Towards Automatic Musical Instrument Timbre Recognition

Tae Hong Park

A DISSERTATION

PRESENTED TO THE FACULTY

OF PRINCETON UNIVERSITY

IN CANDIDACY FOR THE DEGREE

OF DOCTOR OF PHILOSOPHY

RECOMMENDED FOR ACCEPTANCE

BY THE DEPARTMENT OF MUSIC

November 2004

ProQuest Information and Learning Company
300 North Zeeb Road
PO Box 1346
Ann Arbor, MI 48106-1346

Abstract

This dissertation is comprised of two parts – focus on issues concerning research and development of an artificial system for automatic musical instrument timbre recognition and musical compositions. The technical part of the essay includes a detailed record of developed and implemented algorithms for feature extraction and pattern recognition. A review of existing literature introducing historical aspects surrounding timbre research, problems associated with a number of timbre definitions, and highlights of selected research activities that have had significant impact in this field are also included. The developed timbre recognition system follows a *bottom-up*, *data-driven model* that includes a pre-processing module, feature extraction module, and a RBF/EBF (Radial/Elliptical Basis Function) neural network-based pattern recognition module. 829 monophonic samples from 12 instruments have been chosen from the Peter Siedlaczek library (Best Service) and other samples from the internet and personal collections. Significant emphasis has been put on feature extraction development and testing to achieve robust and consistent feature vectors that are eventually passed to the neural network module. In order to avoid a *garbage-in-garbage-out* (GIGO) trap and improve generality, extra care was taken in designing and testing the developed algorithms using various dynamics, different playing techniques, and a variety of pitches for each instrument with inclusion of attack and steady-state portions of a signal. Most of the research and development was conducted in Matlab. The compositional part of the essay includes brief introductions to *"A d'Ess Are," "Aboji," "48 13 N, 16 20 O,"* and *"pH-*

SQ." A general outline pertaining to the ideas and concepts behind the architectural designs of the pieces including formal structures, time structures, orchestration methods, and pitch structures are also presented.

Acknowledgements

First and foremost I would like to thank Paul Lansky for his thoughtful guidance throughout my studies at Princeton. His seemingly bottomless insights and his economic, but always profound guiding words without a doubt helped me in all facets of my development as a composer and researcher. I am particularly thankful for his kindness, support, countless discussions on all walks of life, and his vision regarding the Princeton University Composition Program whereby instituting an environment where students from diverse backgrounds are given the opportunity to study and engage in research in the broadest sense and most interesting ways. I am greatly indebted to Perry Cook, a wonderful and caring teacher who was and always will be an endless source for ideas and great inspiration. My experiences as his preceptor in 3 undergraduate courses were just one of the many highlights working with him at Princeton. Many thanks to Steve Mackie whose intricate knowledge of acoustic instruments and instrumentation have been invaluable in helping me gain serious interest in compositional projects that were initially outside my normal scope of concentration. I am grateful to Dan Trueman for the very effective composition lessons and critiques which continually helped me with my compositions. I am especially thankful for his generous advice in helping me organize Listening in the Sound Kitchen 2003. I would like to acknowledge Barbara White for her insightful and detailed feedback on my compositions and papers which were handed to her in black and always came back in red. Thanks also to Paul Koonce who has been very influential in my compositional work and thought

processes during and after his tenure at Princeton. I wish to thank Peter Westergaard and can only say that I regret not coming to Princeton earlier to study with him before his retirement. My gratitude to Sun Yuan Kung at the engineering school for his help with neural networks and the numerous meetings we have had.

Thanks to my colleague Paul Botelho and in particular Daniel Biro who has been the most critical of my work. Warm thanks to Marilyn Ham, Cindy Masterson, Irene McElroy, Greg Smith, and Kyle Subramaniam for their always friendly and professional help, and especially Paula Matthews whose kindness I am truly grateful. Mary Roberts and Jim Allington for their technical expertise and deep knowledge of all things wired. I also wish to express my warmest thanks to Jon Appleton who has been doubtlessly one of the most supportive and helpful friends during the course of my graduate studies.

I would like to further express my immeasurable gratitude to my parents and brothers who have at all times been supportive and encouraging throughout my life. My oldest brother Kihong Park has especially guided me in critical and difficult times with his invaluably logical and thoughtful comments and advice.

Lastly, I would like to thank my wife Kyoung Hyun Ahn without whom it would not have been possible to get to this point in my life. For her unwavering support, encouragement, long nights (more like early mornings) waiting for me, and

experiencing every difficult and happy moment with me at Princeton I am deeply indebted.

Table of Contents

Chapter 1 Motivation

The motivation for choosing a Ph.D. thesis on the subject of automatic timbre recognition is multifaceted. The most important motivation is in the pursuit of understanding musical sound. Generally speaking, a musical sound can be described with 4 perceptually oriented parameters – pitch, magnitude, duration, and timbre. The first three components of a musical sound can be generally characterized using a one-dimensional scale. Pitch can be defined by its fundamental frequency in Hertz (Hz), the magnitude of a sound can be described in decibels (dB), and time is measurable using the units of seconds. Timbre however, is reluctant to be put on a one-dimensional scale and furthermore does not conveniently lend itself to be defined by some sort of "timbral unit." It is impossible to assert that the flute is a 0.4 timbral unit whereas the double bass is a 0.9 timbral unit for example. Hence, it seems that musical timbre is multidimensional in nature and it also appears that it is the component of musical sound that is the most difficult to understand. Trying to better understand this last ambiguous component of sound has been the foremost reason for choosing the topic of timbre research.

The second reason for pursuing a dissertation topic on timbre is in the actual implementation of the system. I aim to answer the simple question "musically trained people can do it with relative ease, how hard would it be to design a robust artificial system to do similar tasks?" This research field has been a long-term project for me which started with my master's thesis at Dartmouth College

entitled “Salient Feature Extraction of Musical Instrument Signals” (Park 2000). The master’s thesis focused on aspects of salient feature extraction techniques and implementation of the algorithms which were developed mostly in Matlab and then ported to Java.

Lastly the motivation for writing the essay was in part to provide a comprehensive and detailed document presenting and contributing new approaches and suggestions for timbre research for experts and non-experts alike, in the hope that underlying concepts, techniques, and algorithms would be available without ambiguity and actually help the reader in his or her own research and studies in musical timbre.

Chapter 2 Introduction

The curiosity humans have had with musical sounds can be traced to as far back as ancient Chinese dynasties and to the Greek philosopher Pythagoras of Samos. Pythagoras, who is largely credited for discovering integer ratio relationships of string vibration, marks one of the first steps in the quest for understanding the mysteries of musical timbre. Timbre has always been a subject surrounded by ambiguity and mystery. As a matter of fact, it is still a term that experts in fields such as psychology, music, and computer science have a somewhat difficult time defining clearly. Numerous descriptions of timbre, especially in the past, have dealt with semantic descriptions such as cold, warm, bright, dull, short, long, smooth, rough … etc., which however do not really reveal concrete and tangible information about its structure, dimension, or mechanism.

One might be tempted to ask "we identify, recognize, and differentiate musical timbre quite easily, so why is it so hard to understand and design a machine to do it?" This is a valid point but in many cases the things that humans do with little effort, such as smelling, touching, seeing, and recognizing objects are the most difficult aspects to understand, and especially difficult to model on a computer system. As a matter of fact, it seems that in many cases machines and humans have an inverse relationship when it comes to accomplishing perceptual tasks. Tasks that machines do effortlessly are done with greater difficulty by humans and vice-versa. Through continued research, especially accelerated by the advent of more and more powerful computers, we have witnessed advances

bringing us a little closer towards understanding the complexities of timbre. However, we are still at the point where it is difficult to robustly design machines to “listen” to monophonic tones outside the highly controlled laboratory environment where samples are recorded with great care and in many cases normalized in pitch, dynamics, and acoustic space properties. The level of difficulty rises exponentially for polyphonic sound sources and designing systems that can listen to “regular” CD tracks and discern timbral idiosyncrasies.

So, what makes timbre so difficult to understand? The reasons of course are far too complex to put into one sentence, but perhaps after reading this thesis we will have a better idea about some aspects of musical instrument timbre and gain insights on some of its dimensions and structure. In the meantime, I will assert for now that the main problems underlying the complexity may be attributed to the following:

1 Timbre is a perceptual and subjective attribute of sound, rather than a purely physical one, making it a somewhat “non-tangible” entity.
2 Timbre is multidimensional in nature, where the qualities and importance of features, and the number of dimensions are not fully understood.
3 There are no current existing subjective scales to make judgments about timbre.

4 Timbre is an interdisciplinary research area that covers fields such as acoustics, music, computer science, engineering, and psychology.

5 There are no standard sets of sound examples against which the researcher can test their developed models.

The essay is divided into 10 chapters with an appendix for flowcharts, derivations of equations, other supplemental information, and references. The first 5 chapters give outlines of the thesis and review of literature in timbre research. Chapters 6 and 7 present feature extraction, pattern recognition algorithms, and chapter 8 results of the system's performance. The penultimate chapter summarizes and concludes the technical part of the essay addressing problems and insights gained in timbre research and the final chapter outlines of the musical composition part of the thesis.

According to Puterbaugh (Puterbaugh 1999), “At one point, timbre referred to a kind of bell (i.e. tabular bell) …” evolving later to “…the sound of a bell, then the sound quality of an instrument in general, and finally to the quality of sound in general.” Generally speaking, as mentioned above a musical sound can be described by four factors: *pitch, loudness, duration,* and *timbre*. The first three terms are believed to be one-dimensional, and are better understood primarily due to the existence of their physical correlates. That is, pitch is measured in terms of fundamental frequency (in most cases), loudness explained through intensity, duration determined by the lifetime of a tone or musical phrase (unless too short). This is not to say that the three components, especially time are necessarily perceived in a linear nor absolute way. For example, a piece of music or musical tone in the context of composition can have the exact same duration of say, 2 minutes, but can be perceived differently depending on the complexity, density, musicality, and many other factors. The absence of total clarity for all three components; namely pitch, loudness, and duration is due to the fact that they are to some degree perceptual attributes and not purely physical attributes. Timbre is an even more complex component of sound as it cannot be mapped to a one-dimensional scale and furthermore is not uncoupled from the other one-dimensional components. As one would expect, it has had and probably still has many aliases such as “tone quality,” “klangfarbe” (Helmholtz 1954), “sound color” (Slawson 1985), and “tone color” (Levarie, Levy 1981) to name a few. It is somewhat ironic that we try to define timbre as we still

do not exactly know what it is. Yet, it is something that most, if not all people perceive and differentiate without too much difficulty, especially those persons who are trained musicians although on a subjective level (see section 5.1 for details on human performance on identification tasks). Words like dark, bright, shrill, flat, funky, weird, funny, bland, rough, smooth, cold, warm, pure, rich, and countless other semantic descriptions are used on a daily basis among professionals working with timbre and non-professionals alike. Indeed, there have not been much better ways to communicate timbral metaphors, especially in the realm of musical composition. Perhaps the vagueness underlying the definition of timbre is one reason why people like Bregman and Slawson (Slawson 1985) find the need to take the word out of their dictionary altogether saying, "Until such time as the dimensions of timbre are clarified perhaps it is better to drop the term timbre" (Bregman 1990). Martin furthermore states that, "… it is empty of scientific meaning and should be expunged from the vocabulary of hearing science" (Martin 1999). Not too long has passed since, and before taking such drastic measures, I think it would be worthwhile to look at the term "timbre" and decide for ourselves if it needs to be "expunged."

3.1 Definitions

In this section I will present a number of definitions and comments about timbre given by renowned researchers in this field and summarize some salient timbral features that have been discovered throughout the years, starting with Helmholtz.

Helmholtz's (Helmholtz 1954) definition from the original 1877 translation reads:

> "...the amplitude of the vibration determines the force or loudness, and the period of vibration the pitch. Quality of tone can therefore depend upon neither of these. The only possible hypothesis, therefore; is that the quality of tone should depend upon the manner in which the motion is performed within the period of each single vibration."
>
> "When we speak in what follows of musical quality of tone, we shall disregard these peculiarities of beginning and ending, and confine our attention to the peculiarities of the musical tone which continues uniformly."
>
> "The quality of the musical portion of a compound tone depends solely on the number and relative strength of its partials simple tones, and in no respect on their differences of phase."

Fletcher (Fletcher 1934)

> "...timbre depends principally upon the overtone structure; but large changes in the intensity and the frequency also produce changes in the timbre."

Seashore (Seashore 1938)

> “In general, we may say that, aside from accessory noises and inharmonic elements, the timbre of a tone depends upon (1) the number of harmonic partials present, (2) the relative location or locations of these partials in the range from the lowest to the highest, and (3) the relative strength or dominance of each partial. … depends upon its harmonic structure as modified by absolute pitch and total intensity… we must also take phase relations into account.”
>
> “The German word for it is ‘klangfarbe’; the French word (frequently borrowed in English) is 'timbre'.”

Licklider (Licklider 1951)

> “… it can hardly be possible to say more about timbre than that it is a 'multidimensional' dimension.”

American National Standards Institute (ANSI 1960, 1973)

> “12.9 Timbre. Timbre is that attribute of auditory sensation in terms of which a listener can judge that two sounds similarly presented and having the same loudness and pitch are dissimilar.”

> “Timbre depends primarily upon the spectrum of the stimulus, but it also depends upon the waveform, the sound pressure, the frequency location of the spectrum, and the temporal characteristics of the stimulus.”

American Standards Association (ASA 1960)

> “Timbre is that attribute of sensation in terms of which a listener can judge that two sounds having the same loudness and pitch are dissimilar.”

Schouten (Schouten 1968)

> “In most textbooks timbre is defined as the overtone structure or the envelope of the spectrum of the physical sound. This definition is hopelessly insufficient, as I hope to prove by demonstrating that timbre can be expressed in terms of at least five major parameters … 1.The range between tonal and noise-like character, 2.The spectral envelope, 3.The time envelope in terms of rise, duration and decay, 4.The change both of spectral envelope (formant glide) or fundamental frequency (micro-intonation), 5.The prefix, an onset of a sound quite dissimilar to the ensuing lasting vibration.”

Scholes (Scholes 1970)

> “Timbre means tone quality-coarse or smooth, ringing or more subtly penetrating, “scarlet” like that of a trumpet, “rich brown” like that of a cello, or “silver” like that of the flute. These color analogies come naturally to every mind … The one and only factor in sound production which conditions timbre is the presence or absence, or relative strength or weakness, of overtones.”

Plomp (Plomp 1976)

> “ … timbre depends upon several parameters of the sound including the spectral envelope and its change in time, periodic fluctuations of the amplitude or the fundamental frequency, and whether the sound is a tone or noise.”

> “Clearly...timbre is determined by the absolute frequency position of the spectral envelope rather than by the position of the spectral envelope relative to the fundamental… [von Bismarck] found that sharpness as the major attribute of timbre is primarily related to the position of the loudness centre on an absolute frequency scale rather than to a particular shape of the spectral envelope …Low frequency tones do indeed sound dull and high-frequency tones sharp...”

"… the spacing of the harmonics, determined by the fundamental frequency, is responsible for the timbre dissimilarity of sounds with different pitch but similar spectral envelopes."

Rasch and Plomp (Rasch, Plomp 1982)

"Timbre is, after pitch and loudness, the third attribute of the subjective experience of musical tones… Especially important is the relative amplitude of the harmonics. … temporal characteristics of the tones may have a profound influence on timbre as well … Both onset effects (rise time, presence of noise or inharmonic partials during onset, unequal rise of partials, characteristic shape of rise curve, etc.) and steady state effects (vibrato, amplitude modulation, gradual swelling, pitch instability; etc) are important factors in the recognition and, therefore, in the timbre of tones."

Bregman (Bregman 1990) Comments on the ASA definition

"This is, of course; no definition at all … it implies that there are some sounds for which we cannot decide whether they possess the quality of timbre or not. In order for the definition to apply; two sounds need to be able to be presented at the same pitch, but there are some sounds … that have no pitch at all … Either we must assert that only sounds with pitch can have timbre, meaning that we cannot discuss the timbre of a tambourine or of the musical

sounds of many African cultures, or there is something terribly wrong with the definition."

"Until such time as the dimensions of timbre are clarified perhaps it is better to drop the term timbre."

Martin (Martin 1999)

"Although the word timbre appears in the abstract of this dissertation… previous two chapters … conclusion of Chapter 7 … it is empty of scientific meaning and should be expunged from the vocabulary of hearing science."

As seen from the comments and milestones presented above, some definitions were problematic and even false, where others contributive in narrowing down its scope. Below I have given a summary of the above descriptions in two categories: false and possible attributes of timbre. The validity for categorization will be expanded and elaborated in the following sections.

False and problematic assertions:

1. Helmholtz's notion of the ear accomplishing frequency analysis much like the Fourier transform *without* regard to phase and his view that the "quality of music" must be concentrated on the steady-state portion *alone*, was for a long time largely unquestioned until

Chapin and Firestone (Chapin, Firestone 1934). However, it has been found that phase and the non-steady-state portions of a tone *do* play a significant role in characterizing tones – especially with relation to the fundamental frequency.

2. Seashore's "accessory noise and inharmonic elements" (Seashore 1967) do contribute to timbral quality and are not mere "accessories."
3. ASA's definition from 1960 actually says what timbre is *not* and not what it is and leaves out sounds that are non-pitched (ASA 1960).
4. Relative strengths and weaknesses of overtones being the only features for timbre (Scholes 1970).

Possible attributes of timbre:

1. Number of harmonics, relative strengths of harmonics.
2. Harmonic structure, spectral envelope change over time.
3. Loudness and fundamental frequency.
4. Phase, inharmonic partials, synchronicity of partials
5. Presence of noise.
6. Multidimensionality.
7. Temporal characteristics of stimulus especially rise time.
8. Steady-state and attack portions

4.1 Classical Theory of Timbre: Importance of the Steady-State and the Controversy over Phase

Helmholtz, who was inspired by Ohm's *acoustical law* (Ohm 1843) and the Fourier transform decomposing a signal into frequency and phase components, applied those concepts in trying to explain timbral attributes of tones. Helmholtz asserted that timbre depends "… solely on the number and relative strength of its partials, simple tones, and in no respect on their differences of phase" (Helmholtz 1954). However, this is not to say that Helmholtz did not investigate phase in his experiments, which he actually did using synthetic test tones. Nevertheless, he concluded that the phase patterns of the harmonic series contributed minimally to the perception of timbre and thus could be ignored while at the same time expressing ambiguity towards the influence of phase on upper harmonics that physically give rise to roughness in timbre perception. Helmholtz also stressed the importance of the "musical tone, which continues uniformly," i.e. the steady-state part of a tone, disregarding "…peculiarities of beginning and ending …" thereby neglecting some of the temporal aspects of musical tones. Far less attention was paid to those timbral aspects that were transient and observable in normal listening experiences. The concept behind his thoughts became to be known as the *classical theory* and has without a doubt contributed greatly to the research in timbre.

This view of phase's insignificance was generally unquestioned until Chapin (Chapin, Firestone 1934) and Fletcher (Fletcher 1934) revealed some of the flaws in Helmholtz' thinking. Fletcher showed that at high intensities, phase alterations contributed small amounts of change in the sensation of a tone. The controversy over the importance of phase was further clarified some 30 years later by Plomp and Steeneken (Plomp, Steeneken 1969; Plomp 1976). Using *Multidimensional Scaling* (MDS) techniques, they were able to find two tones that had identical harmonic amplitude characteristics. The pair of tones however, showed phase patterns that were maximally different from each other, therefore ideal in investigating the influence of phase on the perception of timbre. They concluded that phase did indeed influence the perception of the tone, although on a much lesser degree than the magnitude components of the frequencies. Also, measurements indicated the subtlety to be equivalent to the difference between two closely related vowels. Furthermore, it was discovered that the qualitative effect of phase on timbral perception was inversely proportional to the fundamental frequency, i.e. the higher the fundamental frequency the less phase contribution there is to the overall timbre perception.

Building on the concepts presented by Helmholtz, the notion of spectral envelope and the significance of the steady-state, the formant model (see section 6.2.8.1 for details on formants and steady-state) was applied to musical tones (Fletcher 1934). The ideas of formants were borrowed from research practices in speech research, now the foundation of current voice coders in digital communication. It

is common to make use of visual metaphors to describe the hilly and mountainous spectral contours of formants, which are basically strong "frequency amplitude poles" in the frequency spectrum that hold up a tent, so to speak. Bartholomew (Bartholomew 1945), who was interested in determining why certain voices sounded "good" investigated female and male voices, leading to the development of his *formant theory*. Bartholomew's formant theory is usually taken as a supplement and not a replacement to the classical theory of Helmholtz (Saldanha, Corso 1964). He proposed that "… the characteristic tone quality of an instrument is due to the relative strengthening of whatever partials lie within a fixed or relatively fixed region of the musical scale…" However, the formant model is rather limited, in that it is restricted to the prediction of the steady-state parts only. I have exploited this fact (Park 2000) and used it as a means to compute the noise quality of a tone using *Linear Predictive Coding* (LPC) explained in section 6.2.8.

As mentioned above, semantic descriptions in timbre have been used ubiquitously and have also played an important part in its communication. For example, Helmholtz used *roughness, sweetness, fullness, brightness,* and so forth in his scientific writings although some did not directly map to tangible and measurable quantities. To that end, perhaps in the spirit of the classical theory, Lichte (Lichte 1941) applied verbal timbral measurement techniques in his investigations and had listeners evaluate sets of steady-state tones with systematically altered spectral envelopes. He reported three semantic features

that he considered important – brightness, fullness, and roughness. He interpreted these attributes as follows:

1. Brightness: The midpoint of energy distribution of the frequency scale.
2. Fullness: The relative presence or absence of odd or even harmonics. (Later we will see its correlation to nasality and hollowness characterized by the dominance of odd harmonics.)
3. Roughness: The presence of harmonics generally higher than the 6^{th} harmonic.

In a different study, von Bismark (von Bismark 1974) had subjects rate speech, musical, and synthetic tones with 30 *a priori* selected verbal attributes applying the *Semantic Differential* (SD) method (Osgood, Suci, Tannenbaum 1957). SD comes from the communication literature and is conventionally used to study people's reactions to stimulus words. It is also used to study different connotations for words through psychological “distance measurements.” The verbal attributes in von Bismark’s study were selected in such a way as to have bipolar characteristics: e.g. cold-warm, full-empty, soft-hard, and pure-rich. Some salient verbal pairs he found for timbre were dull-sharp, compact-scattered, full-empty, and colorful-colorless. The dull-sharp element was further found to be related to the midpoint of the spectral energy, and the compact-scattered pair to tonal-noise character of a tone.

4.2 Further Attributes of Timbre: Spectral, Temporal Features and MDS

The origins of *Multidimensional Scaling* (MDS) can be found in the field of psychometrics, which was proposed and coined by Torgerson (Torgerson 1952). MDS is a technique designed to help understand human judgment achieved through similarity/dissimilarity measurements of objects and has also been applied in timbre research. Subjects are presented with *pairs* of tones and asked to scale the similarity or dissimilarity between them. Very much like the SD method discussed in the previous section, adjectives of opposite extremes such as dark-bright are used. However unlike SD, in MDS *pairs* of tones are presented to the subject. The data analyzed in MDS systems are obtained via N number of sources (the subjects) each relating to M entities (sounds) in a pair-by-pair basis. With the acquired data (usually a higher order dimension), an MDS procedure is applied ultimately rendering a low dimensional space. These dimensions are often reduced to 2 or 3 known as *timbre spaces* (see figure 4.1). The relative distances are typically computed with *Euclidean* metrics and the resulting axes are analyzed for any relation to spectral and temporal features of a sound. It has been suggested with great optimism that it is theoretically possible to use up to five specific rating scales to accurately identify almost any timbre (Howard 2001).

Plomp (Plomp 1970) was one of the forerunners to try MDS investigations with musical tones. He studied its application on steady-state portions of 9

instrumental tones that were divided into 18-1/3 octave sub-bands resulting in 18 dimensions.

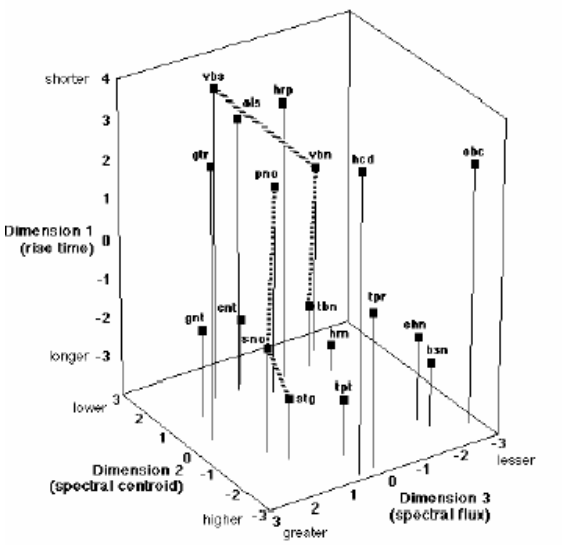


Figure 4.1 Timbre space (McAdams et al. 1995)

He continued to use MDS to investigate the contribution of phase to timbre as briefly discussed in the preceding section (Plomp 1976). He conducted his experiments using complex tones with 8 harmonics that were equalized in fundamental frequency and loudness. As it turned out, Plomp was able to find that phase contributed to timbre perception, especially with fundamental frequencies that resided on the low end of the frequency scale. However, as the fundamental was increased the effect of phase on timbre perception seemed to decrease accordingly. Also, he found fixed spectral envelopes to be more important than time-variant envelopes. He explained his assertion through what he called the *spectral space* (measure of spectral differences in complex tones) saying, "… the more similar the sound spectra are of two complex tones, the more similar they are in timbre." In summary, not only was it important to see the

significance of phase in a complex tone, but it was also important that the timbre space obtained through MDS showed salient timbral dimensions which correlated to physical aspects of tones.

Similarity studies performed prior to 1975 have primarily dealt with steady-state portions of a sound and harmonic qualities of timbre. Temporal attributes such as onsets and decays were not included. Grey (Grey 1975, 1977) however, carried out significant MDS work with tones that were temporally complete as they included the *attack, sustain,* and *decay* portion of a tone (flutes, double reeds, single reeds, brasses, and strings were included) although the tones were also normalized in fundamental frequency (Eb above middle C), loudness, and duration. His studies revealed three acoustical dimensions:

1. Spectral energy distribution: Observed as spectral bandwidths and concentrations of frequency amplitude components in the steady-state portion.
2. Synchronicity: The synchronicity in the *collective* attacks and decays of upper harmonics, i.e. observation of tone onsets/offsets as to whether upper harmonics enter/leave in close time alignments.
3. Low amplitude, high precedent energy: most often inharmonic energy, during the attack phase characterized by components of high-frequency, low-amplitude values.

Notable findings of his experiments were not only the possibility of new timbral features but also feasible evidence suggesting a data reduction scheme that may occur somewhere within our hearing system. As shown in figure 4.2 (b), the data reduction occurred in the order of 100:1, implemented with the so-called *line-segment approximation* that had between 5 to 8 nodes (note the smooth harmonic envelopes decimated to rough representations using only a fraction of the original data points). When synthesized with the reduced data sets, an impressive tone quality, very close to its original was achieved. The drastic modifications in the complex time-variant functions suggested that micro-fluctuations found in the analyzed tones may not be that significant in the *recognition* of timbre.

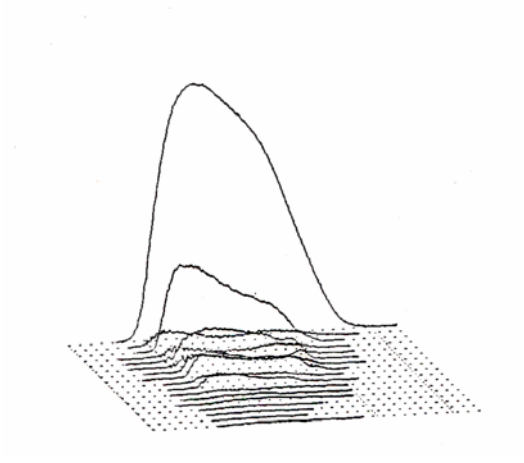


Figure 4.2 (a) Original analysis (Grey 1975)

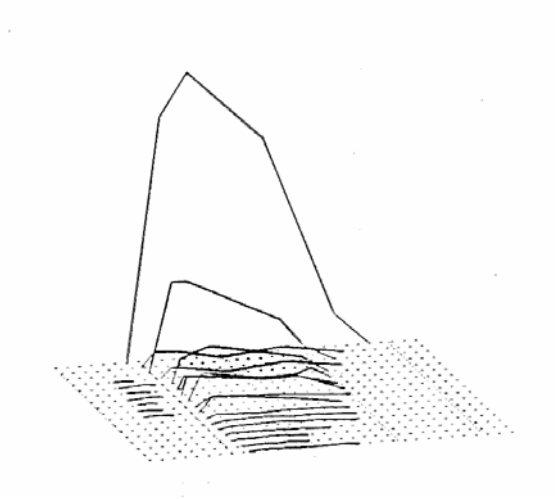


Figure 4.2 (b) Line-segment approximations (Grey 1975)

Further insights that may imply a data reduction scheme in the human auditory system are observed in the fact that for digital telephony of speech data, a sampling rate of 8 kHz is used as a standard. This means that only frequencies up to 4 kHz are available at any given time. Furthermore, due to data compression techniques such as LPC and codebook based excitation methods such as CELP; a further limited representation of the original 4 kHz is actually transmitted to the receiver. Remarkably, even under these poor conditions brought about through decimation and data compression, it is still possible to recognize the speaker and indeed, musical instrument sounds. This is not to say that the sound quality is optimal which it is not, nor am I suggesting that "CELP based synthesis" produces all the micro subtleties desirable for a sophisticated digital compositional tool; what it does suggest however, is that for the *recognition* of a sound object the whole range of spectral information is not

necessarily required. In short, the sound *quality* to a "reasonable extent" is not critical in the *recognition* of a musical instrument.

Other studies with MDS resulted in observations of additional salient features such as the *spectral centroid* (Iverson, Krumhansl 1993) explained further in section 6.1.9. The salience of the spectral centroid was additionally substantiated by Krimphoff et al. (Krimphoff, McAdams, Winsberg 1994) who found two more significant features: the *log-rise time* and *irregularity* (see section 6.2.2 and 6.1.10). McAdams et al. further elaborated on *spectral irregularity* substituting it with the *spectral flux* (McAdams, Beauchamp, Meneguzzi 1999). An interesting observation that can be made with timbre space is that timbre judgments seem to be based on physical properties of tones rather than some sort of non-acoustical grouping scheme. In Grey and Gordon's words "This may indicate that clustering is based more upon perceived features of tone rather than to some cognitive recognition or class-membership naming function" (Grey, Gordon 1978).

As previously mentioned, past studies have primarily focused on harmonic tones, that is, tones that elicit a sense of pitch and with absence of noise components – the classical approach to timbre. However, a recent study by Lakatos (Lakatos 2000) included instruments divided into 17 harmonic and 18 percussive sounds mostly taken from the MUMS library (McGill University Master Samples). The experiments were conducted with 34 human subjects asked to do similarity

judgments using discrete sliders consisting of 500 points. The tests for harmonic, non-harmonic, and a combination of both were conducted separately within the context of MDS. The harmonic tests resulted in a two-dimensional timbre space of log-rise time and spectral centroid. The percussive experiments rendered a 3-dimensional timbre space with attack and spectral centroid, but the third dimension was somewhat difficult to categorize and was correlated to the verbal description of richness. The combination test resulted in a two-dimensional space and again identified attack time and spectral centroid as the salient timbral features. One interesting observation in this study was the ability of timbre perception to be independent of duration, as the same 2-dimensions (attack time and spectral centroid) were found in both harmonic and percussive tone tests.

With the application of similarity experiments in MDS, a number of salient features such as spectral centroid, attack time and the like have been discovered. It is encouraging that features like the spectral centroid, referring to the brightness of a sound, have been "cross-validated" as prominent attributes by different researchers (e.g. Helmholtz 1954, Lichte 1941, Krumhansl 1989, McAdams 1995, Lakatos 2000). In summary, the following important key points can be made:

1. Timbre is unambiguously multidimensional.
2. Observation of spectral envelope, especially centroid compared to the physical correlate brightness is important.

3. The attack portion of a tone, particularly the *rise time*, plays an important role in timbre.

Although MDS studies have been praised for their contribution to timbre research, they have also been criticized particularly due to their inefficiency in obtaining similarity spaces for large number of test tones. According to Lakatos (Lakatos, Scavone, Cook 2000) when the number of stimuli (M) is greater than 50, MDS methods use all M(M-1)/2 similarity judgments from a paring of M stimuli. For example, if M = 50, the number of tests will be 50(49)/2 = 1225. This is indeed an enormous figure for a relatively small number of stimuli. Another problem is the number of stimuli that are used for comparison. In other words, in the MDS method, only *pairs* of tones are played for each subtest and subsequent pairs are usually selected in a random order. Hence, this may not necessarily be adequate in comparison to using three or more tones for each similarity test in a desired order. Also, the predominant use of bipolar scales may bias the subject to think in a one-dimensional way, which could possibly be counterintuitive since timbre is multidimensional.

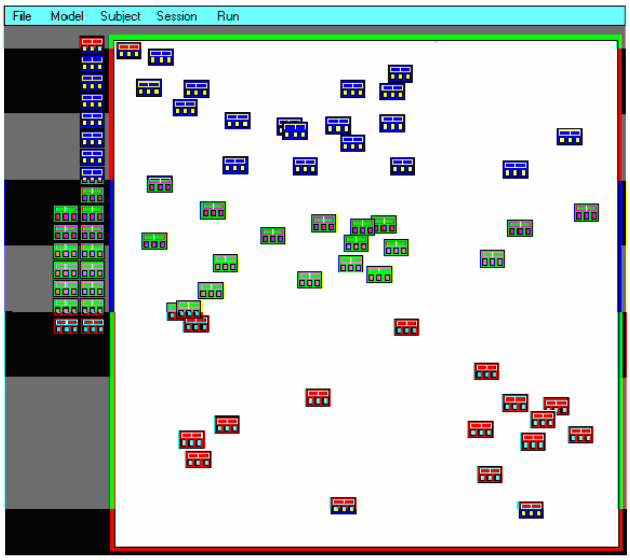


Figure 4.3 Data collection GUI (Lakatos, Scavone, Cook 2000)

Lakatos et al. suggest a more flexible, interactive, and efficient method for collecting data using a 2-dimensional GUI based software (see figure 4.3), in which the subject can move around, play, group, and select a number of sound-icons simultaneously. However, they add that the system "… is not intended as a substitute for traditional MDS analyses; rather, its primary purpose is to collect data using a format that increases participants' motivation …"

4.3 Auditory Scene Analysis (ASA)

Coming from a different angle in understanding sound perception, *Auditory Scene Analysis* puts forward a theory primarily based on the concept of *streaming* and *cues* that occur during the perception of sound sources. This idea comes from the "cocktail party effect" pointed out by E. C. Cherry in 1953 (Cherry 1953). The cocktail party effect may be explained in an environment, such as a

cocktail party where simultaneous conversations transpire at the same time in proximity or further away. As we may have experienced, we are able to maintain simultaneous representations of sound groups in a relatively independent fashion. ASA has in large been studied by Bregman, cumulating to the publishing of the book “Auditory Scene Analysis: The Perceptual Organization of Sound” (Bregman 1990). In the book Bregman attempts to explain psychoacoustic phenomena through intuitive experiments and tries to make models based on ASA concepts.

Streaming generally refers to the tendency to group together sound components that fall into similar frequency ranges. Some of the theories in this area stem from the psychology literature and use such ideas as *good continuation*, *similarity*, and *proximity*. An example of good continuation is a drawing that looks like a big “X,” which we normally perceive as one line diagonally going up (or down) from left to right (or R → L) and the other diagonally coming down (or up). Even if the “X” had been drawn with two triangles (with one side missing) balanced on top of each other, we would still very likely perceive the “X” as constructed with two straight and continuous lines. An example of similarity may be easily explained when grouping similar sounds together, or similar visual objects into belonging to the same group, as in segregating the string section from the brass section in an ensemble, or grouping the kick drum with an open E on the bass (lowest note on a conventionally tuned electric bass). Proximity is readily observable in examples where a listener groups together pitches that are in close proximity,

sometimes used in solo instrument pieces where the composer renders an auditory illusion of having three instruments playing in the upper, middle, and lower registers.

According to Bregman two types of auditory groupings exist, *primitive grouping* and *schema-driven grouping*. Primitive grouping is defined as audio-information driven mechanism and is believed to be an innate grouping scheme. Schema-driven grouping is based on learned knowledge of acoustic environments and in contrast can be regarded as a prediction-driven scheme. The machine implementation of such a system in exploiting grouping rules, cues, and other concepts is called *Computational Auditory Scene Analysis* (CASA). Grouping components together (on micro and macro levels) as coming from the same source is referred to as *fusion* or *coherence*, whereas the opposite phenomenon of grouping sounds as coming from different sound sources is called *fission* or *segregation*. Grouping boundaries occur in many aspects of a sound source, including the onset, offset, common periodicity (pitch), modulation, harmonic content, and spatial location to name a few. Ellis for example, implemented such a *prediction-driven model* (Ellis 1996), whereby the analysis of a sound source proceeds by making a prediction of the observed cues expected in the next frame based on the past frames (hence, the name *prediction-driven* or *top-down model*). The prediction result is then compared to the actual information arriving from the front-end and finally reconciliation between the predicted and actual

information is made. Ultimately, through iteration the internal abstract model is constantly updated and trained with new sound sources (see figure 4.4).

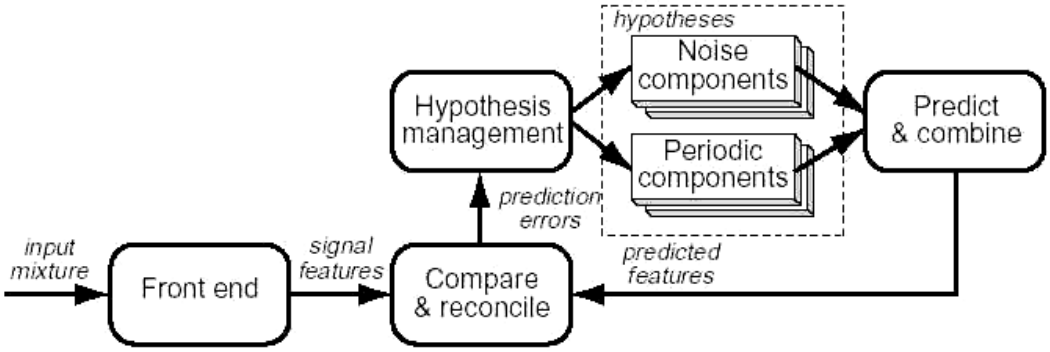


Figure 4.4 Ellis' Prediction-driven CASA system (Ellis 1996)

Chapter 5 Recent Research in Automatic Instrument Recognition

Numerous research topics in sound recognition have been conducted in diverse areas such as discriminating between types of motor vehicles, detecting malfunction in machines and components, pitch recognition, voice recognition, babble recognition (infant vocalizations), face recognition, handwriting recognition, zip code recognition, bird call recognition, recognition of impulsive sounds (glass breaks, door slams, human screams), and monitoring frog communities. However, the majority of machine learning and recognition of sounds have been in the speech research field. The trend may be partially attributed to the huge availability of resources for speech sounds, ease of acquiring such data, and the economical viability and vast application areas of such systems. Hence, it is not surprising that some of the classification, recognition, and signal processing techniques such as LPC and cepstral coefficient also used in musical timbre research come from our friends in the speech community. In this section I will give an outline of past and recent research projects pertaining to automatic instrument recognition by machines.

5.1 Human Recognition Performance of Musical Instrument Sounds

Before going on to machine-based musical instrument recognition systems, I will first briefly discuss the performance of humans in the musical instrument recognition task. It is somewhat surprising that few researchers actually have conducted tests in recognition of musical instrument under *realistic* conditions with human subjects. For example, reported studies have mainly utilized isolated

tones from only a few instrument types, often using a single or a limited pitch range, the same dynamics or a narrow dynamic range. It has been pointed out in the past that transitional information between notes may actually provide substantial information in the perception of musical quality (Saldanha, Corso 1964), and in another study Kendall emphasized the importance of context within musical passages in the perception and recognition of instruments (Kendall 1986). Nevertheless, such studies have been scarce in testing the performance of human recognition of musical instruments. It is commonly believed that trained musicians can “easily” identify instruments upon hearing a musical tone, however the following experimental results suggest that the error rates are perhaps a little higher than one would expect.

Table 5.1 shows a summary of recognition percentages by human subjects. The first 5 tests used single tones and the latter, most recent experiments utilized monophonic phrases (Brown 2001). Houix, Martin, and Brown (Brown 2001) conducted two recent examples of testing human recognition of musical instruments. Houix included short solo instrument excerpts that were played to a group of 15 musicians who were asked to classify 60 samples into arbitrary categories. The number of categories was specified to the subjects in advance. As seen in table 5.1 the best results can be seen in discriminating flute timbres from the rest with a 93 % accuracy, and the worst in clarinet recognition at 71% recognition with an overall 85 % recognition rate.

	Date	Oboe	Sax.	Clar.	Flute	Overall	Number of Instruments
Eagleson/Eagleson	1947		59	45	20	56	9
Saldanha/Corso	1964	75		84	61	41	10
Berger	1964					59	10
Clark/Milner	1964					**90**	3 (flute, clar., oboe)
Strong/Clark	1967a					85	8
Campbell/Heller	1978					72	6 (2-note legato)
Kendall	1986					84	3 (trumpet, clar., violin)
Brown	1999	85	**92**			**89**	2 (oboe, sax.)
Martin	1999					46	27 (isolated tone)
						67	27 (10 sec excerpts)
Houix/McAdams/Brown	2001	**87**	**87**	71	**93**	85	4 (oboe, sax., clar., flute)

Table 5.1 Percent correct for instrument recognition experiments by humans

One test conducted by Martin was based on 10 sec. excerpts of 27 instrument types. The test results for isolated short tones were 46% for individual recognition of instruments and 92% for family recognition (Strings, brasses, double reeds …). Test results with longer excerpts improved the recognition percentage to 67% for single instruments and 97% for instrument families. Hence, it shows that longer excerpts of musical tones actually improve the recognition rate considerably (21% for individual instruments, 5% percent for instrument families). Some interesting observations were made within the families of the instruments, especially in the isolated tone test. For example, viola and violins were sometimes mistaken for each other and some confusion between the flute and piccolo was also noticeable.

Figure 5.1 shows another study done more recently on human performance obtained from 88 conservatory undergraduate and graduate students, as well as

faculty from the Peabody Conservatory of Music using samples from the McGill University Master Samples CDs (Srinivasan, Sullivan, Fujinaga 2002). The isolated tones were extracted from the CDs without alteration which were between 4~7 seconds long. The subjects were presented with instruments categorized into 3 sections (3 different tests) listed in table 5.2 and 1 section (1 test) table 5.3. Each subject was forced to answer all the questions and was given 5 seconds to register an answer. Figure 5.1 shows the results of the tests for ear-training (ET) students, composition (Comp) students, and faculty members.

2-instrument	2-instrument	9-instrument
Oboe	Clarinet	Flute
Sax	Trumpet	Oboe
	Violin	Clarinet
		Bassoon
		Sax
		Trumpet
		Trombone
		Violin
		Cello

Table 5.2 List of instruments divided into smaller groups

27-instrument		
Violin	English horn	Tenor sax
Viola	Bassoon	Baritone sax
Cello	Contrabassoon	Bass sax
Double bass	Eb clarinet	Trumpet
Piccolo	Bb clarinet	French horn
Flute	Contrabass clarinet	Alto trb.
Alto Flute	Soprano sax	Tenor trb.
Bass Flute	Alto sax	Bass trb.
Oboe		Tuba

Table 5.3 List of instrument all in one group

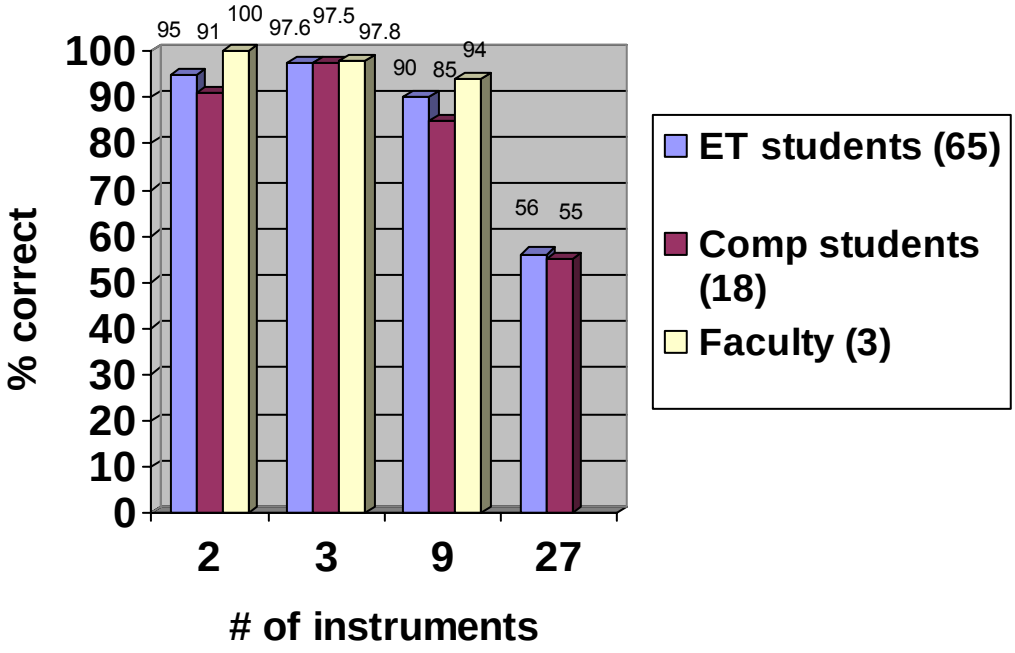


Figure 5.1 Results for 1st test without training

From the experimental results it is interesting to observe that even trained musicians have difficulty in identifying instruments when the number of instrument types are expanded to 27 instruments and asked to choose from them. However, when the number of instruments in a group was decreased to a selected few, the subjects did very well.

Figure 5.2 shows the min, max, and average performance for human subjects from the data presented above. Upon closer inspection of the studies it is rather difficult to come to a clear consensus on how humans do in identifying instrument sounds. The range for individual instrument recognition varies anywhere between 41% to 100% depending on the number of instruments used and the length of each excerpt. Furthermore, in many cases only a few examples of

each instrument was played, using limited techniques and pitch base; and in some cases it is unknown how many examples of each instrument was used to determine the results. The tests do however clearly suggest that instrument family recognition is easier than individual instrument recognition.

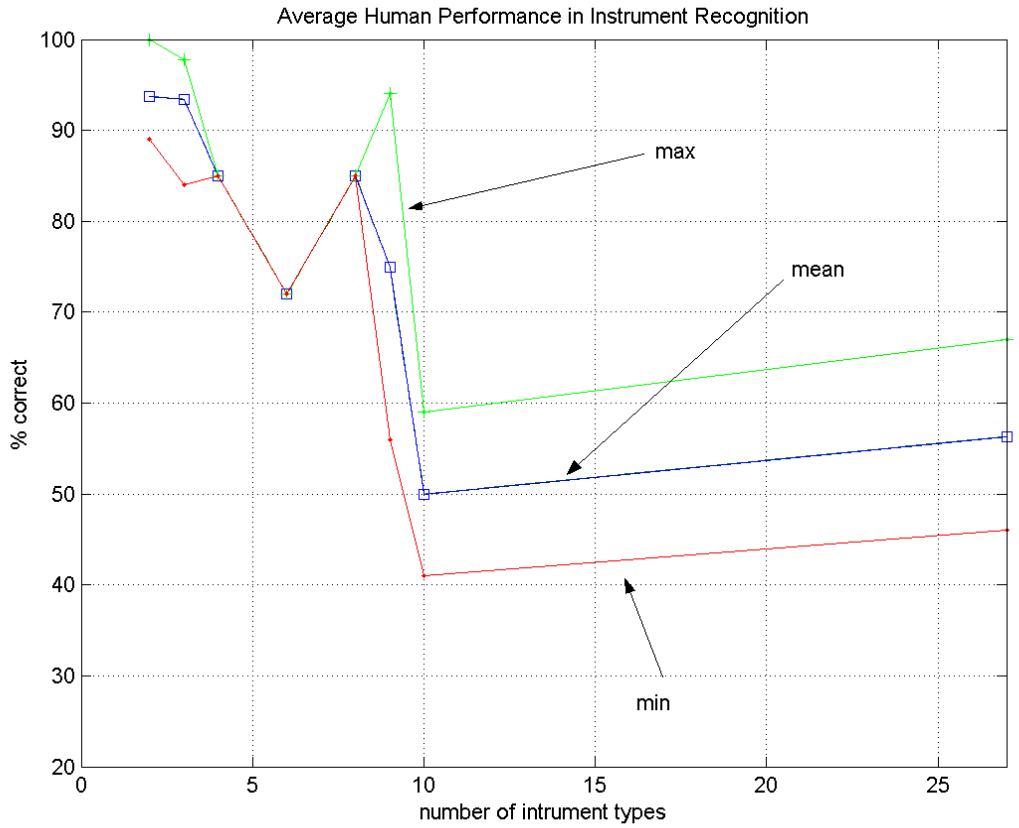


Figure 5.2 Min, max, and average performance for human subjects

5.2 Automatic Recognition Models

Recognition can be defined as a sound object currently being heard and its correspondence to something that has been heard in the past. The final matching of the target sound occurs with some lexicon located in long-term memory (McAdams 1993). Hence, recognition requires some sort of memory

and learning experience. In a machine based recognition system, the memory and learning experience paradigm is usually found in the forms discussed hereafter.

The recognition models for instruments and timbre in machines may be generally divided into two categories, the *top-down model* and the *bottom-up model*. The majority of the recognition research pertaining to timbre has used *bottom-up models* also known as *data-driven models*. Figure 5.3 shows the basic steps: transduction of acoustic signal, pre-processing, feature extraction, training, and classification.

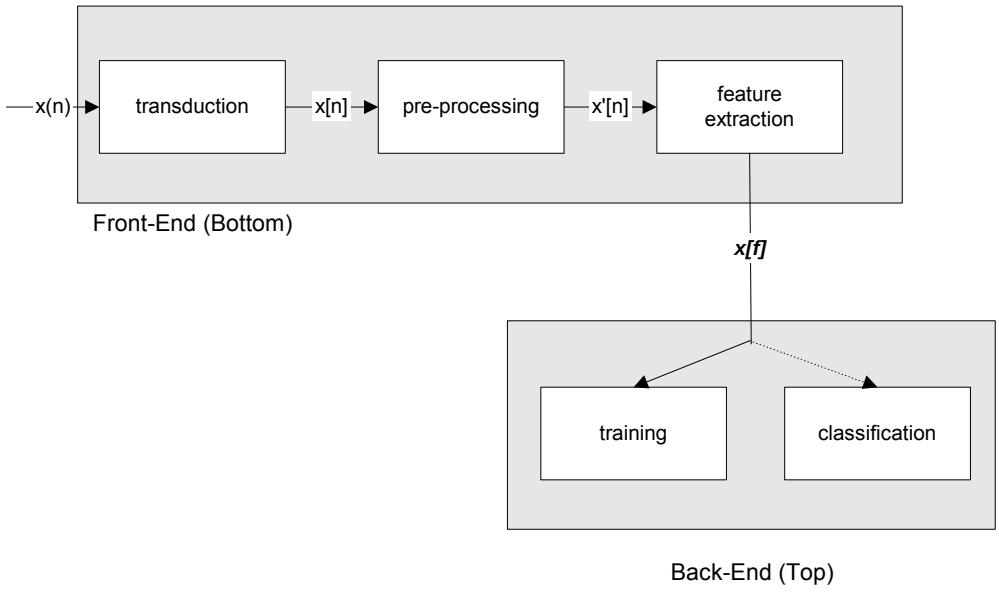


Figure 5.3 Basic bottom-up model

The information flow for the bottom-up model is from the bottom (front-end: transduction, pre-processing, and feature extraction) to top (back-end:

recognition module). The top-down model on the other hand, CASA (Ellis 1996) being the most representative, uses predictive means from internal abstract models (top) that meet features and cues analyzed from the front-end side (bottom) as seen in figure 5.4. The predicted information is reconciled with the actual extracted data and the abstract model is updated via computed errors. However, top-down models cannot exist as pure high-level systems alone as it needs some sort of data, reliable features, and cues from the lower levels.

