Musical Timbre Analysis

HIROKO MORIOKA

MSc by Research in Pattern Analysis and Neural Networks

ASTON UNIVERSITY

September 2003

Acknowledgements

First, I would like to thank my supervisor Dr. Ian Nabney, for his continuous support and his time throughout the production of this thesis. This thesis could not have been possible without his guidance. Thank you!!

I would also like to thank all the MSc students, Rémi Barillec, Benoît Fabre, Frédéric Lê, Benoît Maroillez, Jody Miotke, Evangelos Sagianos and all the people from NCRG for their support and encouragement, especially Mr. Alex Brulo for his support for IT problems and some ideas towards this project, Ms. Vicky Bond for helping me with many issues, and Prof. David Lowe for his advice and support.

Finally, I would like to thank my family for supporting me throughout my studies.

ASTON UNIVERSITY

Musical Timbre Analysis

Hiroko Morioka

MSc by Research in Pattern Analysis and Neural Networks, 2003

Thesis Summary

Automatic music analysis has long been attempted, yet it is still far from achieving practically useful results. Musical audio signal consists of perceptual features such as pitch, loudness, duration and timbre, and structural aspects features such as harmony, melody, rhythm, and tempo. Complexity analysis is performed on segments of music in order to better understand the underlying generator of signals, then Independent Component Analysis was used in an attempt to extract such underlying sources. This is proved to be unsuccessful, so we investigated an alternative method of feature extraction. A simple harmonic model was used to model musical signals and an optimisation method for the parameters of the model was discussed.

Keywords: Pattern analysis, Music recognition, Independent Component Analysis, Complexity analysis, Harmonic analysis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The advance of the Internet has enabled us to handle huge amounts of data, and with the expansion of the Internet community, multimedia content, including audio data, is growing exponentially. There have been significant improvements in search engines to search and to explore the information available on the Internet, yet it remains difficult to search through the actual information content of multimedia data automatically.

With the support of the continuous development of different types of instruments and technology, styles of music have evolved tremendously in a multitude of genres. When we search for music, we are often interested in a particular style of music and genre is not a rigid classification. For example in jazz music, a piece may sound very like classical music, and another may sound very like a hip hop piece. This is because of their similarity in the type of instruments they use and how they are played. When music is played, a listener perceives the atmosphere and mood created, and looks for distinctive features in the piece to help identify the style. Such features can be explained in terms of components of music and how they are organised.

There are four perceptual components in a musical note; *pitch, loudness, duration* and *timbre* [28]. Physical properties of pitch, loudness and duration are better understood than timbre. Pitch is closely related to the fundamental frequency, which is the lowest frequency in harmonic vibration and it is also referred to as the 1st harmonic of the note or f_0 (Formant 0). Loudness of a sound can be explained by its intensity, which is proportional to the square of the amplitude of a sound played, and duration simply is the temporal duration of a tone. The definition of timbre is more vague. 'Timbre' refers to a quality of sound, or 'colour' of sound. It allows us to distinguish between the same note played on musical instruments of different types. The timbre of music has an important role in differentiating instruments, and tone is a key component of the process of distinguishing musical styles.

The structural aspects of music, such as harmony, melody, rhythm, and tempo are another important issue. Harmony is a relationship between tones played at the same time (e.g. chord and triad), while melody is the relationship between tones played one after the other. Rhythm is an important element in melody, it affects the progression of harmony, and many dance movements in classical music have a characteristic rhythmic foundation. Tempo is a characteristic rate or rhythm of activity.

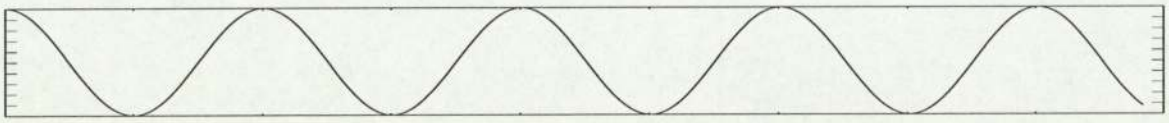

Figure 1.1: Pure sine wave of 440Hz.

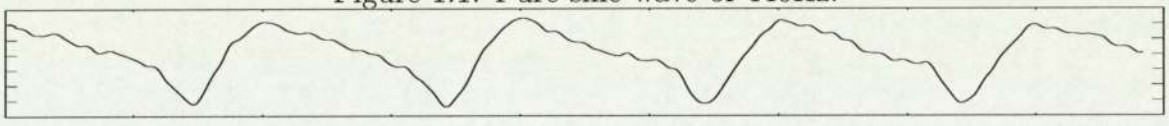

Figure 1.2: Sound wave of Piano at 440Hz.

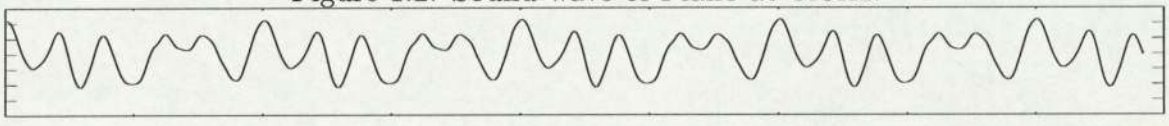

Figure 1.3: Sound wave of Violin at 440Hz.

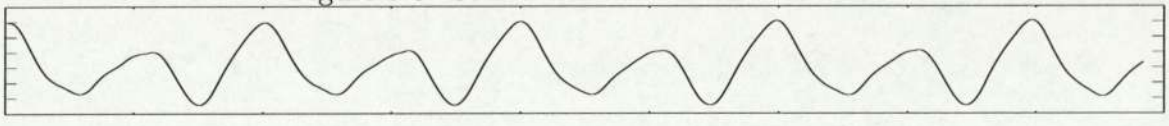

Figure 1.4: Sound wave of flute at 440Hz.

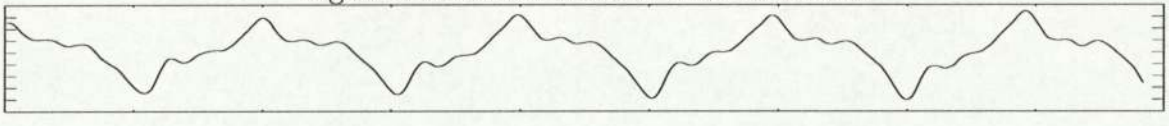

Figure 1.5: Sound wave of Clarinet at 440Hz.

In Figure 1.1, a pure sine wave at 440Hz (orchestral A or A4) is sampled for 0.01 seconds at sampling rate of 44100Hz. In Figures 1.2, 1.3, 1.4 and 1.5 respectively, sound waves of piano, violin, flute and clarinet are shown. Although the shape of waves differ, it is clear that the basic pattern which repeats is unique to each instruments. In these graphs, it is not hard to imagine that it contains vibrations of higher frequencies in addition to the fundamental frequency.

In Figures 1.6, 1.7, 1.8 and 1.9, the frequency spectrum of the signals above are shown. It is clear from the plots, that each note played by the instrument consists of partials or the harmonics of the fundamental frequency, which are integer multiples of the fundamental, and different relative intensities of such harmonics give a unique colour of sound. The series of those harmonics of a note is called a *harmonic series*. An example of harmonic series starting arbitrarily on C2 is shown in Figure **??**. Those figures show that there are many more harmonics observed on the frequency spectrum of the piano, while the violin contains more distinctive harmonics and the flute has stronger peaks at lower harmonics. There are very distinctive characteristics shown on the frequency spectrum of clarinet, where much stronger peaks at odd harmonics are recorded. The study [7] also shows the characteristics in the intensity of partials and the anharmonicity of the notes played by instruments. Thus the tone of the notes are produced mainly by the harmonics existing within the signal.

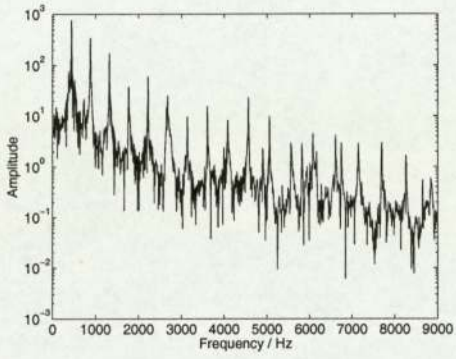

Figure 1.6: Spectrum of piano played for 0.1 second at 440Hz.

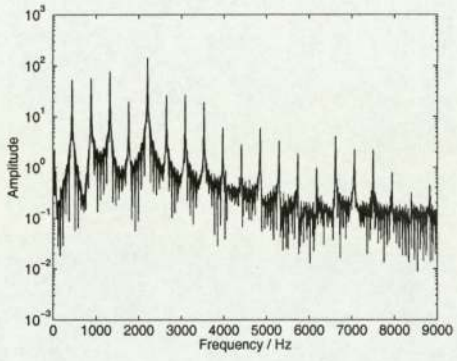

Figure 1.7: Spectrum of violin played for 0.1 second at 440Hz.

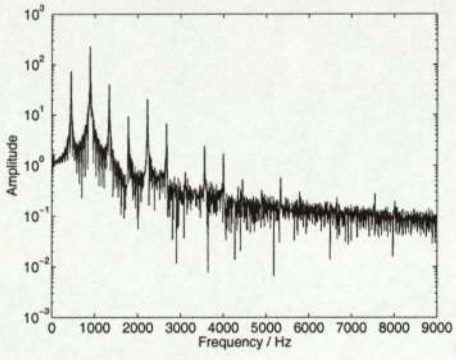

Figure 1.8: Spectrum of flute played for 0.1 second at 440Hz.

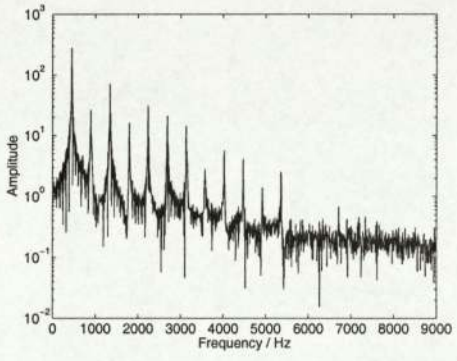

Figure 1.9: Spectrum of clarinet played for 0.1 second at 440Hz.

The piano and the violin are instruments whose tones are produced by strings. The vibration of the strings travels through the air and reaches our ear when we recognise it as a musical sound. Violin strings are vibrated by the action of bow on the string, or simply plucking the string, in which case it clearly produces different timbre. When it is played, the body of a violin acts as a medium for the string to transmit its vibration to the air so that it is audible to the ear. Thus the quality of sound produced by violin depends not only on the resonance of the string itself, but also how those vibrations are transmitted through the bridge to the body of the violin and thence to the air. Benade's study on resonance curves for a violin suggests that the main air resonance would enhance the D string and the main wood resonance would enhance the A string [5].

