requiert généralement l'inclusion de certaines informations perceptuellement non pertinentes. Dans cette thèse, nous étudions le codage perceptuel en utilisant une représentation destinée à ne contenir que l'information sonore, indépendamment du fait que la reconstruction puisse être effectuée de manière efficace. La représentation perceptuelle que nous utilisons est basée sur un modèle à canaux multiples de la membrane basilaire pour lequel chaque canal est décomposé en éléments de l'enveloppe et du signal porteur. Nous supposons que l'information contenue dans le signal porteur est également présente dans les enveloppes et supprimons donc les composantes du signal porteur. Les composantes de l'enveloppe sont réduites à l'aide d'un modèle de masquage transmultiplexeur pour former notre représentation parcimonieuse des enveloppes sonores (RPES).

Nous présentons un algorithme de reconstruction itératif pour la RPES qui fait une estimation des composantes du signal porteur à partir des enveloppes codées. L'algorithme a deux étapes. À la première étape, deux ensembles d'enveloppes sont produits: le premier dilate les échantillons des enveloppes clairsemées tandis que le deuxieme fournit des limites pour la reconstruction itérative. À la deuxième étape, les éléments du signal porteur sont estimés en utilisant une méthode d'analyse par synthèse itérative adaptée de méthodes conçues pour la reconstruction de coefficients de la transformée de grandeur. Le système est évalué à l'aide de tests subjectifs et objectifs sur des signaux de parole et audio. Nous constatons que certains types de signaux audio sont très bien reproduits par cette méthode alors que d'autres démontrent de la distorsion audible. Nous concluons que, sauf dans certains cas spécifiques où une partie de l'information du signal porteur est indispensable, la majorité de l'information sonore est présente dans la RPES et peut être reconstruite en utilisant des méthodes itératives.

Acknowledgments

There are many people who have been critical to the work presented in this thesis. I would like to thank first my advisor Prof. Peter Kabal for his commitment, time and support throughout my studies. Prof. Fabrice Labeau has also provided invaluable advice and support. Many thanks also go to my fellow students (and former fellow students) Abdul, Amr, Benoît, François, Hafsa, Mahmood, Mohamed, Qipeng, Tiago, and many others both in the TSP lab at McGill and at other Universities. I thank them for being there when I wanted to do *just one more test!*

Special thanks go out to my friends and family for being my support outside of the lab. Especially I would like to thank Madeline for being my support, my taskmaster, motivation, and editor without whom this thesis would not have been possible.

Contents

List of Figures

List of Tables

Acronyms

AF	Auditory Filter
BM	Basilar Membrane
CB	Critical Bandwidth
DFT	Discrete Fourier Transform
ERB	Effective Rectangular Bandwidth
FB	Filter Bank
FIR	Finite Impulse Response
FFT	Fast Fourier Transform
IHC	Inner Hair Cell
JND	Just Noticeable Difference
MDCT	Modified Discrete Cosine Transform
MUSHRA	Multiple Stimuli with Hidden Reference and Anchor
SAE	Sparse Auditory Envelope
SAES	SAE Sample
SAER	SAE Representation
STFT	Short-Time Fourier Transform

Chapter 1

Introduction

1.1 Perceptual audio coding

Audio coding has been an area of active research for several decades both in academia and industry. It has reached a remarkable level of maturity and its application has become ubiquitous in everyday consumer items. Today, if a device has a way to store a few megabytes of data and is capable of producing sound, chances are one can use it to play back music encoded in the popular "mp3" format. Yet, there continues to be interest in improving the sound quality of the stored audio and in reducing the amount of storage required.

The efficiency of an audio coding scheme is a function of the number of bits per second (bitrate) required for storage and the distortion the audio signal exhibits when reproduced. Usually, a higher bitrate results in lower distortion and to express the theoretical efficiency as a numerical quantity, we use the rate-distortion (RD) function. We aim to make a tradeoff between the bitrate and the distortion, either by fixing the bitrate and minimizing distortion or minimizing the bitrate to achieve some fixed level of distortion. While measuring the bitrate is a simple matter of counting the number of bits needed by the audio codec over some given period of time, evaluating the distortion is a more complex task. We often measure the distortion using a simple signal to noise ratio, but the ultimate judge of quality for an audio signal is a human being and it is hard to predict how much distortion the listener can hear and the degree to which different types of distortion are audible. A better estimate of distortion can

be achieved by using a computational model of a typical human listener. Audio coding that exploits this is termed perceptual coding and is the main subject area of this thesis.

1.1.1 Block-transform perceptual coding

The use of auditory models to aid in the design of audio codecs began in the 1980's with Johnston [Johnston, 1988], and his pioneering work has led to many of the transform based codecs that are common today [Painter and Spanias, 2000]. In a typical transform codec as depicted in Fig. 1.1, the audio signal to be encoded is first split into short-time segments, which are then transformed into a frequency domain representation using a fast block transform. A model of the human auditory system then evaluates this frequency domain representation of the sound to determine regions of the frequency spectrum where added noise is inaudible. This information is used by the quantizer block to encode the frequency domain representation in such a manner as to reduce the quantization noise in the audible portions of the spectrum.

There are several reasons why this approach has been very successful. Block transforms can be performed very efficiently using fast transforms (such as the Modified Discrete Cosine Transform, MDCT, or the Fast Fourier Transform, FFT) and the properties of the resulting transform coefficients are well understood in the context of data compression. Furthermore, the frequency domain properties of the auditory system have been investigated thoroughly and can be exploited easily by processing the transform coefficients.

A disadvantage of this approach is that block based processing of sound signals makes temporal properties of the auditory system difficult to exploit, since the human hearing system is not based on fixed-length time segments. Thus, the processing of some audio events (such as sharp transitions) depends on when they occur. Block-based coders can be made more flexible in this respect using techniques such as dynamically changing the length of frames to better fit the current signal properties [Spanias et al., 2007], but there is no unified, consistent approach to dealing with both the time and frequency properties. Block-transform codecs can be regarded as using a perceptual model guiding the quantization of the transform coefficients, hiding the artifacts where they cannot be perceived. A more direct use of perceptual models

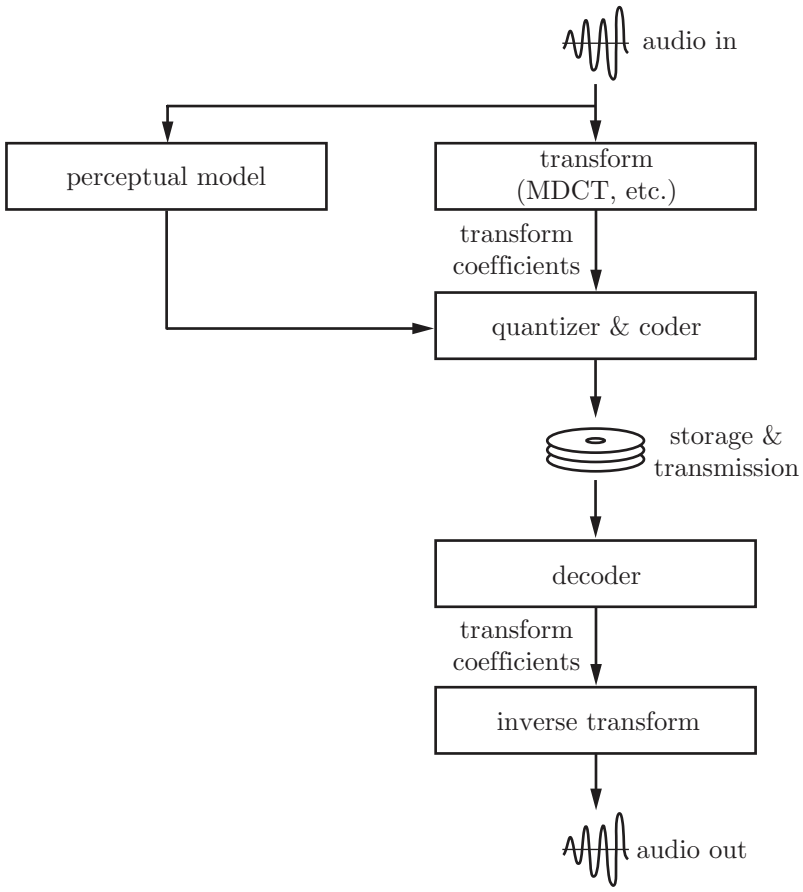


Fig. 1.1: Diagram of block-transform perceptual coding

that has appeared recently is perceptual domain coding.

1.1.2 Perceptual domain coding

A recent new approach to perceptual audio coding is perceptual domain coding (PDC), which uses the parameters of an auditory model directly to encode the audio signal. As shown in Fig. 1.2, the structure of such a scheme is actually quite simple. It is based on the idea that the auditory model can be used as the primary transform of the sound signal into a representation which can be quantized directly. Quantization can thus take into account perceptual properties, especially if the representation has a single-letter squared-error criterion that reflects the audibility of distortions [Kubin

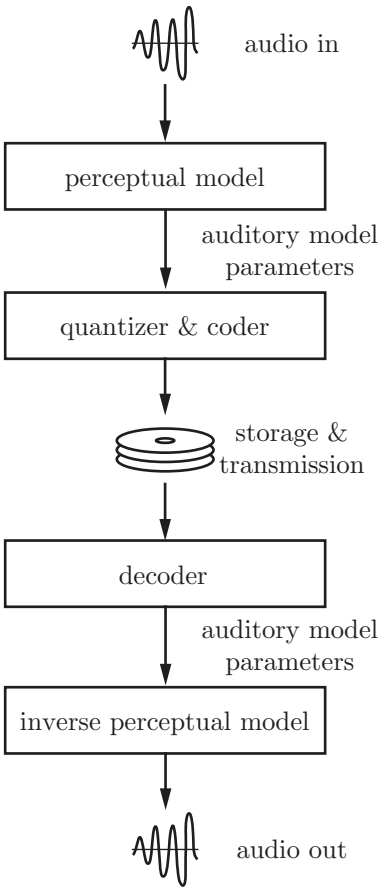


Fig. 1.2: Diagram of perceptual domain coding

and Kleijn, 1999]. However, in these codecs the model is constrained to be easily invertible to reduce the computational load on the decoder.

To date, no perceptual model has been designed specifically for audio coding. Generally, the models have been designed either purely for research purposes (to further the understanding of the human auditory system) or for the objective evaluation of perceived quality (eg. PEAQ [ITU-R, 2001]). As a result, the problem of parameter granularity is addressed only in terms of a noticeable difference detection or as a globally aggregated number. In their initial analysis stage, models also often increase the amount of data manyfold, with the aim towards accuracy and flexibility at the expense of efficiency. Our ultimate goal for a coding-oriented model is to strike a balance between accuracy, data efficiency and computational complexity.

This balance is in part determined by the depth to which the auditory system is modeled. The human auditory system is a complex biological system that is closely knit into other cognitive functions, such as the memory, motor, and of course linguistic structures [Moore, 2003]. Models can be designed to simulate only the periphery of the inner ear, such as the movement of the Basilar Membrane (BM), they can model the neuronal signals in the auditory nerve, or even take higher order effects into account. Modelling each step from the periphery to the cortex should allow for higher coding efficiency as more inaudible information is discarded. The problem is that higher order functions cannot be directly observed, which makes accurate modeling of these functions very challenging. In this thesis we tackle the human auditory system by modeling the envelopes of early auditory neuron excitations, which we term auditory envelopes.

1.1.3 Auditory envelopes representation

Based on the frequency analysis performed by the BM and the temporal smoothing of neural transduction, auditory envelopes (AE) are a representation that has been used in several research auditory models. AE properties have been studied to understand the audibility of modulation patterns. This representation has the property of being relatively high-level, discarding a significant amount of information. However, AE have not been used in perceptual coding since reconstruction from envelopes is computationally difficult. If we lift the constraint on computational complexity for the decoder, this reconstruction can be done using iterative methods: synthesis by analysis with refinement.

1.1.4 Reconstruction using synthesis by analysis

The computational difficulty of inverting auditory models arises from the fact that as the level of modeling increases, information is discarded. An auditory model is therefore a many-to-one mapping of a signal to a representation and the inversion of audio from the model parameters is not unique. In effect, some of the discarded information must be recreated to create an audio signal that is perceptually equivalent to the original signal. Given the nonlinearities in the auditory system, direct model inversion may be numerically unstable.

Perceptual domain audio codecs are designed to encode and decode signals intended for human listeners. Since the coded representation is based on an analysis stage that emulates the human auditory system, we can look at the decoder in a different way: as a transmultiplexing system, *en*coding a set of percepts into a single-dimensional channel (the audio signal), to be decoded by the human ear. Introduced in [Feldbauer and Kubin, 2004] to design a masking model for perceptual domain coding, the transmultiplexer model uses the analysis stage from the encoder to model the human listener, effectively turning the codec "inside out".

While originally used to design and optimize a perceptual domain encoder, in this thesis we propose to take this idea a step further and make the transmultiplexer model an integral part of the decoder, rather than just an aid in the design of the encoder. Adding a perceptual model to the decoder allows us to immediately check the quality with which the coded information will be perceived. Then, if the model indicates that the quality is not satisfactory, the decoder can adjust the reconstructed audio signal to better fit the encoded data. This leads to the concept of iterative synthesis-by-analysis. Iterative methods are not usually used at the decoder in audio coding systems due to the high computational cost, but the continued increases of available computational power even in handheld portable devices make this subject a worthwhile topic for study.

1.2 Perspective and goal of the thesis

The origin of the ideas in this thesis can be traced back to earlier work in perceptual modeling and audio coding. Perceptual models are often used indirectly in the development cycle of audio codecs to evaluate the performance and coding efficiency as part of objective quality measures; doing so is far cheaper and quicker than performing repeated subjective testing with many subjects whose responses can be difficult to interpret. There have even been proposals to automate the tweaking of codec parameters by using perceptual models [Holters and Zölzer, 2009].

The "target audience" to a codec developer then almost seems to be the model itself rather than a real listener, though of course developers of auditory models and subjective audio measures strive to make the model and human listener consistent

with each other. The question then arises, if we are developing to satisfy the demands of some computational construct, why not include this construct directly into the decoder to check the quality of the reconstruction? Furthermore, if we do so and if the perceptual model is evaluating the signal based on parameters X, Y and Z, should it not be sufficient to transmit just those parameters? After all, the original signal is not available at the decoder so if the extra information is needed for the perceptual model, how much of the original information is required?

This thesis is the result of our work tackling these questions. It draws on combining concepts from the science of psychoacoustics to evaluate perceptual audio representations with signal processing and linear programming theory to evaluate the iterative system that implements the decoder. The overall system is very complex so we focus on iterative audio reconstruction from auditory envelopes. Given the maturity of the audio coding field, we cannot hope to design and implement a complete audio codec that competes with those that are commercially available and used in consumer devices. Rather, we use components that can be understood individually and in combination with each other to explore the feasibility of iterative reconstruction for perceptual coding from auditory envelopes.

1.3 Thesis contributions

There are several contributions in this thesis:

- We introduce a new perceptual audio representation based on the sparse sampling of auditory envelopes. In contrast to representations that are designed for computationally simple inversion, this representation is designed to only contain the perceptually relevant information. The algorithm to generate the representation consists of extracting the auditory envelopes, subsampling and sparsification using a masking model. We show that some types of signals can be reconstructed very well from this representation. However, we also show that some types of audio signals cannot be reproduced accurately, and we examine the cause for this.

- A two-stage algorithm is presented to reconstruct an audio signal from the above perceptual representation. The sparse sampling of auditory envelopes

is expanded into two sets of full-rate envelopes, one set for known envelope values and the other set for limits within which the reconstructed envelopes must fit. This stems from the idea that in the perceptual representation, a missing sample is not without information, but was removed due to the application of the masking model. The second step is the iterative estimation of information discarded from the perceptual representation at the encoder, using a synthesis-by-analysis loop applying the envelope constraints from the previous stage.

- We present a method of analyzing the iterative reconstruction from subchannel envelopes of FIR filterbanks using a circulant matrix notation and frame theory. While this notation is limited to fully oversampled FIR filterbanks, it is simple to understand and allows for straightforward numerical analysis.

1.4 Outline of the thesis

This thesis is organized as follows. Chapter 2 gives an overview of the human auditory system focusing in particular on the inner ear and early neural transduction to show the basis for auditory envelopes as a representation of auditory percepts. We discuss the basics of modeling these structures computationally, then describe some applications of perceptual modeling in audio coding. In particular, perceptually motivated coders are introduced upon which we base a new perceptual representation to illustrate the key concepts of our research.

In Chapter 3, some background mathematical notation and techniques are presented that tie into the basic scheme of iterative reconstruction from perceptual subband envelopes. We describe subband filtering using circulant matrix notation and frame theory. The reconstruction from subband signal estimates and subband signal envelopes is presented and analyzed using the framework used by Griffin and Lim [Griffin and Lim, 1984], which was used originally for reconstruction of signals from magnitude-only transform coefficients.

The central topic of the sparse auditory envelope representation and its iterative reconstruction method is presented in Chapter 4. We describe the general framework and issues that must be addressed, beginning with an abstract discussion of iterative reconstruction from perceptually coded representations. The sparse auditory envelope

representation is adapted from a perceptually motivated audio coder by Feldbauer [Feldbauer, 2005] and we show that the new representation discards the fine temporal structure of auditory subband signals, which is typically both difficult to encode at low bitrates and of limited perceptual importance. The various stages of the encoder and decoder are described from a perceptual modeling perspective. The encoder begins with a highly redundant envelope representation, which is first subsampled and then sparsified. The sparsification is achieved using a transmultiplexer based masking model to remove from the representation envelope samples that are assumed to be inaudible due to temporal and simultaneous masking. We then describe the algorithm to reconstruct an audio signal from the sparse envelope representation. This also is a multi-step process: in the first step the full auditory envelopes are estimated from the sparse representation and in the second step the carrier signals associated with the envelopes are rebuilt using the iterative framework from Chapter 3.

The details of the implementation of the reconstruction algorithm are described in Chapter 5, along with the results of testing this implementation on speech and audio signals. The quality of the reconstructed signals is evaluated by subjective analysis and we examine the properties of the algorithm using objective measures as introduced in Chapter 3. Using these results, we discuss the premise of using the sparse auditory envelope representation and iterative reconstruction for perceptual coding in the conclusion of the thesis in Chapter 6.

Chapter 2

Auditory perception and modeling

The topic of this thesis is based on the processing of sound by the human auditory system. To show how the algorithms presented in later chapters are rooted in the human auditory system, this chapter will provide a brief overview of the physiological structures of the ear, as well as a signal processing view of modeling these structures. Also, some applications of auditory models are presented, focusing on coding using pulse-based and modulation domain techniques.

2.1 Overview of the auditory system

Like all sensory organs, the human ear is a very complex instrument. Of particular interest to the research presented here is the processing of information in the inner ear. It is useful to briefly discuss the outer and middle ear as well as they act as direction-dependent filters for the incoming sound. Detailed descriptions of the entire auditory system can be found in [Moore, 2003; Allen, 1985; Zwicker and Fastl, 1999], but important concepts are summarized below.

2.1.1 The outer and middle ear

The outer ear, consisting of the earlobe (pinna) and the ear canal, is the only visible part of the ear. It focuses the incoming sound into the ear canal and alters the sound depending on direction. The ear canal ends at the ear drum (the tympanic membrane), where the pressure waves are converted into mechanical movement. In the middle ear,

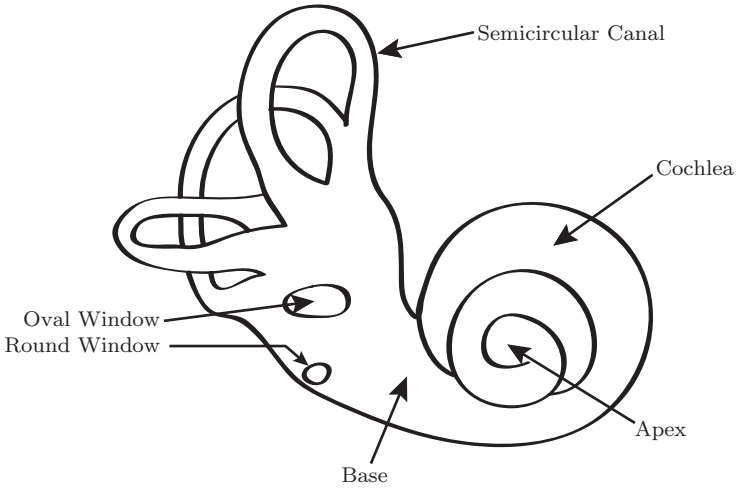
three small bones (ossicles) connect the ear drum to the oval window of the cochlea in the inner ear. These bones act as a set of levers to match the impedance of air to the fluid in the cochlea, ensuring efficient transmission of the sound signal.

The effect of the outer and middle ear on incoming sound is mostly regarded as a linear passive filter, emphasizing frequencies between 500 and 4000 Hz. There are nonlinear effects which appear when certain sounds are present, such as noise exceeding the threshold of pain. However, these are outside the scope of signals considered here since there is little interest in reproducing these effects in audio coding.

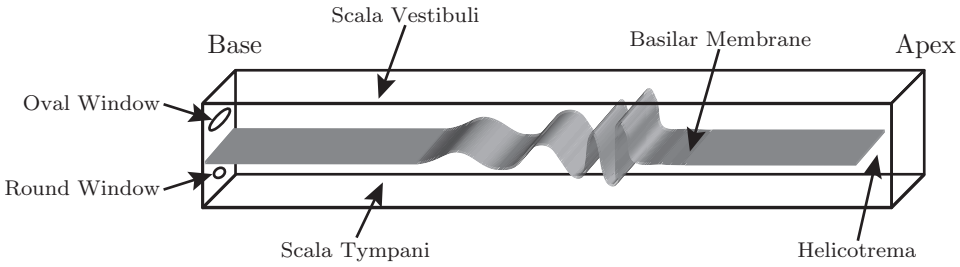
2.1.2 Inner ear anatomy

The inner ear, a structure embedded in the hard temporal bone, is shown in Fig. 2.1a. This structure contains both the balance organ (in the semicircular canals) and the auditory organ in the cochlea. The cochlea is a snail-shaped structure of three fluid-filled channels, called the scala vestibuli, the scala media, and the scala tympani. These channels are separated by two membranes, the basilar membrane (BM) and Reissner's membrane. Reissner's membrane separates the scala vestibuli and the scala media, but since this membrane is very light and thin, these two channels are usually treated as a single unit [Zwicker and Fastl, 1999]. The three channels curl up in parallel up to the apex of the cochlea, where the scala vestibuli and the scala tympani are joined at the helicotrema. At the base of the cochlea, the channels are connected to the middle ear, by the oval window and the round window. The oval window is the interface between the ossicles (specifically, the stapes or stirrup) and the scala vestibuli. Here, the movement of the ossicles is converted into pressure variations of the fluid in the cochlea. The round window is connected to the scala tympani to equalize the pressure.

The scala vestibuli and the scala tympani are separated by the BM. Since the fluid medium is essentially incompressible, pressure changes must be equalized through the BM to the scala tympani and the round window. The BM varies in stiffness and thickness along its length and resonates at high frequencies at the base of the cochlea and low frequencies at the apex. Thus, periodic pressure variations will create a travelling wave pattern on the BM as shown by the schematic view of the BM in Fig. 2.1b. Given a simple sinusoidal input, the maximal displacement of the BM



(a) Inner Ear Structure



(b) Cochlea as unwound canal

Fig. 2.1: Schematic depiction of the ear and the BM in response to a stimulus, with the cochlea unwound, being shown as a straight and uniform canal [Zweig et al., 1976].

will be at a point dependent on the frequency of that sinusoid and, for more complex signals, there will be multiple local maxima. The BM can be thought of as performing a frequency analysis similar to a Fourier Transform.

The BM supports the Organ of Corti, consisting of a set of sensory cells, the inner and outer hair cells (IHC and OHC). These cells convert the mechanical movement of the BM into neural impulses. The IHC primarily attach to afferent neurons (transmitting information to the brain) and the OHC to efferent neurons (transmitting information from the brain). There are about 3 500 IHC and 12 000 OHC connected to about 30 000 neurons; however, only about 1 800 of those are efferent. It is thought that the OHC play an active role in changing the characteristics of the BM and are controlled by the auditory complex of the brain [Moore, 2003].

The auditory nerve connects the cells within the cochlea to the central nervous system. There, it is attached to the auditory cortex in several different places, including the ventral cochlear nucleus and the dorsal cochlear nucleus. These forward the auditory information to the superior olive (SO) and higher level structures in the brain. However, the SO is notable for being one of the first structures to combine stimuli from both ears and thus plays an important role in the spatial perception of sound [Moore, 1991]. In the brain in general, higher level tasks are often quite difficult to localize and all but impossible to examine in isolation.

2.2 Modelling auditory properties

To develop models of the human auditory system, researchers have studied it using both anatomical examinations and functional observations. Since it is a part of a living system and both highly complex and delicate, anatomical examinations are only done on either cadavers or animal models. On the other hand, functional observations are done by presenting certain controlled stimuli to volunteer test subjects who are then asked to perform some task based upon their perception. This may be as simple as indicating whether or not they can detect a target stimulus in the presence of a background stimulus, such as a tone in the presence of broadband noise. From the results of these tests, many processes in the ear can be inferred. However, functional observations of this kind are prone to variations and bias of the individual subjects.

Together, anatomical and functional observations are complementary for understanding auditory perception and can be used to construct auditory models. Viewing sound as a signal carrying information [Zwicker and Feldkeller, 1967], an auditory model in the context of this thesis is a signal-processing algorithm that takes as input a representation of sound (usually a digital sound file) and turns it into a set of parameters that represent stimuli internal to the auditory system. Possible model parameters are the deflection of the BM, the firing rate of a group of auditory neurons, and a more abstract detection of "objects". For the signal analysis in this thesis we model the BM and neural transduction only since auditory object detection is an area where fundamental research is still being conducted.

2.2.1 Modelling the BM movement using auditory filters

A key concept in the understanding of the auditory system from a functional perspective is the Critical Bandwidth (CB). Originally postulated by Fletcher [Allen, 1996], the CB describes the extent to which sounds at different frequencies interact and thus can be used to predict masking behaviour in the auditory system.

A common way to describe the frequency-specific behaviour of the properties of the auditory system is to model it as a bank of parallel auditory filters (AF). A great deal of research has been done to quantify the interactions of stimuli at different frequencies more precisely, commonly using threshold-of-hearing experiments with human subjects [Moore, 2003], but also by direct observation of the BM in sedated animals [Békésy, 1953]. For a comprehensive review see [Robles and Ruggero, 2001]. In frequency domain, the AF shape can be described as a rounded exponential ("roex(p)", where p is a parameter to fit observations), with the bandwidth described in terms of the Effective Rectangular Bandwidth (ERB). The AFs are highly overlapping in frequency since each point of the BM defines its own AF, with a critical frequency determined by its distance from the round window.

Auditory filters can be specified in the frequency domain, where the input signal is first transformed by use of a short-time block transform such as the Fast Fourier Transform (FFT) or the Modified Discrete Cosine Transform (MDCT). The individual auditory filter responses are then computed by a weighted integration using the filter shapes in frequency domain, as given by functional observations. Because of the

transform, this frequency domain view of auditory filters processes the audio signal in blocks, transforming a fixed length segment of audio at a time. However, in the auditory system, stimuli are processed over timescales that are not easily described by simple intervals. While it is possible to work around this limitation, it makes the implementations more complicated.

Alternatively, auditory filters can be modeled in time domain using an approximation of the impulse response of the BM. This is closer to being a model of the physiological processing in the ear and is described in more detail below, as it is the basis of the model used in the following chapters.

Gammatone filters

A common model for the AF is the gammatone filter, with a causal impulse response in the form of

$$g_m(t) = \begin{cases} a_m t^{(n-1)} e^{-2\pi b_m t} \cos(2\pi f_m t + \phi_m), & (t > 0), \\ 0, & (t \leq 0), \end{cases} \tag{2.1}$$

where f_m is the critical frequency and n, b_m, and ϕ_m are constants fitted to empirical data. The parameter b_m defines the bandwidth of the filter and is directly related to the CB, while n determines the slope of the temporal envelope. The subscript $m = 1, \ldots, M$ is used to indicate that Eq. (2.1) defines a set of discrete filters that sample the BM at a set of (spatial) points. The spacing of these samples from the base to the apex depends on several factors that will be discussed in Chapter 5.

These formulae were used by Patterson for creating "auditory images" for both simple stimuli and complex sounds [Patterson et al., 1992] and have since found widespread use as an auditory pre-processing step for many perceptually motivated audio processing applications. Flanagan [Flanagan, 1962] first used gammatone functions to match the measurements by Békésy, but the "modern" formulation was developed to match data obtained using the reverse-correlation ('revcor') technique on recordings of nerve-fiber responses in cats [de Boer and de Jongh, 1978], with the constants fitted to subjective measurements in humans. Values commonly used are

$n = 4$ and $b_m = 1.019\,\mathrm{ERB}(f_m)$ [Moore, 2003] where

$$\mathrm{ERB}(f_m) = 24.7(4.37 f_m/1000 + 1). \tag{2.2}$$

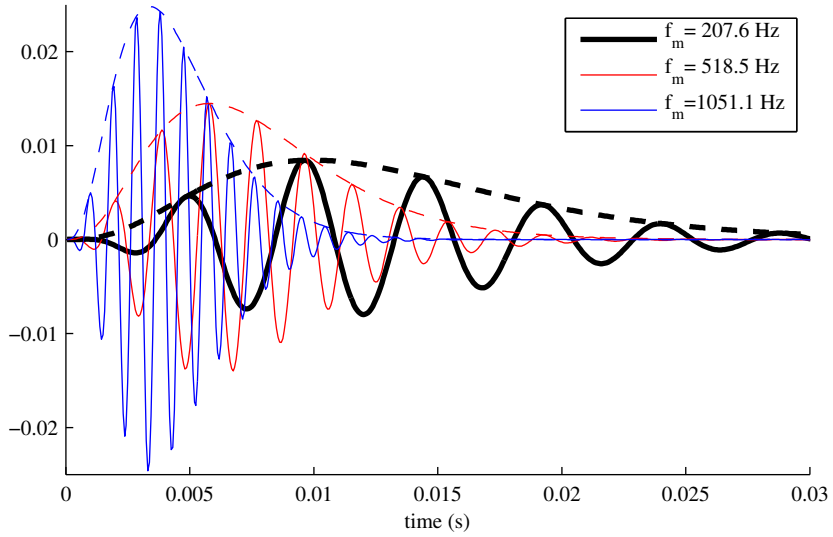
Typical responses of gammatone filters are shown in Fig. 2.2a. The dashed lines show the envelopes $g_{e,m}(t) = a_m t^{(n-1)} \exp(-2\pi b_m t)$, with the interior solid lines representing the actual response with $\phi_m = 0$. The value of a_m is set such that the gain at the critical frequency is 1, as shown in the frequency domain versions in Fig. 2.2b. For clarity a set of filters very far apart in frequency (spaced at 5 ERB) is shown. This does not show the large overlap of adjacent filters in frequency domain. At a spacing of 1 filter per ERB, the crossover attenuation of the filters is at -3dB.

2.2.2 Auditory envelopes for modeling neural transduction

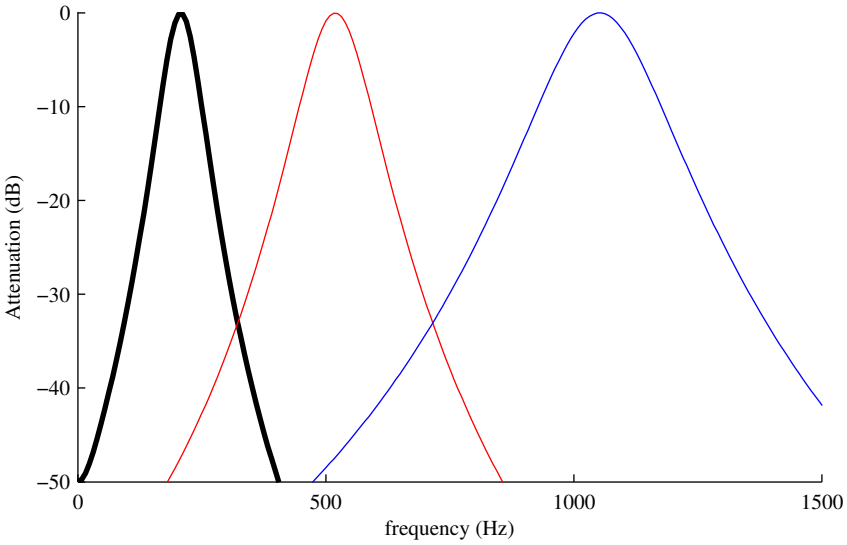
As described above, conversion of the BM deflection into neural impulses is accomplished by the hair cells. In auditory models based on gammatone filters, hair cells are not modeled individually, but rather as groups. Since a gammatone filter channel models the movement of a point on the BM, the neural transduction model is a description of the combined responses of the hair cells near this point.

The hair cell response to the BM movement is generally modeled by a nonlinearity followed by a lowpass filter [Patterson and Holdsworth, 1996]. The implementation of the nonlinearity in particular can vary in complexity between models. Based on simulating intracellular processes, a temporal derivative followed by a sigmoidal function is used in [Yang et al., 1992; Chi et al., 2005]. Later models use a half-wave rectifier [Dau et al., 1996a;b; Jepsen et al., 2008] which produces effectively the same signal but at lower complexity. In this thesis, we use a Hilbert envelope decomposition to extend the half-wave rectifier to complex signals as in [Ghitza, 2001]. The Hilbert envelope of a signal has fewer high-frequency components than the rectification of a real signal and thus requires less filtering than the other methods. We call the envelopes of the gammatone filter outputs the auditory envelopes (AE).

A common property of all of the methods to compute the hair cell model output is that the resulting signal is strictly positive with attenuated high-frequency components. The combination of rectification and lowpass filtering models the stochastic



(a) Time-domain impulse response of three gammatone filters.



(b) Frequency-domain response of those same filters.

Fig. 2.2: Time and frequency domain responses of select gammatone filters

nature of the individual hair cell responses and the summation of the large number of individual cells. From a physiological viewpoint, the signal computed by the neural transduction model (one for each gammatone filter channel) represents the cumulative rate of firing of a group of hair cells in the narrow section of the BM tuned to that particular critical frequency. We may also regard it as being proportional to the probability of an individual hair cell at the centre of the spot represented by the gammatone filter firing a neural impulse. The complete set of all auditory channel responses is termed the *auditory pattern* (AP).

In Fig. 2.3, we show representations of a short sample of speech, the word "turned" spoken by a female speaker. The top figure shows the waveform sampled at 16 kHz, followed by the spectrogram with FFT length of 128 samples, or 8ms. Each block is windowed using a Hamming window. Next, we show the auditory pattern as obtained from the Dau model [Dau et al., 1996a;b], plotted in a similar manner to the spectrogram. The data was obtained using the Dau model implementation from the HUTear2 toolbox [Härmä and Palomäki, 1999]. Comparing the spectrogram to the auditory pattern, the logarithmic stretching of the frequency axis is evident by the spacing of the harmonics around the 1.2 second mark. From the auditory pattern, one auditory channel (with a critical frequency of 532.8 Hz) is shown in the bottom plot. Both the log of the Hilbert envelope of the gammatone filter output (line with high-frequency spikes) as well as the final auditory channel response (smooth line) are shown. This plot shows that, compared to the Hilbert envelope of the auditory channel signal, the Dau model channel response is a strongly low-pass envelope.

The question of the nonlinear processing and lowpass filtering within each auditory channel is especially interesting with respect to coding, since these affect the frequency response and thus are indicative of the amount of information carried within each channel. A notable study of the auditory channel frequency characteristics from a perceptual viewpoint can be found in [Ghitza, 2001], based on the Modulation Transfer Functions in [Chi et al., 1999]. Ghitza found that to preserve speech quality, the minimum bandwidth of the envelope information must be roughly half the CB. More recent studies have proposed a better model of the varying timescales across the audible frequency range by using a multiresolution representation [Chi et al., 2005]. These concepts are combined in the model we describe in Chapter 4, which

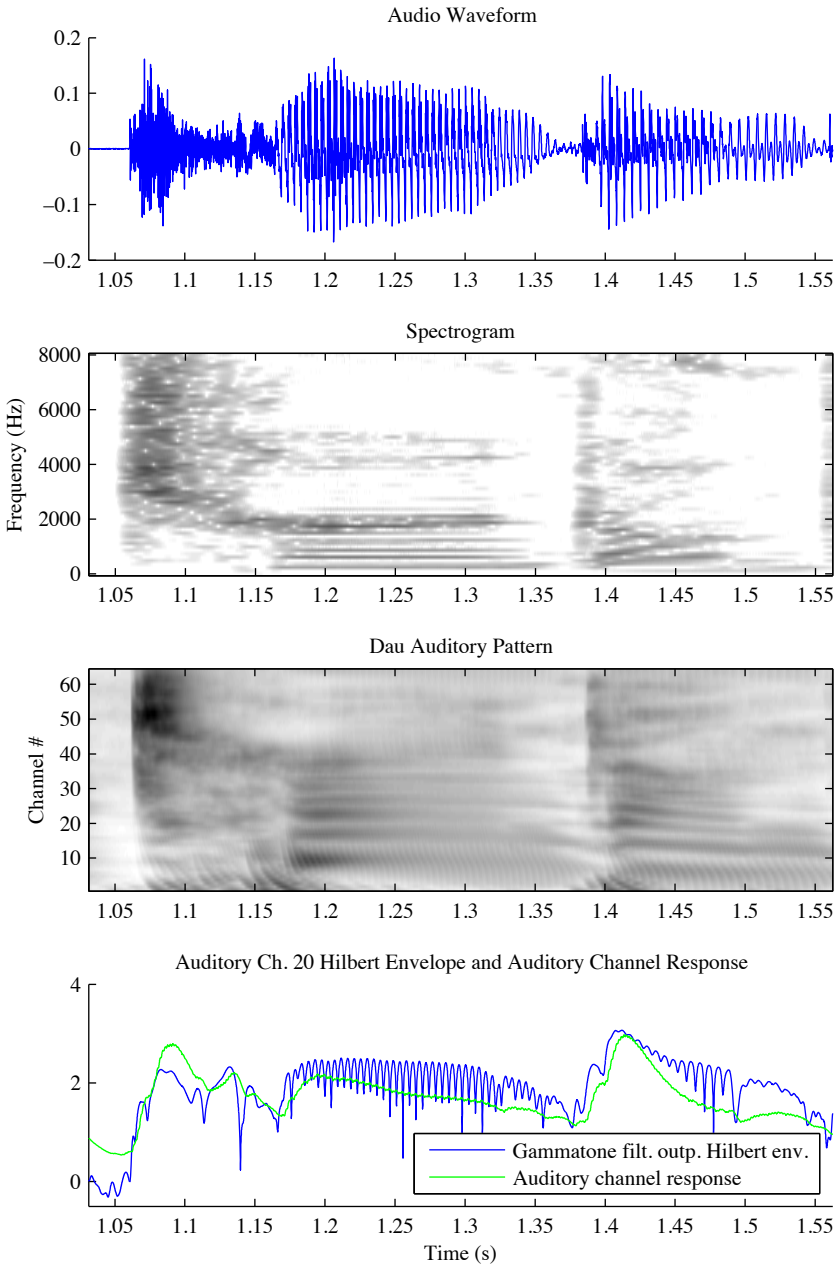


Fig. 2.3: Visualisations of audio waveform and auditory model representation

uses a sparse sampling of low-pass filtered Hilbert envelopes from the outputs of a gammatone filterbank.