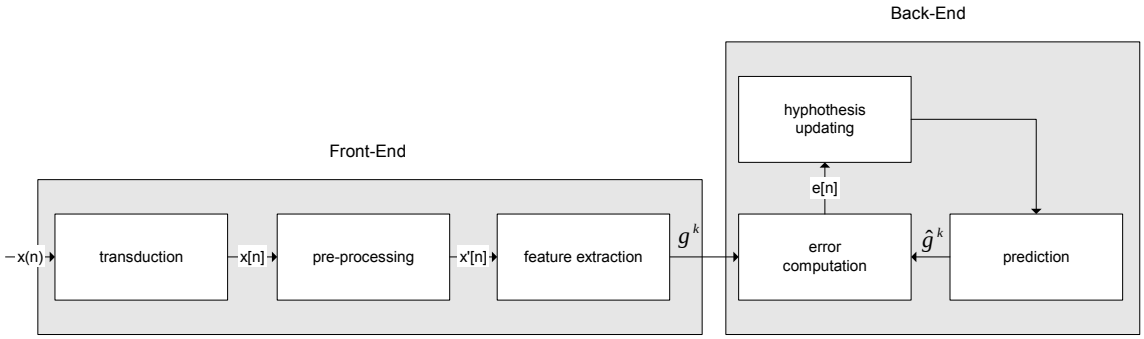


Figure 5.4 Top-down model

Computer systems that implement machine recognition often apply some sort of machine learning technique categorized into *supervised* and *unsupervised learning*. Unsupervised learning systems do not need supervisors: they infer and learn from example data and automatically produce classes. *Clustering* algorithms are a type of unsupervised classifiers that basically assume that patterns of the same class are likely to cluster close together in a given pattern space. An example is the *k-means* algorithm that has found much popularity among researchers. The popularity of k-means may be attributed in part to the ease of its implementation, absence of training (no supervision), and the

advantage of obtaining automatic clusters. K-means which is a distance-measure based clustering algorithm computes distances between pattern samples generally with Euclidian metrics. However, k numbers of classes have to be chosen beforehand and an extensive number of examples must be stored in memory before running the algorithm on a particular input pattern. This is known as *lazy learning*, i.e. storing all its training samples and computing distances between its entire training samples. The major drawbacks of such systems are high sensitivity to irrelevant features, large memory requirements, significant computation load, and unhelpfulness in gaining insight on features. The *k-NN* (k-nearest neighbor) is similar to the k-means algorithm where *k* is the number for "nearest neighboring clusters." When k numbers of "nearest neighbors" are found for a particular pattern, a voting scheme determines the pattern's membership from k number of possibilities. Both methods are often used as rough approximations of the pattern space before application of a fine-tuning algorithm.

Supervised learning on the other hand, as the name implies, requires a *teacher* to help the machine in the learning process. *Pattern recognition* and *discrimination* also refer to this type of system, where the construction of a classification procedure from a set of patterns is achieved with the knowledge of the desired or true classes, hence *training* the algorithm. Pattern recognition in general is categorized as a supervised learning system as it needs a "teacher" to train the system to correctly classify input patterns through features obtained

from the front-end. *Neural networks*, *Gaussian classifiers*, *Hidden Markov Models* (HMM), *Bayesian classifiers*, and *binary trees* are also types of supervised learning systems. In studying the salience of timbral features, the binary tree method is well suited as it results in a taxonomic organization of sound (see section 5.3.3 for details). Each node bifurcates into two new possible nodes where in some cases *average entropy* is used in making its decisions (Jensen, Arnspang 1999). A large number of feature related questions are asked at each node, then the *goodness of split* is computed, and finally the question that renders the best goodness of split is chosen. For example a simple question may be a pattern that satisfies a "rise-time greater than 40 ms AND a centroid less than 400 Hz." The question that renders the smallest average entropy is chosen, or according to Bahl et al. "… seeking questions which minimize entropy is just another way of saying that questions are sought which are maximally informative about the event being predicted" (Bahl et al. 1989). The results from the training phase can then be used to classify new instruments as they become available. With this sort of method insight may be gained by viewing the choices that are made at each node, which may improve our knowledge of the brain's classification procedures.

5.3 Machine Recognition Performance of Musical Instruments

The following is a review of selected classification techniques used by recent researchers. The algorithms and their performances are summarized in table 5.4 (Herrera-Boyer et al. 2000). Although some algorithms seem to perform better than others, upon closer inspection it should be noted that some of the information can be a little misleading: there is insufficient data to fully assess a particular method and hence its accuracy (%).

Method	*Researcher(s)*	*Database size (sounds/classes)*	*Accuracy % success*	*Comments*
K-Nearest Neighbors				Memory intensive/lack of generalization
	Martin & Min (1998) Fujinaga (1998-2000) Ronen & Klapuri (1999)	1023/14 1200/39 1498/30	61 - 79% 68% 75%	family previous to class decision real-time recog.; GA enhanced mixed architecture; + Gaussian classifier
Bayesian Classifiers				
	Martin & Kim (1998) Brown (1999)	1023/14 30/2	71% 85%	
Discriminant Analysis				Fast computation/post-hoc feature selection
	Herrera (unpublished)	120/8	75%	quadratic discriminant functions
Binary Trees	Jensen (1999) Wieczorkowska (1999)	150/5 n.a./18	n/a 68%	
Support Vector Machines				Better generalization than others
	Marques (1999)	esti. 5000/8	70-83%	
Artificial Neural Networks				Very slow training procedure
	Kaminskyj et al. (1995) Kostek (1995-2000) Cemgil et al. (1997)	240/4 n.a.(esti. 120)/4 40/10	97% 90% 94-100%	
Higher Order Statistics				
	Dubnov et al. (1997)	n.a./18	n/a	Details not available
Rough Sets				
	Kosteck (1998) Wieczorkowsak (1999)	n.a. (est. 120)/4 n.a./18	80% 90%	

Table 5.4 Summary of automatic instrument recognition performance

For example, although artificial neural networks (ANN) in table 5.4 seem to outperform all other classifiers (except one of the *rough set classifiers*), the *number* and *type* of instruments employed greatly differ from some of the other studies. The same is true for the rough set results, which report good performance, but the lack of instruments and classes used (4 classes only) do not give an adequate representation of the robustness of those algorithms.

5.3.1 K-Nearest Neighbor

K-NN has been widely used in timbre research with good results considering the simplicity of its algorithm. Martin and Kim used k-NN techniques within a *hierarchical and non-hierarchical* system (Martin, Kim 1998). The hierarchical system refers to a tree, somewhat like the binary tree mentioned before, but with greater number of branches at each node. As seen in figure 5.6, an instrument is first divided between pizzicato and sustained groups then further into strings, woodwinds, and brasses. Martin and Kim believe that in humans recognition of instruments at least partially, happens in this manner: classification of instruments occurs by division according to their features, which become progressively low-level until the bottom of the tree is reached. The k-NN algorithm works by having all the training data in memory while stipulating the “k nearest neighbors” beforehand. Hence, when a sample is to be classified at a certain node, the algorithm finds the “k nearest clusters” and the new sample is given membership according to its multidimensional location and a membership vote. K-NN techniques have been shown to be robust and yield good results.

However, they provide little insight into the relative importance of the various individual features. To alleviate this problem, Martin and Kim implemented a "step-forward" approach. The step-forward method basically steps through each feature individually, leaving the ones that are most salient and discarding those that are most inferior. When all features have been stepped through, a best "current set" of features remains. The algorithm continues by testing all possible combinations of the "current set," adding new salient features and leaving out poor ones while testing the training data. Using the best 10 features on every node, the system's performance improved considerably for instrument family identification. Table 5.4 shows some of the features they used in their experiment.

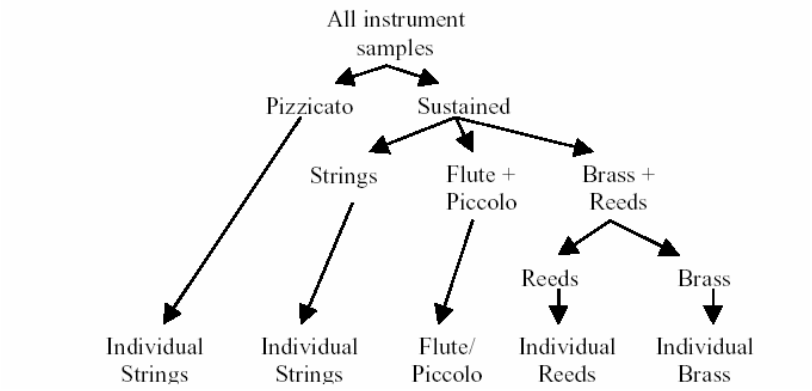


Figure 5.6 Instrument taxonomy (Martin, Kim 1998)

Average pitch over steady state
Pitch variance
Average spectral centroid
Variance of spectral centroid
Maximum slope of onset (dB/msec)
Onset duration
Vibrato frequency/amplitude
Tremolo frequency/amplitude
Spectral centroid modulation
Odd/even harmonic ratio

Table 5.4 Some of the features in Martin and Kim's experiment

Fujinaga also used a k-NN algorithm and applied it to a recent real-time implementation for instrument recognition (Fujinaga 1999). In his system he used a combination of k-NN and *genetic algorithms* (Holland 1975). GAs are often used in solving complex problems that are "impossible" or difficult to solve – by means of "evolutionary survival." The weights are converted into "genes" (bit encoded) and through "evolutionary processes," the strongest genes – those that render the highest recognition rates are allowed to survive. Although, there is no guarantee of the best solution, a near-optimal solution is usually obtained. By changing the weights (ω_i, equation 5.1) in the distance measurements of the k-NN, a change in the feature space occurs. If the changes are manipulated to obtain an optimal solution for clustering, the recognition rate will improve accordingly. The problem then is to find the optimum weights. This may be quite easily done for a two-dimensional system but for features that are 20 or more, the problem becomes extremely difficult to solve.

$$d = \left(\sum_{i=1}^{N} \omega_i \left(x_i - y_i \right)^2 \right)^{1/2} \quad (5.1)$$

Fujinaga reported some considerable improvements by applying GAs but also mentions that the performance is somewhat instrument dependent. Some of the features used in his experiments include the centroid, skewness, and estimated pitch. *N* is the number of dimensions, *d* is the resulting *N* dimensional features space, x_i and y_i are points in the features space.

5.3.2 Bayesian Classifiers

Bayesian classifiers are statistical classifiers that involve a learning step in which the probabilities for the classes, conditional probabilities for a given feature, and a given class are estimated based on their frequencies over the training data. These estimates correspond to the learned hypothesis formed by counting the frequency of various data combinations within the training examples, and can be used to classify each new instance.

One of the difficulties in training an algorithm is deciding which features to include and also simply the number of feature vectors involved. The size of training data increases exponentially with the number of features used, in other words the more features a system employs the more training data it requires. Hence, one of the dilemmas is the reduction of feature vectors in order to gain a manageable testing environment. Martin and Kim (Martin, Kim 1998) employed Fisher multiple discriminant analysis (McLachlan 1992) to lighten the sheer number of training requirements by reducing features. The Fisher technique in their system is used on each node of the hierarchical system which projects higher dimensional features spaces into lower ones. The analysis yields the *mean* feature vector and *covariance matrix* in the reduced timbre space of a Gaussian density for each class, which is used in turn to form a *maximum a posteriori* (MAP) classifier with the introduction of prior probabilities. Table 5.6 shows a comparison of k-NN and Fisher method used in their tests.

	Hierarchical Methods		Non-Hierarchical
	Fisher + MAP	**k-NN**	**k-NN**
Pizzicato vs. continuant	98.8%	97.9%	97.9%
Instrument family	85.3%	79.0%	86.9%
Individual instruments	71.6%	67.5%	61.3%

Table 5.6 Fisher and k-NN method comparison (Martin, Kim 1998)

Other methods for reducing the feature vectors also exist. One is called *Principal Component Analysis* (PCA). PCA involves a mathematical procedure that transforms a number of possible *correlated* components into a reduced number of *uncorrelated* principal components. *Sequential forward generation* (SFG) is another dimension-reducing algorithm that works by starting with an empty set of features. It adds new features by selecting those features which give the best results to the system. The process is iterated until no further improvement is made. *Sequential backward generation* (SBG) is similar to SFG but the procedure is "reversed." The process starts with a full set of features and proceeds by eliminating those features that give the poorest results until no further improvements are made (Liu 1998). The advantage of using the last two methods are obviously the ease of implementation, gaining insight about saliency of features, however, the downside is that the algorithms sometimes converge to sub-optimal solutions.

5.3.3 Binary Trees

Binary trees are constructed in a tree-like manner where each node has the ability to branch into two newer nodes. The binary tree is built by asking various

"questions" at each node during the training phase. A sound thus gets classified into two possible groups at each consecutive node until the bottom of the tree is reached. The "goodness" of each split occurring at each node is determined by the *questions asked* and the *average entropy*. That is, the best split is determined by selecting the question that minimizes the average entropy of the system. All possible combinations of questions (feature thresholds) are asked at each node and the best sets of questions (feature thresholds) that minimize the average entropy remain, thus completing the tree. As seen in figure 5.7 each node consists of a number of questions based on spectral and temporal features (Jensen 2001).

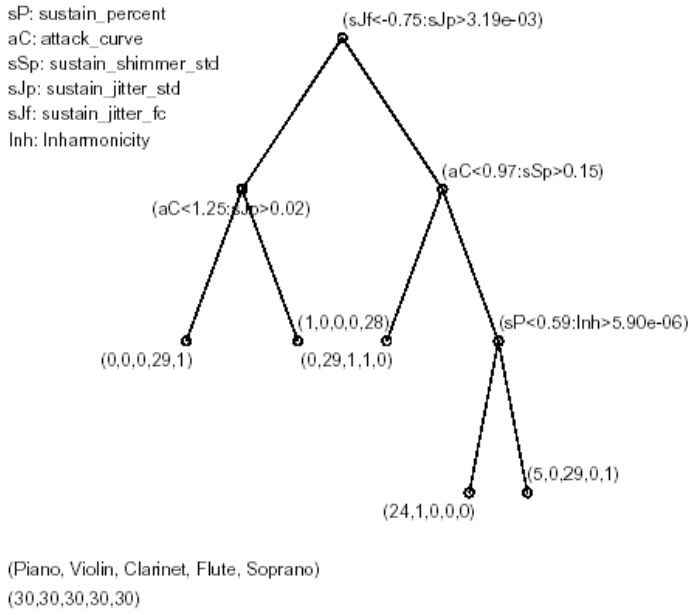


Figure 5.7 Binary tree (Jensen 2001)

Once the tree is built it can be used to classify new sounds. Also, the binary tree is helpful in tracing timbre and feature selection related decisions that are made during the classification procedure, which may give some possible clues about the decision making processes in our hearing system. Furthermore, if some misclassification occurs the tree structure helps in visualization of decision-making paradigms, which may help tweak existing features or add new ones to the respective nodes. Some of the features used in this test were attack curve, shimmer, jitter, inharmonicity, and spectral centroid.

$$H(X) = \sum_{j=1}^{N} \left[p_j(x) \log_2 p_j(x) \right] \quad (5.2)$$

The entropy *H(X)* is defined as shown in equation 5.2 where *p(x)* is the probability density function of sample *x*, and *j* is the index of the sample *x*.

5.3.4 Artificial Neural Networks (ANN)

ANNs, also referred to as *connectionist architectures, parallel distributed processing*, and *neuromorphic systems*, are information-processing systems inspired by the processing model of the human brain. ANNs try to mimic some of the observed properties of biological neural behavior and draw on the analogies of adaptive biological learning. They are composed of a large number of highly interconnected processing elements that are similar to neurons and are tied together via *weighted connections* comparable to synapses. Training the system essentially occurs by example in iteration with input and known output or target

values resulting in the strengthening of connection weights (synapses). The connection weights essentially "store" the knowledge necessary to solve a given problem, in our case instrument classification. Both supervised and unsupervised or self-organizing ANNs exist which form clusters automatically.

One of the most widely used flavors of ANNs is the *Multi-Layered Perceptron* (MLP). MLPs learn through input examples, desired output targets, and error criteria. The error is computed by comparing the output of the network to a known response. *Backpropagation* (BP) algorithms are frequently used to make subtle adjustments to the weights in order to minimize the error. This sequence is iterated until an optimum set of weights is found. ANNs work best in situations when solutions are extremely difficult to obtain (similar to GAs). Some of the drawbacks with ANNs are the tedious tweaking of the parameters, computationally demanding nature during the training phase, and over-fitting (use of excessive bad examples) which can lead to degradation of generality. However, once the network has learned, classification is done in a fast and rapid manner. See chapter 7 for details on ANN.

Kaminskyj et al. used backpropagation in conjunction with a k-NN algorithm and had impressive results: 97% accuracy (Herrera-Boyer et al. 2000). Although the ANN community reported notable results, concrete assessments of the algorithms are difficult to make as only a limited database and single octave pitch range was used in one example (40 sounds from 10 classes).

In this chapter I will elaborate on existing and developed feature extraction algorithms in both the time and frequency domain. Feature extraction is one of the most critical modules in a pattern recognition system as consistent, accurate, and robust algorithms are conducive to enhancement of the pattern recognition module and hence the performance of the system as a whole. Bottom-up based recognition architectures are especially prone to fall into GIGO (garbage-in-garbage-out) loopholes, and as consequence many researchers have found the prospect of improving robustness of feature extraction algorithms a fertile area for increasing pattern classification performance. Approaching the problem of automatic timbre recognition with substantial weight given to feature extraction in a sense follows the paradigm that people with "better" ears have the potential to "better" recognize sound objects with everything else being equal.

6.1 Frequency Domain

Most of the research in the frequency domain analysis has been based around the *Fourier transform*. This may be due to the fact that the hearing system on one hand seems to do frequency analysis much like the Fourier transform. Specifically, the *Discrete Fourier Transform* (DFT) version and its efficient *Fast Fourier Transform* (FFT) version comprise the backbone of many studies in the frequency domain. One well-known example easily observed in the spectral domain is the absence or existence of even and odd harmonics in a signal. It has been found that when the dominant frequency components lie on the odd

harmonic path of a spectrum, a *nasal* character is heard. This has been observed primarily in the *lower* registers of clarinet spectra, where odd harmonic dominance renders a *hollow* sound (Backus 1976). The dominance of odd harmonics is also a feature of "square waves" which can be represented as the sum of odd harmonics with a decrease in amplitude for each harmonic by $1/A_j$ where A is the amplitude, $k = 2*j + 1$ the odd harmonic number, and $j = 0, 1, 2$ *…etc*.

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N} \quad (6.1)$$

The DFT as shown in equation 6.1 is of course the "slow" Fourier transform version where *X[k]* is a complex number with magnitude and phase components at frequency *bin k,* DFT length *N* (usually equal to the length of the window), *sampled* input signal *x[n],* and discrete time index *n*. The *Long Term Average Spectra* (LTAS) shown in equation 6.2 is a good method for viewing the average spectra of sounds as the background noise is averaged towards 0 and the steady state (if present) is accentuated and made clearer.

$$LTAS = 1/J \sum_{j=0}^{J-1} Xj \quad (6.2)$$

J is the number of frames, j the frame index, X_j the j^{th} frame's spectral vector containing magnitude and phase components. However, when using LTAS,

much transient information is lost, depending on analysis window size and DFT size. When applying the DFT, care should be taken in the selection of parameters as a compromise between frequency and time resolution always exists: increased time resolution (transitory characteristic) leads to a degraded frequency resolution resulting in frequency smearing. One way to elude this problem is using the *Short-Time Fourier Transform* (STFT) suggested by Allen (Allen 1977, Allen and Rabiner 1977). The STFT shown in equation 6.3 can be simply thought of as windowing a signal, but rather than advancing or *hopping* the starting point of the signal *x[n]* by the window size, windows are *overlapped* and advanced depending on the overlap length as seen in figure 6.1. In effect, this lessens some of the degradation of *time-frequency smearing* and is applied in most DFT-based spectral analysis practices.

$$X[k] = \sum_{n=0}^{N-1} w[k - mD]x[n]e^{-j2\pi kn/N} \quad (6.3)$$

Also different window types such as rectangular, Hamming, Blackwell, and others exist for extracting a slice of a signal. Each window has its own particular shape which determines the side-lobe characteristics. Please refer to appendix section A.6 for details.

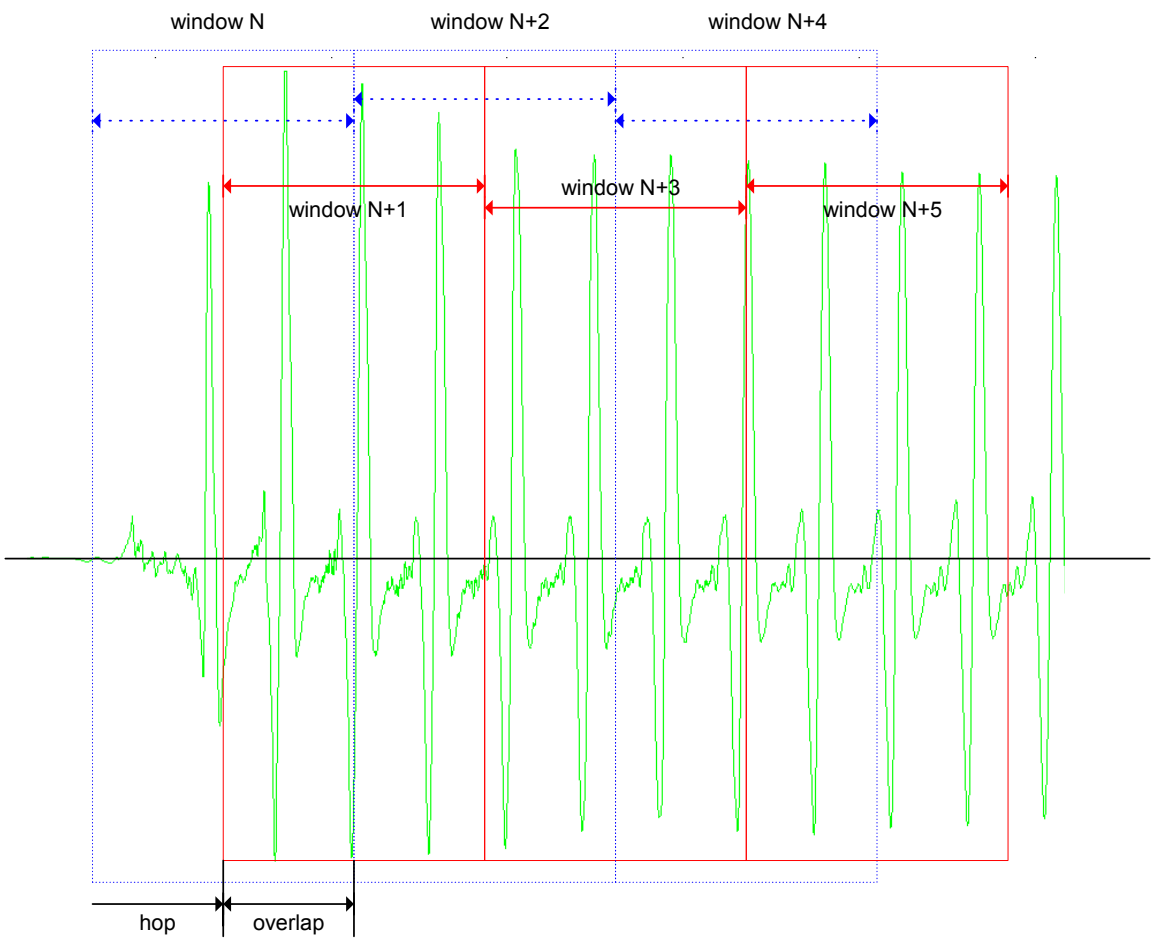


Figure 6.1 Window hop and overlap in STFT

6.1.1Harmonic Analysis

One of the most prominent features from spectral analysis has to do with the behavior of harmonics and indeed the knowledge of locations of harmonics themselves. Although harmonic analysis pertains primarily to feature extraction for pitched instruments; it is important as a large number of instruments are based on pitch-producing resonant structures. As we shall see in this section, armed with the information regarding harmonics and their behavior, features such as *shimmer, jitter, inharmonicity, spectral envelope, harmonic synchronicity,*

and *tristimulus* characteristics may be extracted. The harmonic analysis algorithm developed in this thesis is an enhanced version of the author's previous harmonic analyzer algorithm (Park 2000), essentially making the current analyzer more robust to the vastly different number of playing styles, dynamics, and pitches that change the harmonic structure of a musical tone.

The harmonic analysis algorithm is basically divided into three sections (figure 6.2). The first section uses LTAS mean spectrum as a guide for searching the most salient harmonics and determining the harmonic length (distance between harmonics). Using an averaged spectrum is helpful in guiding the search process as it suppresses noisiness and brings out peaks that are salient. Naturally, mean spectra loose transient information – subtle differences in harmonic locations and magnitudes which is essentially what is desirable during the initial stages of this algorithm. The second stage comprises of separately analyzing for peaks using both log LTAS and linear LTAS spectra. After determining the harmonic length (distance between harmonics) and using the results from the log and linear LTAS peak analysis, ideal locations of harmonics for the LTAS spectrum are determined. The results at this stage can be used to compute inharmonicity as explained in section 6.1.2. Once the harmonic length is computed, a search for harmonics pertaining to each spectral frame in the STFT vector is performed. Furthermore, by applying an "adaptive harmonic length" updating method, it is possible to better track harmonics that compress or expand as harmonic numbers increase towards Nyquist or decrease towards DC.

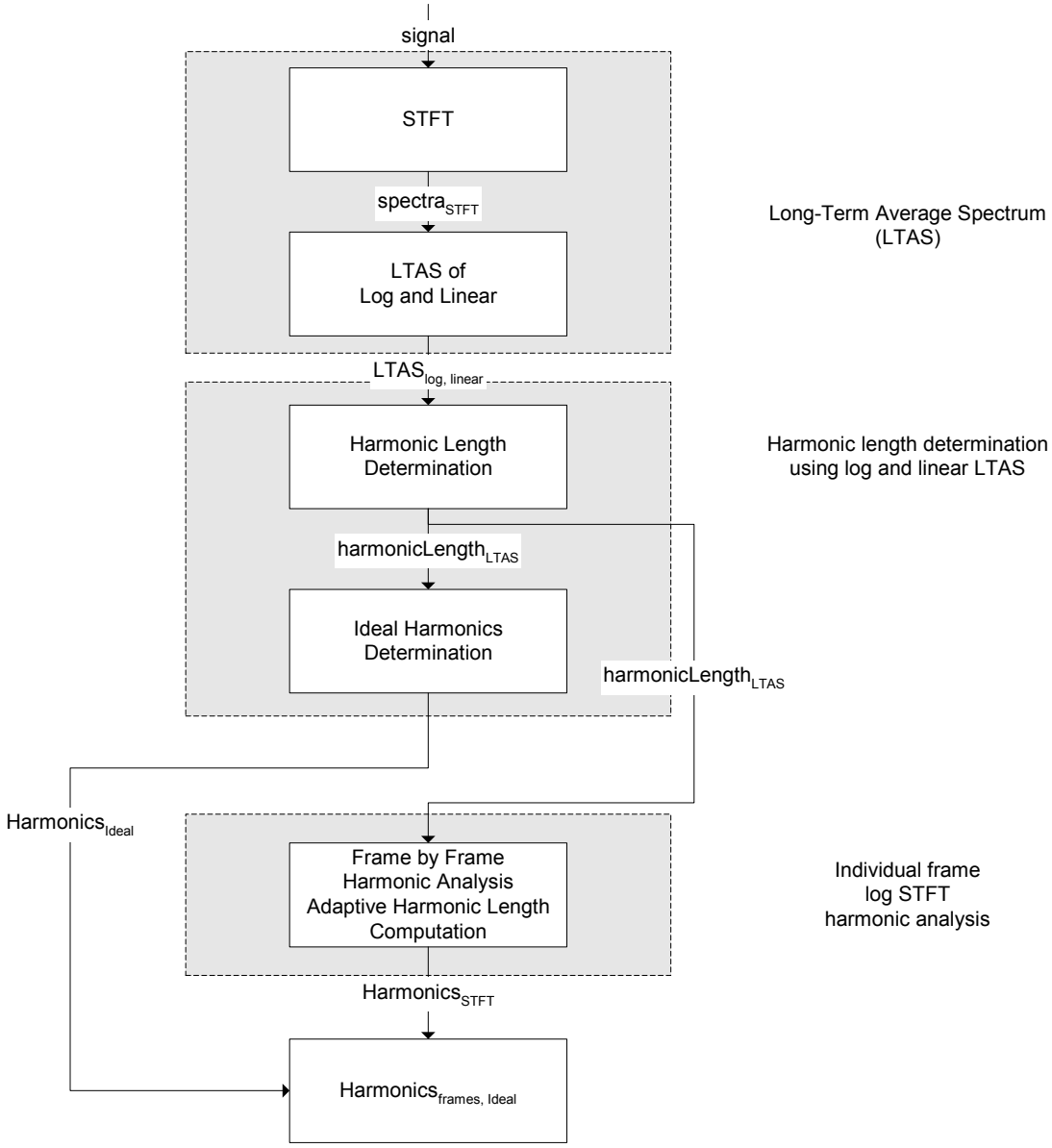


Figure 6.2 Basic flow for Harmonic Analysis

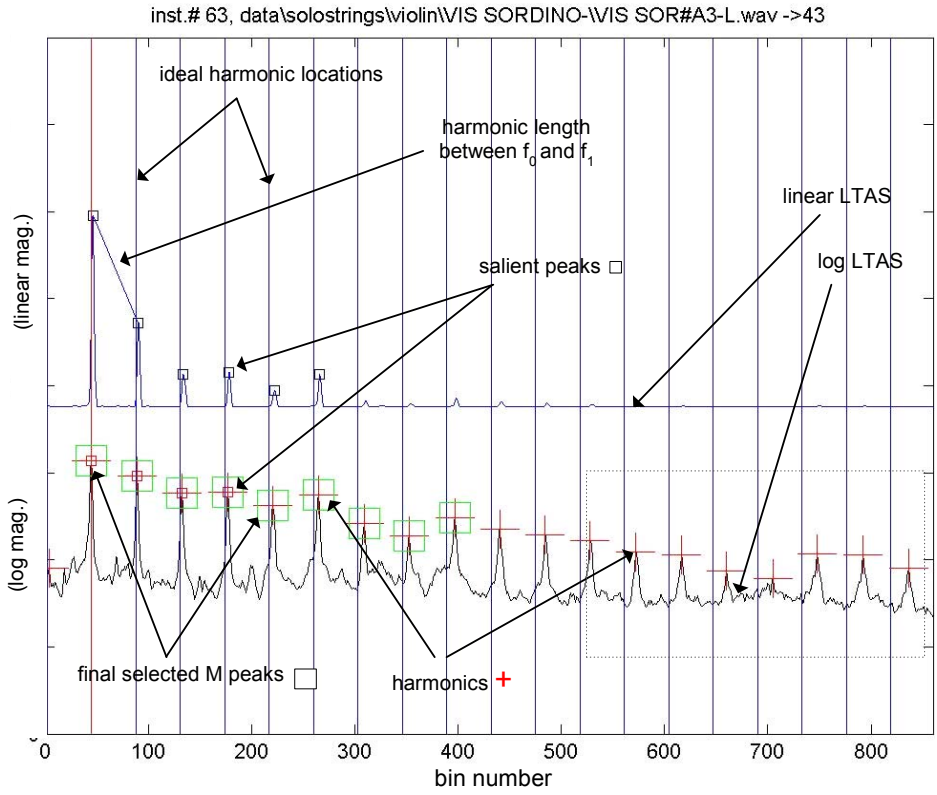


Figure 6.3 Harmonic analysis

Figure 6.3 shows a violin sample and its harmonics obtained through the harmonic analysis algorithm. As we can see, as the harmonic number increases towards Nyquist an expansion of the harmonic locations can be observed. Using a static harmonic length and search area window to determine each harmonic location would be less than ideal as the search area and harmonic length often changes dynamically. However, using the proposed method of adaptive harmonic length along with the updated harmonic locations, it is possible to take into account the contraction or expansion (expansion in this example) of harmonics in a spectrum.

The harmonic length is updated using:

$$harmonicLength = abs(currentHarmonic - lastHarmonic) \quad (6.4)$$

The search area is defined by its left boundary and right boundary where the threshold value *thresh* is a fractional number less than 1.0:

$$nextHarmonicNum = currentHarmonicNum + harmonicLength \quad (6.5)$$

$$leftBound = nextHarmonicNum - thresh * harmonicLength \quad (6.6)$$

$$rightBound = nextHarmonicNum + thresh * harmonicLength \quad (6.7)$$

Figure 6.4 shows a zoomed-in section of the violin spectrum and a problem that arises when using a static harmonic length and search area window measure. Overlapping windows in some case will introduce incorrect selection of harmonics which may result in omission or plainly erroneous picking of a peak that is not a harmonic. However, when using the dynamic harmonic length method the search area will adapt accordingly and help in selecting accurate harmonics. Please refer to appendix section A.8 for details on algorithm.

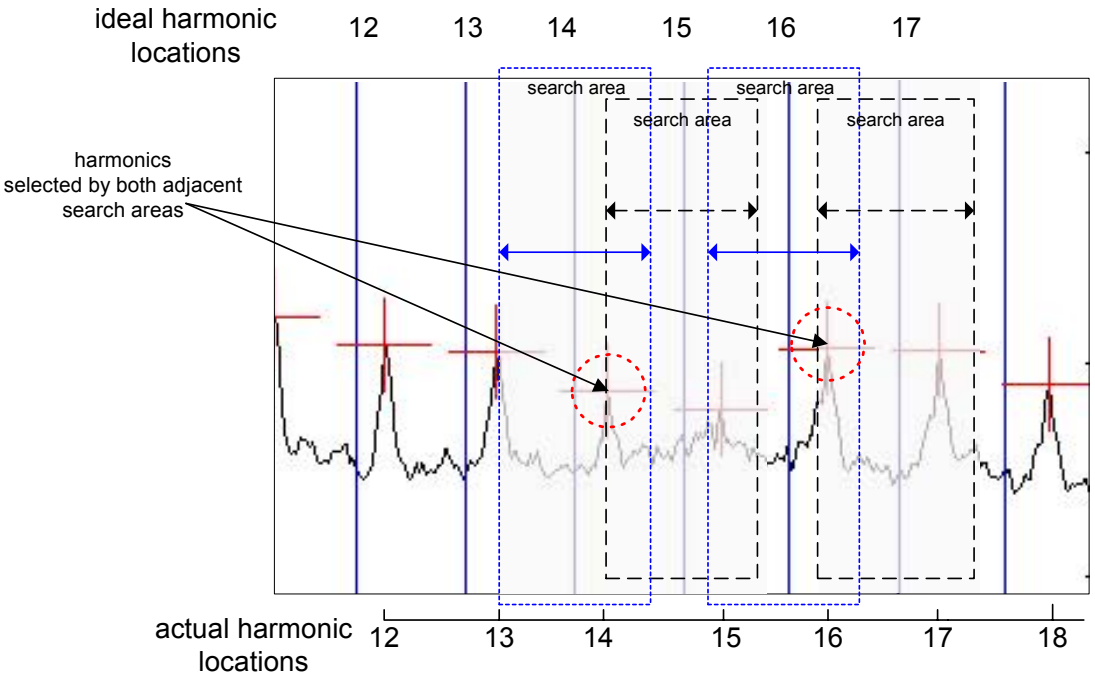


Figure 6.4 Static harmonic length and search area

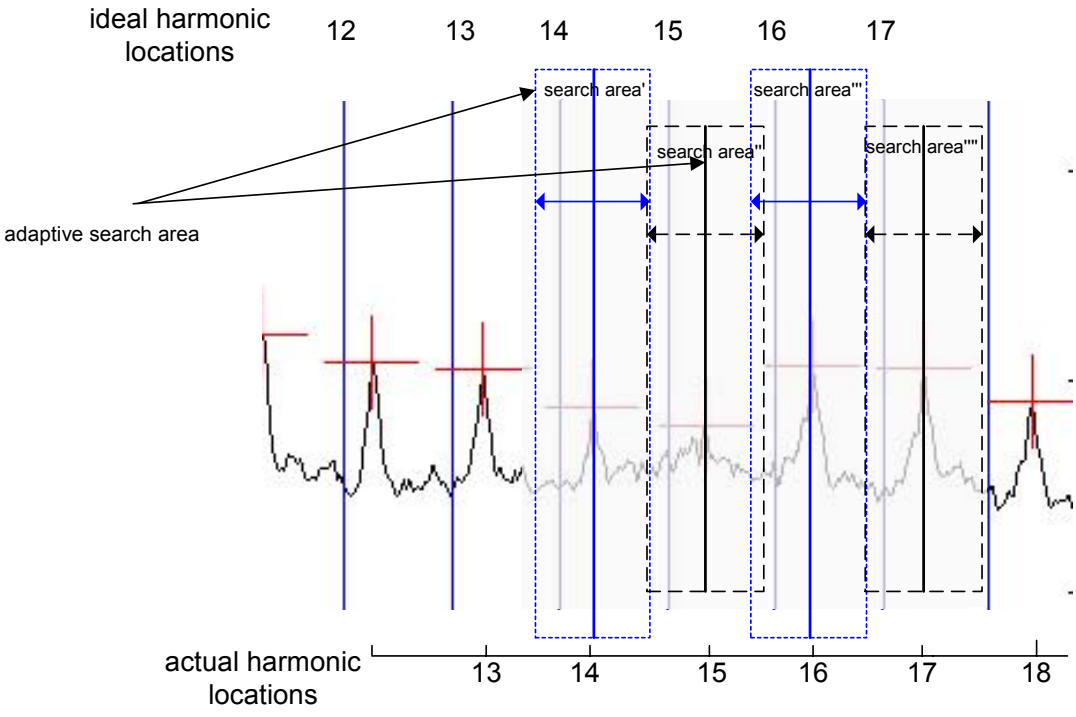


Figure 6.5 Adaptive harmonic length and search area

6.1.2 Inharmonicity

Inharmonicty can be described as the error between measured harmonics and their theoretical ideal harmonics. It may defined as:

$$h_n = \frac{\left| f_n - nf_0 \right|}{nf_o},\ n \in [2, N] \quad (6.8)$$

N denotes the total number of harmonics, *n* the current harmonic number, f_n the n^{th} harmonic frequency component and f_0 the fundamental frequency. The net inharmoncity of a given frame can then be computed as the sum of each component with respect to the fundamental denoted as *H* in the following equation*:*

$$H = \sum_{n=1}^{P} h[n] \quad (6.9)$$

Inharmonicity is observable for example, in stringed instruments such as the piano or electric guitar where the stiffness of the strings causes inharmonicity especially with higher partials. The stiffness together with string tension contributes to the restoration of string vibration to their rest state.

6.1.3 Harmonic Expansion/Compression

As we have seen in figure 6.3 compression and expansion of harmonics are sometimes noticeable in spectra. Harmonic expansion and compression feature

represents a method which will give information regarding how much the harmonic series is expanding or compressing with one number – its slope. Figure 6.6 depicts this idea where the middle *"y=x"* with slope 0.5 is the point of reference. Hence, harmonics that are expanding would have a slope that is greater than 0.5 and harmonics that are compression would have a slope smaller than 0.5, where the extent of expansion and compression would be dictated by slope m.

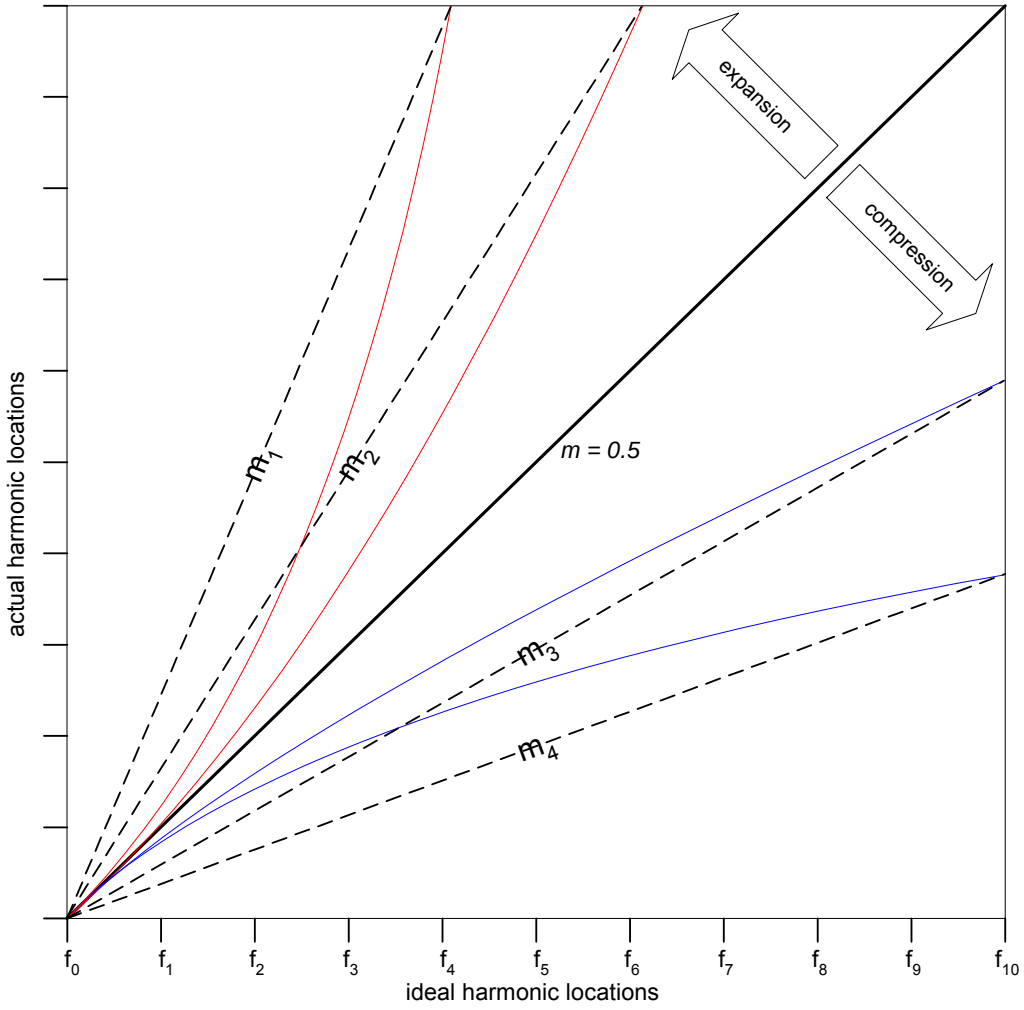


Figure 6.6 Harmonics expansion/compression curves

For a more precise measure of the expansion/compression characteristics, line-fitting algorithms may be applied instead of a single scalar representation.

6.1.4 Harmonic Slope

Harmonic slope represents a measurement of the spectral envelope using the computed harmonic structure of a sound. The idea for this feature was arrived at in trying to come up with one number to represent the spectral envelope. As we will see in chapter 8, it has had some success as a salient feature for both family and individual instrument classification.

6.1.5 Shimmer and Jitter

Shimmer and *jitter* refer to short-time, subtle irregular variations in the amplitude envelope and frequency envelope of spectra respectively. Figure 6.7 shows a shimmer plot for 9 harmonics of a French horn sample. Note that the “random” amplitude modulation is especially present in the higher harmonics and the lower harmonics have virtually no shimmer properties. In the case of jitter for the same French horn sample there seems to be little irregularity for all of the harmonics. As a matter of fact all of the harmonics 3 and 6 are very stable.

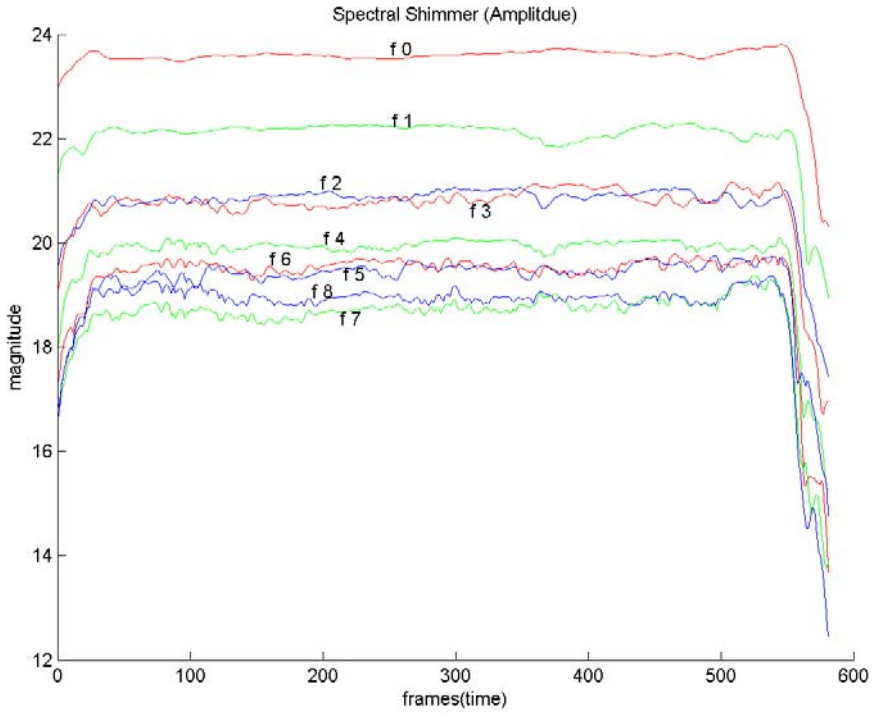


Figure 6.7 Shimmer

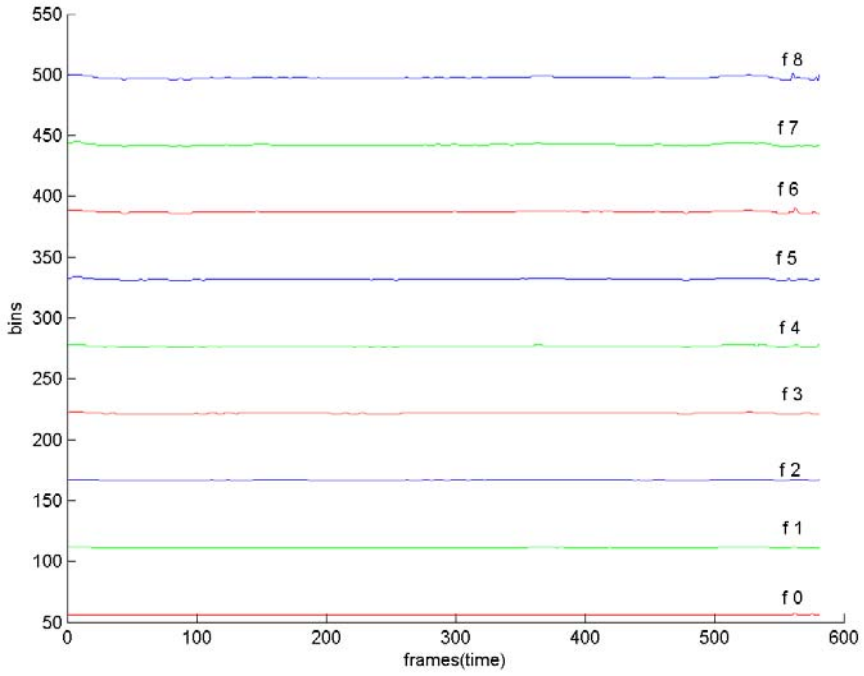


Figure 6.8 Jitter

Jitter is also characteristic in instruments such as the string family. For bowed strings, the unstable interaction between the bow and string – constant action of micro-level pulling and releasing, results in frequency *jitter.*

An alternate approach for computing jitter and shimmer with a 24 sub-band bark scale (see appendix for details on bark frequency scale) may be used. Generally, each band corresponds to one harmonic but if more than one harmonic is found in one of the 24 bands, the average of the harmonics and center frequency of that particular band is taken. Shimmer and jitter, which are characteristics of noise, are believed to have Gaussian normal distribution.

6.1.6 Spectral Envelope

The spectral envelope, which embodies a wealth of information, is really a "zoomed-out" view of the power spectrum. What determines the shape of the envelope is basically the location of dominant peaks on the frequency scale. As described in the section 4.2 one surprising finding by Grey was that through line-segment approximation of the envelopes (Grey 1977), he was able to get very good re-synthesis results of the original tones with a 100:1 data reduction. On the distribution of the harmonics, it has been suggested that no harmonics higher than the 5^{th} to 7^{th}, regardless of the fundamental frequency, are resolved individually. Studies have shown that the upper harmonics rather than being perceived independently are heard as a group (Howard, Angus 2001). Further support for this phenomena is made by Hartman who, according to Puterbaugh

(Puterbaugh 1999), suggests that for a signal with fundamental frequency below 400 Hz, only the first 10 harmonics play an individual role: harmonics greater than 10 affect the timbre *en masse*.

Numerous methods exist in determining the spectral envelope. One method is by salient peak detection of the power spectrum. This method is summarized in figure 6.9, where first the power spectrum is computed via the FFT, followed by salient peak picking and finally interpolation.

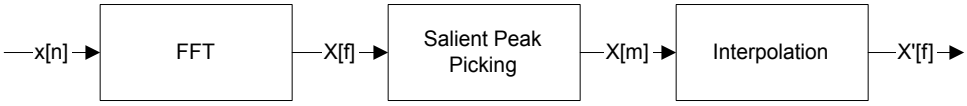


Figure 6.9 Peak detection method for spectral envelope

6.1.7 Synchronicity

Synchronicity is defined as the degree of time alignment of upper harmonics. Synchronicity in the onset part of a sound can be clearly observed in many acoustic instruments. For example, in woodwind instruments in general the starting time of the fundamental occurs first, followed by the 2^{nd} and 3^{rd} harmonics (Howard, Angus, 20001). Beauchamp (Beauchamp 1969) also made some observations in synchronicity and reported that higher harmonics usually needed greater attack durations than the fundamental frequency and that the fundamental intensity had strong correlations with the intensity of a tone. The general finding that lower frequencies to peak before higher ones was backed by Risset and Mathews (Mathews, Risset 1969) who noticed similar results while

investigating the trumpet in close scrutiny using analysis and synthesis methods (verification via aural feedback using a virtual trumpet model). However, this does not mean that the upper harmonics in the attack portion are always lagging. For example, in overblown organ pipes 2nd harmonics start earlier than the fundamental, which can be explained as an initial jump of the fundamental to the 2nd harmonic (Howard, Angus, 20001). One method of measuring synchronicity may be to monitor lower harmonics individually (as mentioned before, it is believed that harmonics above the 7th ~10th are not heard separately), and upper harmonics (greater than 7th ~ 10th harmonic) heard together as a group.

6.1.8 Tristiumulus

The *tristimulus* measure has been suggested in the early eighties (Pollard, Jansson 1982) as a timbre equivalent to the color attributes in vision (based on the peripheral encoding by the human eye). The tristimulus theory of color perception in vision research seems to imply that any color can be obtained from a mix of the three primaries, red, green, and blue. Although most visible colors can be represented using RBG, some cannot (Goldstein 1989). Tristimulus is defined by the following three equations.

$$z = tristimulus \quad 1 \quad = \frac{H[1]}{\sum_{k=1}^{N} H[k]} \quad (6.10)$$

$$y = tristimulus \quad 2 \quad = \frac{H[2] + H[3] + H[4]}{\sum_{k=1}^{N} H[k]} \quad (6.11)$$

$$x = tristimulus\ \ 3\ \ = \frac{\sum_{k=5}^{k=N} H[k]}{\sum_{k=1}^{N} H[k]} \quad (6.12)$$

Where *H[N]* is the upper most harmonic, and *k=1* refers to the fundamental component. Figure 6.10 shows a diagram where the lifetime of tone is traced through the tristimulus diagram using the x and y dimensions. The circles are the approximate steady-state portions which start out at the other end of the line.