Piano produces its sound by striking three strings set on the bridge by the pin, that are used to alter the tension on the string. The three strings are typically tuned so the strings for each note covers spreads up to 8cents around the note[1]. This slight detuning of the strings in piano is proven to give longer lasting notes compared to those tuned exactly together is [5].

Both flute and clarinet are woodwind instruments. Woodwind instruments make use of an air column whose natural frequencies are properly arranged prior to playing. A

[1]A cent is a unit that is 1/100 of a tempered semi-tone [1].

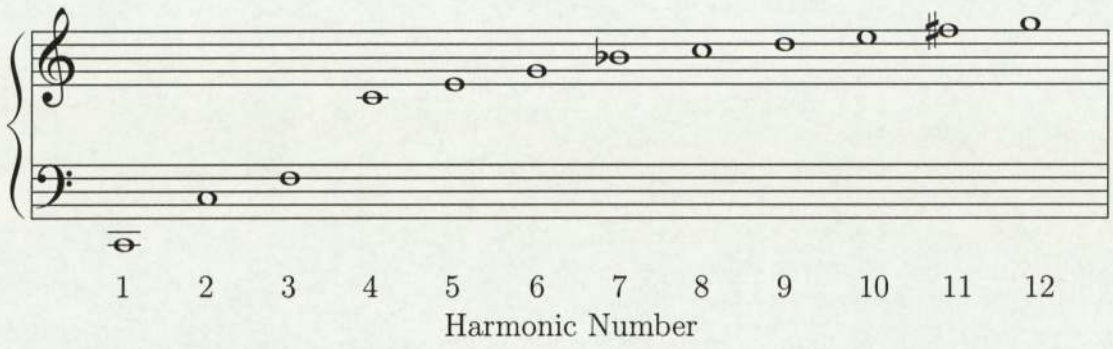

Figure 1.10: The first twelve notes of the harmonic series.

clarinet acts as a closed air column instrument with a mouthpiece acoustically acting as a closed end and it does not produce even harmonics. On the other hand, the flute is made in the form of an open cylindrical air column, where the air is blown into from the mouth hole. When flute is played, a skilled player can alter the strength of even and odd harmonics to produce tones that gives different impressions of the note. However, in most cases physical properties of the flute means that it produces a tone whose harmonics are precisely at integer multiples, although in some cases, this harmonicity balance may intentionally be broken by the technique that is used to produce a popping sound in which exact harmonicity would produce tone that may sound 'odd' [5].

1.1 Motivation

Recently, with the aid of computers, the area of automatic music analysis has attracted many people such as musicologists, computer scientists, and music lovers. There are many active researchers internationally and numbers of commercial software system have been developed for automatic retrieval of musical data using templates for this musical data. Because of the need for a wide range of expertise in music analysis, events organised by consortia of research groups, such as the Digital Music Research Network and conferences such as International Conference on Music Information Retrieval (ISMIR), enable us to view the current state of art in Music Information Retrieval (MIR).

There are numerous practical approaches to the problem, and one of the possible applications of the field of study is a content-based searching system, which enables users to search for a sound signal by the actual data content in the waveform. As mentioned earlier, there are many features in the musical signals, which should be considered when identifying the 'style' of music. Timbre is one of the most fundamental components of the music, and it is also related to the instrumental content in the piece. Providing a robust semantic labelling system of such components will help us to understand better the contents of the music signals.

The development of MPEG-7 has broadened the possibility of information retrieval from multimedia. It is a standard for describing the multimedia content data using XML developed by Motion Pictures Expert Group. There are several 'description

schemes' (DSs) available in MPEG-7. The content DS has two main levels: *syntactic* and *semantic* [19] [27] [30]. Such description schemes can be broken down to several lower levels of information to form hierarchical structure. Each lower-level DS block may contain structural descriptions of the multimedia content, such as the information related to the tonal structure of music, and/or the colour found in the image captured, and also the physical and logical aspects of this information. Such low level descriptions are used as building blocks for mid-level and high-level descriptions to enable easy access to the multimedia informaation needed. MPEG-7 addresses different types of applications, therefore it gives flexible and extensible framework to describe different types of multimedia data. This standard given to multimedia data means that given the data that describes the multimedia data stored in the DS block, it enables users to easily search the content of the multimedia data. This leads us to explore the possibility of describing multimedia data in a most efficient and meaningful way.

1.2 Approach

Musical signals consist of a mixture of one or more sound sources producing sound waveforms. Each sound source consists of its own internal dynamic, which is temporally well structured. Our aim is to characterise such sound sources by analysing the structure of musical signals. First, complexity analysis was performed on segments of music in order to better understand the underlying generator of signals, then Independent Component Analysis was used in an attempt to extract such underlying generators. This proved to be unsuccessful, so we investigated an alternative method of feature extraction. A simple harmonic model was used to model musical signals and an optimisation method for the parameters of the model will be discussed.

In Chapter 2, we review research done on Audio Information Retrieval and related areas of interest. In Chapter 3, some methods to analyse the complexity of the musical signal are reviewed and applied. Extraction of underlying sources of the musical signal has also been attempted. In Chapter 4, we describe the simple harmonic model used in [14], and suggest a method for optimisation of the parameters. Finally in Chapter 5, there is a summary of the study, and possible directions for future work are suggested.

Chapter 2

Literature Survey

Many studies have been performed on audio source recognition in the past, and the problem of Audio Information Retrieval (AIR) is widely recognised [16]. When solving such problems, there are two key tasks; parameterising features of the music and managing such feature information for retrieval. Practically, for the classification of musical instruments within a recorded musical signal, segmentation of the data should be performed with care. In a typical piece of music, many parts contain complex mixes of sound sources (i.e. musical instruments). This suggests the need for appropriate segmentation method that is capable of extracting a single source within the musical data prior to the harmonic analysis for the recognition of timbre. In this project, we will perform sound source separation followed by the structural analysis of harmonics produced by each sound source. In this chapter, we provide a brief survey of the relevant literature.

2.1 Sound Source Separation

In an environment where a mixture of sound sources exist (e.g. at a party, or on the train), a human listener has the ability to identify and focus on a sound source of interest (e.g. a friend's voice). This problem is called the 'Cocktail Party Problem' [11], and it was one of the earliest studies in Sound Source Separation. Since then, researchers have been attempting to build a system that is capable of separation of sound sources making use of the cocktail party effect. There have been two main ways of solving this problem. The first approach is Blind Source Separation (BSS), where no assumption is made about the sources except that they are statistically independent. The second is the model-based approach, where the model can be anything from the harmonicity of sounds to rhythmic complexity and onset/offset detection. As one example of the model based approach, a pattern matching algorithm was successfully used in conjunction with the recorded sound database to separate percussive beats from music in [18]. This approach showed good results providing there are enough model templates stored within the system, and thus it was computationally expensive in the general case.

Blind Source Separation was first used by Jutten and Herault [24], and Comon [13]. Comon further studied the problem of ICA (Independent Component Analysis) [12], in which he proposed cost functions related to the minimisation of mutual informa-

tion between sources. In this approach, an estimate of mutual information is needed, which involves approximating the pdf using polynomial expansions of Gram-Charlier or Edgeworth. This leads to the use of higher-order cumulants, which are highly sensitive to outliers. This problem has led to the development of other approximations of entropy. In [20], Hyvärinen suggested using maximum entropy together with a 'measuring' function, which is the objective function introduced to approximate differential entropy. This method has shown more accuracy than conventional cumulant-based approximations used in [12].

On the other hand, the use of the infomax principle first advocated by Linsker [25] became a popular method for blind source separation in linear systems. This idea was developed from the redundancy reduction across the input signals as coding strategy in neurons denoted by Barlow [2] [3]. The aim of this process is to maximise the mutual information between the inputs and outputs of a neural network. To achieve this, each neuron should be encoded as statistically independent from other neurons. Bell and Sejnowski derived stochastic gradient learning rules for the maximisation of mutual information following the principle described by Laughlin in 1981 [4].

For most ICA algorithms, the number of source signals which can be extracted is limited to the number of recordings N provided. One of the problems in applying this for the separation of musical audio data is that in most cases the musical data is either mono or stereo recordings. This has led to the development of an algorithm to enable the separation of the single recorded signal into underlying sources. In [23], an ICA method of extracting multi-source brain activity from a single EM channel was described. This was performed by viewing the EEG activity as a dynamical system, and build an embedding matrix from a series of delay vectors. This embedding matrix was used in ICA and desired signal was successfully extracted from a single channel signal with numbers of underlying sources.

2.2 Parameterisation of feature space

The term 'feature' has many different interpretations in musical signal analysis. It ranges from surface features such as timbre, pitch, tempo and loudness to more contextual features such as melody and rhythm. As mentioned, the timbre of music strongly relates to the frequency content of a sound signal. There are many methods for the spectral analysis of a signal, the simplest and the most popular of which is the Fourier Transform. However, psychophysical studies on human perception has led to the development of the warped frequency scale, in which it follows the 'subjective pitch' of pure tones [32]. One of the common warped frequency scale used to model the features of musical signal is Mel-Frequency Cepstrum Coefficients (MFCC). The Mel-frequency is a subjective pitch proposed by Stevens, Volkman and Newman in 1937. MFCC provides a reduction in the number of spectral features compared with the Fast Fourier Transform. The use of MFCC for music modelling was shown to be applicable [26] and it is now widely used in music signal processing community. Brown has also used MFCC for the parameterisation of features for the identification of the various woodwind instruments played in solo, and using Gaussian mixture models, it showed bet-

ter results in identification of instruments than the untrained human listener [8] [9].

Another popular method to extract the features of the signal is Singular Value Decomposition (SVD). The method is closely related to Principal Component Analysis (PCA), and in the context of signal processing, it characterises the time series by its most relevant components in a delay embedding space (a more detailed explanation can be found in Section 3.2.1). SVD is widely used to reduce the dimensionality of data whilst retaining the maximum variation. The use of singular value spectra is popular as a complexity measure for EEG signal[23] [33], and is often used as a subspace analysis tool and/or pre-processing tool for BSS in biomedical and audio signal processing communities [10] [39].

A recorded musical signal, which was originally produced by several instruments, is a product of several different underlying sources (four for a quartet). Thus the system dimensionality is also related to the complexity of each component signal, while it is restricted to lower dimensional manifolds by the recording process. Although it is hard to define exactly the 'dimension' of a recorded signal, the dimensionality of the data is closely related to the complexity of the data.