2.2.3 Higher order modeling

Higher-level modeling is possible by further processing the neural transduction response. However, processing can vary greatly between different models and make comparisons difficult. Lateral inhibition may be applied to the auditory channel responses [Yang et al., 1992] to model interactions between neurons that are close in terms of critical frequency. Other models apply modulation analysis to segregate modulation patterns by frequency content of the channel responses [Jepsen et al., 2008].

The processes of how the brain interprets the information from the BM or the auditory neurons are simulated by higher order cognitive models that tend to be even more abstract. These processes include the perception of pitch [de Cheveigné, 2005], binaural information [Thompson and Dau, 2008], and the processing of "auditory objects". The field of Auditory Scene Analysis (ASA) describes how audible sensations are grouped in the brain into these auditory objects, much like the visual system sees shapes. Models are still somewhat rough, and while there are applications of ASA for speech recognition, pitch tracking, and music segregation [Bregman, 2007], the level of detail with which auditory objects can be described is not yet suitable for audio coding.

2.3 Applications of perceptual analysis and synthesis

Auditory models started as methods to either visualize audio signals or to predict audibility of stimuli in some condition. Auditory model inversion, the process of synthesizing an audio signal from parameters obtained using an auditory model, has been used to test model accuracy and to perform speech enhancement or separation [Yang et al., 1992; Slaney, 1995; Kollmeier, 2005]. In these applications there is no data to transmit over a finite capacity channel, thus quantization and coding of parameters is not needed. Still, reconstruction or model inversion is typically a difficult and computationally complex problem, since multiple constraints in time

and frequency domain must be met. Furthermore, two signals that sound very similar (and so should have nearly identical perceptual representations) can be very different in time-domain. However, auditory modeling and model inversion has recently been the basis for several proposed codecs. In the following section, some methods are summarized that use a gammatone filterbank (FB) or equivalent decomposition as the initial stage.

2.3.1 Pulse based methods and matching pursuits

Two related methods of encoding the signal as a set of pulses are by subchannel peak-picking and matching pursuits of gammatone impulse atoms. The reconstruction from these methods is quite similar mathematically, being the summation of gammatone filter impulse responses that are scaled and translated. The analysis methods are very different, as will be explained in this section.

Subchannel signal peak-picking methods

An early method for coding of narrowband speech using auditory pulses is presented in [Kubin and Kleijn, 1999], shown in Fig. 2.4. This method uses an analysis FB of gammatone filters to model the BM followed by a simple model of neural transduction. The transduction model is a half-wave rectification of the filter responses followed by a power-law compression to model the IHCs. A peak picking process models the combination of neuronal groups, lowering the frequency of pulses. The resulting auditory representation is a set of impulses of varying amplitude in each channel of the FB, similar to Patterson's pulse-ribbon model [Patterson et al., 1992]. Figure 2.4 shows the multichannel nature of the pulse coding methods and how each stage of processing roughly corresponds to the processing in the human auditory system. The decoder compensates for the processing of the pulse amplitudes then generates the reconstructed audio signal using a synthesis FB.

The reconstruction of the speech signal (after decoding the auditory representation) is performed by first reversing the power-law expansion on the impulses and compensating for the energy loss due to the peak picking. The resulting pulse trains are passed through a synthesis FB, where for each channel the filter impulse response is the time-reverse of the analysis filter. The resulting quality is robust to quantization

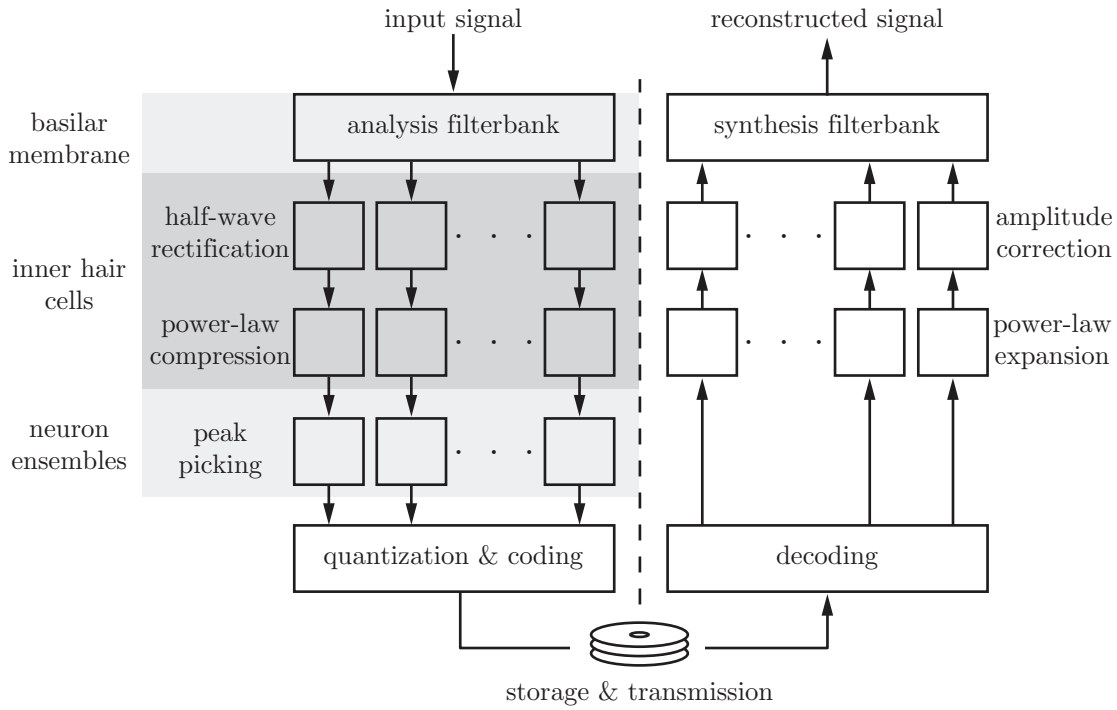


Fig. 2.4: Invertible auditory pulse coder [Kubin and Kleijn, 1999], redrawn from [Feldbauer, 2005].

of the pulse amplitude when at least one bit per impulse is used. However, a large number of pulses is needed, more than the number of samples in the original signal.

In [Ambikairajah et al., 2001] a method for wideband speech and audio coding is presented, based on the above ideas. The problem of the number of impulses is addressed, and an algorithm is introduced where impulses are removed from the channels by incorporating a masking model similar to that of MPEG [Brandenburg and Stoll, 1994] across channels and accounting for temporal post-masking using an exponential decay function. Within a channel, a pulse is discarded if its amplitude is below the decay function of a previous pulse.

Also addressed in [Ambikairajah et al., 2001] and a followup [Lin et al., 2002] is the problem of quantizing and coding the temporal pulse position. The pulse positions of the higher frequency channels (above 1.5kHz) are coded less precisely using a vector quantizer, while lossless coding is used in the lower frequency bands.

To reduce the number of impulses required by the auditory representation for coding, Feldbauer introduces the concept of the transmultiplexer as part of a masking model [Feldbauer and Kubin, 2004; Feldbauer et al., 2005]. As shown in Fig. 2.5a, the encoder described in [Feldbauer et al., 2005] modifies the system shown in Fig. 2.4 by moving one of the pulse amplitude correction steps from the decoder to the encoder, and more importantly, adding a masking model before the pulse amplitude compression and coding stages. This masking model will be briefly described here, and in more detail in Chapter 4.

Like the coder shown in Fig. 2.4, the input signal is analyzed by a gammatone FB followed by a half-wave rectifier and a peak-picking function. The adjustment of pulse amplitudes to compensate for the loss in energy due to the peak-picking process is now performed by the encoder, such that the masking model that follows uses the correct energy per channel to estimate masking thresholds. The masking model finds pulses that do not contribute significantly to the reconstructed audio and removes those from the auditory representation. To ensure the overall energy in the channel is maintained, the pulse amplitudes are corrected again before being compressed, quantized, and encoded into the bitstream.

The decoder shown in Fig. 2.5b is actually a simpler version of the one shown in Fig. 2.4 since the amplitude correction step has been moved to the encoding stage. The decoder rebuilds the pulse representation by decoding the stream and the pulses are converted into an audio signal using the synthesis FB of reverse-time gammatone impulses as with the earlier coder. Feldbauer adds an equalizing postfilter to reduce the frequency domain ripple of the overall system.

The masking model is based on observing a single isolated impulse as it is passed through the synthesis FB and then analyzed again. At the decoder, the audio signal due to that pulse will simply be the impulse response of the corresponding channel filter of the synthesis FB, a time-reversed gammatone pulse. Analyzing this pulse with the auditory FB yields an excitation pattern that spreads in time and in the FB frequency decomposition from the initial pulse position. This excitation pattern is compared to adjacent pulses in the original auditory pulse pattern to identify pulses that are masked by their stronger neighbours.

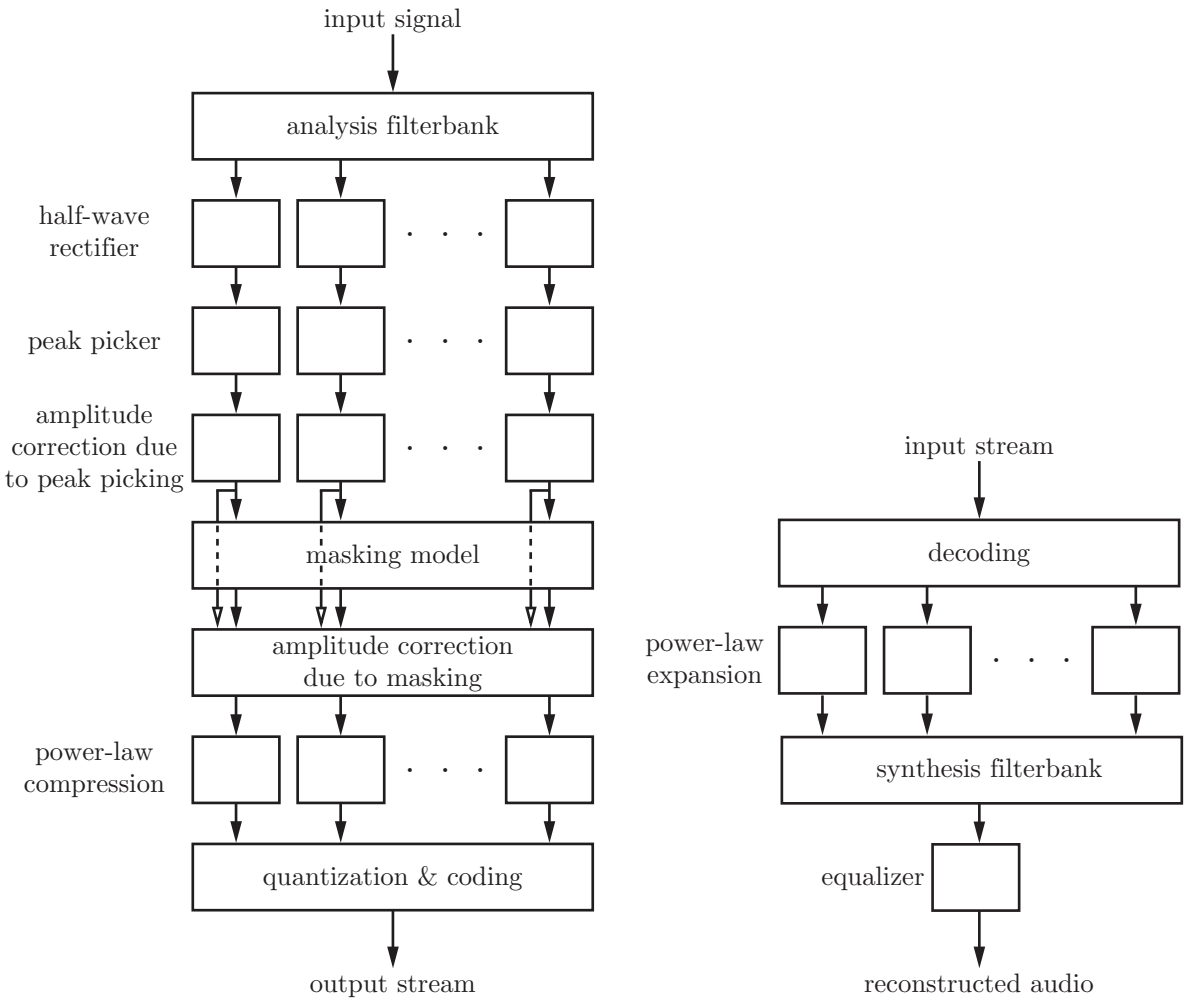


Fig. 2.5: Sparse auditory pulse coder [Feldbauer, 2005].

Matching pursuits

An alternative to the above methods is used in [Pichevar et al., 2007] where the audio signal is decomposed into gammatone impulses directy using a matching pursuits algorithm [Mallat, 1993]. This method is based on research suggesting that speech signals are matched to auditory filter responses [Smith and Lewicki, 2006], though later research disputes this [Strahl and Mertins, 2008]. The decomposition results in a description that is similar but more flexible than the above, since the Feldbauer auditory impulses are basically gammatone atoms with time, amplitude, and frequency parameters, where the frequency parameter is restricted to the frequencies of the synthesis FB. The matching pursuits approach of [Pichevar et al., 2007] has a search space that is more flexible in frequency, and also includes a chirp parameter. The drawbacks of this method are that the search space is very large and that the chirp parameter also needs to be encoded.

The major problem of pulse-based perceptual coding is the required precision of pulse timing information. This problem can be illustrated by considering two pulses in adjacent bands and overlapping in time. If the timing of one of the pulses is modified due to quantization, the pulse response due to the synthesis filters may result in additive interference or subtractive interference. In addition, on a more global scale the ear is insensitive to small timing changes. As a result, the pulse timing information tends to be encoded with higher precision than is theoretically necessary.

2.3.2 Envelopes and modulation domain processing

While auditory pulses of the methods described above can be interpreted as being a model of internal auditory events, a more abstract view is used in some coding and sound processing applications. As an alternative to the pulse-based processing, it is possible to regard the output from the gammatone filters as an amplitude and phase (or frequency) modulated sinusoid. From a coding perspective, this representation contains the same information as the gammatone filter output, but the channel signal is now represented by two generally slowly varying signals (an amplitude modulation and a phase modulation) rather than a sequence of individual impulses. From the perceptual perspective, the question of audibility of the precise pulse timing becomes

the question of how well the phase modulation is perceived, and the question of the audibility of the pulse amplitudes becomes the question of how well the amplitude modulation is perceived.

This type of representation of the audio signal by a set of modulated carriers is known as modulation domain representation and has been used in some form for several decades. The analysis functions are typically not gammatone filters, but windowed sinusoid functions at regular intervals. One such type of modulation analysis is the Short-Time Fourier Transform (STFT), used because it can be computed very efficiently.

Any natural signal can be represented using this modulation analysis. This was found originally to be useful for speech coding and was later adopted for audio coding in general. In particular, subband coding has been used for speech starting with the Channel Vocoder [Dudley, 1940]. Dudley's vocoder reproduced speech using an oscillator or noise source feeding into bandpass filters whose output is amplitude modulated, where the amplitude modulators and oscillator pitch were found by analyzing the source speech. This concept was refined with the Phase Vocoder by Flanagan and Golden [Flanagan and Golden, 1966]. In the Phase Vocoder, the analysis stage is a set of bandpass filters, where the amplitude and phase of the outputs are sampled at regular intervals. These parameters are then quantized and transmitted. Flanagan and Golden found that the channel amplitude modulations were bandlimited to about 20–30 Hz, but the phase component was not bounded. Instead, the derivative of the phase component was better behaved for transmission. The loss of the additive phase constant during reconstruction by integration was not considered a problem. Effectively, the phase vocoder thus transmitted the instantaneous frequency and amplitude of each channel. Subsequently, subband amplitude processing has found applications such as time-scale modification of audio [Laroche and Delson, 1999], noise reduction, and speech recognition [Kingsbury et al., 1998].

Of particular interest here is the research into perceptual coding of modulation *spectra*, that is, the analysis of the modulation functions in terms of their frequency content. The relative spectral transform - perceptual linear prediction (RASTA-PLP) technique by Hermansky *et al.* [Hermansky et al., 1992] was one of the first techniques to exploit the dynamics of spectral amplitudes, originally in the context of

speech recognition. RASTA-PLP is based on computing the power spectrum of the input signal, then transforming it into a log-compressed critical-band spectrum. This critical-band spectrum is bandpass filtered to remove the effects of the acoustic environment and noise. RASTA-PLP was modified and extended for use as a noise reduction method [Hermansky and Morgan, 1994], with reconstruction of the modified signal by the overlap-add technique.

A comprehensive approach to modulation spectra began with Greenberg [Greenberg, 1996]. An early method of representing speech using the modulation spectrum aimed at speech recognition [Greenberg and Kingsbury, 1997] uses a critical-band FB followed by half-wave rectification and envelope detectors. While not mentioned directly by the authors, this front-end processing can be regarded as a simplified version of the perceptual models that were developed at the same time [Dau et al., 1996a]. The auditory envelopes thus generated are then analyzed using a STFT to produce the modulation spectrogram. The modulation spectrogram is a three-dimensional representation of speech, representing the speech signal in terms of modulation frequencies within each channel over time. Like RASTA-PLP, the original motivation was in the context of speech recognition, but the modulation spectrogram concept was then investigated for perceptual modeling [Chi et al., 1999] and has strongly influenced the study of modulation patterns in auditory perception.

In [Schimmel and Atlas, 2005a;b], Schimmel and Atlas introduce a modification on the envelope processing used by Greenberg and others by proposing the use of *coherent* envelope detection for audio processing. They point out that the traditional envelope analysis results in a signal whose energy is not well concentrated in low frequency, since the "carrier" is not a simple sinusoid; this was also noted by Ghitza [Ghitza, 2001]. By replacing the envelope with a "modulator" function that can be negative, the subband signal can be decomposed into a modulator and carrier signal that that can be processed more easily. This processing has been applied successfully to source separation [Schimmel et al., 2006; Schimmel, 2007], showing that an audio signal can be reconstructed with good perceptual quality from filtered modulators. However, while useful in applications, there is no physiological correlate of the coherent modulating functions since the modulator is a complex function retaining frequency information of the carrier component. This information is generally lost at the neural

transduction stage of the auditory system.

In general, modulation domain processing is useful for audio processing and perceptual audio analysis, as evident by the diverse applications. The particulars of this modulation domain processing vary greatly but the methods presented above outline the key techniques that influenced the development of the method presented in the following chapters.

2.4 Summary

The human auditory system is a complex structure but, over the last few decades, functional models have been developed for use in audio enhancement and coding. In particular, the internal structures can be modeled using a gammatone FB, followed by nonlinear processes that mimic neural transduction, lateral inhibition, and temporal adaptation.

From their use in scientific investigation of human psychoacoustics, these models have been adapted for practical coding applications. By analyzing the audio signals using bandpass filters based on the gammatone FB model, the audio signal is represented as a set of subchannel signals that are then transmitted either as a set of auditory pulses or a set of low-frequency modulating functions. The following chapters build upon these techniques to explore a new paradigm of auditory based coding.

Chapter 3

Mathematical Background

This chapter will present the mathematical framework to analyze the perceptual reconstruction methods that will be discussed in the later chapters. After introducing the basic notation, the analysis of filters will be presented in terms of circulant matrices, which have useful properties with regards to the frequency domain. This notation is then extended to filterbanks (FBs) and FB systems with matched analysis/synthesis filters. We then introduce a brief overview of the theory of discrete finite frames and will show how it can be used to analyze the matched filter FB system. The frame algorithm and its use in the design of the matched filter FB are introduced. Finally, we discuss the reconstruction of a signal from the Hilbert envelopes of its sub-band representation by iterative refinement and examine the convergence of the reconstruction algorithm.

3.1 Notation

This thesis deals with audio processing and so the basic signals being dealt with are one-dimensional discrete-time signals. Using a sampling frequency f_s, the time index is the integer n, representing the time instant $t = n/f_s$, so

$$x[n] \equiv x(t/f_s). \tag{3.1}$$

We assume that $x(t)$ is appropriately bandlimited to avoid aliasing. Signals are in general assumed to be complex valued.

We define the inner product of $x[n]$ with another signal $y[n]$ as

$$\langle x, y \rangle = \sum_{n=-\infty}^{\infty} x[n]y^*[n], \tag{3.2}$$

and a norm

$$\|x\|^2 = \langle x, x \rangle = \sum_{n=-\infty}^{\infty} |x[n]|^2, \tag{3.3}$$

where the asterisk indicates complex conjugation.

In this thesis, we deal mostly with finite length signals. This allows simplification of the notation of many operations by using vector notation. In general, we use vectors of size N, chosen to be larger than the signal length. Zero padding is used where appropriate and this is made explicit in the text.

Using vector notation, the signal $x[n]$ of length L_x can be represented by the column vector $\mathbf{x}$ of size N by

$$\mathbf{x} = [x[0]\ x[1]\ \ldots\ x[L_x - 1]\ 0\ \ldots\ 0]^T, \tag{3.4}$$

or $\mathbf{x}_{[k]} = x[k-1]$, where $\mathbf{x}_{[k]}$ is the kth element of $\mathbf{x}$, and $x[k]$ is defined to be nonzero only for $k = 0, \ldots, L_x - 1$.

Using equally sized vectors, this allows us to rewrite the above inner product as

$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{y}^H \mathbf{x} \tag{3.5}$$

and the Euclidian norm as

$$\|\mathbf{x}\|^2 = \mathbf{x}^H \mathbf{x}, \tag{3.6}$$

using the Hermitian transpose, that is $\mathbf{x}^H$ is a row vector whose elements are the complex conjugates of those of $\mathbf{x}$.

3.1.1 Frequency domain

To analyze the discrete time signals in frequency domain, the standard Fourier Transform is used,

$$\mathcal{F}x(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}. \tag{3.7}$$

For finite length signals, we can write the Discrete Fourier Transform (DFT),

$$\mathcal{F}\mathbf{x}_{[k]} = \sum_{n=1}^{N} \mathbf{x}_{[n]} e^{-2\pi j \frac{(k-1)(n-1)}{N}}, \qquad k = 1, \ldots, N, \tag{3.8}$$

where $\mathcal{F}\mathbf{x}_{[k]}$ is the kth element of the column vector $\mathcal{F}\mathbf{x}$,[1] for $k = 1, \ldots, N$. Note that $\mathcal{F}\mathbf{x}_{[k]}$ is $\mathcal{F}x(e^{j\omega})$ sampled at $\omega = 2\pi(k-1)/N$. This transformation can be described in matrix notation by declaring an $N \times N$ Fourier transform matrix $\mathbf{W} \in \mathbb{C}^{N\times N}$ with elements

$$\mathbf{W}_{[m,n]} = \frac{1}{\sqrt{N}} e^{-2\pi j \frac{(m-1)(n-1)}{N}}, \tag{3.9}$$

where m and n are the row and column indices respectively. So, Eq. (3.8) can be restated as

$$\mathcal{F}\mathbf{x} = \sqrt{N}\mathbf{W}\mathbf{x}, \tag{3.10}$$

noting that we choose N to ensure that $\mathbf{W}$ is of the appropriate size. The scale factor $\frac{1}{\sqrt{N}}$ is introduced to make the matrix $\mathbf{W}$ unitary, so that we obtain $\mathbf{W}^{-1} = \mathbf{W}^H$. It should also be noted that $\mathbf{W}\mathbf{W}^{-1} = \mathbf{I}$, and so Parseval's theorem [Proakis and Manolakis, 1996] can be applied,

$$\|\mathbf{W}\mathbf{x}\|^2 = \mathbf{x}^H\mathbf{W}^H\mathbf{W}\mathbf{x} = \mathbf{x}^H\mathbf{x} = \|\mathbf{x}\|^2. \tag{3.11}$$

3.1.2 Subband domain

An important representation used in this thesis is the subband domain, which is obtained by filtering the signal with a set of bandpass filters. We restrict ourselves here to finite impulse response (FIR) filters, with impulse response $g_m[n]$ for the mth

[1] We avoid using bold uppercase letters for the discrete frequency representation to avoid confusion with the notation for matrices.

filter ($m = 1, \ldots, M$) of the filter bank. All impulse responses are of length L_g or less.

Linear convolution and circulant matrices

Using standard convolution notation, each channel m has the output

$$c_m[n] = \sum_{l=0}^{L_g-1} g_m[l]x[n-l], \qquad \begin{array}{l} m = 1, \ldots, M, \\ n = 0, \ldots, L_x + L_g - 1. \end{array} \tag{3.12}$$

We can now show how the convolution operation can be described as a multiplication of a vector by a matrix. In particular, we will use Circulant matrices derived from the FIR filter impulse responses. A Circulant matrix is a special form of a Toeplitz matrix, with useful properties that will be discussed below. For a comprehensive discussion of Toeplitz and Circulant matrices, see [Gray, 2006].

We express the FIR filter with response $g_m[n]$ as the matrix $\mathbf{G}_m \in \mathbb{C}^{N\times N}$, with elements

$$\mathbf{G}_{m,[l,n]} = g_m[(l-n) \bmod N] \tag{3.13}$$

or using the vector notation $\mathbf{g}_{m,[k]} = g_m[k-1]$, $\mathbf{G}_{m,[l,n]} = \mathbf{g}_{m,[((l-n) \bmod N)+1]}$, that is, each column of $\mathbf{G}_m$ is a shifted and wrapped copy of $\mathbf{g}_m$. We call $\mathbf{g}_m$ the prototype vector for $\mathbf{G}_m$.

To visualize, assume that

1. $g_m[n] = 0$ for $n < 0$ and $n \geq L_g$ (causal with impulse response length L_g),
2. $N > L_g$,

then the matrix has the form

$$\mathbf{G}_m = \begin{bmatrix} g_m[0] & 0 & \cdots & g_m[2] & g_m[1] \\ g_m[1] & g_m[0] & & g_m[3] & g_m[2] \\ g_m[2] & g_m[1] & & g_m[4] & g_m[3] \\ \vdots & & \ddots & & \vdots \\ g_m[L_g-1] & g_m[L_g-2] & & 0 & 0 \\ 0 & g_m[L_g-1] & & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & & g_m[0] & 0 \\ 0 & 0 & \cdots & g_m[1] & g_m[0] \end{bmatrix}. \tag{3.14}$$

With this definition of $\mathbf{G}_m$, the multiplication

$$\mathbf{c}_m = \mathbf{G}_m\mathbf{x}, \qquad m = 1, \ldots, M, \tag{3.15}$$

is equivalent to

$$c_m[n] = \sum_{l=0}^{N-1} g_m\big[(l-n) \bmod N\big]x[l], \qquad \begin{matrix} m = 1, \ldots, M, \\ n = 0, \ldots, N-1. \end{matrix} \tag{3.16}$$

This equation is equivalent to Eq. (3.12) if $N > (L_g + L_x)$. If this condition is not satisfied, the filtering will result in *circular convolution*, where the tail of the response (beyond index N) is added to the beginning of the filter response. This also means that while any FIR filter operating on a finite-length signal can be represented by a circulant matrix multiplication, if the vectors are large enough, the converse is not true.

While the above example of a filter is a causal FIR filter, the circulant matrix notation can be used for non-causal filters as well. Given a filter with impulse response $a[n]$ which is nonzero only for $n = -L_a + 1, \ldots, 0$, its vector representation $\mathbf{a}_{[k]}$ is nonzero for $k = 1$ and $k = N - L_a + 2, \ldots, N$. Using Eq. (3.13) to construct the circulant matrix $\mathbf{A}$, the multiplication $\mathbf{b} = \mathbf{A}\mathbf{x}$ is a valid linear convolution if $\mathbf{x}$ is zero-padded at the beginning, that is, $\mathbf{x}_{[k]}$ is zero for $k = 1, \ldots, L_a - 1$.

We now introduce some properties of Circulant matrices. Assume that $\mathbf{A}$ and $\mathbf{B}$

are Circulant matrices of equal size with eigenvalues $\{\alpha_k\}$ and $\{\beta_k\}$, respectively. The eigenvalues of $\mathbf{C}$ are $\{\gamma_k\}$.

1. Using the definition of $\mathbf{W}$ in Eq. (3.9), $\mathbf{A} = \mathbf{W}^H \text{diag}(\mathcal{F}\mathbf{a})\mathbf{W}$, that is, the columns of $\mathbf{W}$ are the eigenvectors of a circulant matrix and the elements of the DFT of the prototype vector are the eigenvalues, $\alpha_k = \mathcal{F}\mathbf{a}_{[k]}$.
2. If $\mathbf{C} = \mathbf{A} + \mathbf{B}, \mathbf{C}$ is circulant and $\gamma_k = \alpha_k + \beta_k$.
3. If $\mathbf{C} = \mathbf{A}^H, \mathbf{C}$ is circulant and $\gamma_k = \alpha_k^*$.
4. If $\mathbf{C} = \mathbf{AB}, \mathbf{C}$ is circulant, with $\gamma_k = \alpha_k \beta_k$.
5. If $\mathbf{A}$ is nonsingular, $\mathbf{C} = \mathbf{A}^{-1}$ is circulant and $\gamma_k = 1/\alpha_k$.
6. Circulant matrices commute, $\mathbf{AB} = \mathbf{BA}$.

The proofs for these properties can be found in [Gray, 2006]. However, from the first property (illustrated in Appendix A), showing the remaining is straightforward.

With respect to linear filter operations, first we consider the properties of transposing circulant matrices and of multiplying two circulant matrices. From the discussion of non-causal filters above, it should be apparent that the filter represented by the matrix $\mathbf{H}_m = \mathbf{G}_m^H$ is simply a filter with an impulse response that is the complex conjugate time-reverse of $\mathbf{g}_m$, as can be shown by exchanging the row and column indices in Eq. (3.13). The cascading of two filters $\mathbf{A}$ and $\mathbf{B}$ can be expressed using the associative property of matrix multiplication, where $\mathbf{y} = \mathbf{ABx} = \mathbf{A}(\mathbf{Bx})$. As stated above, care must be taken that the size of the vectors and matrices are such that circular convolution is avoided.

Filterbanks using circulant matrices

The vector notation can be extended to parallel FIR filterbanks. In particular, we define the subchannel signal $\mathbf{c}$ as

$$\begin{aligned} \mathbf{c} &= [c_1(0)\ c_1(1)\ \ldots\ c_1(N-1)\ c_2(0)\ \ldots\ c_M(N-1)]^T \\ &= [\mathbf{c}_1^T\ \mathbf{c}_2^T\ \ldots\ \mathbf{c}_M^T]^T, \end{aligned} \tag{3.17}$$

a column vector of size MN which is a vertical concatenation of the subchannel signal vectors. Note that the kth element of the subchannel signal $\mathbf{c}_m$ is the lth element of $\mathbf{c}$, where $l = (m-1)M + k$. Then, the filter operation of a signal $\mathbf{x}$ by a set of M filters described by $\mathbf{G}_m$ can be written as

$$\begin{aligned} \mathbf{G} &= [\mathbf{G}_1^T \ \mathbf{G}_2^T \ \ldots \ \mathbf{G}_M^T]^T, \\ \mathbf{c} &= \mathbf{G}\mathbf{x}. \end{aligned} \tag{3.18}$$

The matrix $\mathbf{G}$ is thus a *vertical* concatenation of the channel filter matrices and has an overall size of $MN \times N$. This is the analysis filterbank that splits the signal $\mathbf{x}$ into M channels.

To complement the analysis filterbank, we can construct the synthesis filterbank that combines the M channel signals back into a single channel. We define the matrix $\mathbf{H}$ as

$$\mathbf{H} = [\mathbf{H}_1 \ \mathbf{H}_2 \ \ldots \mathbf{H}_M], \tag{3.19}$$

the *horizontal* concatenation of submatrices $\mathbf{H}_m$, which are the circulant matrices representing the channel synthesis filters. This matrix is of size $N \times MN$.

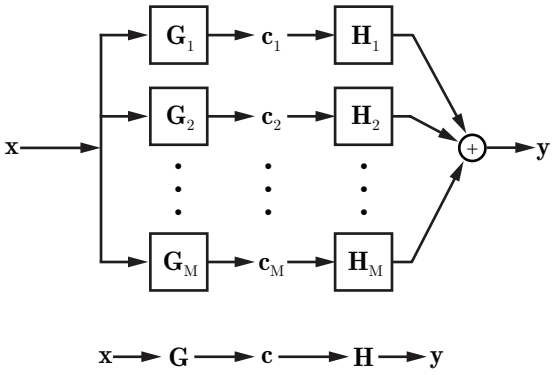


Fig. 3.1: Subchannel analysis and synthesis filters

The structure of a simple subband analysis/synthesis system is shown in Fig. 3.1.

Using the definitions of $\mathbf{G}$ and $\mathbf{H}$ we can now describe the entire system using

$$\mathbf{y} = \mathbf{HGx} = \Big(\sum_{m=1}^{M} \mathbf{H}_m \mathbf{G}_m \Big) \mathbf{x}, \tag{3.20}$$

assuming that the channels are without noise or distortion. Note that while neither $\mathbf{G}$ or $\mathbf{H}$ are circulant, the summation term of this equation, $\mathbf{U} = \sum_{m=1}^{M} \mathbf{H}_m \mathbf{G}_m = \sum_{m=1}^{M} \mathbf{U}_m$, is a circulant matrix and typically is a realizable filter whose impulse response characterizes the filterbank system.

Matched filters

It is now of interest to consider a filterbank system where each channel's synthesis filter is defined as the time-reverse complex conjugate of the analysis filter. As described above, the matrix form of these synthesis filters can be stated as $\mathbf{H}_m = \mathbf{G}_m^H$, so $\mathbf{U}_m = \mathbf{G}_m^H \mathbf{G}_m$. Using the properties of circulant matrices above, it can be shown that for all vector element indecies k,

$$\mathcal{F}\mathbf{u}_{m,[k]} = \mathcal{F}\mathbf{g}_{m,[k]}^{*} \mathcal{F}\mathbf{g}_{m,[k]} = \left| \mathcal{F}\mathbf{g}_{m,[k]} \right|^2, \tag{3.21}$$

that is, the STFT coefficients of the filter cascade in each channel are real and positive. Since $\mathbf{U} = \sum_{m=1}^{M} \mathbf{U}_m$, the STFT of the impulse response of the entire filterbank is also real and positive. We further note that a real-valued frequency domain response implies that the impulse response is non-causal and symmetric about the origin in time domain. These "matched filter" systems are well known in communication theory.

3.2 Redundant representations and frame theory

In the previous section, we introduced the notation for filterbanks using FIR filters. We used circulant matrices that can be used to describe linear filters without decimation, that is, the sample rate of the output of the filter is the same as for the input. Thus one signal is turned into multiple subband signals, all of which are at the same sampling rate as the original signal.

In this context, subband filtering can be regarded as a redundant analysis of a

signal. Clearly, the transformation of a vector $\mathbf{x}$ of size N into a vector $\mathbf{c}$ of size MN is an M-fold increase in the number of values representing the same amount of data, and thus is *potentially* largely redundant. Redundancy means that under certain conditions, given a subset of samples of the subband signals, it is possible to recover the original signal without error. In terms of the notation presented above, this means that we can recover $\mathbf{x}$ from an incomplete version of $\mathbf{c}$.

In this section, we introduce some key concepts from frame theory [Mallat, 1998], which provides a framework for analyzing redundant representations. In addition to providing a method to determine the precision with which the representation can be inverted, frame theory provides algorithms to compute the reconstruction and methods to analyze the sensitivity of the redundant representation to distortions.

The discussion here deals with discrete finite frames that we apply to the finite-signal representation of $\mathbf{x}$ and the filterbank structures described in the previous section (a treatment of discrete finite frames specifically can be found in [Pei and Yeh, 1997]). Detailed analysis of filterbanks using frame theory can be done using polyphase matrices [Bölcskei et al., 1998] or state-space representations [Chai et al., 2007]. A detailed discussion of STFT-like processing with general short-time transforms and relation to FIR filters can also be found in [Dembo and Malah, 1988]. There, reconstruction from transformed representations with and without modification is discussed, without invoking frame theory directly. The circulant matrix representation and frame theoretical approach used here is a bit simpler, at the expense of not being as general. This analysis is restricted to fully oversampled FIR filterbanks, which is sufficient in the present context.

The frame condition

A set of vectors $\{\mathbf{e}_m\}$ is a frame of the space $\mathcal{S}$ if the frame condition is satisfied,

$$A\|\mathbf{x}\|^2 \leq \sum_m |\langle \mathbf{x}, \mathbf{e}_m \rangle|^2 \leq B\|\mathbf{x}\|^2, \qquad \forall \mathbf{x} \in \mathcal{S}, \tag{3.22}$$

with $A > 0$ and $B < \infty$. This frame condition implies that $\{\mathbf{e}_m\}$ must span $\mathcal{S}$, since otherwise it would be possible to have a nonzero $\mathbf{x}$ for which $\langle \mathbf{x}, \mathbf{e}_m \rangle = 0, \forall m$, giving $A = 0$. By contradiction, if $\{\mathbf{e}_m\}$ spans $\mathcal{S}$, $A > 0$.

We can regard $b_m = \langle \mathbf{x}, \mathbf{e}_m \rangle$ as the coefficients of the redundant representation of $\mathbf{x}$. Writing $\mathbf{b} = [b_1 \ b_2 \ \dots \ b_M]^T$,

$$\mathbf{b} = \mathbf{E}\mathbf{x}, \tag{3.23}$$

where $\mathbf{E}$ is the *analysis operator*. The set of vectors $\{\mathbf{e}_m^H\}$ forms the rows of $\mathbf{E}$. We also define a *frame operator*, which we denote $\mathbf{S}$. The frame operator is defined as

$$\mathbf{S}\mathbf{x} = \sum_m \langle \mathbf{x}, \mathbf{e}_m \rangle \mathbf{e}_m, \tag{3.24}$$

so

$$\sum_m |\langle \mathbf{x}, \mathbf{e}_m \rangle|^2 = \langle \mathbf{x}, \mathbf{S}\mathbf{x} \rangle = \mathbf{x}^H \mathbf{S}\mathbf{x}, \tag{3.25}$$

and the frame condition can be rewritten as

$$A\|\mathbf{x}\|^2 \leq \mathbf{x}^H \mathbf{S}\mathbf{x} \leq B\|\mathbf{x}\|^2. \tag{3.26}$$

The frame bounds A and B are the infimum and supremum of the eigenvalues of $\mathbf{S}$. The frame operator can be found from the analysis operator by

$$\mathbf{S} = \mathbf{E}^H \mathbf{E}. \tag{3.27}$$

If the frame condition is fulfilled, $\mathbf{x}$ can be recovered completely from $\mathbf{b}$, using the *synthesis operator*

$$\mathbf{P} = \mathbf{S}^{-1}\mathbf{E}^H, \tag{3.28}$$

so

$$\mathbf{x} = \mathbf{P}\mathbf{b} = \mathbf{S}^{-1}\mathbf{E}^H \mathbf{b}. \tag{3.29}$$

It should be noted though that for redundant frames, there are infinite possible synthesis operators that recover $\mathbf{x}$ from $\mathbf{b}$ exactly. If $\mathbf{b}$ is perturbed by white noise of equal variance on all elements, Eq. (3.29) is optimal in the mean-squared error sense to recover $\mathbf{x}$ from $\mathbf{b}$, and $\mathbf{P}$ is called the *pseudo inverse* of $\mathbf{E}$.

The ratio B/A indicates the numerical properties of reconstruction, and it should be apparent how it relates to the eigenvalues of $\mathbf{S}$ and thus $\mathbf{S}^{-1}$. If $B/A = 1$, the frame is called *tight* and $S = A\mathbf{I}$, therefore $\mathbf{S}^{-1} = (1/A)\mathbf{I}$, which means that the frame

is its own inverse, where $\mathbf{x} = (1/A)\mathbf{E}^H\mathbf{b}$. If $B/A \approx 1$, the frame is called *snug*, and the property of energy preservation can be assumed with little error.