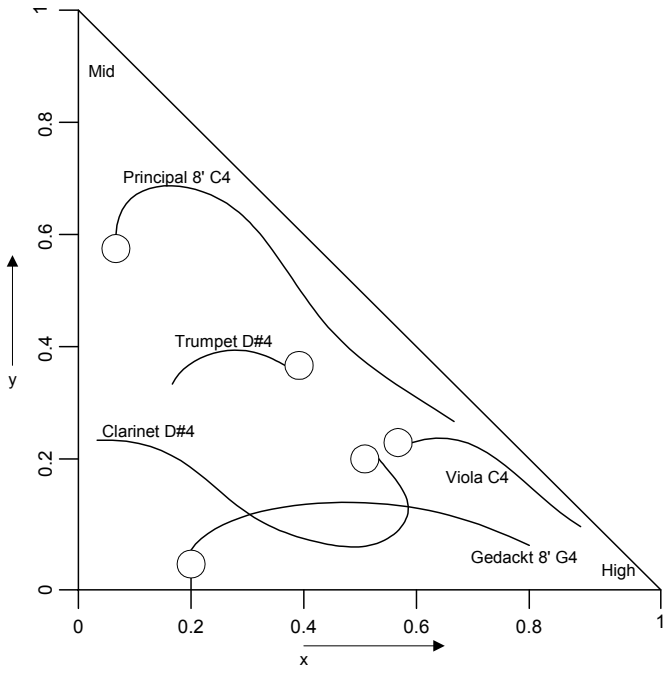


Figure 6.10 Example of a 2-dimensional tristimulus plot (Howard, Angus 2001)

6.1.9 Spectral Centroid

The *spectral centroid* discussed in previous sections corresponds to a timbral feature that describes the brightness of a sound. This important feature has

been elicited in the past (Helmholtz 1954, Lichte 1941) and experimentally observed in MDS based investigations (Krumhansl 1989, McAdams et al 1995, Lakatos 2000). The spectral centroid can be thought of as the center of gravity for the frequency components of a signal. It exists in many variations including its mean, standard deviation, square amplitudes, log amplitudes, use of bark frequency scale (Sekey, Hanson 1984), and the harmonic centroid (see appendix for details about bark frequency scale). During the lifetime of a sound, the centroid changes as seen in figure 6.11 and furthermore as one might expect, it varies characteristically with intensity (it has been suggested to use ratio of centroid to intensity by some researchers). For example, a trumpet blown in middle C with pianissimo and forte dynamics result in different spectral centroid characteristics. The centroid, currently one of the MPEG-7 timbre descriptors, is defined as:

$$SC_{Hz} = \frac{\sum_{k=1}^{N-1} kX[k]}{\sum_{k=1}^{N-1} X[k]} \quad (6.13)$$

X[k] is the magnitude corresponding to frequency bin *k, N* is the length of the DFT and *SC* is the spectral centroid in Hertz. Generally, it has been observed that sounds with dark qualities tend to have more low frequency content and those with brighter sound dominance in high frequency (Backus 1976) which can be inferred by the value of the centroid. McAdams (McAdams 1995) found this

trait by summing up to 30 harmonics for the computation of centroids and calculated it with 12 ms windows.

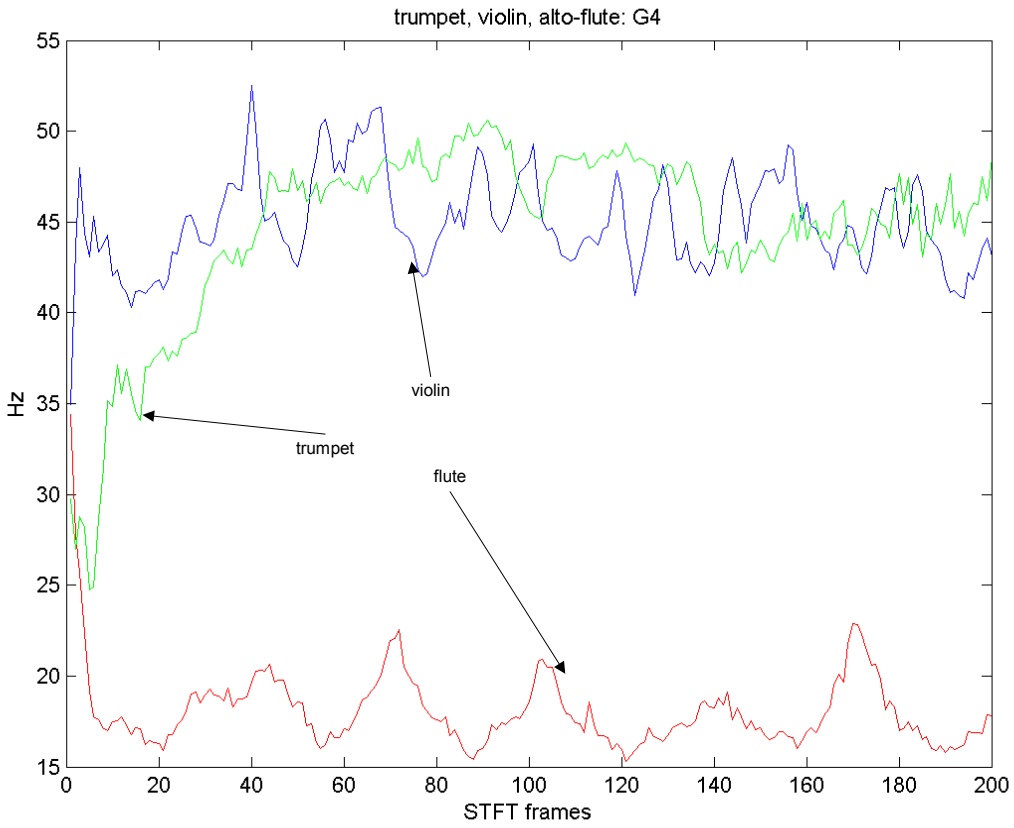


Figure 6.11 Spectral centroid of alto flute, trumpet, and violin: fortissimo G4

It has also been suggested (Kendall, Carterette 1996) that the centroid be normalized in pitch hence making the spectral centroid a unit-less and *relative* measure since it is normalized by the fundamental frequency f_0. Some researchers have therefore included both the normalized and absolute versions of the centroid (Eronen, Klapuri 2000).

$$SC_{relative} = \frac{\sum_{k=1}^{N-1} kX[k]}{f_0 \sum_{k=1}^{N-1} X[k]} \quad (6.14)$$

However, with the addition of f_o it is only useful in the context of *pitched* sounds as purely percussive or inharmonic sounds usually are absent of a fundamental frequency. Furthermore, for obvious reasons, a precise pitch-extraction algorithm has to be devised for accurate centroid measurement which is not a trivial problem to solve.

6.1.10 Spectral Irregularity

Spectral irregularity (Krimphoff 1994) also referred to as *spectral smoothness* (McAdams 1999) basically shows the irregularity of a signal usually computed with the STFT where the average of the current, next, and previous amplitude values are compared with the current amplitude value. Bregman (Bregman 1990) remarks that the smoothness of a spectrum promotes integration of partials to a same source and a single higher intensity partial is more likely to be perceived as an independent sound. It has also been found to be useful in revealing complex resonant structures of string instruments.

$$SI = \sum_{k=1}^{N-1} \left| 20\log(X[k]) - \frac{20\log(X[k-1]) + 20\log(X[k] + 20\log(X[k+1]))}{3} \right| \quad (6.15)$$

A recent modified version also exists and is expressed as in equation 6.16.

$$SS = \frac{\sum_{k=1}^{N-1} (X[k] - X[k-1])^2}{\sum_{k=1}^{N-1} X[k]^2} \quad (6.16)$$

Unlike the conventional spectral irregularity algorithm equation 6.16 is different that it highlights the power of the spectrum due to the nonlinear square operator.

6.1.11 Spectral Flux

The *spectral flux* defines the amount of frame-to-frame fluctuation in time. It is computed by the *2-norm* difference between consecutive STFT frames.

$$SF = \left\| X[f] - X_p[f] \right\| \quad (6.17)$$

where the general *q-norm* is:

$$\left\| X[f] \right\| = \left(\sum_{k=0}^{N-1} X[k]^q \right)^{1/q} \quad (6.18)$$

X[f] denotes the magnitude components of frame *f* and $X_p[f]$ the previous *(p)* frame's magnitude components of same vector size as *X[f]*. *SF* also known as the *delta magnitude spectrum* has also been used to discriminate speech and musical signals. It exploits the fact that speech signals generally change faster than musical signals, noting that in human speech there is a constant game of

ping-pong being played between consonants and vowel sounds. In musical signals however, drastic changes tend to vary on a lesser degree.

The spectral flux computation in this dissertation was implemented in two ways. The first method was just a computation of the overall flux whereas the 2nd method divided the total STFT frames into two groups – the first half reflecting more of the attack part and the 2nd half reflecting more of the steady-state portion. The difference of the two norms was computed and used to reflect a *spectral flux shift.*

6.1.12 Log Spectral Spread

The *log spectral spread* obtained as a salient dimension from MDS experiments with sonar sounds (Tucker 2001) is defined as the energy found between bounded pairs of upper and lower frequencies. The "left and right" bounds are found by applying a threshold value with respect to the maximum amplitude of the spectrum (e.g. –10 dB off the maximum) followed by locating the upper and lower frequency bounds, and finally taking the log of the result. The spread is somewhat similar to the spectral distribution found by Grey (Grey 1977) and is also compared to the *richness* of a sound, however no attempts have been made to quantify this factor. The left-bounded frequency by itself has also been detected as a prominent dimension in his timbre space. Spectral spread may be helpful along with envelopes to observe qualities of instruments such as trombone, French horn, and tuba which generally lack in high frequency content,

whereas the trumpet primarily due to its brightness is rich in upper harmonics (Howard, Angus, 2001). The spectral spread has also been specified as one of the MPEG-7 audio descriptors.

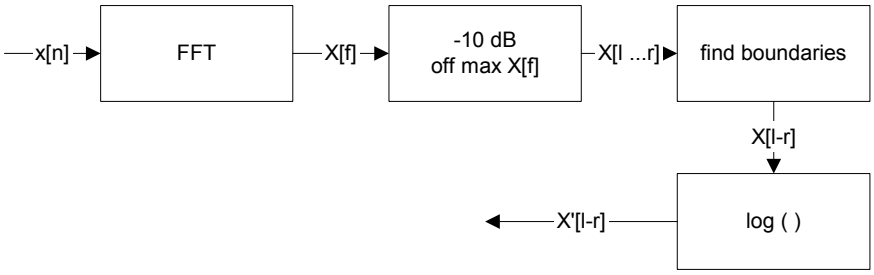


Figure 6.12 Log Spectral Spread Flow Chart

6.1.13 Roll-off

The *roll-off* point in Hertz is defined as the frequency boundary where 85% of the total power spectrum energy resides. It is commonly referred to *skew* of the spectral shape and is frequently used in differentiating percussive and highly transient sounds (which exhibit higher frequency components) from more constant sounds such as vowels.

$$SR = R \quad (6.19)$$

R is:

$$\sum_{k=0}^{R} X[k] = 0.85 \sum_{k=0}^{N-1} X[k] \quad (6.20)$$

X[k] are the magnitude components, *k* frequency index and *R* the frequency roll-off point with 85 % of the energy.

6.1.14 Phase

Before researchers like Chapin, Firestone, Fletcher, Plomp (Chapin, Firestone 1934; Fletcher 1934; Plomp, Steeneken 1969) and others were able to show the effect of phase in timbre perception it was not considered a relevant feature of timbre as discussed in section 4.1. It was however discovered that the influence of phase on timbre perception is usually with less dominance, especially with increase in fundamental frequency (Plomp 1976). For example, it has been found experimentally (Plomp, Steeneken 1969) that the maximal impact of phase on timbre perception for harmonic tones occur when a perfectly in phase harmonic tone (all its individual harmonics are in phase with respect to each other) and a harmonic tone with alternating 90 degree phase for each harmonic (e.g. $f_0^{[0]} f_1^{[90]} f_2^{[0]} f_3^{[90]} f_3^{[0]} \ldots$) are compared. Furthermore, the influence of the fundamental frequency on phase was confirmed in 1987 where experiments showed that phase influence on timbre was only perceptible for tones with fundamental frequency below 400 Hz (Patterson 1987). Essentially, for tones with fundamental frequency greater than 400 Hz the auditory system becomes phase deaf, as Ohm (Ohm 1843) and Helmholtz originally asserted (1875). It seems that phase becomes more prominent as the fundamental frequency is around 200 Hz or below even when the sound is limited to its first 12 harmonics. These results suggest that for pitched notes below middle C (approx. 260 Hz)

and most men's and women's voices depend on phase characteristics. Phase is also clearly perceptible in situations where it varies cyclically, resulting in the perception of beats. Phase however, although an important attribute, does not appear perceptually relevant in some acoustic surroundings as it gets blurred – for example in reflective spaces such as large cathedrals (Roads 1989). Thus, the impact of phase on timbral perception is substantially dependent on the acoustic environment and fundamental frequency in the case of pitched sounds. Phase for each frequency component is computed as:

$$X[k] = X_{real}[k] + jX_{imag}[k]$$
$$\phi[k] = \angle X[k] = \tan^{-1}\left(\frac{X_{imag}[k]}{X_{real}[k]}\right) \quad (6.21)$$

6.1.15 Spectral Flatness Measure

The *spectral flatness measure* is defined as the ratio between the *geometric mean* (Gm), and the *arithmetic mean* (Am) as shown in equation 6.22. It gives insight on the noise content of a signal and has been used in speech research to extract voiced and unvoiced speech signals (Yantorno 2000). As SFM approaches 0 the signal becomes more sinusoidal and as SFM approaches 1 the signal becomes more flat and de-correlated.

$$SFM_{dB} = 10\log\left(\frac{Gm}{Am}\right) = 10\log_{10}\left(\frac{\left(\prod_{k=0}^{k=N-1}|X(k)|\right)^{\frac{1}{N}}}{\frac{1}{N}\sum_{k=0}^{N-1}|X[k]|}\right) \quad (6.22)$$

Equation 6.22 can be further extended to refer to the tonality (dominant sinusoidal components) denoted by α (Johnston 1988) which ranges from -60 dB to 0 dB and is computed via equation 6.23 with SFM_{dBmax} set to –60 dB. The closer α is to 1 the more "tonal" the signal becomes and the closer α is to 0 the noisier it is. This method does not imply any sort of harmonicity in the signal but suggests the absence and presence of dominant sinusoidal components which help in segmentation of signal.

$$\alpha = \min\left(\frac{SFM_{dB}}{SFM_{dB\max}}, 1\right) \quad (6.23)$$

6.2 Time Domain

6.2.1 Amplitude Envelope: Attack, Steady-state, and Decay

The *amplitude envelope* represents the shape of a tone's amplitude contour. It is generally divided into three sections, the *attack* or *onset*, the *steady-state* or *sustain* and the *decay or release*. Various algorithms exist in computing the envelope. One widely used method is the *Root Mean Square* (RMS) method, which is related to the average power of a signal, unlike the *peak-level* or *average*. The *average* changes little with time and hence performs poorly in capturing transient portions, whereas the peak level reacts too readily to the changing peaks giving too much transient information. The RMS measure on the other hand has been found to closely correspond to the way we hear loudness and is expressed as:

$$RMS = \sqrt{\left(\frac{1}{L} \sum_{n=0}^{L-1} x[n]^2 \right)} \quad (6.24)$$

L is the length of window and *n* sample number. Low pass filtering after RMS helps smooth out the envelope. An alternative method of obtaining the envelope is using the RMS "without the R," i.e. without square-rooting the averaged summation part. The result is known as the *short-time energy* envelope. Although it is commonly used for extracting envelope information, it accentuates high frequency components due to the non-linear square operation. One way of circumventing the square operation is taking the absolute value as seen in

equation 6.26 also known as half-wave rectification (low pass filtering after rectification also helps).

$$ShortTimeEnergy = \frac{1}{L}\sum_{l=0}^{L-1} x[l]^2 \quad (6.25)$$

$$Mag_{average} = \frac{1}{L}\sum_{l=0}^{L-1} \left| x[l] \right| \quad (6.26)$$

All of the methods discussed above use some sort of short-time windowing while stepping through the whole waveform. As attack times are often in the order of tens of milliseconds; although it varies between different sound sources, a reasonable window length and window step size (hop size, overlap percentage) must be chosen. The importance of the envelope's segments (ADSR – attack, decay, sustain, release as the synthesizer user community likes to call it) can be easily demonstrated by simply playing a familiar sound such as the piano in reversal – it is difficult to recognize it as a piano tone. This suggests that the shape or the order of the envelope's segments is very important in the *recognition* of timbre. Also, it has been found that the timbre of some orchestral instruments such as trumpets, flutes, trombones, and French horns display strong correlation with their temporal envelopes, in part due to the fact that their spectral envelopes tend to lack uniqueness (Strong 1963).

6.2.1.1. Attack

The attack as we've seen in previous MDS studies has constantly appeared as one of the more salient timbral features. It is characterized not only by its rise-time as shown below but also by its slope. For example, the attack of a flute has a long onset, while other instruments in the same family have a sharper attack. For brass instruments, the attack portion is characterized by unsteady pitch values until it reaches the steady-state and stabilizes into the target value. In bowed strings, initial high frequency scratch occurs when starting to bow until bowing stabilizes (Handel 1995). In the context of music (e.g. melodic lines, phrasings etc.) however, it seems that the onset portion (in the perception of timbre) is less important as the boundaries between onsets and offsets often get blurred. In one investigation, in order to better understand the effect of attack on the perception of timbre, the steady-state and attack portions were isolated (Iverson, Krumhansl 1993). The surprising conclusion from this study indicated that although the attack contained important features, those same features were also found in the steady-state portion. Hence, in their study, timbre was similar regardless if the whole tone, just the attack or only the steady-states were included. It seems that although the onset has been found critical by many researchers, recent studies also suggest that its place in the identification of instruments is somewhat context dependent (Martin, Kim 1999).

6.2.1.2. Steady-state and Decay

The steady-state corresponds to the portion of a sound where the amplitude level is stable and a constant pitch (quasi-periodic) is observed. In this region amplitude modulation and frequency modulation can frequently be observed (e.g. flute, violin, voice …). The decay portion of the envelope relates to the dying phase of a sound, ending with ultimate loss of energy. Generally the lower harmonics decay rates are longest and higher harmonics and partials die out considerably faster as figure 6.13 shows.

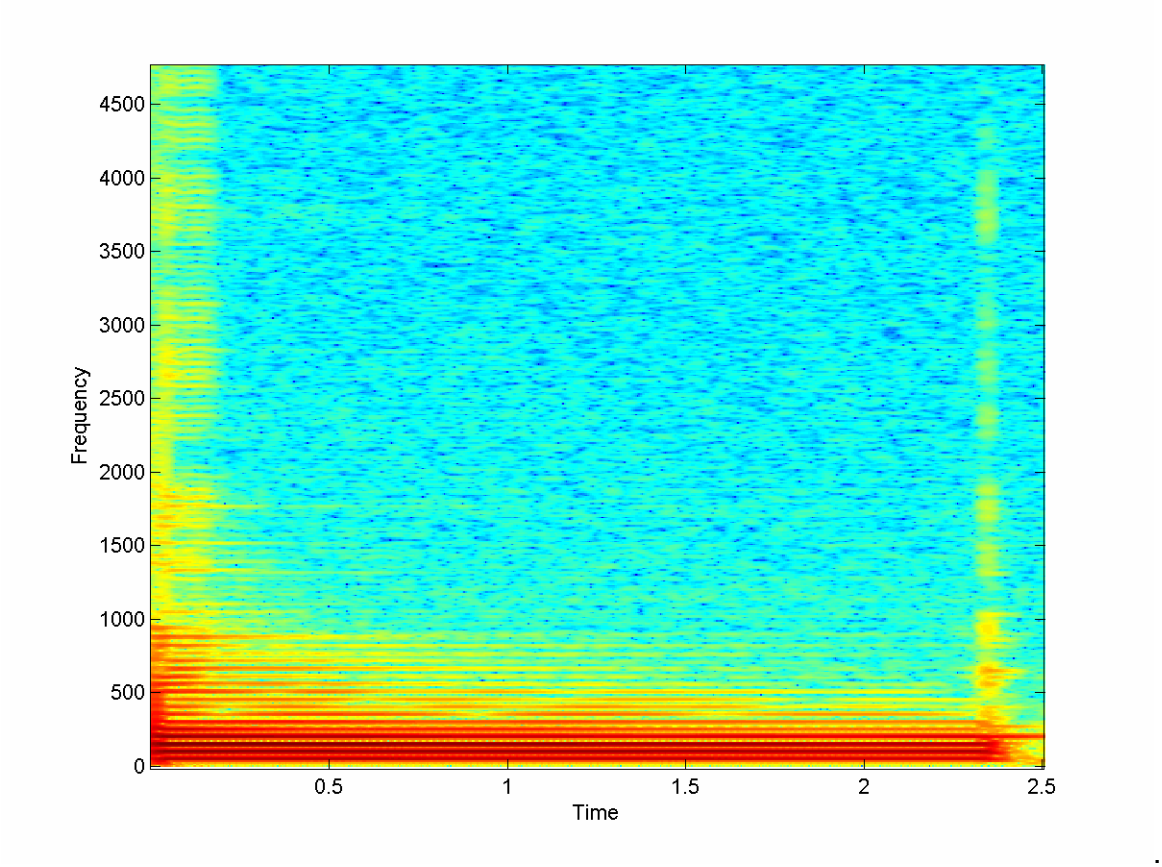


Figure 6.13 Spectrogram of electric bass guitar pluck @ 22050

6.2.2 Attack time (rise-time)

The traditional *attack time* or *rise-time* of a sound is simply defined as shown in equation 6.27 (sometimes used without the log operator). It computes the time between a start-point (determined via a threshold value) and the time location of the maximum amplitude of a sound. The log-rise time has been found to be an important dimension in MDS and other timbral studies as discussed before, where it is often found as one of the three dimensions of a given MDS timbre space (Saldanha, Corso 1964; Scholes 1970; Krimphoff 1993; McAdams 1995, Lakatos 2000).

$$LRT = \log\left(t_{\max} - t_{thresh}\right) \quad (6.27)$$

However, a caveat is that the threshold value and maximum gain values do not give an absolute definition of where the attack starts and ends. As a matter of fact, no concrete measurement studies have been published that I know of which describe a clear and unambiguous measurement technique to describe the attack. A slight variation for equation 6.27 also exists in the form of equation 6.28.

$$LRT = \log\left(t_{threshMax} - t_{threshMin}\right) \quad (6.28)$$

Here the only difference is setting a threshold value for the maximum magnitude of a signal with a maximum threshold coefficient, sometimes set to 2% of the

maximum magnitude (Misdariis, Smith, Pressnitzer, Susini, McAdams 1998). However, this approach although sometimes better for *some* envelopes, is again a static approach to a dynamic problem and may cause errors and inconsistencies when computing the attack time.

An alternative method to the above described algorithm is the one proposed in this dissertation below using a “mid-term” envelope and “envelope contour” analysis for more flexible computation of the attack time.

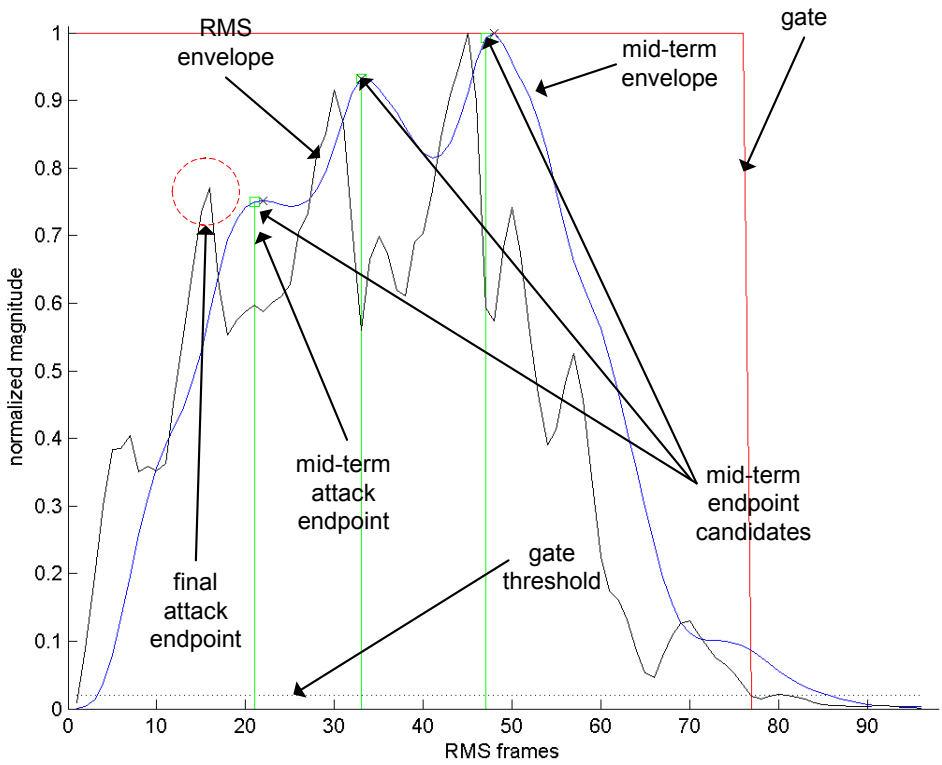


Figure 6.14 RMS envelope and envelope contour analysis of violin sample.

With the traditional method using t_{max} to determine the attack endpoint, an error of approximately 20+ RMS frames would occurs for the example in figure 6.14

as the maximum magnitude value would be used to determine that attack end point. The $t_{threshMax}$ method of 2% would also be virtually the same and contribute very little improvement. However, using the envelope contour analysis of the mid-term (low passed RMS envelope) three possible candidates for attack end points are computed for the above example. The peaks are determined through a global magnitude threshold value and a number of other threshold values that detect sudden changes in amplitude ultimately resulting in possible candidates for peaks that reflect surges in positive slope changes.

Using the “mid-term” attack endpoint the final attack endpoint of the RMS envelope is determined. This is achieved by selecting an RMS peak that is observed to occur before the mid-term envelope’s endpoint candidate as low passing a signal introduces delays. The final attack-time is computed using the modified version of equation 6.28. Please refer to appendix for details.

$$LRT = \log\left(t_{RMS\_peak} - t_{thresh}\right) \quad (6.29)$$

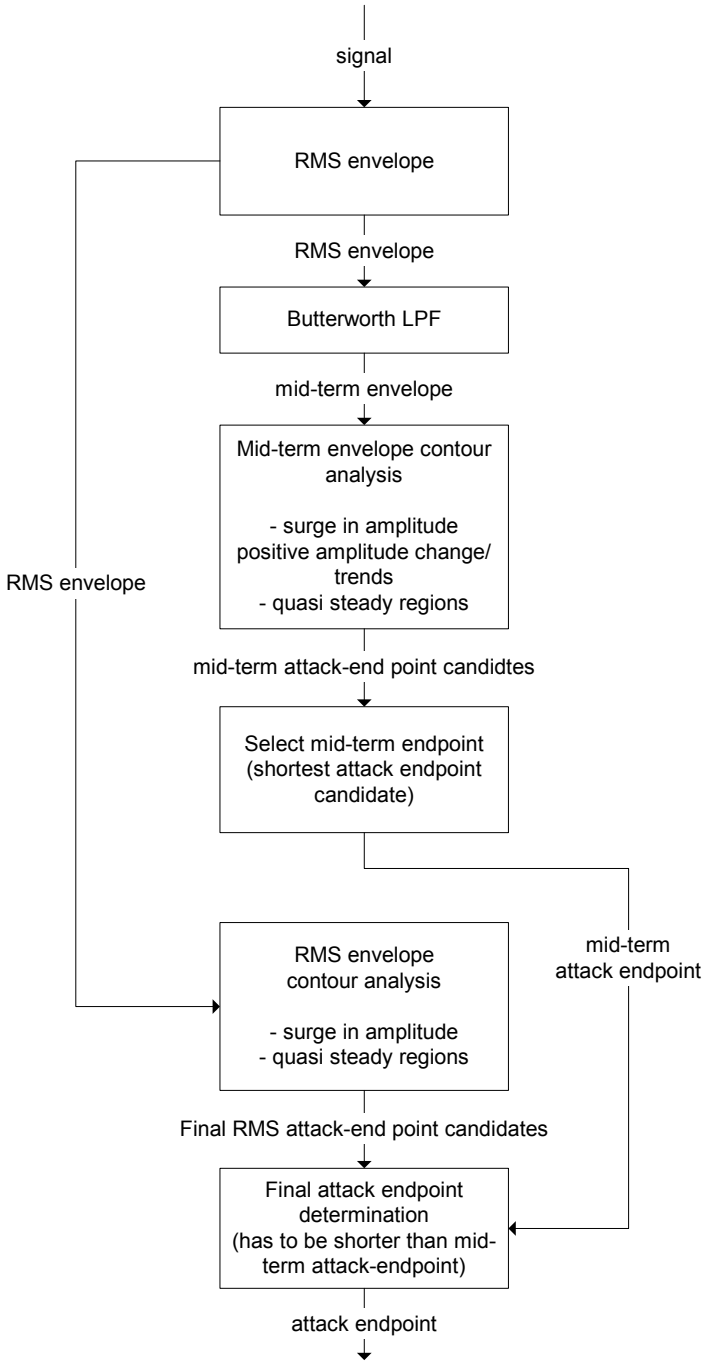


Figure 6.15 Basic flowchart for attack endpoint computation

6.2.3 Amplitude Modulation (Tremolo)

Amplitude modulation is omnipresent in musical tones, especially in performance situations and is often accompanied by frequency modulation, both being strongly coupled to each other (Backus 1976). As seen in figure 6.16 (Park 2000) low frequency oscillation can be seen in the amplitude envelope.

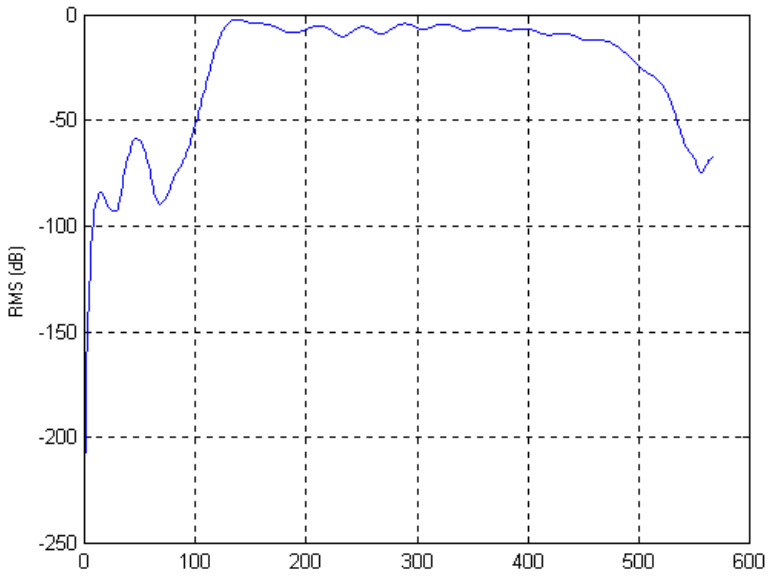


Figure 6.16 Soft tremolo on saxophone

The amplitude modulation frequency typically ranges from 4~8 Hz and is usually found in the steady-state portion of a sound. It can be computed with pitch tracking (see section 6.2.5.1) methods of the sustain portion of the sound. For modulation frequencies between 9 ~ 40 Hz, a “roughness” and “graininess” perception is noted, which also contributes to the resulting timbral quality.

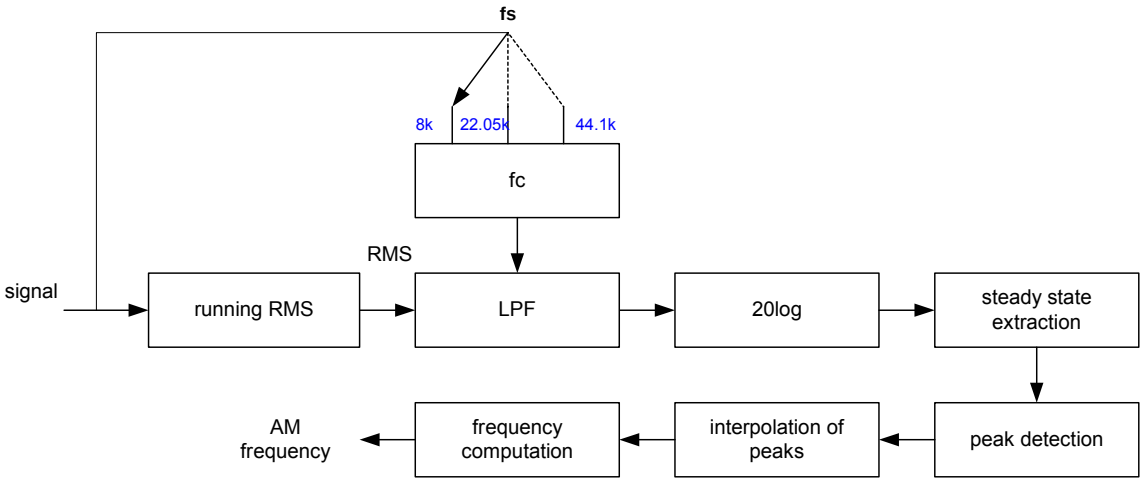


Figure 6.17 Block diagram of AM extraction (Park 2000)

6.2.4 Temporal Centroid

The *temporal centroid* has been found important as signal descriptors for highly transient and percussive sounds. It has been used along with log-attack time and spectral centroid in MDS experiments by Lakatos (Lakatos 2000). It is defined as the energy weighted mean of the time signal given in equation 6.30 and is also a part of the MPEG-7 audio descriptors.

$$TC = \frac{\sum_{n=1}^{L-1} nx[n]}{\sum_{n=1}^{L-1} x[n]} \quad (6.30)$$

x[n] is the input signal, *n* time index and *L* the length of the signal.

6.2.5 Pitch

Pitch is a perceptual aspect of sound. It corresponds to the quasi-periodic harmonic nature of sounds and is commonly referred to as the fundamental frequency. Humans can under normal and healthy conditions hear *pitches* from about 20 to 4200 Hz and hear *frequencies* of up to 20 kHz. It is commonly mistaken to think that the human hearing system is capable of discerning pitch up to 20,000 Hz. This is however not the case – for frequencies above approximately 4.2 kHz a "sense" of high frequency content is felt rather than a precise pitch value. When defining timbre, pitch along with loudness and duration is usually not included. However, this rather over-simplistic assumption that it is independent and uncoupled from other dimensions of timbre, has not been verified nor proven (Krumhansl 1989). Krumhansl later reported that whilst pitch and timbre can be *manipulated* independently they are not *perceived* independently (Krumhansl, Iverson 1992). In any case, the pitch range of musical instruments alone can actually deduce a lot of information. For example, the pitch range of an instrument can tell us that a high-pitched string sound cannot be a cello, or it can give us clues about the physical resonant structure of acoustic instruments – the cello playing an open string C cannot be a viola for instance (unless the viola for some reason has been tuned down which is for most cases highly unlikely).

Historically, pitch has been closely linked with what is called the *place theory*. That is, the inner ear's basilar membrane's surface seems to be excited "place-

wise" by a traveling wave in motion: each "place" (location) on the basilar membrane corresponding to a specific frequency (Bekesy 1960). This model takes after the idea of the ear performing spectral analysis analogous to a DFT spectrum analysis. However, some problems exist with such a model. Although our hearing system is capable of high fidelity pitch detection, with the place theory model, only coarse pitch perception is possible. This is in part because the waves cannot peak in such precise locations on the basilar membrane. It has been observed that for low frequencies (approximately lower than 200 Hz), the place theory breaks down as the basilar membrane at the far end is excited as a *whole* and *overlapping* manner thereby making the perception of low pitch next to impossible. Furthermore, it fails to explain the missing fundamental phenomenon, since humans can clearly perceive pitch without the fundamental frequency via subsequent harmonics. A solution to this problem was suggested which viewed the hearing system to function in a temporal manner. That is, the basilar membrane does not specifically react to an absolute location for its frequency information, but depends on the neural firing rates. Thus pitch is determined via timings of a *group* of neurons firing in synchrony. This is called *phase-locking* of nerve fibers. The reason the nerve fibers are thought to work in close phase-locked synchrony is due to the limitation of the discharge rate of each nerve fiber which is around 300~500 Hz. Clearly a single fiber will not be able to account for pitches higher than 500 Hz, but a group of phase shifted nerves working together will. The controversy still lingers and hybrid spectro-

temporal models also have been suggested to explain some fallacies with both spectral and temporal models.

There are a number of different ways of computing pitch, one of the most straightforward ones being the *zero-crossing* method. As the name implies, it can is computed while stepping through a waveform and measuring the time between successive crossings of the "0 axis." However, before computing the zero-crossing rate, the signal should be passed through a DC offset removal filter to take out the mean so that the signal oscillates about the zero magnitude axis. This method particularly works well with signals that behave well within the context of harmonicity. For example, in a perfect sine wave, every second zero-crossing would be equivalent to one period ($T = 1/f$) or its fundamental frequency. However, for natural, complex signals high frequency noise problems should be accounted for and algorithmic enhancements such as down-sampling and low pass filtering is needed for robust pitch detection.

6.2.5.1. Autocorrelation Method for Pitch Extraction

The *autocorrelation* method for pitch computation is one of the most widely used algorithms. It is essentially a similarity comparison of a signal with itself with one waveform stationary and the other shifting in time. Therefore the strongest peak always occurs where lag = 0 (no shifting) and the next maximum peak that follow the "0 lag peak" will normally correspond to the fundamental frequency. The algorithm is defined as:

$$acf_{xx}[\tau] = x[\tau] * x[-\tau] = \sum_{n=0}^{N-\tau-1} x[n]x[n+\tau] \quad (6.31)$$

τ is the lag, *acf[* τ *]* is the corresponding autocorrelation value. When $\tau = 0$, $acf_{xx}[\tau]$ becomes the signal's power. A typical autocorrelation plot is shown in figure 6.18. The "fast autocorrelation" method is used whenever CPU demand is an issue by exploiting the fact that multiplication in the time domain corresponds to convolution in the frequency domain and vice-versa. Hence, autocorrelation may be also computed by taking the FFT of a signal and self-multiplying it in the frequency domain.

$$x[\tau] * x[-\tau] = X[f]X^{*}[f] = \left|X[f]\right|^{2} \quad (6.32)$$

$$acf_{xx}[\tau] = DFT^{-1}\{\left|X[f]\right|^{2}\} \quad (6.33)$$

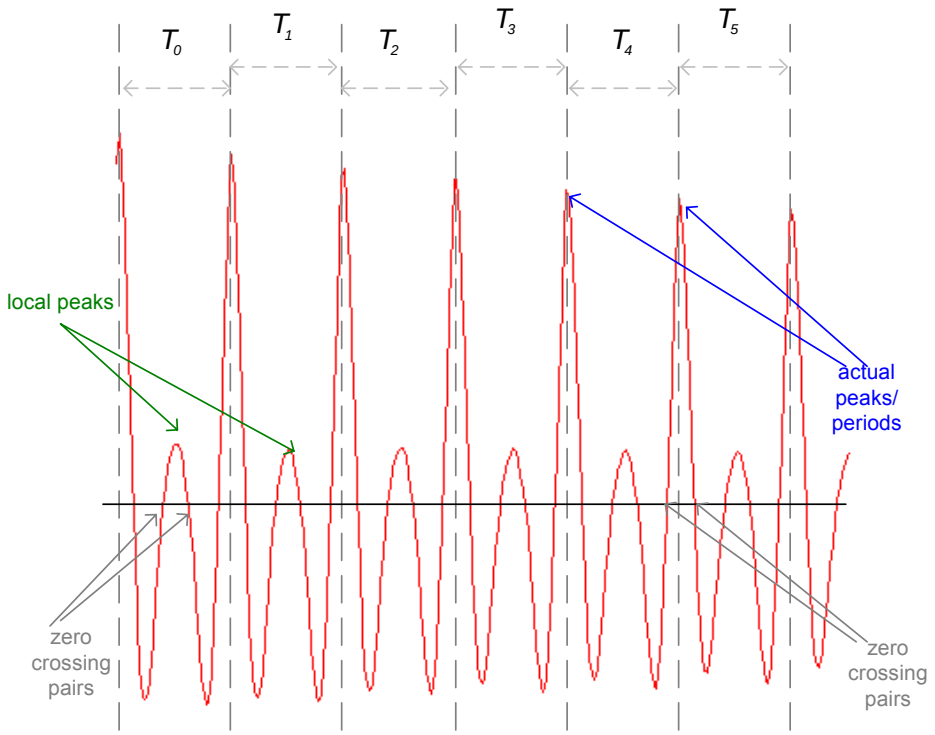


Figure 6.18 Autocorrelation plot

An algorithm for pitch extraction is outlined in figure 6.19 (Park 2000) with the inclusion of interpolation of peaks for better time resolution. It should be noted that the autocorrelation works best in low-range frequencies, as more samples are available. In summary, as the pitch increases a decrease in time resolution (samples) occurs and as the pitch decreases the time resolution (samples) increases resulting in more accurate pitch computation due to the $f = 1/T$ relationship (*f* is frequency, *T* is period).

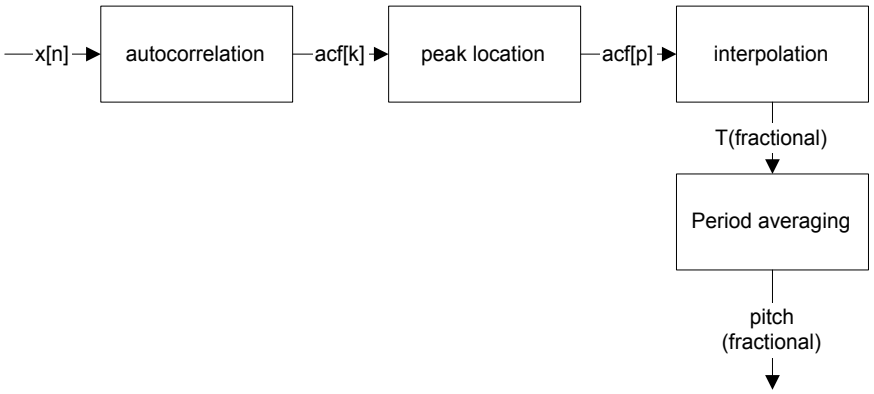


Figure 6.19 Autocorrelation pitch extraction

6.2.5.2. Autocorrelation with Adaptive Lag Length Method

An enhanced algorithm that I have developed using an "adaptive lag length" method is presented in this section. With the adaptive lag length method it is possible to get better transient pitch information compared to the conventional static lag length methods which require a lag length of at least twice the *lowest* frequency. In order to encompass fundamental frequencies down to 20 Hz for example, a lag length that corresponds to this frequency would require a minimum of 4410 samples (2205*2 frame size @ *fs* = 44,100 Hz) for the computation of the fundamental frequency. In practical applications an even larger lag length is required for accurate and consistent pitch computation. That would mean a frame size greater than 100 ms which is less than ideal for highly transient sounds. With the dynamic lag length method however, the lag length will dynamically adjust to the period – i.e. higher pitches need less number of samples whereas lower pitches require more samples. Figure 6.20 summarizes this method:

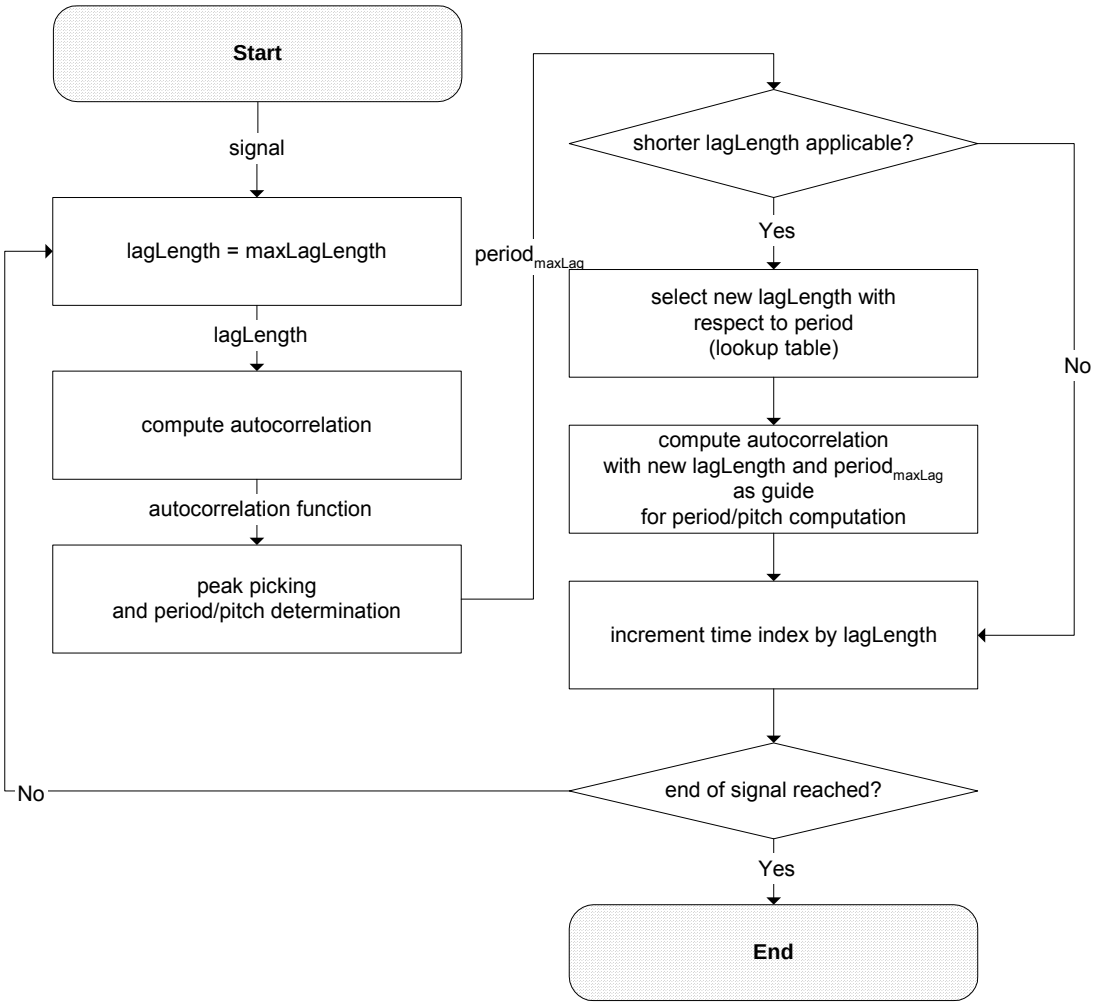


Figure 6.20 Basic flow of adaptive lag length algorithm

When computing the period with a shorter lag length, the original maxLag length period is used as a guide for searching for peaks (adaptive periods). This approach helps in minimizing errors as shorter lag lengths are more prone to incorrect selection of peaks in the autocorrelation function which correspond to the fundamental frequency. It should also be noted that after each frame increment the autocorrelation function is always computed with the maximum lag length in order to analyze the next frame's rough period.

6.2.5.3. Other Pitch Extraction Methods

Another popular method is using the DFT to find the fundamental itself (it is often the maximum magnitude of partials in a harmonic signal) or observing the harmonic relationships between the fundamental and subsequent harmonics. However, difficulties arise with spectra that have even or odd harmonics missing or fundamental harmonics missing altogether. An alternative to FFT pitch extraction is the cepstrum method defined in equation 6.34.

$$\hat{s}[n] = DFT^{-1}\{\log DFT(x[n])\} \quad (6.34)$$

$\hat{s}[n]$ is the real cepstrum component from input signal *x[n]* which can then be analyzed for peaks corresponding to the period and hence the fundamental frequency. According to Moorer (Moorer 1975) the cepstrum method is believed to work well in noisy environments as it highlights the loudest components in a signal.

6.2.6 Frequency Modulation

Frequency modulation (vibrato) is closely coupled with amplitude modulation (tremolo) and usually occurs in the steady-state portion of a sound as explained before. The method for computing low frequency oscillation is similar to vibrato extraction (see figure 6.21).

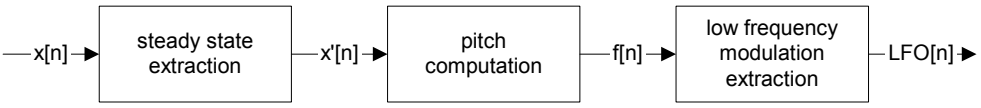


Figure 6.21 Frequency Modulation Extraction Block Diagram

6.2.7 Zero-Crossing Rate (ZCR)

The *zero-crossing rate* is defined as the time-domain zero crossings within a windowed portion of a sound normalized by the window length. As mentioned in section 6.2.5 it is used to find the pitch of a signal with some signal conditioning and stipulations on maximum allowed rates. Essentially, for complex tones and signals, ZCR gives some insight to the degree of noisiness, for example in the perception of vowels opposed to consonants in speech or in detecting endpoints in speech. However, it has also been used in a variety of timbre related research. Some examples include classification of percussive sounds; albeit only discriminating between kick and snare drums (Gouyon, Pachet, Delerue 2000) and temporal segmentation (Rossignol et al. 2000) of notes. The algorithm is shown in equation 6.35.

$$ZCR = \frac{1}{N-1}\sum_{n=1}^{n=N-1}\left|S\{x[n]\} - S\{x[n-1]\}\right| \quad (6.35)$$

$$S\{n\} = \begin{cases} x[n] \geq 0, & 1 \\ else & 0 \end{cases}$$

N is the length of signal (or window), *x[n]* the signal, and *ZCR* is the average number of zero-crossings within a portion of the signal.

6.2.8 Linear Predictive Coding (LPC)

Linear predictive coding is another sibling that comes from the speech research literature and has found its way to the music community. The theory can simply be thought as an analysis algorithm predicting the current sample, by using *p* number of past samples individually weighted as shown in equation 6.36.

$$\hat{s}[n] = \sum_{k=1}^{k=p} a_k s[n-k] \quad (6.36)$$

$\hat{s}[n]$ denotes the approximated sample, $s[n-k]$ past samples, a_k corresponds to weights of past samples and *p* the total number of past samples. In trying to minimize the error $e[n]$ between $s[n]$ and $\hat{s}[n]$ the optimum weights are computed which are called LPC coefficients. The synthesis filter becomes an all-pole filter in the form of equation 6.39 which can be excited by noise (non-pitched sounds) or pulse (pitched sounds) a shown in figure 6.22.

$$e[n] = s[n] - \hat{s}[n] = s[n] - \sum_{k=1}^{k=p} a_k s[n-k] \quad (6.37)$$

$$A(z) = 1 - \sum_{k=1}^{k=p} \alpha_k z^{-k} \quad (6.38)$$

the all zero filter then becomes an all pole filter

$$H(z) = \frac{1}{A(z)} \quad (6.39)$$

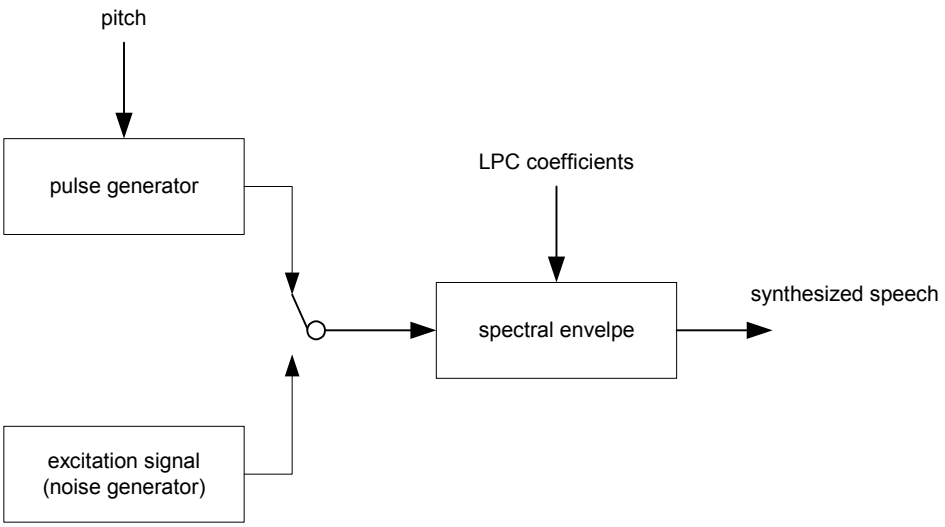


Figure 6.22 LPC synthesis model

The LPC method is also sometimes referred to the short-term prediction filter, that is, it is used to predict the “short-term” trends (“well behaving” portions of a signal such as the steady-state are predicted well whereas onset regions which are more unstable are predicted with less accuracy) and does a poor job predicting the “long-term” samples (known as long-term prediction), which refers to pitch estimation. I have exploited these shortcomings of the LPC algorithm and used it for analyzing noise-content in musical signals (Park 2000). By simply analyzing the LPC coefficient for a window and filtering (predicting) the same signal gives the residual (noise) signal as shown in figure 6.23. The pre-emphasis filter (1^{st} order FIR high pass filter) is used to flatten the spectrum due to the predominant low frequency content in musical signals, thus helping in the computation for better representative LPC coefficients. The long-term attributes (fundamental frequency) are still present and need to be extracted also, which can be done using pitch extraction methods and comb filtering.