In this thesis, a recorded musical signal is first examined for its complexity, and an embedding matrix will be constructed using the results obtained from the complexity analysis. The ICA algorithm will then be applied to the embedding matrix in order to extract information related to the underlying sources of the recorded musical signal.

2.3 Harmonic analysis

As briefly discussed in the introduction, the tone structure of a note can be represented as a sum of a fundamental frequency and its partials. Now, sound is a vibration of the air, and musical instruments are the creators of this air vibration. When the air vibration reaches our ear, it translates the vibration of the air into a electrical signal that we perceive as a sound. Benade recorded and analysed a sound produced by a nylon-stringed guitar [5] to show the harmonics of each sound produced. The results he obtained are shown in Table 2.1. The result clearly suggests that the harmonics of a fundamental pitch are at frequencies which are *approximately* integer multiples.

Combining this knowledge of musical acoustics and the fundamental theory of signal processing, *additive synthesis* was one of the first attempts made in sound synthesis. Following the leading study in musical instruments tones by Risset and Mathews in 1969, many attempts have been made to model such tone structures as a summation of sine waves and cosine waves [35] [34]. This approach to the analysis of sound sources requires two steps, peak detection in frequency domain, and groupings of such peaks. It is easy to find local maxima in the magnitude spectrum, but some problems may arise as some of these peaks may not be related to the main tonal structure. In [34], Hidden Markov Models (HMM) were used to put detected peaks together as a group of partials, and the *additive synthesis* of sinusoidal and the residual noise method was proven to be successful for musical signal analysis and synthesis. However, this ap-

String Number	Pitch Name	Characteristic Harmonics: Lowest	2nd	3rd	4th	5th
1	E4	300	600.9	900.2	1200.0	1500.9
2	B3	300	599.2	900.0	1200.1	1500.0
3	G3	300	602.0	902.8	1204.6	1504.1
4	D3	300	600.6	900.0	1204.5	1508.2
5	A2	300	595.4	897.0	1198.1	1500.0
6	E2	300	603.7	900.0	1201.9	1500.0

Note: To aid comparison, the sounds of the strings have been transposed by variable-speed tape recorder to make the frequencies of their lowest component match.

Table 2.1: Measured values of Components of a Set of Guitar Strings.

proach relies heavily on the spectrum of the musical signal, as discussed further in [34], thus it is difficult to model fast transient in the musical signal (e.g. a musical signal created by piano consists of sounds created by attacking of the key and the following sustained note). This problem was overcome by introducing High Resolution Matching Pursuit (HRMP) together with a set of decomposition vectors called a dictionary.

2.4 Pitch Detection

Pitch, as discussed in the introduction, is a perceptual feature of an audio signal. There have been two main theories of how humans perceive pitch, *place* theory and *temporal* theory [28]. Place theory has two postulates. The first suggests that there is some sort of spectral analysis taking place in the inner ear, such that different frequencies excite different places along the basilar membrane (BM), and the second suggests that the pitch of a stimulus is related to the excitation pattern produced by that stimulus. Temporal theory suggests that the nerve firings tend to occur at a particular phase of the stimulation waveform, and thus intervals between successive neural impulses are used to approximate the period of the waveform. The theory could not work at very high frequencies, since phase locking does not occur for sinusoids with frequencies above about 5kHz [28]. On the other hand, a difficulty arises for complex tones when we fit the place theory. The complex tones will produce distribution of excitation with many maxima. This problem somewhat relates to the harmonic analysis problem of Rodet [34] mentioned in Section 2.3. In it, the problems have been overcome by applying statistical analysis on the peaks detected to resolve 'best-fitting' fundamental frequency, f_0. Thus the problem of pitch detection is equivalent to f_0 estimation.

Autocorrelation analysis is one of the simplest and commonly used techniques for pitch detection [31]. It measures the similarity of waveforms at different time intervals. Let us consider a simple periodic wave shown in Figure 2.1, in which autocorrelation function is shown in Figure 2.3. As can be seen from the plots, the similarity is exact at a time lag of zero, and as we increase the time lag to half of the period of the waveform, the correlation decreases to a minimum, since the waveform become totally out of phase with comparison to the original. Likewise, the correlation attains its maximum as the time lag increases to one period of the waveform. The problem arises

when we consider harmonically complex waveform. If we look at an example shown in Figure 2.2, in which autocorrelation function is shown in Figure 2.4, the correlation reaches its local maxima at around half of the period, as well as at one period of the waveform. This means that some form of peak detection algorithm that is able to distinguish between large and small peaks is needed to successfully apply autocorrelation function as a pitch estimator.

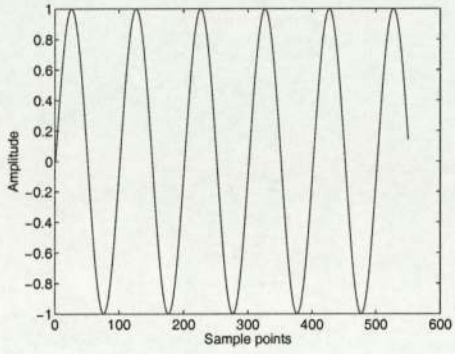

Figure 2.1: Example of a pure sine waveform.

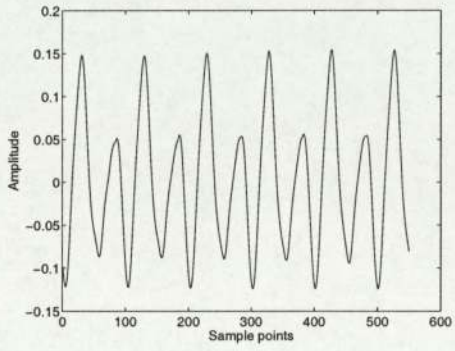

Figure 2.2: Example of flute waveform.

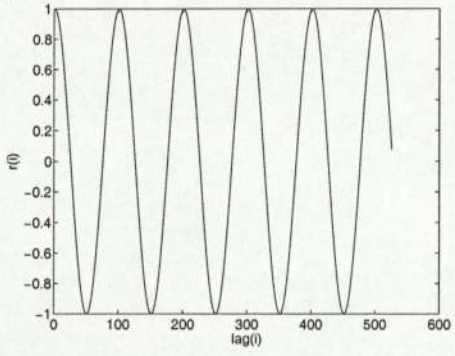

Figure 2.3: Autocorrelation function calculated from the waveform in Figure 2.1.

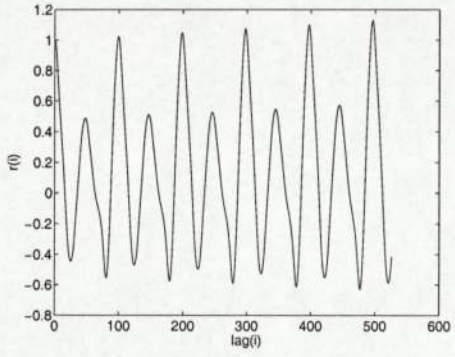

Figure 2.4: Autocorrelation function calculated from the waveform in Figure 2.2.

The YIN f_0 estimator was developed based on an autocorrelation algorithm [15], and it was developed as a pitch detection algorithm for a speech signal. While the approach of using a lagged window is similar, it uses the difference function in an attempt to minimise the difference between the waveform, instead of maximising the product. To reduce the occurrence of harmonic errors (f_0 being estimated as harmonics of true f_0), it employs a cumulative mean function to have less emphasis on higher harmonics. It has been tested on a database of a mixture of speech signals and used the signal of a laryngograph (an apparatus that measures electrical resistance between electrodes placed across the larynx) as a 'ground-truth' estimate to derive the error. It was shown to have 99% accuracy within 20% of the ground-truth estimate, 94% within 5%, and 60% within 1%. More details of the algorithm can be found in Section 4.3.2.

Other than the methods mentioned above, there are many pitch estimation methods that are based on the nonlinearity of human perception of pitch. As briefly discussed in Section 2.2, many studies have been performed in the psychology of pitch percep-

tion for both pure and complex tones. Some pitch extractors make use of such knowledge in human perception in an attempt to extract the pitch of various kinds of audio data including musical audio and speech audio. One example is the use of Lyon's cochlea model in Slaney's Auditory Toolbox [38], in which the input audio signal is filtered using a model of the human auditory system. It consists of series of filters that model the travelling pressure waves with Half Wave Rectifiers (HWR) to detect the energy in the signal and several stages of Automatic Gain Control (AGC) [36] [37].

Chapter 3

Signal Decomposition

In this chapter, we shall discuss a method for extracting underlying sources from a single channel recorded musical signal. A small sample taken from a music audio signal was analysed in terms of complexity, and the decomposition of such a signal into its components was attempted. First, a brief overview of Independent Components Analysis and related algorithms will be discussed in Section 3.1, followed by the complexity analysis of music in Section 3.2.1. Then the algorithm will be evaluated and results are shown in Section 3.3 and 3.4.

3.1 Sound Source Separation

As briefly discussed in Section 2.1, Independent Component Analysis (ICA) was developed as a tool for sound source separation. Because of the nature of this statistical method, it can sometimes be used to find 'interesting' features in multidimensional data. The algorithm looks for underlying factors/components from such data by minimising the statistical dependence between them. First, we shall look into the definition of ICA, followed by the definition of statistical independence. Then the FastICA algorithm described by Hyvärinen [21] will be discussed.

3.1.1 Definition of ICA

If we observe n random variable $\mathbf{x} = \{x_1, \ldots, x_n\}$, modelled as a linear combination of n latent random variables $\mathbf{s} = \{s_1, \ldots, s_n\}$, then,

$$\mathbf{x} = A\mathbf{s}, \tag{3.1}$$

where A is an $n \times n$ mixing matrix. If A is invertible, this model can be re-written using the constant weight matrix W:

$$\mathbf{s} = W\mathbf{x}, \tag{3.2}$$

so that the linear transformation of the observed variables can be obtained and generating variables extracted.