3.2.1 The frame algorithm

In some cases, it is impossible or at least impractical to find a synthesis operator. In the case of discrete finite frames, this is typically due to computational constraints making direct computation of $\mathbf{S}^{-1}$ impossible. An alternative method to find $\mathbf{x}$ from $\mathbf{b}$ is an iterative method called the frame algorithm [Daubechies, 1992] or the Extrapolated Richardson algorithm [Mallat, 1998].

The frame algorithm effectively inverts the frame by using the adjoint analysis operator with a scale factor as an approximate synthesis operator. Then the error of the estimate is corrected by projection into the frame. In effect, $\mathbf{S}^{-1}$ is approximated by a scaled identity matrix, $\gamma\mathbf{I}$; then, in the next iteration, the error between that estimate and the projection thereof onto the frame is added. Using $\hat{\mathbf{x}}^{(i)}$ as the ith estimate of the original signal $\mathbf{x}$, the full update equation is

$$\hat{\mathbf{x}}^{(i)} = \gamma\mathbf{E}^H\mathbf{b} + (\mathbf{I} - \gamma\mathbf{S})\hat{\mathbf{x}}^{(i-1)}, \tag{3.30}$$

where we typically set $\hat{\mathbf{x}}^{(0)} = \mathbf{0}$. Note that we use $x^{(i)}$ as the value of x in the ith iteration of the iterative loop, and x^i to indicate x raised to the ith power.

We can calculate the error at the ith iteration as

$$\begin{aligned}
\mathbf{x} - \hat{\mathbf{x}}^{(i)} &= \mathbf{x} - \gamma\mathbf{S}\mathbf{x} + (\mathbf{I} - \gamma\mathbf{S})\hat{\mathbf{x}}^{(i-1)} \\
&= (\mathbf{I} - \gamma\mathbf{S})\mathbf{x} + (\mathbf{I} - \gamma\mathbf{S})\hat{\mathbf{x}}^{(i-1)} \\
&= (\mathbf{I} - \gamma\mathbf{S})^i\mathbf{x}.
\end{aligned} \tag{3.31}$$

As noted above, if the frame defined by $\mathbf{S}$ is tight, $\mathbf{S}^{-1} = (1/A)\mathbf{S}$, and the estimate will be the exact solution on the first iteration if γ is chosen to be $1/A$. If the fame is *not* tight, we can calculate a bound on the error for each iteration $\mathbf{e}_x^{(i)} = \mathbf{x} - \hat{\mathbf{x}}^{(i)}$, which depends on the frame bounds and the scale factor. Using $\mathbf{R} = (\mathbf{I} - \gamma\mathbf{S})$, the

norm of the error can be calculated simply as

$$\begin{aligned}\|\mathbf{e}_x^{(i)}\|^2 &= \|\mathbf{R}^i\mathbf{x}\|^2 \\ &\leq \|\mathbf{R}^i\|^2\|\mathbf{x}\|^2,\end{aligned} \tag{3.32}$$

where $\|\mathbf{R}^i\|^2$ is the spectral norm, which can be found from the largest eigenvalue of $(\mathbf{R}^i)^H(\mathbf{R}^i)$. Recall that the eigenvalues of $\mathbf{S}$ are bounded by the frame bounds A and B (see Eq. (3.26)), so the eigenvalues of $\mathbf{R}$ are bounded by $|1-\gamma A|$ and $|1-\gamma B|$ and the norm of $\mathbf{R}^i$ can be found from A, B and γ as

$$\|\mathbf{R}^i\|^2 = \max(|1-\gamma A|^{2i}, |1-\gamma B|^{2i}). \tag{3.33}$$

If the frame bounds are known, we can find the value of γ to minimize the norm and thus the error, with

$$\begin{aligned}\|\mathbf{e}_x^{(i)}\|^2 &\leq \min_{\gamma} \max(|1-\gamma A|^{2i}, |1-\gamma B|^{2i})\|\mathbf{x}\|^2 \\ &= \left(\frac{B-A}{B+A}\right)^{2i}\|\mathbf{x}\|^2,\end{aligned} \tag{3.34}$$

for

$$\gamma = \frac{2}{B+A}. \tag{3.35}$$

Since $\frac{B-A}{B+A} < 1$ for $A, B \neq 0$, the frame algorithm is guaranteed to converge for this value of γ.

3.2.2 Frame theory and filterbanks

We can now use frame theory to analyze FBs using the circulant matrices. Using the notation from the previous section, the analysis FB $\mathbf{G}$ is the analysis operator and the subchannel signals $\mathbf{c}$ are the frame coefficients. If the frame formed by this analysis operator satisfies the frame condition, that is $\mathbf{S} = \mathbf{G}^H\mathbf{G}$ satisfies Eq. (3.26), the original signal $\mathbf{x}$ can be reconstructed from the subchannel signals. However, in the context of circulant matrices representing FBs, Eq. (3.28) cannot be used to design a synthesis filter for $\mathbf{G}$. While $\mathbf{S}$ is nonsingular (by satisfying the frame condition) and circulant (since $\mathbf{S} = \mathbf{G}^H\mathbf{G} = \sum_{m=1}^{M} \mathbf{G}_m^H\mathbf{G}_m$), a synthesis filter constructed by

$\mathbf{H} = \mathbf{S}^{-1}\mathbf{G}^H$ would not be realizable as a linear filter. The submatrices $\mathbf{H}_m$ of $\mathbf{H} = [\mathbf{H}_1\ \mathbf{H}_2\ \ldots\ \mathbf{H}_M]$ in this case would be circulant, but it is not possible to guarantee that the impulse response length is bounded, that is, the order of $\mathbf{H}_m$ might be too high[2]. Thus, it can not be guaranteed that the resulting circulant matrix can be realized as a linear time-invariant filter due to circular convolution.

In Section 3.1.2, we introduced the matched filter filterbank and noted that the system response is defined as $\mathbf{U} = \sum_{m=1}^{M} \mathbf{U}_m = \sum_{m=1}^{M} \mathbf{G}_m^H \mathbf{G}_m$. The notation shows the equivalence of the matched filter system to the projection of the signal onto the frame defined by the analysis filter. Furthermore, if we now modify the synthesis filterbank by adding a scale factor γ, the estimate $\hat{\mathbf{x}}$ of $\mathbf{x}$ if passed through the matched filterbank system can be stated as

$$\hat{\mathbf{x}} = \gamma \mathbf{G}^H \mathbf{c}. \tag{3.36}$$

Comparing Eq. (3.36) with the frame algorithm update equation Eq. (3.30) it can be observed that Eq. (3.36) is simply the first iteration of the frame algorithm (assuming $\mathbf{x}^{(0)} = \mathbf{0}$). Thus, we can apply the same equations as above to determine the bound on the reconstruction error $\|\mathbf{x} - \hat{\mathbf{x}}\|^2$. By substituting $i = 1$, the upper bound of the error is

$$\|\mathbf{x} - \hat{\mathbf{x}}\|^2 \leq \max(|1 - \gamma A|^2, |1 - \gamma B|^2)\|\mathbf{x}\|^2, \tag{3.37}$$

where A and B are the bounds of the frame defined by $\mathbf{G}^H\mathbf{G}$. As above, the optimal scale factor if A and B are known is $\gamma = 2/(A + B)$ and the resulting maximal error is

$$\|\mathbf{x} - \hat{\mathbf{x}}\|^2 \leq \left(\frac{B - A}{B + A}\right)^2 \|\mathbf{x}\|^2. \tag{3.38}$$

Frequency-weighted error

The error between the original and reconstructed signal as calculated above assumes that any error present in the reconstruction is equally significant. When dealing with audio signals this is not necessarily true since some types of errors (such as a DC

[2]This can be thought of as designing a filter in frequency domain using the inverse STFT. Given just the power spectrum, we cannot tell if the time-domain impulse response will be shorter than the transform size.

offset) are completely inaudible. The proper way to handle this is of course to use an auditory model and this will be discussed in more detail later. However, we can introduce the concept of frequency-weighted error here, where the error for some parts of the frequency spectrum is considered less significant.

A simple way to handle this is to calculate the error signal in frequency domain using the $\mathbf{W}$ matrix as defined in Section 3.1.1. We define the frequency domain weighting using a vector $\mathbf{v}$ scaling the Fourier transform values of the error, giving the weighted error

$$\|\mathbf{e}_{\mathrm{W}}\|^2 = \|\mathrm{diag}\{\mathbf{v}\}\mathbf{W}(\mathbf{x} - \hat{\mathbf{x}})\|^2. \tag{3.39}$$

The error bound in this case depends on the range of the eigenvalues of $\mathrm{diag}\{\mathbf{v}\}\mathbf{W}(\mathbf{I} - \gamma\mathbf{G}^H\mathbf{G})$. Finding γ to minimize the norm of this expression is relatively simple if the eigenvalues of $\mathbf{G}^H\mathbf{G}$ are known[3].

3.3 Signal estimation from modified subband signals

In the previous discussion, the focus has been on reconstruction of the original signal $\mathbf{x}$ with full knowledge of the subchannel signals $\mathbf{c}$ and with constraints on the reconstruction filters. In this section, we will reverse this problem to some degree. Here, we consider the problem of estimating a signal $\hat{\mathbf{x}}$ given a *modified* channel signal $\mathbf{c}'$ that may have been obtained from some "original" $\mathbf{x}$, but what is more important in this context is that we find the estimate $\hat{\mathbf{x}}$ that minimizes the error between $\mathbf{G}\hat{\mathbf{x}}$ ($\hat{\mathbf{x}}$ when passed through the analysis filter) and $\mathbf{c}'$. Later in this section, we will consider finding an estimate of $\mathbf{x}$ given the *envelopes* of $\mathbf{c}$ to prepare for the discussion of auditory model inversion in the next chapter. The discussion here will closely follow a paper by Griffin and Lim [Griffin and Lim, 1984] that outlined a method to estimate a signal given STFT magnitudes.

We define the error to be minimized as

$$D(\mathbf{c}', \mathbf{G}\hat{\mathbf{x}}) = \|\mathbf{c}' - \mathbf{G}\hat{\mathbf{x}}\|^2. \tag{3.40}$$

[3]In Chapter 5 we use a weighting vector that is simply binary by measuring the frame bounds over a limited frequency range.

Expanding this and finding the gradient[4], we get

$$\frac{\partial D(\mathbf{c}', \mathbf{G}\hat{\mathbf{x}})}{\partial \hat{\mathbf{x}}^*} = -\mathbf{G}^H\mathbf{c}' + \mathbf{G}^H\mathbf{G}\hat{\mathbf{x}}, \tag{3.41}$$

and equating Eq. (3.41) to zero, the signal that minimizes the difference is

$$\hat{\mathbf{x}} = (\mathbf{G}^H\mathbf{G})^{-1}\mathbf{G}^H\mathbf{c}'. \tag{3.42}$$

This result is equal to the synthesis frame stated in Eq. (3.29) and many of the same arguments from frame theory apply. Thus if $(\mathbf{G}^H\mathbf{G})^{-1}$ cannot be implemented, the frame algorithm can be used to obtain the estimate iteratively, or, if $\mathbf{G}$ can be designed to form a snug frame, the Hermitian transpose serves as a synthesis operator with small error.

3.3.1 Envelopes and carriers

We now define the concept of envelopes as a particular modification of a subband signal. An envelope of a signal is generally taken to be a non-negative real quantity that is always larger than the magnitude of its underlying signal, so

$$\overline{x}[n] \geq |x[n]|. \tag{3.43}$$

For complex signals, typically this equation is used with the equality, so we use

$$\overline{\mathbf{c}}_{[k]} = |\mathbf{c}_{[k]}| \tag{3.44}$$

as the envelopes[5] of the subband signals. The envelope of real signals is often obtained from the Hilbert envelope

$$\overline{x}[n] = |x[n] + j\mathcal{H}\{x\}[n]|, \tag{3.45}$$

[4]We use differentiation with respect to $\mathbf{x}^*$, see [Haykin, 1999].

[5]We use an overline $(\bar{\cdot})$ to indicate the envelope in general and the absolute value notation $(|\cdot|)$ as the magnitude specifically.

where $\mathcal{H}\{x\}[n]$ is the nth sample of the Hilbert transform of the signal x. Alternative definitions often relax the requirement of the envelope to be non-negative (calling it the *amplitude* or *modulator* instead) in order to ensure that the resulting signal is continuous or closed under convolution [Cohen et al., 1999; Schimmel, 2007]. Since we generally assume signals to be complex, we only use the absolute value operator.

Complementary to the envelopes, we define the carrier components $\mathring{\mathbf{c}}_{[k]}$ such that

$$\mathbf{c}_{[k]} = \overline{\mathbf{c}}_{[k]}\mathring{\mathbf{c}}_{[k]}. \tag{3.46}$$

By using the magnitude as envelope, the carrier is the phase component of the complex signal with unit magnitude and thus can be found using $\mathring{\mathbf{c}}_{[k]} = \mathbf{c}_{[k]}/|\mathbf{c}_{[k]}|$. If $|\mathbf{c}_{[k]}|$ is 0, we define $\mathring{\mathbf{c}}_{[k]} = 1$, but typically we will avoid this problem as described below. Thus, $\overline{\mathbf{c}}$ and $\mathring{\mathbf{c}}$ are the envelopes and carriers[6] of $\mathbf{c}$.

3.3.2 Estimating a signal from the subband envelopes

In this section we consider the problem of finding a signal estimate based on subband envelopes. The relationship of subband envelopes to perceptual representations of audio signals will be discussed in the following chapter, but in the remainder of this chapter, we approach the problem from a more general viewpoint.

Of interest is the error between the given envelopes $\overline{\mathbf{c}}$ and the subband envelopes of the estimate $\hat{\mathbf{x}}$, so we modify Eq. (3.40) such that

$$D_M(\overline{\mathbf{c}}, \overline{\mathbf{G}\hat{\mathbf{x}}}) = \| \, |\mathbf{c}| - |\mathbf{G}\hat{\mathbf{x}}| \, \|^2. \tag{3.47}$$

Note that the expression $|\mathbf{c}|$ indicates a vector of the same size as $\mathbf{c}$ with all elements being the magnitude of the corresponding element in $\mathbf{c}$, while the notation $\|\mathbf{c}\|^2$ is the (scalar) norm of $\mathbf{c}$. A closed-form expression of $\hat{\mathbf{x}}$ minimizing $D_M(\overline{\mathbf{c}}, \overline{\mathbf{G}\hat{\mathbf{x}}})$ is not a simple task due to the nonlinearity of the envelope operator. Some recent research shows that direct reconstruction from magnitude-only is possible for certain classes of frames [Balan et al., 2006; Bodmann et al., 2008]. Since we aim to develop a method for a frame representing a perceptual analysis, we use an iterative method based on

[6] Recall that $\mathbf{c}$ is the collection of subband signals $\mathbf{c}_m$ for $m = 1, \ldots, M$.

[Griffin and Lim, 1984] to find the estimate of $\hat{\mathbf{x}}$ using convex projection.

Griffin and Lim described a method to estimate a signal from the modified STFT (MSTFT) and modified STFT magnitude (MSTFTM) for use in time-scale modification of speech and speech enhancement. Using a windowed STFT, a signal $x[n]$ is represented by

$$X_W(mS,\omega) = \sum_{n=-\infty}^{\infty} w[mS-n]x[n]e^{-j\omega n}, \tag{3.48}$$

where $w[n]$ is a window function, S is a constant integer indicating the "step size" in samples from one segment to the next, m is the segment index, and ω is the frequency index within each segment. Noting that for an arbitrary *modified* STFT $Y_W(mS,\omega)$ in general there is no sequence whose STFT is given by $Y_W(mS,\omega)$, instead Griffin and Lim derive a method to find the sequence $x[n]$ whose STFT $X_W(mS,\omega)$ minimizes the distance

$$D\big(x[n], Y_W(mS,\omega)\big) = \sum_{m=-\infty}^{\infty} \frac{1}{2\pi} \int_{\omega=-\pi}^{\pi} |X_W(mS,\omega) - Y_W(mS,\omega)|^2 d\omega, \tag{3.49}$$

using

$$x[n] = \frac{\sum_{m=-\infty}^{\infty} w[mS-n]y_W[mS,n]}{\sum_{m=-\infty}^{\infty} w^2[mS-n]}, \tag{3.50}$$

where $y_W[mS,n] = \frac{1}{2\pi}\int_{\omega=-\pi}^{\pi} Y_W(mS,\omega)e^{j\omega n}d\omega$.

This method is termed least-squares error estimation from the MSTFT (LSEE-MSTFT). The function to find $x[n]$ from the MSTFT is then extended to the MSTFTM. Derived from image-processing methods, their method is an iterative algorithm that, at each iteration, decreases the error measure

$$D_M\big(x[n], |Y_W(mS,\omega)|\big) = \sum_{m=-\infty}^{\infty} \frac{1}{2\pi} \int_{\omega=-\pi}^{\pi} \big[|X_W(mS,\omega)| - |Y_W(mS,\omega)|\big]^2 d\omega. \tag{3.51}$$

Key to the iterative algorithm is the computation of the MSTFT estimate $\hat{X}_W^{(i)}(mS,\omega)$ at each iteration by combining the phase component of the previous estimate with

the MSTFTM $|Y_W(mS,\omega)|$,

$$\hat{X}_W^{(i)}(mS,\omega) = |Y_W(mS,\omega)|\frac{X_W^{(i)}(mS,\omega)}{|X_W^{(i)}(mS,\omega)|}, \tag{3.52}$$

from which the sequence $x^{(i+1)}[n]$ is calculated using LSEE-MSTFT,

$$x^{(i+1)}[n] = \frac{\sum_{m=-\infty}^{\infty}\frac{1}{2\pi}\int_{\omega=-\pi}^{\pi}\hat{X}_W(mS,\omega)e^{j\omega n}d\omega}{\sum_{m=-\infty}^{\infty}w^2[mS-n]}. \tag{3.53}$$

The similarity to the problem of finding $\mathbf{x}$ given $\overline{\mathbf{c}}$ should be apparent. We replace $|Y_W(mS,\omega)|$ with $\overline{\mathbf{c}}$, which are the envelopes to which the signal must be fitted, and $X_W^{(i)}(mS,\omega)$ with $\mathbf{c}^{(i)}$, which is the subchannel decomposition of the current estimate $\hat{\mathbf{x}}^{(i)}$. The frame theory motivated method to find $\mathbf{x}$ from modified subband signals as discussed in Section 3.3 is analogous to LSEE-MSTFT, and we can modify Eq. (3.52) to become

$$\hat{\mathbf{c}}_{[k]}^{(i)} = \overline{\mathbf{c}}_{[k]}\frac{\mathbf{c}_{[k]}^{(i)}}{|\mathbf{c}_{[k]}^{(i)}|}, \qquad k = 1,\ldots,(MN), \tag{3.54}$$

or $\hat{\mathbf{c}}_{[k]}^{(i)} = \overline{\mathbf{c}}_{[k]}\mathring{\mathbf{c}}_{[k]}^{(i)}$, where $\mathbf{c}^{(i)} = \mathbf{G}\hat{\mathbf{x}}^{(i)}$. This equation gives us the subchannel signals with the envelopes constrained to $\overline{\mathbf{c}}$.

The next estimate is then calculated using the equivalent to Eq. (3.53),

$$\hat{\mathbf{x}}^{(i+1)} = \mathbf{H}\hat{\mathbf{c}}^{(i)} \tag{3.55}$$

which in terms of elements of $\hat{\mathbf{x}}^{(i+1)}$ can be stated as

$$\hat{\mathbf{x}}_{[n]}^{(i+1)} = \sum_{k=1}^{MN}\mathbf{H}_{[n,k]}\overline{\mathbf{c}}_{[k]}\mathring{\mathbf{c}}_{[k]}^{(i)}. \tag{3.56}$$

We label this sequence as the *refinement function* $\hat{\mathbf{x}}^{(i+1)} = R(\hat{\mathbf{x}}^{(i)},\overline{\mathbf{c}})$, the function that generates a better guess of a signal matching $\overline{\mathbf{c}}$ given a previous estimate.

Visually the structure of this iterative loop is shown in Fig. 3.2, showing how the output $\hat{\mathbf{x}}^{(i)}$ is generated purely from $\overline{\mathbf{c}}$. Note that the analysis filter $\mathbf{G}$ calculates the channel signals from the *previous* estimate and that for $\mathbf{c}_{[k]}^{(i)} = 0$, we define

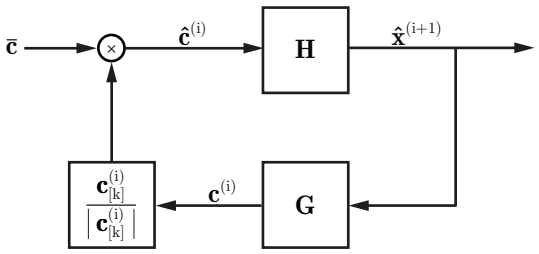


Fig. 3.2: Iterative loop to find estimate $\hat{\mathbf{x}}$ from envelopes $\overline{\mathbf{c}}$.

$\mathbf{c}^{(i)}_{[k]}/|\mathbf{c}^{(i)}_{[k]}| = 0$.

We address here the issue of circular convolution. Consider the filter cascade of $\hat{\mathbf{c}}^{(i)}$ being first passed through the filterbank $\mathbf{H}$ then through the filterbank $\mathbf{G}$. This is a *transmultiplexing* system: the M signals in $\hat{\mathbf{c}}^{(i)} \in \mathbb{C}^{MN}$ are combined into one signal $\hat{\mathbf{x}}^{(i+1)}_{[n]} \in \mathbb{C}^{N}$, then expanded back out into M signals $\mathbf{c}^{(i+1)} \in \mathbb{C}^{MN}$. Assume that all filters in $\mathbf{H}$ are anticausal (or strictly non-causal, with zero causal impulse response) of length L_h and all filters in $\mathbf{G}$ are causal of length L_g. Then, in order to avoid circular convolution in the iterative estimation loop, all subchannel *envelopes* $\overline{\mathbf{c}}_m$ must be zero-padded at the beginning with L_h zeros and at the end with L_g zeros: Eq. (3.54) will force these zeros onto $\hat{\mathbf{c}}^{(i)}$ and these subchannel signals in any iteration will result in an estimate $\hat{\mathbf{x}}^{(i+1)}_{[n]}$ without circular convolution yet zero padded by L_g samples at the end. Thus the next subchannel signals $\mathbf{c}^{(i+1)}$ are obtained without circular convolution.

3.3.3 Convergence

When using iterative algorithms, the question of stability and convergence must be addressed. An algorithm that is unstable or that converges only very slowly is useless in any practical setting since computational resources are finite. It is also useful to know how likely the algorithm is to converge to a local rather than a global minimum.

Given the above notation for the iterative estimation algorithm, we can now show what conditions must be met for the result $\hat{\mathbf{x}}^{(i)}$ to converge to a solution. A common method to use is the global convergence theorem [Luenberger, 1973], which works for general nonlinear iterative descent algorithms. Similar algorithms have been shown to converge [Tom et al., 1981; Quatieri et al., 1981] but do so by assuming that the

update equation uses nonexpansive transforms. This is not necessary for the global convergence theorem.

We define $\mathbf{x}, \hat{\mathbf{x}}^{(i)} \in \mathcal{S}$, and the solution set as $\Gamma \subset \mathcal{S}$. This means that if $\hat{\mathbf{x}}^{(i)} \in \Gamma$, $\hat{\mathbf{x}}^{(i)}$ is considered an acceptable final output from the iterative procedure. The refinement function Eq. (3.55) generates a sequence $\{\hat{\mathbf{x}}^{(i)}\}_{i=0}^{\infty}$, starting from $\hat{\mathbf{x}}^{(0)}$.

Applying the global convergence theorem states that for the functions $\hat{\mathbf{x}}^{(i+1)} = R(\hat{\mathbf{x}}^{(i)}, \overline{\mathbf{c}})$ and $D_M(\overline{\mathbf{c}}, \overline{\mathbf{G}\hat{\mathbf{x}}})$, if

1. all points $\hat{\mathbf{x}}^{(i)}$ are contained in a compact set $\Psi \subset \mathcal{S}$,
2. (a) $D_M(\overline{\mathbf{c}}, \overline{\mathbf{G}\hat{\mathbf{x}}^{(i+1)}}) < D_M(\overline{\mathbf{c}}, \overline{\mathbf{G}\hat{\mathbf{x}}^{(i)}})$, for $\mathbf{x}^{(i)} \notin \Gamma$ and
   (b) $D_M(\overline{\mathbf{c}}, \overline{\mathbf{G}\hat{\mathbf{x}}^{(i+1)}}) \leq D_M(\overline{\mathbf{c}}, \overline{\mathbf{G}\hat{\mathbf{x}}^{(i)}})$, for $\mathbf{x}^{(i)} \in \Gamma$
3. the mapping defined by $R(\hat{\mathbf{x}}^{(i)}, \overline{\mathbf{c}})$ is closed outside Γ,

then the limit of any convergent subsequence of $\{\hat{\mathbf{x}}^{(i)}\}$ is a solution.

Of these conditions, the first one can be interpreted as requiring that the sequence cannot *diverge* even if the sequence does not converge. For example, the sequence $\{\sin(kl)\}_{l=0}^{\infty}$ does not converge to a limit as $l \to \infty$ (for $k \neq 0$), yet it is a compact set with interval $[-1, 1]$. In the context of estimating the signal $\mathbf{x}$ from the envelopes $\overline{\mathbf{c}}$, this means that we must ensure that any guess generated by the refinement function is finite and bounded for all $\hat{\mathbf{x}}^{(i)}$. This follows simply from the property of $\mathring{\mathbf{c}}_{[k]}^{(i)}$ being closed and bounded: $|\mathring{\mathbf{c}}_{[k]}^{(i)}| = 1$ for all k. From Eq. (3.56), each element of $\hat{\mathbf{x}}^{(i)}$ is a linear combination of elements in $\mathring{\mathbf{c}}^{(i)}$, thus the elements in $\hat{\mathbf{x}}^{(i)}$ are also in a closed and bounded set, satisfying the first condition.

The second condition states that the function $R(\hat{\mathbf{x}}^{(i)}, \overline{\mathbf{c}})$ must be a *strictly* decreasing function with respect to the error measure $D_M(\overline{\mathbf{c}}, \overline{\mathbf{G}\hat{\mathbf{x}}^{(i)}})$, *unless* the solution set has been reached. Once the solution set has been reached, the error must either decrease or remain the same for any further iterations. This condition requires detailed analysis of the refinement algorithm and the function that computes the error. In the context of estimation from modified envelopes, Γ is a disconnected region, since there are usually multiple "correct" estimates (for example, if some signal $\hat{\mathbf{x}}$ is a solution, the signal $-\hat{\mathbf{x}}$ will also be a solution). Thus, the algorithm would be allowed to jump between multiple points within Γ yet still be considered as having converged.

Following the steps of [Griffin and Lim, 1984], we show that $D_M(\overline{\mathbf{c}}, \overline{\mathbf{G}\hat{\mathbf{x}}^{(i)}})$ decreases at every step under certain conditions, by using the error measure described by Eq. (3.40). Constraining $\hat{\mathbf{c}}^{(i)}$ to magnitude $\overline{\mathbf{c}}$ using Eq. (3.54) minimizes $D(\hat{\mathbf{c}}^{(i)}, \mathbf{G}\hat{\mathbf{x}}^{(i)})$ for $\hat{\mathbf{x}}^{(i)}$ fixed. Thus we have

$$D(\hat{\mathbf{c}}^{(i)}, \mathbf{G}\hat{\mathbf{x}}^{(i)}) \leq D(\hat{\mathbf{c}}^{(i-1)}, \mathbf{G}\hat{\mathbf{x}}^{(i)}), \tag{3.57}$$

since the elements of $\hat{\mathbf{c}}^{(i)}$ are of the same magnitude as the elements in $\hat{\mathbf{c}}^{(i-1)}$, but the phase component is aligned with $\mathbf{G}\hat{\mathbf{x}}^{(i)}$. We also write out the next iteration as

$$D(\hat{\mathbf{c}}^{(i+1)}, \mathbf{G}\hat{\mathbf{x}}^{(i+1)}) \leq D(\hat{\mathbf{c}}^{(i)}, \mathbf{G}\hat{\mathbf{x}}^{(i+1)}). \tag{3.58}$$

Now given a way to estimate the next signal $\hat{\mathbf{x}}^{(i+1)}$ which minimizes $D(\hat{\mathbf{c}}^{(i)}, \mathbf{G}\mathbf{x})$ for any given $\hat{\mathbf{c}}^{(i)}$, we find that

$$D(\hat{\mathbf{c}}^{(i)}, \mathbf{G}\hat{\mathbf{x}}^{(i+1)}) \leq D(\hat{\mathbf{c}}^{(i)}, \mathbf{G}\hat{\mathbf{x}}^{(i)}), \tag{3.59}$$

where equality holds if and only if $\hat{\mathbf{x}}^{(i+1)} = \hat{\mathbf{x}}^{(i)}$ since otherwise $\hat{\mathbf{x}}^{(i+1)}$ is not at the minimum. We now combine Eq. (3.58) with Eq. (3.59) to get

$$D(\hat{\mathbf{c}}^{(i+1)}, \mathbf{G}\hat{\mathbf{x}}^{(i+1)}) \leq D(\hat{\mathbf{c}}^{(i)}, \mathbf{G}\hat{\mathbf{x}}^{(i)}). \tag{3.60}$$

We note now that $\hat{\mathbf{c}}^{(i)}$ and $\mathbf{G}\hat{\mathbf{x}}^{(i)}$ by Eq. (3.54) always have the same phase for all elements, thus we expand $D(\hat{\mathbf{c}}^{(i)}, \mathbf{G}\hat{\mathbf{x}}^{(i)})$ element-by-element

$$\begin{aligned}
D(\hat{\mathbf{c}}^{(i)}, \mathbf{G}\hat{\mathbf{x}}^{(i)}) &= \langle \hat{\mathbf{c}}^{(i)} - \mathbf{c}^{(i)}, \hat{\mathbf{c}}^{(i)} - \mathbf{c}^{(i)} \rangle \\
&= \sum_k \left| \hat{\mathbf{c}}^{(i)}_{[k]} - \mathbf{c}^{(i)}_{[k]} \right|^2 \\
&= \sum_k \left| (\overline{\mathbf{c}}_{[k]} - |\mathbf{c}^{(i)}_{[k]}|) \frac{\mathbf{c}^{(i)}_{[k]}}{|\mathbf{c}^{(i)}_{[k]}|} \right|^2 \\
&= \sum_k \left| \overline{\mathbf{c}}_{[k]} - |\mathbf{c}^{(i)}_{[k]}| \right|^2 \\
&= D_M(\overline{\mathbf{c}}, \overline{\mathbf{G}\hat{\mathbf{x}}^{(i)}}).
\end{aligned} \tag{3.61}$$

Hence we show that Eq. (3.60) can be rewritten as

$$D_M(\overline{\mathbf{c}}, \overline{\mathbf{G}\hat{\mathbf{x}}^{(i+1)}}) \leq D_M(\overline{\mathbf{c}}, \overline{\mathbf{G}\hat{\mathbf{x}}^{(i)}}), \tag{3.62}$$

with equality if and only if $\hat{\mathbf{x}}^{(i+1)} = \hat{\mathbf{x}}^{(i)}$, showing that the distance measure always decreases until a fixed value is reached. At this point, the envelopes $\overline{\mathbf{G}\hat{\mathbf{x}}^{(i)}}$ will not change from one iteration to the next and thus the gradient $\nabla D_M(\overline{\mathbf{c}}, \overline{\mathbf{G}\hat{\mathbf{x}}})$ with respect to $\hat{\mathbf{x}}$ will be zero, indicating $\hat{\mathbf{x}}^{(i)} \in \Gamma$, with Γ closed.

However, note that Eq. (3.59) is only true if we can calculate $\hat{\mathbf{x}}^{(i+1)}$ to minimize $D(\hat{\mathbf{c}}^{(i)}, \mathbf{G}\hat{\mathbf{x}}^{(i+1)})$ *exactly*. In the framework of a synthesis filterbank, this can be guaranteed only if the analysis filterbank is a tight frame as described in Section 3.2.2. The approximation of the optimal synthesis operator by the matched filterbank if the frame is snug rather than tight results in the possibility of a small additive error to the estimates $\hat{\mathbf{x}}^{(i+1)}$ and $\mathbf{c}^{(i+1)}$. The bound can be calculated from the frame bounds as described in Section 3.2.2 and thus it is important to ensure the filterbank frame is as close to tight as possible. It is possible to add more terms of the frame algorithm to compensate for a non-tight frame but this would be at the expense of higher computational cost.

Finally, we consider the third condition, the requirement to have a closed mapping. The concept of a closed mapping applies to point-to-set mappings and is a general form of the concept of continuity in point-to-point mappings. Since the function $R(\hat{\mathbf{x}}^{(i)}, \overline{\mathbf{c}})$ is a point-to-point mapping, we need to show that it is continuous. It should be apparent, however, that the given function is *not* continuous in general. At issue is the calculation of the carrier in Eq. (3.54), which clearly is not continuous if $\mathbf{c}_{[k]}^{(i)}$ is infinitesimally small. Typically this is not a problem since in the iterative loop this will only occur at samples where $\overline{\mathbf{c}}$ is also very small, thus any small change that will cause the phase to 'flip' at any given subchannel sample $\mathbf{c}_{[k]}^{(i)}$ is likely to also be a small change in the magnitude constrained $\hat{\mathbf{c}}_{[k]}^{(i)}$. Interestingly, in [Griffin and Lim, 1984] it is claimed that the update equation *is* continuous without detailed proof, even though the update function contains a similar phase term, as stated in Eq. (3.52). However, unlike our analysis using discrete finite frames, LSEE-MSTFTM is analyzed in the continuous frequency domain and thus can behave differently at the critical points.

3.3.4 Implementation issues

From the analysis above, it must be concluded that for the algorithm we present to estimate a signal from subchannel envelopes using a physically realizable FIR filterbank, convergence to a global minimum *cannot* be guaranteed. However, the analysis highlights the reasons that prevent convergence and thus we can design a workaround to ensure that the algorithm will at least find a reasonable suboptimal estimate. The definition of a reasonable estimate will depend on the application. This will be explained for audio signals specifically in the next chapter and is used in the implementation of the iterative algorithm to terminate processing.

Simulated annealing

We note that for nonlinear iterative algorithms in general, the proof of convergence does not guarantee speed of convergence nor that the solution is reachable in a finite number of iterations. Thus it is possible that the gradient becomes very shallow with the estimate staying near a local minimum even if convergence to a global minimum is guaranteed. Often, it is possible to perturb the current estimate by a small random amount and reach a state where the gradient is more steep. An example of such a method is the technique known as simulated annealing, which models an annealing melt going from a hot chaotic state to a near optimal crystalline structure. Originally proposed for discrete optimization [Kirkpatrick et al., 1983; Aarts and Korst, 1989] and later adapted to continuous functions [Vanderbilt and Louie, 1984; Corana et al., 1987], simulated annealing can be found in several applications including perceptual audio coding [Holters and Zölzer, 2009].

By itself, Eq. (3.54) is undefined for $\mathbf{c}^{(i)}_{[k]} = 0$, which is a rare occurrence if the input signal is nonzero. To completely avoid the need to handle this case, we can make a small modification to the algorithm. At the same time, this change will introduce a controllable amount of perturbation, implementing a simplified version of simulated annealing. This modification is to change the step which finds the next set of carriers from the current signal estimate estimate from $\mathbf{c}^{(i)} = \mathbf{G}\hat{\mathbf{x}}^{(i)}$ to

$$\mathbf{c}^{(i)} = \mathbf{G}\hat{\mathbf{x}}^{(i)} + \mu\mathbf{r} \tag{3.63}$$

where $\mathbf{r}$ is a vector of MN zero-mean random variables of unit variance and μ is a scale factor to control the perturbation. The scale factor can be controlled dynamically, being effectively the "temperature" control of the simulated annealing process, but should never be reduced to 0. A typical control of μ would be to observe the error gradient and inject more noise at one instance to try to push the current estimate out of a local minimum, while keeping μ very low at other times.

Modified envelopes

The discussion about the envelopes implies that $\overline{\mathbf{c}}$ is computed from a signal $\mathbf{x}$, thus there exists some set of carriers that when combined with the envelopes yield the original channel signals $\mathbf{c}$. However, no such assumption is made in the above analysis of the iterative estimation algorithm, so the algorithm should work just as well if given a set of *modified* envelopes $\overline{\mathbf{c}} = |\mathbf{c}'|$. The main problem of modified envelopes is simply that no perfect solution can be guaranteed to exist: there might not be any signal estimate $\hat{\mathbf{x}}$ that can be generated for which $D_M(\overline{\mathbf{c}}, \overline{\mathbf{G}\hat{\mathbf{x}}})$ is within the frame error bounds.

Terminating condition

In the case of modified subband envelopes the minimal possible error is difficult to predict. While in general a simple threshold on D_M is a reasonable solution, we can also consider a method to terminate the iterative loop by measuring the change in the error $D_M(\overline{\mathbf{c}}, \overline{\mathbf{G}\hat{\mathbf{x}}})$, the relative change in error from one iteration to the next. However, this must be coupled to the simulated annealing factor μ, since addition of the perturbing noise may change the difference in D_M for one iteration from negative to positive (that is, adding noise will increase the error). To differentiate between a local minimum and the final result, an actual implementation of the estimation algorithm should observe the change in D_M for multiple iterations. If after a noise injection the gradient still indicates shallow convergence, the algorithm should terminate.

3.4 Illustrative example

In the following chapter, we will apply the methods described here to analyze a perceptual model based on a complex gammatone FB. However, to illustrate the concepts presented in this chapter, it is useful to show some simple FBs without being constrained to the properties of the human auditory system. With these FBs, we can show the issues of estimation from envelopes with tight-frame and snug-frame FBs. The reconstruction algorithm is trimmed down to a very simple loop with fixed noise injection to avoid division by zero.

The example FBs we use have eight channels ($M = 8$) and are based on the windowed STFT. Each channel analysis filter is of length $L_g = 8$ with impulse response

$$g_m[n] = 0.2w(n)e^{-2\pi j(m-1)n/M}, \tag{3.64}$$

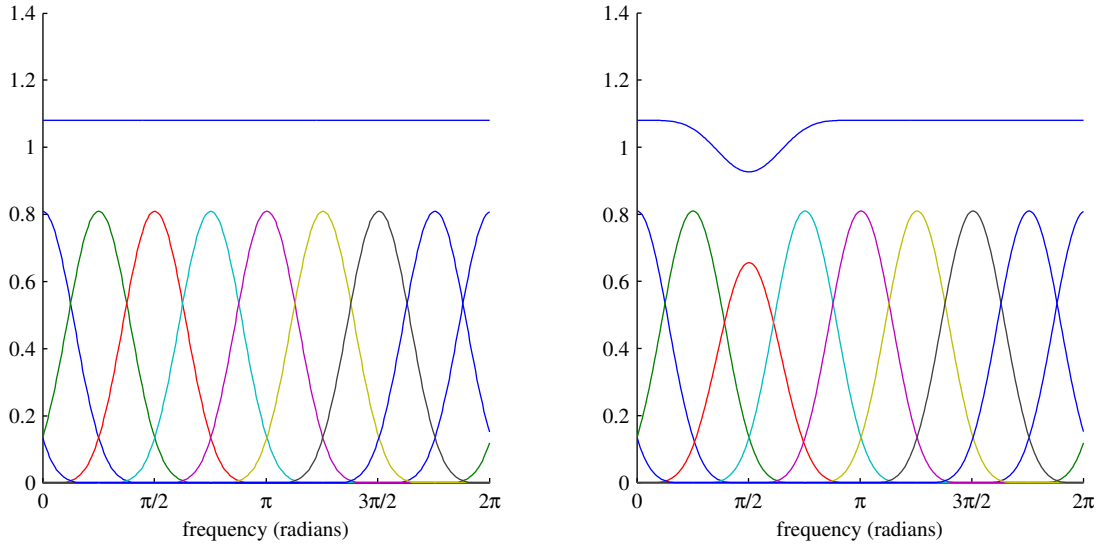
with $m = 1, \ldots, M$ and $n = 0, \ldots, (L_g - 1)$. The window $w(n)$ is the Hanning window of length 8. The scaling factor of the impulse response is chosen to make the plots more clear. From these prototype vectors, we generate circulant matrices and combine them into the first FB labeled $\mathbf{G}_T$. The second FB is derived from $\mathbf{G}_T$ by scaling the filter of index $m = 3$ by a factor of 0.9 and this FB is labeled $\mathbf{G}_S$. We can now show that the frame defined by $\mathbf{G}_T$ is *tight* and the frame defined by $\mathbf{G}_S$ is *snug*.