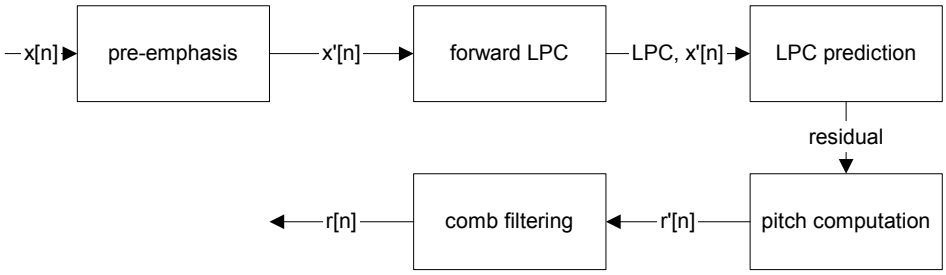


Figure 6.23 Noise content analysis block diagram

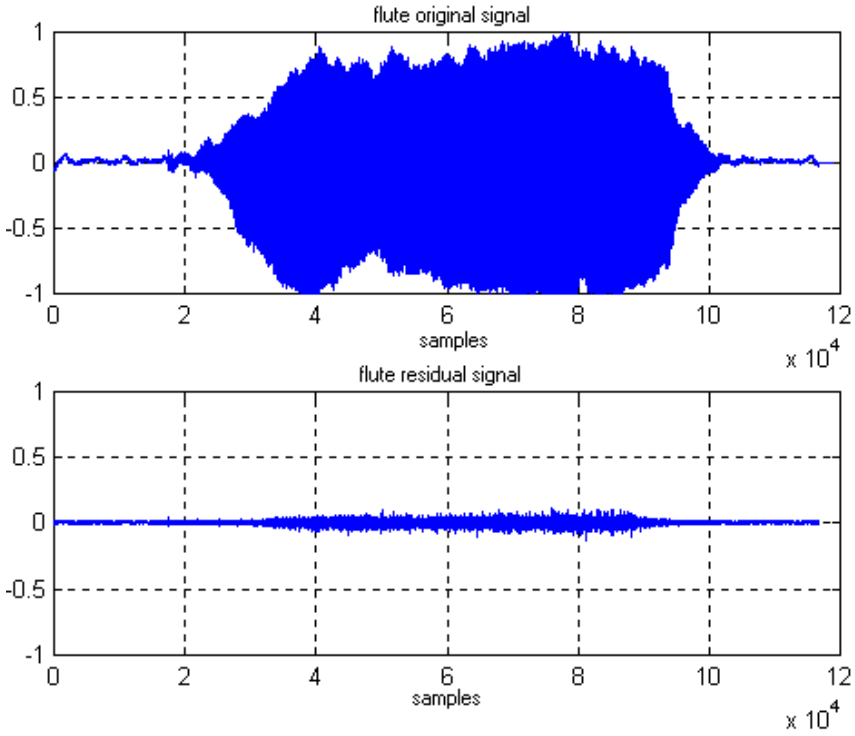


Figure 6.24 Flute noise content analysis

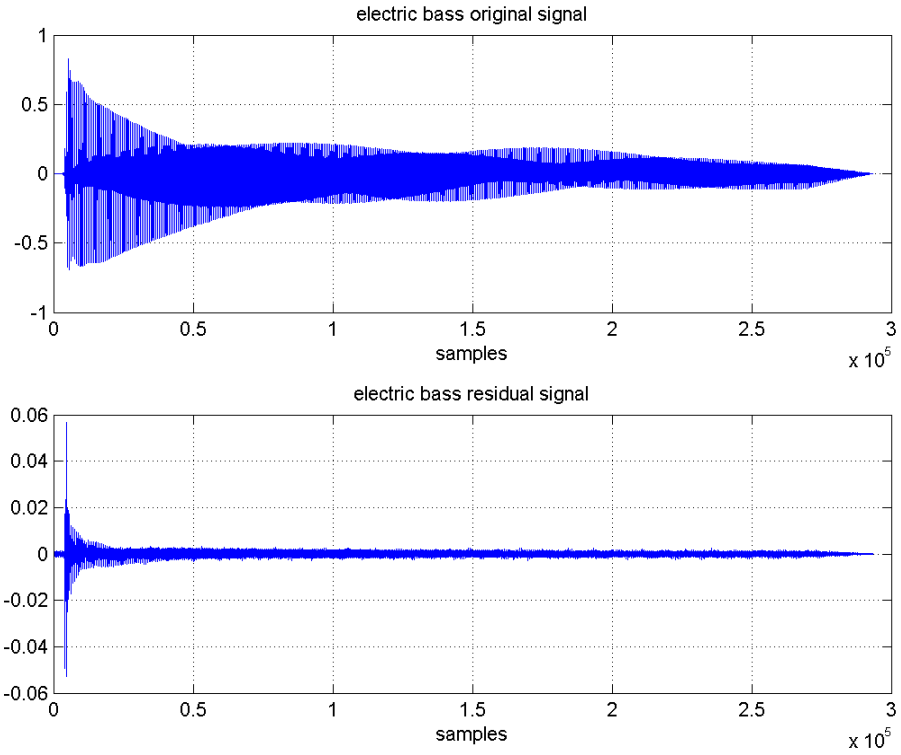


Figure 6.25 Electric bass noise content analysis

Figure 6.24 illustrates the noise content of the flute. It can be seen that the "breathiness" is present throughout the duration. In figure 6.25 it can be noted that noisiness is most prominent during the attack portion of the electric bass.

6.2.8.1. Formants

The LPC synthesis filter as mentioned before is an all-pole filter, which is essentially a resonant filter describing the vocal tract. The vocal tract can be compared to a conglomerate of small sections of tubes and it turns out that the LPC model is very useful in describing the *formant* structure of human speech. Formants are resonance locations typically ranging from 3 to 5 for human vowel sounds, which are distinct in location along the spectrum. Due to the "tube-

model” paradigm inherent in LPC systems, some musical instruments exhibit stronger formant characteristics than others. For example, woodwind instruments such as the oboe and clarinet generally show one or two pronounced formant regions (Backus 1976, Brown 1999). Also, the nasality and hollowness in timbre is noticeable in formants that are often found between the 3rd and 6th harmonics (Roederer 1979). Luce (Luce 1967) found that un-muted brass instruments exhibit a single formant, and Strong (Strong 1963) interpreted spectra in terms of their formant characteristics pertaining to the oboe, clarinet, and bassoon, further suggesting the relevance of formant structures in some musical instruments.

In this chapter I will present the pattern recognition module built on a BP (Backpropagation) and RBF/EBF *(Radial/Elliptical Basis Function)* neural network utilized in this dissertation. The neural network approach for pattern recognition is particularly well suited to complex systems where solutions are very difficult to compute algorithmically – which is the case for timbre recognition. Also, the bottom-up model in conjunction with a neural network loosely resembles the architecture and mechanism of one possible human auditory perception model. In the artificial model, sound pressure is converted from acoustic energy to electrical energy through transduction using a microphone, sampled for digital representation, pre-processed, analyzed and extracted of salient features; and finally piped to the neural network for classification as seen in figure 7.1. A similar structure exists in the human auditory system. From the outer to the middle and finally the inner ear, acoustical energy is ultimately transduced to electrical energy via the inner and outer hair cells, which are then passed to the brain for analysis and processing. Before jumping into RBFNs and EBFNs I will briefly go over some of the fundamentals of neural networks.

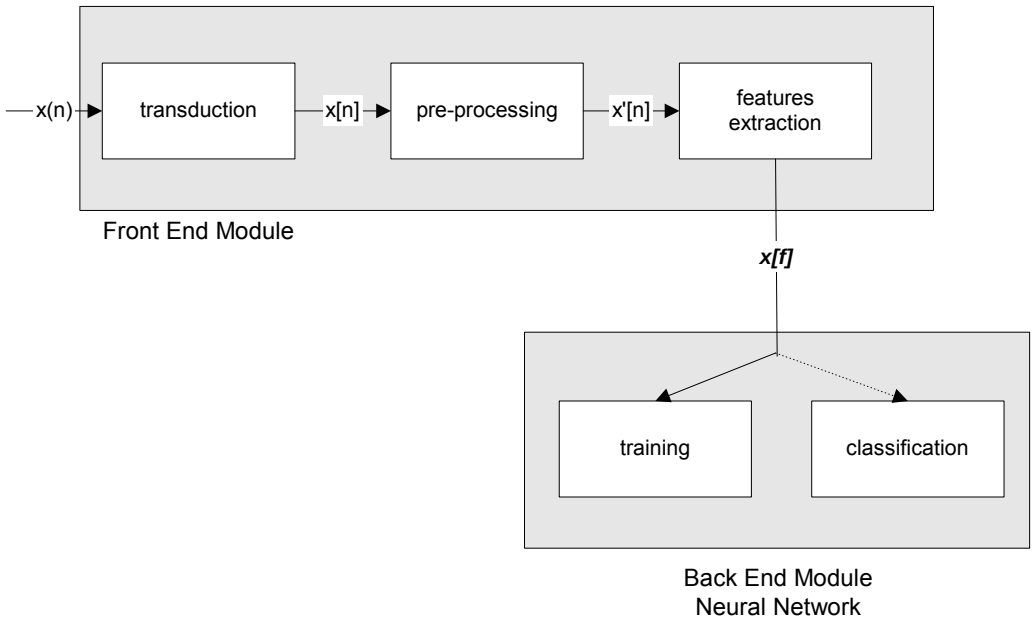


Figure 7.1 Bottom-up model in conjunction with neural network

7.1 What and Why Neural Networks?

Neural networks are inspired by the human brain's nervous structure somewhat modeled by the way biological neural networks process information. They go by many different names including *connectionism, parallel distributed processing, neuro-computing*, as well as *Artificial Neural Networks* (ANN). ANNs are different from *von Neumann* machines which are directly dependent and limited by the consequences of finding an algorithm that can describe a given problem. As seen in figure 7.2 the von Neumann model is based on memory, data, instruction, and CPU paradigm.

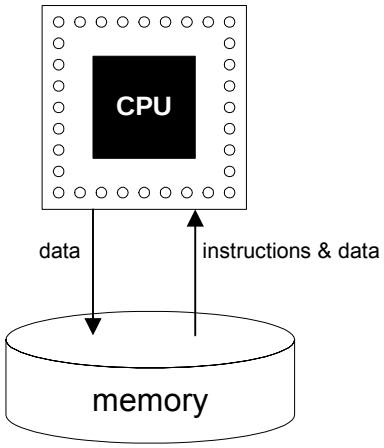


Figure 7.2 von Neumann model

This rather seemingly simple and promising algorithmic approach actually produces great complexity in formalizing algorithms that describe everyday tasks such as visual recognition of familiar faces, categorizing faces that we have never seen before, recognizing handwritten characters from different people, identifying micro speech patterns, and many other tasks including musical timbre recognition.

This is where neural networks come in. Neural networks are well suited for finding solutions that are difficult to formulate using algorithms. They may not necessarily give one as much insight on the exact mechanics of a system as direct algorithmic solutions do, but when it is not possible to design a generic and at the same time specific enough algorithm, a neural network approach may seem like an attractive alternative. ANNs are also desirable in situations where a large number of example data exist (I have used 829 number of samples for 3 instrument families, and 12 instrument types; see appendix section A.9 for details) as they have the ability to generalize from training data. Finally, ANNs

are suitable for problems where it is required to determine the structure from the existing example data.

McCulloch and Pitts published the first studies in ANNs in 1943 where initial simulations using formal logic and development of neural network models based on simple neurological behavior were presented (McCulloch, Pitts, 1943). Figure 7.3 shows a simple artificial neural network made up of an input layer and an output layer. However, networks with only one input and output layer are not powerful enough to address complex problems (a classic example is the inability of such a network to compute the XOR function). Adding hidden layers improves the learning ability of a network and is the standard method for designing an ANN (hidden layers are layers of nodes that are between the output and input nodes).

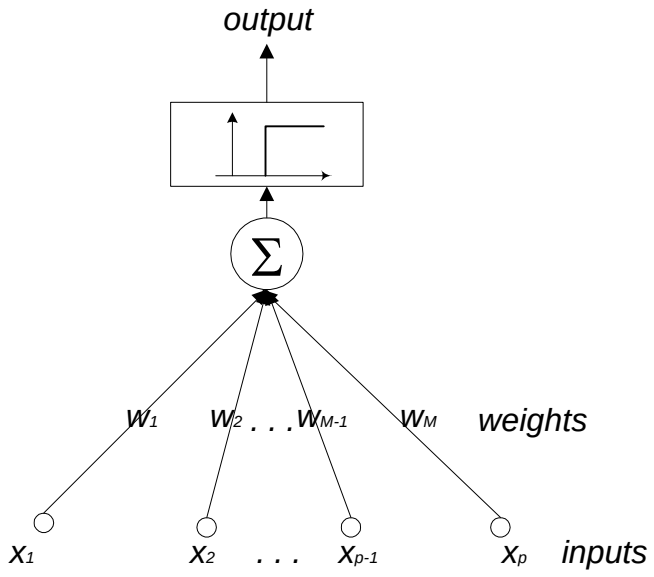


Figure 7.3 Simple neural network with step function

It should be noted however, that just adding hidden layers can prove to be counterproductive in practice. Some of the problems that arise from too many

layers include slow convergence in backpropagation learning due to introduction of additional local minima and numerical degradation of error signals during back-propagation of excessive number of layers (Kung 1993). Generally speaking, two-layer networks should be adequate as universal approximators of any non-linear functions and three-layer networks suffice to separate any convex or non-convex polyhedral decision region from its background (Lippmann 1987). Obviously as the number of layers increase the complexity of the network increases accordingly. Biological neurons have approximately 10,000 inputs, which in turn are connected to other neurons. For artificial neurons the number of input nodes are much smaller due to more practical reasons such as complexity and available processing power.

7.1.1 Basis Functions and Activation Functions

As seen in figure 7.3, the output is essentially a linear combination of weighted inputs causing the output to go “high” or “low” depending on the cumulative contribution of each of the inputs and their corresponding weights. The sum of weighted inputs before being subjected to an activation function is called the basis function.

The basis function in figure 7.3 is called the *Linear Basis Function* (LBF) and is expressed as shown in equation 7.1, also known as the McCulloch and Pitts (MCP) model (Minsky, Papert 1969). Equation 7.1 is the generic representation

of a basis function where u_i denotes the input to the *output* node *i* (in this case we only have one output), w_{ij} the weights, and x_j the input.

$$u_i(w, x) = \sum_{j=1}^{N} w_{ij} x_j \quad (7.1)$$

The output's value of "high" and "low" is determined by an *activation function*, which "activates" an output according to the input it receives. The activation functions are commonly among others, step functions (figure 7.3), Gaussian (figure 7.10, equation 7.3), or sigmoid functions (figure 7.4, equation) where the shape of the sigmoid is determined by *r* in equation 7.2. For the bell shaped Gaussian distribution (normal distribution) *x* is input, μ is the mean, σ^2 is the variance. In essence the processing ability of a neural network is stored and determined by w_{ij}, where the objective is to find suitable weights which will render a desired output.

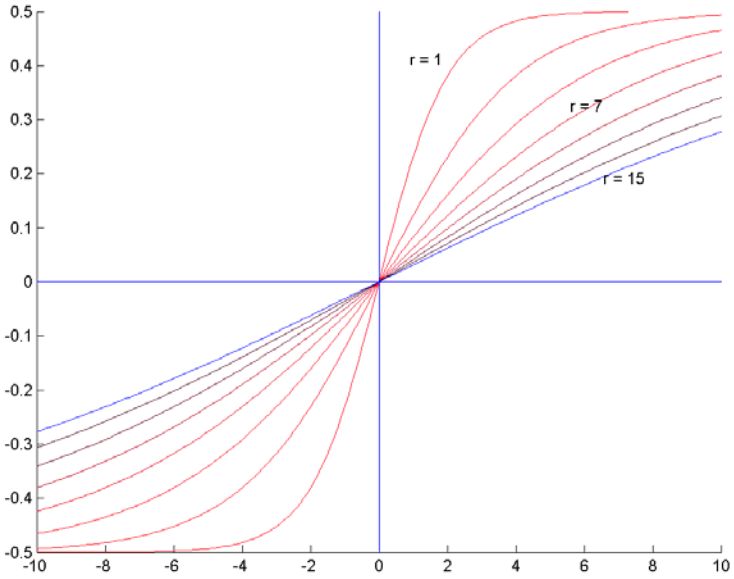


Figure 7.4 Sigmoid functions

$$y(x) = \frac{1}{1+e^{-x/r}} \quad (7.2)$$

$$y(x) = ce^{-(x-\mu)^2/2\sigma^2}, \quad c = 1/\sigma\sqrt{2\pi} \quad (7.3)$$

This process of finding appropriate weights is referred to as the *network learning* achieved through iterative training via learning patterns.

7.1.2 Learning Rules: Unsupervised and Supervised

Training a neural network to find the appropriate weights is achieved in two ways – through *supervised* and *unsupervised learning*. Supervised learning requires a "teacher" to guide the system in computing its weights, whereas unsupervised learning does not require a teacher and hence adjusts its weights automatically.

There are many different methods for weight adaptation but most are based on the oldest learning rule known as *Hebb's Rule*.

7.1.2.1. Unsupervised Learning

Hebb's Rule

Hebb's Rule (Hebb 1949) named after Donald Hebb follows a very simple design. Basically it is a rule that allows for strengthening of connections between neurons that have similar activity. For example, if neuron A is connected to neuron B and both A and B are highly "active" in the same direction (negative or positive), then the weight connecting A and B is strengthened. This is an unsupervised learning method as the weights are updated automatically from the input patterns without guidance from a teacher. A slight enhancement of Hebb's Rule is the *Hopfield Law*.

Hopfield Law

The Hopfield Law is similar to the Hebbian method except for the addition of a *learning rate* usually denoted as η, a scalar multiplier that strengthens or weakens respective weights incrementally.

Kohonen Learning Rule

Kohonen's Learning Rule inspired by basic behavior of biological systems capitalizes on *neural competitive learning* for updating weights. Basically the winner (the neuron with largest output) is given the power to inhibit other

competitors but at the same time “help” selected neighboring neurons update their weights. The “helping” and sensitivity to neighborhood or history provides a way to avoid some nodes from being completely unlearned and also helps enhance topological properties from the example patterns. However, only the winner is allowed to output. Such self-organizing networks (also known as *Self-Organizing Feature Maps* – SOFM) do not require a teacher and use clusters with memberships and centroids (means or centers of clusters) that get adjusted whenever new input patterns are introduced to the system.

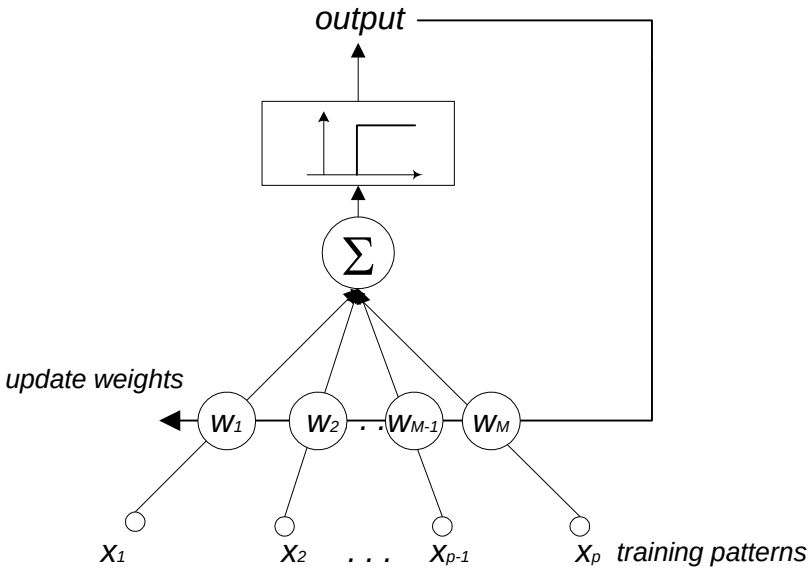


Figure 7.5 Unsupervised learning

7.1.2.2. Supervised Learning

Supervised learning as the name suggests, requires a “teacher” to train the network. The aim is to teach a neural network to perform a given task by adjusting its weights in order to reduce the error (cost function) between the

desired output (teacher) and actual output (computed). The methodology of using the error between the computed output and the desired output or the "delta" is called *Delta Rule Learning*. Delta Rule as the name implies applies continuous tweaking of the weights for reducing the delta (difference) between the desired output and the computed output. Backpropagation (BP) also known as *Error-Backpropagation, Windrow-Hoff Learning Rule*, and *Least Mean Square Learning Rule* is an example of Delta Rule Learning and is one of the most widely used methods for training a neural network. In BP the error usually computed as the mean squared error is "back-propagated" into the system until the bottom-most layer is reached. This method is one of the most popular methods for a supervised learning network and usually includes learning rate parameters similar to the one specified in Hopfield Rule.

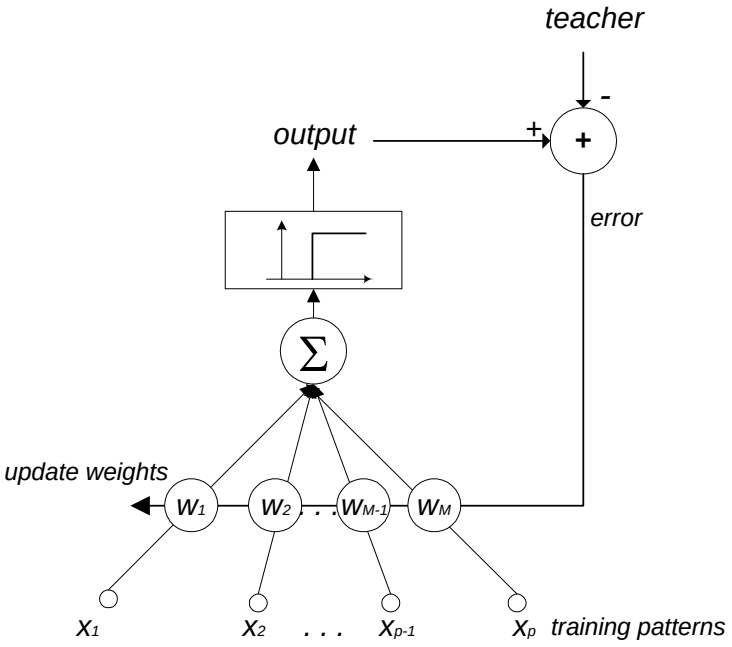


Figure 7.6 Supervised learning

7.2 RBF/EBF Neural Networks and Backpropagation (BP)

This section introduces the architecture and design of the pattern recognition module used in this dissertation. It is based on feed-forward Radial/Elliptic Basis Functions (RBF/EBF) with Gaussian distribution and backpropagation (BP) learning networks (supervised learning).

7.2.1 RBFN and EBFN

RBFs surfaced as a possible variant of ANNs in the late 80s and have been used by researchers in basically two areas – functional approximation for time series modeling (Park, Sandberg 1991; Poggio, Girosi 1990) and pattern classification. In the area of pattern classification they have been used for tasks such as speech recognition (Niranjan, Fallside 1990; Mak 1993, 1996), speech prediction (Birgmeier 1996), phoneme recognition (Berthold 1994), and face recognition (Sato, Shah, Aggarwal 1998; Thomaz, Fetiosa, Veiga 1998; Huang, Shimizu, Kobatake 2002). Interestingly, RBFs have especially had impressive results in the vision community where one experiment showed 1.92 error percentage (Er et al 2002) using the face ORL (Olivetti Research Laboratory) database consisting of 40 individual faces with 10 images for each individual face. In another face detection study using a modified RBFN, a detection rate for 270 images was between 99.25% ~ 100% (Huang 2002). These are quite impressive and promising results for face recognition which is in part also a perceptual problem much like timbre recognition. It is interesting to observe that a number of the

issues concerning difficulties in pattern recognition of faces are somewhat similar to timbre recognition issues as outlined in table 7.1.

Face recognition	Timbre recognition
Faces are highly variable	Timbres are highly variable
Individual facial idiosyncrasies Pose Expression Facial hair, make-up	Manufacturer's idiosyncratic timbre Direction of sound projection Performance expressivity Filtering, production (recordings)
Lighting conditions Background Scaling	Auditory space Background noise Dynamics

Table 7.1 Similarity problems for face recognition and timbre recognition

That is not to say that table 7.1 suggests face recognition and timbre recognition share exactly the same problems which can be solved with the same feature extraction algorithms, nor is it suggested that both pattern recognition tasks are equally well suited for a given EBFN/RBFN. It is however worthwhile to note that some of the difficulties that are subtle in nature yet significant in face recognition are also seemingly present in the timbre recognition task. Approaching the problem from a purely pattern recognition approach without consideration of musical and psychoacoustic theories or the mechanics involved in visual and auditory perception, and further simplifying the architecture to the passing of data from the front-end to the back-end; the problem from the neural network point of view becomes a task of learning from example training sets. Considering the impressive results RBFNs have had in face recognition applications, coupled with

the fact that there have been no studies made with RBFNs in timbre classification at the time of writing this essay (although they have been used for sound synthesis (Röbel 1995), and morphing (Drioli 1999)), it may be meaningful to consider its possibilities for musical instrument timbre classification.

7.2.1.1. RBFN and EBFN Characteristics

For complex neural network-based pattern classification, Multi-Layer Perceptrons (MLP) have been widely used. MLPs are basically concatenated LBFs with multiple hidden layers and activation functions as seen in figure 7.7. RBF/EBF basis functions fundamentally differ from MLPs in the following ways. As we have seen in the previous section, LBF architectures require inputs to be weighted before being summed as shown in figure 7.7. The sum is then subjected to an activation function such a step function or sigmoid function causing the output to go “high” or “low” depending on some threshold value. The training consists of adjusting the weights that connect the inputs to the outputs in each layer. For RBF/EBFs on the other hand, the inputs are directly patched to each basis function and the output of the activation functions are then weighted and summed. Also, RBFNs take non-linear input spaces and output linear activation outputs via only one hidden layer (basis function). Using inherent nonlinear approximation properties, RBFNs/EBFNs have the capability to model very complex patterns, which the MLPs can only achieve through multiple intermediary hidden layers (Haykin 1994). RBFN/EBFNs also have faster learning capacity, are easier to

implement, are less complex in structure, and are computationally more efficient than MLPs.

.

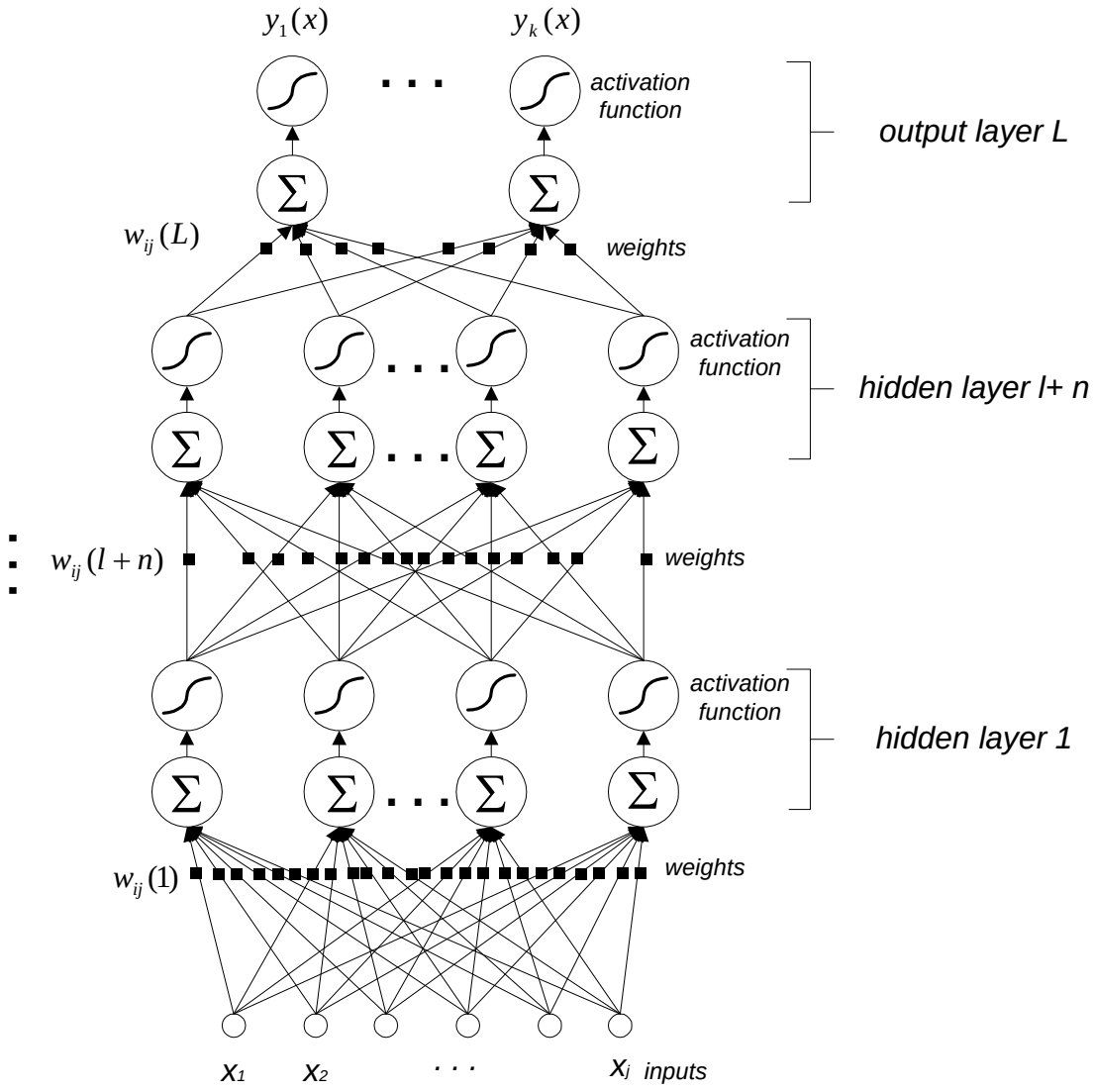


Figure 7.7 MLP

RBF/EBFs are built on centroids (means), spreads (standard deviation from mean), and activation functions. The centroids with their spreads corresponding to a cluster obtained via the training samples for each individual activation

function are denoted as $\phi(.)$ in figure 7.8. As with MLPs, RBF/EBFNs are also trained to adjust its weights but in addition, the spreads and centers of each cluster are also updated. For feature spaces in two dimensions a circular cluster is formed for RBFs and elliptic cluster for EBFs. For 3 dimensional feature spaces spherical clusters are rendered for RBFs and ellipsoids for EBFs. For dimensions greater than 3, hyperspheres and hyperellipsoids are rendered for RBFs and EBFs respectively. Hence, the only difference between RBFs and EBFs is that for RBFs, the spreads are constant in all directions whereas the spreads for EBFs are variant giving them more flexibility in forming clusters (RBFs are special cases of EBFs where the diagonal covariance matrix is equal).

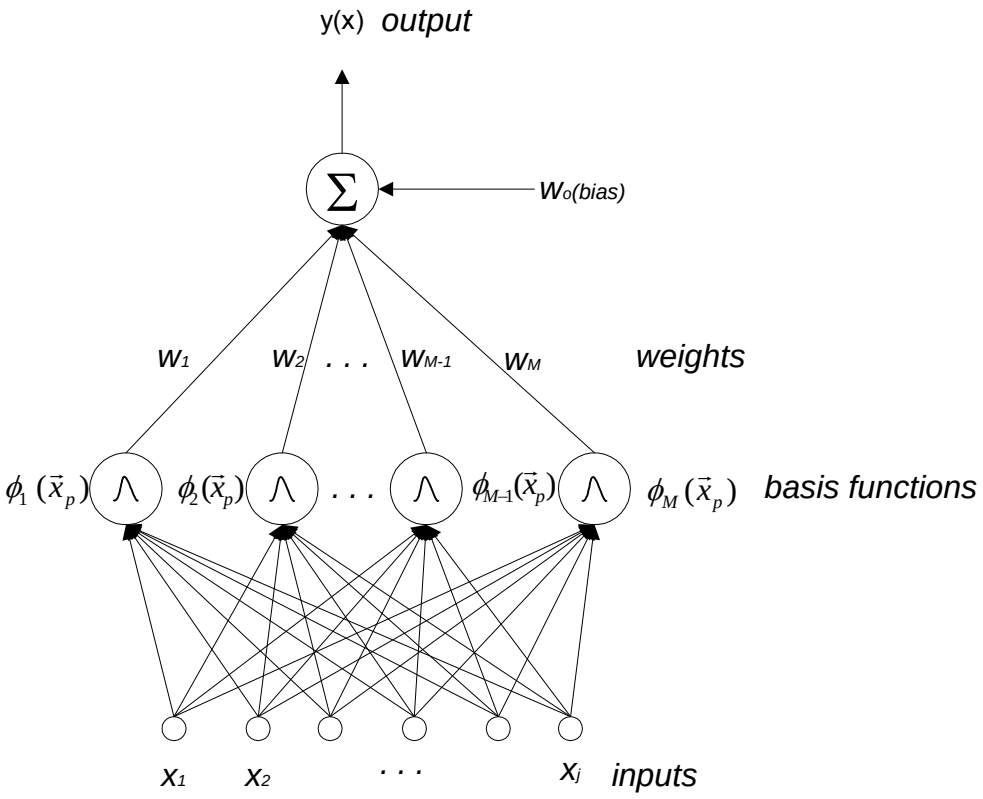


Figure 7.8 Basic RBF/EBF neural network

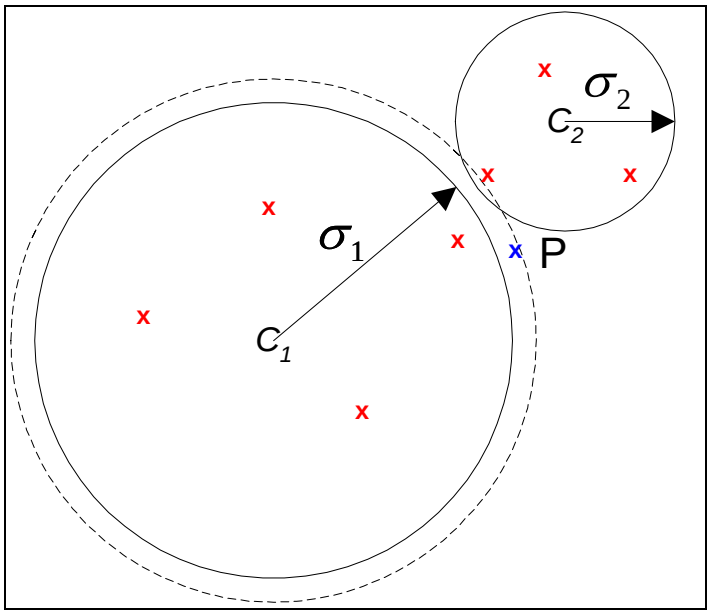


Figure 7.9 RBF clusters with width σ_1 and σ_2

7.2.2 RBF/EBF Activation Functions

The most common kinds of activation functions for RBFN/EBFNs are *Thin Plate Spline* and *Gaussian* activation functions (equation 7.4).

$$h(r) = e^{-r^2/2\sigma^2} \quad (7.4)$$

Thin Plate Spline activation functions are usually used for time series modeling whereas Gaussian functions have found much popularity in classification applications (Mak 1993, 1996; Kovacevic, Loncaric 1997; Moody 1989; Kumar, Srinivas 2001). As we can observe from equation 7.4, figure 7.9, and figure 7.10 as input data points are further away from the mean (C_1/C_2 in figure 7.9, μ in figure 7.10), the activation output decreases exponentially. Hence patterns

located at large distances from the mean (cluster centers) will fail to activate a particular basis function while maximum activation is achieved by data samples closest to a cluster's mean. In addition for Gaussian activation functions, as each cluster has its own Gaussian distribution, each cluster's boundary will be independently defined by both its unique mean and spread. As seen in figure 7.9 although sample P is closer to centroid C_2 than C_1 in Euclidian distance, the spread of class 1 can be adjusted to allow *P* to have greater class membership to class 1 than class 2.

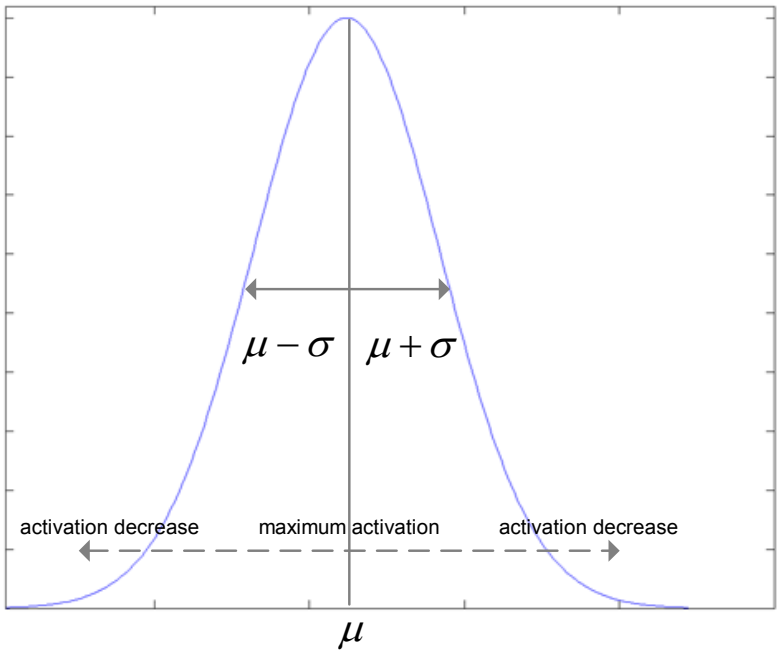


Figure 7.10 Gaussian normal distribution

The RBF basis function is defined by the Euclidian distance *r* and the activation function by $\phi(.)$, where x_p is the input sample number *p*, μ_j is the centroid for

cluster *j*, *n* is the dimension of input vector x_p (*N* = 3 for 3D space), and σ is the spread for cluster *j*.

$$r_{Euclidian} = \left\| x_p - \mu_j \right\| = \sqrt{\sum_{n=1}^{N} (x_{np} - \mu_{nj})^2} \quad (7.5)$$

$$\phi_j(r) = e^{-r_{Euclidian}^2 / 2\sigma^2} \quad (7.6)$$

For EBFs as the spread is not constant in all directions the Mahalnobis distance is used for distance computation as shown in equation 7.7. The Mahalanobis method takes into account correlations (covariance matrix Σ) between data points. RBF is hence a special case of EBF where the covariance matrix is diagonal with diagonal elements equal. For the EBF activation function:

$$r_{Mahalanobis} = \sqrt{(x_p - \mu_j)^T \Sigma_j^{-1} (x_p - \mu_j)} \quad (7.7)$$

$$\phi_j(x_p) = e^{-r_{Mahalanobis}^2 / 2} \quad (7.8)$$

$$= \exp\left\{ -\frac{1}{2} (x_p - \mu_j)^T \Sigma_j^{-1} (x_p - \mu_j) \right\} \quad (7.9)$$

x_p is the input, μ_j is the mean for cluster *j*, Σ^{-1} is the inverse covariance matrix, ϕ_j is the activation function output for cluster *j*, and *T* is matrix transposition.

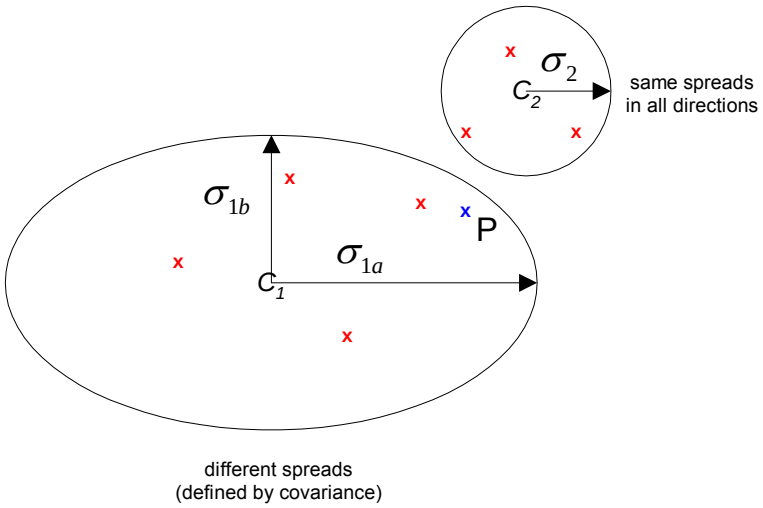


Figure 7.11 EBF/RBF clustering patterns

Figure 7.11 illustrates the same data points from figure 7.9 but with two classes having different spread characteristics. Class 1 uses an elliptical spread, and class 2 the original radial spread. It can be seen that for the EBF-based cluster, class membership of patterns are more flexible and pattern *P* is given membership with greater elasticity. For the RBF only example (figure 7.9) however, with the widening of radial spread σ_1 boundary and membership issues become more difficult.

7.2.3 Backpropagation (BP) and Network Training

Having setup the RBF/EBF basis functions what remains is adjusting the weights, spreads, and means of the network. One of the most popular and effective method is backpropagation (LeCun et al 1989). BP has been proven highly effective in training neural networks as it is not just given reinforcement for a given task but is also fed information about the errors which are filtered back

through the system and used to adjust the weights ultimately improving performance. The BP algorithm independently proposed by Werbos, Parker, and Rumelhart (Werbos 1974; Parker 1985; Rumelhart 1986) effectively uses the delta rule method introduced in section 7.1.2.2 along with gradient descent (partial derivative) computation (see appendix section A.3 how gradient descent techniques guarantee finding at least a local minima). The general gradient type learning formula for each layer is given by:

$$w_{ij}[n] = w_{ij}[n-1] + \Delta w_{ij}[n] \quad (7.10)$$

The objective is to adjust the change in weights Δw_{ij} of a network that minimize the error *E* usually computed as the least-squares-error:

$$E = \frac{1}{2}\sum_{p=1}^{P}\sum_{i=1}^{N}[d_i^{(p)} - y_i^{(p)}(L)]^2 \quad (7.11)$$

$d_i^{(p)}$ is the target (teacher), $y_i^{(p)}$ the actual network output, *L* is the topmost output layer number of a multi-layer network, *P* the number of training patterns, *(p)* pattern index, and *N* is the dimension of the output space.

The weight change Δw_{ij} along with a learning rate scalar η and network error *E* can then be expressed as:

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} \quad (7.12)$$

By using gradient descent and *Chain Rule* approach $\frac{\partial E}{\partial w_i}, \frac{\partial E}{\partial \sigma_{ij}}, \frac{\partial E}{\partial c_{ij}}$ for the weights, covariance/standard deviation, and centers respectively we get the following update equations for our RBF/EBF networks:

$$e(x^{(p)}) = \frac{1}{2}\sum_{j=1}^{J}\{d_j^{(p)} - y_j(x^{(p)})\}^2 \quad (7.13)$$

$$w_i(t+1) = w_i(t) + \eta_w \sum_{p=1}^{N} e(x^{(p)})\Phi_i(x^{(p)}) \quad \text{when } i = 1,2,\ldots,M \quad (7.14)$$

$$w_i(t+1) = w_i(t) + \eta_w \sum_{p=1}^{N} e(x^{(p)}) \quad \text{when } i = 0 \quad (7.15)$$

$$\sigma_{ij}(t+1) = \sigma_{ij}(t) + \eta_\sigma \sum_{p=1}^{N} e(x^{(p)}) w_i \Phi_i(x^{(p)}) \frac{\left(x_j^{(p)} - c_{ij}\right)^2}{\sigma_{ij}^{\ 3}(t)} \quad (7.16)$$

$$c_{ij}(t+1) = c_{ij}(t) + \eta_c \sum_{p=1}^{N} e(x^{(p)}) w_i \Phi_i(x^{(p)}) \frac{\left(x_j^{(p)} - c_{ij}\right)}{\sigma_{ij}^{\ 2}(t)} \quad (7.17)$$

e is the error between the target *d* and actual computed output *y* for pattern index *p*; w_i the weights with center index i; η_w, η_σ η_c learning rates for weights, variance, and centroids respectively; Φ_i the activation function for center i; σ_{ij} the standard deviations between input dimension j and centroid i; *t* the time index; and finally c_{ij} the mean for centroid i's j^{th} dimension.

Many BP networks also include a bias w_o as shown in equation 7.15 and figure 7.8. The bias can be thought of as a unit which has a constant activation of 1. The bias is used to help in the overall weight training by giving extra degree of flexibility to the each layer. If for example the output uses a sigmoid function, the offset value w_o will influence where on the sigmoid function the weighted sum of activation outputs will correspond to, which in turn ultimately affects the final activation of the output and hence a pattern's classification.

7.2.3.1. Network Training – two stages

The training of the network is divided into two stages. The first stage consists of guessing initial parameters for the means, weights, covariance, and standard deviations (spreads). The second stage iteratively trains the network and updates parameters using equations 7.14 ~ 7.17. Each pass through all the sample patterns is referred to one epoch.

Stage 1 – guessing of the initial parameters can be achieved in two ways. One method is quite simply through random assignments of values of the initial parameters. The other more sophisticated method for initialization is through unsupervised learning via distance-based clustering algorithms.

In this dissertation I have used the k-means method to compute the initial parameters. With the k-means algorithm it is possible to obtain k number of

clusters, their respective centers, and class membership information. Using the membership information it is possible to compute the covariance matrix and the standard deviations. For RBFNs the standard deviations are computed by taking the norm of square-root of the diagonal covariance matrix components (radial spreads are constant in all dimensions). For the EBFN the respective dimensional standard deviation characteristics are preserved and used for the network training process.

To obtain the initial weights *w*, $\frac{\partial E}{\partial w} = 0$ is used to solve for *w* with respect to the total error squared *E* (see appendix section A.5 for derivation):

$$\text{Set}\ \frac{\partial E}{\partial w} = 0, where \quad E = \left(\vec{d} - A\vec{w}\right)^T \left(\vec{d} - A\vec{w}\right) \quad (7.18)$$

solving for w yields

$$w = (A^T A)^{-1} A^T d \quad (7.19)$$

A is the activation output in matrix form, *T* matrix transpose, *d* the target matrix, and *w* the computed initial weights. Figure 7.12 summarizes the initialization process.

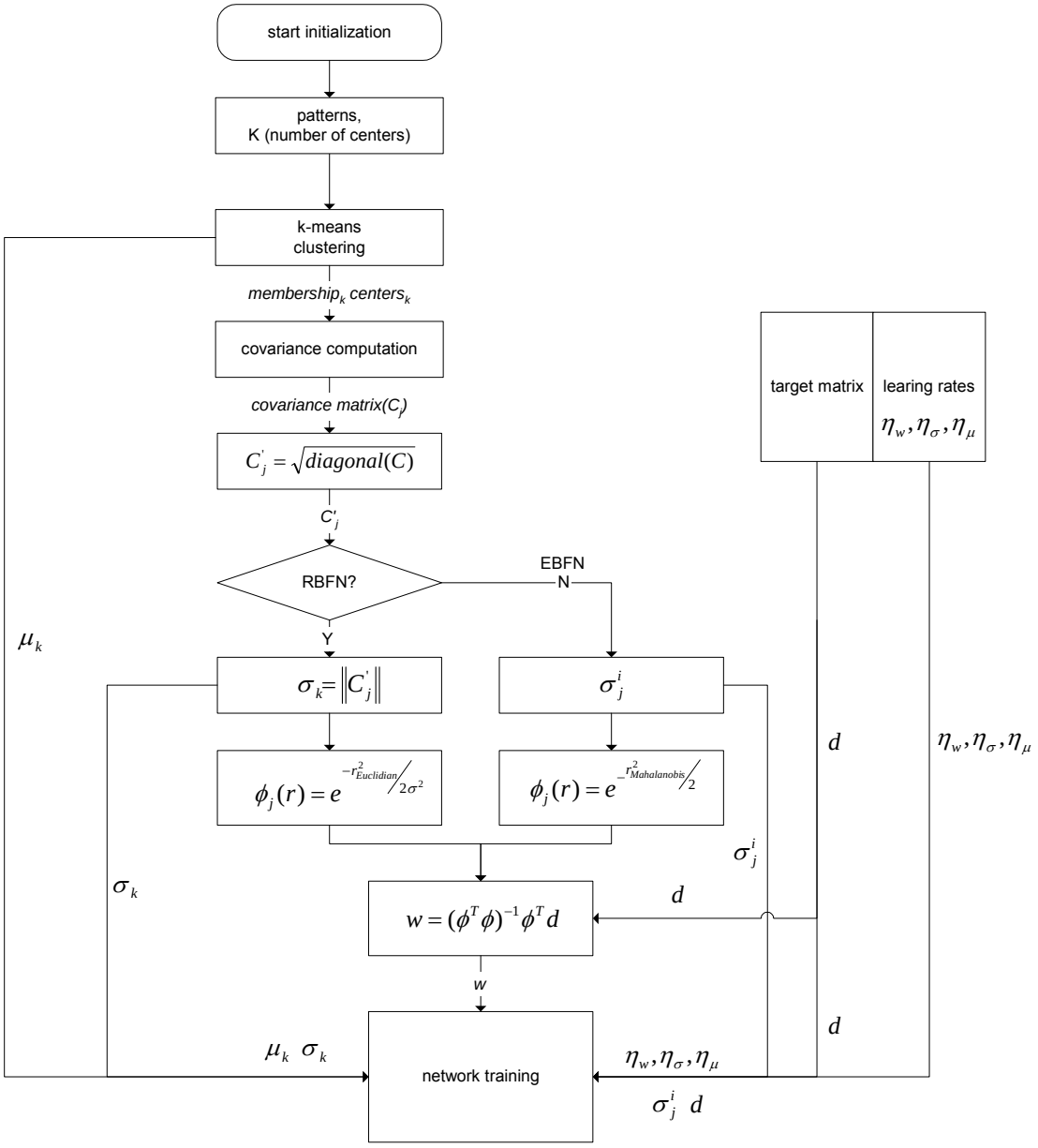


Figure 7.12 Initialization of network parameters

Once all the parameters are initialized the training phase can be started. It is a fact that training and tweaking the network parameters are often tedious and difficult. However, once the network is trained and a desirable performance

achieved, the actual recognition part of the network (not the training part) is extremely fast and robust to input data.

7.2.3.2. Network Initialization – one more stage

During the initialization stages of the network consisting of the k-means algorithm (means and spreads computation) and $\frac{\partial E}{\partial w}$ (weights computation), problems sometimes arise with the inverse matrix computation. That is, issues with singular or near singular matrices become problematic especially as the number of centroids increases. Generally speaking, more centroids achieve better results for a particular pattern space. However, the more centroids used, the more *specific* it becomes to the training samples used in the training process – this can lead to loss of generality. On a technical level, as the number of centroids increase, the spreads of centroids decrease causing a chain reaction of instability to the network. The size of centroids and instability is closely related to the inverse matrix operation. That is, the inverse term $(A^T A)^{-1}$ in the $\frac{\partial E}{\partial w} = 0$ solution may become unstable or blow-up (divide by zero) depending on the severity of singularity. This is in part due to centroids with very small spreads. For this problem, I have used singular value decomposition (Golub, Van Loan 1996) which estimates the inverse matrix while avoiding divide-by-zero scenarios.