3.1.2 Independence and Uncorrelatedness

A collection of n random variables $x_1, x_2, \ldots, x_n$ are said to be independent if and only if:

$$p(x_1, \ldots, x_n) = \prod_{i=1}^{n} p(x_i). \tag{3.3}$$

In addition, independent variables have the following properties:

$$R_{x_i,x_j} = \varepsilon[x_i, x_j] = \varepsilon[x_i]\,\varepsilon[x_j] \qquad \text{for } i \neq j \tag{3.4}$$

$$C_{x_i,x_j} = \varepsilon\left[(x_i - m_{x_i})(x_j - m_{x_j})\right] = 0 \qquad \text{for } i \neq j \tag{3.5}$$

where R_{x_i,x_j} and C_{x_i,x_j} are the cross-correlation and covariance of x_i, x_j respectively, and $\varepsilon[\cdot]$ denotes expectation. If Equation 3.4 is true, random variables x_i and x_j $(i \neq j)$ are also said to be uncorrelated, but satisfying only Equation 3.4 does not mean they satisfy Equation 3.3, thus independence implies uncorrelatedness, but uncorrelatedness does not imply independence.

3.1.3 Whitening

Whitening is a transformation process to remove correlation of variables. As discussed in Section 3.1.2, uncorrelatedness does not imply independence, nevertheless, whitening is a useful pre-processing technique to be applied before ICA. The correlation of the components in vector $\mathbf{x}$ is removed by a linear transformation, so that we obtain a new 'whitened' vector $\tilde{\mathbf{x}}$.

$$\tilde{\mathbf{x}} = Q\mathbf{x}, \tag{3.6}$$

whose covariance matrix equals the identity matrix,

$$\varepsilon\left[\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T\right] = I. \tag{3.7}$$

Generally, a covariance matrix can be re-written in the form:

$$EDE^T = C,$$

where E is the orthogonal matrix of eigenvectors and D is the diagonal matrix of its eigenvalues. Now, if we apply the following transformation to $\mathbf{x}$;

$$\tilde{\mathbf{x}} = D^{-1/2}E\mathbf{x},$$

its covariance will become:

$$\begin{aligned} \varepsilon\left[\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T\right] &= D^{-1/2}E\,\varepsilon\left[\mathbf{x}\mathbf{x}^T\right]E^T D^{-1/2} \\ &= C^{-1/2}CC^{-1/2} \\ &= I. \end{aligned} \tag{3.8}$$

Let

$$Q = D^{-1/2}E,$$

then,

$$\tilde{\mathbf{x}} = Q\mathbf{x}.$$

The whitening matrix Q is by no means the only whitening matrix for x. Let us consider an orthogonal matrix R, then it is clear that any matrix RQ is also a whitening matrix. If

$$\tilde{\mathbf{x}} = RQ\mathbf{x},$$

then;

$$\begin{aligned} \varepsilon\left[\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T\right] &= RQDQ^TR^T \\ &= I. \end{aligned} \tag{3.9}$$

Now, from Equation 3.1, Equation 3.6 can be re-written as:

$$\tilde{\mathbf{x}} = RQA\mathbf{s} = \tilde{A}\mathbf{s}.$$

This can be re-written as:

$$\begin{aligned} \varepsilon\left[\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T\right] &= \tilde{A}\,\varepsilon\left[\mathbf{s}\mathbf{s}^T\right]\tilde{A}^T \\ &= \tilde{A}\tilde{A}^T \\ &= I. \end{aligned} \tag{3.10}$$

Since the new mixing matrix $\tilde{A}$ is also orthogonal, the search for the mixing matrix can be restricted to the space of orthogonal matrices with $n(n-1)/2$ degrees of freedom, while for an arbitrary matrix, n^2 parameters should be estimated. Since whitening is a simple procedure which reduces the complexity of the problem tremendously, it is a good idea to employ whitening as a pre-processing step of ICA.

3.1.4 Non-Gaussianity and Independence

The *central limit theorem* is one of the most important statistical results, which explains why the Gaussian distribution occurs so frequently in nature. It is the core of ICA and it states that the distribution of a sum of identically distributed independent random variables tends towards a Gaussian distribution as the number of random variables increases [22]. This means that, if we have a sum of two independent random variables, the distribution of the sum will be closer to Gaussian than either of the original random variables.

Now, consider the general ICA model to estimate $\mathbf{s}$ stated in Equation 3.2. To estimate one of the independent components, one can consider a linear combination of variables x_i. Let us denote this by $y = \mathbf{b}^T\mathbf{x} = \sum_i b_i x_i$, where $\mathbf{b}$ is a vector to be determined, and this can be re-written as: $y = \mathbf{b}^T A\mathbf{s}$. Thus y can also be expressed as linear combination of s_i, and let us define the coefficients $\mathbf{r}^T = \mathbf{b}^T A$. Then,

$$y = \mathbf{b}^T\mathbf{x} = \mathbf{r}^T\mathbf{s} = \sum_i r_i s_i.$$

As stated in *central limit theorem*, $y = \mathbf{r}^T\mathbf{s}$, a linear combination of s_i, is usually more Gaussian than any of individual s_i, and becomes least Gaussian when it equals one of

the s_i, because of its statistical independence. Obviously, in this case, only one of the elements of r_i is nonzero. Since $\mathbf{b}^T\mathbf{x} = \mathbf{r}^T\mathbf{s}$, we can pick the optimal $\mathbf{b}$ by monitoring the distribution of $\mathbf{b}^T\mathbf{x}$. Thus by finding the vector $\mathbf{b}$ which maximises the non-Gaussianity of $\mathbf{b}^T\mathbf{x}$, we could find one of the independent components [22].

3.1.5 Mutual Information

As discussed in Section 3.1.4, ICA looks for the vector or direction $\mathbf{b}$, so that the projection of the data $\mathbf{x}$ is most non-Gaussian. There are several different methods for estimating such a $\mathbf{b}$ and one of the most popular is to use the concept of mutual information. Mutual information is a measure of the relationship between members of a set of random variables. Mutual information can be defined as follows. First, let us consider the differential entropy H of a random vector $y = \{y_1, \ldots, y_n\}$ with density $g(.)$:

$$H(y) = -\int g(y) \log g(y)\, dy. \tag{3.11}$$

Differential entropy can be interpreted as a measure of randomness in the same way as entropy. If the random variable is concentrated in small intervals, then the differential entropy becomes small. Note that the differential entropy is a relative measure of randomness, and is at its maximum with a Gaussian distribution. This leads to the definition of negentropy J, which can be used as a measure of non-Gaussianity:

$$J(y) = H(y_{gauss}) - H(y), \tag{3.12}$$

where y_{gauss} is a Gaussian random vector with the same covariance matrix as y. By normalising differential entropy obtained in Equation 3.11, negentropy $J(y)$, would equal zero for a Gaussian variable, and always be non-negative. Knowing this, the mutual information I can be expressed as:

$$I(y_1, \ldots, y_n) = J(y) - \sum_i J(y_i). \tag{3.13}$$

Mutual information is a natural measure of the dependence between random variables, thus it can be applied for finding ICs by ICA. As shown in Equation 3.2, ICA of a random vector $\mathbf{x}$ is an invertible transformation to $\mathbf{s}$. Thus the matrix $\mathbf{W}$ is determined so that the mutual information of the transformed components s_i is minimised. Since negentropy is invariant under any linear invertible change of coordinates [12], the problem of finding a transformation $\mathbf{W}$ that minimises the mutual information can be interpreted as finding directions to project data $\mathbf{x}$ in which the negentropy is maximised, thus ICA is sometimes referred to as a form of projection pursuit.

3.1.6 Estimation of Negentropy

As mentioned in Section 3.1.5, the concept of negentropy is the key to extract independent sources from mixed signal, and it is very well justified by statistical theory as a measure of non-Gaussianity. However, estimation of negentropy can be a difficult task as it may require estimation of a pdf, for example using the polynomial expansion known as Edgeworth expansion or Gram-Charlier expansion [12]. This involves the use

of higher-order cumulants, and thus it is sensitive to outliers as their values may heavily depend on only a few erroneous observations. In [20], an approximation of differential entropy using a contrast function derived further from a conventional polynomial expansion method was shown. The result of the classical method of negentropy approximation using the polynomial density expansion is of the form:

$$J(y) \approx \frac{1}{12}\varepsilon\left[y^3\right]^2 + \frac{1}{48}kurt(y)^2. \tag{3.14}$$

This was generalised to use expectations of general nonquadratic function. In the simplest case, the new approximation of negentropy is of the form [21]:

$$J(y_i) \propto \left[\varepsilon\left[G(y)\right] - \varepsilon\left[G(v)\right]\right]^2, \tag{3.15}$$

where c is a constant, v is a Gaussian variable of zero mean and unit variance, and G is a contrast function, which practically can be any non-quadratic function. For symmetric variables, using cumulant based approximation in [12], $G(y_i) = y_i^4$.

As Equation 3.15 is generalised to use any non-quadratic function as a contrast function, there are many choices of $G(y)$ that can be used. In [21], they have tested three choices of $G(y)$:

$$G_1(y) = \frac{1}{a_1}\log\cosh(a_1 y), \qquad g_1(y) = \tanh(a_1 y), \tag{3.16}$$

$$G_2(y) = \frac{1}{a_2}\exp(-a_2 y^2/2), \quad g_2(y) = y\exp(-a_2 y^2/2), \tag{3.17}$$

$$G_3(y) = \frac{1}{4}y^4, \qquad g_3(y) = y^3, \tag{3.18}$$

where $1 \leq a_1 \leq 2$, $a_2 \approx 1$ are constants, and g_1, g_2 and g_3 are derivatives. The conclusion was that G_1 is a good general-purpose contrast function, while G_2 worked better when expected independent components are highly super-Gaussian, and G_3 (kurtosis) is suited for estimating sub-Gaussian independent components. Figures 3.1, 3.2 and 3.3 shows the distribution of piano, violin and flute played at 440Hz. This suggest that the sound wave of instruments are sub-Gaussian, thus using G_3 as a contrast function in order to extract such independent components from recorded signal would be an appropriate option.

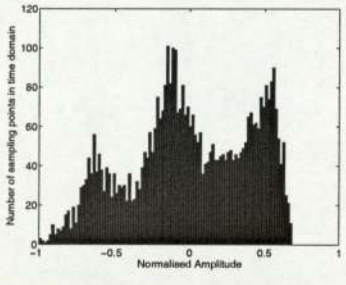

Figure 3.1: The distribution of piano played at 440Hz.

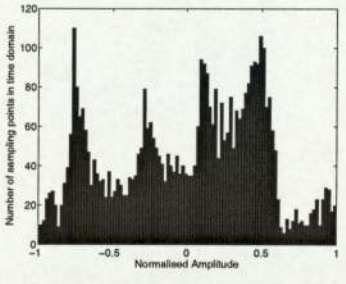

Figure 3.2: The distribution of violin played at 440Hz.