The frequency-domain responses (power spectra) of the FBs are shown in Fig. 3.3, where the top line in each graph shows the sum of the spectra for all filters. This is equivalent to the frequency response of the matched-filter FB system with no scaling of the synthesis FB, as shown by Eq. (3.20) with $\mathbf{H} = \mathbf{G}^T$.

	A	B	B/A	γ_{opt}	$\left(\frac{B-A}{B+A}\right)^2$
$\mathbf{G}_T$	1.08	1.08	1	0.9259	3.054e-30
$\mathbf{G}_S$	0.9261	1.08	1.1662	0.9970	0.005885

Table 3.1: Table of frame parameters for $\mathbf{G}_T$ and $\mathbf{G}_S$ with $N = 256$.

The circulant matrices were generated with $N = 256$ and the resulting parameters calculated from $\mathbf{G}^H\mathbf{G}$ are shown in Table 3.1. The factor $\gamma_{opt} = 2/(B + A)$ is used to implement the FB system minimizing the error bound, as in Eq. (3.36), with the error



(a) Tight-frame FB

(b) Snug-frame FB

Fig. 3.3: Filterbanks with tight and snug frame bounds

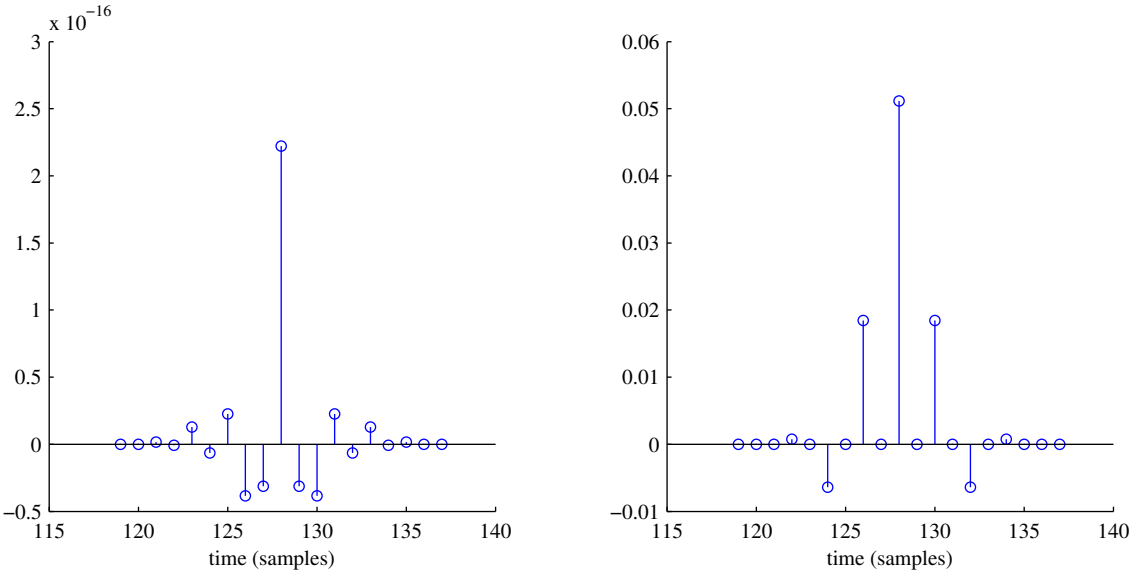
bound shown in the rightmost column. Note that due to finite numerical precision, $\mathbf{G}_T$ is not *exactly* tight.

We illustrate the reconstruction error by creating a unit impulse signal $\mathbf{x}$ of size N with the unit sample at $n = 128$. Figure 3.4 shows the resulting error (real part) $\mathbf{e} = \gamma \mathbf{G}^H \mathbf{G}\mathbf{x} - \mathbf{x}$. For the tight frame FB, the error $\|\mathbf{e}\|^2$ is less than $7 \cdot 10^{-32}$ and for the snug FB, less than 0.005, both within the predicted bounds.

3.4.1 Example estimation from envelopes

We can now illustrate the effect of the tightness of the frame on the estimation of a signal from its subband envelopes. The test signal used is a speech signal sampled at 8 kHz, the first part of the word 'twisted' spoken by a female speaker. The fragment is about 0.22 seconds long and was padded with zeros at the beginning and end to set $N = 2048$ as the base size of the vectors and matrices.

The signal and its envelope representation obtained by filtering with the tight frame FB are shown in Fig. 3.5. Clearly visible in the envelope representation is the broadband onset which is equally strong in all subchannels, followed by the voiced



(a) Tight-frame FB

(b) Snug-frame FB

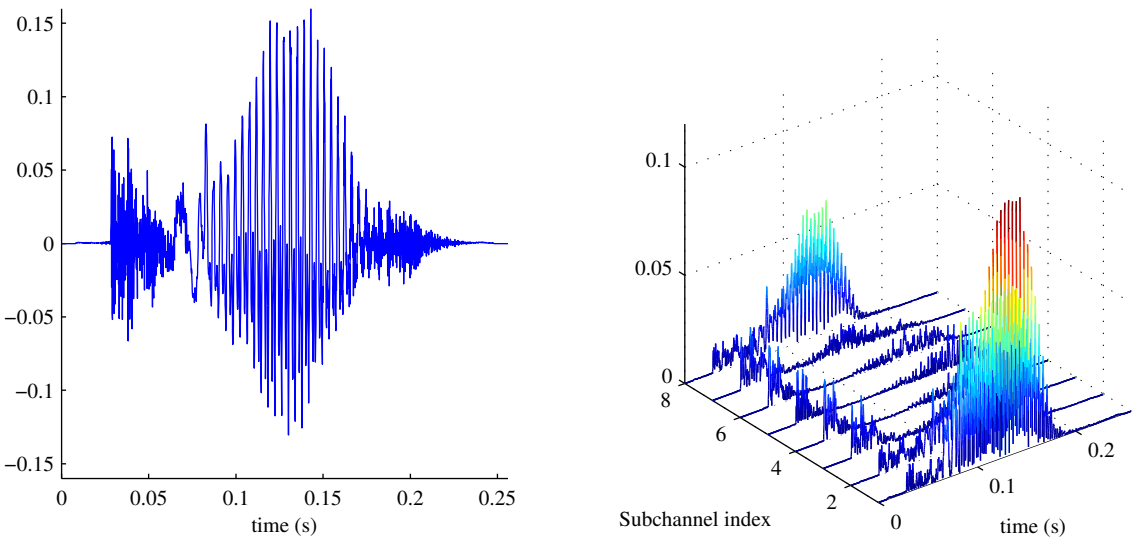
Fig. 3.4: Reconstruction error, impulse at $n = 128$.

section '-wi-' which is concentrated in the lowest frequency subchannels[7] and finally the fricative 's' is dominant in the high frequency subchannels at the end of the signal. These envelopes were then used to estimate the original signal, using the tight frame FB, the snug frame FB, and the snug frame 'optimal' (non-LTI filter) synthesis operator from Eq. (3.42). Note that the envelopes for the snug frame FB are identical to the tight frame case with the 3rd subchannel envelope scaled by 0.9. For the purpose of illustrating the evolution of D_M, the algorithm is implemented in its basic form, running for a fixed number of iterations with constant noise injection.

To facilitate comparison of convergent behaviour with different signals, the Signal to Error Ratio from [Griffin et al., 1984] is used, defined for the MSTFTM as

$$\mathrm{SER}[x(n), |Y_W(mS,\omega)|] = 10\log_{10}\left(\frac{\sum_{m=-\infty}^{\infty}\frac{1}{2\pi}\int_{\omega=-\pi}^{\pi}|Y_W(mS,\omega)|^2 d\omega}{\sum_{m=-\infty}^{\infty}\frac{1}{2\pi}\int_{\omega=-\pi}^{\pi}[|X_W(mS,\omega)|-|Y_W(mS,\omega)|]^2 d\omega}\right). \quad (3.65)$$

[7]Note that in this complex frequency representation, the high frequency bands are channels 4 and 5.



(a) Time-domain signal

(b) Envelopes

Fig. 3.5: Time-domain signal and subchannel envelopes of the word fragment 'twis-(ted)' spoken by a female speaker, sampled at $f_s = 8000$ Hz.

The denominator of this equation is simply $D_M[x(n), |Y_W(mS, \omega)|]$, and the numerator is constant for a given $Y_W(mS, \omega)$, giving a scaled inverse of the error function. This function has the advantage of being independent of signal energy. For our FB using circulant matrices, we use the equivalent

$$\mathrm{SER}(\overline{\mathbf{c}}, \overline{\mathbf{G}\hat{\mathbf{x}}}) = 10 \log_{10} \left(\frac{\|\mathbf{c}\|^2}{\| \, |\mathbf{G}\hat{\mathbf{x}}| - |\mathbf{c}| \, \|^2} \right). \tag{3.66}$$

The algorithm is started with the subchannel signals initialized to the envelopes with random phase. In comparison with zero-phase initialization, this was found to be useful to ensure rapid convergence at the beginning of the iterative procedure, since otherwise the initial guess would be a signal that is mostly DC and thus is strongly attenuated by the (bandpass) synthesis filters.

In Fig. 3.6, we show the increase of $\mathrm{SER}(\overline{\mathbf{c}}, \overline{\mathbf{G}\hat{\mathbf{x}}})$ over 1000 iterations for the set of envelopes that represent the speech signal. To generate the graph, the noise added at each iteration was scaled by a constant $\mu = 10^{-9}$, which is small enough to show monotonic decrease in the error. What can be seen clearly is that the algorithm

Algorithm 1 Iterative Reconstruction of $\hat{\mathbf{c}}$ from $\overline{\mathbf{c}}$

$i \leftarrow 0$;
$\hat{\mathbf{c}} \leftarrow \mathbf{r}$;
$\hat{\mathbf{c}}_{[k]} \leftarrow \overline{\mathbf{c}}_{[k]} \frac{\hat{\mathbf{c}}_{[k]}}{|\hat{\mathbf{c}}_{[k]}|} \qquad k = 1, \ldots, (MN)$;
repeat
  $\hat{\mathbf{x}} \leftarrow \mathbf{H}\hat{\mathbf{c}}$;
  $\hat{\mathbf{c}} \leftarrow \mathbf{G}\hat{\mathbf{x}} + \mu\mathbf{r}$
  $\hat{\mathbf{c}}_{[k]} \leftarrow \overline{\mathbf{c}}_{[k]} \frac{\hat{\mathbf{c}}_{[k]}}{|\hat{\mathbf{c}}_{[k]}|} \qquad k = 1, \ldots, (MN)$;
  $i \leftarrow i + 1$;
until $i = L$.

converges to a small error (high SER) very quickly, then settles to a steady state. The iterative algorithm using a snug frame (dashed line) converges to an error value of about 0.02 (SER = 21.8, with $\|\mathbf{c}\|^2 = 2.92$), which is close to the value that can be expected due to the frame bound, $(\frac{B-A}{B+A})^2\|\overline{\mathbf{c}}\|^2 = 0.0172$. For the tight frame (solid line) the SER after 1000 iterations is far higher. In theory, the limit would be about 10^{-15} due to the added noise but it would take an infinite number of iterations to reach that value.

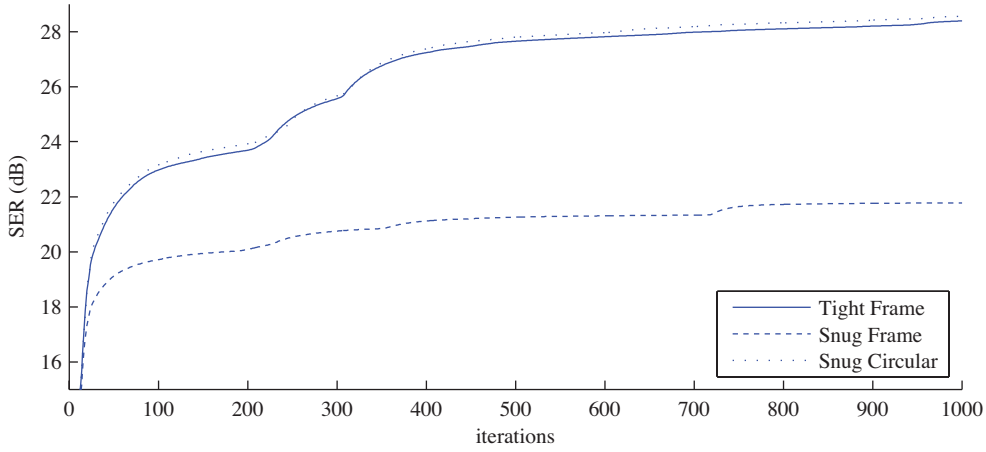


Fig. 3.6: Error measure over 1000 iterations

The dotted line represents the estimation of the signal using $\mathbf{G}_S$ as analysis operator and $(\mathbf{G}_S^H\mathbf{G}_S)^{-1}\mathbf{G}_S^H$ as synthesis filter, ignoring for the moment the circular convolution effect. It can be seen that overall the behaviour is very similar to the

tight frame case, showing the importance of ensuring tight frame bounds.

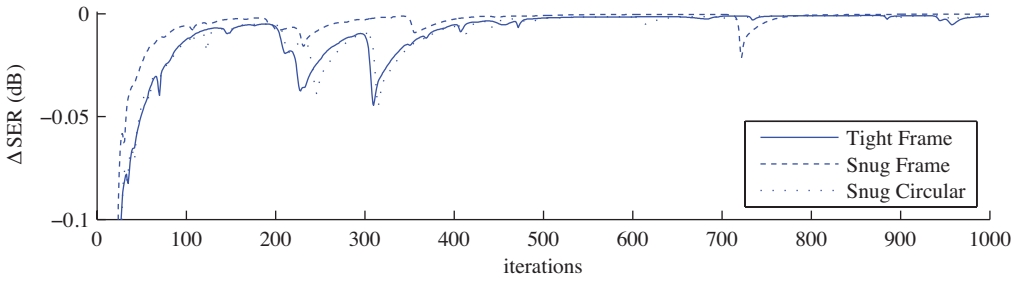


Fig. 3.7: Difference of error measure between iterations

The effect of local minima[8] is visible as a vague stair-like pattern in Fig. 3.6, but clearer in Fig. 3.7. The improvement in the error becomes lower with each iteration, but occasionally the estimate breaks out of the current local minimum and improves more quickly again for a short stretch.

Finally, we compare the original signal and its estimate (using the tight frame FB). The top graph of Fig. 3.8 shows the beginning of the signal in solid, with the estimate as a dashed line, while the lower graph shows the corresponding original envelope (channel 1) and the error $\overline{\mathbf{c}} - \overline{\mathbf{G}\hat{\mathbf{x}}^{(i)}}$ at the final iteration. The beginning part of the signal was reconstructed exactly, whereas the second part generally has reverse polarity to the original. The envelopes match well for long stretches, but where the energy (and thus envelope) is low, it is possible that a phase inversion occurs. The envelope of the estimate cannot match the original well at the instance when the phase "flips".

This is a significant property of the algorithm and can contribute to audible artifacts for pitched sounds, as will be discussed later. Envelopes are made to match within connected regions, that is, the signal has to be consistent with parts of itself that overlap in either time or frequency within the bounds of the filter impulse response. In this example, the impulse response is very short (15 samples for the combination of synthesis and analysis filters) and the filters overlap in frequency domain

[8]Note that there are not true local minima since the algorithm continues to converge to the global minimum even without noise perturbation. However, the error improvement in these sections is very small from one iteration to the next.

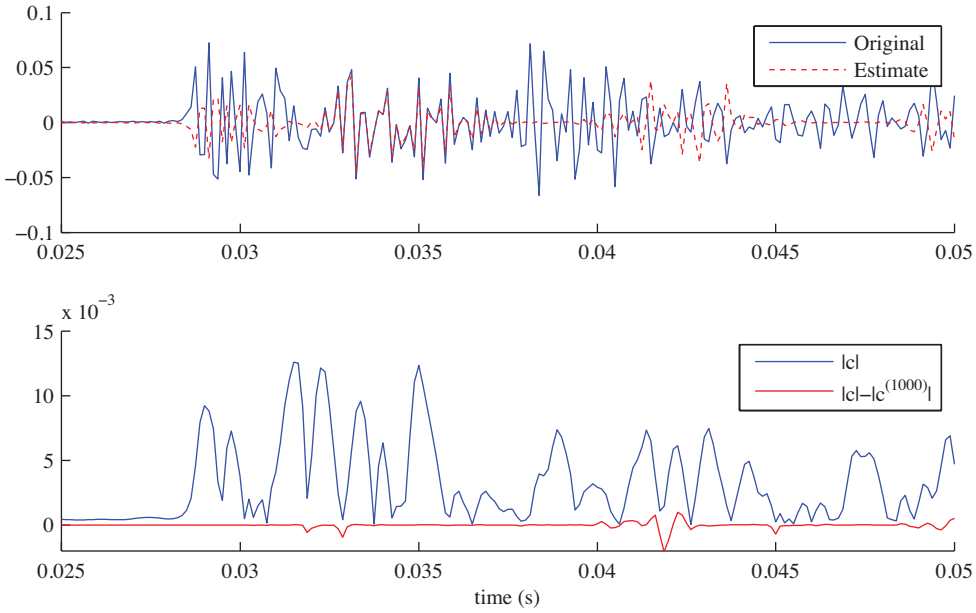


Fig. 3.8: Original signal and resulting estimate (top), channel 1 envelope of original and estimate error (bottom)

only with their immediate neighbours. In the following chapter, we describe other types of filters (specifically auditory-system based gammatone filters) that are more spread out, both in time domain (longer impulse response) and frequency (covering a larger region of the spectrum).

Given the simple FB in this section, even though an estimate was constructed whose subband envelopes matched the original signal very closely, the resulting signal subjectively is very different from the original. The estimate has a very rough characteristic, having lost its voiced pitch to a type of incoherent buzz. This loss of quality can be attributed to the fact that frequency information is lost in discarding the carrier information, and the envelopes do not match the characteristics of envelopes in the human auditory system.

3.5 Summary

This chapter presented a novel framework and tools to analyze an algorithm that finds the estimate of a signal based on the envelopes of a subband decomposition. Repre-

senting filters using circulant matrices and FBs as concatenated circulant matrices, linear convolution can be described in a compact manner if the filters and signals are designed such that circular convolution is avoided.

The circulant matrix representation of FBs can also be used to describe the FBs in terms of frame theory. The analysis and synthesis FB matrices can be viewed as analysis and synthesis operators for discrete finite frames, allowing for direct numerical computation of frame bounds and thus the accuracy with which the synthesis FB can reconstruct the original signal after decomposition by the analysis FB. Frame theory methods to construct a synthesis operator given an analysis operator can be applied directly to the FB representation. We show that the well-known matched FB structure can be regarded as the frame inverse if the corresponding frame is tight.

The concept of splitting a signal into an envelope and a carrier component is introduced, in the specific form of an analytic (complex) signal being split into the real, strictly positive amplitude and a carrier which is the unit-magnitude phase component of the signal. We then introduce an algorithm that attempts to compute an estimate of a signal given a set of subband envelopes. This algorithm is based on earlier work that was aimed at estimation from short-time Fourier transform magnitudes. The new algorithm is analyzed for convergence using the global convergence theorem. We conclude that, for the algorithm implemented using FB, convergence can be expected in general but cannot be guaranteed.

We conclude with a brief example showing the algorithm implemented with an 8-channel FB. One implementation uses a FB designed to be a tight frame, the other a snug frame. These implementations show the basic behaviour, in particular the problem where the algorithm spends a considerable number of iterations at a local minimum. In the next chapter, we will introduce perceptual envelopes from a gammatone FB and a signal estimation algorithm from those perceptual envelopes based on the algorithm presented above.

Chapter 4

Perceptual Representation and Iterative Reconstruction

In this chapter we present the central theme of this thesis, describing the perceptual representation and the iterative method to reconstruct an audio signal from that representation. We begin by outlining the issue of perceptual representations and iterative reconstruction from a general standpoint, showing the parts needed to implement such a coding system. The parts can be roughly summarized as a psychoacoustic model to convert the original audio signal into a set of perceptual parameters, a method of calculating a single-variable error between the original perceptual parameters and the set of parameters of the reconstructed audio, and a function to modify the reconstructed signal such that the error between the two sets of parameters is reduced. The iterative approach to reconstruction allows for a new approach to designing perceptual representations for audio coding.

A representation of audio signals by sparsely sampled auditory envelopes designed for iterative reconstruction is presented. This representation is based on perceptual coding methods that were briefly described in Chapter 2. We describe the auditory pulse representation that was used as a starting point in more detail in this chapter. Our new representation encodes a monaural signal by sampling the auditory envelopes. The auditory envelopes are computed by a gammatone filter subband decomposition followed by a magnitude envelope extraction and low-pass filtering. A masking model evaluates the auditory envelope samples to discard samples deemed

perceptually irrelevant, resulting in a sparse auditory envelope representation (SAER).

The envelope representation lacks the instantaneous phase information in each auditory channel that is needed for reconstruction by linear filtering. We call this instantaneous phase signal the subband carrier since it can be regarded as a frequency-modulated sinusoid whose amplitude is modulated by the envelope signal. From a coding point of view, it is desirable to discard this carrier information since it is very difficult to encode, yet it can be reconstructed based on the envelopes. This suggests the synthesis-by-analysis approach to reconstruction and we describe an iterative method for model inversion. These methods have been proposed in the context of auditory models before [Slaney et al., 1994; Heming et al., 2003; Chi et al., 2005]; we show that such methods can be modified to incorporate a sparse representation similar to pulse coding [Feldbauer et al., 2005] that is suitable for coding. This will demonstrate the feasibility of iterative reconstruction for decoding the parameters from a perceptual coder.

In particular, we examine the reconstruction of the carrier information which must be done in a way such that the subband channel signals are consistent with each other. There is considerable overlap in frequency between auditory channels and a wrong estimate of the carrier will cause interference, resulting in audible artifacts which can be detected by comparing the envelopes of the reconstructed signal to the original auditory envelopes. The detection of artifacts is used to adjust the carrier information for the next iteration of the synthesis-by-analysis loop, eventually finding a set of carrier signals matching the auditory envelopes. However, modification of the original auditory envelopes due to sparsification and quantization adversely affects the ability to reconstruct an accurate carrier signal, which we examine in the implementation described in Chapter 5.

A major criticism that can be made against such iterative reconstruction methods is the high computational cost. The availability of ever faster and power-efficient processing capabilities weakens this argument somewhat, but is is still important to limit the computational requirements. In iterative methods, the computational cost of a single pass of the reconstruction algorithm is multiplied by the number of passes. To mitigate this, we include methods to lower the computational load within the loop and a mechanism to terminate the loop once reconstruction quality is satisfactory.

4.1 Perceptual representations of audio signals: general concepts

A perceptual model is an algorithm that converts a sound signal into a set of model parameters that are a perceptual representation of the sound. To explain the idea of the reconstruction of a new audio signal from that perceptual representation, we must first consider what reconstruction is acceptable and whether or not the model parameters have been modified (for example by quantization for transmission).

Suppose we have two audio signals x_1 and x_2 in a space $\mathcal{S}$ and we have a perceptual model giving us sets of parameters, $F_{x_1} = \mathcal{P}(x_1)$ and $F_{x_2} = \mathcal{P}(x_2)$ inside a parameter space $F_{x_1}, F_{x_2} \in \mathcal{A}$. Since an audio signal may be modified in such a way that changes are inaudible, $F_{x_1} = F_{x_2}$ does *not* imply $x_1 = x_2$, that is, the function $\mathcal{P}(\cdot)$ is many-to-one. More generally, we assume that there is an *audible difference* function $D_{\mathcal{P}}$ and a threshold $\tau_{\mathcal{M}}$ such that

$$D_{\mathcal{P}}(F_{x_1}, F_{x_2}) \leq \tau_{\mathcal{M}} \tag{4.1}$$

means that x_1 and x_2 are indistinguishable from each other by an average human listener. For any individual parameter of a signal, this is usually called the "just noticeable difference" (JND).

We call the set of "signals sounding like x", $\mathcal{S}_x$. $\mathcal{S}_x$ can be a disconnected subspace of $\mathcal{S}$. For example, most people cannot tell if a sound is played back through a speaker with reversed polarity (for some nonzero x, both x and $-x$ are in $\mathcal{S}_x$). Thus, large disconnected regions in $\mathcal{S}$ can project to a small region in $\mathcal{A}$.

Inversion of a perceptual model $\mathcal{P}$ means that we are trying to estimate a signal $\hat{x}$ from a set of perceptual parameters F'_x. It it typically assumed that F'_x is derived in some way from a set of features F_x that are calculated from an actual signal x, and thus F'_x can be regarded as a perceptual representation of the signal x plus some error, ϵ_Q (typically introduced due to quantization and coding[1]), with

$$D_{\mathcal{P}}(F_x, F'_x) = \epsilon_Q. \tag{4.2}$$

[1]It is possible to generate entirely synthetic perceptual parameters but here we assume parameters are derived from a real signal.

Perceptual models, as described in Chapter 2, are generally nonlinear complex algorithms. They are typically also redundant to some degree or involve sampling of a redundant representation. Any set of perceptual features not derived directly from a signal in $\mathcal{S}$ may not have an exactly matching signal in $\mathcal{S}$, that is, for some F'_x it is *impossible* to find a signal $\hat{x} \in \mathcal{S}$ such that $\mathcal{P}(\hat{x}) = F'_x$. Essentially, the perceptual features of a modified representation might not be self-consistent. The best a model inversion algorithm can do in this instance is to find the estimate $\hat{x} \in \mathcal{S}$ that minimizes the difference function $D_{\mathcal{P}}(F'_x, F_{\hat{x}})$, which is the reconstruction error. In other words, the task of the decoder is to find an audio signal whose perceptual parameters best match the received information. Since a decoder has no knowledge of the original x or F_x before quantization, even the optimal reconstruction of F'_x might be outside of $\mathcal{S}_x$, that is the difference between x and $\hat{x}$ might be audible even if all parameters were quantized below their individual JND, or collectively $\epsilon_Q \leq \tau_{\mathcal{M}}$. This issue must be taken into account when designing the encoder, in particular if the perceptual parameters are quantized.

4.1.1 Reconstruction by iterative estimation

A key concept we introduce in this chapter is the reconstruction of audio from coded perceptual parameters using an iterative method. To find an estimated signal $\hat{x}$ from the set of features F'_x, the algorithm begins with a guess of what the signal $\hat{x}$ might be and then determines how good the guess is in a perceptual sense. This estimate, or rather its perceptual analysis, is compared to the received parameters and processed to generate a new estimate. This process is repeated until the perceptual analysis of the estimate is considered sufficiently similar to the received parameters.

Generating the new estimate is a critical step in this process. We denote the sequence of signals as $\hat{x}^{(i)}$, with $\hat{x}^{(0)}$ being the initial guess and label the refinement function $\mathcal{R}$. This function at each iteration computes

$$\hat{x}^{(i+1)} = \mathcal{R}(F'_x, \hat{x}^{(i)}), \tag{4.3}$$

that is, given the current estimate $\hat{x}^{(i)}$ and the target parameters F'_x, a new estimate $\hat{x}^{(i+1)}$ is generated. This function should be designed such that applying it results in

an improvement in perceptual quality relative to the given perceptual features F'_x, so

$$D_{\mathcal{P}}(F'_x, F_{\hat{x}^{(i+1)}}) \leq D_{\mathcal{P}}(F'_x, F_{\hat{x}^{(i)}}). \tag{4.4}$$

We note that in this iterative method of reconstruction every estimate $F_{\hat{x}^{(i)}}$ is computed from a signal in $\mathcal{S}$. Thus, even though F'_x might not be realizable in $\mathcal{S}$, if Eq. (4.4) is satisfied, as $i \to \infty$ a reconstruction given F'_x should be found that is best in terms of the perceptual model.

In terms of perceptual models, this allows us to change the approach to designing a perceptual representation intended for coding applications. Rather than designing a representation for ease of direct inversion (which typically necessitates coding information that is perceptually irrelevant), we make our representation easy to refine given an initial estimate of the signal (which does not need to be transmitted). In addition, this approach allows for testing perceptual representations for completeness by subjective validation. If the representation is *incomplete* in terms of encoding all perceptually relevant information and the iterative algorithm converges to a state where the error $D_{\mathcal{P}}(F'_x, F_{\hat{x}^{(i)}})$ is below the audible difference threshold $\tau_{\mathcal{M}}$ as estimated by the model, then human listeners should be able to detect a difference between the original signal and the reconstruction. Since presumably inaudible information was discarded at the encoder, the decoder *must* generate the missing information from just the transmitted data and this artificially generated information might be different from the discarded information.

It is difficult to predict what data the iterative reconstruction algorithm can interpolate from the information that is given, so one approach to develop a new auditory representation is to trim down an existing perceptual representation. To demonstrate this, we modify an existing perceptual coder that uses direct model inversion.

4.2 Computing an auditory representation

The perceptual representation we use in this thesis is a sparse sampling of auditory envelopes, where the auditory envelopes are obtained using a gammatone FB. This representation has its origin in a sparse pulse representation described by [Feldbauer, 2005], and to describe our representation (termed SAER), we first describe Feldbauer's

sparse pulse representation method and the algorithm to compute it. In particular, we focus on the pulse masking model, which sparsifies the pulse representation. We then describe the algorithm to compute the SAER, highlighting the differences and similarities to Feldbauer's method.

4.2.1 Feldbauer's sparse pulsed auditory representation

The full structure of the encoder in [Feldbauer, 2005] is shown in Fig. 2.5a. For the purpose of illustrating the sparse pulse auditory representation, we use a simplified version in which the amplitude compensation stages as well as the stream encoding sections are omitted since they are not part of the psychoacoustic model. This simplified structure is shown in Fig. 4.1.

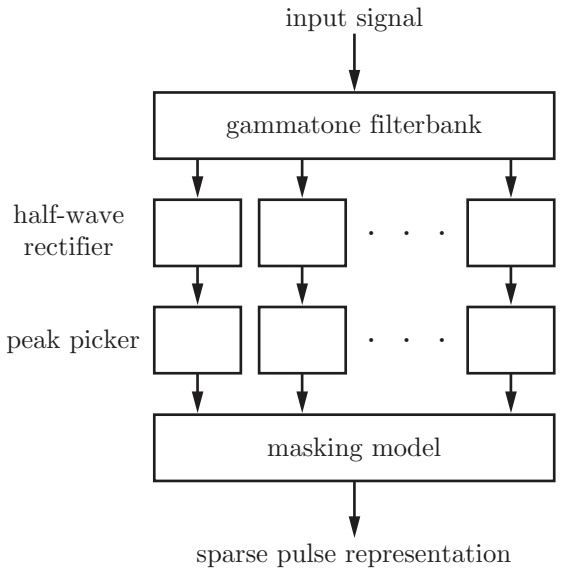


Fig. 4.1: Simplified version of the Feldbauer encoder

The filters of the analysis gammatone FB are described by the general expression Eq. (2.1). The specific choices of the number of channels, channel gains, and phase constants are dictated by the requirement of allowing for reconstruction with low distortion. For the implementation of our model, the details of the FB will be discussed in the next chapter. Key to the analysis FB is that each channel is an emulation of the BM frequency analysis for a local group of inner hair cells (IHC). The auditory

subchannel signals from the FB outputs have a narrowband bandpass characteristic with significant overlap in frequency domain, which allows the encoder to account for interactions of stimuli at different frequencies.

The transformation of the subband signals into a pulse-based representation of the original audio signal begins by half-wave rectification. Within each auditory channel, the half-wave rectifier block in Fig. 4.1 can be described by the transfer function

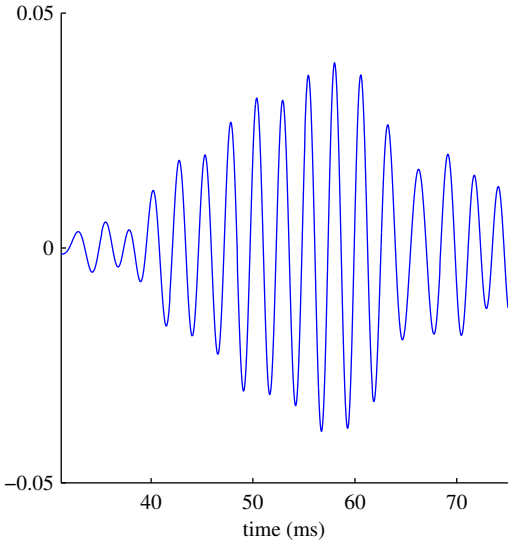
$$x_{\mathrm{Rout}}[n] = \max(x_{\mathrm{FBout}}[n], 0)^c, \tag{4.5}$$

where $x_{\mathrm{FBout}}[n]$ is the channel signal from the gammatone filterbank and $x_{\mathrm{Rout}}[n]$ is the output. The exponent $c = 0.4$ is used to perform power-law companding as part of the inner hair cell (IHC) model. The peak picking block that follows the half-wave rectification sparsifies its input by setting all samples of the input to zero except for local maxima using the function

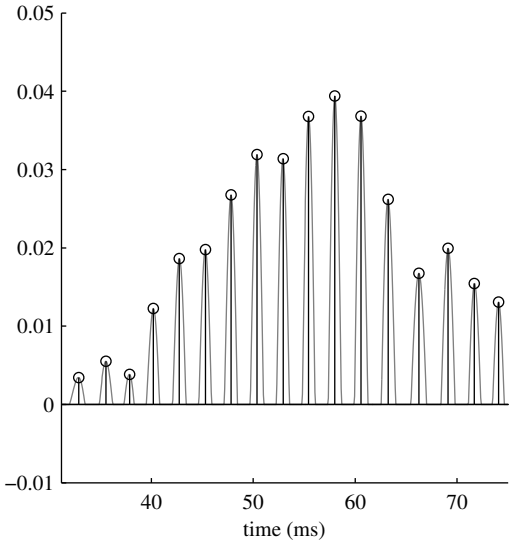
$$x_{\mathrm{Peaks}}[n] = \begin{cases} x_{\mathrm{Rout}}[n], & x_{\mathrm{Rout}}[n] > x_{\mathrm{Rout}}[n-1], \text{and} \\ & x_{\mathrm{Rout}}[n] > x_{\mathrm{Rout}}[n+1], \\ 0, & \text{otherwise.} \end{cases} \tag{4.6}$$

The resulting signal in each subchannel is a sparse stream of pulses synchronized with the peaks. To clarify this pulse selection, Fig. 4.2a shows a short section of a subchannel signal where the channel centre frequency is 374 Hz. The input signal is a male voice speaking the word "distance". The figure shows the signal section corresponding to the first vowel ("i"). Sub-figure 4.2b shows the peak-picked signal with the half-wave rectified signal as a gray line (we set $c = 1$ to show the match to the peak value). Note that the signal in Fig. 4.2a bears resemblance to an AM signal; however, the underlying carrier is not a monochromatic sinusoid but instead has a variable instantaneous frequency. The peaks in Fig. 4.2b, synchronized to this carrier wave, are therefore not spaced at regular intervals and their deviation from regular spacing turns out to contain important information for proper reconstruction.

As a whole, the representation of the signal at this stage is the set of pulses in all the subchannels. In [Kubin and Kleijn, 1999], 20 channels are used for audio signals sampled at 8 kHz and Feldbauer expands this to 50 channels in the system described



(a) Original Subchannel signal



(b) Half-wave rectified signal and peak picking

in [Feldbauer, 2005]. It should be apparent that although the individual subchannel pulse signals are sparse, the total number of pulses exceeds the original number of samples in the signal due to the large number of channels. The full representation of the short audio signal from above is shown in Fig. 4.2, with the pulses as vertical bars whose intensity indicates the amplitude. This short segment of speech is only 450 samples long at a sampling rate of 8 kHz, yet the pulse representation is 2246 pulses in 50 channels (the channel centre frequencies span from 26 Hz to 3329 Hz). Note that the rate of pulses increases with the channel number as can be expected, since the centre frequency is a function of the channel number. On average, the time interval between adjacent pulses within an auditory channel should be $1/f_c$, where f_c is the channel centre frequency.

In order to code audio signals efficiently using this pulse representation, the number of pulses in the auditory representation must be reduced significantly. This task is primarily performed by the masking model, which will be described in more detail below, in particular the transmultiplexer concept that implements both temporal and simultaneous masking properties. In his implementation, Feldbauer reports that the masking model reduces the number of pulses used to represent the audio signal such that the number of pulses after applying the masking model is less than the number

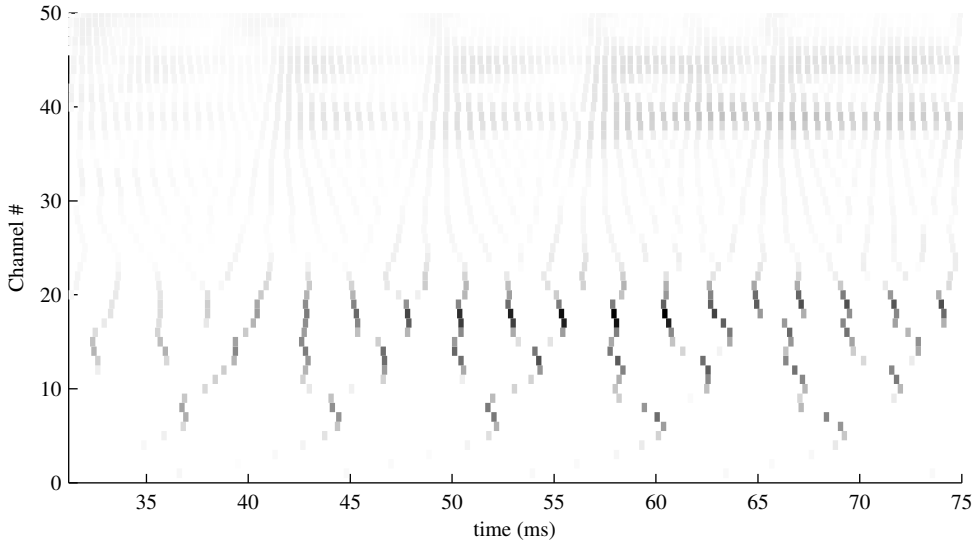


Fig. 4.2: Peak-picked representation of an audio signal showing multiple subband channels

of samples for a given time segment of audio.

The transmultiplexer based masking model

We now examine the masking model based on the concept of the transmultiplexer in more detail. The transmultiplexer originally provided the idea for the development of the iterative reconstruction algorithm that will be explained later in this chapter. In both cases, the key idea is that the ultimate *receiver* of the reconstructed audio signal is a human listener. We can therefore substitute an auditory model for this listener's hearing system to evaluate the (predicted) perception of the reconstructed output signal.