EBFNs are particularly prone to singularity problems due to the additional requirement of the inverse covariance matrix in the Mahalanobis distance

algorithm. The cause of instability is of course once again some centroids' spreads becoming too small (close to singularity) and sometimes even becoming 0 (singular). Perhaps this is the reason why there is a scarcity of EBFNs being used in classification tasks – difficulty in controlling the stability of centroids. A number of different approaches were investigated to increase the consistency of EBFNs as they are more flexible in pattern recognition but at the same time more unstable.

Some of the ideas which did not do so well included using a running average filter to adaptively prohibit quick expansions or contraction of spreads. Another approach was to simply limit centers from getting extremely big or dangerously small using absolute minimum and maximum spread values in a timbre space normalized to +/- 1.0. After further investigating the behavior of misbehaving centroids, I found that in most cases, the misbehaving centroids went through a "big-bang" phase. In other words, the spreads of a particular centroid became smaller and smaller until it reached a near 0 value (near singularity) and then exploded, causing the network to break during the training phase. My approach of catching those big-bang centroids during the training phase itself was not producing consistently good results and was actually not the best approach as the supervisor would use "higher forces" to disturb the flow of the backpropagation algorithm.

It turned out that the best way, and consequently the simplest way to get more stable performance was identifying and taking care of possible problematic centroids during the initialization phase and not the training phase – before they get the chance to get out of control and blow up by which time it is too late. I have tried reducing or increasing the spreads of centroids with extreme spreads without good results. However, the method which rendered the best results was eliminating those centroids with extreme spreads (small and large) from the network altogether. Fortunately, this rather drastic approach works in most situations as the number of potential extreme centroids were generally less than 3.

Figure 7.13 illustrates the instability of RBFN/EBFNs during training for classifying instruments into 3 different families (woodwinds, brasses, strings) with four input dimensions (spectral spread, spectral centroid, attack time, and shimmer), 2000 epochs, 1 ~ 100 centroids, and 5 training iterations for each centroid configuration. As we can see for EBFNs, the error rate starts becoming very unstable around 35 centroids and turns into a blind guessing game by 77 centroids. For the RBFN it is not quite as severe although the mean correct percentage also becomes unstable around 70 centers using the same testing environment.

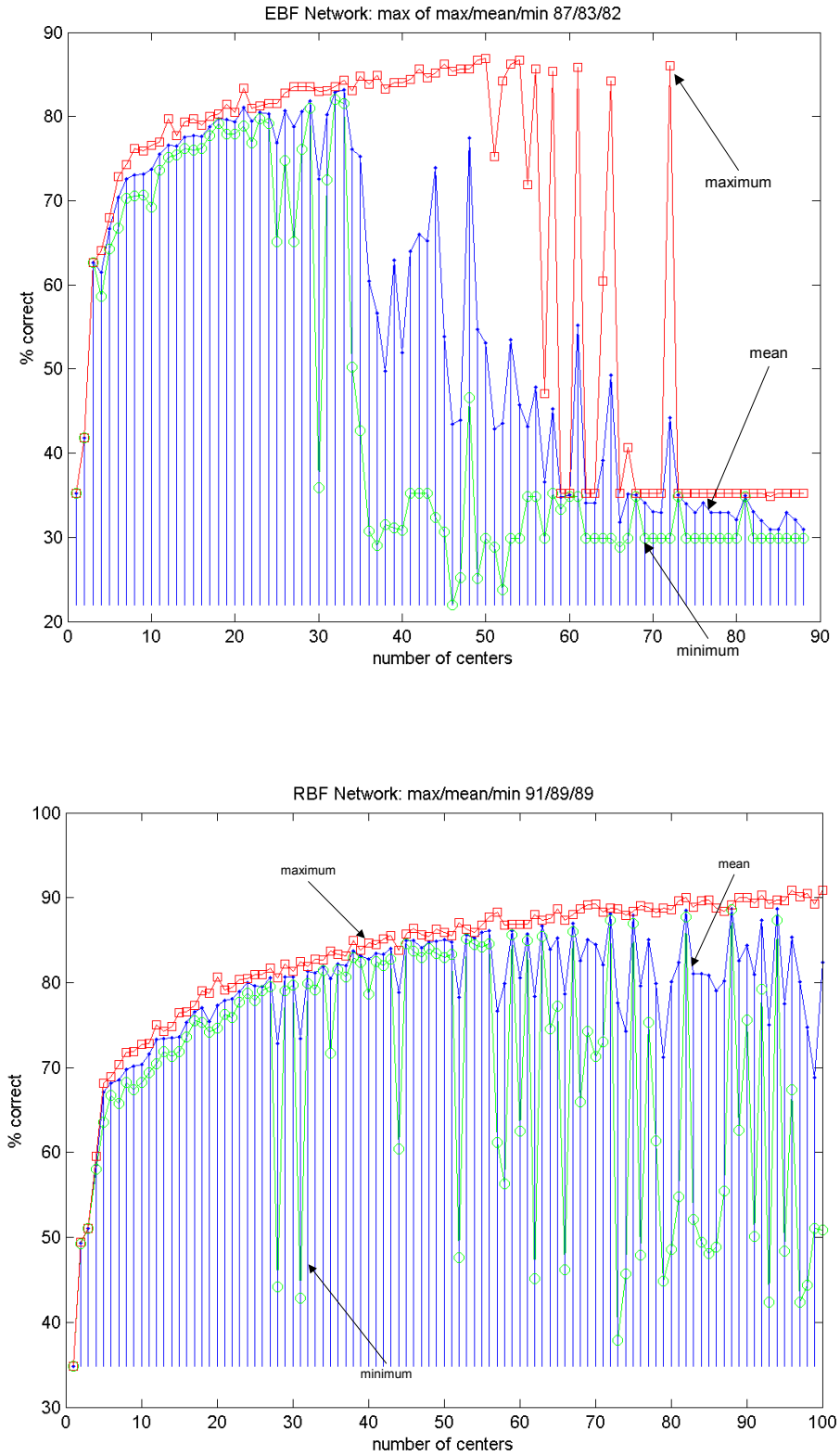


Figure 7.13 Instability of RBF/EBF networks

7.2.3.3. Further Fine-Tuning: Nearest Centroid Error Clustering (NCC)

At the end of the day, neural networks for pattern recognition are in essence fine-tuning systems which need all the help they can get from the supervisor and also other algorithms for classification tasks. In this thesis for example, I have used k-means for initial guessing of the neural network parameters – the spreads (σ) and centroid locations (μ) obtained from the k-means are fine-tuned using the RBF/EBF network. However, another stage for further fine-tuning was applied. In this section I will describe an algorithm I have developed that further fine-tunes the network automatically using error feedback which incidentally occur mostly between class boundaries, especially in those areas where ambiguity and overlap of class boundaries are particularly problematic.

Designing and training a network blindly with clustering algorithms and simple backpropagation is tedious to say the least, and furthermore does not particularly take advantage of the supervisor's knowledge of error prone areas (during training) for a specific problem. Considering these facts, would it not be sensible to place "an appropriate" number of "smaller" and finer centroids in those areas where errors occur or at least seem to be lumped together? To better explain the "nearest centroid error clustering" (NCC) algorithm let's look at figure 7.14.

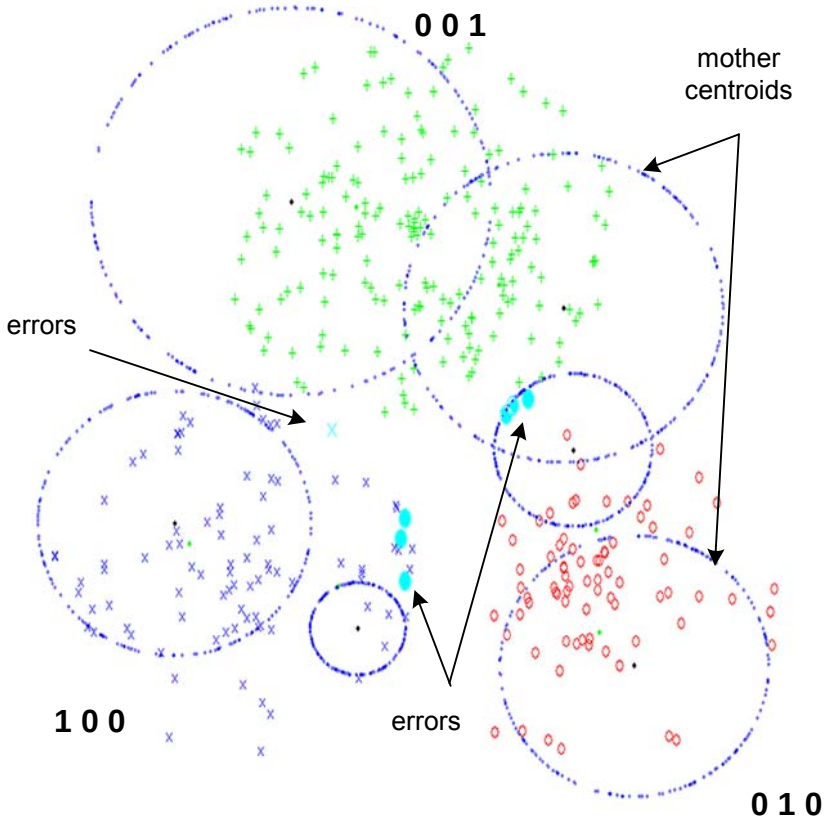


Figure 7.14 Initial training with a 6 centroid RBFN

As seen in figure 7.14, 6 centroids have been arbitrarily chosen for the k-means algorithm to classify patterns in a two dimensional space with Gaussian noise using 2 inputs, 3 outputs, 3 classes – 001, 010, 100 (these are the desired outputs) . We notice that for this problem the initial training (k-means + initial NN training) does a good job in the classification task – approximately 94% correct classification only after 200 epochs for 360 samples. It can also be noted as previously mentioned, that the errors occur in areas where two or more class boundaries overlap or are close together. Hence, if one could automatically place additional centroids at those problematic locations one could possibly

foresee an improvement in the performance of the network after further training – an additional fine-tuning training stage concentrating on localized regions with small centroids. This in essence is the basic idea of the proposed method.

The algorithm determines the locations of these “problematic areas” using information from “mother centroids” (original 6 mother centroids for this example) and spawning new smaller children centroids nearest to a particular mother centroid (see flowchart in appendix A.8.2 for details). The reason for using the mother centroids as a guide for computing the new children centroids is because the mother centroids are already “roughly tuned” in size and location to a particular pattern space. The spreads of the mother centroids or the size of the mother centroids tend to be in the mid to large-size range. In general, the mother centroids are larger when there are less centroids and become smaller when the total number of centroids increase. This is like trying to fit circles or ellipses in a rectangular space (in this example); the more circles and ellipses one has, the more centroids with smaller spreads will result and vice-versa. Although just blindly increasing the number of centroids is an option for better performance (albeit not the most intelligent nor most effective), there is no consideration of the error feedback reflecting the pattern space. The nearest centroid error clustering method on the other hand achieves rapid and more consistent increased performance utilizing learned information about the pattern space without having to manually set the number for additional centroids. The following diagrams show how the algorithm works for the 3-class, 2-dimensional pattern space.

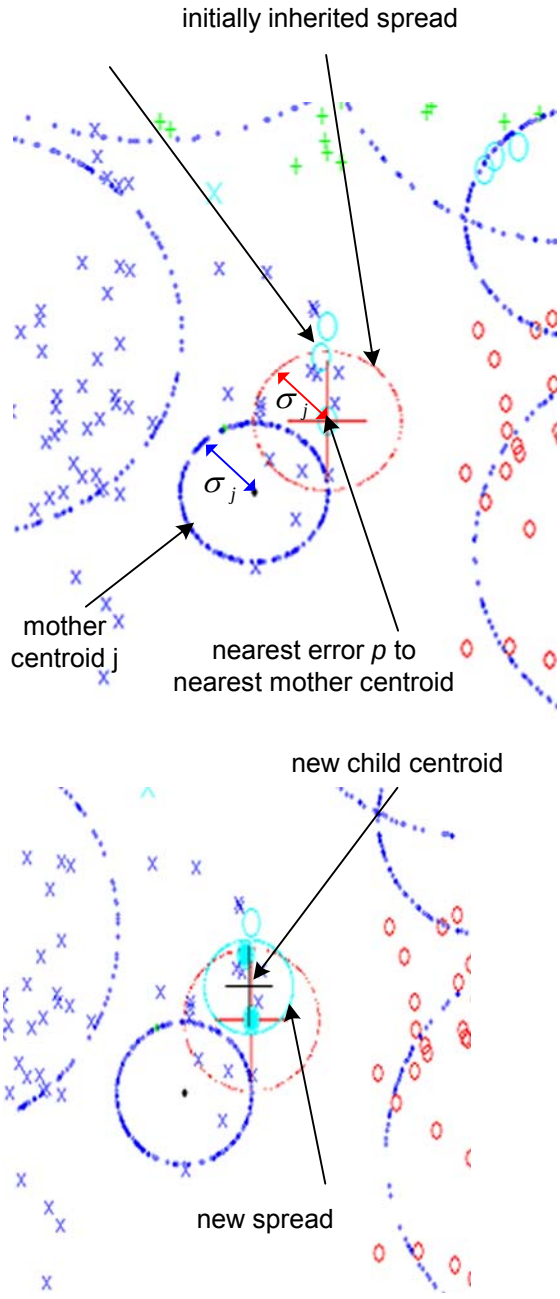


Figure 7.15 Spawning a new child centroid

Figure 7.15 shows a zoomed-in section of figure 7.14. As seen on the top of figure 7.15 the error pattern and mother centroid which render the minimum

distance is initially chosen. Once the error pattern *p* and mother centroid *j* is selected, the new child centroid pertaining to pattern *p* inherits the mother's spread σ_j (for RBFs this is a scalar, whereas for the EBF it is a vector). The mother's spread σ_j is then used to find any encompassing error pattern neighbors satisfying the general case hyperellipsoid equation 7.20 (*N* is the dimension index; *m* and *p* pattern and center indexes). If any error pattern members are found within σ_j, a new spread is computed via the covariance function and a new center is computed via the arithmetic mean reflecting the members of the child centroid as shown on the bottom of figure 7.15. In this example, the new child centroid has one "sibling."

$$\left(\frac{(q_m^{(1)} - q_p^{(1)})}{\sigma^{(1)}}\right)^2 + \left(\frac{(q_m^{(2)} - q_p^{(2)})}{\sigma^{(2)}}\right)^2 + \ldots + \left(\frac{(q_m^{(N)} - q_p^{(N)})}{\sigma^{(N)}}\right)^2 \leq 1 \quad (7.20)$$

$$\boldsymbol{\mu}_p = \frac{1}{M}\sum_{m=1}^{M} \mathbf{e}_m, m \in \{members \quad of \quad new \quad child \quad centroid \quad p\} \quad (7.21)$$

On the other hand if there is only a single member within spread σ_j (the nearest neighboring child centroid to the nearest mother centroid), the new child's spread is just scaled linearly to decrease its span of influence.

$$\sigma_p = \alpha \cdot \sigma_j^{mother}, \text{ where } 0 < \alpha < 1 \quad (7.22)$$

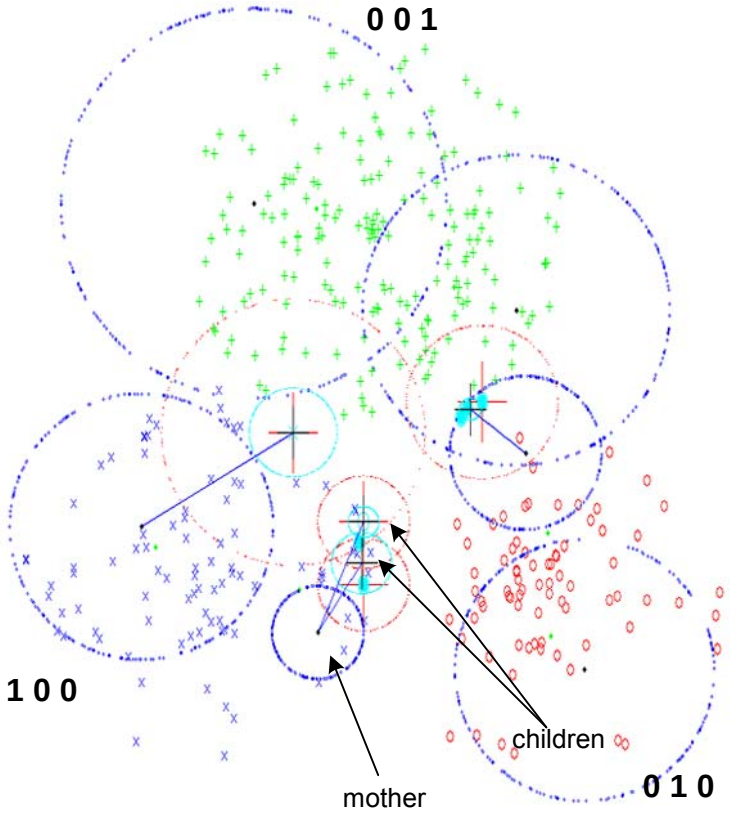


Figure 7.16 Final results after clustering

This process is repeated and new centroids with reduced spreads (σ) and new "error-pattern-influenced" centers (μ) are automatically obtained. In order to resume training the network with the newfound children centroids (using only the mother centroids), the weights of the newly introduced children centroids have to be initialized. The weight initialization can be done in two ways – trying to manually approximate it via teacher and mother centroid information and doing it using equation 7.19 via $\frac{\partial E}{\partial w}$. A number of ways are possible for the manual method, including taking the mean of the all the weights in network or a subset of a selection of mother centroids; giving them no bias at all by initializing them to a

scalar value among other possibilities. To use as much information possible from the existing mother centroids, which after all represent a good approximation of the feature space, I tried utilizing parameters regarding location of the error patterns, target vector corresponding to the children centroids, and the mother centroids' weight patterns to initialize the children's weights. It can be observed that the weights corresponding to the output form strong connections to those output components that reflect a particular class. For example, in the 3-class, 2-input, and 3-output pattern space the weights connected to the largest centroid is 0.1, -0.01, and 0.95. Not surprisingly this weight pattern corresponds to the desired output pattern 0, 0, 1. In other words, the weights of the upper-left centroid are trained to help the cumulative output of the basis activation functions to correspond to a particular target value. Exploiting this tendency of the behavior of the weights, it is then possible to initialize the children centroids using the mother centroids' weights as a guide since the mother centroids (weights, centroids, spreads) are already roughly tuned. However, the children centroids will not always have the same weight pattern as the mother weights. By weight patterns I am referring to the same strong weight components pointing towards a particular class (1 0 0, 0 0 1, 0 1 0). In such cases, I have just used a swapping method whereby the weight component that is congruous to the error pattern's correct output pattern is chosen. Figure 7.17 shows the final result of the 3 class classification problem with a 6% increase in performance (100%) and a total of 10 centroids from 6.

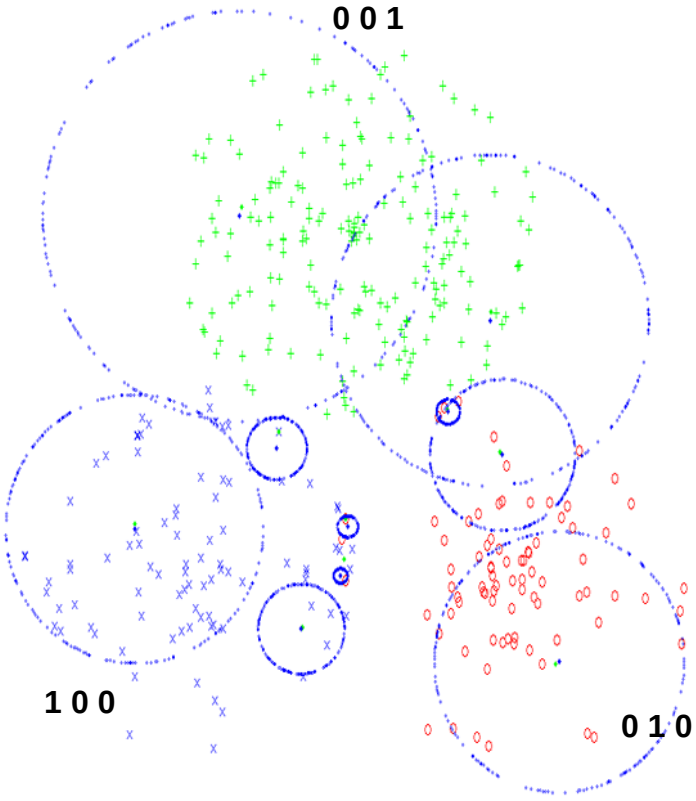


Figure 7.17 100% with new children centroids

Although the manual method for weight initialization resulted in improvement, it was not always consistently reliable. Using the $\frac{\partial E}{\partial w}$ to compute the weights, i.e. equation 7.19, yielded better and more consistent results.

In theory this additional fine-tuning method could be applied a number of times until a desired performance is achieved. However, due to the nature of the algorithm, there will usually be an increase of the number of total centroids as long as errors exist, and as a result over-fitting issues may arise. A partial solution to over-fitting was to include an optional parameter which would control whether to include “single-member” children centroids as these tend to address

only very localized error patterns. This methodology would not only lessen the overall increase of centroids, (generally a good idea), but also only include those new children centroids with spreads that have at least two members associated with it in total. Alternatively, it would also be possible to group together neighboring single-member children centroids and form a bigger centroids to encompass at least one other child centroid. However, further problems would arise with this method as the distance between "nearest" single-member children centroids could in actuality be quite distant, and not really help the network in the fine tuning process.

7.3 RBFN/EBFN Test on 2-Dimensional Classification Problem

The following examples show the implemented RBFN/EBFN network's performance classifying a two dimensional pattern space into two classes. The two-dimensional patterns (p = [x-coordinates, y-coordinates]) generated via Gaussian white noise consists of points in a rectangular space with width 2 and height 1 ratio. As seen in figure 7.18 the two classes are separated by dividing the rectangular area into 4 regions where the pattern membership is denoted as "x" and "o." The top-bottom triangular regions make up one class and the left-right triangular regions make-up the other class. The objective is to train the network to classify the top-bottom triangular regions (x) into one group and left-right triangular regions (o) into another. The network in figure 7.18 used 4 centers and a single binary output. The output was trained to output a number as close to 1 as possible for the top-bottom triangular regions and -1 for the left-

right triangular regions. The error *e[n]* was simply computed as $e[n] = actual_p[n] - target_p[n]$ where *n* is the epoch index and *p* the sample number.

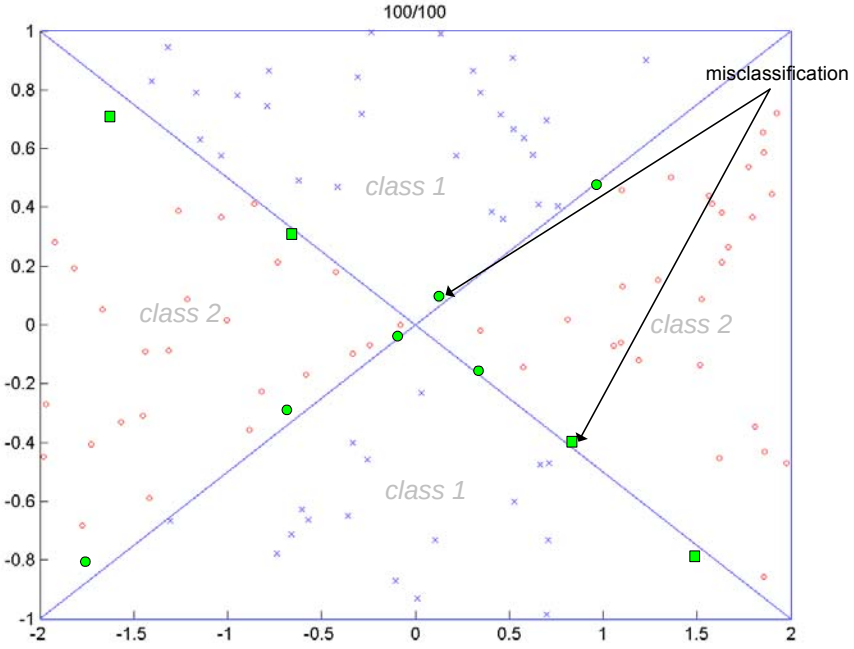


Figure 7.18 EBFN pattern classification results

The result for this problem using EBFN and 4 centers was approximately 82%~91% correct classification using 100 sample points after training the network for 100 epochs. Using RBFNs resulted in approximately 80% ~ 86% classifications.

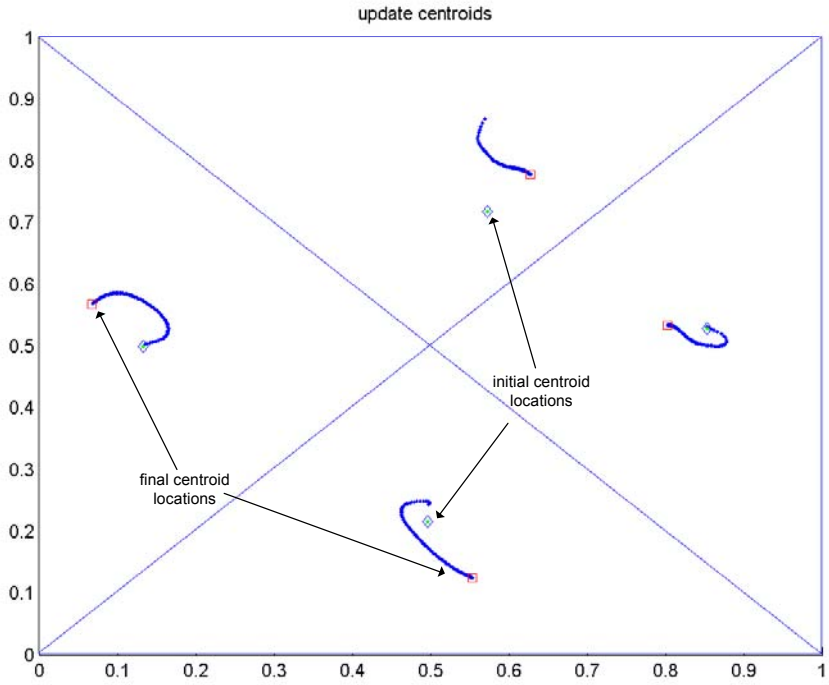


Figure 7.19 Centroid locations updates

It is not that surprising the EBFNs in this case and perhaps in general, perform better than RBFNs due to their independence in dimensional spread– “x axis” spread and “y axis” spreads in this example. It is somewhat like trying to fit a circle into the four triangular regions (RBFNs) versus trying to fit ellipses (EBFNs). However, with only 4 centers, the performance results vary a lot due to the high dependency on the initial rough estimate by the k-means algorithm. Increasing the number of centroids to 12 centroids as seen in figure 7.20 as well as more epochs (5000, although at around 3000 epochs things are pretty stable) the performance of the system increases considerably with an average correctness of approximately 94% ~ 96% for RBFNs.

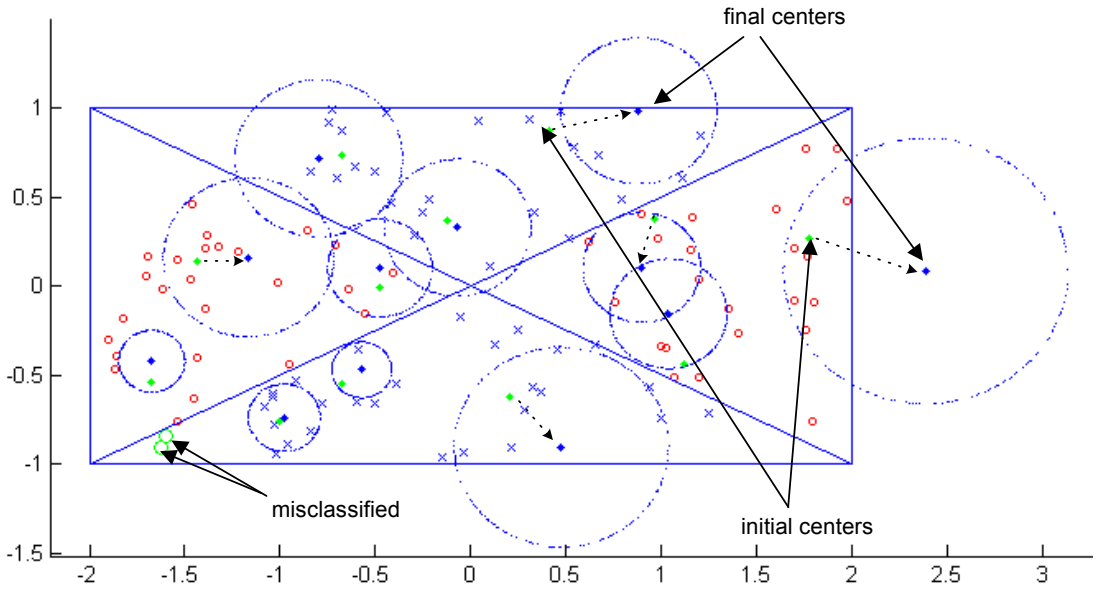


Figure 7.20 Twelve centroids RBFN behavior: 98% correct

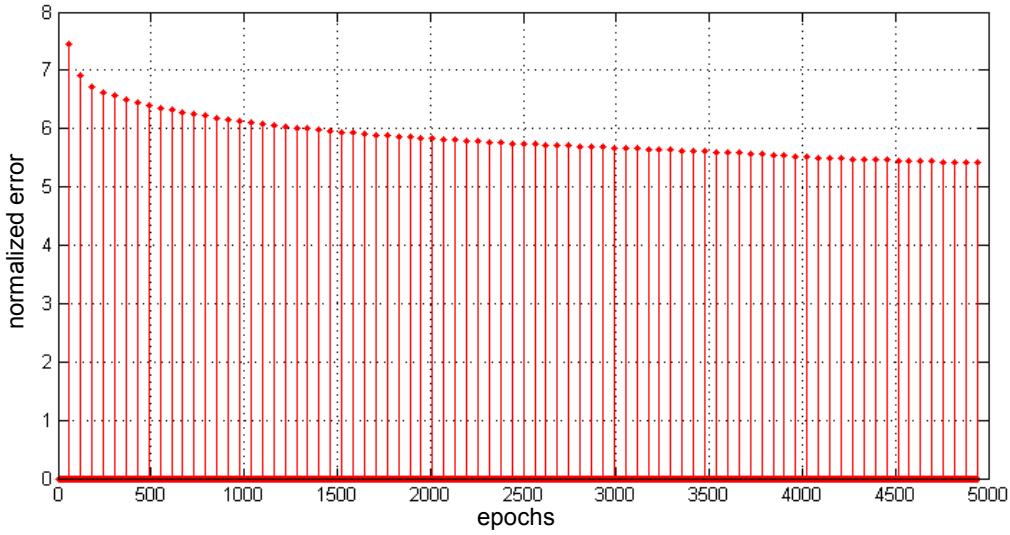


Figure 7.21 Twelve centroids RBFN error plot

It is also noticeable that the errors mostly are clustered at the class boundaries especially where the four diagonal lines intersect. Figure 7.23 shows the

application of the nearest centroid error clustering (NCC) method starting on a 4 centroid initial training example at 85% correct classification which is increased to 94% (15 centroids) after training is complete in figure 7.24.

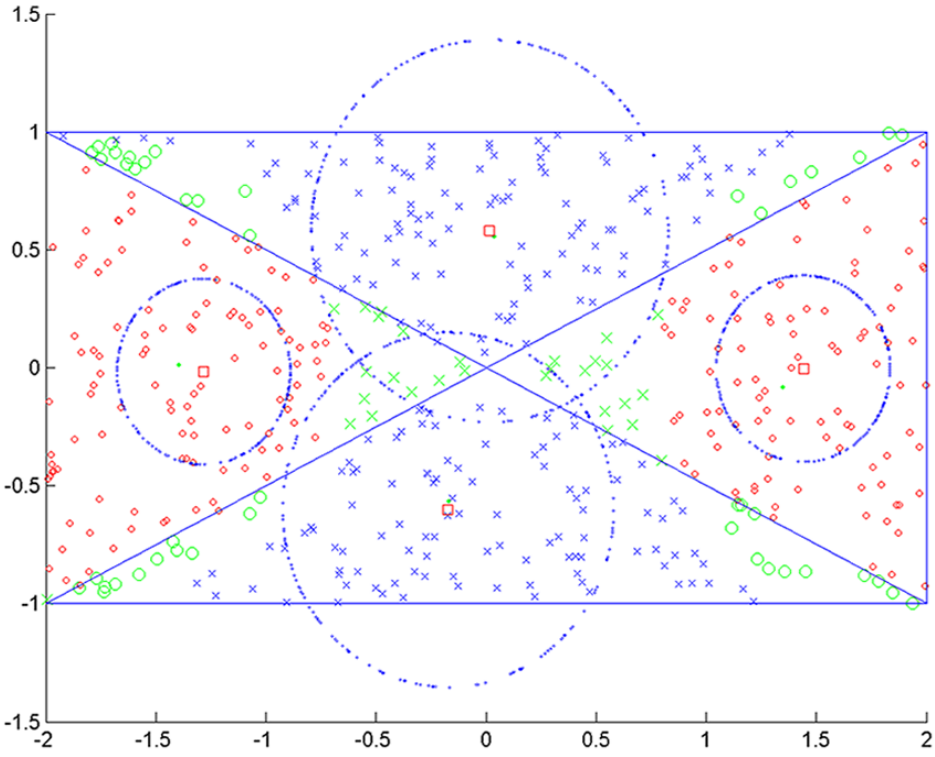


Figure 7.22 4 centers at 500 epochs @ 85 %

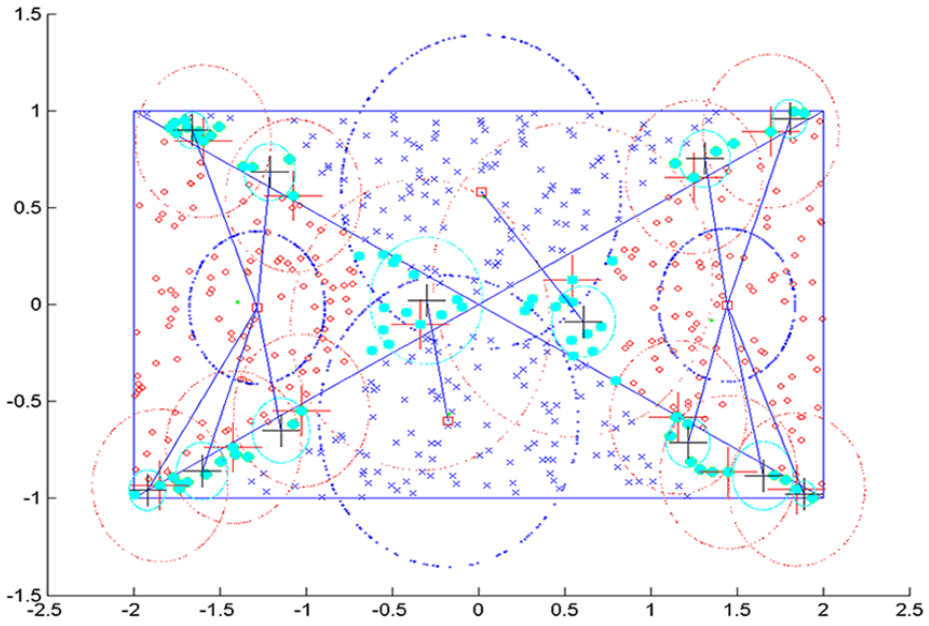


Figure 7.23 Spawning of children centroids: 15 centroids 85 %

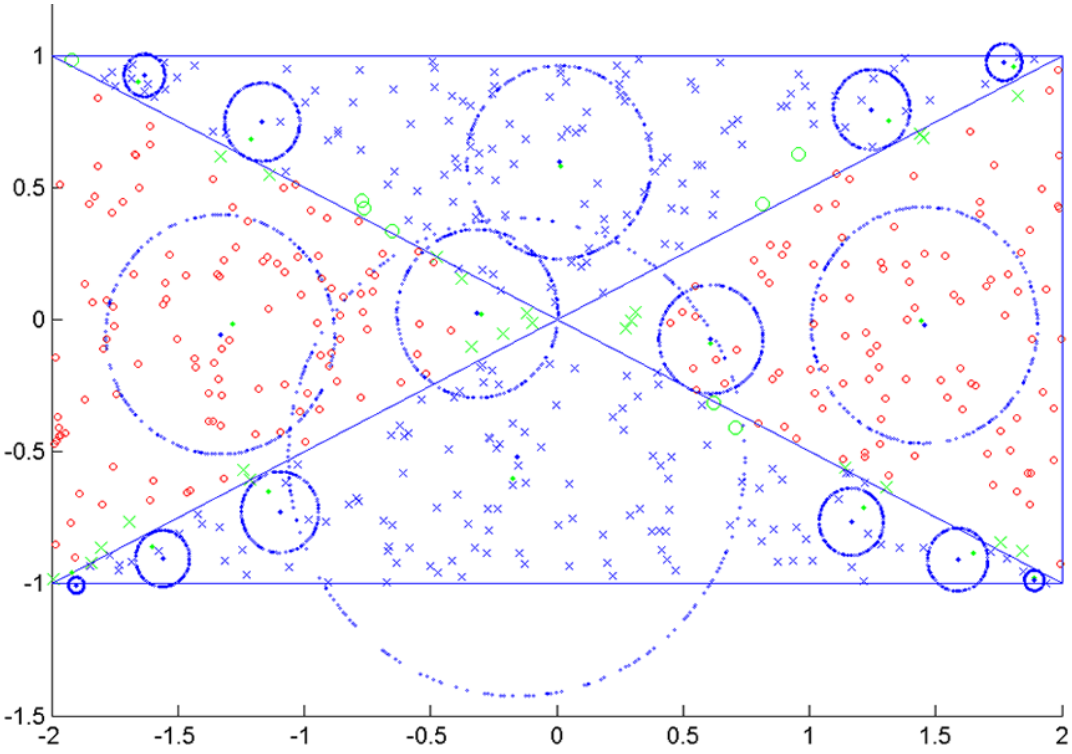


Figure 7.24 Retraining after spawned children: increase from 85% to 94%

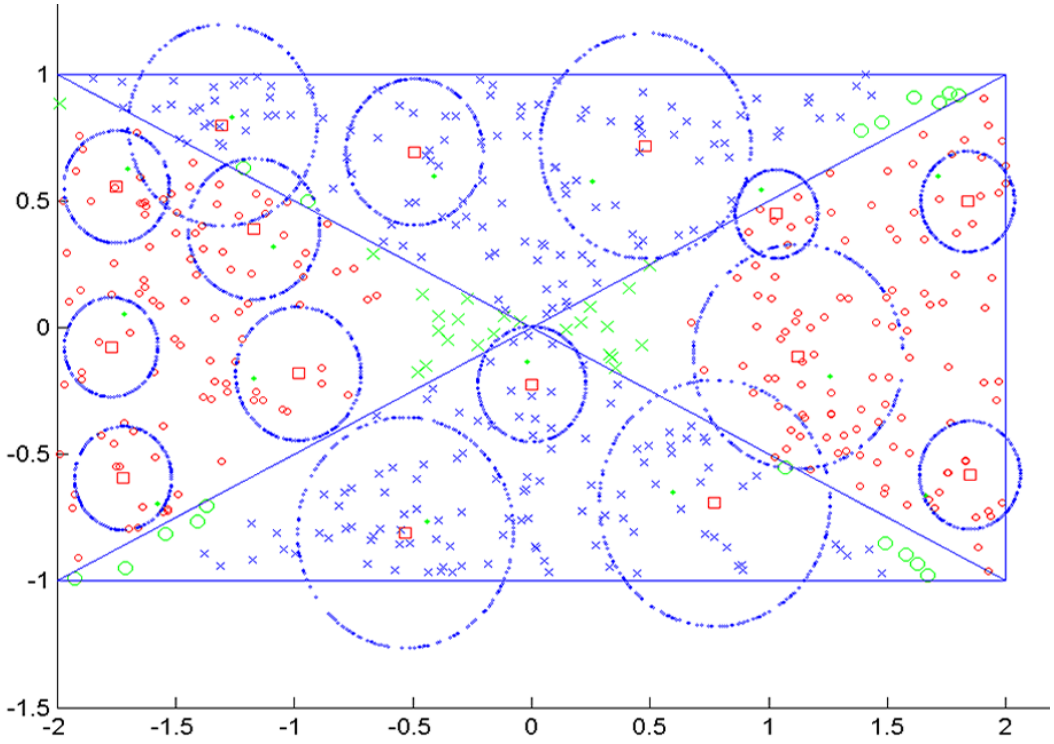


Figure 7.25 Using 15 centers from the onset without spawned retraining: 92%

Figure 7.25 shows training of the network using the same number of centroids (15) without the nearest centroid clustering algorithm (NCC) resulting in 92% correct classification. It can be noted that with the application of the NCC method, the centroids more accurately “draw” the class boundaries compared to the normal training method.

8.1 General Testing Environment

The testing environment for automatic instrument timbre recognition is summarized in figure 8.1. The database consists mostly of Western orchestral instruments divided into three sections – strings, brasses, and woodwinds (table 8.1) totaling 829 samples and 12 instrument types as listed in table 8.2. A total of 12 features were used (table 8.3), with feature sets of lower dimensions resulting in superior system performance (see section 8.3.1 for details).

All samples were temporally complete; however the decay part of a sample was not used for feature extraction (only first two seconds of samples). In order to have as many "realistic" sounds as possible, various dynamics were used for each of the instruments including *pianissimo, piano, mezzo-forte, forte,* and *fortissimo*. Also, different techniques and articulations such as *long, short (staccato)*, *pizzicato*, *con sordino, detaché, vibrato, non-vibrato, and espressivo* techniques were used. Finally, pitches were not normalized in any way nor limited to one specific note; rather a large range of pitches for the majority of the instruments depending on sample availability were included. Appendix section A.9 lists details of the instruments, articulations, pitches, and performance techniques included in the tests.

The development of the algorithms was primarily done in Matlab, which in most cases were optimized using linear algebra and matrix manipulations.

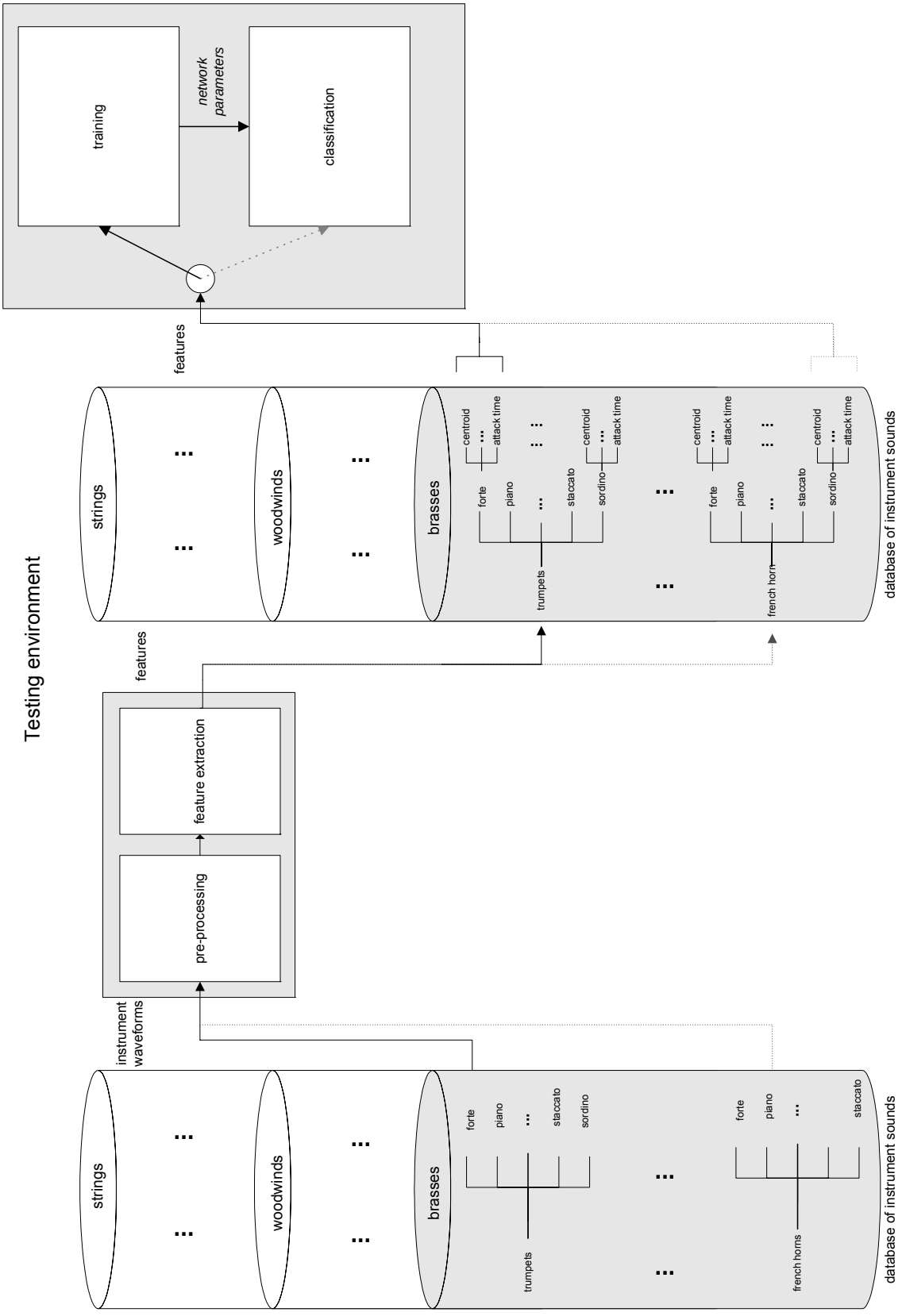


Figure 8.1 Testing environment

Family	Number of samples
Strings	292
Woodwinds	190
Brasses	248
Total	**829**

Table 8.1 Instrument family list

Instruments	Number of examples
1.Electric bass	10
2.Violin	105
3.Cello	102
4.Viola	75
5.Bb clarinet	100
6.Flute	99
7.Oboe	55
8.Bassoon	35
9.French horn	56
10.Trumpet	78
11.Trombone	82
12.Tuba	32
Total	**829**

Table 8.2 Instrument list

Feature number	Feature name
1	Shimmer
2	Jitter
2	Mean spectral spread
4	Mean spectral centroid
5	Noise content
6	Inharmonicty
7	Attack time
8	Harmonic slope
9	Harmonic expansion/compression
10	Spectral flux shift
11	Temporal centroid
12	Zero crossing rate
Total	12 features

Table 8.3 Full set of features used in tests

8.2 PCM File Specifications

The sound samples were all single notes recorded in isolation (no polyphonic notes). Sampling rate was set to 22,050 Hz (down-sampled if f_S > 22050 Hz) and the first 2 seconds of each instrument was excerpted for feature extraction and classification.

Sampling rate	**22,050 Hz**
Resolution	16 bits
Duration	2 seconds excerpts

Table 8.4 PCM file characteristics

8.3 Testing Procedures

The implemented feature extraction algorithms were thoroughly tested with *all* of the samples and verified manually whenever possible. The system's classification performance and training was conducted in the following ways:

1. Using the leave-one-out procedure.

   80 %of the total 829 samples were used for the training of the network and the remaining 20 % of samples were used for cross-validation.

2. Training the system for instrument family recognition

   Determination of best feature sets.

   Separate feature sets resulting in separate network parameter sets.

3. Training the system for individual instrument recognition

   Determination of best feature sets.

Separate feature sets resulting in separate network parameter sets.

The leave-one-out cross-validation procedure was implemented using “pattern shuffling” before being subjected for training. That is, the original 829 patterns were shuffled randomly, split into 80% training and 20% cross-validation sets, and then subjected to network learning and cross-validation procedures.

8.3.1Salient Feature Selection

In order to obtain best performance results from a neural network it is important to know which feature sets achieve those results. Preliminary tests suggested that feature sets smaller than the maximum number of features (12) yielded higher system performance. A method similar to sequential backward generation (SBG) suggested by Liu (Liu 1998, see section 5.4) which steps through a selection of prominent feature combinations and picking out those feature sets that result in best performance was developed. One of the difficulties in selecting the best possible features via network training is that there are five parameters that affect the performance results of RBF/EBF networks. Those parameters are the number of centroids, number of epochs, number of input dimensions, number of output dimensions, and cross-validation results. In general, as the number of epochs increase, the performance of the system increases accordingly – the same is true for number of centroids. However, with the increase of epochs, over-fitting becomes an issue and the optimal number of epochs is difficult to determine as the number of epochs in turn depend on the centroid population as

well as input and output dimensions size. Although an optimal set of features may achieve good results using *x* number of centroids during the training phase, this may not necessarily be the best set of features or number of centroids for cross-validation. Figure 8.2 shows an example where the training results for a 54-centroid network is higher than a 51-centroid network, but the cross-validation results are the opposite (lower group is cross-validation and higher group is training results). Additionally, even though a network may be initialized with the same number of centroids (for example 40 centroids), each time it is trained to adapt to the input patterns, the classification results will not be exactly the same and in some instances may vary a lot (this is due to k-means clustering which includes a randomization process in its algorithm). To maximize the selection of the best set of features, each feature set was tested 5~10 times each with different number of centroids (30 ~ 50 centroids). The feature set that rendered the best performance was selected.

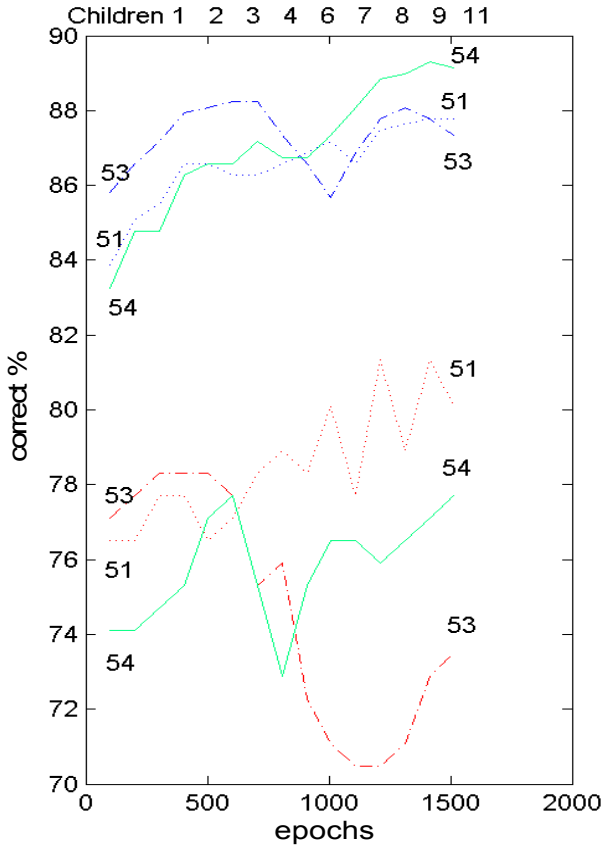


Figure 8.2 Better training results but poorer cross-validation results for 54-centroid children network

Tables 8.5 and 8.6 show prominent features stets for family and individual instrument classification obtained from the "best feature selection" process (the tables were constructed using output data from 11 separate tests). The algorithm starts off with a full set of features {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12} and gets filtered down to a feature set that yields the best results.