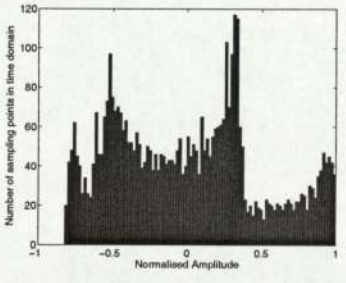

Figure 3.3: The distribution of flute played at 440Hz.

3.1.7 Fixed-point ICA Algorithm

Fixed-point algorithms, also known as FastICA were developed by Hyvärinen [21] for a more robust and computationally efficient algorithm, which can be used for ICA in practical analysis. The FastICA uses a fixed-point iteration scheme to find the direction, $\mathbf{w}$, such that the projection $\mathbf{w}^T\mathbf{x}$, maximises non-Gaussianity. In Section 3.1.6, the efficient approximation method of negentropy using contrast function was discussed. Using Equation 3.15, to find one independent component as $y_i = \mathbf{w}^T\mathbf{x}$, it gives a new objective function:

$$J_G(\mathbf{w}) = [\varepsilon\left[G(\mathbf{w}^T\mathbf{x})\right] - \varepsilon\left[G(v)\right]]^2, \tag{3.19}$$

where $\mathbf{w}$ is an m-dimensional weight vector constrained so that $\varepsilon\left[(\mathbf{w}^T\mathbf{x})^2\right] = 1$. By finding the optima of $\varepsilon\left[G(\mathbf{w}^T\mathbf{x})\right]$, one can maximise J_G.

According to Kuhn-Tucker conditions, the optima of $\varepsilon\left[G(\mathbf{w}^T\mathbf{x})\right]$, under the constraint $\varepsilon\left[G(\mathbf{w}^T\mathbf{x})^2\right] = \|\mathbf{w}\|^2 = 1$ are obtained at points where [21]:

$$\varepsilon\left[\mathbf{x}g(\mathbf{w}^T\mathbf{x})\right] - \beta\mathbf{w} = 0, \tag{3.20}$$

where β is a constant that can be evaluated to give $\beta = \varepsilon\left[\mathbf{w}_0^T\mathbf{x}g(\mathbf{w}_0^T\mathbf{x})\right]$, where $\mathbf{w}_0$ is the value of $\mathbf{w}$ at optimum and g denotes the derivative of G.

Let us denote the left hand side of Equation 3.20 by F, and using Newton's method, the Jacobian matrix $JF(\mathbf{w})$ is obtained:

$$JF(\mathbf{w}) = \varepsilon\left[\mathbf{x}\mathbf{x}^T g'(\mathbf{w}^T\mathbf{x})\right] - \beta\mathbf{I}.$$

Note that g' denotes the derivative of g. As the data is sphered, we can approximate;

$$\varepsilon\left[\mathbf{x}\mathbf{x}^T g'(\mathbf{w}^T\mathbf{x})\right] \approx \varepsilon\left[\mathbf{x}\mathbf{x}^T\right]\varepsilon\left[g'(\mathbf{w}^T\mathbf{x})\right] = \varepsilon\left[g'(\mathbf{w}^T\mathbf{x})\right]\mathbf{I}.$$

Thus the Jacobian matrix becomes diagonal, so it can easily be inverted. So we obtain the approximative Newton iteration:

$$\mathbf{w}^+ = \mathbf{w} - \frac{\varepsilon\left[\mathbf{x}g(\mathbf{w}^T\mathbf{x})\right] - \beta\mathbf{w}}{\varepsilon\left[g'(\mathbf{w}^T\mathbf{x})\right] - \beta}. \tag{3.21}$$

$\mathbf{w}^+$ obtained here is normalised for the stability, so the new value becomes $\mathbf{w}^* = \mathbf{w}^+/\|\mathbf{w}^+\|$. This can be simplified by multiplying both sides of 3.21 by $\beta + \varepsilon\left[g'(\mathbf{w}^T x)\right]$, and it gives the fixed-point algorithm defined in [21]:

$$\mathbf{w}^+ = \varepsilon\left[\mathbf{x}g(\mathbf{w}^T\mathbf{x})\right] - \varepsilon\left[g'(\mathbf{w}^T\mathbf{x})\right]\mathbf{w}, \tag{3.22}$$

with the normalisation $\mathbf{w}^* = \mathbf{w}^+/\|\mathbf{w}^+\|$ to give new value $\mathbf{w}^*$.

This process is repeated until the value of $\mathbf{w}$ and $\mathbf{w}^*$ converged to the same direction so that their dot product is equal to one. Note that because of the normalisation performed to obtain $\mathbf{w}^*$, y estimated will have unit variance.

To obtain the whole matrix $\mathbf{W}$, this process is repeated as follows. After estimating the

pth independent component $\mathbf{w}_p$, we run the algorithm shown in Equation 3.22 to estimate $\mathbf{w}_{p+1}$, and after every iteration step, subtract the 'projections' $\mathbf{w}_{p+1}^T \mathbf{w}_i \mathbf{w}_i$ for all $i = 1, \ldots, p$ estimated independent components from $\mathbf{w}_{p+1}$, and then renormalise [21]:

$$\mathbf{w}_{p+1}^{+} = \mathbf{w}_{p+1} - \sum_{i=1}^{p} \mathbf{w}_{p+1}^T \mathbf{w}_i \mathbf{i}, \; for i = 1, \ldots, p \tag{3.23}$$

$$\mathbf{w}_{p+1}^{*} = \mathbf{w}_{p+1}^{+} / \sqrt{\mathbf{w}_{p+1}^{+T} \mathbf{w}_{p+1}^{+}} \tag{3.24}$$

In the next section, we will discuss the method of estimating independent sound sources from a WAVE audio format file using this Fast-point algorithm.

3.2 Audio as a dynamical system

One of the problems that arises when applying the ICA algorithm to a digital audio file is the limitation in the number of recordings available in the sound file. As discussed earlier, the ICA model approximates the source signal s by estimating $\mathbf{w}_i$ for $i = 1, \ldots, n$. Thus the number of sources that can be estimated is limited to the number of available recordings, and this is, for audio recordings, either one or two.

To overcome this problem, one can consider a musical signal as a form of deterministic nonlinear *dynamical system.* Takens' Theorem[1] states that, for a dynamical system manifold A of dimension d, one can construct a $(2d + 1)$-dimensional vector from the data that represents the system.

When applying this theorem, an *embedding matrix* is constructed using a rectangular sliding window of window size m,

$$X = \begin{pmatrix} x_t & x_{t+\tau} & \cdots & x_{t+N\tau} \\ x_{t+\tau} & x_{t+2\tau} & \cdots & x_{t+(N+1)\tau} \\ \vdots & \vdots & \ddots & \vdots \\ x_{t+(m-1)\tau} & x_{t+(m)\tau} & \cdots & x_{t+(m+N-1)} \end{pmatrix}.$$

Our intention is to extract frequency-related information from the independent components, after applying ICA, and thus the delay between the windows is fixed to $\tau = 1$. When constructing an embedding matrix, one has to ensure that the window size is 'large' enough, so that the matrix created captures the information content in the signal, but not so large that we capture non-stationarity. To investigate this, we employ Singular Value Decomposition (SVD) to monitor how the singular value spectrum converges as the window size changes.

[1] See [6], which provides a good explanation of the theorem.

3.2.1 Complexity Analysis

In [23], the complexity of an embedding matrix was analysed using the singular spectrum obtained from the matrix. In a musical signal, pure tones (pure sine wave signal) have lower complexity (lower dimensionality). As discussed in Section 1, tones created by instruments consist of different ratios of harmonics of the fundamental frequency, and their amplitude may increase or decrease with time depending on how instruments are played. This is because timbre often depends on volume and pitch. When such partials change over time, the complexity of the signal increases. It may also increase as more tones are played together or played in sequences. The complexity of random noise is of very large dimensionality and considered very complex. Such complexity can be studied by analysing singular values of the embedding matrix.

Using SVD, the embedding matrix X can be re-written in the form:

$$X = U\Sigma V^T,$$

where U and V are orthogonal matrices of singular vectors, such that the matrix U is the matrix of projections of X on to eigenvalues of XX^T and Σ is a diagonal matrix of singular values $\Sigma = diag(s_1, \ldots, s_n)$. The singular values are equal to the square roots of the eigenvalues of XX^T. Thus monitoring the singular spectrum does not directly determine the contents of the actual data in embedding matrix X, but it enables us to have a measure of the structure of the data. For example, the data shown in Figures 3.4 and 3.5 would share the same singular spectrum Σ so the complexity of the embedding matrices are the same, but the actual data in matrices U and V are different.

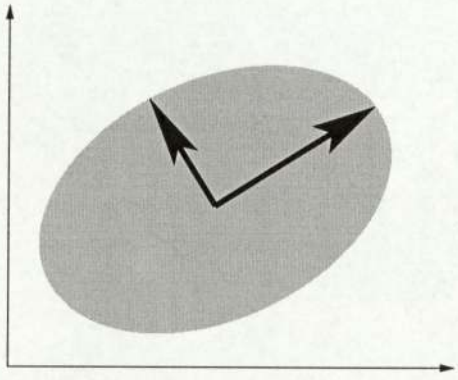

Figure 3.4: Data Space 1 with Singular Spectrum Σ.

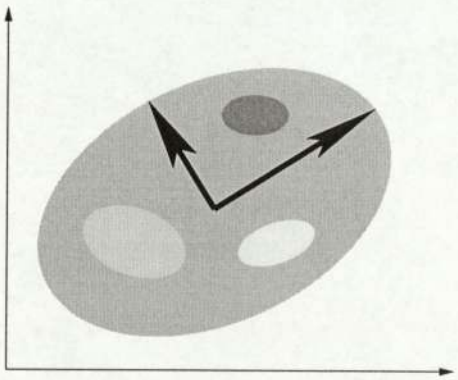

Figure 3.5: Data Space 2 with Singular Spectrum Σ

3.3 Complexity of Music

To analyse the complexity of a music signal, WAVE file format mono audio with $44100Hz$ sampling rate and 16 bit depth was used. The list of samples used can be found in Appendix and are provided by RWC Music Database [17]. The WAVE file format records amplitude at each sampling point, and bit depth denotes the number of

bits used to represent such amplitudes. Samples of length 0.1 second were taken from various points in several different genres of music[2].