We can thus ask how much any given impulse that is part of a pulsed auditory image contributes to the auditory image generated in the auditory system of the listener. As shown in Fig. 4.3, we do this by first reconstructing the audio based on the sparse pulse representation then analyzing this reconstructed audio signal using a model of the auditory system. One can use the same model that was used to analyze the sound to be encoded. This becomes the transmultiplexer view of perceptual-

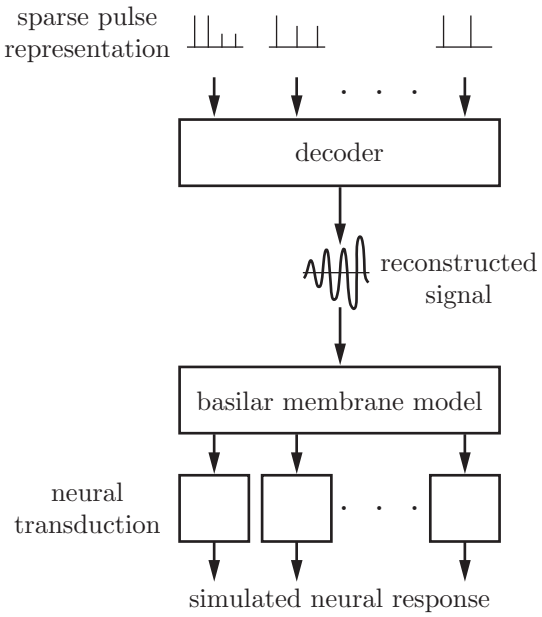
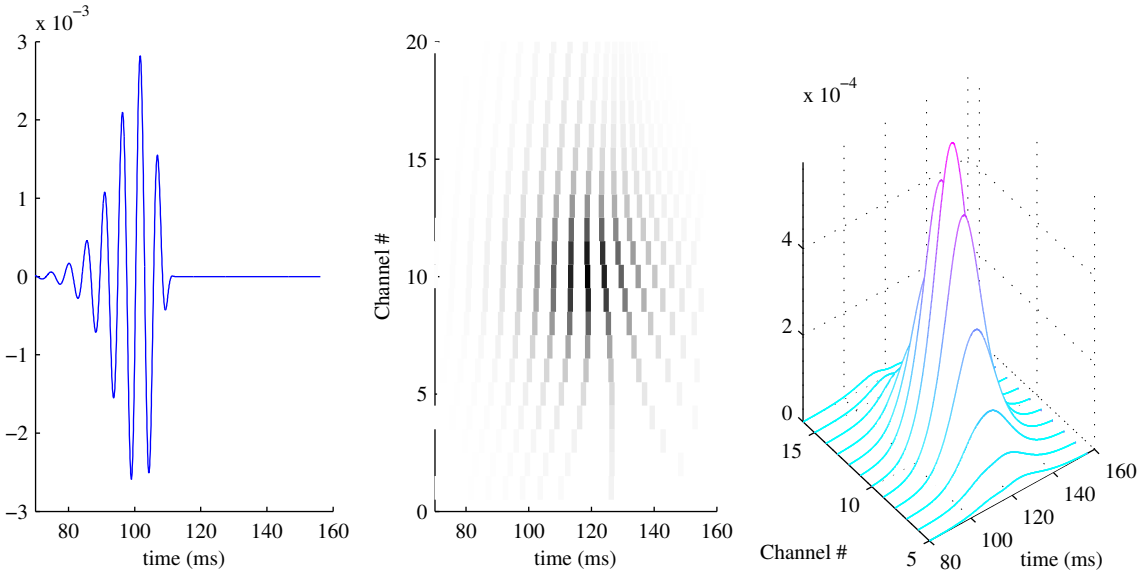


Fig. 4.3: Transmultiplexer view of sparse pulse coding

domain coding: the multichannel signal (the sparse pulse auditory representation) is transmitted via a single-channel carrier (the reconstructed audio signal) to a receiver that transforms it back into a multichannel pulsed representation (the internal neural pattern in the human listener). Compared to the usual view of a perceptual domain coder, the encoder and decoder are inverted.

How can this construct be used to sparsify a pulsed representation? Ideally, given the pulse representation one could perform an exhaustive search, recomputing the auditory representation for any subset of pulses from the original representation. Clearly, with thousands of pulses for even small segments of audio, such an exhaustive search is not feasible.

Instead, Feldbauer considers the effect that any given pulse has on its immediate time-frequency neighbourhood. Take a single isolated pulse in a pulsed auditory representation: the synthesis filterbank (shown in Fig. 2.5b) will turn this single pulse into a reverse-time gammatone impulse with frequency and length dictated by the channel wherein this impulse is located. This reverse-time gammatone is the resulting audio signal that is passed through the analysis filterbank in the transmultiplexer



(a) Synthesized audio from isolated pulse

(b) Pulse pattern of synthesized audio

(c) Envelope pattern of synthesized audio

Fig. 4.4: Reverse time pulse (Channel 10, centre frequency of 184.7 Hz), its pulse pattern after transmultiplexing, and the associated envelope pattern

setup. Due to the overlap of the gammatone filters in frequency, the (narrowband) reverse-time gammatone impulse will evoke a pulse pattern that spreads over many channels adjacent to the original pulse channel. Also, the pulse will be spread in time by the filter responses, resulting in a spread pattern such as that shown in Fig. 4.4. Figure 4.4a shows the synthesized audio from a pulse in channel 10 (centre frequency of 184.7 Hz) and Fig. 4.4b shows the resulting pulse pattern after reanalysis.

We can see that the isolated pulse expands into a large set of pulses in multiple channels. If now the original pulse is part of a pulsed auditory pattern, nearby pulses whose amplitudes are less than that of the original pulse will not affect this pattern greatly, that is, the effect of the smaller pulses on the transmultiplexed pattern is masked by the dominant pulse. To determine which pulses are dominant and which smaller pulses are masked, the transmultiplexed *envelopes*[2] [Feldbauer and Kubin, 2004] are computed since the actual temporal positions of pulses within the spread

[2]Specifically, the Hilbert envelopes as defined by Eq. (3.45).

pattern are assumed to be irrelevant. In Fig. 4.4c, the envelopes of the pulse pattern shown in Fig. 4.4b are shown. The envelopes are basically a smooth curve connecting the peaks of the pulse pattern.

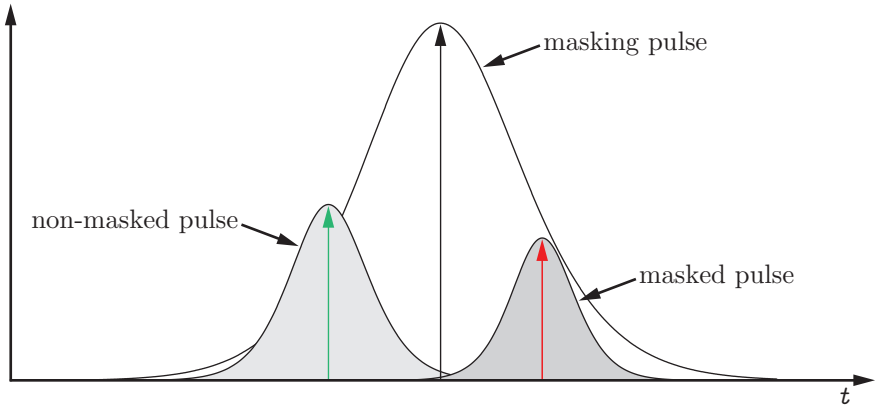


Fig. 4.5: Schematic representation of the masking decision

The implementation of this masking model operates on short sections of the pulse representation. Within this section, the pulses are sorted by amplitude and evaluated beginning with the largest amplitude pulse. This pulse is declared the *masking pulse* and the surrounding pulses are compared to the envelope that the masking pulse would generate, as shown in Fig. 4.5. Any pulse below the envelope is considered to be masked by the masking pulse and is removed from the pulse representation.

Of the remaining pulses, the next largest pulse is selected and the process is repeated until every pulse of the original full pulsed auditory representation has been evaluated as a masker or has been removed from the representation. Then, the amplitude of the remaining pulses is adjusted to account for the energy lost due to the removal of pulses and this sparse auditory representation is coded for transmission or storage.

Feldbauer introduces an additional empirical factor to the masking model, the *impact factor*, used to control the severity of the sparsification. This factor (r_I in the description of the implementation in the next chapter) is used to attenuate or amplify the transmultiplexed pulse, allowing the pulse masking effect to be either under- or overestimated. It therefore allows a tradeoff between the number of pulses in the final

representation and the quality of the audio signal at the decoder. The impact factor is also used in the implementation of the transmultiplexer masking model for envelope samples as described below.

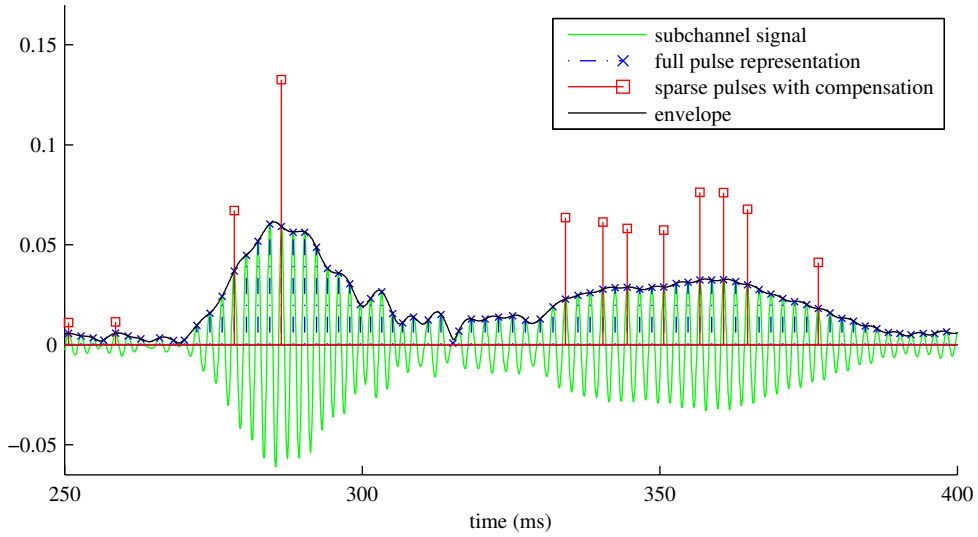


Fig. 4.6: Subchannel signal and pulse based representations

Fig. 4.6 shows the subband signal of a speech sample, the peak-picked pulse representation, and the sparse representation with amplitude compensation. This figure also shows the Hilbert envelope of the subband signal and that the peak-picked pulse representation effectively samples the envelope at a rate determined by the subband carrier. Note that this single-channel view does not show the effects of simultaneous masking (masking due to pulses in adjacent channels) that explains the lack of retained pulses in the region of 290–320 ms.

4.2.2 Auditory envelope representation

The encoder described by Feldbauer generates a good sparse auditory representation that can be used to reconstruct an audio signal with high fidelity at low computational cost. However, this representation focuses on the amplitude of the pulses, assuming that the pulse positions are encoded with little or no error. Especially for the higher frequency channels, a small error in the pulse position can cause interference with

the subchannel signal in an adjacent band, resulting in audible distortion of the reconstructed signal. To avoid this interference, the encoder is required to store or transmit the precise timing information for the pulses. This is exacerbated by the fact that the higher frequency channels also use more pulses per unit time than the lower frequency channels and this requirement for storing large amounts of precise timing information is in contrast to the human auditory system, where the hair cells have a limited temporal acuity [Moore, 2003].

We can address this problem by using iterative reconstruction as described in the previous section. Thus, the perceptual representation that we use is a modification of the pulse based representation in which the half-wave rectification and peak-picking sections are replaced with an envelope detector, specifically an operator that extracts the Hilbert envelope of the subband signal. The Hilbert envelope extractor is equivalent to the envelope follower used in the isolated pulse masking model described above, but is applied to the original gammatone filter output. The resulting subband envelopes (or auditory envelopes) have a generally low-pass characteristic and lack the precise timing information that is contained in the pulse positions of the peak-picked representation: this timing information will be reconstructed by the decoder using an iterative loop. This modified encoder is shown in Fig. 4.7. Like the pulse-based encoder, it begins with a gammatone filter analysis resulting in a redundant representation which is then sparsified by a masking model.

The decomposition of a signal into subband envelopes can be computed efficiently and, more importantly, we have shown in the previous chapter that it is possible to reconstruct a signal from just these envelopes. The relationship between the envelope and pulse representations is pointed out in Fig. 4.6 and it can be seen that the pulsed representation and the envelope are very similar but the pulsed representation retains the timing information of the signal. The figure also shows the sparsified pulse representation (with the amplitude compensation that allows for reconstruction of the signal with correct subband energy). Effectively, the sparse representation discards the timing information of the missing pulses but retains their amplitudes by adjusting the retained pulses. Still, the decoder needs the precise timing information of retained pulses for accurate reconstruction. This can be avoided with the envelope representation.

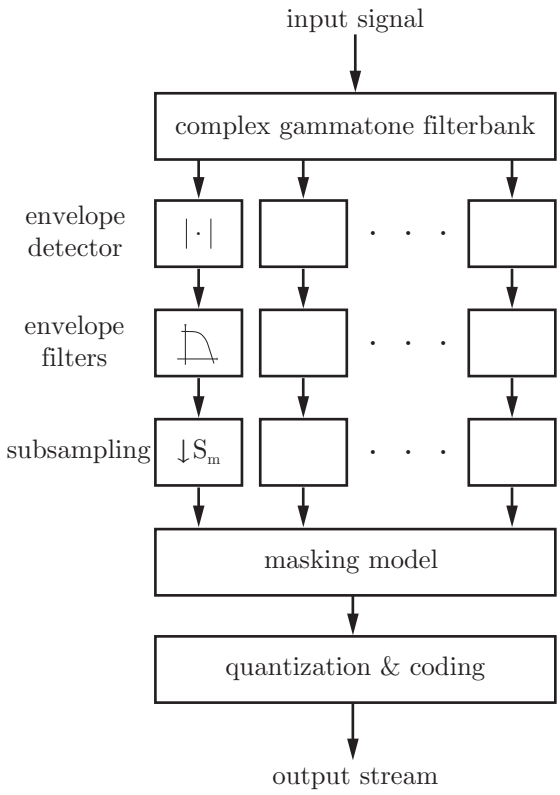


Fig. 4.7: Sparse auditory envelope encoder

Thus, the envelopes are a representation of the peaks of the subband signals without the detailed timing information. In contrast to the peak-picked representation, the envelopes can be evaluated based on their frequency characteristics in addition to the temporal properties on which the transmultiplexer masking model is based. The sparsification algorithm that we use exploits the frequency domain characteristics by subsampling the envelopes and the time domain characteristics by sparsification using the transmultiplexing model.

Sparse envelopes by sampling and salient feature detection

Up to now, we have implicitly assumed that the analysis filterbank and the envelope extraction operate at the sampling rate of the original signal. Thus, while the envelopes are a representation of the audio signal without the pulse timing information,

in terms of samples it is still a large amount of data and highly redundant. As a first step to improve the efficiency of the representation, the envelopes are subsampled. The peak-picking may be regarded as an adaptive sampling of the envelopes, but the rate of this adaptive sampling is not dependent on the audible information of the envelopes, instead it is tied to the instantaneous frequency of the carrier signals. The sparsification by the masking model addresses this to some degree but remains tied to the fine temporal structure rather than the perceptible characteristics of the envelopes.

We look at the perception of auditory subband envelopes, which has been studied in the past by many researchers. In particular, we note that the Hilbert envelope of auditory subbands can be compared to the low-pass filtered neural transduction model that has been used in perceptual models such as those described in Chapter 2. More specifically, the effects of filtering the Hilbert envelopes of subband signals have been investigated by Drullman [Drullman et al., 1994] and Ghitza [Ghitza, 2001], mainly in the context of speech perception. Ghitza finds that the frequency content of auditory envelopes is inaudible if higher than roughly half the bandwidth of the auditory filter. Thus we can argue that the subband envelopes can be more efficiently processed by subsampling at a rate equal to the reciprocal of the auditory channel bandwidth, preceded by a lowpass filter equal to half the channel bandwidth to avoid aliasing (see Fig. 4.7).

On the other hand, using the Hilbert envelope as a perceptual representation was criticized by Schimmel [Schimmel, 2007], pointing out three problems with Hilbert envelopes (termed incoherent envelopes in his analysis). The first problem is that the bandwidth of the magnitude envelope exceeds the bandwidth of the subband signal [Dugundji, 1958]. The second is that since the envelope is a real signal, its spectrum is conjugate symmetric about DC, while the subband signal spectrum is not symmetric about its centre frequency. The third problem is that the envelope representation is not closed under convolution, since it is strictly positive but its convolution by a filter response may become negative. In general, modulation domain filtering is not well defined [Li and Atlas, 2005]. Schimmel addresses these issues by using a coherent modulation representation rather than a strictly real and positive envelope. However, as with representations for audio coding, his problem analysis and

solution are motivated by allowing simple non-iterative reconstruction, in particular for modified representations[3].

In the iterative reconstruction, we find that since the operations performed on the envelopes do not need to be exactly invertible, we can avoid the issues raised by Schimmel. First, since it is assumed that the frequency content of the envelope exceeding half the auditory channel's bandwidth is inaudible, it is sufficient to reconstruct a signal where only the lower frequency spectra of the envelopes match. Second, the induced symmetry in the envelope spectrum is not enforced on the underlying subband signal during reconstruction and we allow the subband signal to have an asymmetric spectrum since the refinement algorithm only compares envelopes in time domain. Finally, the problem that the envelope representation is not closed under convolution is more problematic in the context of subsampling and quantization. However, since an envelope signal tends to have a very strong DC component, the antialias filter for sampling is unlikely to cause the subsampled envelope signal to become negative. If this situation occurs, it is likely to be below the audible threshold and half-wave rectification can be used.

Initial experiments with reconstruction from filtered and sampled envelopes showed promising results with little or no perceptible distortion for some classes of audio (in particular speech signals) [Thiemann and Kabal, 2007]. It was found that the sampled envelope representation is still a highly redundant representation, with 1.4 envelope samples required per monaural audio sample (using a signal sampled at 16 kHz and 62 subband filters with centre frequencies from 40 Hz to 6930 Hz). The model used in this thesis extended the FB to 65 channels (spaced 0.5 ERB apart) to increase the audio bandwidth and as result has a higher redundancy (see the following chapter), but we introduce a sparsification block to the model that reduces the number of envelope samples significantly.

Transmultiplexer based sparsification on subband envelopes

To sparsify the sampled envelope representation, we implement a transmultiplexer based masking model similar to the model described above. In fact, in the final

[3]The target application of Schimmel's method is noise reduction by modifying the perceptual representation of the noisy signal.

implementation, the envelope samples can be treated as pulses and expanded to envelopes of pulses for comparison to the adjacent envelope samples. However, since the transmultiplexer is based on modeling the reconstruction from pulses, the "library" of envelopes must be modified to emulate the reconstruction from a sparse envelope sample rather than an auditory pulse. For a given channel, the resulting transmultiplexed pattern in our implementation differs from the auditory pulse pattern simply by a gain factor. The details are described in the following chapter.

We observe that since the envelope samples are generally spaced further apart than the pulses in the auditory pulse based representation, the temporal masking effect of the transmultiplexer model is greatly reduced even with the stretched patterns. However, the transmultiplexer model still removes envelope samples due to simultaneous masking (in frequency domain), primarily in higher frequency bands.

The masking model is also used to handle sparsification of the samples due to the threshold in quiet. Given an assumption of the absolute volume of the signal being encoded, the threshold of hearing for the envelope in each channel can be calculated. Envelope values below this threshold are not encoded.

Quantization and coding

The final step shown in Fig. 4.7 is the quantization and encoding of the SAER into a bitstream. From the perspective of perceptual coding, the important effect of this step is the addition of noise to the sample amplitudes due to quantization and we assume that once quantized the auditory envelope samples are received at the decoder without error. However, the quantization is a modification of the envelopes in addition to the sparsification and for this reason the next chapter will present the results of applying a simple scalar quantizer to the SAER.

In Fig. 4.8, the subband signals of Fig. 4.6 are shown with the smoothed subband envelopes and the sampling thereof. We also show the adjacent channel to illustrate the effects of simultaneous masking: the peak at 280 ms in channel 20 is not coded by AE samples since the samples in channel 21 are dominant. As in Fig. 4.6, simultaneous masking from adjacent channels removes a significant number of envelope samples. In the following section, this effect is made more visible by the maximum envelope limit calculation. Both the sparse pulse and sparse envelope representations encode

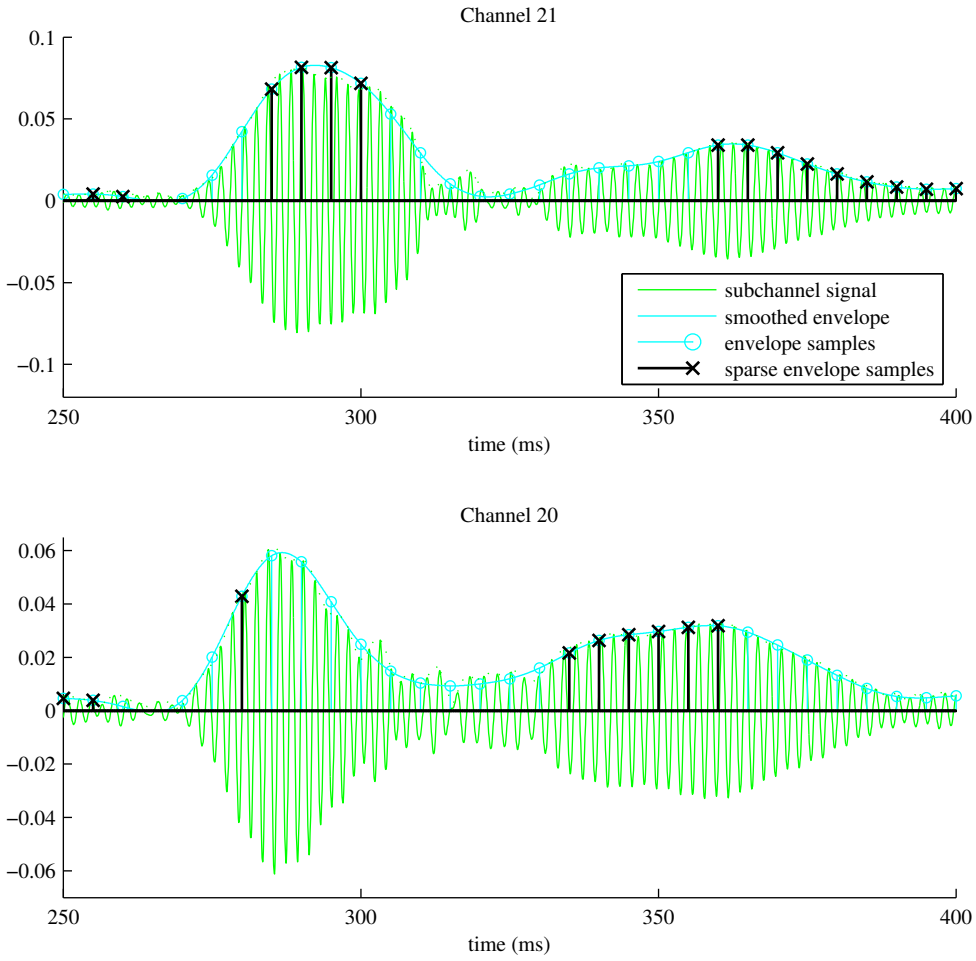


Fig. 4.8: Sparse sampling of a lowpass filtered auditory envelopes

this particular subband signal with a similar number of samples. However, because the sparse envelope representation is a sampling of a low-frequency signal at regular intervals, the sample position (modulo the sampling rate) is known implicitly and does not need to be encoded as side information.

4.2.3 Shortcomings of perceptual subband envelope representation

As we have described it in this chapter, the envelope based perceptual representation that we propose for testing the iterative reconstruction algorithm is a simplified ver-

sion of common perceptual models that have been developed for investigating human perception. These models were used to study masking properties and modulation pattern perception, but do not address the perception of pitch. In fact, temporal fine structure cues are often considered essential for pitch perception [Zeng et al., 2004]. However, we will demonstrate that pitch is encoded in the envelopes indirectly, but this lack of pitch sensitivity in the model translates into poorer performance when reconstructing sounds that are highly pitched, such as pure tones.

We demonstrate this problem by showing the envelope representation of two pure sinusoids. While pure sinusoids even of short duration are not common in typical audio signals, they illustrate the problem that is common to all signals with little or no amplitude or frequency modulation. The spacing of channels used in this example is 0.5 ERB, from the implementation described in the next chapter.

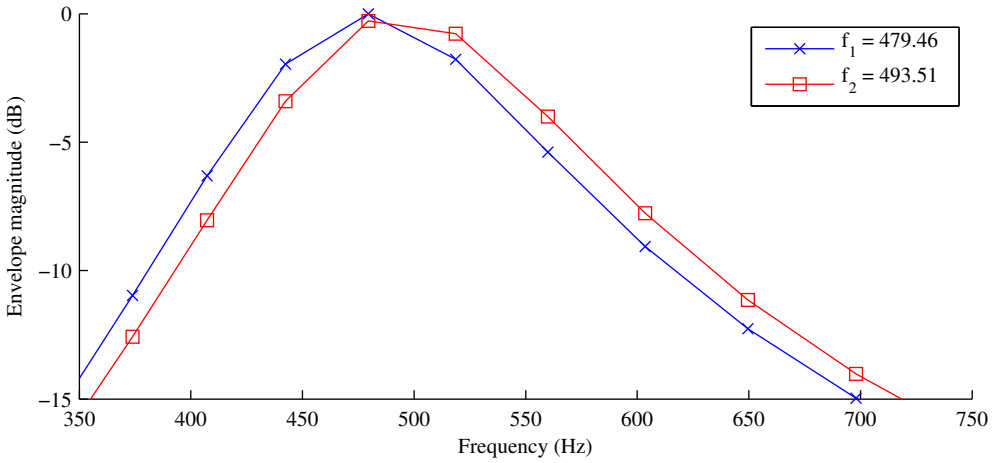


Fig. 4.9: Auditory envelope representations of two sinusoids

Figure 4.9 shows the envelope magnitudes in dB of the two sinusoids across several auditory channels, at about 0.5s after the onset of each sinusoid. The line markers show the value of the envelope at the filter centre frequencies. The frequency of the first sinusoid is chosen to coincide with the centre frequency of an auditory channel at $f_1 = 479.46$ Hz, and the second is chosen such that its frequency is half a semitone higher (a scale factor of $\sqrt[24]{2}$), $f_2 = 493.51$ Hz. The difference in pitch between these two tones is quite audible, but the envelope amplitudes are very similar. In

the auditory channel with centre frequency f_1, the difference between the envelopes is 0.28 dB. The next higher auditory filter (at 518.6 Hz) shows a difference of about 1 dB. In other auditory channels the difference is only slightly larger, never exceeding 1.7 dB.

We can see that the frequency information is present in the envelopes, but in a very indirect and subtle manner. In psychoacoustic theory, this is the essence of the place theory of pitch, which states that the percept of pitch is based on the location on the BM where the stimulus is strongest [Moore, 2003] and that closely spaced auditory channels are needed to accurately capture pitch [Smith et al., 2002]. Computationally finding the pitch on the stimulus is very difficult however, in particular if the envelopes are corrupted by noise (for example due to quantization). Knowing that pitch differences even smaller than we used in this example have been shown to be audible, we can expect the reconstruction algorithm to have problems reproducing pitched sounds accurately, especially once the masking algorithm and quantization have been applied.

The solution to this problem is the explicit inclusion of some pitch information. In the pulse representation, pitch is implicit in the pulse timing information, but this information is present even when not required for perceptually transparent encoding. A challenge for future research is to find a model that combines the envelope representation with pitch information which is omitted where pitch distortion is not perceptible. This will then change the structure of the encoded parameters and as a result, the decoder's reconstruction algorithm will need to be altered to match the encoded parameters.

The details of pitch perception are still an active area of research and thus beyond the scope of the research presented here. For the experimental iterative reconstruction algorithm in this thesis, the envelope model is deemed adequate and, in fact, the reconstruction method shows the strengths and shortcomings of the envelope model.

4.3 Reconstruction from the sparse envelope representation

This section presents the algorithm to synthesize a signal based on the sparse subband envelope representation described above. In particular, we describe the refinement

function to compute the next signal estimate given the envelope representation of the current estimate. The refinement function is based on the estimation algorithm presented in the previous chapter but modified to account for the sparse envelope representation.

The overall structure of the algorithm to turn the coded SAER into an audio signal is shown in Fig. 4.10. The algorithm can be divided into three basic steps: the decoding to convert the input stream back into the the sparse envelope representation (labeled "decoder"), the conversion of the sparse envelope representation into a set of envelope constraints (the "fixed expansion" and "envelope limit" blocks), and the iterative reconstruction of the audio signal using the envelope limits (the looped structure including the "compare & adjust" block and the filterbanks). Note that the "compare & adjust" block uses data from both the fixed expansion and the envelope limit blocks and controls when the reconstructed signal is passed to the output. This is indicated by the dashed line controlling the switch.

The role of the first step is simply to unpack the bitstream provided by the encoder back into the sparse envelope representation that was described in the previous section. It is assumed that the encoding and decoding of the bitstream is lossless and transparent to the overall system with the exception of some error due to quantization of the envelope amplitudes. The focus here is to show the concept of iterative reconstruction from a perceptual context.

4.3.1 Fixed envelopes and envelope limits

The next part of the reconstruction algorithm converts the SAER into two sets of constraints to which the auditory envelopes of the reconstructed signal are made to conform. The first set is simply an expansion (upsampling) of the sparse envelope samples by piecewise constant segments. We call this set the fixed envelopes.

Recall that when generating the SAER, the envelopes are sampled at regular intervals prior to sparsification. Thus, the lack of an envelope sample at the reconstruction stage conveys some information: the missing sample was removed due to having an amplitude less than the transmultiplexed pattern of some other sample. This means there is an upper bound that the reconstructed envelopes must not exceed wherever the SAER is missing samples.

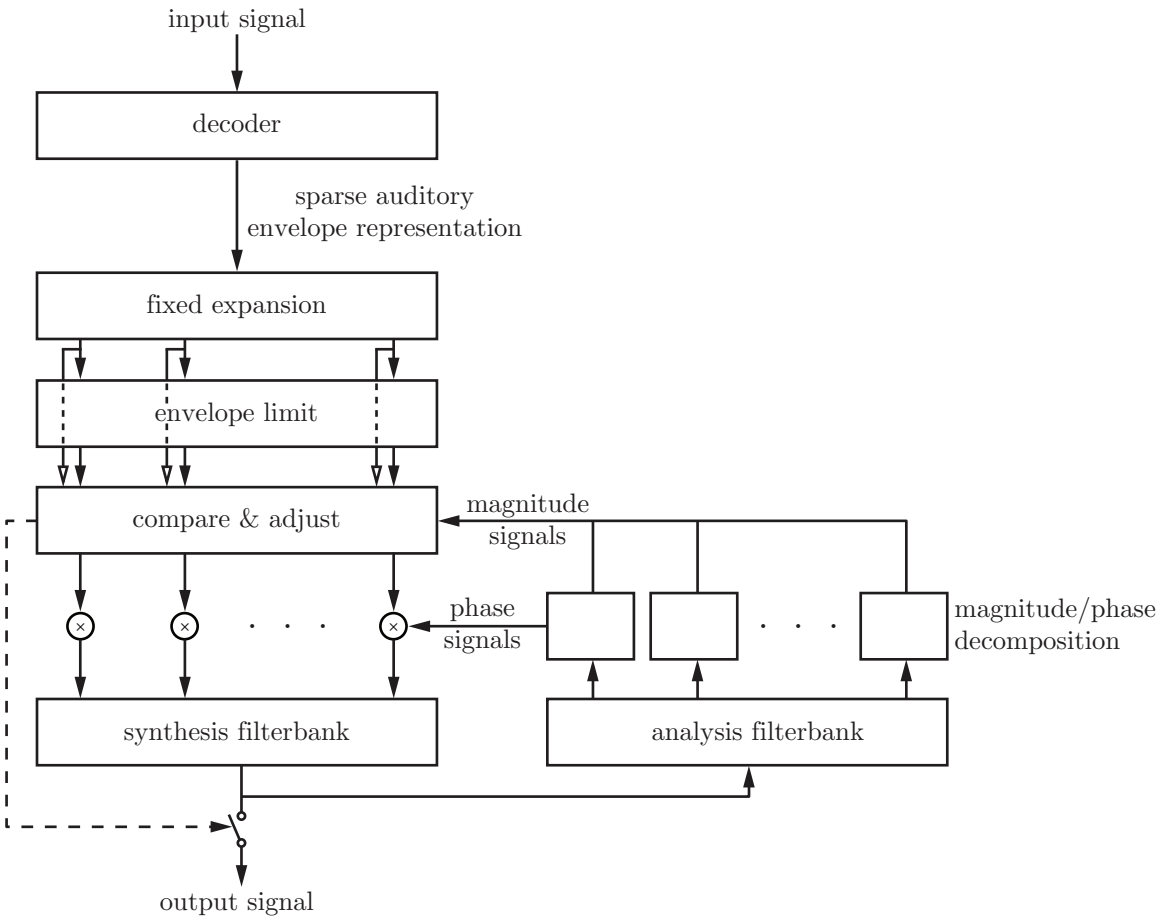


Fig. 4.10: The reconstruction algorithm for the sparse envelope representation

With this in mind, we calculate a limit on the envelopes in the sparsified regions of the envelope representation by applying the transmultiplexer model to the sparse representation. In envelope regions that are not defined by the fixed envelopes, the transmultiplexed envelope pattern due to the transmitted samples becomes the upper limit that the envelope is allowed to reach during reconstruction. If during the iterative loop the envelope of the estimate exceeds the upper limit, the subchannel signal estimate is adjusted. We can regard this limiting of the envelopes as an optimized recalculation of the transmultiplexer masking model outside the reconstruction loop. The envelope limit is a precomputation of the masking model without explicit evaluation of each envelope sample within the loop. Note that the limiting value is nonzero even if there are no masking envelope samples in the spectro-temporal vicinity. In this case, the limit is given by the threshold in quiet, as it was computed at the encoder's masking stage. This is the amount of noise the decoder is allowed to generate if no samples are present since this noise will be inaudible to the human listener.

The relationships between the smoothed envelopes, the fixed envelopes, and the upper envelope limits are shown in Fig. 4.11. The samples of the smoothed envelopes are interpolated into a stepwise-constant curve at points where the envelopes are known due to the envelope samples. The piecewise parabolic-like curves are the upper envelope limits generated by the transmultiplexed envelope samples both in the time domain and in adjacent channels (see, for example, the curves at around 290 ms and at around 360 ms, respectively). We can clearly see that every section of the auditory envelopes are defined either by a fixed envelope section or an envelope limit. The stepwise approximation of the envelopes by the fixed sections does introduce some error but is adequate for the implementation.

4.3.2 The iterative reconstruction loop to determine the carriers

The last step of the decoder is the iterative reconstruction of the audio signal estimating the carriers to match the envelope data. This process is similar to the cochleagram inversion as described in [Slaney, 1995], and we describe it here as a modification of the algorithm described in Section 3.3.2 and Fig. 3.2. In Fig. 4.10, we can associate the synthesis filterbank with the block labeled **H** in Fig. 3.2 and the analysis filterbank with the block labeled **G**. In Fig. 4.10, the individual channels are shown as parallel

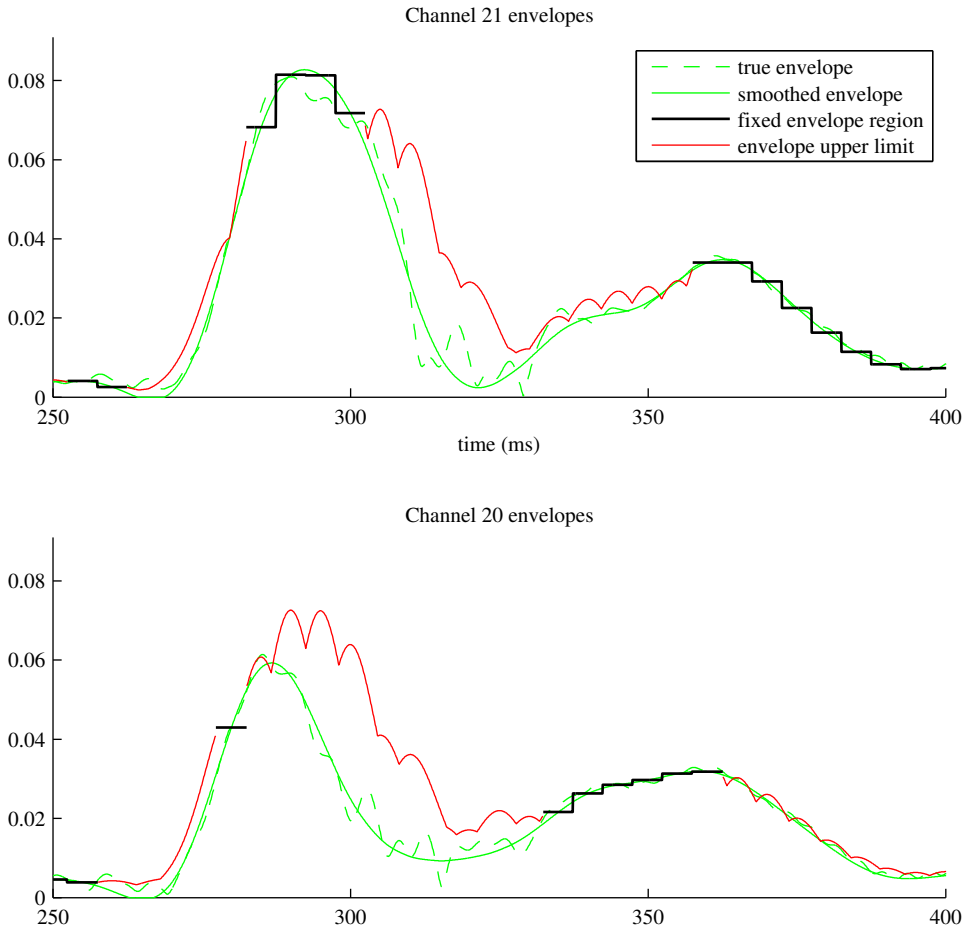


Fig. 4.11: The smoothed envelopes and the reconstruction target specification in two adjacent channels. The envelopes are specified by fixed envelope sections (labelled $\overline{\mathbf{c}}'$ in Section 4.3.2) and the maximum envelope limits ($\tilde{\mathbf{c}}$) that affect multiple channels at the same time offset.

branches of the analysis and synthesis filterbank, which in Fig. 3.2 are combined in the vector $\mathbf{c}^{(i)}$.

The key difference between the algorithm here and Algorithm 1 is that we do not have a full set of envelopes ($\overline{\mathbf{c}}$ in Algorithm 1) to which to adjust the estimate. Instead, we only have the set of sparsified envelope samples and the set of upper limits calculated in the previous stage of the decoder. These two sets influence the generation of the next iteration estimate in different ways.

Amending the notation from the previous chapter, we use $\overline{\mathbf{c}}'$ to denote the fixed envelopes. In the sparsified sections, $\overline{\mathbf{c}}'$ is set to zero. From this the previous stage generates the envelope upper limits $\tilde{\mathbf{c}}$ using the transmultiplexer patterns. As in Section 3.3.2, $\hat{\mathbf{c}}^{(i)}$ is the magnitude constrained ith estimate of the subband signals and $\hat{\mathbf{x}}^{(i)}$ is the audio signal calculated from the previous subband signal estimate. The subband signals from the current signal estimate are denoted $\mathbf{c}^{(i)}$. The first estimate is initialized with $\mathring{r}$, random complex numbers of unit magnitude.