Cross-validation %	1.Shimmer	2.Jitter	3.Spectral spread	4.Spectral centroid	5.Inharmonicity	6.Attack time	7.Harmonic slope	8.LPC noise content	9.Harmonic expansion/ contraction	10.Spectral flux	11.Temporal centroid	12.Zero crossing rate
88	O	O	O	O			O	O	O		O	O
85	O	O	O	O		O		O	O		O	O
84	O	O		O		O	O	O	O	O	O	O
83	O		O	O	O	O	O	O	O	O	O	O
82	O	O	O	O			O	O			O	O
82	O	O	O	O				O	O			
71	O	O		O	O	O	O	O	O	O	O	O
65	O	O	O									
61	O	O	O									

Table 8.5 Feature set selection for family classification

Cross-validation %	1.Shimmer	2.Jitter	3.Spectral spread	4.Spectral centroid	5.Inharmonicity	6.Attack time	7.Harmonic slope	8.LPC noise content	9.Harmonic expansion/ contraction	10.Spectral flux	11.Temporal centroid	12.Zero crossing rate
71		O	O	O			O	O		O	O	O
69		O	O	O		O	O	O		O	O	O
69	O	O	O	O		O	O	O		O	O	O
68	O		O	O				O	O	O	O	O
66		O	O	O		O	O	O	O	O	O	O
65	O	O	O	O		O	O	O		O	O	
65	O	O	O	O		O	O	O			O	O
64	O	O	O	O	O	O	O	O	O	O	O	O
62	O		O	O			O				O	
56	O		O	O			O	O	O	O	O	O
44	O		O	O	O	O	O	O	O			
42			O	O			O					
30			O				O					

Table 8.6 Feature set selection for individual instrument classification

It can be observed from the above tables that for family classification, shimmer, jitter, spectral spread, and spectral centroid especially seem to be salient

frequency domain features. Also, in order get highest results, other spectral features such as harmonic slope, and harmonic expansion/contraction were found to be salient features. For time domain features, LPC noise content in particular was an important feature along with temporal centroid and zero crossing rates.

There was a considerable amount of overlap between individual instrument features and family features as seen above. For individual instruments, features 2 ~ 4 seemed quite important for high classification rates along with spectral slope and spectral flux which were not as critical in family classification. Unlike table 8.5, harmonic expansion/contraction and shimmer was not as omnipresent in the higher performance feature sets and attack time was not included in the best 71% cross-validation feature set. Generally speaking, attack time was a little more relevant for individual instrument classification, and as before LPC noise content, temporal centroid, and zero crossing rates were important features. Interestingly, inharmonicity seemed to be the least desired feature for both family and individual instruments, although variants of it like the harmonic slope and harmonic expansion/contraction seemed to be chosen as salient features.

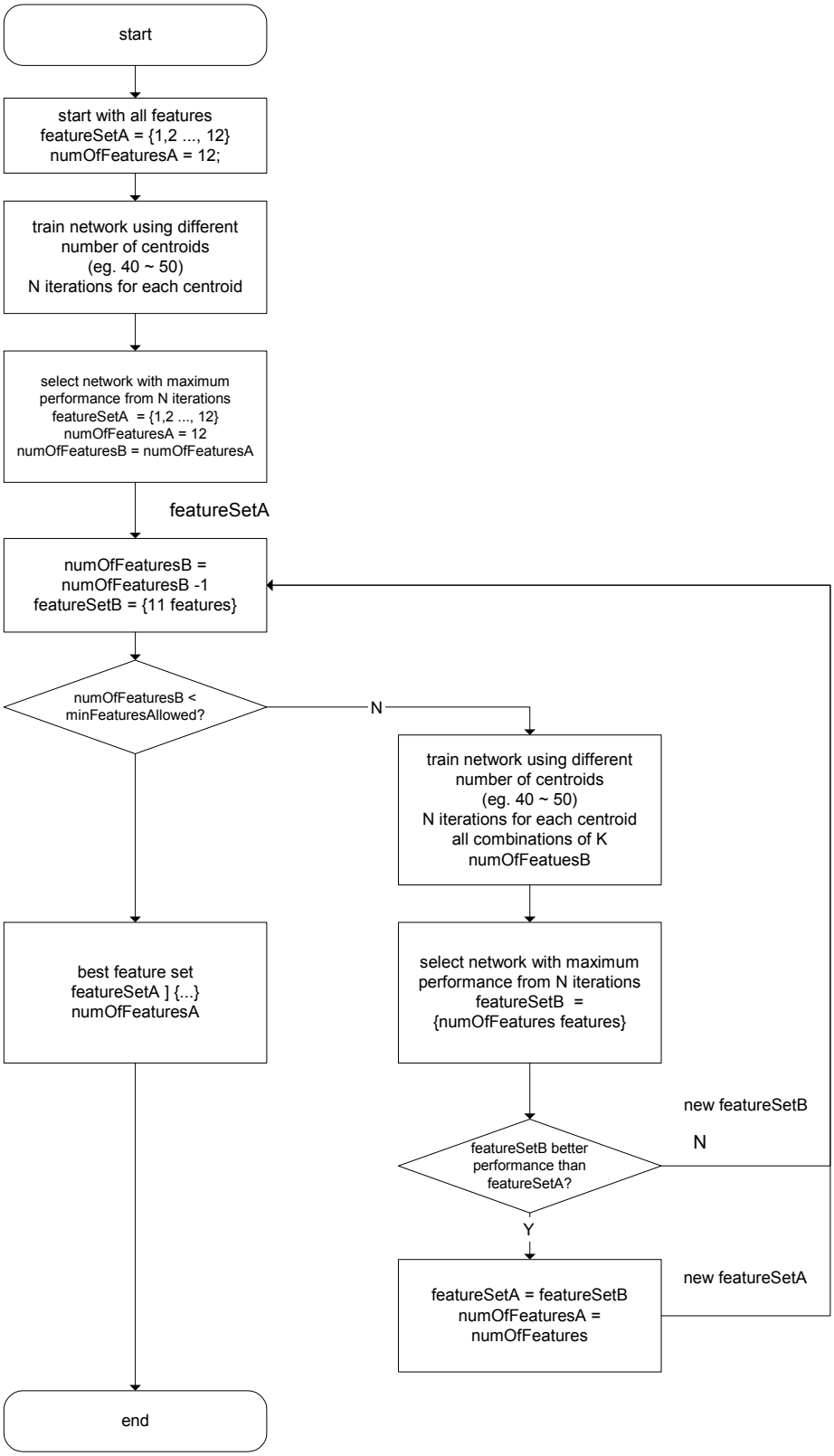


Figure 8.3 Salient feature selection process

8.4 System Performance and Results

One interesting result occurred when determining family membership not via parameters obtained through "family network training" using three outputs (strings, woodwinds, and brasses), but applying parameters and output values obtained through an estimated "individual instrument network training." The alternate "estimated" family classification method (through "individual instrument network training") was simply achieved by considering the cumulative (net) output of each instrument belonging to a particular family category as seen in table 8.7 and equation 8.1.

Strings q(0)				**Woodwinds q(1)**				**brasses q(2)**			
0.5				1.8				- 0.7			
Bass	Violin	Cello	Viola	Clarinet	Flute	Oboe	Bassoon	French horn	Trumpet	Trombone	tuba
-0.1	0.9	0.2	-0.5	0.7	0.8	0.5	-0.2	-0.1	-0.2	-0.2	-0.2

Table 8.7 Estimated family classification example

$$y_q^{family} = sum(y_{l-m}^{individual}) \quad (8.1)$$

It turned out that the "estimated" method for classifying family membership helped in increasing the performance of the individual instrument and family classification – this was achieved and developed using a "confidence level" (CL) method described below (see figure 8.4 below for basic outline of algorithm).

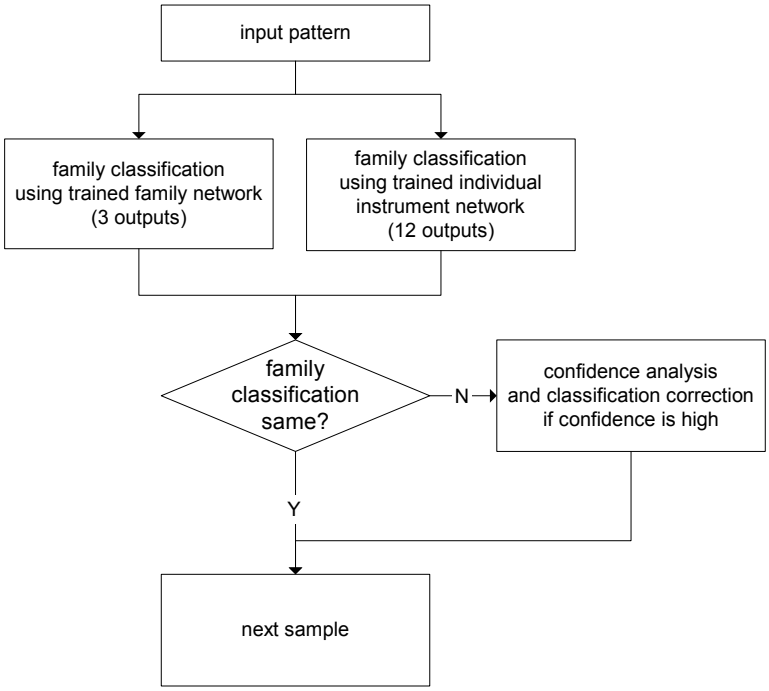


Figure 8.4 Outline of “confidence level” analysis

While considering the “confidence levels” of the network outputs it was possible to use the “estimated” family classification results (network trained with 12 outputs) and the “family trained” results (network trained with 3 outputs) to further increase the classification success rate by an average of approximately 3 percent for families and 2 percent for individual instruments. By analyzing those network classifications that resulted in incongruent family classifications (between the “family trained” network of 3 outputs and “estimated” individual instrument trained network of 12 outputs) it was observed that in some cases one method was classifying an instrument into a family category with little “confidence” while the other method with high “confidence.” Figure 8.5 shows an example of such a situation.

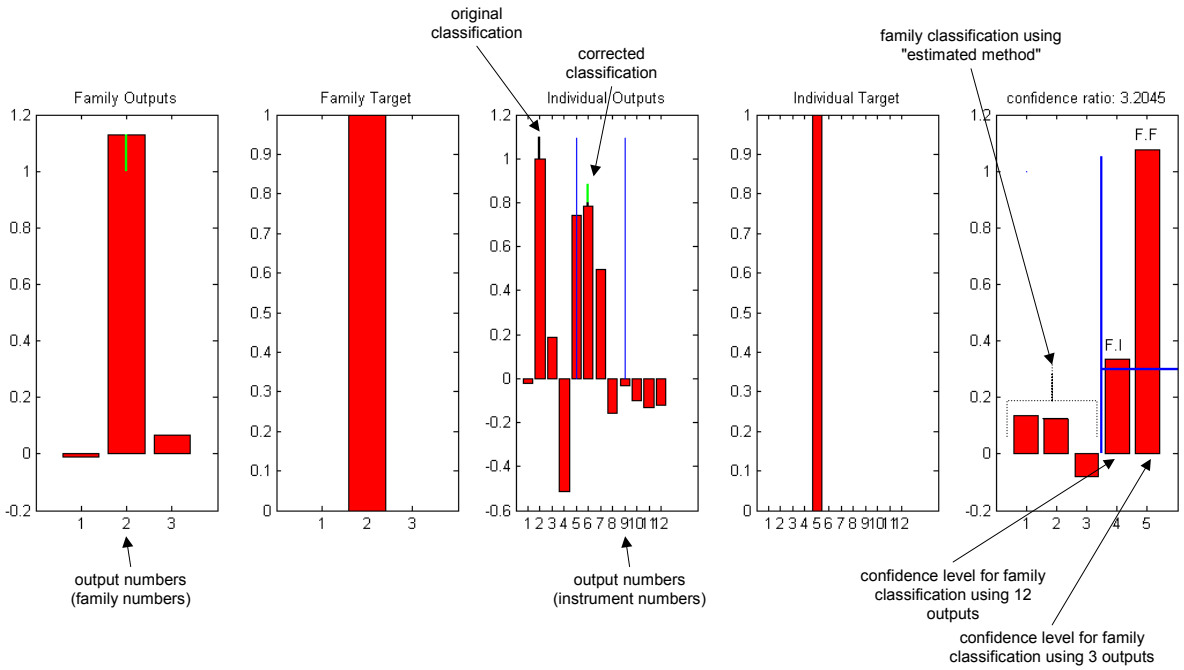


Figure 8.5 Confidence level (CL) outputs

The “confidence level” was computed using all three “new” outputs from the “estimated” family classification method and the “family trained” network respectively as shown in equation 8.2. For the example in figure 8.5, the “confidence level” was highest for the “family trained” network denoted as F.F. in the rightmost bar graph (please refer to appendix A.8.2 figure A.12 for details on algorithm). The “estimated” family had *low* confidence that the pattern belonged to family 1 (strings) whereas the “family trained” network had *high* confidence that it belonged to family 2 (woodwinds). Narrowing down the search for individual instruments by looking only in the woodwinds family group, a correction in the individual instrument classification was made (instrument # 5 clarinet – confusion between #5 clarinet and #2 violin in this example).

$$Confidence_{family} = y_{\max} - (y_k + y_l) \quad (8.2)$$

8.4.1 Instrument Family Recognition

For the family recognition task I have used three families – strings, woodwinds, and brasses. The network was trained with 3 possible target values where each output corresponded to a particular family as shown in table 8.8.

Target vector (output)	Family
001	Strings
010	Woodwinds
100	Brasses

Table 8.8 Target output patterns for instrument family

The 9 member feature set that resulted in best performance for family classification was *shimmer, jitter, spectral spread, spectral centroid, harmonic slope, noise content, harmonic expansion/contraction, temporal centroid, and zero crossing rates*. Table 8.9 summarizes the best results for family classification.

	RBFN		EBFN	
	Normal	**NCC**	**Normal**	**NCC**
Correct classification (%)	78	97	81	92
Number of features	9	9	9	9
Number of centroids	50	132	50	124
Number of epochs	2000	1500	2000	1500
Cross-validation (%)	63	68	65	69

Table 8.9 Best training results for families without cross-validation

	RBFN		EBFN	
	Normal + CL	**NCC + CL**	**Normal + CL**	**NCC + CL**
Correct classification (%)	73	88	85	85
Number of features	9	9	9	9
Number of centroids	45	54	40	57
Number of epochs	2000	1600	9000	7000

Table 8.10 Best results for family recognition with cross-validation

As we see in table 8.9 and figure 8.2 system performance without cross-validation tends to be higher, whereas performances with inclusion of cross-validation about 10% lower. This is likely a case of pattern over-fitting most noticeable with the all inclusive NCC (Nearest Centroid Error Clustering) algorithm. That is, NCCs that include both multi-member *and* single member centroids. By enabling only multi-member centroids to be included in the fine-tuning process, better results were obtained for cross-validation – this methodology was used for all cross-validation tests and network training. The best performance for family recognition was approximately 88% for RBFNs after NCC and CL (confidence level) correction.

One interesting finding was that EBFN performances were a little lower than RBFNs. Although EBFNs have not been exhaustively trained and tested as the computation load and stability was a major issue, the current results do suggest for complex pattern spaces RBFNs have more robust classification results.

Figure 8.6 shows the confusion matrix run on *all* the data samples using the network parameters (number of centroids, means, weights, features) learned from the 88% (family) cross-validation network to get a better overview of the network's errors (cross-validation samples + training samples). The confusion matrix was constructed using the actual output values of the network which are not 0s or 1s (binary) but are continuous values that have been normalized between 0 and 1. The confusion matrix is divided into three groups. The first group of three bars represents confusion of the system when presented with stringed instrument sounds. From the graphs we can see that the least confusion for string instrument classification was with brasses and a little confusion existed in the case of the network misclassifying some instruments as woodwinds. The 2nd group shows the confusion of the system when presented with woodwind instruments samples. Most of the confusion was with string instruments. The last group reflects the confusion of the system when reacting to brass instruments. In the brass section, we can note that the network had noticeable confusion with string instruments but also substantial errors with woodwind instruments.

Figure 8.7 shows the binary version of the confusion matrix ("correct or not – correct"). The rows denote the presentation of instruments and columns the reaction to the presented instruments by the recognition system. The diagonal matrix elements are the number of correct recognition, the elements in the other locations the misclassified numbers. The last column shows the total correct family classification (%).

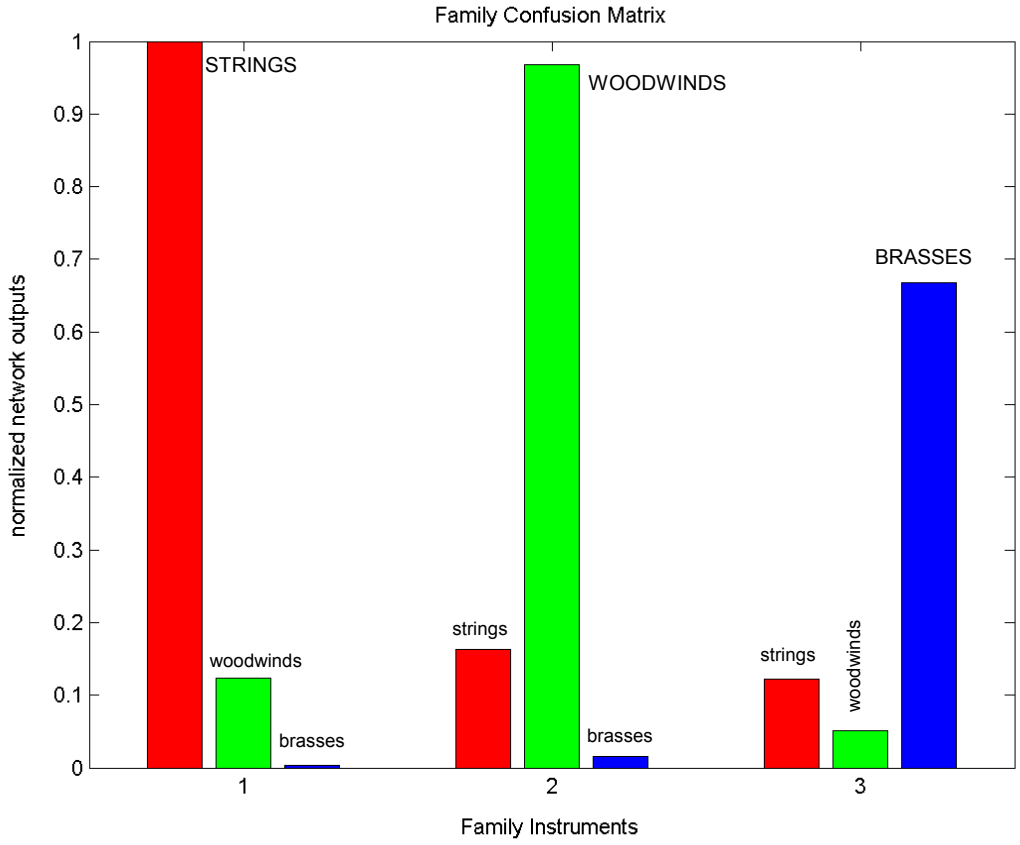


Figure 8.6 Confusion matrix for instrument family classification

	Strgs.	Wwinds.	brasses	%
Strgs.	**278**	10	4	***95***
Wwinds.	24	**261**	4	***90***
brasses	22	24	**202**	***81***

Figure 8.7 Confusion matrix for instrument family (binary format)

As we can see, the best classification results were for the strings in the mid 90s, woodwinds in lower 90s, and brasses in the lower 80s. We will see in the next section that one instrument, namely the French horn was the main cause for performance degradation for family classification and also individual instrument recognition.

8.4.2 Individual Instrument Recognition

In the case of individual instrument recognition, 12 instruments were used as listed in table 8.11. Hence, the target vector had 12 outputs mapped to each

instrument type opposed to 3 for family classification.

Target vector (output)	Instruments
000000000001	Electric bass
000000000010	Violin
000000000100	Cello
000000001000	Viola
000000010000	Clarinet
000000100000	Flute
000001000000	Oboe
000010000000	Bassoon
000100000000	French horn
001000000000	Trumpet
010000000000	Trombone
100000000000	Tuba

Table 8.11 output patterns for individual instruments

The 8 member feature set that rendered best system performance was *jitter, spectral spread, spectral centroid, harmonic slope, LPC noise content, spectral flux, temporal centroid, and zero crossing rates*. Unlike salient family features shimmer, harmonic expansion/compression was absent from the best feature set. Furthermore, spectral flux which was absent in the family feature set was present in the individual feature set. Tables 8.12 and 8.13 summarize the best results for individual instrument classification.

	RBFN		EBFN	
	Normal	**NCC**	**Normal**	**NCC**
Correct classification (%)	70	95	78	97
Number of features	8	8	8	8
Number of centroids	45	214	35	199
Number of epochs	3000	2000	3000	2000
Cross-validation (%)	48	51	52	53

Table 8.12 Training results for individual instruments without cross-validation

	RBFN		EBFN	
	Normal + CL	**NCC + CL**	**Normal + CL**	**NCC + CL**
Correct classification (%)	57	71	63	67
Number of features	8	8	10	8
Number of centroids	45	59	38	74
Number of epochs	10000	8000	10000	8000

Table 8.13 Results for individual instrument recognition with cross-validation

As in family classification, training without cross-validation resulted in superior system performance. However, with cross-validation the best performance for RBFNs was approximately 71% and 67% for EBFNs.

The confusion matrix in figure 8.8 was also computed with all patterns using the network parameters that yielded the best cross-validation results (from the 20% cross-validation pattern set). As in the family confusion matrix, the individual graphs represent the 12 instruments in each bar graph group computed via actual network non-binary outputs. The upper case letters denote the correct output and the lower case the instrument that caused greatest confusion during classification. Table 8.14 lists the greatest confusion that occurred for each instrument.

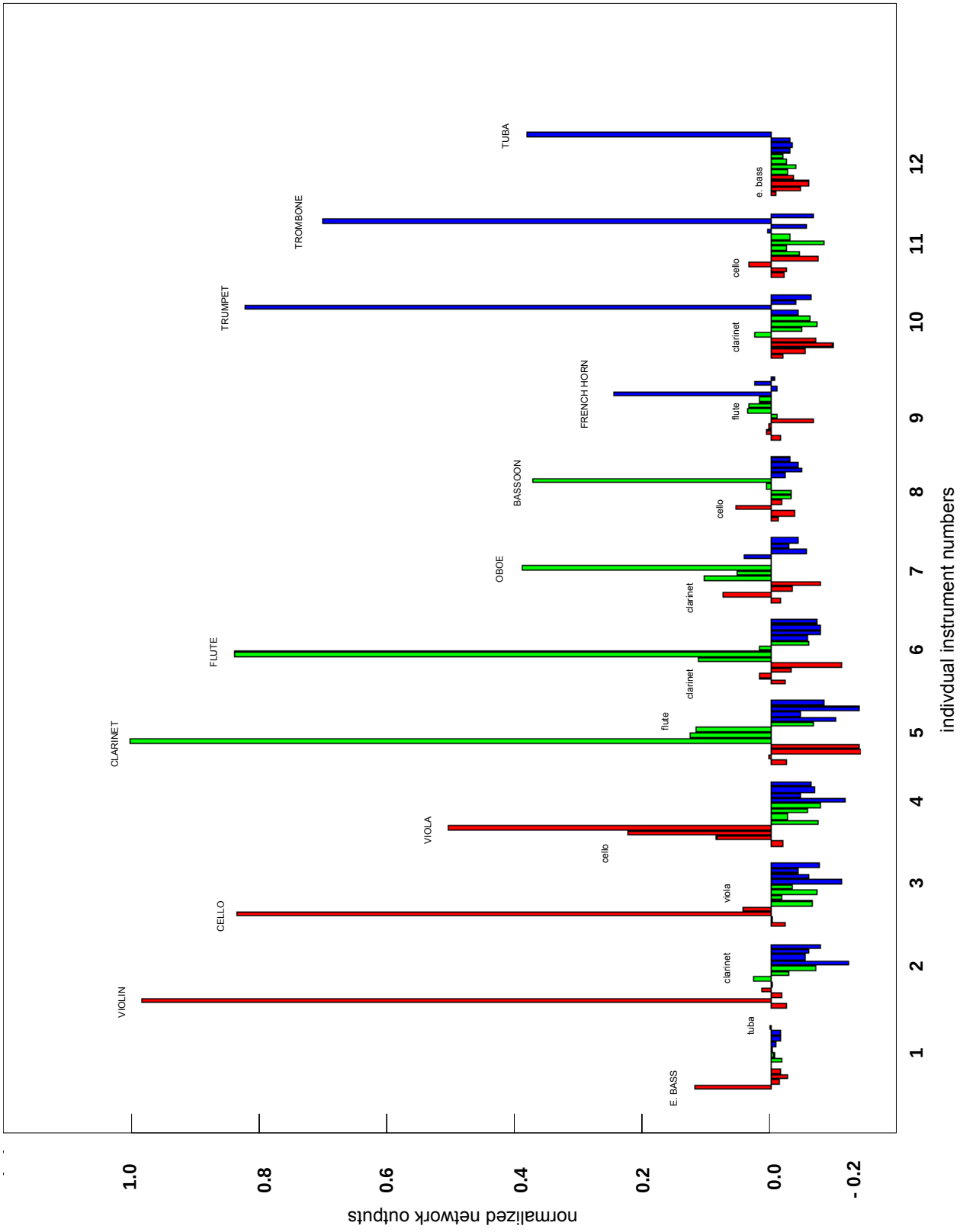


Figure 8.8 Confusion matrix for individual instrument classification

Instrument presented to NN	NN confusing it mostly with
1.Electric bass	12.tuba
2.Violin	5.Bb clarinet
3.Cello	4.Viola
4.Viola	3.Cello
5.Bb clarinet	6.Flute
6.Flute	5.Bb clarinet
7.Oboe	5.Bb clarinet
8.Bassoon	3.Cello
9.French horn	6.Flute
10.Trumpet	5.Bb clarinet
11.Trombone	3.Cello
12.Tuba	1.Electric bass

Table 8.14 Greatest confusion between presentation and reaction to patterns

The confusion for cello was viola; for viola cello; clarinet the flute; flute the clarinet; and oboe the clarinet. The viola and clarinet especially showed characteristics that suggest confusion within its own family group where the viola was frequently confused by the cello and violin. The clarinet's confusion was mostly with flutes and oboes. Interestingly for brasses such as the trumpet, the network confused it mostly with the clarinet. In a number of cases the brass family confusion was not within its own family group but with instruments in other instrument families. Table 8.15 lists the least confusion (most "distant") that occurred for each instrument.

Instrument presented to NN	NN confusing it least with
1.Electric bass	3.Cello
2.Violin	9.French horn
3.Cello	9.French horn
4.Viola	9.French horn
5.Bb clarinet	4.Viola
6.Flute	4.Viola
7.Oboe	4.Viola
8.Bassoon	10.Trumpet
9.French horn	4.Viola
10.Trumpet	3.Cello
11.Trombone	7.Oboe
12.Tuba	3.Cello

Table 8.15 Least confusion between presentation and reaction to patterns

Figure 8.9 shows the "binary" version of the "soft" confusion matrix (figure 8.7). As before, the rows denote the presentation of instruments and columns the reaction of the system to the presented instruments. The diagonal matrix elements are the number of correct classification, the elements in the other locations the incorrect number of instruments classified. The last column shows the total number of correct classification percentage for each individual instrument. What we can immediately see from the "binary" confusion matrix is that French horn classification is the poorest (10^{th} row) at 32%. The best results are for the tuba (last row) at 91% correct classification (most of its errors are within its own family) and the electric bass (1^{st} row). However, there were only 10 samples for the electric bass and hence, the results for the electric bass are inconclusive. Interestingly, for the brass instruments more than the usual cross-family error observations can be made (excluding the tuba). The following figures (8.10~ 8.21) show detailed "binary" confusion matrices breaking-down the

error occurrences for each instrument type according to dynamics and performance techniques.

	eb.	vln.	vcl.	vla.	clar.	flut.	obo	bsn.	hrn.	trp.	trb.	tub.	%
eb.	**9**	.	.	.	1	.	.	.	.	.	.	.	*90*
vln.	.	**83**	4	9	4	3	1	.	.	1	.	.	*79*
vcl.	1	8	**75**	10	1	.	2	.	.	1	4	.	*68*
vla.	.	14	16	**44**	.	.	.	1	.	.	.	.	*59*
clar.	.	4	1	.	**82**	11	1	.	.	1	.	.	*82*
flut.	.	7	3	2	9	**73**	4	.	1	.	.	.	*74*
obo	.	4	.	.	9	4	**38**	.	.	.	.	.	*69*
bsn.	.	.	3	.	1	1	.	**29**	.	.	1	.	*85*
hrn.	1	3	5	.	1	6	10	3	**18**	2	2	5	*32*
trp.	.	1	3	4	2	1	.	.	.	**67**	.	.	*86*
trb.	.	4	5	1	1	5	1	1	4	7	**53**	.	*65*
tub.	.	.	.	.	.	.	.	.	.	.	3	**29**	*91*

Figure 8.9 Confusion matrix for individual instruments (binary format)

	eb.	vln.	vcl.	vla.	clar.	flut.	obo	bsn.	hrn.	trp.	trb.	tub.	#s	%
Finger Plucked	.	.	.	.	1	.	.	.	.	.	.	.	**9/ 10**	**90**

Figure 8.10 Electric bass detailed confusion matrix

	eb.	vln.	vcl.	vla.	clar.	flut.	obo	bsn.	hrn.	trp.	trb.	tub.	#s	%
Long piano	.	.	.	3	.	.	1	.	.	.	.	.	**11/ 15**	**73**
Pizzicato	.	.	.	.	.	.	.	.	.	.	.	.	**19/ 19**	**100**
Short detaché	.	.	.	.	.	.	.	.	.	.	.	.	**15/ 15**	**100**
Con sordino	.	.	5	.	.	1	.	.	.	.	.	.	**9/ 15**	**60**
Spiccato	.	.	.	.	4	2	.	.	.	1	.	.	**23/ 30**	**77**
Con Espre-ssivo	.	.	1	5	.	.	.	.	.	.	.	.	**5/ 11**	**45**

Figure 8.11 Violins technique detailed confusion matrix

	eb.	vln.	vcl.	vla.	clar.	flut.	obo	bsn.	hrn.	trp.	trb.	tub.	#s	%
Long fortissimo	.	5	.	3.	.	.	2	.	.	.	.	.	**35/45**	**78**
Long piano	1	.	.	2	1	.	.	.	.	.	.	.	**3/7**	**43**
Pizzicato	.	.	.	2	.	.	.	.	.	.	.	.	**22/24**	**92**
Con sordino	.	1	.	3	.	.	.	.	.	.	.	.	**12/16**	**75**
Staccato	.	2	.	.	.	.	.	.	.	1	4	.	**4/11**	**36**

Figure 8.12 Cello technique detailed confusion matrix

	eb.	vln.	vcl.	vla.	clar.	flut.	obo	bsn.	hrn.	trp.	trb.	tub.	#s	%
Long piano	.	5	7	.	.	.	.	.	.	.	.	.	**29/41**	**70**
Pizzicato	.	3	6	.	.	.	.	1	.	.	.	.	**11/21**	**52**
Con sordino	.	6	3	.	.	.	.	.	.	.	.	.	**4/13**	**31**

Figure 8.13 Viola technique detailed confusion matrix

	cb.	vln.	vcl.	vla.	clar.	flut.	obo	bsn.	hrn.	trp.	trb.	tub.	#s	%
Fortissi-mo	.	.	.	.	.	.	.	.	.	.	.	.	**11/ 11**	**100**
Long forte	.	.	1	.	.	3	1	.	.	1	.	.	**7/ 13**	**54**
Long piano	.	.	.	.	.	1	.	.	.	.	.	.	**11/ 12**	**92**
Mezzo-forte	.	.	.	.	.	1	.	.	.	.	.	.	**20/ 21**	**95**
Pianissi-mo	.	.	.	.	.	5	.	.	.	.	.	.	**10/ 15**	**67**
Staccato	.	4	.	.	.	1	.	.	.	.	.	.	**23/ 28**	**82**

Figure 8.14 Clarinet technique detailed confusion matrix

	cb.	vln.	vcl.	vla.	clar.	flut.	obo	bsn.	hrn.	trp.	trb.	tub.	#s	%
Long forte	.	5	2	1	.	.	2	.	.	.	.	.	**3/ 13**	**23**
Long piano	.	2	.	.	3	.	2	.	1	.	.	.	**4/ 12**	**33**
No vibrato (*ff*)	.	.	.	.	.	.	.	.	.	.	.	.	**13/ 13**	**100**
No vibrato (*pp*)	.	.	.	.	1	.	.	.	.	.	.	.	**12/ 13**	**92**
Staccato	.	.	.	1	3	.	.	.	.	.	.	.	**19/ 23**	**83**
With vibrato (*ff*)	.	.	.	.	.	.	.	.	.	.	.	.	**13/ 13**	**100**
With vibrato (*pp*)	.	.	1	.	2	.	.	.	.	.	.	.	**9/ 12**	**75**
Long forte	.	.	.	.	.	.	.	.	.	.	.	.	**13/ 13**	**100**

Figure 8.15 Flute technique detailed confusion matrix

	cb.	vln.	vcl.	vla.	clar.	flut.	obo	bsn.	hrn.	trp.	trb.	tub.	#s	%
Long forte	.	.	.	.	1	.	.	.	.	.	.	.	**10/ 11**	**91**
Long piano	.	.	.	.	1	.	2	.	.	.	.	.	**8/ 11**	**73**
Staccato	.	1	.	.	4	1	.	.	.	.	.	.	**16/ 22**	**73**
With vibrato	.	3	.	.	2	1	.	.	.	.	.	.	**5/ 11**	**45**

Figure 8.16 Oboe technique detailed confusion matrix

	cb.	vln.	vcl.	vla.	clar.	flut.	obo	bsn.	hrn.	trp.	trb.	tub.	#s	%
Long	.	.	.	.	1	1	.	.	.	.	.	.	**11/ 13**	**85**
Staccato	.	.	3	.	.	.	.	.	.	.	1	.	**18/ 22**	**82**

Figure 8.17 Bassoon technique detailed confusion matrix

	eb.	vln.	vcl.	vla.	clar.	flut.	obo	bsn.	hrn.	trp.	trb.	tub.	#s	%
Fortissi-mo	.	.	.	.	.	2	.	.	.	.	.	.	**0/ 2**	**0**
Long forte	.	.	3	.	.	3	6	1	.	.	.	.	**0/ 13**	**0**
Long piano	.	.	.	.	1	.	1	2	.	.	.	5	**4/ 13**	**31**
Pianissi-mo	1	.	.	.	.	1	.	.	.	.	.	.	**0/ 2**	**0**
Staccato	.	3	2	.	.	.	3	.	.	1	1	.	**14/ 26**	**54**

Figure 8.18 French horn technique detailed confusion matrix

	eb.	vln.	vcl.	vla.	clar.	flut.	obo	bsn.	hrn.	trp.	trb.	tub.	#s	%
Long forte	.	.	.	3	.	.	.	.	.	.	.	.	**8/ 11**	**72**
Long piano	.	1	3	1	.	1	.	.	.	.	.	.	**5/ 11**	**45**
Con sordino	.	.	.	.	.	.	.	.	.	.	.	.	**8/ 8**	**100**
Staccato (normal)	.	.	.	.	2	.	.	.	.	.	.	.	**21/ 33**	**64**
Staccato (sordino)	.	.	.	.	.	.	.	.	.	.	.	.	**15/ 15**	**100**

Figure 8.19 Trumpet technique detailed confusion matrix

	eb.	vln.	vcl.	vla.	clar.	flut.	obo	bsn.	hrn.	trp.	trb.	tub.	#s	%
Long forte	.	4	1	1	.	1	.	.	.	6	.	.	**1/ 14**	**1**
Long piano	.	.	3	.	1	4	1	1	1	.	.	.	**2/ 13**	**15**
Staccato (piano)	.	.	1	.					.	.	.	.	**26/ 27**	**96**
Staccato (forte)	.	.	1	.	.	.	.	.	3	1	.	.	**23/ 28**	82

Figure 8.20 Trombone technique detailed confusion matrix

	eb.	vln.	vcl.	vla.	clar.	flut.	obo	bsn.	hrn.	trp.	trb.	tub.	#s	%
Long forte	.	.	.	.	.	.	.	.	.	.	.	.	**13/ 13**	**100**
Long piano	.	.	.	.	.	.	.	.	.	.	.	.	**8/ 8**	**100**
Staccato	.	.	.	.	.	.	.	.	.	3	.	.	**8/ 11**	**73**

Figure 8.21 Tuba technique detailed confusion matrix

8.4.3 Discussion on Results

The presented materials in this dissertation demonstrate that with appropriately trained RBF/EBF neural networks via salient features it is possible to design an ANN system to automatically recognize musical instrument sounds. However, the results should be taken with a grain of salt, as the number of *different* examples (from different sound libraries) for each instrument was limited in breath. The majority (86%) of the samples came from a single library (Peter Siedlaczek), although other samples from personal collections and from the Internet were used in training and evaluating the system (clarinets, flutes, and electric bass – 14%). Furthermore, most samples were recorded in clean and clear recording studio environments. Although the robustness of the system has not been tested with samples that were subjected to noise or samples recorded in very different acoustic environments with addition of reverb for example, it is not unlikely that the network may not perform as well in those situations.

It is evident from the tests that the system recognizes instrument families with fewer errors than individual instruments, which is congruent with human performance tendencies as discussed in section 5.1. This is hardly an unexpected discovery, but nevertheless a confirmation that the system is not generally behaving in an unpredictable manner.

The best performance for family instruments was approximately 88 percent using RBF networks with feature set shimmer, jitter, spectral spread, spectral centroid,

harmonic slope, LPC noise content, harmonic expansion/contraction, temporal centroid, and zero crossing rates. EBFNs did not do quite as well, but as mentioned before EBFNs were not exhaustively tested due to time constraints imposed by the computational load and instability of EBF networks (it took more than a week with a 3.2 GHz Intel machine run on Windows XP Professional and Matlab 6.5 to determine the "best" feature sets for the individual instruments at 10,000 epochs and 7 iterations). The majority of the instability for the EBFNs had to do with singularity or near-singularity issues discussed in chapter 7. With careful control of centroid widths perhaps the more flexible EBFNs could be trained to produce better results.

It was discovered that the networks can be *actively* trained to classify *training* samples in the upper 90 percentile, but this seemed to cause over-fitting and result in less-than acceptable cross-validation results both for family and individual instrument classification. Also, as the experimental results show, it was possible to quickly and robustly increase the network's performance using the Nearest Centroid Error Clustering (NCC) method developed in this thesis starting with a relatively low number of centroids and jumping to a higher centroid configuration with the insertion of "children centroids" in error prone areas. Simply increasing the number of centroids without NCC generally did not result in acceptable results and in most cases introduced instability making the higher order networks very difficult and sometimes impossible to use. For individual instruments the best performance (with cross-validation) was 71% using RBFNs.

Again, as in family classification, EBFNs had a slightly lower success rate. The inferiority of the EBF networks ather a surprise as initial tests, albeit on synthesized two-dimensional pattern spaces resulted in better performance.

Some features such as the spectral centroid, spectral spread, temporal centroid, zero crossing rates, and attack time which have been used by a number of other researchers have been cross-validated in this thesis as important features. Additionally, new features such as harmonic slope, LPC-based noise content analysis, and harmonic expansion/contraction have also shown to increase classification performance of the neural network.

Looking back at the confusion matrix for individual instruments (figure 8.9 it is apparent that the system has difficulty in classifying French horn timbres. However, interestingly enough classification for the other brass instruments was on the average better than the other families – if we take out the worst performing instrument from each family (strings – viola, woodwinds – oboe, brasses – French horn), the average correct classification for each family group would be 79%, 80%, and 81% respectively (increase from 74%, 77%, 69%). Also, except for the tuba which has all its errors within its family (also highest individual instrument classification performance at 91%), the other brass instruments, especially the trumpet and trombone have their errors outside their family in both the strings and woodwinds groups. Although the first reaction may be that this result is potentially a “serious network classification problem” since it

seemingly can't even classify an instrument into the correct family, it may not necessarily be an undesirable result. For the French horn for example, most of the confusion occurs with the oboe (10 patterns, especially with long forte notes – see figure 8.18), flute (6), and cello (5 samples). Although further research needs to be conducted, it is very possible that tweaking misclassifications of instruments with "cross-family" error is fundamentally easier to do than "within-family" misclassifications – one simple is argument is that family recognition is easier for both machines and humans. However, it remains to be seen if additional features or tweaking of existing features to improve classification will indeed improve French horn errors.

Additional observations can be made from confusion matrices in figures 8.10 ~ 8.21 which illustrate classification errors based on dynamics, articulations, and techniques for each individual instrument. Generally, it seems that classification is dependent on dynamics and techniques used. One interesting result in this study shows that in a number of cases, the system's performance was better for samples that used some variation of "short-note techniques." This is summarized in table 8.16. The results are rather surprising as I initially expected more errors to occur with pizzicato and staccato techniques for example due to *transient* characteristics of such sounds. The findings also implicate that the system may possibly work well with percussive instruments especially those percussive instruments that are pitched, this however remains to be seen.

It is not that unexpected then, that in some cases long notes tend to cause problems for the system. As we can see in table 8.17, a number of instruments regardless of family membership display misclassification characteristics when presented with patterns that are long piano and long forte techniques – this is true for strings, woodwinds, and brass instruments.

Instrument	Technique, dynamics
Violins	Pizzicato: 100% Short - detaché: 100%
Cellos	Pizzicato: 92%
Clarinets	Staccato: 82%
Flutes	Staccato: 83%
Oboes	Staccato: 73%
Bassoons	Staccato: 82%
French horns	Staccato: 54%
Trumpets	Staccato (muted): 100%
Trombones	Staccato (piano): 96%

Table 8.16 High system performance with specific techniques and dynamics

Instrument	Technique, dynamics
Violins	Con espressivo: 45% Con sordino: 60%
Cellos	Long piano notes: 43%
Violas	Con sordino: 31%
Clarinets	Long forte notes: 54%
Flutes	Long forte notes: 23% Long piano notes: 33%
Oboes	Vibrato: 45%
French horns	Long forte/pianissimo notes: 0%
Trumpets	Long piano notes: 45%
Trombones	Long forte notes: 1% Long piano notes: 15%

Table 8.17 Poor system performance with specific techniques and dynamics

Also, *con sordino and con espressivo* techniques seem to confuse the system in its classification of stringed instruments. However, not all instruments adhered to the aforementioned trends and some deviances can be seen in the confusion matrices. Viola for example did not perform as well as the other stringed instruments with pizzicato techniques. Another exception to the lower performance of long-note is the clarinet with 92% accuracy for long forte notes and fortissimo notes at 100%.

Comparing the developed neural network system with other artificial systems as summarized in table 5.4, it is possible to note some similarities and differences in performance as well as testing environment. For k-NN based models, Fujinaga, Martin, Ronen & Klapuri (see table 5.4) reported approximately 61% ~ 75% accuracy rates for individual instruments which are close to the rates obtained with this neural network model. Also, the number of samples that were used (1023 to 1498) is comparable to what was used in this dissertation if we take into consideration the number of classes that were applied. For other neural networks systems such as the one used by Kaminskyj (Herrera-Boyer et. al 2000) a high 90-percentile performance was reported. However, the number of instruments (4) and number of samples (240) employed to evaluate the neural network seem to be less-than-ideal to confidently make an assessment. The same is true for Kostek who has had excellent results with 94~100% accuracy using 40 samples classified into 10 classes (on the average 4 samples per class). Kostek also reported 97% for correctly classifying four instruments (bass

trombone, trombone, English horn, and contra bassoon). However, the pitch information was provided to the system and training patterns and cross-validation patterns came from the same stereo audio file – one channel for training and the other channel for cross-validation (Eronen 2001).

Comparing the system with the human counterpart, it can be said that the general difficulty or ease for identifying instrument families and individual instruments seem to be the same – families are easier to recognize whereas individual instruments less so. As summarized in table 5.1, human recognition performance ranges from 41% ~ 90% for individual instrument recognition, and upper 90 percentile for family recognition. Again, some of these numbers tend to be misleading as the number of instrument types and the numbers of examples for each instrument vary greatly from report to report. For example, a 90% correct classification of “individual instruments” by humans was conduced with only 3 instruments (flute, clarinet, and oboe), another example at 92% only included the oboe and saxophone (albeit according to Brown (Brown 2001) woodwind instruments are difficult to distinguish from each other due to their similar characteristics in attack, modes of excitation, and frequency overlap – which is actually true for other instrument families also). In other tests such as the one conducted by Martin (Martin 1999) a 46% accuracy was reported for isolated tone tests using a similar testing environment as the one applied in this research (however Martin used 27 instruments not 12 as is the case for this dissertation).

Some of the problems that may have attributed to the system's performance are as follows.

1. The types of actual pattern examples (non-synthesized) used for the training and testing.

   The sample-base included a wide range of dynamics (pianissimo, piano, mezzo-forte, forte, and fortissimo), performance techniques (long, short, staccato, pizzicato, con sordino, detaché, vibrato, non-vibrato, and espressivo), and a variety of pitches (see appendix A.9). Timbral qualities change considerably with change in pitch, dynamics, different articulations, as well as performance techniques such as pizzicatos and detachés. Hence, without sufficient number of examples it may be difficult to train the network to cover a larger sample space and perform adequately in cross-validation situations. For example, the lack of number of samples was evident for some instruments such as the tuba (32 samples), bassoon (35 samples), and electric bass guitar (10 samples). Interestingly for the strings and woodwinds families, confusion in classification generally occurred within one's own class. For brass instruments the confusion seemed to jump to other family classes – when presented with trombone samples confusion occurred mostly with the cello and flute. The acoustical implications and feature similarities have not been investigated during the writing of this dissertation, but such a prospect

would be a worthwhile endeavor to further gain insights on features and timbral structures.

2. Feature extraction accuracy

   It may be that some of the feature extraction algorithms are not robust enough to help output consistent values. This may be the case for inharmonicity computation as it was not one of the salient features that increased classification performance both for family and individual instrument classification. However, other features using harmonic structure information such as the harmonic slope and harmonic contraction/expansion did help in obtaining the best classification results. The poorest performing instrument was the French horn, which may suggest that the current set of features is inadequate for robust French horn classification. Figure 8.9 suggests that the errors for French horn (68% error) occur mostly between woodwinds – 9 errors in the string family, 20 in the woodwind family, and 9 within its own family.

3. Further testing of EBFNs

   EBFNs have not been exhaustively tested due to time constraints and instability. Further testing EBFNs and finding the optimal network training parameters may possibly improve overlall performance as

EBFNs have shown to be more flexible and robust in preliminary classification experiments conducted in this dissertation.

What I have tried to accomplish in the technical part of the dissertation was to come away having a better understanding about the structure of musical instrument sounds, build an artificial system for recognizing musical instrument timbre, and to document as much as possible my own research findings and existing research results.

In summary, an outline of the history of timbre and technical research have been presented in the beginning chapters while later chapters focused on detailed technical issues regarding automatic timbre recognition. A number of new features and pattern classification related algorithms were developed, implemented, enhanced, and utilized in this dissertation. Some of the "new" and enhanced features included noise content analysis using LPC, harmonic expansion/contraction, enhanced attack-time computation, and a harmonic tracking algorithm among others. For pattern classification, notable contributions were the Nearest Centroid Error Clustering (NCC) algorithm which increased the performance of the network substantially by automatically inserting smaller and more refined centroids in areas prone to (mostly) boundary related errors, and also the "confidence levels" (CL) algorithm for correcting errors in classification. The NCC algorithm in particular worked robustly in increasing the number of centroids using error feedback, and as a result increased the network's performance. With the absence of the NCC stage difficulty was experienced in increasing the number of centroids (and hence classification performance) with

the standard network learning models and clustering algorithms by simple “lazy-learning” initialization processes. Given the fact that RBFN/EBFNs have not been used for automatic timbre recognition (during the writing of this essay), this dissertation may serve as a starting point for researchers interested in using such networks for their own research.

One of the major difficulties in using neural networks was by far the amount of time it took to train the networks. The training time increased exponentially with the number of input dimensions, output dimensions, and number of centroids. Using Matlab (although all time-critical modules had been optimized using matrix manipulations to minimize computation load) did not help in work efficiency, especially during the later system verification test stages – “number crunching and waiting for results” stages. In hindsight testing the system could have been more efficient if faster compiled languages such as C/C++ or Java were used during the later stages – but that would have also entailed having to port code to another environment which introduces a whole different set of uncertainties.

Future work in improving classification performance of this system for automatic musical instrument timbre recognition is summarized in table 8.18.

Improvement suggestions	Description
More pattern examples	Other sample libraries (McGill Master Sample Libraries and other sources) Different recording conditions (different space, noisy spaces)
"More" and "better" features	Finding new features Fine tuning existing features More specialized features (for French horn especially) Separate analysis of attack/steady-state
Applying a hierarchical model	Training sub-networks Sub-networks specialized to instrument families
Improving initialization procedure	Applying clustering algorithms such SOM (Self-Organizing Maps)
Multiple "confidence level" comparisons	Multiple trained networks for family classification
Higher level intervention	Pitch information Performance technique information

Table 8.18 Possible improvements for classification system.

Using additional patterns from other sources may not necessarily improve the system in the short run, but will without a doubt test the system's robustness and help fine-tune feature extraction algorithms as well as the neural network classification model. Using a hierarchical approach whereby increasing the performance of family classification to a maximum and training networks for each family type separately may be another possible area for improvement of this system. The drawback of course is that the hierarchical system would very much depend on the robustness of the system in classifying instrument families or some other top-level tree-based branching structure. With the current system using a family-based hierarchical structure, the best-case scenario would render 88% *maximum* performance for individual instrument classification. Another

fertile area for improvement is the initialization process during network training. It cannot be stressed enough that the initialization stage is one of the most important parts in any pattern classification system. Other clustering algorithms such as Self-Organizing Feature Maps (SOFM) may be an option for the initialization process. Also, taking the idea of "confidence level" a step further it may be worthwhile trying out training a number of different networks to simultaneously classify patterns into families and coming to a consensus on a final family membership utilizing confidence level votes derived from all the networks. Other possibilities may be another high-level "correction" scheme for improvement may be using features to verify and correct a network's classification result. For example, pitch information may be exploited in excluding certain instruments from a possible list of choices, while other features such as vibrato and other features may further help limit the choice of possible candidates.

Other tests that could be conducted are using different samples rates for features extraction and classification. As discussed in section 4.2 sampling rates and spectral decimation do not seem to play a critical role in human speech *recognition* and *identifying* synthesized sounds. It would be interesting to test the system under lower sample rates – 8kHz and 11kHz for example, and observe the results. Perhaps lower sampling rates will not degrade the performance by much, which would further suggest our hearing system's flexibility in sound object identification under poor sound conditions.

Future work pertinent to this dissertation includes an immediate goal in expanding the number of instruments, number of examples, and inclusion of percussive instruments. Preliminary findings discussed in chapter 8 suggest that there is a good possibility of the system performing well with transient sounds. Also, other pertinent future work includes a short-term goal – uploading the source programs and essay on the Internet as open source documentation (please check http:/music.princeton.edu/~park in the near future); a mid-term project that entails porting the Matlab code into a standalone Java application (part of the Matlab code has already been ported), and a long-term project to design a compositional Java application for manipulating and editing timbral qualities of sound objects. For example with this new Java application it would be possible to extract the noise-content of a signal using LPC, change the sound object's noise quality and mix it back to the signal. Another example may be changing the harmonic compression and contraction characteristics of harmonic sounds using phase vocoding techniques.

The compositional portion of the dissertation includes four pieces, *"A d'Ess Are," "Aboji," "48 13 N, 16 20 O,"* and *"pH-SQ."* The following sections will give brief descriptions of the pieces.

10.1 "A d'Ess Are"

10.1.1 Concepts, Structures, and Form

"*A d'Ess Are*" was inspired by the theme of timbre in musical composition. My initial starting point was using timbral metaphors and "timbral features" for orchestration, arrangement, and composition of the piece. To force myself to compose in a constrained environment, I loosely utilized *Fibonacci* numbers to help me construct the time-architecture, occurrence of events, and pitch selection process. The title of the piece which also in part depicts the form of the piece is pronounced ADSR (attack decay sustain release). Hence, the piece is more or less divided into four sections A, D, S, R.

The time-structure of the piece adheres to the Fibonacci number system in the following way.