3.3.1 Framework

Singular spectra of different segments of music were calculated for $k = 1, \ldots, K$ for $K = 77$ windows where the window size m increases with an interval of 5 so $m = 5(k+3)$ and $20 \leq m \leq 400$. The power of each spectrum increases as the window size gets larger since it start capturing more complexity of the signal. However, we are not so interested in the actual power of complexity, but the structure of the complexity captured in the embedded matrix. To do this, we first calculate the relative singular values Σ' for each singular spectrum by normalising it with the maxima of the spectrum, so that $\Sigma' = s_i / \max_i |s_i|$ for all $i = 1, \ldots, n$, then the power of each spectrum $\|\Sigma_k\|$ is calculated for $S = \{\Sigma'_1, \ldots, \Sigma'_K\}$. Let us denote the maximum power

$$P = \max_{\Sigma \in S} \|\Sigma\|.$$

Then the power ratio of each spectrum is ;

$$ratio_k = \frac{\|\Sigma_k\|}{P} \qquad \text{for } k = 1, \ldots, K.$$

For each spectrum $S = \{\Sigma'_1, \ldots, \Sigma'_K\}$, this ratio is used to stretch in the x-direction to obtain Σ^*_k, so it enables us to monitor how the power spreads on the spectra as the window size increases. The source code for this MATLAB operation is as follows;

```
for k = 1:K
 ratios = ratio(k):1/ratio(k):50/ratio(k);
 ratios = ratios - 1;
 spectrum_range = 0:1:49;
 Snew = interp1(ratios,Soriginal,spectrum_range)';
end
```

where the MATLAB function

```
YI = interp1(X,Y,XI)
```

interpolates to find YI, the values of the underlying function Y at the points in the vector XI, and the vector X specifies the points at which the data Y is given. After this operation, the change in structure of the spectra can be observed, thus we can monitor the change in the complexity of the embedded matrix.

The Sum-Squared Error (SSE) was used to monitor the convergence of the singular spectra as window size increases;

$$E_k = \sum_{n=1}^{N} (\Sigma_k^{*n} - \Sigma_{k+1}^{*n})^2,$$

where Σ_k^{*n} is nth singular value of the kth spectrum obtained and E_k is the error between kth and $(k+1)$th spectra.

[2]Please see Appendices for the details of some of the segments of music used.

3.3.2 Results

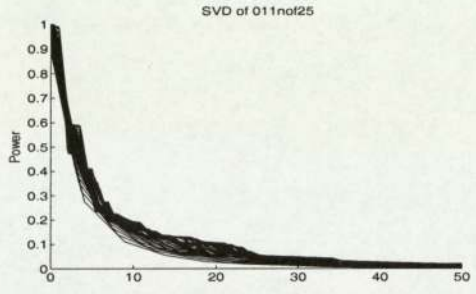

Figure 3.6: Normalised Singular Spectra of 011nof25.wav (piano A2, 110Hz).

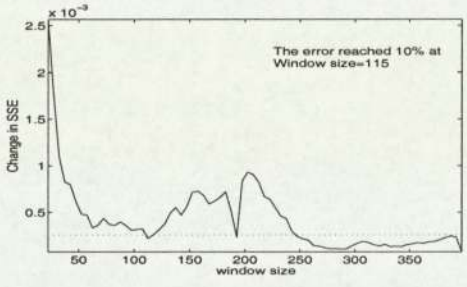

Figure 3.7: SSE calculated between the Singular Spectra of 011nof25.wav (piano A2,110Hz)at successive window sizes.

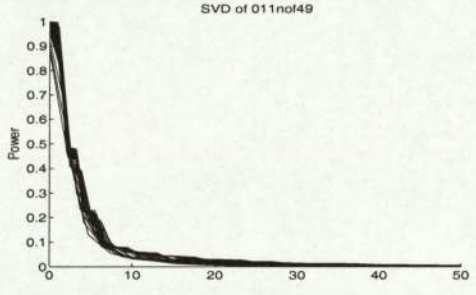

Figure 3.8: Normalised Singular Spectra of 011nof49.wav (piano A4, 440Hz).

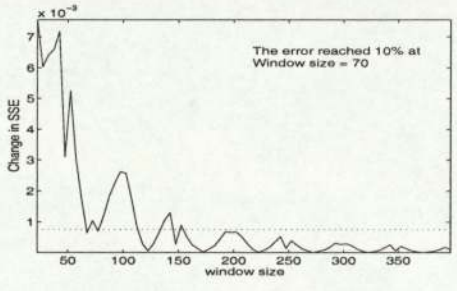

Figure 3.9: SSE calculated between the Singular Spectra of 011nof49.wav (piano A4,440Hz)at successive window sizes.

In Figures 3.6 and 3.8 C normalised singular spectra Σ' obtained from a piano playing A2 and A4 are plotted for all k, and Figures 3.7 and 3.9 show the SSE calculated between the normalised singular spectra. From Figures 3.6 and 3.8, spectra convergence to some structure at certain window sizes can be monitored. The spectra repeatedly converge to new structures as the window size increases. This convergence in structure can be monitored easier by the SSE plot on Figures 3.7 and 3.9. The SSE decreases as it converges to a structure and increases when new structure is captured as the window size grows. For note A2, the SSE reaches its 10% of the starting error when the window size is at 115, but it soon captures new structure, and does not stabilise until quite a large window size. For note A4, it reaches stabilisation quite quickly and the repeat in pattern in SSE shows that it repeat in capturing the similar structure. The repeat in the sequence occurs as the window size increases to $m \approx 100$.

These results, however, agree with the frequency range it captured by each window. The lowest frequency that could be captured by a window size of m is $\frac{F_S}{m}$, where F_S is the sampling frequency. Thus for $F_S = 44100$, it will be $\frac{44100}{400} = 110$Hz, in which it just captures all the complexity in a signal played at 110Hz.

In Figures 3.10 and 3.12, normalised singular spectra for violin and flute playing A4

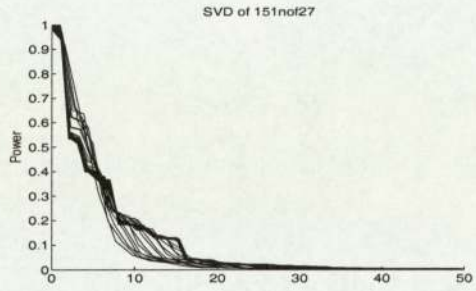

Figure 3.10: Normalised Singular Spectra of 151nof27.wav (violin A4, 440Hz).

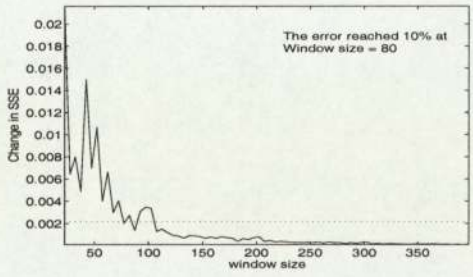

Figure 3.11: SSE calculated between the Singular Spectra of 151nof27.wav (violin A4,440Hz)at successive window sizes.

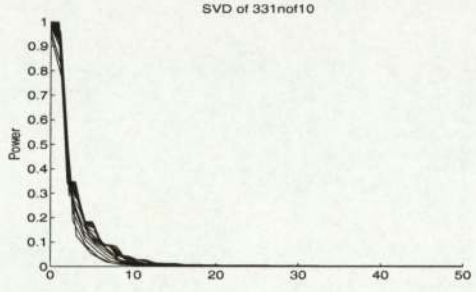

Figure 3.12: Normalised Singular Spectra of 331nof10.wav (flute A4, 440Hz).

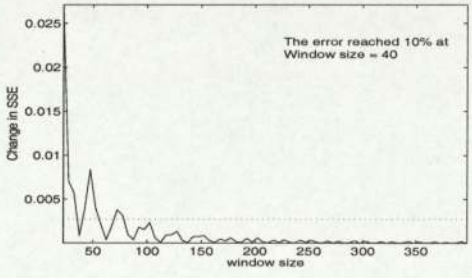

Figure 3.13: SSE calculated between the Singular Spectra of 331nof10.wav (flute A4,440Hz)at successive window sizes.

are plotted together with corresponding SSE plots in Figures 3.11 and 3.13. Again, the convergence to the structure can be monitored. The SSE of these samples decreases steadily and reaches stabilisation quicker than those of the piano. This means that the complexity of the piano is greater than that of violin and flute.

Figures 3.14 and 3.16 plot the normalised singular spectra of sample playing violin and piano chord respectively, and Figures 3.15 and 3.17 show their change in SSE with the window size grow. From these plots, it is confirmed that the complexity of the signal takes role in the convergence to the structure, while the frequency component takes role in the stability.

3.3.3 Conclusion to the singular spectra analysis

The normalised singular spectra were obtained for different window sizes. From the plots, the convergence to the structure in singular spectra is monitored as the window size grows to capture more complexity of the signal. To monitor these structures easily, SSE of successive singular spectra were calculated and plotted. It was shown that as the window size increases, it repeats convergence in structure, and finally reaches the stabilisation. While the size of window to converge to the structure relates to the complexity of the signal, it also has a strong correlation with the frequency content of the signal. As we would not want to capture noise from the audio signal, window size

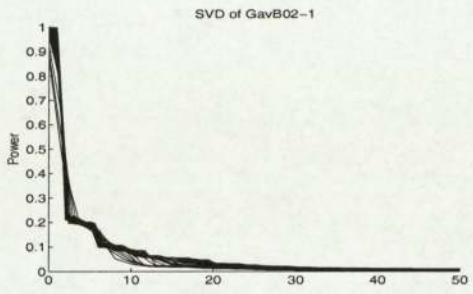

Figure 3.14: Normalised Singular Spectra of GavB02-1.wav (violin chord B4 & E5, 494Hz & 659Hz).

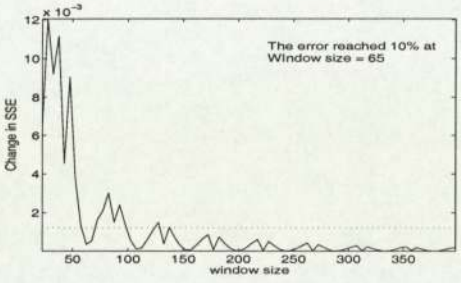

Figure 3.15: SSE between the Singular Spectra of GavB02-1.wav (violin chord B4 & E5, 494Hz & 659Hz) at successive window sizes.

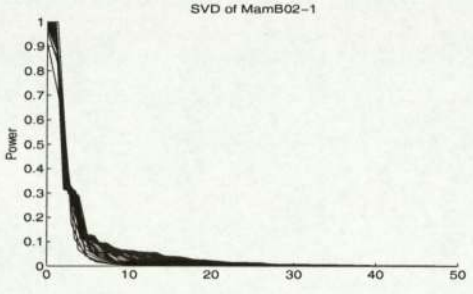

Figure 3.16: Normalised Singular Spectra of MamB02-1.wav (piano chord E4 & G5, 330Hz & 784Hz).