Algorithm 2 Iterative reconstruction algorithm with envelope limits

$i \leftarrow 0$;
$\hat{\mathbf{c}}^{(0)}_{[k]} \leftarrow \overline{\mathbf{c}}'_{[k]}\mathring{r} \qquad k = 1, \ldots, (MN)$;
repeat
  $i \leftarrow i + 1$;
  $\hat{\mathbf{x}}^{(i)} \leftarrow \mathbf{H}\hat{\mathbf{c}}^{(i-1)}$;
  $\mathbf{c}^{(i)} \leftarrow \mathbf{G}\hat{\mathbf{x}}^{(i)} + \mu\mathbf{r}$
  $\hat{\mathbf{c}}^{(i)}_{[k]} \leftarrow \overline{\mathbf{c}}'_{[k]} \frac{\mathbf{c}^{(i)}_{[k]}}{|\mathbf{c}^{(i)}_{[k]}|} \qquad \forall k$ where $\overline{\mathbf{c}}'_{[k]} \neq 0$;
  $\hat{\mathbf{c}}^{(i)}_{[k]} \leftarrow \min(\tilde{\mathbf{c}}_{[k]}, \hat{\mathbf{c}}^{(i)}_{[k]}) \frac{\hat{\mathbf{c}}^{(i)}_{[k]}}{|\hat{\mathbf{c}}^{(i)}_{[k]}|} \qquad \forall k$ where $\overline{\mathbf{c}}'_{[k]} = 0$;
until $D_M(\overline{\hat{\mathbf{c}}^{(i)}}, \overline{\mathbf{c}^{(i)}}) < \tau$ or $i = L$.

Refinement step and error measure

Reflecting the argument of reaching an acceptable solution as part of the reconstruction algorithm, we introduce a terminating condition based on the Signal-to-Error Ration (SER) as presented in Chapter 3, with some important modifications. These modifications are necessary due to the sparse representation. The reconstruction algorithm does not have a full version of the original envelopes. However, the envelopes $\overline{\hat{\mathbf{c}}^{(i)}}$ are the current best estimate of the original envelopes, since the iterative loop forces $\overline{\hat{\mathbf{c}}^{(i)}}$ to conform to $\overline{\mathbf{c}}'$. For all k that are specified in the sparse envelope representation, $\overline{\hat{\mathbf{c}}^{(i)}_{[k]}}$ is equal to $\overline{\mathbf{c}_{[k]}}$ but, for other values of k, $\overline{\hat{\mathbf{c}}^{(i)}_{[k]}}$ is equal to $\overline{\mathbf{c}^{(i)}_{[k]}}$ *unless* $\hat{\mathbf{c}}^{(i)}_{[k]}$ is corrected by the envelope limits, in which case $\overline{\hat{\mathbf{c}}^{(i)}_{[k]}}$ is equal to $\tilde{\mathbf{c}}_{[k]}$.

So, $D_M(\overline{\hat{\mathbf{c}}^{(i)}}, \overline{\mathbf{c}^{(i)}})$ measures the amount of modification the current iteration of the loop made to the signal estimate to fit its envelopes to the target sparse representation.

If the signal is close to matching the target envelopes, the modifications are small, so this difference metric should be a good indication of the reconstructed signal quality and generally decrease at each iteration. However, since it is a global average, it is not a good perceptual measure. The measure as stated above makes no distinction between low-level noise present throughout the signal and a large noise burst isolated in a short section. A human listener would find the latter far more disturbing and thus the square error measure over the entire signal is necessarily not a useful measure as a stopping condition.

To overcome this limitation, we introduce a modification of the iterative reconstruction algorithm that processes the audio file in short segments. In this modified implementation, the error is calculated just on these short segments and thus can be used for a better estimate of perceived error based on the current signal level. Based on this segmental error measure, we will examine the relationship of D_M to perceived quality in Chapter 5.

4.3.3 Modifying the algorithm for different perceptual representations

As presented, the reconstruction algorithm is specifically designed for the sparse envelope representation of audio signals. As pointed out earlier in this chapter, this representation could in the future be improved by adding some phase information. Here, we briefly examine the changes that are needed to accommodate an alternative perceptual representation that includes some temporal detail.

Any modification to the algorithm must maintain the basic premise that the signal should be modified from the previous estimate only when necessary. We can envision a situation where the phase and/or instantaneous carrier frequency information is transmitted for select sections of a subband. A simple implementation could replace the carrier signal at the given section with the transmitted information, causing a discontinuity at the transitions that will be smoothed out in subsequent iterations of the algorithm. A somewhat more elaborate method would be to use a smooth modification of the instantaneous frequency of the carrier signal to avoid discontinuities. This is an area which must be addressed once a more complete perceptual representation is developed. Such an envelope-and-phase representation can be used as a generalization of models using real-valued subband signals, expanding the use of

the iterative reconstruction algorithm to many established auditory models.

4.3.4 Nonlinear effects of loudness in the hearing system

In many perceptual models including that used by Feldbauer, nonlinear and time-varying effects of the IHC transduction, in particular companding and adaptation, are an important part of the perceptual representation. In this thesis we focus on the reconstruction of audio from a sparse envelope representation, thus we only consider these nonlinear effects in the difference function D_M and the quantizer. Effectively, the power-law companding of Eq. (4.5) is moved into a modified form of Eq. (3.47),

$$D_M(\overline{\hat{\mathbf{c}}^{(i)}}, \overline{\mathbf{c}^{(i)}}) = \| \ |\hat{\mathbf{c}}^{(i)}|^{0.4} - |\mathbf{c}^{(i)}|^{0.4} \ \|^2, \tag{4.7}$$

scaling the error to be more representative of perceived loudness. Incorporating the effects of IHC temporal adaptation into the analysis model and the reconstruction algorithm is beyond the scope of this thesis and is an opportunity for future research.

4.4 Conclusion

In perceptual coding systems, iterative reconstruction methods are typically not used due to the computational requirements. However, in this chapter we present the argument that iterative reconstruction allows for using a signal representation that is strictly based on the perceptual content of the audio signal. In particular, when considering high frequency sounds, the fine temporal structure of signal is far less important than the overall amplitude modulation. Furthermore, by design, the iterative reconstruction algorithm compares the output to the encoded perceptual representation in the perceptual domain, thus having an internal measure of how well the reconstruction matches the representation. Given this information, the reconstruction algorithm can be used to evaluate perceptual representations if the internal reconstruction quality measure is compared to subjective evaluations. We use this capability to test a perceptual representation that cannot easily be inverted by non-iterative methods.

Based on a physiologically motivated audio coding system using auditory subband

pulse encoding, we develop a perceptual representation using the Hilbert envelope of the auditory subbands. This representation can be viewed as encoding only the subband energy at the rate that the ear can perceive the changes without the temporal specifics (the phase of the carrier). Additionally, a masking model is applied that discards envelope sections that are inaudible. The resulting sparse envelope representation is comparable to the output of perceptual models that have been used in auditory research. We show that the proposed representation might be problematic for tonal sounds.

We complete this chapter with a high-level overview of the reconstruction algorithm for the sparse envelope representation. In particular we show how the masking model can be precomputed, reducing the computation necessary in the iterative loop. In the following chapter, we describe an implementation of this algorithm in more detail and present the results of encoding various types of audio signals.

Chapter 5

Implementation and Results

This chapter details the implementation of the perceptual model (encoding the signal into the sparse envelope representation) that is described in Chapter 4 and the implementation of the iterative method to reconstruct an audio signal from the model parameters (transforming the sparse envelope representation into an audio file). The perceptual model description will start with the design of the analysis filterbank (FB), showing how the time and frequency characteristics of the filters in the FB can be chosen such that the iterative reconstruction can converge towards a solution, then show how the FB outputs are converted into the sparse auditory envelope representation (SAER). This conversion is done by first extracting and subsampling the envelopes, then sparsifying the envelope samples using the transmultiplexer masking model. In the reconstruction algorithm the sparse envelope samples are converted into the fixed envelope sections and the envelope limits. We continue by describing the iterative reconstruction loop using FBs, highlighting the advantages and disadvantages of a system that employs finite-delay processing.

By subjective testing using the Multiple Stimuli with Hidden Reference and Anchor (MUSHRA) [ITU-R, 2003] protocol, we evaluate the quality with which the model can represent speech and audio signals and estimate the efficiency in terms of bits required to encode signals at a reasonable quality. Objective measurements of the Signal-to-Error Ratio (SER) are used to compare the reconstructed signal to the original as well as determine the convergence of the iterative algorithm. These measures are used to gain an understanding of the dynamics of the algorithm when

processing different types of audio signals, some of which are encoded very well using the SAER.

5.1 Encoding the audio signal into a sparse envelope representation

In the previous chapter, the conversion of the audio signal into a sparse envelope representation was described from the viewpoint of perceptual modeling, presenting it as an extension of the way the representation described by Feldbauer [Feldbauer, 2005] is computed. This section will give a more detailed description of the implementation of our sparse auditory envelope sample (SAES) extraction.

5.1.1 Peripheral auditory analysis

At the core of the perceptual model is the decomposition of the input audio signal into a set of auditory channel signals, which are then processed to find the Hilbert envelopes within each channel. These auditory channel signals are found using a set of M complex gammatone filters, with responses given by

$$g_m[n] = a_m \left(\frac{n}{f_s}\right)^3 \exp\left(-2\pi 1.019\ \mathrm{ERB}(f_m)\frac{n}{f_s}\right) \exp\left(-2\pi i f_m \frac{n}{f_s}\right), \quad m = 1, \ldots, M. \quad (5.1)$$

This FB is a slight modification of the standard gammatone FB as described by Eq. (2.1), where the real-valued sinusoidal modulation term is replaced with a complex sinusoidal term. The reason for using a complex term is that the resulting response can easily be decomposed into an envelope and carrier component by a magnitude operator. The formulation of Eq. (5.1) can also be regarded as two filters in quadrature (for the real and the imaginary component). Another modification is that the constant phase offset is missing from Eq. (5.1). This is not needed since the masking calculation is purely based on the envelopes of the responses.

The FB is designed with the goal of modeling the human auditory system while also being suitable for the iterative reconstruction algorithm. We can design this

gammatone FB using the framework presented in Chapter 3 to form a snug frame by choosing appropriate values for the total number of channels M, the channel centre frequencies f_m, and the filter scale values a_m.

The total number of channels in the FB is directly related to the spacing between channel centre frequencies and the overall bandwidth that the FB covers and also to the amount of redundancy in the representation, which we wish to control to retain suitability for coding. The design parameter we focus on is the spacing of channel frequencies, and we seek to find as wide a spacing to still allow reconstruction with good quality. We use a spacing of about 2 filters per ERB. To get a uniform spacing on the ERB scale, we find the set of frequencies computed using

$$f_m = \frac{1000}{4.37}\left(\frac{\text{ERB}\#_m}{24.7} - 1\right), \tag{5.2}$$

where the $\text{ERB}\#_m$ are chosen to be $1, 1.5, 2, \ldots 32$ for a total of 65 filters. The filter at the bottom of the spectrum has a centre frequency of 26.03 Hz and the filter at the top of the spectrum has a centre frequency of 7743 Hz. Our implementation is designed with sampling frequency $f_s = 16\,000$. Technically, Eq. (5.1) specifies infinite-impulse filters, but we chose to implement the FBs using FIR filters, cutting off the impulse response at 1800 samples or 112.5 ms. For the lowest frequency filter, the amplitude of the impulse response decays to less than 1.5×10^{-5} of the peak.

Some other modifications to the FB are made to suit the iterative reconstruction algorithm. In addition to the complex formulation already noted in Eq. (5.1) above, we find that the simple uniform spacing in the ERB scale did not result in a sufficiently flat response as needed for iterative reconstruction. The addition of an equalizing postfilter as Feldbauer describes was not found to be a good solution due to the additional spreading and delay it would entail due to the longer effective impulse response. Instead, we adapted the frame-theoretic analysis of gammatone filterbanks found in [Strahl and Mertins, 2009]. One of the methods proposed by Strahl and Mertins to optimize the frame bound ratio is to "nudge" the filter centre frequencies to lower the spectral ripple and tilt of the overall frequency domain response, ensuring a nearly flat response over the frequency band of interest.

The frequency response of the analysis/synthesis filterbank combination is shown

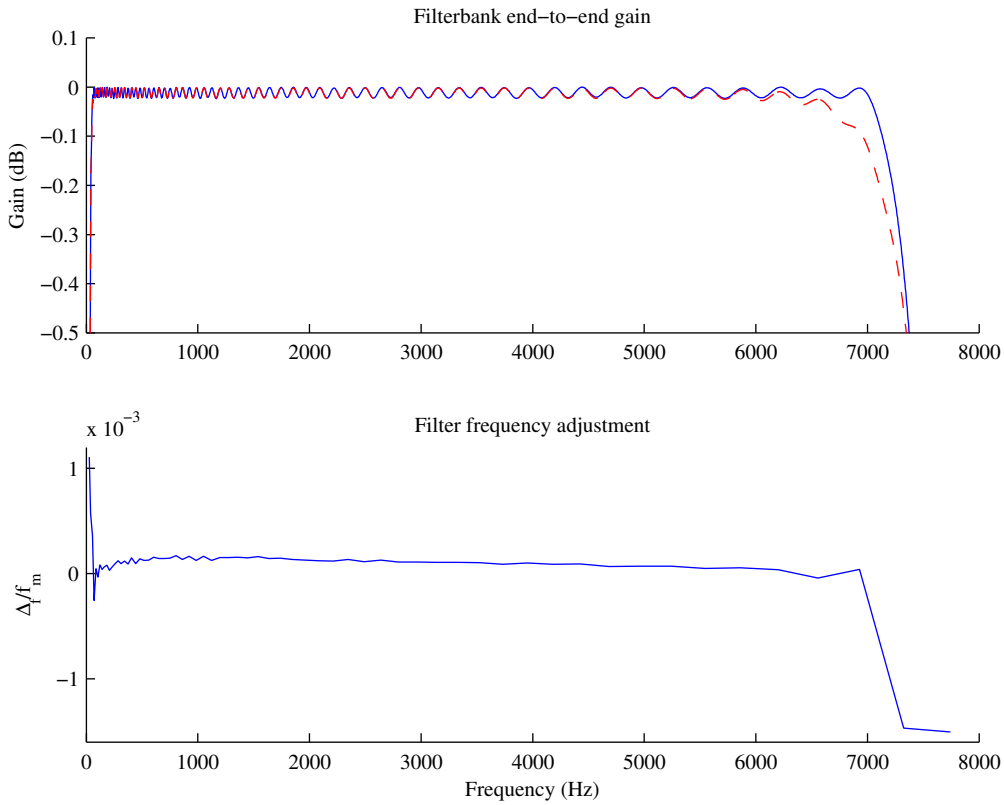


Fig. 5.1: Filterbank end-to-end gain and frequency adjustment to flatten response of the gammatone FB. The upper graph shows the original FB response as a dashed line, and the adjusted FB response as a solid line. The bottom graph shows the adjustment to the center frequencies of the individual filters as a relative shift.

in the top of Fig. 5.1. The frequency response of the original filter spacing (shown as a red dashed line) can be seen to be very flat already, within 0.03 dB of the maximum from 66 Hz to 6350 Hz. Above 6350 Hz, the gain decreases rapidly. For typical non-iterative reconstruction methods, this roll-off is usually not a significant problem, being just a slight lowpass filtering of the output signal. However, in the iterative reconstruction framework, this was found to be problematic. The repeated synthesis-analysis of the signal and envelope correction amplified the error in the region from 6 to 7 kHz. Due to the nonlinear processing of the envelope correction step, this error spreads to other frequencies. To correct this roll-off, the filter centre frequencies were

adjusted by numerical optimization with the filter ripple (from the lowest auditory channel frequency to the highest auditory channel frequency) as the cost function to be minimized. The resulting adjustments are shown in the bottom graph of Fig. 5.1. Note that the frequency adjustment is shown relative to the centre frequency, so there are only small adjustments for most of the filters, but the last two filter frequencies are lowered by more than 10 Hz such that the top filter now has a centre frequency of 7732 Hz. When combined with the other filters in the FB, this alters the gain in the top bands sufficiently to flatten the FB response. The lowest filter is also shifted by a small amount, which is not visible in the top of Fig. 5.1.

The small adjustment can be quantified in terms of the frame bound ratio (B/A, see Chapter 3). For the original gammatone FB, the frame bound ratio of the is 1.0028 measured from 70 Hz to 6000 Hz but 1.014 from 55 Hz to 7000 Hz. With the adjustment, the modified FB has a frame bound of 1.0027 over the larger frequency range.

5.1.2 Envelope computation, filtering and subsampling

With the input signal split into 65 complex-valued narrowband signals, we compute the magnitude of each subband signal sample resulting in a fully oversampled but real-valued set of envelopes. Each of these envelopes is then low-pass filtered and subsampled. The filters applied to avoid aliasing during the envelope sampling process are linear-phase FIR filters of order 512 with cut-off frequency given by the bandwidth parameter b_m as used in Eq. (5.1). As stated in Chapter 2, for each subband m with centre frequency f_m, $b_m = 1.019\ \mathrm{ERB}(f_m)$. Based on this cutoff frequency, the integer subsampling factor of the envelopes is

$$K_m = \left\lfloor \frac{f_s}{2b_m} \right\rfloor . \tag{5.3}$$

The filter cutoff frequency is set to be at least 50 Hz, for a maximum subsampling rate of 160. This limit was used to facilitate implementation based on short-time segments.

5.1.3 Sparsification using a transmultiplexer masking model

At this stage, the original audio signal is represented by a set of real-valued samples of the filtered envelopes, similar to the full pulse based representation used by Kubin and Kleijn in [Kubin and Kleijn, 1999]. The actual number of samples to encode a given segment of audio is increased; on average there are 2.4 envelope samples per input sample at $f_s = 16\,000$.

As described in the previous chapter, the number of envelope samples is reduced by applying the transmultiplexer based masking model to the envelope samples. The implementation we use is analogous to the procedure described in [Feldbauer, 2005], where the envelope samples take the place of auditory pulses. For completeness, we describe this implementation here in more detail.

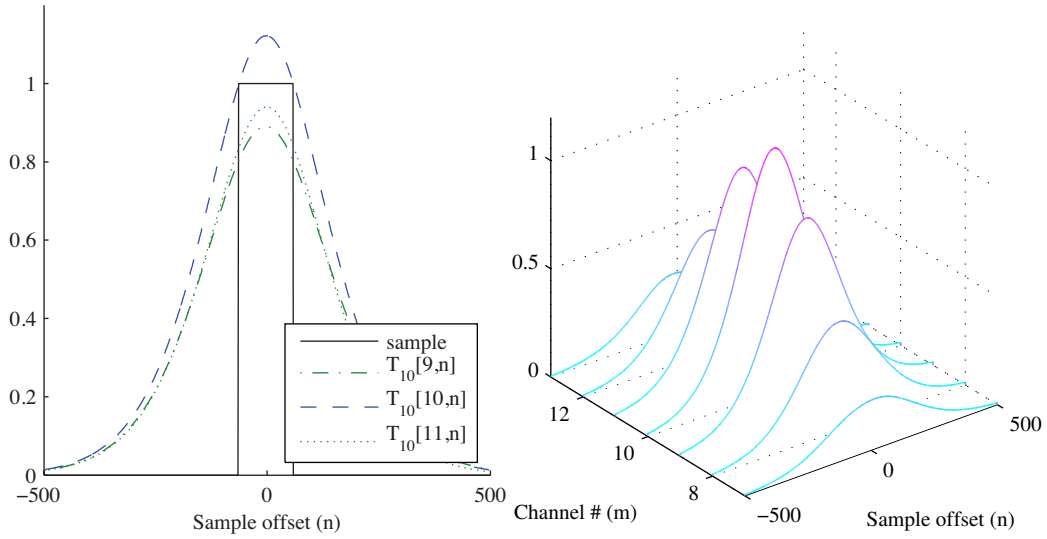
We denote the transmultiplexed envelope patterns as $\mathbf{T}_k[m, n]$ for the pattern of an envelope sample in channel k, and they are simply computed by

$$\mathbf{T}_k = r_S \overline{\mathbf{G}\mathbf{h}_k}, \tag{5.4}$$

which is the impulse response of the synthesis FB channel k ($\mathbf{h}_k$ is the first column of $\mathbf{H}_k$) passed through the analysis FB and is equivalent to the transmultiplexer *pulse* setup in [Feldbauer, 2005]. We scale the pattern by $r_S = 1.12$ to account for the fact that envelope samples are spread over time compared to auditory pulses. For the algorithm below, we store the patterns such that the peak of the pattern $\mathbf{T}_k[m, n]$ is at $n = 0$, and it extends backwards and forwards in time. The choice of r_S is illustrated by Fig. 5.2, showing a unit envelope sample as it would appear in $\overline{\mathbf{c}}'$ and the corresponding transmultiplexed envelope pattern. It can be seen that the pattern in the same channel (dashed line) matches the sample at the edges. Note that the transmultiplexed envelope patterns can be computed offline.

The sparsification process begins by creating a list of the envelope samples as triplets of amplitude A_p, temporal location n_p, and channel index m_p, then sort this list such that the sample amplitudes are in descending order, so $A_p \geq A_{p+1}$. We assume the signal (or current section of the signal) has a total of P_O envelope samples after subsampling.

The first step in reducing the size of this list is to remove all envelope samples



(a) Sample in channel 10 and transmultiplexed pattern with scale factor $r_S = 1.12$.

(b) Transmultiplexed pattern $\mathbf{T}_{10}$.

Fig. 5.2: Two views of transmultiplexed sample pattern $\mathbf{T}_{10}$ in channel 10 and neighbouring channels. The gain factor r_S is used to match the spread of the pattern to the upsampled unit envelope sample.

that fall below the absolute threshold of hearing in quiet. The threshold of hearing in quiet T_q is a constant value within each auditory channel m and we use a formulation of $T_q(f_m)$ based on an equal loudness contour as derived in [Soulodre, 1998] in dB SPL,

$$T_q(f) = 3.64(f/1000)^{-0.8} - 6.5e^{-0.6(f/1000-3.3)^2} + 10^{-3}(f/1000)^4 - 80.64e^{-4.712(f/1000)^{0.5}}. \tag{5.5}$$

The values of $T_q(f_m), m = 1, \ldots, M$ are converted into envelope thresholds by assuming that the input signal is scaled such that a sinusoid at 1 kHz with peak amplitude 1 is reproduced at 80 dB SPL. This would be considered "very loud" but below the threshold of pain. This is a useful overestimation given that we have no a priori knowledge the volume setting of the end user. Thus, we evaluate every envelope sample p and remove all samples from the list where

$$A_p < 10^{T_q(f_{m_p})/20}, \qquad p = 1, \ldots, P_O, \tag{5.6}$$

to yield a list of size P_T envelope samples. For typical signals, this process removes most of the envelope samples in quiet parts of the signal, but also removes many samples for signals with compact spectra such as narrowband signals.

We can now apply the transmultiplexer based masking model on the remaining envelope samples. Beginning with the envelope sample with largest amplitude ($p = 1$), we iterate over the list and remove samples from the current index p_c onwards based on the envelope masking decision. As explained in the previous chapter, we use the transmultiplexed envelopes for a gammatone impulse in the channel m_{p_c} and scale it using the envelope sample amplitude A_{p_c} and the impact factor r_I. Then, we search the remaining list of envelope samples: using p_t to indicate the index of the envelope sample being tested, we iterate from $p_t = p_c + 1$ to $p_t = P_O$ and compare this sample to the scaled and shifted transmultiplexed envelope pattern $r_I A_{p_c} \mathbf{T}_{m_{p_c}}$. If the sample being tested falls below the pattern, its amplitude is set to 0 to mark it as deleted. The full procedure is shown in Algorithm 3.

Algorithm 3 Transmultiplexer masking model sample removal.

```
for p_c = 1, ..., P_O do
  if A_{p_c} ≠ 0 then
    for p_t = p_c, ..., P_O do
      if A_{p_t} < r_I A_{p_c} T_{m_{p_c}}[m_{p_t}, n_{p_t} − n_{p_c}] then
        A_{p_t} ← 0;
      end if
    end for
  end if
end for
```

It is easy to see how the impact factor directly controls the number of samples removed from the list by either over- or underestimating the amount of masking, and can thus be used to control the amount of sparsification. We indicate the size of the final list with P_{r_I}. This list of the remaining envelope samples then forms the sparse perceptual representation that is sent to the decoder after quantization and entropy coding.

In addition to the impact factor r_I, the number of envelope samples removed by sparsification is very dependent on the type of audio signal, in particular its overall signal bandwidth and the sharpness of its spectral peaks. In the section showing

results below we give some examples of the sparsification achieved with both speech and audio files.

5.1.4 Experimental quantization of envelope samples

To explore the use of the SAER as a viable audio coding method, we implement and test a simple scalar quantizer applied to the envelope samples. The quantization is performed in the log domain of the envelope sample values and thus is robust to variations of scale such as quiet vs. loud passages of a sound file. In addition, implementing a power-law compression of the envelope samples (companding) can be achieved by a simple scaling of the log domain values.

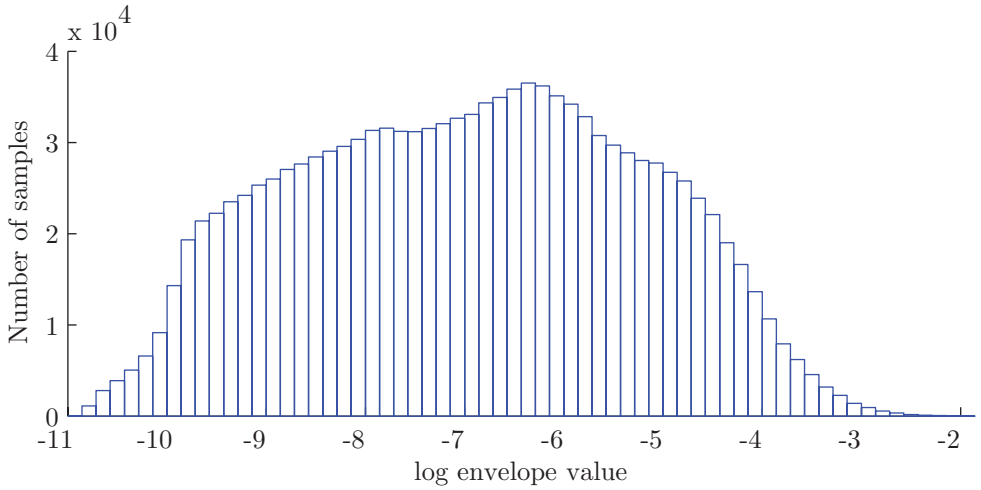


Fig. 5.3: Log-domain distribution of the sparsified envelope values

A histogram of log-domain values of envelope samples is shown in Fig. 5.3, using a total of 64 bins to simulate a 6-bit quantizer as was used in the subjective testing. The histogram shows that the distribution is not very uniform, so it is reasonable to assume that the uniform scalar quantizer used might not be the most efficient method to encode the envelope samples. However, it is useful to show that the representation is robust to some amount of systematic error that is introduced by quantizers. A comprehensive evaluation of various methods such as adaptive quantizers, entropy-constrained quantization, and vector quantizers [Kleijn, 2004] would be necessary for

implementing a full SAER coding system. Since we focus just on evaluating the idea of a sparse envelope representation and the iterative reconstruction, studying quantization and entropy coding is beyond the scope of this thesis.

5.2 Reconstruction implementation

The overall structure of the reconstruction algorithm is shown in Chapter 4 in Fig. 4.10. Here we detail the implementation of the two major parts of the overall algorithm. The first part is the creation of the "fixed envelope regions" and the "envelope limits" ($\overline{\mathbf{c}}'$ and $\tilde{\mathbf{c}}$), a non-iterative expansion of the SAER. The second part is the iterative loop to find the carrier information to fit the envelopes.

5.2.1 Computing the fixed regions and the envelope limit

Generating the "fixed envelopes" section $\overline{\mathbf{c}}'$ is quite straightforward from the list of envelope samples p. The array $\overline{\mathbf{c}}'$ is initialized with all zeros, then for each envelope sample p we set

$$\overline{\mathbf{c}}'[m_p, n] = A_p, \qquad n = n_p - \frac{K_m}{2}, \ldots, n_p + \frac{K_m}{2}. \tag{5.7}$$

This is a stepwise constant upsampling of the envelope samples. Since the sampling is sparse and zeroes represent absence of an envelope sample, filtering $\overline{\mathbf{c}}'[m_p, n]$ is not appropriate, although different expansions can be considered (such as linear interpolation). In our implementation, we simply accept a small amount of error added by this step.

Next, the envelope limits are generated analogous to the masking decision algorithm above, with the key difference that the transmultiplexed envelopes of all the envelope samples are combined into full-rate sets of envelopes. As above, we begin with the threshold in quiet. Using the notation $\tilde{\mathbf{c}}[m, n]$ for the envelope limit in channel m at time index n, we initialize the surface with

$$\tilde{\mathbf{c}}[m, n] = 10^{T_q(f_m)/20}, \qquad m = 1, \ldots, M; \forall n. \tag{5.8}$$

We then iterate over the list of envelope samples of size P_{r_I}, adding the contribu-

tion of the transmultiplexed pulse envelopes to the corresponding region of $\hat{\mathbf{c}}$. Each transmultiplexed pulse envelope is added with the same scaling as used in Algorithm 3 above:

$$\tilde{\mathbf{c}}[m,n] = \max(r_I A_p \mathbf{T}_{m_p}[m, n-n_p], \tilde{\mathbf{c}}[m,n]), \qquad p = 1, \ldots, P_{r_I}, \tag{5.9}$$

where m and n span the region of the transmultiplexed pulse envelopes, offset by the time index of the envelope sample n_p. It is important that the same impact factor is used that generated the sparse representation, since the decision to remove samples from the representation is made on a threshold that is equivalent to $\tilde{\mathbf{c}}$.

5.2.2 Iterative reconstruction implementation

We describe the implementation of the iterative reconstruction algorithm using two basic approaches, by either reconstructing the entire signal from beginning to end as a unit or working on fixed-time parts of the signal sequentially (that is, parts of the signal will be "completed" while other parts of the signal are still being reconstructed). The former is more straightforward both conceptually and in its implementation; the latter can process signals of any length and converges to a solution faster.

To reconstruct the entire signal as a single unit starting with $\overline{\mathbf{c}}'$ and $\tilde{\mathbf{c}}$ is straightforward from Algorithm 2, since the circulant matrix notation was used explicitly to describe FIR FBs. So the implementation begins with initializing a set of 65 carrier estimates ($\hat{\mathbf{c}}^{(0)}$) with a random phase signal. In the loop, the signal estimate is generated by filtering all 65 carrier estimates with superimposed envelopes using the synthesis FB, where each channel filter has the complex conjugate, reverse-time gammatone impulse $g_m^*[-n]$ as impulse response. The filter outputs are combined to form the signal estimate $x[n]$ to be reanalyzed with the analysis FB to get the next set of carrier estimates. Implementation of the envelope correction is also straightforward from Algorithm 2.

Care must be taken to account for the filter delay. The iterative reconstruction algorithm is designed based on a zero-delay synthesis-analysis chain and it is assumed that if the analysis filters are causal, the synthesis filters are anticausal. Thus, if both filterbanks are implemented as causal filters, a delay compensation stage must realign

the signal. In the filter design as described, the delay is equal to the FIR filter order in each channel.

5.2.3 Finite-delay implementation of iterative reconstruction algorithm

The reconstruction of the signal by operating on the entire signal at once is an obvious way to implement iterative signal processing algorithms where changes in one part of the signal can affect the signal in causal and noncausal ways. However, operating on entire signals is impractical in many signal processing settings. It precludes operating on very long signals, such as audio streams, and even if signals can be segmented into finite sections that do not affect one another, for each section the receiver would have to delay processing until the entire data for that section has been received. The solution to this problem is to modify the reconstruction algorithm in such a way that the signal is split into short sections where only some of the sections are modified by the iterative algorithm. Once a section of the signal has been completed, it is stored and used by the algorithm to work on the next sections. As the completed section is no longer being modified, it can be passed on to the next stage of processing; in the case of audio signals, it can be output to the loudspeakers.

This type of finite-delay processing (the finite delay refers the the theoretical minimum amount of delay a section of the signal can incur due to reconstruction) has been well used with iterative reconstruction methods using lapped transforms such as LSEE-MSTFTM where the finite transforms provide for a natural segmentation of the signal. In particular, we look at the Real-Time Iterative Spectrum Inversion (RTISI) algorithm [Zhu et al., 2007; Gnann and Spiertz, 2009] as a model for finite-delay processing. However, in contrast to block-transform methods, the time-domain filter-bank based analysis and synthesis structure of our reconstruction algorithm provides no natural segmentation, which allows for greater flexibility.

We present a finite-delay processing implementation where the step-size of processing can be arbitrarily small. Using the circulant matrix notation of Chapter 3, we can show that an efficient implementation is possible that is analogous to the processing of lapped transform methods.

Our implementation uses a “lookback”, or fixed area, of 320 samples (20 ms) and a work area of 2208 samples (138 ms). The fixed and work areas overlap over 160

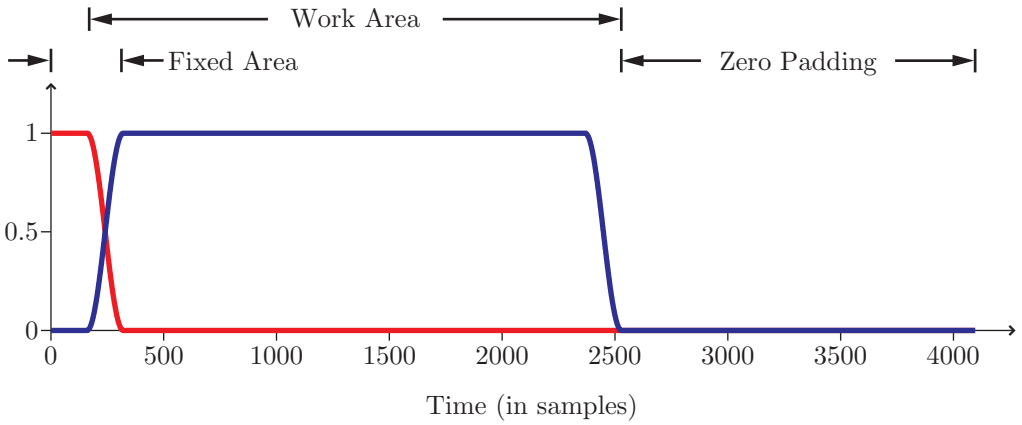


Fig. 5.4: Fixed and working part windows for the processing segment

samples and are windowed as shown in Fig. 5.4 with a Hanning window [Oppenheim and Schafer, 1989] in the crossover region. The choice of step size and overlap regions were determined experimentally, being a tradeoff between speed and quality. If the step size is increased, there are fewer segments per unit time to process, but the overlap is also reduced, such that past frames cannot "seed" the present frame as well. For processing, the segment is zero-padded to 4096 samples such that we can use the Fast Fourier Transform (FFT) for efficient conversion to frequency domain. Once a segment has been processed, the signal is shifted by $a = 80$ samples and the next segment is processed. We denote the windows $w_f[n]$ and $w_w[n]$ for the fixed and work areas respectively, pointing out that the sum of the two windows is constant from the origin to sample 2208 before tapering off to zero.

For per-segment operation, Algorithm 2 is modified as follows. In the initialization, both of the envelope data arrays $\overline{\mathbf{c}}'$ (the fixed envelope areas) and $\tilde{\mathbf{c}}$ (the upper limit for envelopes) for the current segment are windowed with the sum of $w_f[n]$ and $w_w[n]$, and the carrier estimate from the previous segment is shifted and windowed with $w_f(n)$ to give the fixed estimate portion $\dot{\mathbf{c}}$, which in each iteration replaces the beginning of the current carrier estimate.

Note that we use two notations for the elements of $\mathbf{c}$: $\mathbf{c}[m, n]$ refers to the element of $\mathbf{c}$ for channel m at time index n, so $\mathbf{c}_{[k]} = \mathbf{c}[m, n]$ for $k = (m-1)M + n$ (see Eq. (A.9)). After the termination of the iterative loop for the current segment, the

Algorithm 4 Per-segment iterative reconstruction (single segment)

$i \leftarrow 0$;
$\dot{\mathbf{c}}[m,n] \leftarrow \mathbf{c}_{\text{prev}}[m, n+a]w_f[n], \qquad \forall m,n$ in the segment;
$\hat{\mathbf{c}}^{(0)}_{[k]} \leftarrow \overline{\mathbf{c}}'_{[k]}\mathring{r} \qquad k = 1,\ldots,(MN)$;
repeat
  $i \leftarrow i+1$;
  $\hat{\mathbf{x}}^{(i)} \leftarrow \mathbf{H}\hat{\mathbf{c}}^{(i-1)}$;
  $\mathbf{c}^{(i)} \leftarrow \mathbf{G}\hat{\mathbf{x}}^{(i)} + \mu\mathbf{r}$;
  $\mathbf{c}^{(i)}[m,n] \leftarrow \dot{\mathbf{c}}[m,n] + \mathbf{c}^{(i)}[m,n]w_w[n]$;
  $\hat{\mathbf{c}}^{(i)}_{[k]} \leftarrow \overline{\mathbf{c}}'_{[k]} \frac{\mathbf{c}^{(i)}_{[k]}}{|\mathbf{c}^{(i)}_{[k]}|} \qquad \forall k$ where $\overline{\mathbf{c}}'_{[k]} \neq 0$;
  $\hat{\mathbf{c}}^{(i)}_{[k]} \leftarrow \min(\tilde{\mathbf{c}}_{[k]}, \mathbf{c}^{(i)}_{[k]}) \frac{\hat{\mathbf{c}}^{(i)}_{[k]}}{|\hat{\mathbf{c}}^{(i)}_{[k]}|} \qquad \forall k$ where $\overline{\mathbf{c}}'_{[k]} = 0$;
until $D_M(\overline{\hat{\mathbf{c}}^{(i)}}, \overline{\mathbf{c}}^{(i)}) < \tau$ or $i = L$;
$\mathbf{c}_{\text{prev}} \leftarrow \hat{\mathbf{c}}^{(i)}$.

first a samples of $\hat{\mathbf{c}}^{(i)}[m,n]$ are considered "committed" and can be converted into the output signal $\mathbf{x}$ using overlap-add filtering. The implementation of the error measure D_M based is on the SER as described in the previous chapters and calculated on the work area only. For k such that $m = 1,\ldots,M(M = 65)$ and $n = 320,\ldots,2488$ (thus ignoring the overlap and runout),

$$D_M(\overline{\hat{\mathbf{c}}^{(i)}}, \overline{\mathbf{c}}^{(i)}) = \frac{\sum_k (|\hat{\mathbf{c}}^{(i)}_{[k]}|^c - |\mathbf{c}^{(i)}_{[k]}|^c)^2}{\sum_k |\hat{\mathbf{c}}^{(i)}_{[k]}|^{2c}}, \tag{5.10}$$

with $c = 0.4$. Below, we discuss the properties of this error measure with real signals, and the choice of τ for testing. The maximum number of iterations L was set to 20.

The zero-padding indicated in Fig. 5.4 serves a double purpose. First, zero-padding is needed to reduce error due to circular convolution. To avoid circular convolution issues completely, the zero-padding should be increased further, since the impulse response length of the lowest frequency filter is chosen to be 1800 samples. However, the impulse response length for filters decreases rapidly with increasing channel number and the resulting error was found to be negligible. The second purpose of the choice of an overall segment size of 4096 samples is to allow us to use the FFT to efficiently compute the steps in Algorithm 4 to find $\mathbf{c}^{(i)}$ from $\hat{\mathbf{c}}^{(i-1)}$. Since it is not necessary to

compute $\mathbf{x}$ explicitly until the loop is finished, we collapse the second and third step of the loop into $\mathbf{c}^{(i)} \leftarrow \mathbf{GH}\hat{\mathbf{c}}^{(i-1)}$, which can be implemented efficiently in frequency domain (see Appendix A).