Section	**A**	**D**	**S**	**R**
Time (sec)	1	89	144	377
Measure	1	41	64	126

Table 10.1 Fibonacci based time structure

I have tried to orchestrate each of these sections (ADSR) according to the characteristics of those sections present in a musical tone.

As table 10.2 illustrates numerous timbral features and metaphors were used to characterize each section via instrumentation, rhythm, performance techniques, and registers. In other words, during the lifetime of a musical tone the attack is generally characterized by entrance of lower partials first (noisy components in the case of plucked strings), build-up of energy; the decay distinguished by transitory aspects between the attack and steady-state (sustain); the sustain characterized by quasi steady-state aspects and “orderly” behavior of partials; and the release exemplified by loss of energy.

In essence, the Fibonacci series from 1 ~ 377 (1 2 3 5 8 13 21 34 55 89 144 233 377) were used. For the selection of pitches, I have used the Fibonacci series with modulo 12 to wrap it into the 12 tone equal temperament system. Hence, 1 2 3 5 8 13 21 34 55 89 144 233 377 becomes 1 2 3 5 8 1 9 10 11 5 12 5 5, which can be simplified to 1 2 3 5 8 9 10 11 12, and the “anti-Fibonacci” pitches then become 4, 6, 7. With “root note” E the series maps to E F F# G# B C C# D D#. I tried to use combinations of Fibonacci and “anti-Fibonacci” series for the pitches, form, and architecture of the piece not for the sake of using those numbers because they are Fibonacci numbers per se; but rather to limit, restrict, and force myself to work within self-imposed boundaries to help me compose something different – something that I would not normally do otherwise.

Section	Characteristics
Attack	Noise components *Breath noise of instruments* *percussive elements such as bass drum, snare, timpani, pizzicato in strings, staccato notes, scratch tone in string section* Build up in energy *Piano to fortissimo, build up in density of notes and instrumentation* Increase in "high partials" *Lower registers enter and higher register follow, instruments with higher "spectral centroid" enter towards the end of the attack section* Stability *Rhythm starts becoming pattern based towards the end of the attack section.* *"Instability" to "stability"*
Decay	Interfacing section between A and S *Stable "harmonic" characteristics enter.* *Rough "noise-like" residual elements*
Sustain	"Steady state" Characteristics *Harmonic background becomes "stable"* Other Timbral Steady-State Features *Jitter/shimmer characteristics in triangle and piano* *"Amplitude modulation" characteristics through use of harmonic rhythm by cross-fading woodwinds, horns, and trombones* *"High partials" die away – dominance of mid registers*
Release	Loss of Energy *Harmonic background dies away* Dying away of "high-mid partials" *Lower registers die away last*

Table 10.2 Sectional strategies

10.2 “Aboji”

Since coming to Princeton University I have been interested in composing 8-channel tape music. In particular, I have been attracted to tape compositions that are narrative. The usage of narrative strategies in my work started with the tape piece “Omoni.” Omoni was originally a 2-channel tape piece composed at Dartmouth College in 1999, and later rearranged for 8 channel diffusion at Princeton University in the spring of 2001. Omoni was an outcome of trying to compose a piece that would have some sort of relation to everyone who would be listening to it. Thinking for a moment the topic of “mothers” came to mind and that is how the project started. “Aboji” completed in September 2001 is a companion piece to Omoni and was put together by asking interviewees to talk about, comment, on any topic pertaining to “fathers.” (Omoni means mother in Korean, Aboji translates to father). The structure, context, and sonic environment of Omoni and Aboji are both very dependent on the selected interview excerpts where various persons tell their stories about a specific memory or topic on motherhood in Omoni and fatherhood in Aboji.

For Aboji, although the underlying concept of the piece is narrative and the focal point is a presentation of excerpts of interviews, different layers of structure and form also exist which help in connecting and threading together idiosyncratic sections to render a musical composition. For example, the very first sound is a melodic piano line that apparently comes out of nowhere and has seemingly no relation to the piece itself upon first hearing or perhaps even subsequent

listenings. However, that small phrase actually occurs during various points in the piece, sometimes subtly masked and at other times juxtaposed in a particular sonic context. The piano line itself originates from a portion of speech where the rhythmic and melodic features of "… don't pretend you're not with me …" have been extracted and mapped to the piano. At another point in the piece, the sonic ambience from New York City oscillates between the background and foreground, while quotes from old LPs pertinent to the content of the stories fade in and out as well, as the interviewee tells a story about Frank Sinatra, Count Basie, and Meyer Lansky. While numerous stories of a person's father or someone else's father are anecdotally conveyed throughout the piece, the word "father" only surfaces at the very end of the piece.

10.3 "48 13 N, 16 20 O"

In "48 13 N, 16 20 O," I have taken the narrative approach for composition a step further. I have tried to walk the path of the reporter vs. the composer and sought to address issues concerning documentation vs. composition. This project began in the summer of 2002 when I was a guest composer at the Universität für Musik und darstellende Kunst Wien in Vienna, Austria. During my month long stay in Vienna I took my DAT player everywhere I went, recording the city on all levels that I could possibly imagine – from the lower levels of the maze-like subway tracks (U-Bahn), to the upper levels of streetcars, buses, bikes, cars, boats, and on foot in all corners of the city. The city always had a connection to my past as I was born there and lived there "twice" during my childhood adding a

special value to record everyday sounds which were idiosyncratic to that specific location and use those samples as the starting points for composing the piece. After returning to Princeton in the fall of 2002 I began working on the piece and finished it in the middle of 2003.

The compositional strategy of the piece was to start from an abstract soundscape (materials from the latter sections of piece are presented in the opening but masked to a degree of ambiguity to allow the mind to "imagine" things) and move towards a more concrete soundscape eventually ending with the meaning and disclosure of "48 13 N, 16 20 O." During the course of the piece I have tried to focus both on the musical aspects and documentary aspects within a narrative framework. I imagined myself starting at the Beethoven house (one of many!), riding the bus towards the city … getting off and onto a streetcar … entering a Sunday mass at the Stephans Dom … enjoying a hot summer's afternoon of street musicians … finally riding the subway to the Rathaus in the evening to get some Wiener Bier und Wurst at an open-air "concert"… – a compositional documentary focusing on sonic attributes found at longitude and latitude coordinates *"48 13 Nord, 16 20 Ost."*

10.4 "pH-SQ"

pH-SQ written for the Nash Ensemble of London and premiered at Richardson Auditorium at Princeton University April 6, 2003 stands for *p(iano) H(orn)-S(tring)Q(uartet)*. The name of the piece was inspired by Iannis Xenakis'

Concrete PH which is one of the very first electroacoustic music pieces that I heard, although the title is the only relationship and influence of Xenakis' composition on this "acoustic" piece. "pH-SQ" was however somewhat influenced by Bela Bartok's music, especially his 4th string quartet with its dissonance, roughness, rhythmical complexity, and forward moving energy which has been ringing in my ears since first hearing it.

The aspect of friction has in particular been an appealing element for the compositional basis of writing this piece. One specific harmonic friction that I enthusiastically used is the P5/P4 friction. The dissonance that is produced by the juxtaposition of two perfect intervals off by a semi-tone, C-G vs. B-F# for example is omnipresent throughout the piece. Also, the energetic forward charging rhythmical patterns coupled by pizzicato and aggressive piano and horn lines, added to the roughness and harshness in the harmonic, rhythmic, and instrumental material that I was looking for. In order to accentuate high energy moments, which without a doubt exemplify this piece, strategies for releasing the energy and relaxing the momentum of thrust have also been strategically placed in various sections of the piece. Notably the introduction of a "serene" section where the horn plays a solo passage, which in turn gets linked to the piano and then to the string section and finally to the last part of this section where all the instruments come together, foreshadows the return of the more tense and pulsating friction-based materials.

Most sections in the piece have a quasi-tonal center which is clearly audible. Applying the strategy of contrast using tension vs. relaxation, friction vs. fluidity; a "whole tone chord" section was applied to contrast the tonal center paradigm in the latter part of the piece at measure 201. Basically this section comprises of two whole tone chords separated by a semitone, the piano playing one set of the whole tone pitches and the strings playing the other set of whole tone pitches – eventually rendering a feel of the absence of tonal center.

As in "A d'Ess Are," a strategic approach was taken during the initial and intermediary stages of the compositional process. Again, borrowing the Fibonacci series and also the harmonic series as a way to map numbers to pitches, I was able to produce pitches that limited my palette for composition. More specifically, I used the pitches starting on C that rested on the harmonic series up to the 20^{th} harmonic. These were processed through a "Fibonacci filter" resulting in pitches of the 1^{st}, 2^{nd}, 3^{rd}, 5^{th}, 8^{th}, and 13^{th} harmonic; an "anti-Fibonacci filter" resulting in harmonic pitches that were harmonics other than the "Fibonacci harmonics"; and various other processes which colored a particular section in the piece. Again, it was not my intention to utilize a systematic method for generating pitches, rhythms, and structure because there was some hidden and mystical compositional value, rather it was a very practical way of limiting myself to a restricted set of tools with which I was to compose this piece.

Appendix

A.1 The Tone

The term tone has been used in place of sound in the context of timbre experimentation, as it is capable of evoking auditory sensations. It can be further refined in definition to mean sounds that are:

1. Longer than impulses, having a sustain part.
2. Mostly quasi-periodic eliciting a sense of pitch.
3. Produced by synthesized, non-synthesized sounds, and musical instruments including the voice.

Tones are sometimes synthesized and used as stimuli for test subjects. Test tones can be divided into simple tones or pure tones corresponding to sine tones, and complex tones respectively. Complex tones, which contain more than one frequency component (harmonics or strong nodes), can be either harmonic or inharmonic. Harmonic tones pertain to pitched tones and inharmonic test tones usually refer to noisy tones. Timbre experiments in general however, have utilized test tones that are periodic, quasi-periodic, and pure. The choices may have been in part the due to the importance placed on the steady-state portion of a tone, the ease of manipulation of these test tones by the experimenters in trying to isolate a problem using unnatural sine tones.

A.2 Some Nomenclatures

1. Partials: Frequency components that constitute a complex tone, which are not necessarily in integer relationship to each other.
2. Harmonics: Partials that are in integer ratio relationship constituting a harmonic series, i.e. $f_i = i \cdot f_0$ where f_0 is the fundamental frequency denoted as the first harmonic. However, f_0 is not always present in a signal.
3. Overtones: Often used in a musical contexts meaning harmonics that lie above the fundamental frequency. However, this may cause some confusion as the 1st overtone corresponds to the second harmonic. To further confuse things, for a tone having only odd harmonics, the first overtone denotes the 3rd harmonic. It is therefore good practice to use the term harmonics.
4. Centers, centroids, means: These terms in the context of RBFNs/EBFNs refer to the center of a circle/sphere/hypershphere or ellipse/ellipsoid/hyperellipse.
5. Spread, σ, standard deviation: These terms in the context of RBFNs/EBFNs refer to the center of a circle/sphere/hypershphere or ellipse/ellipsoid/hyperellipse.

A.3 Gradient Descent

Gradient descent also known as *steepest descent* is a method for finding local minima in a given function. It guarantees that *at least* a local minima will be

found in any function with a computable derivative. The theory and concept is quite simple to understand if we consider the following scenario:

Let the function in question be called *E(w)*, which in our example could be an error function (or any other function) that is computed comparing a true output to an actual computed output value.

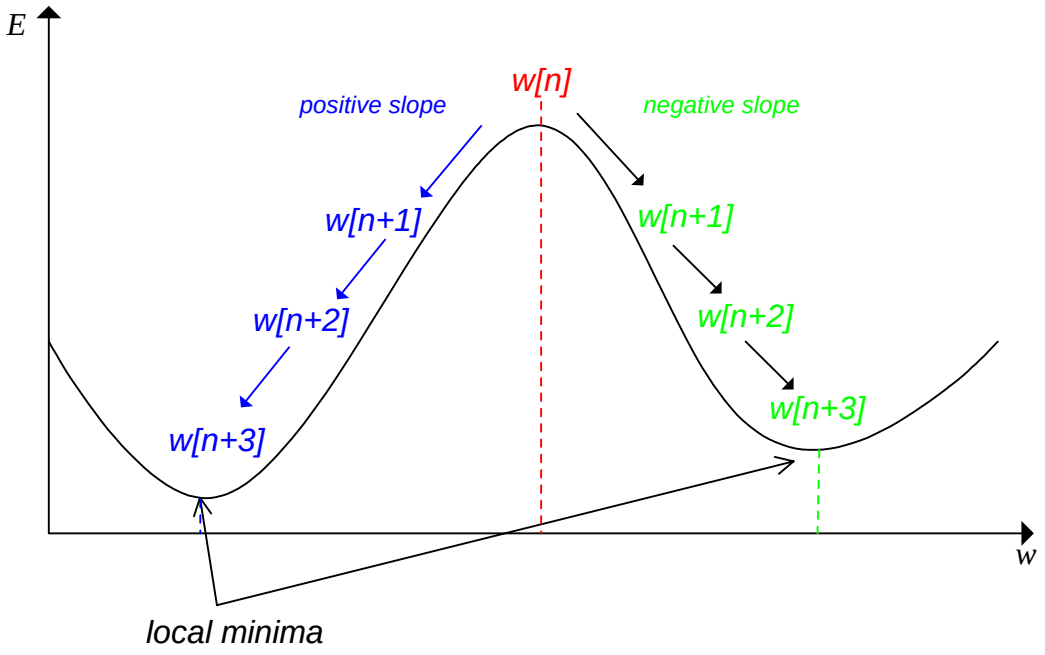


Figure A.1 Gradient descent

We know from section 7.2.3 that the definition of the gradient is:

$$gradient = \frac{\partial E}{\partial w} \quad \text{(A.1)}$$

This basically means that we are computing the slope of function *E(w)* with respect to *w*. In our case function *E* happens to be the error function. We also know from equation 7.10 that the updated weight *w[n]* is computed using the past weight *w[n-1]* and the change in weight Δw, in other words*:*

$$w[n] = w[n-1] + \Delta w \quad \text{(A.2)}$$

$$\Delta w = -\eta \frac{\partial E}{\partial w} \quad \text{(A.3)}$$

Since the change in weight is defined by the change in slope and the scalar η we can rewrite A.2 as:

$$\begin{aligned} w[n] &= w[n-1] + \Delta w \\ &= w[n-1] - \eta \frac{\partial E}{\partial w} \end{aligned} \quad \text{(A.4)}$$

Here η basically dictates the step-size for *w*. Let's say we are starting at some random point *w[n]* as in figure A.1. What happens if the computed slope is negative? It is clear from figure A.1 that we will move towards the right or positive direction of the "*w* axis" until we reach the local minima at around *w[n+3]* (due to the negative sign in equation A.4). Notice that when computing the next *w* at *w[n+4]* the slope will be *positive* which means that *w[n]* will go towards the left (smaller *w*). The same phenomenon happens when we start at *w[n]* and the initial slope is computed as a positive number – due to A.4 we will move left

towards the local minima. Hence, it will be guaranteed that a local minimum will be reached.

A.4 Least Mean Square (LMS) Delta Rule and Gradient Descent

The Least Mean Square (Widrow, Hoff 1960) of a function such as error *E* can be written as:

$$\begin{aligned} E &= \frac{1}{2}(T_p - Y_p)^2 \\ &= \frac{1}{2}(T_p - \sum_{i=1}^{K} w_i x_i^{(p)})^2 \end{aligned} \quad \text{(A.5)}$$

Where *E* is the *Mean Squared Error* (MSE) T_p is the known teacher for pattern *p* and Y_p is the actual output computed from the network. The objective here is to try to minimize *E* in order to find the suitable weights $w_i[n]$. Using gradient descent and *Chain Rule* we have:

$$\begin{aligned} \frac{\partial E}{\partial w_i} &= \frac{\partial E}{\partial Y_p} \cdot \frac{\partial Y_p}{\partial w_i} \\ &= \frac{\partial}{\partial Y}\left[\frac{1}{2}(T_p - Y_p)^2\right] \cdot \frac{\partial}{\partial w_i}\left[\sum_{i=1}^{K} w_i x_i^{(p)}\right] \\ &= \left[2 \cdot \frac{1}{2}(T_p - Y_p) \cdot (-1)\right] \cdot \left[x_i^{(p)}\right] \\ &= -(T_p - Y_p) \cdot x_i^{(p)} \end{aligned} \quad \text{(A.6)}$$

let local error $\delta = (T_p - Y_p)$, *then*

$$\Delta w_i = -\eta \frac{\partial E}{\partial w} = \eta \cdot \delta \cdot x_i^{(p)} \quad \text{(A.7)}$$

and finally we have

$$w_i[n] = w_i[n-1] + \Delta w_i[n] = w_i[n-1] + \eta \cdot \delta \cdot x_i^{(p)} \quad \text{(A.8)}$$

A.5 Derivation of Weight Initialization

The weight initialization as described in section 7.2.3.1 was achieved via k-means. The weight computation is arrived at using the activation outputs, target vector, and total squared error *E* which is computed as the difference between the target vector d and actual output *y*.

$$E = \left(\vec{d} - A\vec{w}\right)^T \left(\vec{d} - A\vec{w}\right)$$

$$= \left(d^T - w^T A^T\right)(d - Aw) = d^T d - w^T A^T d - d^T Aw + w^T A^T Aw \quad \text{(A.9)}$$

Setting $\frac{\partial E}{\partial w} = 0$ we have,

$$\frac{\partial E}{\partial w} = 0 = 0 - \frac{\partial}{\partial w}\left(w^T A^T d\right) - \frac{\partial}{\partial w}\left(d^T Aw\right) + \frac{\partial}{\partial w}\left(w^T A^T Aw\right)$$

$$= -A^T d - \frac{\partial}{\partial w}\left(w^T A^T d\right) + A^T A w + \left(A^T A\right)^T w \quad \text{(A.10)}$$
$$= -2A^T d + 2A^T A w$$

$$\Rightarrow A^T A w = A^T d$$
$$\therefore w = \left(A^T A\right)^{-1} A^T d \quad \text{(A.11)}$$

A.6 Windowing (Park 2000)

To understand why windowing is used in short time Fourier transform, I will show the spectral characteristics when a signal is *windowed* by a simple *rectangular* window. Let us consider the case where *x[n]* is the signal and *w[n]* is the windowing function.

$$x[n] = \begin{cases} a^n, & 0 \leq n \leq N-1 \\ 0, & otherwise \end{cases} \quad \text{(A.12)}$$

$$w[n] = \begin{cases} 1, & 0 \leq n \leq N-1 \\ 0, & otherwise \end{cases} \quad \text{(A.13)}$$

then

$$x_w[n] = x[n]\,w[n] \quad \text{(A.14)}$$

We know that a multiplication in the time domain corresponds to convolution in the frequency domain or more precisely:

$$X_w^f(\Theta) = \frac{1}{2\pi}\{X^f(\Theta) * W^f(\Theta)\} \quad \text{(A.15)}$$

We also observe that for a rectangular window the Fourier transform is

$$\begin{aligned} W^{f}(\Theta) &= \sum_{n=0}^{N-1} e^{-j\Theta n} \\ &= \frac{1-e^{-j\Theta N}}{1-e^{-j\Theta}} \\ &= \frac{\sin(0.5\Theta N)}{\sin(0.5\Theta)} e^{-j0.5\Theta(N-1)} \end{aligned} \qquad \text{(A.16)}$$

Hence, we see that convolution is characteristic of the *Dirichlet kernel. The Dirichlet kernel* causes distortions characterized by the main and side-lobe widths. Each sample in the time domain will render a sync function in the frequency domain causing side effects in the form of spectral smearing due to the finite main-lobe width and side-lobe interference produced by neighboring samples in the signal.

As described above, the choice of windowing functions plays a vital role in short time Fourier analysis. The main criteria in selecting the windowing function is to taper off the abrupt end points of the rectangular window achieving gradual transition. This results in minimized side-lobes magnitudes and minimized main-lobe widths. The behavior of windowing functions can be found in many signal processing textbooks (Porat 1997).

Rectangular window

Rectangular windows are the most common types of windows where the window is determined by ones and zero values – unity gain applied to the entire window as seen in figure A.2. The main lobe width is 4pi/N (N is the number of samples), first side-lobe attenuation is –13.3 dB, and has a roll-off of 20 dB/decade characteristic (roll-off indicates the attenuation rate with respect to frequency). We also notice that there is substantial amount of leakage factor which contributes to frequency smearing due to the side-lobe characteristics; at the same time due to the narrow main-lobe it is well suited for transient based signal.

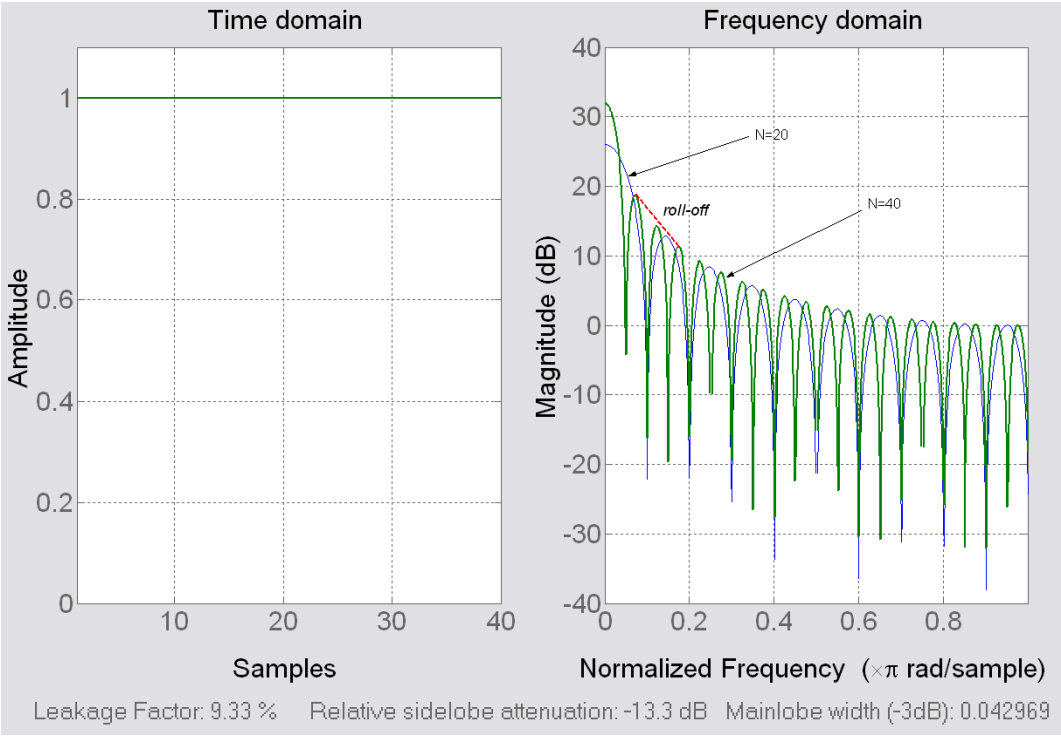


Figure A.2 Rectangular window with 20 and 40 samples length

Hann window

The Hann window also known as the *Hanning* window achieves a side-lobe reduction by superposition. Three *Dirichlet kernels* are shifted and added together resulting in partial cancellation of the side-lobes. The amount of shift is 2pi/(N-1) from the center. The resulting characteristics of the *Hanning* window which is sometimes called the *cosine window* has a first side-lobe level of –32 dB, main lobe width of 8pi/N (N is the number of samples), and a 60 dB/decade roll-off rate. We can notice the considerably sharper roll-off rate but wider main-lobe width and better frequency leaking factor.

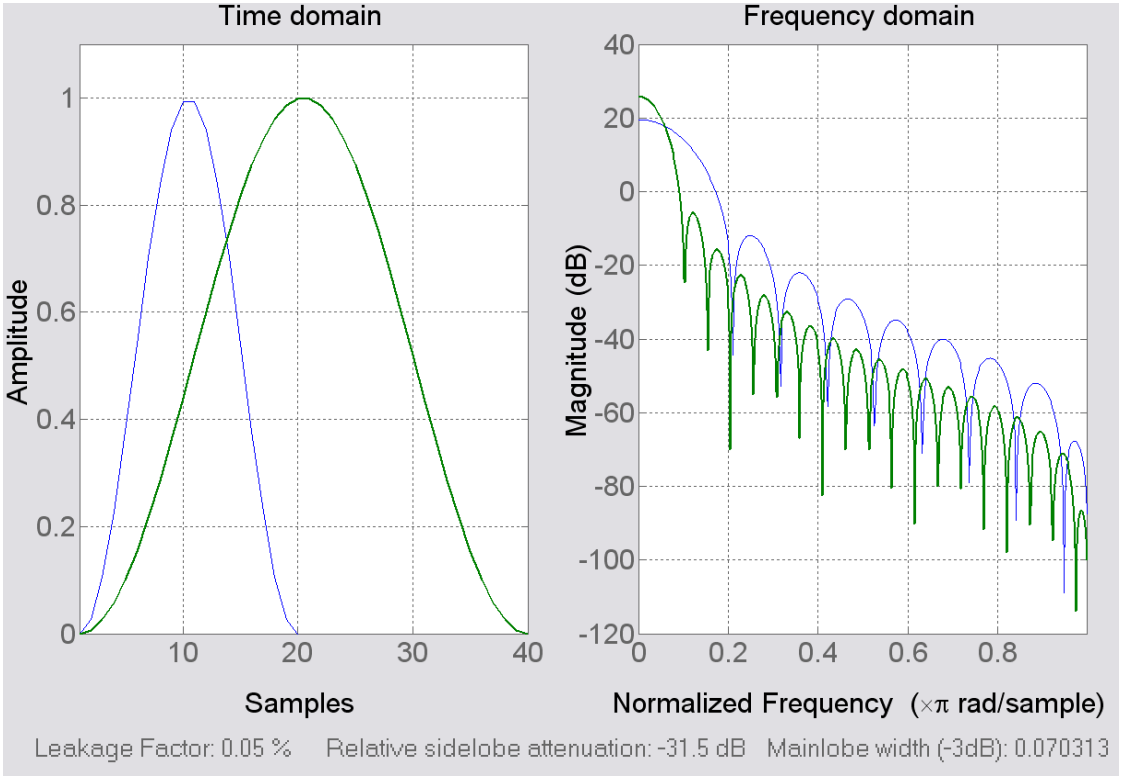


Figure A.3 Hann window characteristics for N = 2 and 40 samples

Hamming window

The *Hamming* window is similar to the *Hanning* window with modifications in weighting the *Dirichlet kernels*. The time and frequency domain windows are as follows. The main-lobe is 8pi/N with –43dB side-lobes and roll-off of 20 dB/decade. One characteristic is the non-zero values at both end points and therefore is sometimes referred to as the half-raised cosine window. N is the number of samples. One interesting observation about the Hamming window is that the first side-lobe is actually smaller than the 2nd side-lobe and has a narrower main-lobe characteristic. Hamming windows are widely used in audio-based spectral analysis and synthesis applications.

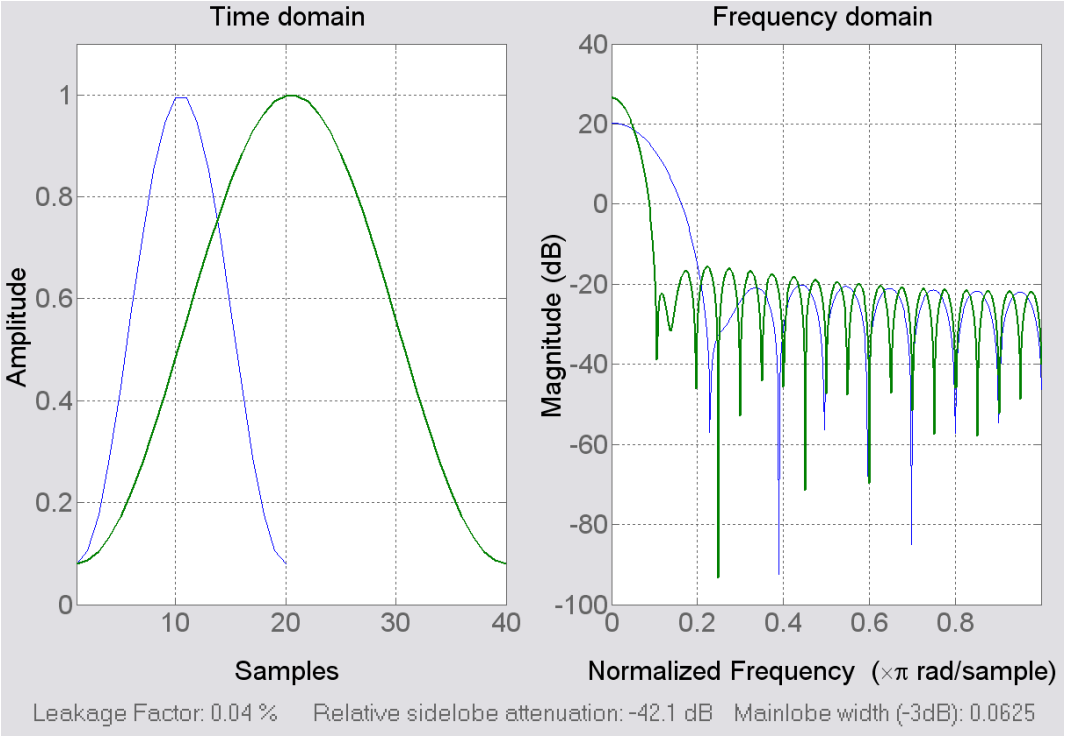


Figure A.4 Hamming window

Blackmann window

The *Blackmann* window has a –58dB side-lobe, a main-lobe width of 12pi/N, and roll-off of a steep 60 dB/decade rate. N is the number of samples.

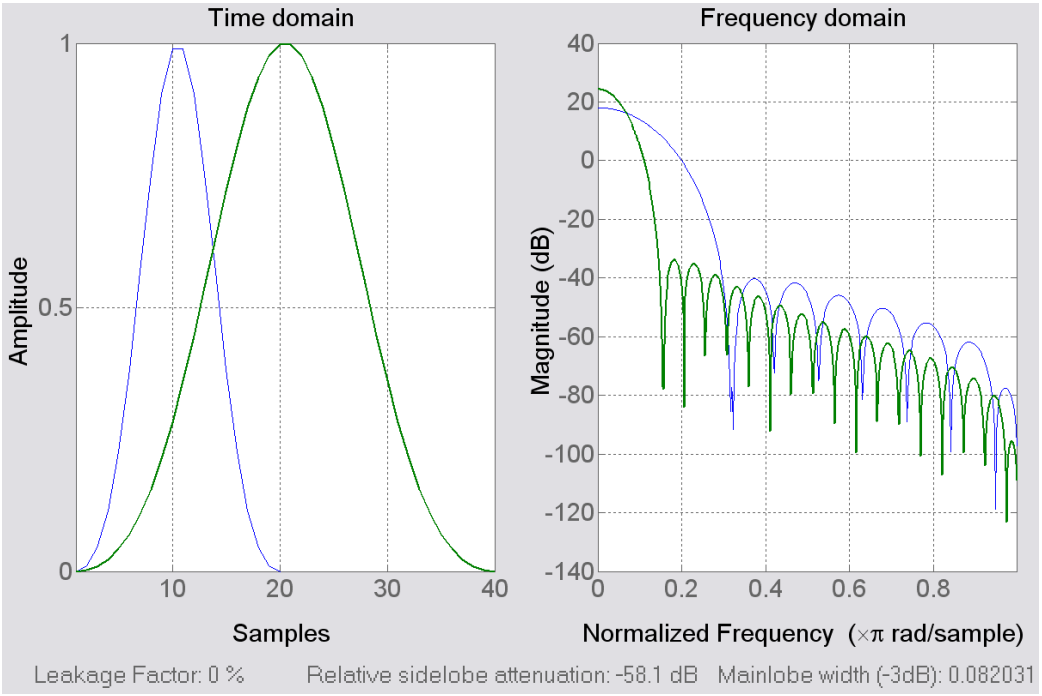
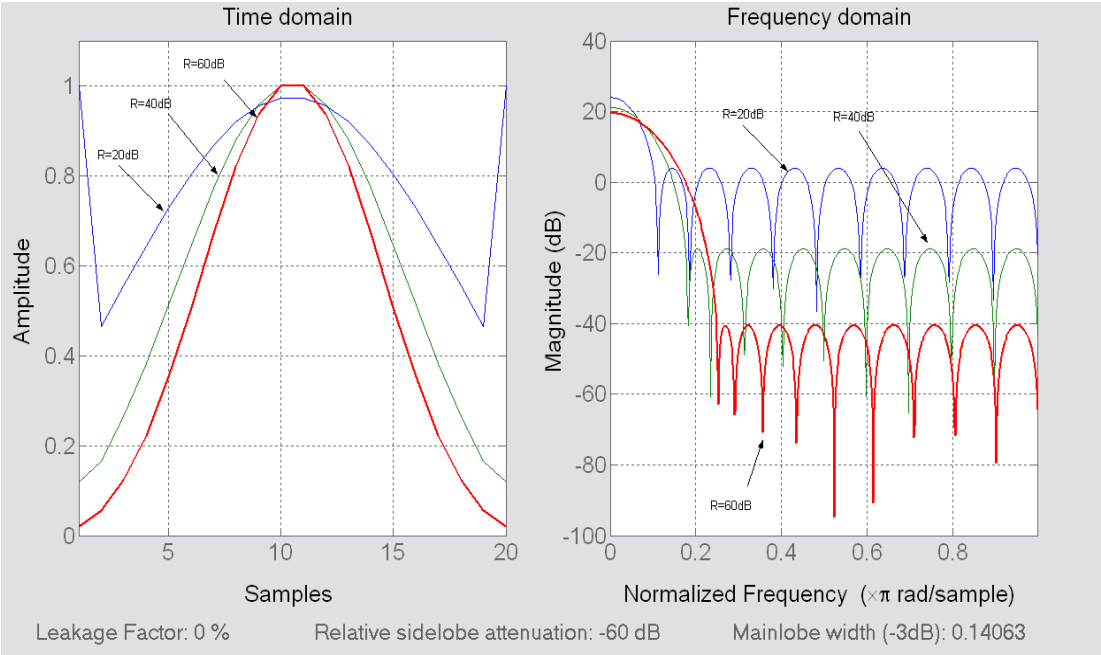


Figure A.5 Blackman window

Other windows – Chebychev and Kaiser

Some window functions such as the Chebychev and Kaiser windows have more direct parametric control over the shape of the windows. The side-lobe attenuation *R* which controls the side-lobe attenuation in the Chebychev window with respect to the main-lobe magnitude is shown in figure A.6 (a). For the Kaiser window the *beta* coefficient determines the side-lobe attenuation characteristics (figure A.6(b)).



A.6(a) Chebychev windows at window length 20 samples

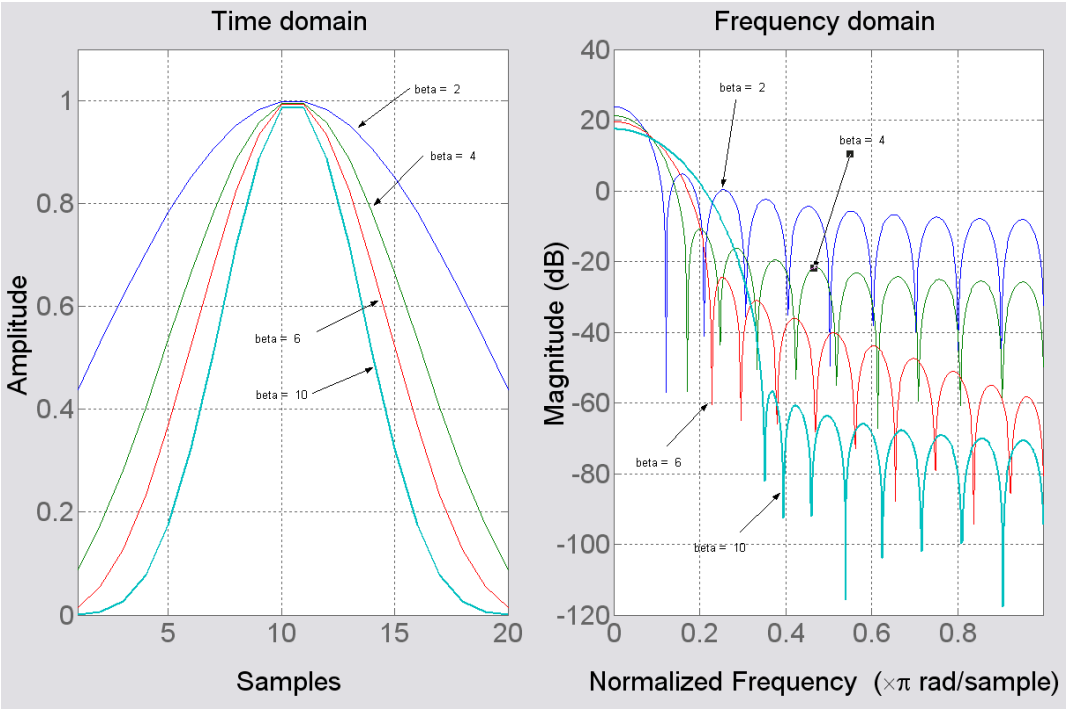


Figure A.6(b) Kaiser windows at window length 20 samples

Figure A.7 shows a comparison plot of some of the windows shown above with window length of 40.

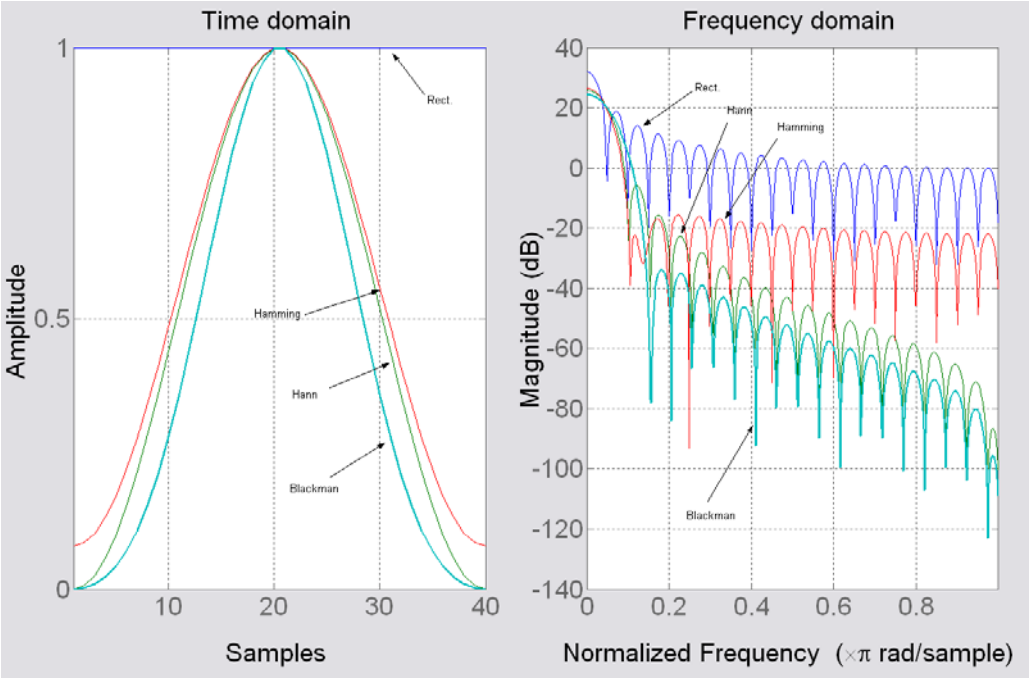


Figure A.7 Comparison of window characteristics:

Rect., Hann, Hamm., and Blackmann

A.7 The Bark Frequency Scale

The *bark frequency scale* is a non-linear frequency scale, which maps frequencies in Hertz to 24 bands of a special psychoacoustic unit called the bark. The bark scale adheres closer to human non-linear hearing where one bark corresponds to the width of one critical band. 24 critical bands with center frequencies and their corresponding band edges are shown in table A.1. A

number of formulas exist in computing the bark scale as shown in equations A.7 to A.23.

Bark Unit	Bark Center Frequencies	Band edges
1	50	0, 100
2	150	100, 200
3	250	200, 300
4	350	300, 400
5	450	400, 510
6	570	510, 630
7	700	630, 770
8	840	770, 920
9	1000	920, 1080
10	1170	1080, 1270
11	1370	1270,1480
12	1600	1480, 1720
13	1850	1720, 2000
14	2150	2000, 2320
15	2500	2320, 2700
16	2900	2700, 3150
17	3400	3150, 3700
18	4000	3700, 4400
19	4800	4400, 5300
20	5800	5300, 6400
21	7000	6400, 7700
22	8500	7700, 9500
23	10500	9500, 12000
24	13500	12000, 15500

Table A.1 Bark frequency scale, center frequency and band edges

Bark algorithms	Authors
$B = 13\tan^{-1}(0.76f/1000) + 3.5\tan^{-1}(f/7500)^2$ (A.17) $B = 8.7 + 14.2\log_{10}(f/1000)$ (A.18)	(Zwicker, Terhardt 1980)
$B = 13.3\tan^{-1}(0.75f/1000)$ (A.19) $B = 12.82\tan^{-1}(0.78f/1000) + 0.17(f/1000)^{1.4}$ (A.20)	(Terhardt 1979)
$B = 6\sinh^{-1}(f/600)$ (A.21)	(Wang, Sekey, Gersho1992)
$B = 7\sinh^{-1}(f/650)$ (A.22)	(Schroeder 1977: according to Paul Carter 2002)
$B = 26.81/[1 + (1960/f)] - 0.53$ (A.23)	(Traunmüller 1990)

Table A.2 Bark frequency scale algorithms, where *B* is in bark and *f* in *Hertz*.

A.8 Additional Flowcharts

A.8.1 Harmonic Analysis

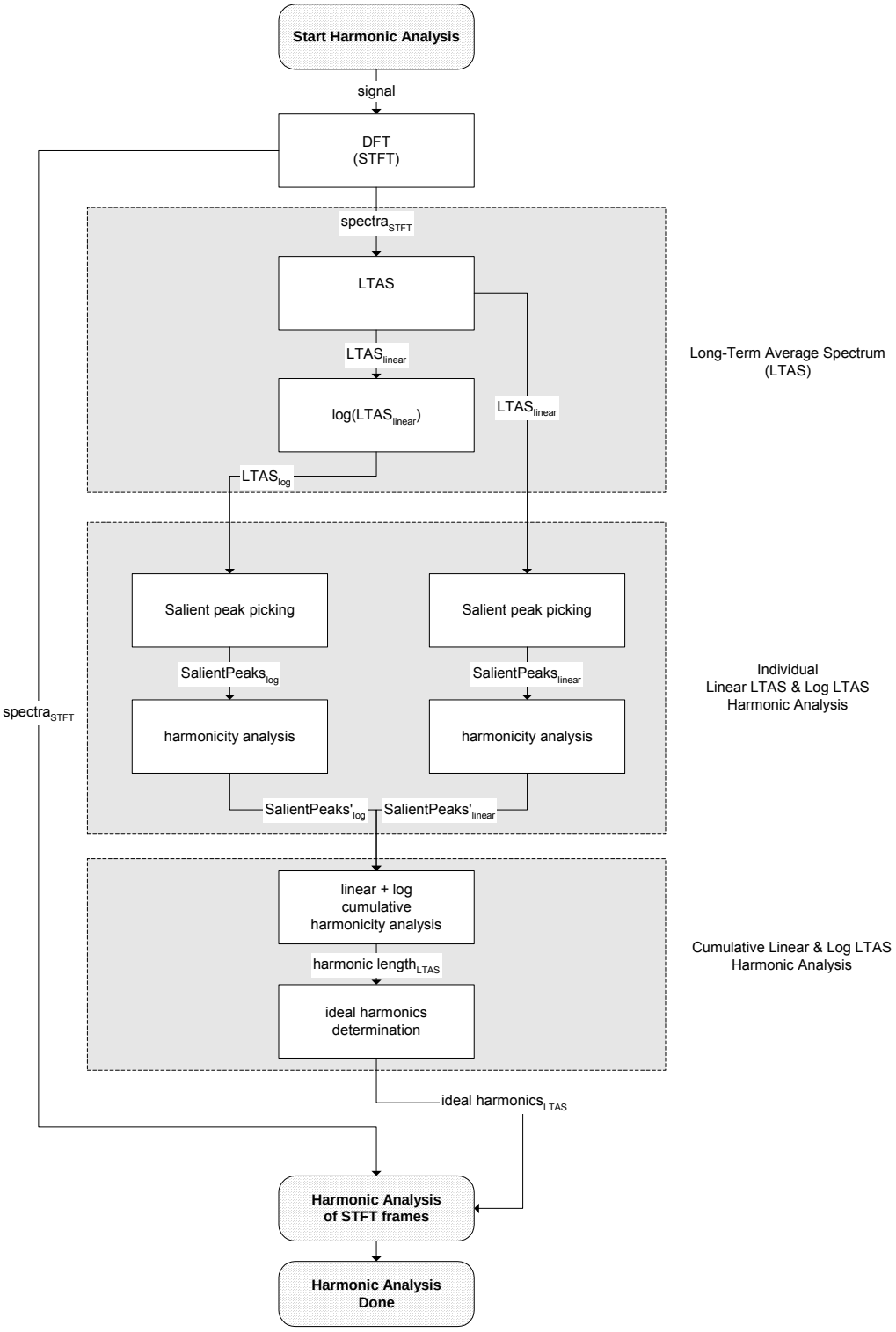


Figure A.8

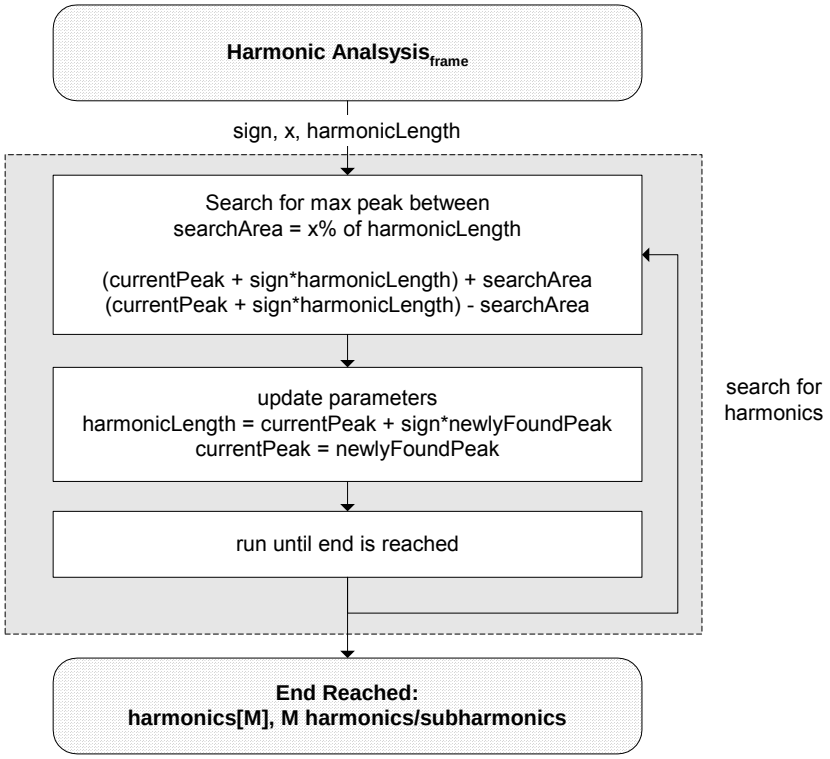

Harmonic Analsysis$_{frame}$
sign, x, harmonicLength
Search for max peak between
searchArea = x% of harmonicLength
(currentPeak + sign*harmonicLength) + searchArea
(currentPeak + sign*harmonicLength) - searchArea
update parameters
harmonicLength = currentPeak + sign*newlyFoundPeak
currentPeak = newlyFoundPeak
run until end is reached
search for
harmonics
End Reached:
harmonics[M], M harmonics/subharmonics

Figure A.9

A.8.2 Nearest Neighbor Error Clustering (NCC)

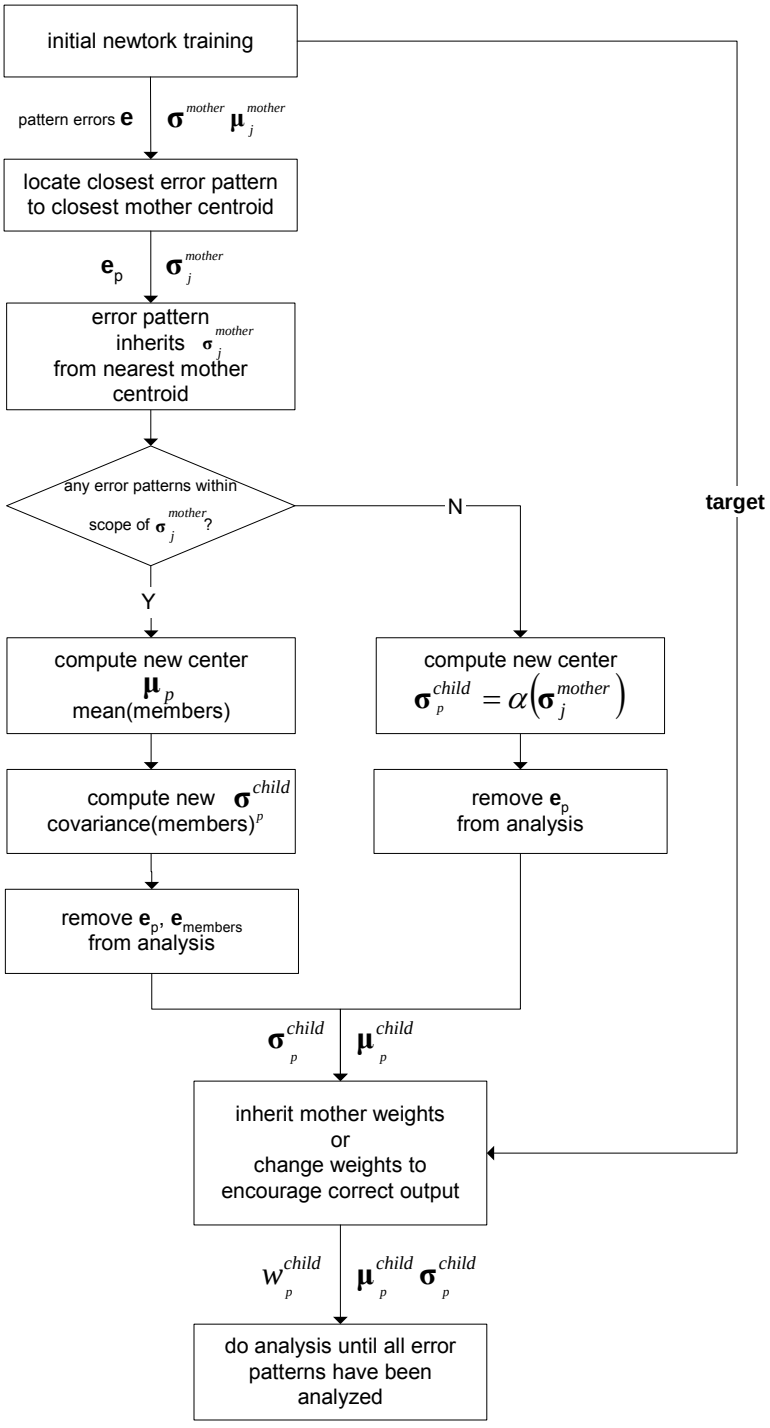


Figure A.10

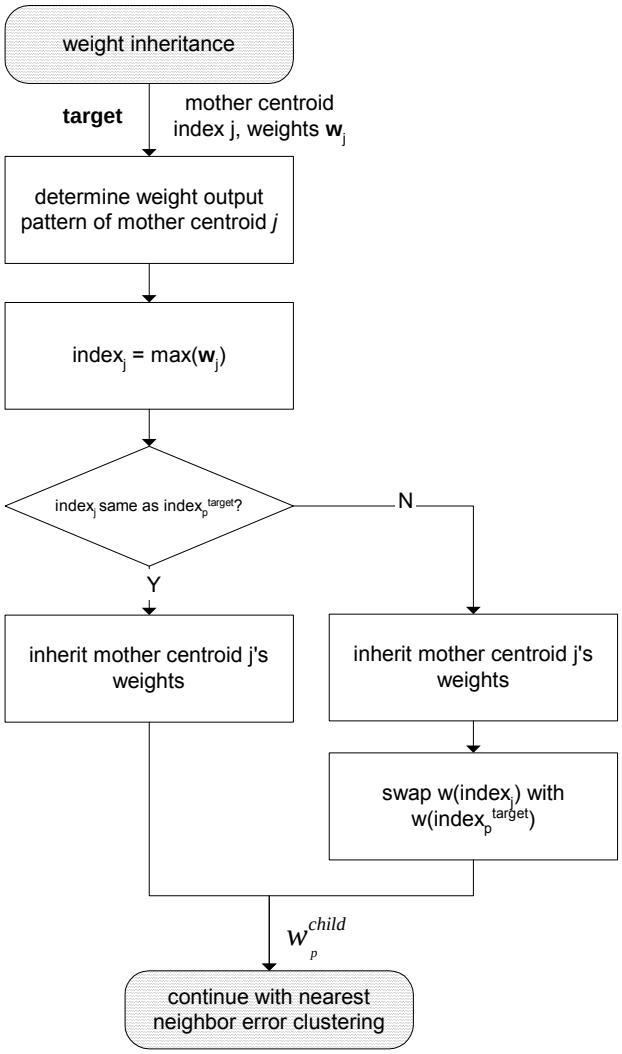

weight inheritance
target
mother centroid index j, weights $\mathbf{w}_j$
determine weight output pattern of mother centroid *j*
$index_j = max(\mathbf{w}_j)$
$index_j$ same as $index_p^{target}$?
N
Y
inherit mother centroid j's weights
inherit mother centroid j's weights
swap $w(index_j)$ with $w(index_p^{target})$
w_p^{child}
continue with nearest neighbor error clustering

Figure A.11

A.8.3 "Confidence Level" based family classification

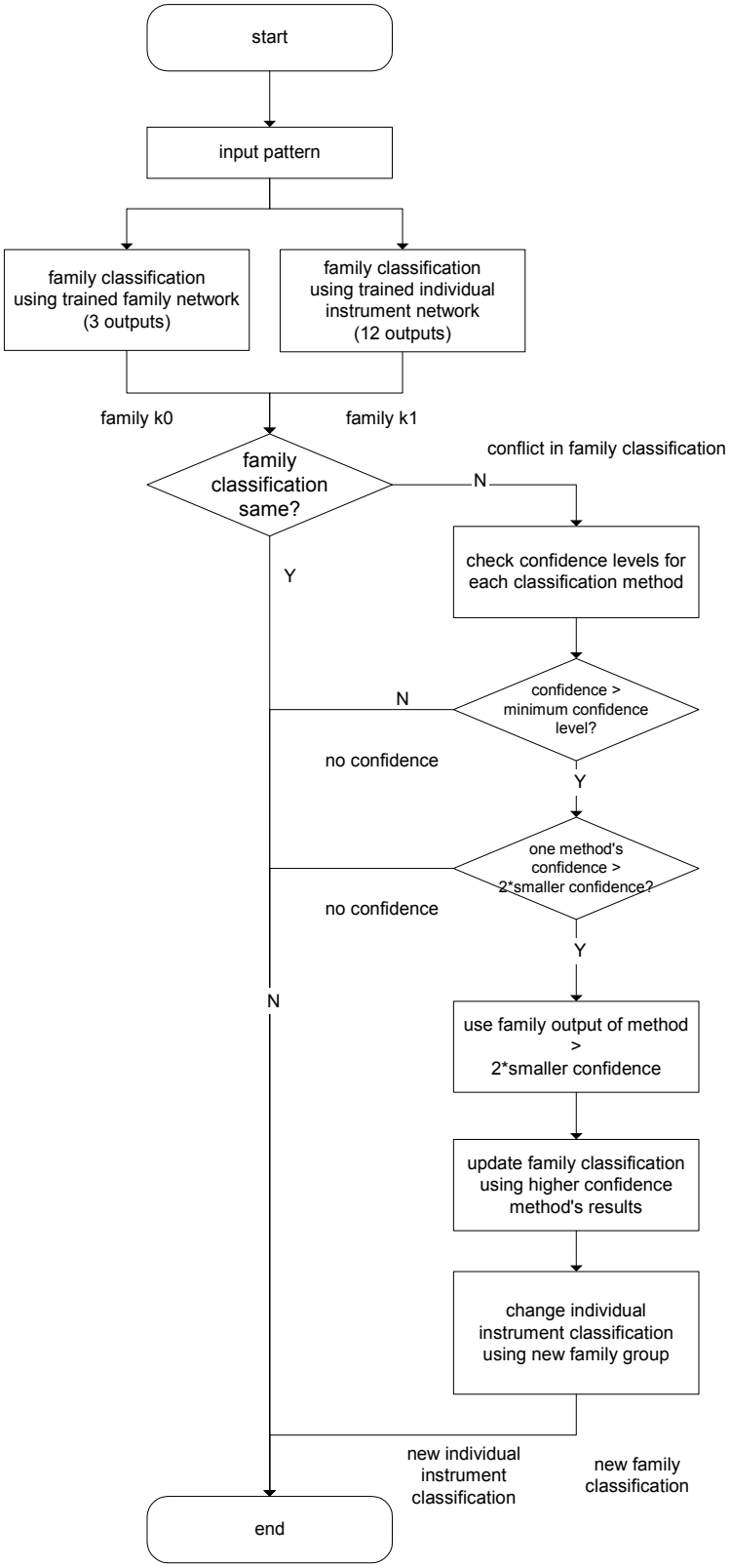


Figure A.12

A.9 Musical Instrument Samples

The table below lists the various techniques, dynamics, and articulations including pizzicato, spiccato, sordino, vibrato, long, sustained, short, muted, pianissimo, piano, mezzo-forte, forte, and fortissimo that were used in testing and developing the system.