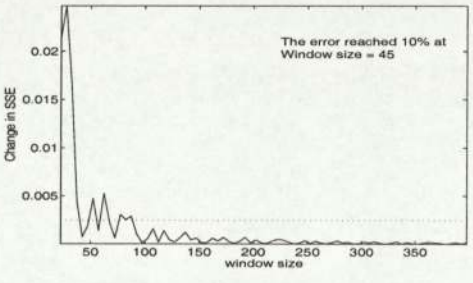

Figure 3.17: SSE calculated between the Singular Spectra of MamB02-1.wav (piano chord E4 & G5, 330Hz & 784Hz) at successive window sizes.

has to be relatively small, but also it has to be large enough to capture all the complexity needed. Thus we have come to choose a window size of $m = 200$ to build the embedding matrix for the later process.

3.4 Application of ICA to the embedding matrix

The embedding matrix was built using the experiments reported above to choose the window size, and FastICA was applied. Recall Equations 3.1 and 3.2, when FastICA is applied, the de-mixing matrix W is estimated, thus the mixing matrix A is easily calculated. In it, each column i shows the contribution of source signal s_i to the recorded signal $\mathbf{x}$ according to Equation 3.1. The magnitude of each column is calculated by taking the square of the values in the column and summing;

$$M_i = \sum_{j=1}^{m} A_{j,i}^2.$$

The magnitudes are sorted in descending order; the first 9 ICs extracted are shown in Figure 3.18.

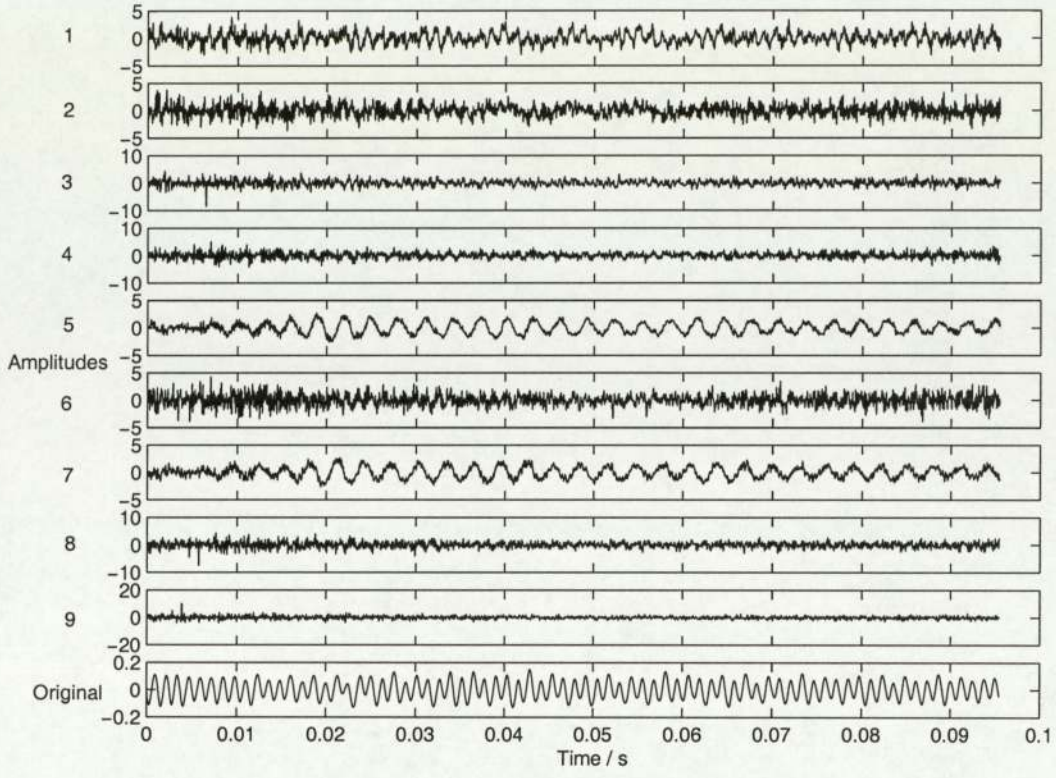

Figure 3.18: First 9 ICs extracted from MamB02-1.wav, piano chord playing E4 & G5, and the original signal used to extract those ICs.

Although there were some frequency components that can be observed, most of the results obtained did not look very useful in the time domain. However, spectrum plots of 1st and 2nd IC shown in Figures 3.19 and 3.20 suggests those signals extracted have some peaks in the frequency components, and some matches with the peaks of original signal.

The Yin fundamental frequency estimator [15] was applied to each IC extracted in an attempt to verify its f_0. The estimated f_0 together with estimated note and its magnitudes are shown in Table 3.4. In Figure 3.21, the magnitude of the independent components is shown. From Figure 3.21, it can be seen that the magnitude of each ICs drops dramatically with its order. Note that the magnitude of ICs can be translated as the importance of each ICs to the original signal. The estimated f_0 of all ICs extracted is plotted in accordance with the magnitude calculated and shown in Figure 3.22.

From the graph, one can clearly see the peaks in the fundamental frequencies extracted. The actual notes played were almost extracted with note G extracted at one octave lower. Other components such as note C, are the residues of the notes played from the moment just before the sample was taken. This will always happen with large instruments since the body of an instrument resonates, and thus the note tends to last longer. This is especially true when a low note is played as the attenuation of the

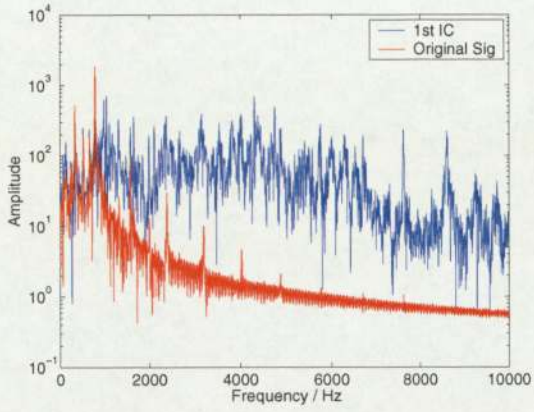

Figure 3.19: Spectrum of the original signal, MamB02-1.wav ,piano chord E4 & G5, plotted against the 1st IC extracted.

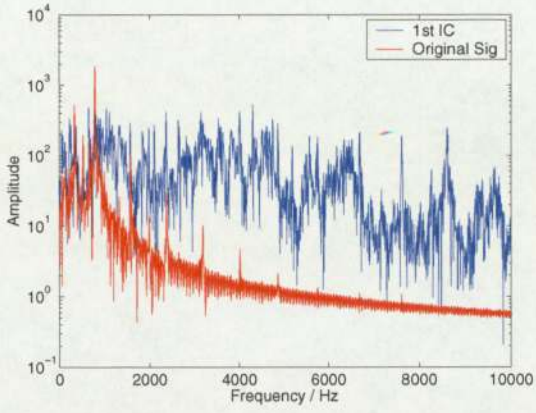

Figure 3.20: Spectrum of the original signal, MamB02-1.wav, piano chord E4 & G5, plotted against the 2nd IC extracted.

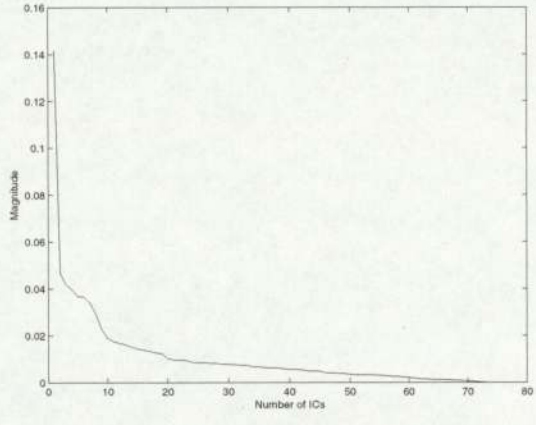

Figure 3.21: Magnitude of independent components shows the weight of each independent components to reconstruct the original signal.

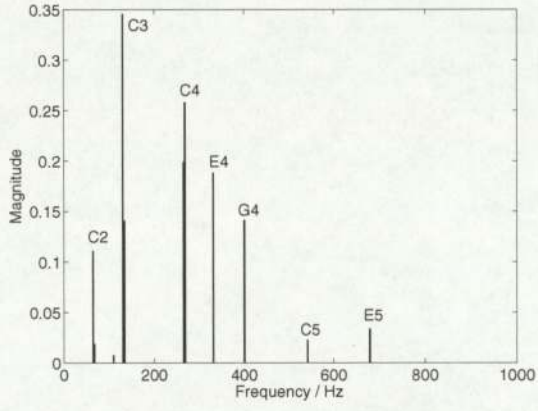

Figure 3.22: f_0 of ICs extracted from MamB02-1.wav (piano chord E4 & G5) plotted in accordance with the magnitude.

sound wave is less at lower frequency.

In Figures 3.23 and 3.24, the magnitude plot of ICs of a violin played at A4 on the D string and the A string are plotted. On Figure 3.24, a very distinct clear peak on A4 is recorded while there are some other peaks around A4 is recorded for the one played on D string on Figure 3.23. This is suspected to be due to a little hand movement such as vibrato on the string.

In Figures 3.25 and 3.26, the magnitude plot of ICs where flute is played at A5 and A6 are shown. In both cases the maxima of the magnitude plot is apparent, and the frequency components at the right fundamentals are extracted. However, the dispersal of peaks at non-related frequencies suggests that many underlying sources that are not explicitly related to the fundamental frequency of the instruments are being extracted.

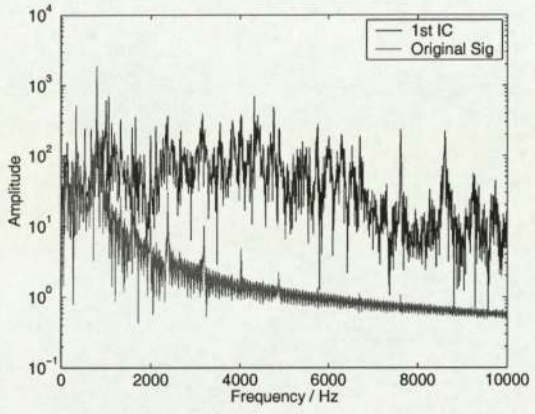

Figure 3.19: Spectrum of the original signal, MamB02-1.wav ,piano chord E4 & G5, plotted against the 1st IC extracted.