With the implementation of the iterative reconstruction we find that the segmented processing of the signal not only allows for finite-delay implementation, but also that the perceived quality of the reconstructed audio is better than the implementation operating on the entire signal at once. This effect is observed in the RTISI algorithm as well [Zhu et al., 2007], where it is observed that the LSEE-MSTFTM algorithm needs a large number of iterations to find a good reconstructed signal since it is starting from a zero phase estimate. The RTISI algorithm working on short segments starts the iteration on each segment with a good estimate from the previous segment. Another interpretation is that the signal is constrained to be consistent between adjacent segments and this can be used to optimize the phase component of the STFT [Le Roux et al., 2008]. Similarly, each new segment being processed by Algorithm 4 is bootstrapped with the carrier from the previous frame through the fixed signal at the onset of the segment, so the carrier can be assumed to fit the envelopes already to some degree. As a result, the carriers in the current segment converge to fit their own set of envelopes more rapidly.

5.3 Evaluation of the model and its reconstruction algorithm

In this section, we present the evaluation of the system described above by computing the sparse envelope representation of various types of audio signal, in particular recordings of speech spoken by individual speakers and segments of music recordings. The source material for the speech recordings is the TSP speech database [Kabal, 2002] and the music recordings are taken from a database specifically designed for subjective audio quality evaluation, the "Sound Quality Assessment Material" (SQAM) database of the European Broadcasting Union (EBU) [EBU, 1988]. Table 5.1a lists the source file names and their length in time and samples (sampling rate of 16 kHz) as well as a description of the type of audio.

ID	Length		Description
CA02	3.86s	(61794 samples)	Speech, child
FA03	3.46s	(55297 samples)	Speech, female
FF32	3.21s	(51407 samples)	Speech, female
FI49	3.30s	(52880 samples)	Speech, female
MG41	2.91s	(46540 samples)	Speech, male
MH46	3.25s	(51973 samples)	Speech, male
MI50	3.56s	(56959 samples)	Speech, male
SQAM48	11.29s	(180578 samples)	Vocal quartet
SQAM60	13.46s	(215293 samples)	Piano concerto
SQAM70	9.98s	(159638 samples)	Country music

(a) Test Material for Evaluation

	envelope samples	r_I		
ID	(pre-masking)	0.8	1.0	1.5
CA02	144822	51516	34480	19569
FA03	130993	45623	30327	17087
FF32	120938	41196	28002	16119
FI49	125866	40255	26968	14906
MG41	111926	40602	27598	15977
MH46	123114	37206	24871	13928
MI50	133565	40504	27492	15533
SQAM48	395378	188420	125715	69113
SQAM60	467726	144479	97327	58410
SQAM70	349660	186906	122658	65803

(b) Number of envelope samples for each file before and after sparsification with various impact factor settings.

	Speech Files (combined)	SQAM48	SQAM60	SQAM70
Samples, original	376850	180578	215293	159638
SAES	199738	125715	97327	122658
SAES/Sample	0.530	0.696	0.452	0.768
Bits/Sample, 6-bit	3.18	4.17	2.71	4.61

(c) Estimate of bit-per-sample rate of raw quantized SAER, using sparsification factor $r_I = 1.0$.

Table 5.1: Statistics of test files, before and after processing.

5.3.1 Subjective evaluation

We performed subjective evaluation based on the MUSHRA protocol, since it allows for rapid testing of multiple test conditions. The test subjects we recruited are graduate students mostly from the McGill Department of Electrical and Computer Engineering, though none are studying speech or audio coding specifically. Overall, we had 5 male and 5 female testers of normal hearing, with an age range of 24 to 35 years old.

All tests sessions were performed in a dedicated acoustically isolated room, with sound being reproduced directly from the computer running the test software over a pair of Beyer-Dynamic DT880 headphones. The software used to perform the tests is "WinTest", provided courtesy of the Groupe de Recherche sur la Parole et l'Audio (GRPA) at the University of Sherbrooke [GRPA, 2010].

The tests are designed to establish first and foremost if the reconstruction algorithm can synthesize a signal of acceptable quality from the envelope representation. We attempt to establish what losses in quality are incurred by the envelope representation itself and how adding the masking model for envelope sample sparsification affects the quality relative to the envelope representation without sparsification. The simple quantizer described above is tested to establish if the sparse envelope representation is robust to noise incurred by quantization. All tests were performed on reconstruction using the finite-delay implementation, except for one test which was performed to verify if the finite-delay implementation is of higher perceived quality than the implementation processing the entire file at once.

In all tests the processed signals were presented along with the original audio signal (the reference) and the lowpass filtered anchor as suggested by the MUSHRA protocol. The anchor is the reference signal passed through a lowpass filter limiting its bandwidth to 4 kHz. In typical MUSHRA tests, this should provide a universally "bad" reference in comparison to full-band audio test signals at 41 or 48 kHz sampling rate. We found that many subjects preferred the anchor to the processed signals in some tests, which can in part be explained by the fact that since the reference has a bandwidth of only 7 kHz the lowpass anchor is still quite similar to the reference. The score for the reference is not shown in the following plots, since in most cases the

subjects had no difficulty identifying it.[1]

Comparison to commercially available codecs was not included in our testing. Our method is still at an experimental stage, making it difficult to get a fair estimate of a bitrate with which the SAER could be encoded. Furthermore, commercial codecs are optimized to perform well for a variety of signals, whereas we show in the following sections that the SAER has significant quality variations depending on signal type. The test setup was designed to reveal these variations and aid us in understanding their causes.

The figures below show statistical analyses of the MUSHRA test scores without normalization. In a typical MUSHRA setup with full-band signals, the lowpass anchor is expected to receive the lowest score and is used to scale all other scores. Instead, we include the score and standard error for the anchor. In the figures, the error bars show the standard error of the mean (SEM). If present, the lines above the bars are used to point out statistical significance, calculated using one-way repeated measure (RM) ANOVA: "ns" indicates that there is no significant difference between the bars, a single star indicates significant difference with a p-value < 0.05, two stars $p < 0.01$ and three stars $p < 0.001$.

5.3.2 Reconstruction from envelopes with and without masking model

The first experiment is to test the overall reconstruction method without quantization at varying levels of the impact factor r_I. Table 5.1b shows the number of sparse envelope samples in the representation for the tested settings of $r_I = 0.8, 1.0$ and 1.5. When comparing Table 5.1a to Table 5.1b, note that applying the masking model in all cases except for SQAM48 and SQAM70 with $r_I = 0.8$ reduces the number of samples when compared to the original (PCM coded) file.

Figure 5.5 shows the overall summary results of reconstructing signals from the envelopes after filtering and after applying the transmultiplexer based masking model with impact factor settings $r_I = 0.8, 1.0$ and 1.5. The drop in perceived quality is very visible (and the difference in scores statistically significant) as the impact factor is increased. However, the most notable feature is the apparent *increase* of

[1] The MUSHRA protocol asks that subjects assign a score of 100 to the item they think is the reference.

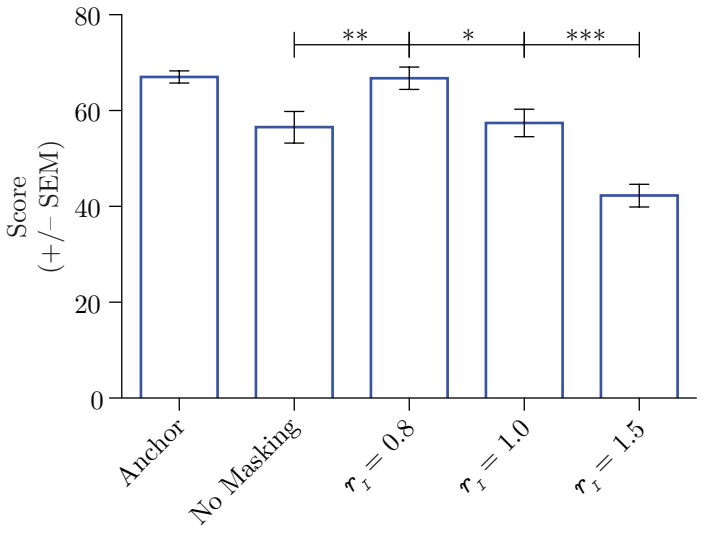


Fig. 5.5: Subjective testing scores evaluating reconstruction using different values of the impact factor r_I. The item labelled "No Masking" is audio reconstructed from the low-pass filtered envelopes without sparsification. The line over the bars shows statistical significance between conditions, with * indicating $p < 0.05$, ** indicating $p < 0.01$, and *** indicating $p < 0.001$.

perceived quality from the reconstruction without sparsification ("No Masking") to the reconstruction from the sparse envelope representation with impact factor $r_I = 0.8$. This result is contrary to expectation since even at that level, the representation is sparsified significantly.

We hypothesize that the low score for the reconstruction using the low-pass filtered envelopes introduces artifacts into the signal since it is a very redundant representation and the iterative algorithm forces the signal onto a set of envelopes that cannot be obtained from a realizable signal (see Sec. 4.1). For example, in Fig. 4.8 and Fig. 4.11, the differences between the low-pass filtered envelopes and the actual Hilbert envelopes are visible in some areas and the algorithm will modify the signal in every iteration to try to match the filtered envelopes.

To gain a further understanding of the issue, we conducted an additional test comparing the reconstruction from low-pass filtered envelopes to reconstruction from the original envelopes without filtering, the results of which are shown in Fig. 5.6. It can clearly be seen that the reconstruction from filtered envelopes is of lower perceived

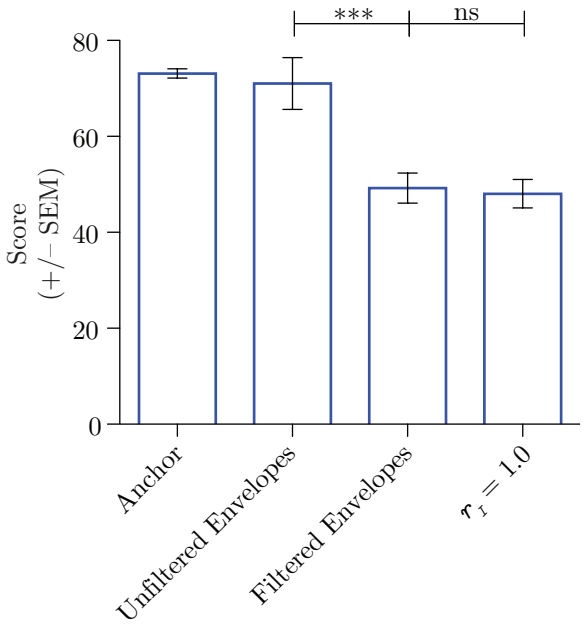


Fig. 5.6: Scores evaluating reconstruction with and without filtering of envelopes. The label "ns" indicates that there is no statistical difference between the results for those two conditions.

quality than the reconstruction from unfiltered envelopes and is effectively equivalent to the reconstruction from the sparsified SAER with $r_I = 1.0$. The reconstruction from unfiltered envelopes instead is judged to be as good as the anchor and if we compare this result to Fig. 5.5, this also implies it is effectively equivalent to the reconstruction from $r_I = 0.8$. This observation leads us to conclude that forcing the signal to fit the lowpass filtered envelopes does introduce artifacts even though the filtering itself does not remove perceptually relevant information. If perceptually important information is removed by the filtering, the reconstruction from $r_I = 0.8$ would not score higher than the smoothed envelope signal from which the sparse representation is derived.

The reduction in artifacts can be explained by the fact that when reconstructing from the sparse representation, the subchannel envelopes of the estimate signal are not limited in frequency in the iterative reconstruction loop. The subchannel signals $\mathbf{c}$ are allowed to have a magnitude less than $\tilde{\mathbf{c}}$ unless forced by $\hat{\mathbf{c}}$. Thus, the high-

frequency content of the envelopes of the reconstructed signal may be different than those of the original signal.

5.3.3 Quantization effects

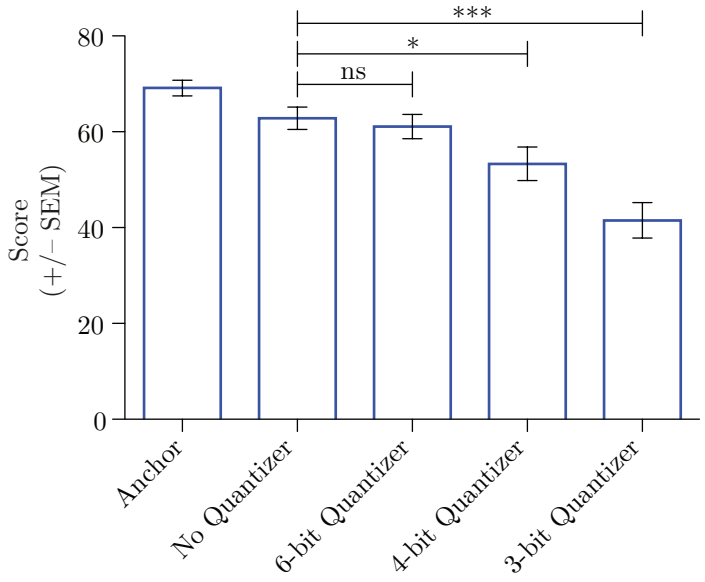


Fig. 5.7: Scores evaluating reconstruction with different quantizers, using a masking model with $r_I = 1.0$.

We test the robustness of the sparse envelope representation to quantization using a uniform scalar quantizer on the log-domain envelope sample values. We tested a 6-bit, 4-bit and 3-bit quantizer (64, 16 and 8 levels respectively). As shown in Fig. 5.7, the degredation in perceptual quality of the 6-bit quantizer was found to be statistically insignificant when compared to reconstruction from the unquantized sparse envelope representation.

Given the subjective evaluation scores, we can calculate a rough estimate of the rate a sparse auditory envelope based coding needs prior to entropy (lossless) coding. Combining the original signal lengths from Table 5.1a with the number of samples in the sparse representation and assuming a 6-bit quantizer, we first calculate the average number of sparse auditory envelope samples (SAES) versus the number of samples in the original sound samples.

Depending on the type of signal, it seems that at the sampling rate of 16 kHz, the sparse auditory envelope representation can be encoded with about 3–4 bits per sample as shown in Table 5.1c, which translates to 48–64 kB/s. Combined with the performance in subjective testing, we conclude that to be used as a viable coding method, the SAER needs to be extended to improve overall sound quality. In the following section we examine some cases where quality is low, resulting in an overall lower score.

5.3.4 Dependence on signal type

During evaluation, it became quickly apparent that the quality of the reconstruction algorithm is highly dependent on the signal content. The problem with highly tonal signals is addressed in Chapter 4 and thus we expected that music samples would perform significantly worse than speech samples. This was confirmed only for two of the music items; surprisingly, the third music item performed very well, in fact better than the speech items.

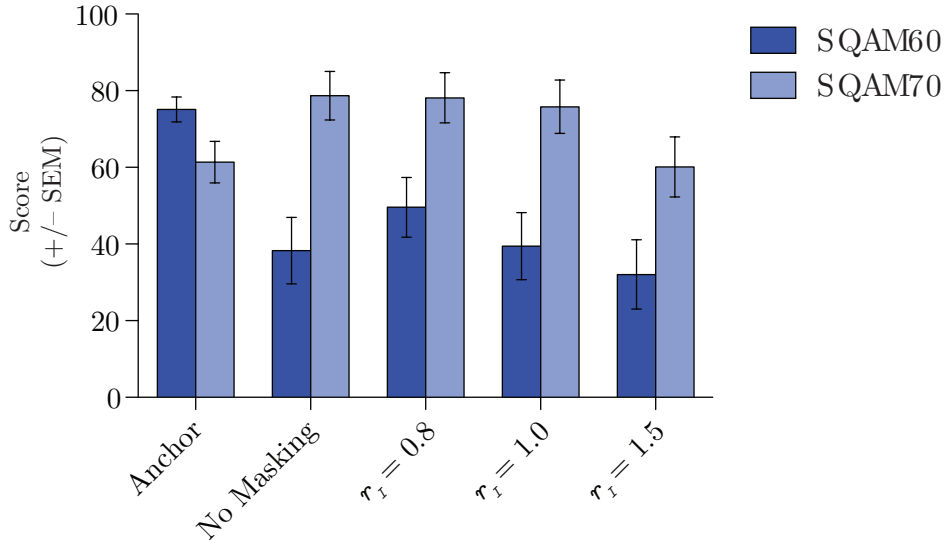


Fig. 5.8: Scores evaluating reconstruction for the two signals with lowest and highest scores.

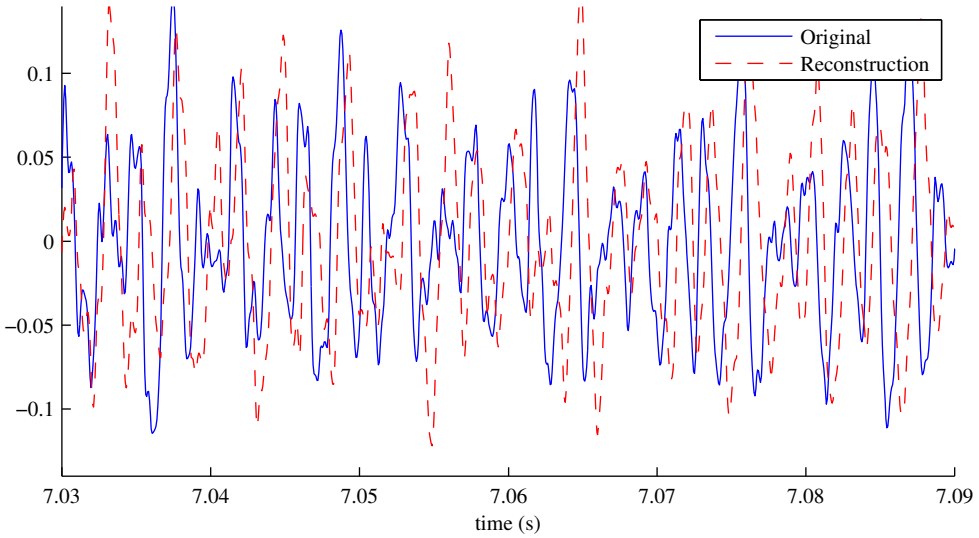
Figure 5.8 shows the two samples which listeners scored consistently with the low-

est and highest scores compared to other samples processed with the same masking settings. SQAM60 is a solo piano piece, where chords are sustained in several instances; pitch distortion was apparent to all listeners, who judged these distortions to be very detrimental to quality. On the other hand, SQAM70 is a piece of country/folk music with a strong rhythm component which the iterative reconstruction algorithm seems to be able to reproduce very well. In several instances, the listeners could not distinguish the reconstructed sound from the reference. Thus the model and reconstruction algorithm achieved *perceptually transparent* quality.

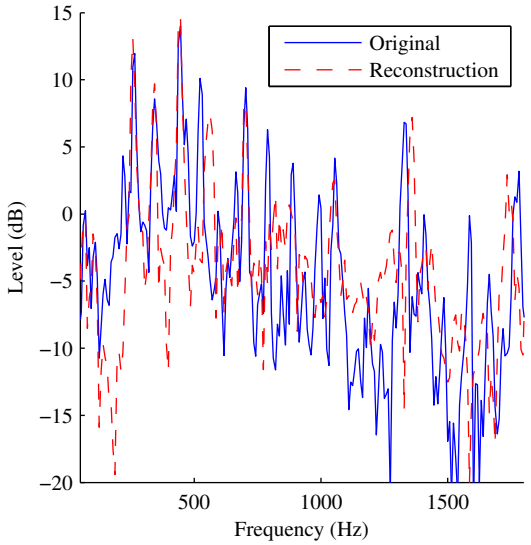
To illustrate how the tonal quality of a signal affects the reconstruction quality we examine some typical segments from the audio signals mentioned above. We begin with a section of SQAM60, showing the time-domain signal and its reconstruction in Fig. 5.9a. The periodicity of the signal is apparent and while the reconstruction does not match the original exactly, a phase-shift is expected but *should* be inaudible. However, if we look at the frequency domain using a STFT in Fig. 5.9b (using a 2048 sample segment of the audio with Hanning window), we note some differences more clearly. In particular, while some frequency peaks are matched quite well, some are shifted (eg. at around 550 Hz, 1300 Hz and 1700 Hz). This particular section is a piano chord and we can assume that the peaks represent the harmonics of the individual notes in the chord. The shifts in frequency of these harmonics are quite noticeable even with casual listening as they play a significant role in pitch detection [Bernstein and Oxenham, 2008; Moore and Glasberg, 2010], producing an sensation of an unsteady "shimmering" pitch.

These peak shifts occur because the reconstruction loop has only limited information about the exact frequency of the harmonics, as shown in Fig. 5.9c. Using block-transforms such as the STFT, the frequency resolution can be increased by using larger blocks, but the frequency-domain shape of the auditory channels is dictated by the BM model, so the sensitivity to signal frequency of each channel is fixed. The "envelope spectrum" in Fig. 5.9c would still only show broad peaks even with more channels, so the pitch of tonal components must be encoded in a more direct way for accurate reproduction. This is a powerful argument for the need to transmit some of the carrier information to specify the frequency of sharp peaks in the spectrum.

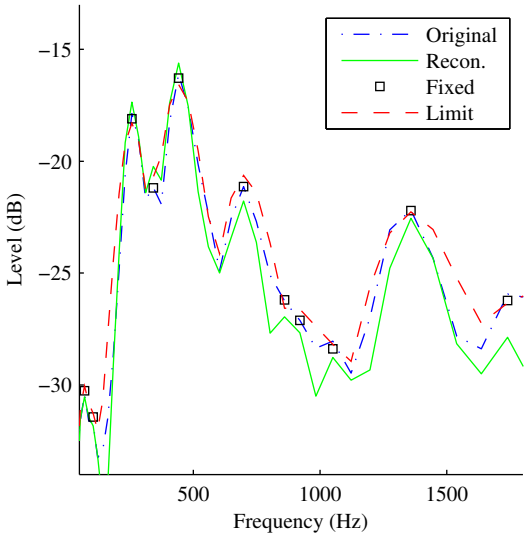
In Fig. 5.10, we examine a segment of another sample, SQAM70, which also has



(a) Short section of SQAM60



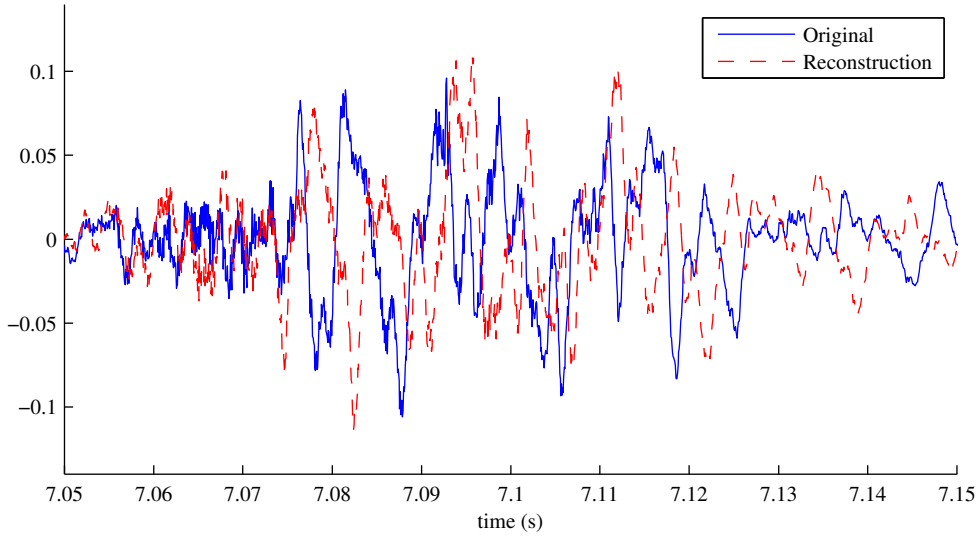
(b) STFTM of section



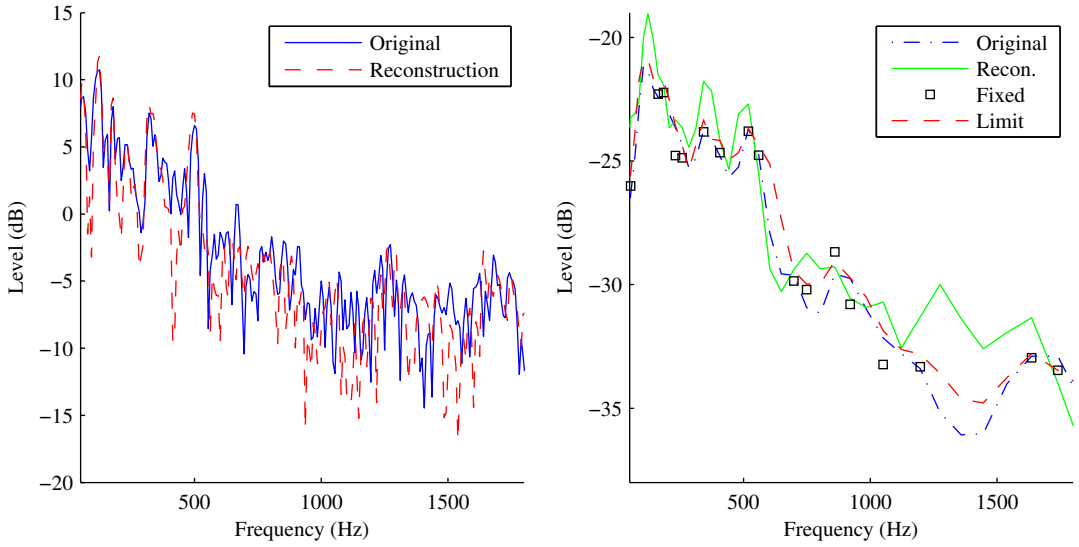
(c) Auditory channel envelopes at t=7.06s

Fig. 5.9: Detailed view of a short section of SQAM60

some pitched components, but is reproduced with little or no audible artifacts. Again, we note some significant phase-shifts in the time-domain signal, but in the frequency domain plot (using 2048 samples as above) of Fig. 5.10b, both the overall shape and



(a) Short section of SQAM70



(b) STFTM of section

(c) Auditory channel envelopes at t=7.09s

Fig. 5.10: Detailed view of a short section of SQAM70

the peaks in the lower frequency portion are matched remarkably well. As in the sound segment described previously, the envelope spectrum does not capture the peaks as well, although the overall spectral shape is similar. If we look back at the time domain

signal, we note that the pitched portion appears to be fairly short and in the STFT spectrum this results in peaks that are more broad than in the previous signal.

Overall we find that the SAER reproduces temporal features of a signal very well at the expense of spectral features. Informal experiments showed that increasing the number of auditory channels increases the quality of reproduced audio only marginally at the cost of increasing the number of samples in the representation and computational complexity. However, modifying the algorithm by adding some information from the original carrier signals (by initially "seeding" the iterative loop with the original carrier) does result in a marked improvement especially for the SQAM60 signal. From the bitrate estimates in Table 5.1c, the SAER of the tonal signal SQAM60 is using almost 2 bits per sample fewer than SQAM70, which would be the budget for the phase information. We expect a modified model accounting for phase could achieve near-transparent quality using those extra bits.

5.3.5 Dependence on reconstruction method

As described at the beginning of this section, all tested signal were reconstructed using the finite-delay implementation as described earlier in this chapter. The preference for the finite-delay processing was established by running a test comparing the reconstruction at $r_I = 1.0$ either by whole-file processing or finite-delay processing. The resulting mean score (combining all test files) for the whole-file processing was 39 ($\pm$3 SEM) versus 58 ($\pm$4 SEM) for finite-delay processing, showing a statistically significant difference with $p < 0.001$.

5.3.6 Objective evaluation based on envelopes

We now compare the auditory envelopes of the reconstructed audio samples to the auditory envelopes of the original sound files. The difference between the envelopes is calculated similar to the SER as in Chapter 3,

$$\mathrm{SegSER}(\mathbf{c}_{\mathrm{orig}}, \mathbf{c}_{\mathrm{recon}}) = 10 \log_{10} \left(\frac{\| \, |\mathbf{c}_{\mathrm{orig}}|^c \, \|^2}{\| \, |\mathbf{c}_{\mathrm{orig}}|^c - |\mathbf{c}_{\mathrm{recon}}|^c \, \|^2} \right), \tag{5.11}$$

with companding $c = 0.4$ to allow comparison to the internal measure described below. We use the same segment size and advance as the reconstruction algorithm

and average the per-segment dB values over each file. The results are presented in two tables. Table 5.2a shows the envelope difference between the reconstruction and the original using the plain envelopes, while for Table 5.2 the envelopes for both reconstruction and original were lowpass filtered prior to calculating the Segmental SER (SegSER).

As might be expected, the reconstruction from the auditory envelopes without any filtering or sparsification (labelled "Plain Envelopes" in the tables and figures) matches the original signal envelopes best. As the modifications to the envelopes are increased (by filtering, sparsification, and quantization), the quality decreases. The quantization effect in particular shows that while the 6-bit quantizer is effectively transparent, the quality decreases significantly as the number of bits is reduced below that value. On the other hand, if we look at some detailed values we notice some striking differences compared to the subjective results. For example, while in subjective testing the SQAM60 sample scored consistently very low (see Fig. 5.8), the envelopes of the reconstruction for this audio sample actually match the original sample envelopes more closely than we observe for other samples. Conversely, the SQAM70 sample in many cases has a lower average SegSER even though listeners had difficulties distinguishing the reproduction from the original at $r_I \leq 1.0$. The decrease in SegSER that we observe from masking with $r_I = 0.8$ to $r_I = 1.5$ (increasing sparsification) is also less than we would expect given the subjective scores. Finally, completely missing from the objective data is the increase in quality that we observed from the reconstruction from filtered envelopes to using the masking model at $r_I = 0.8$. As described in Section 5.3.2 above, we attribute the discrepancy to the fact that the non-sparsified envelope representation forces the reconstructed signal into a set of low-pass envelopes. If the representation is sparse, the reconstruction algorithm can introduce high frequency envelope modulations. While these envelope modulations reduce audible artifacts, they do not necessarily match the original envelopes.

This means that if given two signals whose auditory envelopes are very similar, the audible difference between them can still be very significant. In other words, the sparse auditory envelope representation *in general* is not sufficient to fully describe an audio signal, nor can the error between two sets of envelopes capture subtle audible perceptual differences. However, we also observe that this only applies to specific

	Envelopes		Masking			Quantizer ($r_I = 1.0$)		
Sample	Plain	Filtered	$r_I = 0.8$	$r_I = 1.0$	$r_I = 1.5$	6-bit	4-bit	3-bit
CA02	21.13	16.39	14.47	14.14	13.98	14.11	13.44	12.35
FA03	21.47	16.71	14.88	14.67	14.51	14.55	13.94	12.59
FF32	21.40	15.68	13.65	13.42	13.33	13.34	12.86	11.72
FI49	21.21	15.35	13.69	13.37	13.23	13.29	12.84	11.93
MG41	21.54	15.35	13.49	13.24	13.01	13.23	12.83	11.59
MH46	21.13	15.19	13.54	13.42	13.19	13.38	12.76	12.02
MI50	21.66	15.53	13.22	12.91	12.79	12.94	12.50	11.43
SQAM48	20.73	15.55	14.34	14.15	13.87	14.12	13.72	12.81
SQAM60	20.96	16.18	14.68	14.53	14.36	14.46	14.08	13.01
SQAM70	21.27	16.11	14.27	14.06	13.96	13.98	13.63	12.53
Average	21.25	15.80	14.02	13.79	13.62	13.74	13.26	12.20

(a) Average SegSER between original envelopes and reconstructed envelopes

	Envelopes		Masking			Quantizer ($r_I = 1.0$)		
Sample	Plain	Filtered	$r_I = 0.8$	$r_I = 1.0$	$r_I = 1.5$	6-bit	4-bit	3-bit
CA02	26.02	22.98	17.69	17.22	16.64	17.11	16.19	14.40
FA03	26.35	23.30	18.21	17.78	17.25	17.66	16.69	14.47
FF32	25.95	21.35	16.46	16.14	15.61	16.07	15.29	13.40
FI49	26.25	21.86	17.11	16.64	15.88	16.58	15.70	14.13
MG41	26.90	22.20	17.08	16.68	15.90	16.59	15.93	13.82
MH46	26.33	21.56	17.14	16.70	16.00	16.62	15.75	14.29
MI50	26.95	22.68	16.37	16.07	15.40	16.06	15.24	13.48
SQAM48	25.31	22.25	19.58	19.12	17.74	19.06	18.24	16.24
SQAM60	24.80	22.77	19.27	18.98	17.92	18.93	18.06	15.88
SQAM70	25.99	22.44	18.49	18.03	17.57	17.93	17.21	15.14
Average	26.08	22.34	17.74	17.34	16.59	17.26	16.43	14.52

(b) Average SegSER between original filtered envelopes and reconstructed filtered envelopes

Table 5.2: Average SegSER (in dB) for envelopes of reconstructed audio files

types of signal, since some signals are reproduced very well from this representation alone. This reinforces the observation that while the envelopes capture a significant part of the audible information in a signal, the differences in the carrier information can be very audible in some cases.

5.3.7 Sparsity and D_M

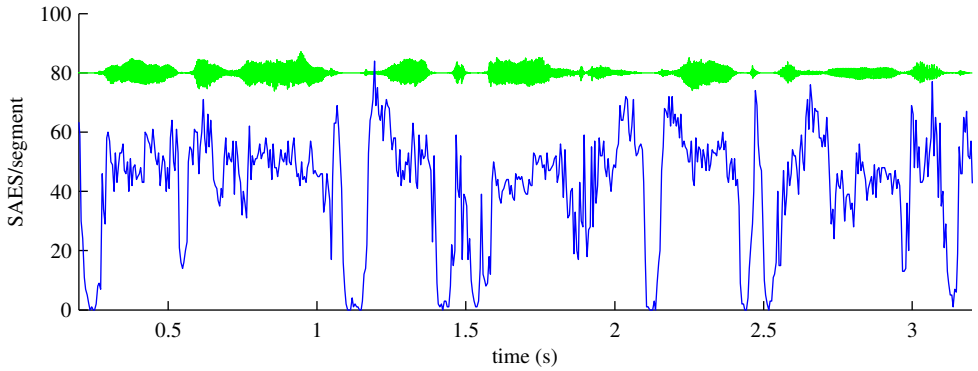
We now examine the distance measure that the iterative loop is minimizing and how it serves as a measure of convergence and perceptual quality. The value of D_M is calculated per segment by Eq. (5.10), then converted to dB using $\text{SegSER}_{\text{int}} = -10\log(D_M)$ (so $\hat{\mathbf{c}}$, the "best estimate" of the envelopes within the iterative loop takes the place of $\mathbf{c}_{\text{orig}}$ in Eq. (5.11)).

	Envelopes		Masking			Quantizer ($r_I = 1.0$)		
Sample	Plain	Filtered	$r_I = 0.8$	$r_I = 1.0$	$r_I = 1.5$	6-bit	4-bit	3-bit
CA02	19.63	18.86	22.08	25.20	27.65	25.20	25.01	24.29
FA03	19.38	18.80	22.25	25.54	27.77	25.50	25.28	24.68
FF32	19.56	17.97	21.64	24.72	27.61	24.68	24.48	23.87
FI49	19.45	18.16	21.25	24.33	27.51	24.30	24.10	23.56
MG41	19.21	18.11	21.22	24.46	27.47	24.44	24.22	23.57
MH46	19.34	17.97	21.14	24.39	27.48	24.41	24.15	23.48
MI50	19.25	18.01	21.16	24.25	27.48	24.16	23.96	23.35
SQAM48	19.09	18.52	20.91	24.03	27.47	24.02	23.80	23.19
SQAM60	19.30	18.72	21.33	24.60	27.63	24.56	24.36	23.79
SQAM70	19.27	18.70	20.78	23.94	27.35	23.90	23.66	23.04
Average	19.35	18.38	21.38	24.55	27.54	24.52	24.30	23.68

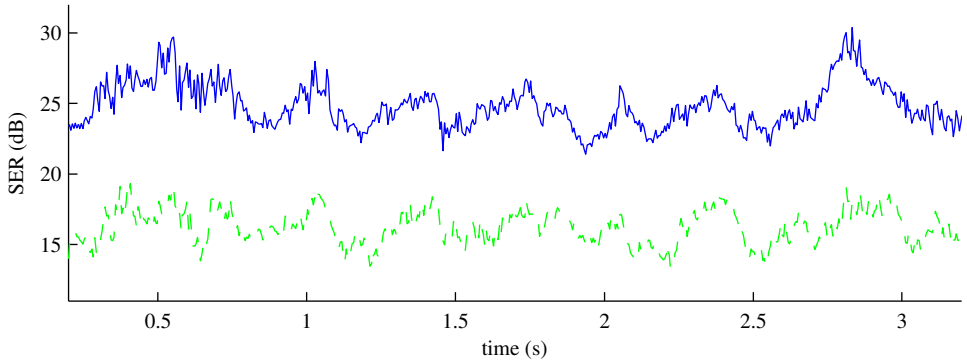
Table 5.3: Internal SegSER for sound samples at different parameters.

The average internal $\text{SegSER}_{\text{int}}$ for all test files at the various values for r_I and quantizer granularity are shown in Table 5.3. The loop termination value is set to 27 dB. Most notable is the fact that for any given sound signal, the average $\text{SegSER}_{\text{int}}$ *increases* (decrease of D_M) as the representation becomes more sparse. However, as a sparsified representation reduces the constraints on the auditory envelopes, it is expected that it becomes easier for the iterative procedure to match the reconstruction to the target envelopes. At $r_I = 1.5$, the iterative loop to find the carrier signals

is terminated before 20 iterations for almost all segments, resulting in an average $\text{SegSER}_{\text{int}}$ greater than 27 dB. This contributes to the decrease in quality for sparse representations, since fewer iterations are performed that would otherwise find a better match for the carrier signals.



(a) SAES per segment for FF32



(b) D_M per segment for FF32

Fig. 5.11: Number of SAES per segment and $\text{SegSER}_{\text{int}}$ for FF32

To show the variation of the $\text{SegSER}_{\text{int}}$ between segments and the relationship with sparsity, Fig. 5.11a shows one of the speech signals and the number of sparse auditory envelope samples (SAES) per 80-sample (5 ms) segment and Fig. 5.11b shows the per-segment $\text{SegSER}_{\text{int}}$ during reconstruction. For this figure, the terminating condition was disabled: the solid line shows the $\text{SegSER}_{\text{int}}$ after the full 20 iterations and the dashed line shows the $\text{SegSER}_{\text{int}}$ after the first 3 iterations. It is apparent that the final $\text{SegSER}_{\text{int}}$ is very variable, depending both on the properties of the signal being

processed and the number of envelope samples within the segment. For example, at 2.7 s, the number of envelope samples is around 40 samples/segment, while the $\text{SegSER}_{\text{int}}$ increases to almost 30 dB.

We conclude that while D_M is a good indication of how well the reconstructed signal matches the encoded representation, the degree of sparsification needs to be taken into account to make the measure more uniform over varying signal types. The addition of phase information to the representation would complicate this metric further and should be a focus for future research.

5.3.8 Computational complexity of the implementation

The implementation of the perceptual model and the iterative method to reconstruct an audio signal from the SAER was primarily designed to show the feasibility of such a representation for audio coding. Thus, no concerted effort was made to optimize computational performance beyond the need to allow for testing of the implementation in a reasonable timeframe. In particular, the model and reconstruction were implemented in a high-level modelling language (MATLAB®), with structuring of the code such that it could be easily understood, tested, and modified. This results in inefficiencies. For example, special cases can be handled to avoid computing unused data, but a regular structure was used to facilitate experimentation. In particular, the use of a fully oversampled filterbank causes a significant amount of memory to be used for storing the auditory channel signals prior to subsampling and sparsification, with the transmultiplexed envelope patterns also stored and processed at full rate. This allowed for testing different subsampling rates and filterbank configurations, but prevents us from making a fair estimate of achievable complexity.