Acronym	Definition
*- L	Left signal
*3A	A3 pitch
*C5B5-0	(*C5B5-0 means C5 pitch) (*C5B5-1 means C#5 pitch) (*C5B5-2 means D5 pitch) (*C5B5-3 means D#5 pitch) etc.
*ff	Fortissimo
*LG F	Long forte
*LG FF	Long-fortissimo
*LG P	Long piano
*mf	Mezzo-forte
*novib	No vibrato
*PIZ	Pizzicato
*pp	Pianissimo
*SHDT	Short detaché
*SOR	Con sordino
*SPC	Spiccato
*STAC	Staccato
*STC	Staccato
*STCP/F	Staccato piano/forte
*STSO	Staccato-sordino
*vib	With vibrato
*SUST	Sustain
*XPR	Con Epressivo

Table A.3 Descriptions of acronyms for sound files

Instruments	Number of examples
1.Electric bass	10
2.Violin	105
3.Cello	102
4.Viola	75
5.Bb clarinet	100
6.Flute	99
7.Oboe	55
8.Bassoon	35
9.French horn	56
10.Trumpet	78
11.Trombone	82
12.Tuba	32

Table A.4 Instruments and number of samples

Technique	Number of examples
Electric bass	
Plucked	10
Violin	
Long piano	15
Pizzicato	19
Short detaché	15
Con sordino	15
Spiccato	30
Con Epressivo	11
Cello	
Fortissimo	45
Long piano	7
Pizzicato	24
Con sordino	16
Staccato	11
Viola	
Long-fortissimo	41
Pizzicato	21
Con sordino	13
Clarinet	
Fortissimo	11
Long forte	13
Long piano	12
Mezzo-forte	21
Pianissimo	15
Staccato	28
Flute	
Long forte	13
Long piano	12

No vibrato (FF)	13
No vinrato (PP)	13
Staccato	23
With vibrato (FF)	13
With vibrato (PP)	12
Oboe	
Long forte	11
Long piano	11
Staccato	22
With vibrato	11
Bassoon	
Sustain	13
Staccato	22
French horn	
Fortissimo	2
Long forte	13
Long piano	13
Pianissimo	2
Staccato	26
Trumpet	
Long forte	11
Long piano	11
Con sordino	8
Staccato(normal)	33
Staccato (sordino)	15
Trombone	
Long forte	14
Long piano	13
Staccato (P)	27
Staccato (F)	28
Tuba	
Long forte	13
Long piano	8
Staccato	11

Table A.4 Number of each technique type used

electric basses
bassA22
bassA#44
bassE22
bassF44
dhigh22
d#high44
fhigh22
f#high44
lowD22
lowD#44

violins
VIS SHDT2A-L
VIS SHDT3A-L
VIS SHDT3C-L
VIS SHDT4A-L
VIS SHDT4C-L
VIS SHDT5A-L
VIS SHDT5C-L
VIS SHDTC6-L
VIS SHDTE3-L
VIS SHDTE4-L
VIS SHDTE5-L
VIS SHDTG2-L
VIS SHDTG3-L
VIS SHDTG4-L
VIS SHDTG5-L
VIS LGP C6-LR
VIS LGP E3-LR
VIS LGP E4-LR
VIS LGP E5-LR
VIS LGP G2-LR
VIS LGP G3-LR
VIS LGP G4-LR
VIS LGP G5-LR
VIS LGP#A2-LR
VIS LGP#A3-LR
VIS LGP#A4-LR
VIS LGP#A5-LR
VIS LGP#C3-LR
VIS LGP#C4-LR
VIS LGP#C5-LR
VIS XPR E3-LR
VIS XPR E4-LR
VIS XPR E5-LR
VIS XPR G2-LR
VIS XPR G4-LR
VIS XPR#A2-LR
VIS XPR#A3-LR
VIS XPR#A4-LR
VIS XPR#C3-LR
VIS XPR#C4-LR
VIS XPR#C5-LR
VIS SOR C6-L
VIS SOR E3-L
VIS SOR E4-L
VIS SOR E5-L
VIS SOR G2-L
VIS SOR G3-L
VIS SOR G4-L
VIS SOR G5-L
VIS SOR#A2-L
VIS SOR#A3-L
VIS SOR#A4-L
VIS SOR#A5-L
VIS SOR#C3-L
VIS SOR#C4-L
VIS SOR#C5-L
VIS1SPC C6-LR
VIS1SPC E3-LR
VIS1SPC E4-LR
VIS1SPC E5-LR
VIS1SPC G2-LR
VIS1SPC G3-LR
VIS1SPC G4-LR
VIS1SPC G5-LR
VIS1SPC#A2-LR
VIS1SPC#A3-LR
VIS1SPC#A4-LR
VIS1SPC#A5-LR
VIS1SPC#C3-LR
VIS1SPC#C4-LR
VIS1SPC#C5-LR
VIS2SPC C6-LR
VIS2SPC E3-LR
VIS2SPC E4-LR
VIS2SPC E5-LR
VIS2SPC G2-LR
VIS2SPC G3-LR
VIS2SPC G4-LR
VIS2SPC G5-LR
VIS2SPC#A2-LR
VIS2SPC#A3-LR
VIS2SPC#A4-LR
VIS2SPC#A5-LR
VIS2SPC#C3-LR
VIS2SPC#C4-LR
VIS2SPC#C5-LR
VIS1PIZ E3-LR
VIS1PIZ E4-LR
VIS1PIZ G2-LR
VIS1PIZ G3-LR
VIS1PIZ G4-LR
VIS1PIZ#A2-LR
VIS1PIZ#A3-LR
VIS1PIZ#A4-LR
VIS1PIZ#C3-LR
VIS1PIZ#C4-LR
VIS1PIZ#C5-LR
VIS2PIZ E3-LR
VIS2PIZ E4-LR
VIS2PIZ G3-LR
VIS2PIZ G4-LR
VIS2PIZ#A2-LR
VIS2PIZ#A3-LR
VIS2PIZ#A4-LR
VIS2PIZ#C4-LR

cellos
CLS1LGFF1D-LR
CLS1LGFF1F-LR
CLS1LGFF2D-LR
CLS1LGFF2F-LR
CLS1LGFF3D-LR
CLS1LGFF3F-LR
CLS1LGFF4D-LR
CLS1LGFF4F-LR
CLS1LGFFA1-LR
CLS1LGFFA2-LR
CLS1LGFFA3-LR
CLS1LGFFA4-LR
CLS1LGFFC1-LR
CLS1LGFFC2-LR
CLS1LGFFC3-LR
CLS1LGFFC4-LR
CLS2LGFF1F-LR
CLS2LGFF2D-LR
CLS2LGFF2F-LR
CLS2LGFF3D-LR
CLS2LGFF3F-LR
CLS2LGFF4D-LR
CLS2LGFF4F-LR
CLS2LGFFA2-LR
CLS2LGFFA3-LR
CLS2LGFFA4-LR
CLS2LGFFC2-LR
CLS2LGFFC3-LR
CLS2LGFFC4-LR
CLS LGP A2-L
CLS LGP A4-L
CLS LGP C1-L
CLS LGP#D1-L
CLS LGP#D3-L
CLS LGP#F3-L
CLS LGP#F4-L
CLS1LGFF1D-L
CLS1LGFF1F-L
CLS1LGFF2D-L
CLS1LGFF2F-L
CLS1LGFF3D-L
CLS1LGFF3F-L
CLS1LGFF4D-L
CLS1LGFF4F-L
CLS1LGFFA1-L
CLS1LGFFA2-L
CLS1LGFFA3-L
CLS1LGFFA4-L
CLS1LGFFC1-L
CLS1LGFFC2-L
CLS1LGFFC3-L
CLS1LGFFC4-L
CLS1PIZ A1-L
CLS1PIZ A2-L
CLS1PIZ A3-L
CLS1PIZ C2-L
CLS1PIZ C3-L
CLS1PIZ C4-L
CLS1PIZ#D2-L
CLS1PIZ#D3-L
CLS1PIZ#D4-L
CLS1PIZ#F1-L
CLS1PIZ#F2-L
CLS1PIZ#F3-L
CLS2PIZ A1-L
CLS2PIZ A2-L
CLS2PIZ A3-L
CLS2PIZ C2-L
CLS2PIZ C3-L
CLS2PIZ C4-L
CLS2PIZ#D2-L
CLS2PIZ#D3-L
CLS2PIZ#D4-L
CLS2PIZ#F1-L
CLS2PIZ#F2-L
CLS2PIZ#F3-L
CLS SOR A1-L
CLS SOR A2-L
CLS SOR A3-L
CLS SOR A4-L
CLS SOR C1-L
CLS SOR C2-L
CLS SOR C3-L
CLS SOR C4-L
CLS SOR#D1-L
CLS SOR#D2-L
CLS SOR#D4-L
CLS SOR#F1-L
CLS SOR#F2-L
CLS SOR#F3-L
CLS SOR#F4-L
CLS1STC A1-L
CLS1STC A2-L
CLS1STC A3-L
CLS1STC C2-L
CLS1STC C3-L
CLS1STC#D1-L
CLS1STC#D2-L
CLS1STC#D3-L
CLS1STC#F1-L
CLS1STC#F2-L
CLS1STC#F3-L

violas
VAS1LGFF2D-LR
VAS1LGFF2F-LR
VAS1LGFF3D-LR
VAS1LGFF3F-LR
VAS1LGFF4D-LR
VAS1LGFF4F-LR
VAS1LGFF5D-LR
VAS1LGFFA2-LR
VAS1LGFFA3-LR
VAS1LGFFA4-LR
VAS1LGFFC2-LR
VAS1LGFFC3-LR
VAS1LGFFC4-LR
VAS1LGFFC5-LR
VAS2LGFF2D-LR
VAS2LGFF2F-LR
VAS2LGFF3D-LR
VAS2LGFF3F-LR
VAS2LGFF4D-LR
VAS2LGFF4F-LR
VAS2LGFF5D-LR
VAS2LGFFA2-LR
VAS2LGFFA3-LR
VAS2LGFFA4-LR
VAS2LGFFC2-LR
VAS2LGFFC3-LR
VAS2LGFFC4-LR
VAS2LGFFC5-LR
VAS3LGFF2D-LR
VAS3LGFF2F-LR
VAS3LGFF3D-LR
VAS3LGFF3F-LR
VAS3LGFF4D-LR
VAS3LGFF4F-LR
VAS3LGFF5D-LR
VAS3LGFFA2-LR
VAS3LGFFA3-LR
VAS3LGFFA4-LR
VAS3LGFFC3-LR
VAS3LGFFC4-LR
VAS3LGFFC5-LR
VAS1PIZ A2-LR
VAS1PIZ A3-LR
VAS1PIZ A4-LR
VAS1PIZ C2-LR
VAS1PIZ C3-LR
VAS1PIZ C4-LR
VAS1PIZ C5-LR
VAS1PIZ#D3-LR
VAS1PIZ#D4-LR
VAS1PIZ#F2-LR
VAS1PIZ#F3-LR
VAS2PIZ A2-LR
VAS2PIZ A3-LR
VAS2PIZ A4-LR
VAS2PIZ C2-LR
VAS2PIZ C3-LR
VAS2PIZ C4-LR
VAS2PIZ C5-LR
VAS2PIZ#D3-LR
VAS2PIZ#D4-LR
VAS2PIZ#F3-LR
VAS1SOR A2-LR
VAS1SOR A3-LR
VAS1SOR A4-LR
VAS1SOR C2-LR
VAS1SOR C3-LR
VAS1SOR C4-LR
VAS1SOR C5-LR
VAS1SOR#D2-LR
VAS1SOR#D3-LR
VAS1SOR#D4-LR
VAS1SOR#F2-LR
VAS1SOR#F3-LR
VAS1SOR#F4-LR

Bb clarinets
BbClar.ff.C5B5-0
BbClar.ff.C5B5-1
BbClar.ff.C5B5-10
BbClar.ff.C5B5-11
BbClar.ff.C5B5-2
BbClar.ff.C5B5-3
BbClar.ff.C5B5-4
BbClar.ff.C5B5-6
BbClar.ff.C5B5-7
BbClar.ff.C5B5-8
BbClar.ff.C5B5-9
BbClar.mf.C5B5-0
BbClar.mf.C5B5-1
BbClar.mf.C5B5-10
BbClar.mf.C5B5-11
BbClar.mf.C5B5-2
BbClar.mf.C5B5-3
BbClar.mf.C5B5-4
BbClar.mf.C5B5-5
BbClar.mf.C5B5-6
BbClar.mf.C5B5-7
BbClar.mf.C5B5-8
BbClar.mf.C5B5-9
BbClar.mf.D3B3-0
BbClar.mf.D3B3-1
BbClar.mf.D3B3-2
BbClar.mf.D3B3-3
BbClar.mf.D3B3-4
BbClar.mf.D3B3-6
BbClar.mf.D3B3-7
BbClar.mf.D3B3-8
BbClar.mf.D3B3-9
BbClar.pp.C5B5-0
BbClar.pp.C5B5-1
BbClar.pp.C5B5-10
BbClar.pp.C5B5-11
BbClar.pp.C5B5-2
BbClar.pp.C5B5-5
BbClar.pp.C5B5-6
BbClar.pp.C5B5-7
BbClar.pp.C5B5-8
BbClar.pp.D3B3-4
BbClar.pp.D3B3-5

BbClar.pp.D3B3-6
BbClar.pp.D3B3-7
BbClar.pp.D3B3-8
BbClar.pp.D3B3-9
CTS LG F A2
CTS LG F A3
CTS LG F A4
CTS LG F C3
CTS LG F C4
CTS LG F #D2
CTS LG F #D3
CTS LG F #D4
CTS LG F #D5
CTS LG F #F2
CTS LG F #F3
CTS LG F #F4
CTS LG F #F5
CTS LG P A2
CTS LG P A3
CTS LG P A4
CTS LG P C3
CTS LG P C5
CTS LG P #D3
CTS LG P #D4
CTS LG P #D5
CTS LG P #F2
CTS LG P #F3
CTS LG P #F4
CTS LG P #F5
CTS1STAC A2
CTS1STAC A3
CTS1STAC A4
CTS1STAC C3
CTS1STAC C4
CTS1STAC C5
CTS1STAC #D2
CTS1STAC #D3
CTS1STAC #D4
CTS1STAC #D5
CTS1STAC #F2
CTS1STAC #F3
CTS1STAC #F4
CTS1STAC #F5
CTS2STAC A2
CTS2STAC A3
CTS2STAC A4
CTS2STAC C3
CTS2STAC C4
CTS2STAC C5
CTS2STAC #D2
CTS2STAC #D3
CTS2STAC #D4
CTS2STAC #D5
CTS2STAC #F2
CTS2STAC #F3
CTS2STAC #F4
CTS2STAC #F5

flutes

FLS LG F A3
FLS LG F A4
FLS LG F A5
FLS LG F C3
FLS LG F C4
FLS LG F C5
FLS LG F C6
FLS LG F #D3
FLS LG F #D4
FLS LG F #D5
FLS LG F #F3
FLS LG F #F4
FLS LG F #F5
FLS LG P A3
FLS LG P A4
FLS LG P A5
FLS LG P C3
FLS LG P C4
FLS LG P C5
FLS LG P C6
FLS LG P #D3
FLS LG P #D4
FLS LG P #D5
FLS LG P #F3
FLS LG P #F4
FLS1STAC A3
FLS1STAC A4
FLS1STAC A5
FLS1STAC C3
FLS1STAC C4
FLS1STAC C5
FLS1STAC C6
FLS1STAC #D3
FLS1STAC #D4
FLS1STAC #D5
FLS1STAC #F3
FLS1STAC #F4
FLS2STAC A3
FLS2STAC A4
FLS2STAC A5
FLS2STAC C3
FLS2STAC C4
FLS2STAC C5
FLS2STAC #D3
FLS2STAC #D4
FLS2STAC #D5
FLS2STAC #F3
FLS2STAC #F4
flute.ff.B3B4.novib1
flute.ff.B3B4.novib10
flute.ff.B3B4.novib11
flute.ff.B3B4.novib12
flute.ff.B3B4.novib13
flute.ff.B3B4.novib2
flute.ff.B3B4.novib3
flute.ff.B3B4.novib4
flute.ff.B3B4.novib5
flute.ff.B3B4.novib6
flute.ff.B3B4.novib7
flute.ff.B3B4.novib8
flute.ff.B3B4.novib9
flute.ff.B3B4.vib1
flute.ff.B3B4.vib10
flute.ff.B3B4.vib11
flute.ff.B3B4.vib12
flute.ff.B3B4.vib13
flute.ff.B3B4.vib2
flute.ff.B3B4.vib3
flute.ff.B3B4.vib4
flute.ff.B3B4.vib5
flute.ff.B3B4.vib6
flute.ff.B3B4.vib7
flute.ff.B3B4.vib8
flute.ff.B3B4.vib9
flute.pp.B3B4.novib1
flute.pp.B3B4.novib10
flute.pp.B3B4.novib11
flute.pp.B3B4.novib12
flute.pp.B3B4.novib13
flute.pp.B3B4.novib2
flute.pp.B3B4.novib3
flute.pp.B3B4.novib4
flute.pp.B3B4.novib5
flute.pp.B3B4.novib6
flute.pp.B3B4.novib7
flute.pp.B3B4.novib8
flute.pp.B3B4.novib9
flute.pp.B3B4.vi11
flute.pp.B3B4.vib1
flute.pp.B3B4.vib10
flute.pp.B3B4.vib11
flute.pp.B3B4.vib2
flute.pp.B3B4.vib3
flute.pp.B3B4.vib4
flute.pp.B3B4.vib5
flute.pp.B3B4.vib6
flute.pp.B3B4.vib7
flute.pp.B3B4.vib8
flute.pp.B3B4.vib9

oboes

OBS LG F A3
OBS LG F A4
OBS LG F C3
OBS LG F C4
OBS LG F C5
OBS LG F #D3
OBS LG F #D4
OBS LG F #D5
OBS LG F #F3
OBS LG F #F4
OBS LG F #F5
OBS LG P A3
OBS LG P A4
OBS LG P C3
OBS LG P C4
OBS LG P C5
OBS LG P #D3
OBS LG P #D4
OBS LG P #D5
OBS LG P #F3
OBS LG P #F4
OBS LG P #F5
OBS VIBR A3
OBS VIBR A4
OBS VIBR C3
OBS VIBR C4
OBS VIBR C5
OBS VIBR #D3
OBS VIBR #D4
OBS VIBR #D5
OBS VIBR #F3
OBS VIBR #F4
OBS VIBR #F5
OBS1STAC A3
OBS1STAC A4
OBS1STAC C3
OBS1STAC C4
OBS1STAC C5
OBS1STAC #D3
OBS1STAC #D4
OBS1STAC #D5
OBS1STAC #F3
OBS1STAC #F4
OBS1STAC #F5
OBS2STAC A3
OBS2STAC A4
OBS2STAC C3
OBS2STAC C4
OBS2STAC C5
OBS2STAC #D3
OBS2STAC #D4
OBS2STAC #D5
OBS2STAC #F3
OBS2STAC #F4
OBS2STAC #F5

bassoons

BNS SUST E1
BNS SUST E2
BNS SUST E3
BNS SUST G1
BNS SUST G2
BNS SUST G3
BNS SUST #A0
BNS SUST #A1
BNS SUST #A2
BNS SUST #A3
BNS SUST #C1
BNS SUST #C2
BNS SUST #C3
BNS1STAC E1
BNS1STAC E2
BNS1STAC E3
BNS1STAC G1
BNS1STAC G2
BNS1STAC #A0
BNS1STAC #A1
BNS1STAC #A2
BNS1STAC #A3
BNS1STAC #C1
BNS1STAC #C2
BNS1STAC #C3
BNS2STAC E1
BNS2STAC E2
BNS2STAC G1
BNS2STAC G2
BNS2STAC #A1
BNS2STAC #A2
BNS2STAC #A3
BNS2STAC #C1
BNS2STAC #C2
BNS2STAC #C3

French horns

FHS LG F B1
FHS LG F B2
FHS LG F B3
FHS LG F D2
FHS LG F D3
FHS LG F D4
FHS LG F F1
FHS LG F F2
FHS LG F F3
FHS LG F F4
FHS LG F #G1
FHS LG F #G2
FHS LG F #G3
FHS LG P B1
FHS LG P B2
FHS LG P B3
FHS LG P D2
FHS LG P D3
FHS LG P D4
FHS LG P F1
FHS LG P F2
FHS LG P F3
FHS LG P F4
FHS LG P #G1
FHS LG P #G2
FHS LG P #G3
FHS1STAC B1
FHS1STAC B2
FHS1STAC B3
FHS1STAC D2
FHS1STAC D3
FHS1STAC D4
FHS1STAC F1
FHS1STAC F2
FHS1STAC F3
FHS1STAC F4
FHS1STAC #G1
FHS1STAC #G2
FHS1STAC #G3
FHS2STAC B1
FHS2STAC B2
FHS2STAC B3
FHS2STAC D2
FHS2STAC D3
FHS2STAC D4
FHS2STAC F1
FHS2STAC F2
FHS2STAC F3
FHS2STAC F4
FHS2STAC #G1
FHS2STAC #G2
FHS2STAC #G3
horn.mf.Bb1B1-0
horn.mf.Bb1B1-1
horn.pp.Bb1B1-0
horn.pp.Bb1B1-1

trumpets

TRS LG F E2
TRS LG F E3
TRS LG F E4
TRS LG F G2
TRS LG F G3
TRS LG F G4
TRS LG F #A2
TRS LG F #A3
TRS LG F #A4
TRS LG F #C3
TRS LG F #C4
TRS LG P E2
TRS LG P E3
TRS LG P E4
TRS LG P G2
TRS LG P G3
TRS LG P G4
TRS LG P #A2
TRS LG P #A3
TRS LG P #A4
TRS LG P #C3
TRS LG P #C4
TRS SOR E3
TRS SOR E4
TRS SOR G3
TRS SOR G4
TRS SOR #A2
TRS SOR #A4
TRS SOR #C3
TRS SOR #C4
TRS1STAC E2
TRS1STAC E3

TRS1STAC E4
TRS1STAC G2
TRS1STAC G3
TRS1STAC G4
TRS1STAC #A2
TRS1STAC #A3
TRS1STAC #A4
TRS1STAC #C3
TRS1STAC #C4
TRS2STAC E2
TRS2STAC E3
TRS2STAC E4
TRS2STAC G2
TRS2STAC G3
TRS2STAC G4
TRS2STAC #A2
TRS2STAC #A3
TRS2STAC #A4
TRS2STAC #C3
TRS2STAC #C4
TRS1STSO E3
TRS1STSO E4
TRS1STSO G3
TRS1STSO G4
TRS1STSO #A2
TRS1STSO #A3
TRS1STSO #A4
TRS1STSO #C3
TRS2STAC E2
TRS2STAC E3
TRS2STAC E4
TRS2STAC G2
TRS2STAC G3
TRS2STAC G4
TRS2STAC #A2
TRS2STAC #A3
TRS2STAC #A4
TRS2STAC #C3
TRS2STAC #C4
TRS2STSO E3
TRS2STSO E4
TRS2STSO G4
TRS2STSO #A2
TRS2STSO #A3
TRS2STSO #A4
TRS2STSO #C3

trombones

TBS LG F E1
TBS LG F E3
TBS LG F G1
TBS LG F G2
TBS LG F G3
TBS LG F #A0
TBS LG F #A1
TBS LG F #A2
TBS LG F #A3
TBS LG F #C1
TBS LG F #C2
TBS LG F #C3
TBS LG F #C4
TBS LG F #E2
TBS LG P E1
TBS LG P E2
TBS LG P E3
TBS LG P G1
TBS LG P G2
TBS LG P G3
TBS LG P #A0
TBS LG P #A1
TBS LG P #A2
TBS LG P #A3
TBS LG P #C1
TBS LG P #C3
TBS LG P #C4
TBS1STCP E1
TBS1STCP E2
TBS1STCP E3
TBS1STCP G1
TBS1STCP G2
TBS1STCP G3
TBS1STCP #A0
TBS1STCP #A1
TBS1STCP #A2
TBS1STCP #A3
TBS1STCP #C1
TBS1STCP #C2
TBS1STCP #C3
TBS1STCP #C4
TBS2STCP E1
TBS2STCP E2
TBS2STCP E3
TBS2STCP G1
TBS2STCP G2
TBS2STCP G3
TBS2STCP #A1
TBS2STCP #A2
TBS2STCP #A3
TBS2STCP #C1
TBS2STCP #C2
TBS2STCP #C3
TBS2STCP #C4
TBS1STCF E1
TBS1STCF E2
TBS1STCF E3
TBS1STCF G1
TBS1STCF G2
TBS1STCF G3
TBS1STCF #A0
TBS1STCF #A1
TBS1STCF #A2
TBS1STCF #A3
TBS1STCF #C1
TBS1STCF #C2
TBS1STCF #C3
TBS1STCF #C4
TBS2STCF E1
TBS2STCF E2
TBS2STCF E3
TBS2STCF G1
TBS2STCF G2
TBS2STCF G3
TBS2STCF #A0
TBS2STCF #A1
TBS2STCF #A2
TBS2STCF #A3
TBS2STCF #C1
TBS2STCF #C2
TBS2STCF #C3
TBS2STCF #C4

tubas

TUS LG F A0
TUS LG F A1
TUS LG F A2
TUS LG F C1
TUS LG F C2
TUS LG F C3
TUS LG F #D0
TUS LG F #D1
TUS LG F #D2
TUS LG F #D3
TUS LG F #F0
TUS LG F #F1
TUS LG F #F2
TUS LG P A1
TUS LG P A2
TUS LG P C2
TUS LG P C3
TUS LG P #D1
TUS LG P #D2
TUS LG P #F1
TUS LG P #F2
TUS1STAC A0
TUS1STAC A1
TUS1STAC C1
TUS1STAC C2
TUS1STAC #D1
TUS1STAC #D2
TUS1STAC #F0
TUS1STAC #F1
TUS1STAC #F2
TUS2STAC A0
TUS2STAC A1

References

Allen, J. B. 1977. Short Term Spectral Analysis, Synthesis, and Modification by Discrete Fourier Transform. IEEE Transactions on Acoustics, Speech, Signal Processing, vol. ASSP-25, no. 3, pp. 235-238.

Allen, J. B., Rabiner, L. R. 1977. A Unified Approach to Short-Time Fourier Analysis and Synthesis. Proc. IEEE, vol. 65, no. 11, pp. 1558-1564.

Andersen, T. H., Jensen, K. 2002. Importance of Phase in Sound Modeling of Acoustic Instruments, Proceedings of the Mosart Midterm Meeting, Esbjerg, Denmark.

ANSI (American National Standards Institute) 1960, 1970.

ASA (American Standard Association) 1960. Acoustical Terminology, New York.

Backus, J. 1976. Input Impedance Curves for the Brass Instruments. Journal of the Acoustical Society of America, 60, pp. 470 - 480.

Bahl, L. R., Brown, P. F.,de Souza, P. V., Mercer, R. L. 1989. A Tree-Based Statistical Language Model for Natural Language Speech Recognition. IEEE Trans. on Acoustics, Speech and Signal Processing, Vol. ASSP-37, No. 7

Bartholomew, B. T. 1945. Acoustic Music. Prentice-Hall, NJ.

Bekesy, G. 1960. Experiments in Hearing. New York: McGraw-Hill.

Berebzweig A., Logan B, Ellis D., Whitman B. 2004. A Large-Scale Evaluation of Acoustic and Subjective Music-Similarity Measures. Computer Music Journal, 28(2):pp 63-76. MIT Press. Cambridge Massachusetts.

Berthold, M. R. 1994. A Time Delay radial Basis Function Network for Phoneme Recognition. Proceedings of the IEEE International Conference on Neural Networks, vol. 7, pp. 4470 - 4473.

Beauchamp J. 1969. A Computer System for Time-Variant Harmonic Analysis and Synthesis of Musical Tones. In Music by Computers, H.F. von Foerster and J.W. Beauchamp, eds., pp. 19-62, John Wiley, New York, NY.

Best Service – Sounds & More. Hanauer Straße 91a, 80993 München, Germany.

Birgmeier, M. 1996. Nonlinear Prediction of Speech Signals Using Radial basis Function Networks. EUSIPCO 1996, vol. 1, pp. 459 - 462.

von Bismark G. 1974. Timbre of Steady Sounds: A Factorial Investigation of its Verbal Attributes. Acoustica, Vol. 30, pp. 146 -159.

Bregman, A. 1990. Auditory Scene Analysis: The Perceptual Organization of Sound. MIT Press: Cambridge.

Brown, J.C., Houix, O., McAdams, S. 2001. Feature Dependence in the the Automatic Identification of Musical Woodwind Instruments. Journal of the Acoustical Society of America 109, pp. 1064-1072.

Carter, P. 2002. Structured Variation in British English Liquids: the Role of Resonance. University of York, Department of Language and Linguistic Science.

Chapin, E. K., Firestone, F. A. 1934. The Influence of Phase on Tonal Quality and Loudness; the Interference of Subjective Harmonics. Journal of the Acoustical Society of America vol. 5, pp. 173 - 180.

Cherry, E. C. 1953. Some Experiments on the Recognition of Speech with One and with Two Ears. Journal of the Acoustical Society of America, Vol. 25, pp.975 - 979.

Drioli, C. 1999. Radial Basis Function Networks for Conversion of Sound Spectra. Proceedings of the DAFX99 Conference. Available on-line at http://www.tele.ntnu.no/akustikk/meetings/DAFx99/papers.html (during writing of thesis).

Ellis, D. P. W. 1996. Prediction-driven Computational Auditory Scene Analysis for Dense Sound Mixtures. ESCA Workshop Keele.

Er, M., Wu S., Lu J., Toh H. 2002. Face Recognition with Radial Basis Neural Networks. IEEE Transactions on Neural Networks, Vol. 13, No. 3.

Eronen A., Klapuri A. 2000. Musical Instrument Recognition Using Cepstral Coefficients and Temporal Features. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2000, pp. 753 - 756.

Eronen A. 2001. Automatic Musical Instrument Recognition. Master of science thesis, Tampere University of Technology, Department of Information Technology.

Fletcher, H. 1934. Loudness, Pitch and Timber of Musical Tones and their Relations to the Intensity, the Frequency and the Overtone Structure. JASA, Vol. 6. No. 2, pp. 59 - 69.

Fraser, A., Fujinaga I. 1999. Toward Real-time Recognition of Acoustic Musical Instruments. ICMC 1999.

Fujinaga, I., Moore, S., Sullivan, D. 1998. Implementation of Exemplar-Based Learning Model for Music Cognition. Proceedings of the International Conference on Music Perception and Cognition, pp. 171-179.

Goldstein, E.B. 1989. Sensation and Perception, Brooks/Cole, California.

Golub, G. H., Van Loan, C 1996. The Singular Value Decomposition and Unitary Matrixes. §2.5.3 and 2.5.6 in Matrix Computations, 3rd ed, pp. 70-71 and 73, Baltimore, MD: Johns Hopkins University Press.

Gouyon, F., Pachet P., Delerue O. 2000. On the Use of Zero-Crossing Rate for an Application of Classification of Percussive Sounds. Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX), Verona, Italy.

Grey, J. M. 1975. Exploration of Musical Timbre. Stanford Univ. Dept. of Music Tech. Rep. STAN-M-2.

Grey J. M. 1977. Multidimensional Perceptual Scaling of Musical Timbre. Journal of the Acoustical Society of America Vol. 61, pp. 1270 - 1277.

Hall, D. E. 1991. Musical Acoustics: An Introduction (2nd edition). Belmont, CAL Wadsworth Publishing Company.

Handel, S. 1995. Timbre Perception and Auditory Object Identification. In Moore, B.C.J. (editor), Hearing. New York: Academic Press, pp. 425 - 463.

Handel, S. 1993. Listening: An Introduction to the Perception of Auditory Events. The MIT Press.

Haykin, S. 1994. Neural Networks: A Comprehensive Foundation. Macmillan.

Hebb, D. O. 1949. The Organization of Behavior. John Wiley, New York.

Helmholtz, H. L 1954. On the Sensation of Tone as a Physiological Basis for the Theory of Music (translation of original text 1877), New York: Dover Publications.

Herrera-Boyer, P., Amatriain X., Batlle E., Serra X. 2000. Towards Instrument Segmentation for Music Content Description: a Critical Review of Instrument Classification Techniques. International Symposium on Music Information Retrieval MUSIC IR 2000.

Holland J. 1975. Adaptation In Natural and Artificial Systems. The University of Michigan Press, Ann Arbor.

Howard, D. M., Angus, J. 2001. Acoustics and Psychoacoustics. Oxford, Boston : Focal Press.

Huang, L., Shimizu A., Kobatake H. 2002. Face Detection Using a Modified Radial Basis Function Neural Network. Proceedings of the 16th International Conference on Pattern Recognition, Vol. 2, pp. 342 - 345.

Iverson, P., C. L. Krumhansl 1993. Isolating the Dynamic Attributes of Musical Timbre. JASA 94(5).

Jensen, K. 2001. Workshop on Current Research Directions in Computer Music. Barcelona, Nov 15-16-17, 2001, Audiovisual Institute, Pompeu Fabra University, MOSART 2001.

Jensen, K., Arnspang, J. 1999. Binary Decision Tree Classification of Musical Sounds. In Proceedings of the 1999 International Computer Music Conference.

Jonston, J. D. 1988. Transform Coding of Audio Signals Using Perceptual Noise Criteria. IEEE on Selected Areas in Communication, Vol. 6, pp. 314 - 323.

Kendall, R. A. 1986. The Role of Acoustic Signal Partitions in Listener Categorization of Musical Phrases. Music Perception. Vol. 4, No. 2, pp. 185 - 214.

Kendall, R. A., Carterette E. C. 1996. Difference Threshold for Timbre Related to Spectral Centroid. Proceedings of the 4th International Conference on Music Perception and Cognition, pp. 91-5.

Klingholz, F. 1987. The Measurement of the Signal-to-noise Ratio (SNR) in Continuous Speech. Speech Communication 6.

Kovacevic, D., Loncaric S. 1997. Radial Basis Function-based Image Segmentation using a Receptive Field. Proceedings of the Tenth Annual IEEE Symposium on Computer-Based Medical Systems, pp. 126 - 130.

Krimphoff, J., McAdams S., Winsberg S. 1994. Characterisation du Timbre des Sons Complexes. II Analyses acoustiques et quantification psychophysique. Journal de Physique IV, Vol. 4, pp. 625 - 628.

Krumhansl, C.L. (1989). Why is Musical Timbre So Hard to Understand? In J. Nielzen & O. Olsson (Eds.) Structure and Electroacoustic Sound and Music (pp.43-53). Amsterdam: Elsevier (Excerpta Medica 846).

Krumhansl, C. L., Iverson, P. 1992. Perceptual Interactions Between Musical Pitch and Timbre. JASA, 18(30), pp. 739 - 751.

Kumar M., Srinivas D. 2001. Unsupervised Image Classification by Radical Basis Functions Neural Networks (RBFNN). 22nd Asian Conference on Remote Sensing.

Kung, S. Y. 1993. Digital Neural Networks, Prentice Hall, Englewood Cliffs, N. J.

Lakatos, S., G. Scavone, Cook , P. 2000. Obtaining Perceptual Spaces for Large Numbers of Complex Sounds: Sensory, Cognitive, and Decisional Constraints. In C. Bonnet (Ed.), Proceedings of the Sixteenth Annual Meeting of the International Psychophysics Society, pp. 245 - 250.

Lakatos, S. 2000. A Common Perceptual Space for Harmonic and Percussive Timbres. Perception & Psychophysics 62(2), pp. 1426 - 1439.

LeCun, Y., Boser B., Denker J. S., Henderson D., Howard R. E., Hubbard W., Jackel L. D. 1989. Backpropagation Applied to Handwritten Zip Code Recognition. Neural Computation, 1(4).

Levarie, S., Levy E. 1981. Tone: A Study in Musical Acoustics. 2nd edition, Greenwood Publishing Group.

Lichte, W. H 1941. Attributes of Complex Tones. Journal of Experimental Psychology, 28, pp. 455 - 480.

Licklider, J. C. R. 1951. Basic Correlates of the Auditory Stimulus. (In Handbook of Experimental Psychology, S.S. Stevens ed.) New York: Wiley.

Lippmann, R.P. 1987. An Introduction to Computing with Neural Nets, IEEE ASSP Magazine. 4: pp. 4-22, p.153.

Liu H., Motoda, H. 1998. Feature Selection for Knowledge Discovery and Data Mining. Kluwer Academic.

Luce, D. 1967. Physical Correlates of Brass-instrument Tones. JASA, 42, pp. 1232 - 1243.

Mak, M.W., Allen W. G., Sexton G. G. 1993. Speaker Identification using Radial Basis Functions. The 3rd IEEE Int. Conf. on Artificial Neural Networks, pp. 138 - 142.

Mak, M.W. 1996. Text-Independent Speaker Verification Over a Telephone Network by Radial Basis Function Networks. Proceedings of the International Symposium of Multi-Technology Information Processing, pp. 145 -150.

Martin, K., Kim Y. 1998. Musical Instrument Identification: A Pattern-Recognition Approach. 136th meeting of the JASA.

Martin, K. 1999. Sound-Source Recognition: A Theory and Computational Model. Ph.D. Dissertation, MIT.

McAdams, S. 1993. Recognition of Sound Sources and Events, in Thinking in Sound: The Cognitive Psychology of Human Audition. S. McAdams and E. Bigand (Eds.) Oxford: Oxford University Press, pp. 146 - 198.

McAdams, S., J. W. Beauchamp, S. Meneguzzi 1999. Discrimination of Musical Instruments Sounds Resynthesized with Simplified Spectrotemporal Parameters, JASA 104(2).

McAdams, S., Winsberg, S., Donnadieu, S., De Soete, G., and Krimphoff, J. 1995. Perceptual Scaling of Synthesized Musical Timbres: Common Dimensions, Specificities, and Latent Subject Classes, Psychological Research 58, 177 - 192.

McLachlan, G. J. 1992. Discriminant Analysis and Statistical Pattern Recognition. New York, NY: Wiley Interscience.

McCulloch, W., Pitts, W. 1943. A Logical Calculus of the Ideas Immanent in Nervous Activity. Bulletin of Mathematical Biophysics, 7: pp. 115 - 133.

Meng, J. E., Shiqian W., Juewi L., Hock L. T. 2002. Face Recognition With Radial Basis Function (RBF) Neural Networks. IEEE transactions on neural networks. Vol 13, No 3.

Minsky, M., Papert S.1969. Perceptrons, An introduction to Computational Geometry. MIT press, expanded edition.

Misdariis, N. , Smith B. K., Pressnitzer D., Susini P., McAdams S. 1998. Validation of a Multidimensional Distance Model for Perceptual Dissimilarities among Musical Timbres. 16th International Congress on Acoustics and 135th Meeting Acoustical Society of America, Seattle, Washington.

Moody, J., Darken C. 1989. Fast Learning in Networks of Locally Tuned Processing Units. Neural Computation 1, pp. 281 - 294.

Moorer, J. A.1975. On the Segmentation and Analysis of Continuous Musical Sound by Digital Computer," Doctoral Dissertation, Department of Computer Science, Stanford University.

Niranjan, M., Fallside F. 1990. Neural Networks and Radial Basis Functions in Classifying Static Speech Patterns. Computer Speech and Language 4, pp. 275 - 289.

Ohm, G. S. 1843. Über die Definitionen des Tones, Nebst Daran Geknüpfter Theorie der Sirene und Ähnlicher Tonbildener Vorrichtungen. Ann. Phys. Chem. 59, pp. 513 - 565.

ORL: Face Identification using Support Vector Machines. ESANN 1999 proceedings - European Symposium on Artificial Neural Networks Bruges (Belgium), ISBN 2-600049-9-X, pp. 195 - 200.

Osgood, C. E., Suci, G., Tannenbaum, P 1957. The Measurement of Meaning. Chicago: University of Chicago Press.

Patterson, R. D., 1987. A pulse Ribbon Model of Monaural Phase Perception. J. Acoust. Soc. Am. 82, pp. 1560 - 1586.

Park, J., Sandberg J. W. 1991. Universal Approximation Using Radial Basis Functions Network. Neural Computation, vol. 3, pp. 246 - 257.

Park T. H. 2000. Salient Feature Extraction of Musical Signals. Master's Thesis. Dartmouth College, Electro-acoustic Music Program.

Parker D. 1985. Learning Logic. Technical Report TR-47, Center for Computational Research in Economics and Management Science, MIT, Cambridge, pp. 99 - 154.

Plomp, R., Steeneken H. 1969. Effect of Phase on the Timbre of Complex Tones. J. Acoust. Soc. Am., 46: pp. 409 - 421.

Plomp, R. 1970. Timbre as a Multidimensional Attribute of Complex Tones. In R. Plomp & G.1 Smoorenburg (eds.), Frequency Analysis and Periodicity Detection in Hearing, Leiden: Sijthoff, pp. 397 - 414.

Plomp, R. 1976. Aspects of Tone Sensation. A Psychophysical study. New York: Academic Press.

Poggio, T., Girosi, F. 1990. Networks for Approximation and Learning. Proceedings of the IEEE, vol 78, Issue 9, pp. 1481 - 1497.

Pollard, H. F., Jansson, E. V. 1982. A Tristimulus Method for the Specification of Musical Timbre. Acustica, vol. 51. pp. 162 - 171.

Puterbaugh, J. D. 1999. Between a Place and Some Location, A View of Timbre through Auditory Models and Sonopoietic Space. Ph. D. Dissertation. Princeton University Music Department.

Rasch, R. and Plomp, R. 1982. The Perception of Musical Tones. In Deutsch, D. (Ed.) Psychology of music. Academic Press: New York.

Mathews, M. and J.-C. Risset. 1969. “Analysis of Instrument Tones.” Physics Today 22(2), pp. 23-30.

Roads, C. 1989. The Music Machine. Selected Readings from Computer Music Journal. Edited by Curtis Roads. The MIT Press, Massachusetts. London, England

Roederer, G. 1979. Introduction to the Physics and Psychophysics of Music. Heidelberg Springer Verlag.

Rossignal, S., J. Soumagne, X. Rodet 2000. Automatic Characterization of Musical Signals: Feature Extraction and Temporal Segmentation of Acoustic Signals. Journal of New Music Research.

Rossignol, S., Rodet, X., Soumagne, J., Collette, J.-L.,Depalle, P., 1999. Feature Extraction and Temporal Segmentation of Acoustic Signals. Journal of New Music Research, Vol. 28 (4).

Röbel, A. 1995. RBF Networks for Synthesis of Speech and Music Signals. In 3rd Workshop Fuzzy-Neuro Systeme '95 (pp. 165-172). Bonn: Deutsche Gesellschaft für Informatik E.V., Gesellschaft für Informatik.

Rumelhart, D.E., McClelland, J.L, & the PDP Research Group 1986. Parallel Distributed Processing: Exploration in the Microstructure of Cognition, Vol. 2 (Cambridge, Mass.: The MIT Press).

Saldhana , E. L., Corso, J. F. 1964. Timbre Cues and the Identification of Musical Instruments. JASA, Vol. 36, No. 11, pp. 2021 - 2026.

Sato K., Shah S., Aggarwal J. K 1998. Partial Face Recognition using Radial Basis Function Networks. Proceedings of the 3rd International Conference on Automatic Face and Gesture Recognition, pp.288 - 293.

Savoji, M. H. 1989. A Robust Algorithm for Accurate Endpointing of Speech. Speech Commin, Vol.8, pp. 45-60.

Scheirer, E., Slaney M. 1997. Proceedings of the 1997 International Conference on Acoustics, Speech, and Signal Processing (ICASSP).

Scholes, P. A. 1970. The Oxford Companion to Music. London: Oxford University Press.

Schouten, J.F. 1968. The Perception of Timbre. In Reports of 6th International Congress on Acoustics, Tokyo, Japan.

Seashore, C.E. 1967. Psychology of Music. (Originally published by McGraw-Hill in 1938, and reprinted) Dover Publications.

Sekey, A., Hanson, B. A 1984. Improved 1-bark Bandwidth Auditory Filter. JASA Vol. 75, No. 6.

Slawson, A. W. 1985. Sound Color. Berkeley: University of California Press.

Srinivasan, A., Sullivan, D., Fujinaga I. 2002. Recognition of Isolated Instrument Tones by Conservatory Students. Proceedings of the 7th ICMPC, poster presentation.

Strong, W. J. 1963. Synthesis and Recognition Characteristics of Wind Instrument Tones. Ph.D. Thesis, MIT.

Terhardt, E. 1979. On the Perception of Spectral Information in Speech. Hearing Mechanisms and Speech, ed. by O. Creutzfeld, H. Scheich & C. Schreiner, 281-91.
Berlin: Springer.

Thomaz, C.E., Feitosa, R.Q., Veiga, A. 1998. Design of Radial Basis Function Network as Classifier in Face Recognition Using Eigenfaces Proceedings of 5th Brazilian Symposium on Neural Networks, pp. 118 -123.

Torgerson. W. S. 1952. Psychometrika. 17. pp. 401 - 419.

Traunmüller, H. 1990. Analytical Expressions for the Tonotopic Sensory Scale. Journal of Acoustical Society of America. 88: pp. 97 - 100.

Tucker S. 2001. Auditory Analysis of Sonar Signals. Ph.D. Transfer Report. Department of Computer Science. University of Sheffield.

Tzanetakis, G., Cook P., 2001. Automatic Musical Genre Classification of Audio Signals. Proceedings International Symposium for Audio Information Retrieval (ISMIR 2001) .

Wang, S., Sekey A., Gersho A. 1992. An Objective Measure for Predicting Subjective Quality of Speech Coders. IEEE Journal on Selected Areas in Communication, vol. SAC-10, pp. 819 - 829.

Werbos, P. 1974. Beyond Regression: New Tools for Predictions and Analysis in the Behavioral Science. PhD Thesis, Harvard University.

Widrow, B., and Hoff, M. E. 1960. Adaptive Switching Circuits. Institute of Radio Engineers, Western Electronic Show and Convention, Convention Record, part 4, pp. 96-104.

Yantorno, R. E. 2000. A Study of Spectra Autocorrelation Peak Valley Ratio (SAPVR) as a Method for Identification of Usable Speech and Detection of Co-channel Speech. Final Report for: Summer research faculty program. Speech Processing Lab, EE&CE, Temple University.

Zwicker, E., Terhardt E. 1980. Analytical Expressions for Critical-band Rate and Critical Bandwidth as a Function of Frequency. Journal of the Acoustical Society of America 68. pp. 1523 - 1525.