The reference implementation (Intel® CPU at 2.83 GHz with 8 GB of RAM) required 27 minutes (12.5 minutes for the computation of the SAER and 14.5 minutes for reconstructing the audio signal) to process the SQAM70 audio sample (10 s long). It is noteworthy that in comparison to typical codec designs, the computational load at the decoder exceeds the load at the encoder, due to the iterative structure of the reconstruction algorithm. Further development should aim to reduce the complexity of the reconstruction algorithm in particular.

5.4 Summary and discussion

This chapter presents some of the details of implementing the method to turn an audio signal into the sparse auditory envelope representation and the iterative inversion to get a signal back from the representation. We first detail the design of the analysis and synthesis filterbank, where special attention is paid to ensure that the response is as flat as possible over the desired frequency range. Low-pass filtering and subsampling of the envelopes is described, followed by the transmultiplexer based masking model, which sparsifies the envelope representation by first creating a sorted list of envelope samples then removes samples from the list based on a masking threshold decision.

The reconstruction of the audio signal from the sparse auditory envelope representation is achieved in two steps. The first step expands the sparse representation to a pair of specifications: points where the reconstructed auditory envelopes must match specified values, and points where the reconstructed auditory envelopes must be within a defined limit. In the next step of the reconstruction algorithm, a set of carrier signals is estimated using iterative refinement to match the constraints.

The implementation of the encoding and reconstruction is tested by processing a set of speech and audio samples and verifying the incurred distortion using both subjective and objective testing. Subjective tests reveal that the quality of the reconstructed audio signals varies greatly with signal type, but the objective comparison of the model parameters shows only little difference; this implies that the sparse auditory envelope representation is insufficient for general audio signals. On the other hand, since some types of signals are reproduced very well, the information present in the carrier is often inaudible.

Sparse auditory envelope samples are therefore a representation of audio signals that can be used for audio coding if we can identify situations where the carrier is important and send the missing information in those specific cases. This seems to be coupled to instances where the sparsification algorithm can remove many of the auditory envelope samples, since tones display sharp peaks in the frequency domain. The amount of side information containing the carrier phase could then be coupled to the sparsification factor: if few samples were removed, it is probably not necessary to send extra information. The development of such an extended model is a topic for future research, but we point out here that the amount of phase information that

an extended model would require is probably fairly low. In the case of the STFT-based algorithms, use of a single bit per transform coefficient has been shown to allow for high-quality reconstruction [Alsteris and Paliwal, 2007]. Our own initial informal experiments showed a marked improvement with a similar addition of the carrier information.

Chapter 6

Conclusion

6.1 Summary of research

Perceptual models have been used in audio coding for several decades now, primarily to direct the quantization stage of a frequency transform encoder. Recently, more direct methods have been proposed in which the bitstream of the coded audio is derived directly from the parameters that the perceptual model computes. However, to date the decoding stage of these perceptual domain codecs is designed with a constraint on computationally efficient reconstruction and thus the encoded bitstream typically includes more information than a strict perceptual analysis would consider relevant.

Considering the ever-increasing availability of powerful processing capabilities both in consumer desktops and even portable devices, we set out to investigate the possibility of using iterative methods at the decoding end of a perceptual domain codec. In particular, if the decoder is given a set of perceptual parameters sufficient to describe the audio signal but not to do a straightforward reconstruction, the decoder can estimate what the audio signal should be, then reanalyze the estimated signal. If the received (target) information is inconsistent with the analysis local at the decoder, the signal is refined and analyzed again. This cycle is repeated until the analysis of the local signal is a match to the target.

To test the feasibility and potential issues of such a decoding scheme, we design and implement a simplified auditory model based on auditory envelopes. Auditory

envelopes are a perceptual representation that models the processing of stimuli of the peripheral auditory system: parallel bandpass filtering by the Basilar membrane (splitting the original signal into auditory channel signals) then nonlinear processing and temporal smoothing by the neural transduction of the inner hair cells. This level of analysis has been used for perceptual domain coding, but to reconstruct a signal accurately, existing decoders need the information contained in the fine temporal structure of the auditory channels, encoded as impulses or a frequency-modulated carrier. This temporal information, either the time-difference between individual pulses or instantaneous frequency of a carrier, is difficult to encode at a low bitrate. It can vary rapidly, yet due to the overlap between auditory channels in frequency domain, small errors can cause interchannel interference during reconstruction. The literature on auditory perception suggests that the temporal information is critical to pitch perception [Zeng et al., 2004], yet present in the envelopes [Yang et al., 1992; Smith et al., 2002]. The iterative reconstruction reveals to what degree the temporal information can be synthesized from the envelopes alone.

We find that reconstruction from subband envelopes can be analyzed using a framework previously used to analyze reconstruction Short-Time Fourier Transform Magnitude (STFTM) data. Using a circulant matrix representation of linear FIR filters, the filtering and reconstruction can be analyzed using frame theory and treated similar to STFTM reconstruction methods, to examine the question of whether the algorithm converges. The subband envelope reconstruction is not an exact match to STFTM reconstruction and convergence cannot be guaranteed, but in general the reconstruction converges to a reasonable estimate.

In its initial form, a subband envelope representation is a highly redundant representation, even though some information has been discarded (the fine temporal information present in the carrier that underlies the envelopes). Based on models of auditory perception, the subband envelopes are low-pass filtered and sampled. Further reduction of the number of samples is achieved by processing the envelope samples using a masking model. This masking model is based on the assumption that the presence of a large amplitude auditory channel signal can inhibit the perception of signal energy nearby in time and frequency (that is, in the same or neighbouring auditory channels), accounting for temporal and simultaneous masking effects in the

human auditory system. The resulting set of envelope samples is the sparse auditory envelope representation (SAER) from which we reconstruct an estimate of the original audio signal.

Reconstruction of an audio signal from the SAER is a two-step process, somewhat mirroring the encoding process. However, neither step is a 'straight' inversion: the inverse of the sparsifying masking model must create a set of envelopes at the full sample rate based on the sparse envelope samples and the audio synthesis from envelopes must recreate the carrier signal in each auditory channel. The first part is non-iterative, creating two sets of envelope data. One set of envelope data is simply the received envelope values, expanded to account for sampling and the other set is an extrapolation of the fixed envelope data mirroring the masking model. In effect, if a particular envelope sample (at a sampling location) is missing in the representation, we know that the masking model at the encoder removed it by deciding that an envelope sample nearby inhibits its perception. Thus, the decoder must ensure that the auditory subchannel signal (of the reconstructed audio) at the location of that missing sample must not exceed the threshold that the encoder used during sparsification. Thus, the intermediate envelope data at the decoder consists of one set of envelopes where the reconstruction analysis must match (the "fixed" set) and one set of envelopes which the reconstruction analysis envelopes must not exceed (the "upper limit" set).

The second part of the decoder then is the synthesis of the audio signal from the two sets of envelopes. For each auditory channel the carrier information is assumed initially to be a random signal. The estimate of the auditory channel signals is synthesized into an audio signal, which is then reanalyzed using the same analysis filters as used by the encoder. The envelopes of the analysis are computed and compared with the envelope sets from the previous stage of the decoder. First the "fixed" sections are applied, then the remaining envelopes are compared to the "upper limit" set. Any portion of the locally analyzed envelopes that exceeds this set is adjusted to match. This loop is repeated until either no further adjustments are needed or the loop has been run a set number of times.

Iterative reconstruction of signals usually requires processing finite signals as a single unit since filtering with delay adjustment (needed to ensure the signal lines up

with itself at each iteration) involves non-causal processing of the signal. This prevents implementation of such algorithms for long signals such as audio streams. Thus a modified version of the reconstruction algorithm is implemented, which reconstructs the signal by short-time sections using overlap with adjacent sections to ensure a coherent overall signals. Also based on Short-Time Fourier Transform (STFTM) methods, this type of finite-delay processing actually results in faster convergence to acceptable signal quality. Due to the overlap, if one section has converged to match the target envelopes, its carrier estimates "seed" the carrier estimates of the following section, which will then find its local minimum faster. A good carrier estimate thus cascades along the reconstructed signal.

The implementation of the encoder of audio into the sparse envelope representation and the matching decoder is tested on a set of speech and audio files by subjective evaluation using the MUSHRA testing protocol. The result reveals that the SAER can be used to get adequate quality overall even with quantization of the representation at a modest bitrate reduction from the original signal. We note that the perceived quality is very dependent on the type of signal encoded, which we take as indication of a shortcoming of the SAER if signals contain pitched sounds.

6.2 Discussion, possible applications and future work

6.2.1 Summary of results

The work presented in this thesis examines the feasibility of the SAER for audio coding and the iterative method of reconstructing an audio signal from this representation. Subjective and objective testing leads us to the following conclusions.

Perceptual model based on envelopes

We find that for some types of audio signals, the SAER can be used to obtain a reconstructed signal with high perceptual quality. This means that the SAER contain a significant portion of the audible information and, in some cases, *all* of the audible information. However, a particularly problematic class of audio signals to reconstruct from envelopes alone are those that are strongly tonal in nature. A tonal signal is one that evokes a pitch sensation and we observe that while pitch directly affects

envelopes, this effect is very subtle in the magnitude of the envelopes. This means that it is very difficult to find the precise pitch of a given set of auditory channel envelopes.

Iterative perceptual model inversion

Although iterative methods have been used to reconstruct signals from magnitude-only transform representations in the past, this thesis presents a novel adaptation tailored to the sparse auditory envelope representation. By generating two sets of envelopes, the fixed envelopes and the upper limits, we precompute the masking effect of the recomputed envelopes in a noniterative way, thus requiring that the computationally expensive iterative loop produce only an estimate of the carrier signals. While the overall computational complexity is still very high, this simplification opens up the possibility of further computational optimizations.

Analysis of FIR filterbank using circulant matrices and frame theory

We analyze the iterative reconstruction from the envelopes of a subband decomposition using a circulant matrix representation of FIR filters. This allows for application of frame theory to FIR FB systems using a simple and compact notation which also lends itself to straightforward numerical analysis. However, there is a loss of generality in comparison to methods such as state-space analysis as this notation is applicable only to fully oversampled FIR FBs.

6.2.2 Criticisms and future work

The research in this thesis aimed to investigate the use of a purely envelope based perceptual representation for audio coding using iterative reconstruction, in part to explore some of the issues that arise with iterative reconstruction from perceptually coded audio. In this respect, we only had qualified success, showing that the SAER is not sufficient for audio coding in general. Iterative decoding of perceptual representations shows promise if one can afford the computational complexity, but care must be taken that the model parameters define the signal well enough.

Thus, future investigations of envelope-based sparse representations should focus

on the phase component, to establish how the input signal can be analyzed to determine if it is necessary to encode part of the carrier signal. Like the envelopes, the phase information would be sparse since our tests show the phase is not required for all signal types. This also means that a bimodal set of constraints could be constructed at the decoder analogous to the fixed and maximum envelope, either forcing the carrier information or constraining it within some limits based on audibility. However, designing such a scheme requires further study of the human auditory system, as phase perception is not as well understood as envelope perception.

A significant effort to understand the role of the fine temporal structure that the phase represents has been in the context of binaural hearing [Moore, 2003]. Extending the SAER and its reconstruction algorithm to stereo audio signals can be an avenue of research to understand the coding aspect of auditory phase. The envelope and phase decomposition should translate well to inter-aural level and time differences as used in [Baumgarte and Faller, 2003; Faller and Baumgarte, 2003].

There are also refinements to the computation of the auditory envelope that can be considered. In this thesis, we used a gammatone FB as model for the BM movement, but recent refinements of BM modeling include multi-path implementations combining a linear and nonlinear branch [Lopez-Poveda and Meddis, 2001], the gammachirp function [Irino and Patterson, 1997; Unoki et al., 2006], and compressive versions of the gammachirp [Irino and Patterson, 2006]. A more accurate modeling of the neural response is expected to yield better sparsification of the envelope samples.

From a more practical standpoint, there are several implementation aspects of the reconstruction algorithm that should be addressed in more detail in future research. Especially for the finite-delay reconstruction algorithm, there are many parameters that can influence the reconstructed signal quality and computational complexity, such as the segment size, the amount of overlap between segments, the loop terminating condition, and noise injection factor. The optimal values for these parameters are likely to depend on the auditory representation and if a new representation is designed a more thorough investigation of these parameters should be performed.

In terms of computational complexity, we note that the implementation as presented in this thesis is far from optimal and one particular problem is that all subchannel signals are processed at the original signal sampling rate. This was done to ease

analysis and implementation as well as retaining flexibility to experiment with various envelope subsampling methods. Once this flexibility is no longer needed, computational efficiency could be increased by performing the analysis/synthesis and envelope correction steps at a lower rate given by the subsampling factor.

6.3 Final remarks

Perceptual audio coding is an area of research that touches on a variety of disciplines from anatomy, biology, and physiology to the mathematics of signal processing and information theory. In this thesis, we look only at one aspect of the broad field: how a particular kind of representation can be used for encoding audio signals and how we can reconstruct a signal from this representation. We believe that this new perceptual representation and the insights gained during its development will serve as a base for future research in auditory perception and audio coding.

Appendix A

Eigenvalues and Eigenvectors of Circulant Matrices

Assume the matrix $\mathbf{C}$ is of size $N \times N$ and has eigenvectors forming the matrix

$$\mathbf{W}_{[m,n]} = \frac{1}{\sqrt{N}} e^{-2\pi j \frac{(m-1)(n-1)}{N}} \tag{A.1}$$

and eigenvalues λ_n, so we can write the eigendecomposition

$$\mathbf{C} = \mathbf{W}^H \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_N \end{bmatrix} \mathbf{W}. \tag{A.2}$$

Writing out this equation per element of $\mathbf{C}$, we get

$$\begin{aligned} \mathbf{C}_{[m,n]} &= \sum_{k=1}^{N} \lambda_k \mathbf{W}^*_{[m,k]} \mathbf{W}_{[n,k]} && \text{(A.3)} \\ &= \frac{1}{N} \sum_{k=1}^{N} \lambda_k e^{2\pi j \frac{(m-1)(k-1)}{N}} e^{-2\pi j \frac{(n-1)(k-1)}{N}} && \text{(A.4)} \\ &= \frac{1}{N} \sum_{k=1}^{N} \lambda_k e^{2\pi j \frac{(k-1)(m-n)}{N}}. && \text{(A.5)} \end{aligned}$$

By the circular property of the complex exponential, for a given set of λ_n this value depends only on $((m-n) \bmod N)$, so

$$\mathbf{C}_{[m,n]} = \mathbf{c}_{[(m-n) \bmod N]} = \frac{1}{N} \sum_{k=1}^{N} \lambda_k e^{2\pi j \frac{(k-1)(m-n)}{N}}, \tag{A.6}$$

meaning $\mathbf{C}$ is a circulant matrix and any circulant matrix can be decomposed into the form of Eq. (A.2). If all λ_k are nonzero, this is a valid eigenvalue decomposition. Note that λ_k is simply the kth (scaled) discrete Fourier transform coefficient of $\mathbf{c}$.

Filter implementations using circulant matrices

The fact that circulant matrices can be diagonalized using the Fourier transform leads directly to the overlap-add and overlap-save techniques of FIR filter implementations [Proakis and Manolakis, 1996]. So, if $\mathbf{A}$ is a circulant matrix representing a filter with impulse response $\mathbf{a}$, the filter operation $\mathbf{y} = \mathbf{A}\mathbf{x}$ becomes

$$\mathbf{y} = \mathbf{A}\mathbf{x} \tag{A.7}$$

$$= \mathbf{W}^H \text{diag}\{\mathcal{F}\mathbf{a}\}\mathbf{W}\frac{1}{\sqrt{N}}\mathbf{W}^H\mathcal{F}\mathbf{x} \tag{A.8}$$

$$= \frac{1}{\sqrt{N}}\mathbf{W}^H(\text{diag}\{\mathcal{F}\mathbf{a}\}\mathcal{F}\mathbf{x}), \tag{A.9}$$

which can be computed very efficiently if the DFT is computed using the Fast Fourier Transform (FFT). Note that $\text{diag}\{\mathcal{F}\mathbf{a}\}\mathcal{F}\mathbf{x}$ is simply an element-by-element multiplication of $\mathcal{F}\mathbf{a}$ and $\mathcal{F}\mathbf{x}$ and that for a fixed filter $\mathcal{F}\mathbf{a}$ is a constant vector (the FFT of the filter response) that can be precomputed.

Frequency-domain implementation of the Transmultiplexer

The (single) filter implementation in frequency domain can be extended to the efficient calculation of the transmultiplexing step in Chapter 5. Recall that the column vector $\mathbf{c}$ of size MN (subdivided into M vectors $\mathbf{c}_m$) is passed through the synthesis filter, represented by $\mathbf{H}$ which is a horizontal concatenation of M circulant matrices $\mathbf{H}_m$ of size $N \times N$. The result of this synthesis is then passed though the analysis filter

described by $\mathbf{G}$, which is a horizontal concatenation of M circulant matrices of size $N \times N$. Using $\mathbf{d}$ similar to $\mathbf{c}$ for the transmultiplexed result, the entire operation can be written as

$$\mathbf{d} = \mathbf{GHc}, \tag{A.10}$$

or per subvector/submatrix,

$$\mathbf{d}_m = \mathbf{G}_m \Big(\sum_{m=1}^{M} \mathbf{H}_m \mathbf{c}_m \Big), \tag{A.11}$$

noting that $\sum_{m=1}^{M} \mathbf{H}_m \mathbf{c}_m$ is simply the intermediate signal, a column vector of size N.

For unstructured matrices, this computation would be very costly in terms of operations. Generating the intermediate signal is a multiplication of an $N \times N$ matrix by an N dimensional vector, an operation of order $O(N^2)$, repeated M times. The analysis to find $\mathbf{d}$ is of the same complexity, thus the transmultiplexer requires $2M$ times $O(N^2)$ operations. However, by transforming the filtering into frequency domain, the $O(N^2)$ operation can be reduced to $O(N)$. The DFT approach is given by

$$\mathcal{F}\mathbf{d}_m = \mathrm{diag}\{\mathcal{F}\mathbf{h}_m\} \Big(\sum_{m=1}^{M} \mathrm{diag}\{\mathcal{F}\mathbf{g}_m\} \mathcal{F}\mathbf{c}_m \Big). \tag{A.12}$$

While this requires the conversion of all subchannel signals $\mathbf{c}_m$ into frequency domain and the inverse transform to get the time domain $\mathbf{d}_m$, using the FFT makes this far more efficient than direct convolution of the individual filters.

References

Aarts, E. and Korst, J. (1989). *Simulated Annealing and Boltzmann Machines*. John Wiley and Sons. In section 3.3.4.

Allen, J. B. (1985). Cochlear modeling. *IEEE Acoustics, Speech, and Signal Processing (ASSP) Magazine*, 2:3–29. In section 2.1.

Allen, J. B. (1996). Harvey Fletcher's role in the creation of communication acoustics. *J. Acoust. Soc. Am.*, 99(4):1825–1839. In section 2.2.1.

Alsteris, L. D. and Paliwal, K. K. (2007). Iterative reconstruction of speech from short-time Fourier transform phase and magnitude spectra. *Computer Speech and Language*, 21:174–186. In section 5.4.

Ambikairajah, E., Epps, J., and Lin, L. (2001). Wideband speech and audio coding using gammatone filter banks. In *Proc. ICASSP 2001*, volume II, pages 773–776, Salt Lake City, UT. In section 2.3.1.

Balan, R., Casazza, P. G., and Edidin, D. (2006). On signal reconstruction without phase. *Appl. Comput. Harmon. Anal.*, 20:345–356. In section 3.3.2.

Baumgarte, F. and Faller, C. (2003). Binaural cue coding — Part I: psychoacoustic fundamentals and design principles. *IEEE Trans. Speech Audio Processing*, 11(6):509–519. In section 6.2.2.

Békésy, G. (1953). Description of some mechanical properties of the organ of Corti. *J. Acoust. Soc. Am.*, 25(4):770–785. In section 2.2.1.

Bernstein, J. G. W. and Oxenham, A. J. (2008). Harmonic segregation through mistuning can improve fundamental frequency discrimination. *J. Acoust. Soc. Am.*, 124(3):1653–1667. In section 5.3.4.

Bodmann, B. G., Casazza, P. G., and Balan, R. (2008). Frames for linear reconstruction without phase. In *Proc. IEEE 42nd Conf. Inform. Science Systems 2008*, Princeton, NJ. In section 3.3.2.

Bölcskei, H., Hlawatsch, F., and Feichtinger, H. G. (1998). Frame-theoretic analysis of oversampled filter banks. *IEEE Trans. Signal Processing*, 46(12):3256–3268. In section 3.2.

Brandenburg, K. and Stoll, G. (1994). ISO-MPEG-1 audio: A generic standard for coding of high-quality digital audio. *J. Audio Eng. Soc.*, 42(10):780–792. In section 2.3.1.

Bregman, A. S. (2007). Progress in the study of auditory scene analysis. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 122–126, New Paltz, NY. In section 2.2.3.

Chai, L., Zhang, J., Zhang, C., and Mosca, E. (2007). Frame-theory-based analysis and design of oversampled filter banks: Direct computationional method. *IEEE Trans. Signal Processing*, 55(2):507–519. In section 3.2.

Chi, T., Gao, Y., Guyton, M. C., Ru, P., and Shamma, S. A. (1999). Spectro-temporal modulation transfer functions and speech intelligibility. *J. Acoust. Soc. Am.*, 106(5):2719–2732. In sections 2.2.2 and 2.3.2.

Chi, T., Ru, P., and Shamma, S. A. (2005). Multiresolution spectrotemporal analysis of complex sounds. *J. Acoust. Soc. Am.*, 118(2):887–906. In sections 2.2.2 and 4.

Cohen, L., Loughlin, P., and Vakman, D. (1999). On the ambiguity in the definition of the amplitude and phase of a signal. *Signal Processing*, 79:301–307. In section 3.3.1.

Corana, A., Marchesi, M., Martini, C., and Ridella, S. (1987). Minimizing multimodal functions of continouous variables with the "simulated annealing" algoritm. *ACM Trans. Mathematical Software*, 13(3):262–280. In section 3.3.4.

Dau, T., Puschel, D., and Kohlrausch, A. (1996a). A quantitative model of the 'effective' signal processing in the auditory system. I. model structure. *J. Acoust. Soc. Am.*, 99(6):3615–3622. In sections 2.2.2 and 2.3.2.

Dau, T., Puschel, D., and Kohlrausch, A. (1996b). A quantitative model of the 'effective' signal processing in the auditory system. II. simulations and measurements. *J. Acoust. Soc. Am.*, 99(6):3623–3631. In section 2.2.2.

Daubechies, I. (1992). *Ten lectures on wavelets*. SIAM, Philadelphia, PA, USA. In section 3.2.1.

de Boer, E. and de Jongh, H. R. (1978). On cochlear encoding: Potentialities and limitations of the reverse-correlation technique. *J. Acoust. Soc. Am.*, 63(1):115–135. In section 2.2.1.

de Cheveigné, A. (2005). Pitch perception models. In Plack, C. J., Fay, R. R., Oxenham, A. J., and Popper, A. N., editors, *Pitch*, volume 24 of *Springer Handbook of Auditory Research*, chapter 6, pages 169–233. Springer Verlag, New York. In section 2.2.3.

Dembo, A. and Malah, D. (1988). Signal synthesis from modified discrete short-time transform. *IEEE Acoustics, Speech, and Signal Processing (ASSP) Magazine*, 36(2):168–181. In section 3.2.

Drullman, R., Festen, J. M., and Plomb, R. (1994). Effect of temporal envelope smearing on speech reception. *J. Acoust. Soc. Am.*, 95(2):1053–1064. In section 4.2.2.

Dudley, H. (1940). The carrier nature of speech. *Bell Systems Technical Journal*, 19(4):495–515. In section 2.3.2.

Dugundji, J. (1958). Envelopes and pre-envelopes of real waveforms. *IRE Trans. Information Theory*, 4(1):53–57. In section 4.2.2.

EBU (1988). Sound Quality Assessment Material, Recordings for subjective tests - Users' Handbook for the EBU-SQAM Compact Disc. Technical Report 3253, European Broadcasting Union. In section 5.3.

Faller, C. and Baumgarte, F. (2003). Binaural cue coding — Part II: schemes and applications. *IEEE Trans. Speech and Audio Processing*, 11(6):520 – 531. In section 6.2.2.

Feldbauer, C. (2005). *Sparse Pulsed Auditory Representations For Speech and Audio Coding*. PhD thesis, Technische Universität Graz. In sections 1.4, 2.4, 2.5, 4.2, 4.2.1, 5.1, and 5.1.3.

Feldbauer, C. and Kubin, G. (2004). How sparse can we make the auditory representation of speech? In *Proc. Int. Conf. on Spoken Language Processing (ICSLP)*, pages 1997–2000, Jeju Island, Korea. In sections 1.1.4, 2.3.1, and 4.2.1.

Feldbauer, C., Kubin, G., and Kleijn, W. B. (2005). Anthropomorphic coding of speech and audio: A model inversion approach. *EURASIP Journal on Applied Signal Processing*, 2005(9):1334–1349. In sections 2.3.1 and 4.

Flanagan, J. L. (1962). Computational model for basilar-membrane displacement. *J. Acoust. Soc. Am.*, 34(8):1370–1376. In section 2.2.1.

Flanagan, J. L. and Golden, R. M. (1966). Phase vocoder. *Bell Systems Technical Journal*, 45:1493–1509. In section 2.3.2.

Ghitza, O. (2001). On the upper cutoff frequency of the auditory critical-band envelope detectors in the context of speech perception. *J. Acoust. Soc. Am.*, 110(3):1628–1640. In sections 2.2.2, 2.3.2, and 4.2.2.

Gnann, V. and Spiertz, M. (2009). Inversion of short-time fourier transform magnitude spectrograms with adaptive window lengths. In *Proc. ICASSP 2009*, pages 325–328, Taipei. In section 5.2.3.

Gray, R. M. (2006). *Toeplitz and Circulant Matrices: A Review*. now Publishers Inc., Hanover, MA. In section 3.1.2.

Greenberg, S. (1996). Understanding speech understanding: Towards a unified theory of speech perception. In *Proc. ESCA Workshop on the "Auditory Basis of Speech Perception"*, pages 1–8. In section 2.3.2.

Greenberg, S. and Kingsbury, B. (1997). The modulation spectrogram: In pursuit of an invariant representation of speech. In *Proc. ICASSP 1997*, volume 3, pages 1647–1650. In section 2.3.2.

Griffin, D. W., Deadrick, D. S., and Lim, J. S. (1984). Speech synthesis from short-time fourier transform magnitude and its application to speech processing. In *Proc. ICASSP 1984*, volume 9, pages 61–64. In section 3.4.1.

Griffin, D. W. and Lim, J. S. (1984). Signal estimation from modified short-time Fourier transform. *IEEE Trans. Acoustics, Speech, and Signal Processing*, 32(2):236–243. In sections 1.4, 3.3, 3.3.2, and 3.3.3.

GRPA (2010). Groupe de Recherche sur la Parole et l'Audio, WinTest download page. `http://www.gel.usherbrooke.ca/audio/wintest.htm`. In section 5.3.1.

Härmä, A. and Palomäki, K. (1999). HUTear - a Free Matlab Toolbox for Modeling of Human Auditory System. In *Proc. Matlab DSP Conference*, pages 96–99, Espoo, Finland. `http://www.acoustics.hut.fi/software/HUTear/`. In section 2.2.2.

Haykin, S. (1999). *Adaptive Filter Theory*. Prentice Hall. In section 4.

Heming, Z., Yongqi, W., and Xueqin, C. (2003). Auditory model inversion and its application. In *Proc. IEEE Int. Conf. Neural Networks and Signal Processing*, volume 2, pages 868–871, Nanjing, China. In section 4.

Hermansky, H. and Morgan, N. (1994). RASTA processing of speech. *IEEE Trans. Speech and Audio Processing*, 2(4):578–589. In section 2.3.2.

Hermansky, H., Morgan, N., Bayya, A., and Kohn, P. (1992). RASTA-PLP speech analysis technique. In *Proc. ICASSP 1992*, volume I, pages 121–124, Los Alamitos, CA, USA. In section 2.3.2.

Holters, M. and Zölzer, U. (2009). Automatic parameter optimization for a perceptual audio codec. In *Proc. ICASSP 2009*, pages 13–16, Taipei. In sections 1.2 and 3.3.4.

Irino, T. and Patterson, R. D. (1997). A time-domain, level-dependent auditory filter: The gammachirp. *J. Acoust. Soc. Am.*, 101(1):412–419. In section 6.2.2.

Irino, T. and Patterson, R. D. (2006). A dynamic compressive gammachirp auditory filterbank. *IEEE Trans. Audio, Speech and Language Processing*, 14(6):2222–2232. In section 6.2.2.

ITU-R (2001). ITU-R Recommendation BS.1387-1, Method for objective measurements of perceived audio quality (PEAQ). In section 1.1.2.

ITU-R (2003). ITU-R Recommendation BS.1534-1, Method for the subjective assessment of intermediate quality level of coding systems. In section 5.

Jepsen, M. L., Ewert, S. D., and Dau, T. (2008). A computational model of human auditory signal processing and perception. *J. Acoust. Soc. Am.*, 124(1):422–438. In sections 2.2.2 and 2.2.3.

Johnston, J. D. (1988). Transform coding of audio signals using perceptual noise criteria. *IEEE J. Selected Areas in Communications*, 6(2):314–323. In section 1.1.1.

Kabal, P. (2002). TSP Speech Database. Technical report, McGill University. `http://WWW-MMSP.ECE.McGill.CA/Documents/Data/index.html`. In section 5.3.

Kingsbury, B. E. D., Morgan, N., and Greenberg, S. (1998). Robust speech recognition using the modulation spectrogram. *Speech Communication*, 25:117–132. In section 2.3.2.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598). In section 3.3.4.

Kleijn, W. B. (2004). A Basis for Source Coding. Lecture notes, KTH, Stockholm. July 2004. In section 5.1.4.

Kollmeier, B. (2005). Auditory models for audio processing - beyond the current perceived quality? In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 178–182, New Paltz, NY. In section 2.3.

Kubin, G. and Kleijn, W. B. (1999). On speech coding in a perceptual domain. In *Proc. ICASSP 1999*, volume 1, pages 205–208, Phoenix, AZ. In sections 1.1.2, 2.3.1, 2.4, 4.2.1, and 5.1.3.

Laroche, J. and Delson, M. (1999). Improved phase vocoder time-scale modification of audio. *IEEE Trans. Audio and Speech Processing*, 7(3). In section 2.3.2.

Le Roux, J., Ono, N., and Sagayama, S. (2008). Explicit consistency constraints for STFT spectrograms and their application to phase reconstruction. In *Proc. Statistical and Perceptual Audition*, Brisbane, Australia. In section 5.2.3.

Li, Q. and Atlas, L. (2005). Properties for modulation spectral filtering. In *Proc. ICASSP 2005*, volume IV, pages 521–524, Philadelphia, PA. In section 4.2.2.

Lin, L., Ambikairajah, E., and Holmes, W. H. (2002). Perceptual domain based speech and audio coder. In Wysocki, W., editor, *6th Int. Sym. DSP for Comm. Sys.*, pages 6–11, Sydney, Australia. In section 2.3.1.

Lopez-Poveda, E. A. and Meddis, R. (2001). A human nonlinear cochlear filterbank. *J. Acoust. Soc. Am.*, 110(6):3107–3118. In section 6.2.2.

Luenberger, D. G. (1973). *Introduction to Linear and Nonlinear Programming*. Addison-Wesley, Reading, MA. In section 3.3.3.

Mallat, S. G. (1993). Matching pursuits with time-frequency dictionaries. *IEEE Trans. Signal Processing*, 41(12):3397–3415. In section 2.3.1.

Mallat, S. G. (1998). *A Wavelet Tour of Signal Processing*. Academic Press. In sections 3.2 and 3.2.1.

Moore, B. C. J. (2003). *An Introduction to the Psychology of Hearing*. Academic Press, 5th edition. In sections 1.1.2, 2.1, 2.1.2, 2.2.1, 4.2.2, 4.2.3, and 6.2.2.

Moore, B. C. J. and Glasberg, B. R. (2010). The role of temporal fine structure in harmonic segregation through mistuning. *J. Acoust. Soc. Am.*, 127(1):5–8. In section 5.3.4.

Moore, D. R. (1991). Anatomy and physiology of binaural hearing. *Audiology*, 30(3):124–134. In section 2.1.2.

Oppenheim, A. V. and Schafer, R. W. (1989). *Discrete-time Signal Processing*. Prentice-Hall, Upper Saddle River, NJ. In section 5.2.3.

Painter, T. and Spanias, A. S. (2000). Perceptual coding of digital audio. *Proceedings of the IEEE*, 88(4):451–515. In section 1.1.1.

Patterson, R. D. and Holdsworth, J. (1996). A functional model of neural activity patterns and auditory images. In *Advances in Speech, Hearing and Language Processing*, volume 3, pages 547–558. JAI Press. In section 2.2.2.

Patterson, R. D., Robinson, K., Holdsworth, J., McKewon, D., Zhang, C., and Allerhand, M. (1992). Complex sounds and auditory images. In Cazals, Y., Demany, L., and Horner, K., editors, *Auditory Physiology and Perception, Proc. 9th Int. Symposium on Hearing*, pages 429–446, Oxford, UK. Pergamon Press. In sections 2.2.1 and 2.3.1.

Pei, S.-C. and Yeh, M.-H. (1997). An introduction to discrete finite frames. *IEEE Signal Processing Magazine*, 14(6):84–96. In section 3.2.

Pichevar, R., Najafzadeh-Azghandi, H., and Thibault, L. (2007). A biologically-inspired low-bit-rate universal audio coder. In *AES 122nd Convention*, Amsterdam, The Netherlands. In section 2.3.1.

Proakis, J. G. and Manolakis, D. G. (1996). *Digital Signal Processing: Principles, Algorithms and Applications*. Prentice-Hall, Upper Saddle River, NJ. In sections 3.1.1 and A.

Quatieri, T. F., Tom, V. T., Hayes, M. H., and McClellan, J. H. (1981). Convergence of iterative signal reconstruction algorithms. In *Proc. ICASSP 1981*, volume 6, pages 35–38, Atlanta, GA. In section 3.3.3.

Robles, L. and Ruggero, M. A. (2001). Mechanics of the mammalian cochlea. *Physiological Reviews*, 81(3):1305–1352. In section 2.2.1.

Schimmel, S. M. (2007). *Theory of Modulation Frequency Analysis and Modulation Filtering, with Applications to Hearing Devices*. PhD thesis, University of Washington. In sections 2.3.2, 3.3.1, and 4.2.2.

Schimmel, S. M. and Atlas, L. E. (2005a). Analysis of signal reconstruction after modulation filtering. In Bajard, J. C., Meloni, N., and Plantard, T., editors, *Study of modular inversion in RNS. Proc. of SPIE.*, volume 5910, pages 152–161. In section 2.3.2.

Schimmel, S. M. and Atlas, L. E. (2005b). Coherent envelope detection for modulation filtering of speech. In *Proc. ICASSP 2005*, volume IV, pages 221–224, Philadelphia, PA. In section 2.3.2.

Schimmel, S. M., Fitz, K. R., and Atlas, L. E. (2006). Frequency reassignment for coherent modulation filtering. In *Proc. ICASSP 2006*, volume 5, pages 261–264, Toulouse. In section 2.3.2.

Slaney, M. (1995). Pattern playback from 1950 to 1995. In *Proc. IEEE Int. Conf. on Systems, Man and Cybernetics*, volume 4, pages 3519–3524, Vancouver, Canada. In sections 2.3 and 4.3.2.

Slaney, M., Naar, D., and Lyon, R. E. (1994). Auditory model inversion for sound separation. In *Proc. ICASSP 1994*, volume II, pages 77–80, Adelaide, SA, Australia. In section 4.

Smith, E. C. and Lewicki, M. S. (2006). Efficient auditory coding. *Nature*, 439(7079):978–982. In section 2.3.1.

Smith, Z. M., Delgutte, B., and Oxenham, A. J. (2002). Chimaeric sounds reveal dichotomies in auditory perception. *Nature*, 416(6876):87–90. In sections 4.2.3 and 6.1.

Soulodre, G. A. (1998). *Adaptive Methods for Removing Camera Noise from Film Soundtracks*. PhD thesis, McGill University, Montreal, Quebec, Canada. In section 5.1.3.

Spanias, A., Painter, T., and Atti, V. (2007). *Audio Signal Processing and Coding*. John Wiley and Sons. In section 1.1.1.

Strahl, S. and Mertins, A. (2008). Sparse gammatone signal model optimized for english speech does not match the human auditory filters. *Brain Research*, 1220:224–233. In section 2.3.1.

Strahl, S. and Mertins, A. (2009). Analysis and design of gammatone signal models. *J. Acoust. Soc. Am.*, 126(5):2379–2389. In section 5.1.1.

Thiemann, J. and Kabal, P. (2007). Reconstructing audio signals from modified non-coherent hilbert envelopes. In *Proc. INTERSPEECH 2007*, pages 534–537, Antwerp, Belgium. In section 4.2.2.

Thompson, E. and Dau, T. (2008). Binaural processing of modulated interaural level differences. *J. Acoust. Soc. Am.*, 123(2):1017–29. In section 2.2.3.

Tom, V. T., Quatieri, T. F., Hayes, M. H., and McClellan, J. H. (1981). Convergence of iterative nonexpansive signal reconstruction algorithms. *IEEE Trans. Acoustics, Speech, and Signal Processing*, ASSP-29(5):1052–1058. In section 3.3.3.

Unoki, M., Irino, T., Glasberg, B., Moore, B. C. J., and Patterson, R. D. (2006). Comparison of the roex and gammachirp filters as representations of the auditory filter. *J. Acoust. Soc. Am.*, 120(3):1474–1492. In section 6.2.2.

Vanderbilt, D. and Louie, S. G. (1984). A monte carlo simulated annealing approach to optimization over continuous variables. *Journal of Computational Physics*, 56(2):259 – 271. In section 3.3.4.

Yang, X., Wang, K., and Shamma, S. A. (1992). Auditory representations of acoustic signals. *IEEE Trans. Information Theory*, 38(2):824–839. In sections 2.2.2, 2.2.3, 2.3, and 6.1.

Zeng, F.-G., Nie, K., Liu, S., Stickney, G., Rio, E. D., Kong, Y.-Y., and Chen, H. (2004). On the dichotomy in auditory perception between temporal envelope amd fine structure cues. *J. Acoust. Soc. Am.*, 116(3):1351–1354. In sections 4.2.3 and 6.1.

Zhu, X., Beauregard, G. T., and Wyse, L. L. (2007). Real-time signal estimation from modified short-time fourier transform magnitude spectra. *IEEE Trans. Audio, Speech and Language Processing*, 15(5):1645–1653. In section 5.2.3.

Zweig, G., Lipes, R., and Pierce, J. (1976). The cochlear compromise. *J. Acoust. Soc. Am.*, 59(4):975–982. In section 2.1.

Zwicker, E. and Fastl, H. (1999). *Psychoacoustics: Facts and Models*. Springer Verlag, 2nd edition. In sections 2.1 and 2.1.2.

Zwicker, E. and Feldkeller, R. (1967). *Das Ohr als Nachrichtenempfänger*. S. Hirzel Verlag. In section 2.2.