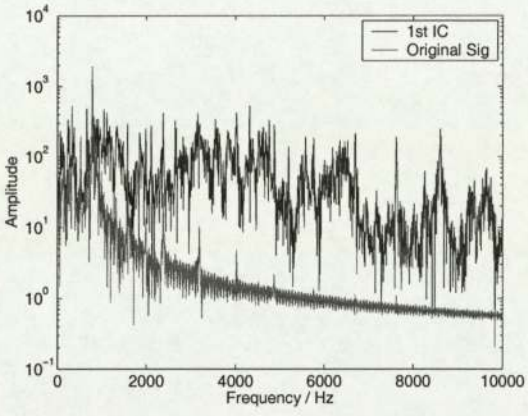

Figure 3.20: Spectrum of the original signal, MamB02-1.wav, piano chord E4 & G5, plotted against the 2nd IC extracted.

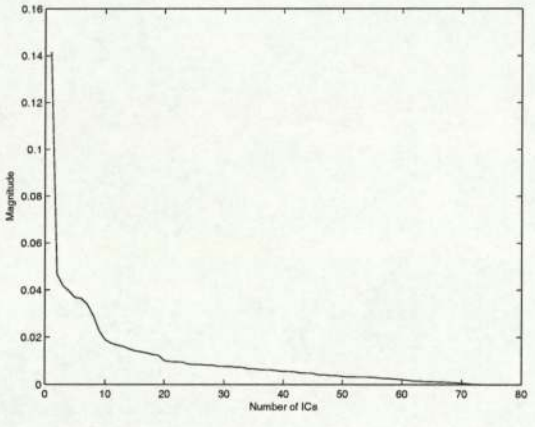

Figure 3.21: Magnitude of independent components shows the weight of each independent components to reconstruct the original signal.

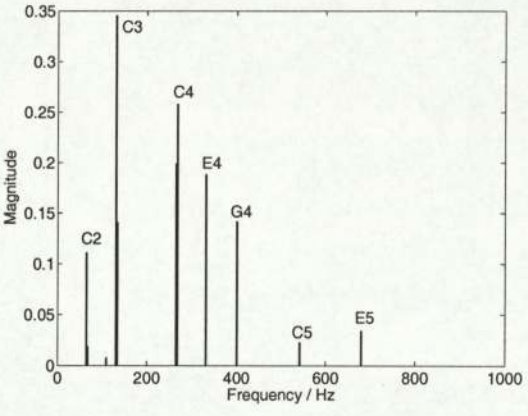

Figure 3.22: f_0 of ICs extracted from MamB02-1.wav (piano chord E4 & G5) plotted in accordance with the magnitude.

sound wave is less at lower frequency.

In Figures 3.23 and 3.24, the magnitude plot of ICs of a violin played at A4 on the D string and the A string are plotted. On Figure 3.24, a very distinct clear peak on A4 is recorded while there are some other peaks around A4 is recorded for the one played on D string on Figure 3.23. This is suspected to be due to a little hand movement such as vibrato on the string.

In Figures 3.25 and 3.26, the magnitude plot of ICs where flute is played at A5 and A6 are shown. In both cases the maxima of the magnitude plot is apparent, and the frequency components at the right fundamentals are extracted. However, the dispersal of peaks at non-related frequencies suggests that many underlying sources that are not explicitly related to the fundamental frequency of the instruments are being extracted.

Original signal: 391.3283Hz (G4 -3 cents)						
IC number in order of Mag	1	2	3	4	5	6
Estimated f_0/Hz	131.617	131.678	65.927	268.192	332.034	131.718
Estimated note + detuning/cents	C3 +11	C3 +11	C2 +14	C4 +43	E4 +13	C3 +12
Magnitude	0.1416	0.0467	0.0418	0.0396	0.0368	0.0365

7	8	9	10	11	12	13	14
331.002	131.929	268.109	132.121	265.814	401.830	131.763	131.782
E4 +7	C3 +15	C4 +42	C3 +17	C4 +27	G4 +43	C3 +13	C3 +13
0.0339	0.0292	0.0227	0.0190	0.0175	0.0168	0.0161	0.0152

15	16	17	18	19	20	21	22
266.524	132.154	265.893	65.995	65.946	131.979	131.94	1572.473
C4 +32	C3 +18	C4 +28	C2 +16	C2 +14	C3 +15	C3+15	G6 +5
0.0143	0.0139	0.0134	0.0127	0.0124	0.0102	0.0097	0.0096

23	24	25	26	27	28	29	30
1567.571	265.227	132.066	268.278	131.842	65.873	65.907	132.560
G6 -0	C4 +24	C3 +17	C4 +43	C3 +14	C2 +12	C2 +13	C3 +23
0.0095	0.0087	0.0085	0.0085	0.0083	0.0082	0.0078	0.0078

Table 3.1: Estimated f_0, notes, and their magnitudes of first 30 ICs extracted from MamB02-1, piano playing chord A4 & G5, 330Hz & 784Hz.

This may be due to the aspiration noise occurring when the flute is played. In Figures 3.27 and 3.28, the magnitude plot of ICs extracted for piano played at A2 and A3 are shown. The peaks are found around the expected fundamental frequency, but the peaks are found dispersed around the maxima. Note that for each note on a piano, there are 3 strings tuned around the desired frequency that are hit by the hammer action. Those strings are slightly detuned so the notes lasts longer when the strings are hit. Thus the dispersal around the desired frequency is thought to be the outcome of such physical properties of the piano.

We have also carried out some analysis for chords. In Figure 3.29, the magnitude plot of a piano chord sample is shown. This shows the peaks at an octave lower than expected, and along with those octave errors, some inexplicable peaks have been produced. Figure 3.30 shows the magnitude plot of the chord played shortly after the chord played for Figure 3.29. It shows peaks where they are expected and residuals from the chord played before, together with some inexplicable peaks. In Figure 3.31, the magnitude plot of a violin chord is shown. This has shown the peaks at the expected pitches with suspected vibrato components around the notes, but also it has shown some erroneous components at lower octaves together with some inexplicable results. However, for the chord played by violin shown in the Figure 3.32, there are clear peaks at the expected frequencies.

3.4.1 Conclusion to the FastICA on the Embedded matrix

Some frequency-related underlying sources of the original signals have been extracted. The notes played by violin showed little error around the f_0 due to the vibrato. The f_0 estimated from the ICs extracted for notes played by flute showed the clear peak at

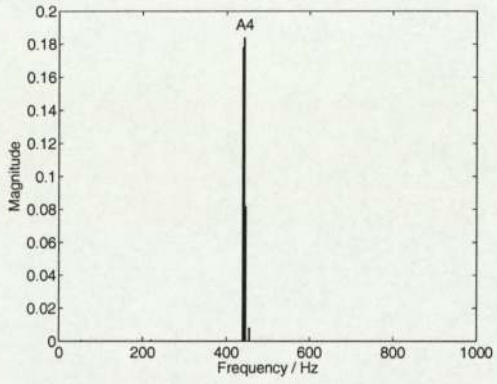

Figure 3.23: f_0 of ICs extracted from 151nof21.wav (A4 violin) plotted in accordance with the magnitude.

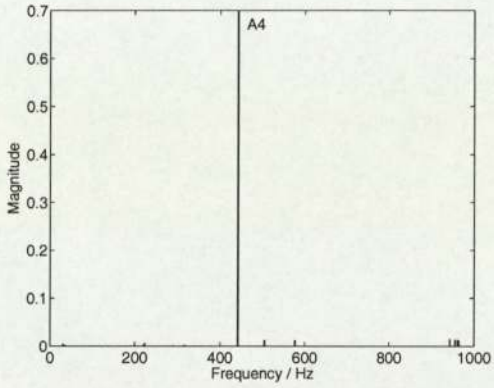

Figure 3.24: f_0 of ICs extracted from 151nof27.wav (A4 violin) plotted in accordance with the magnitude.

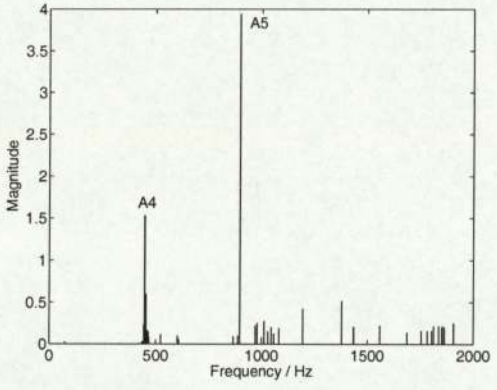

Figure 3.25: f_0 of ICs extracted from 331nof22.wav (A5 flute) plotted in accordance with the magnitude.

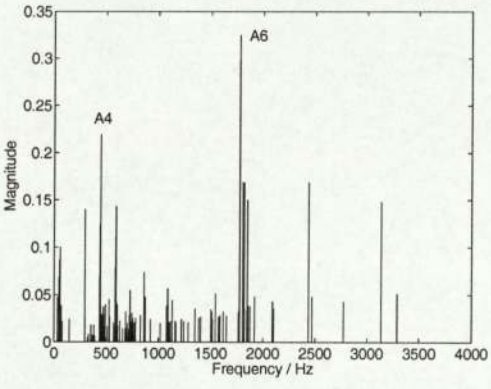

Figure 3.26: f_0 of ICs extracted from 331nof34.wav (A6 flute) plotted in accordance with the magnitude.

the expected f_0 but also showed large dispersal of the f_0 along many frequency ranges. This may be due to the aspiration noise and frication noise produced when flute is played. The notes played by piano showed the dispersal of the peaks around the note played. This is suspected to be caused by the three strings. The results of a single note played somehow reflected the tonal quality of each instruments; however some of the results shown for the chords played were erroneous while some showed good estimation.

Because of the limitation of the material available at the time, it was not possible to test for chords produced by a mixture of instruments. It would be interesting to see how well they could be extracted as the sound sources would be physically independent.

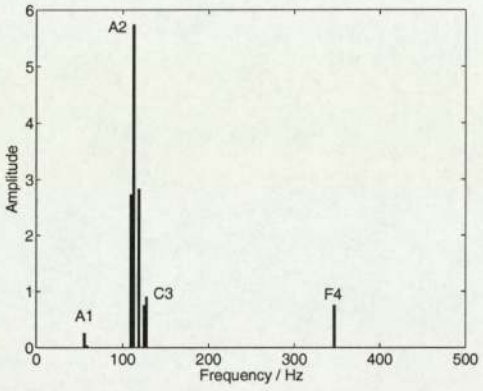

Figure 3.27: f_0 of ICs extracted from 011nof25.wav (A2 piano) plotted in accordance with the magnitude.

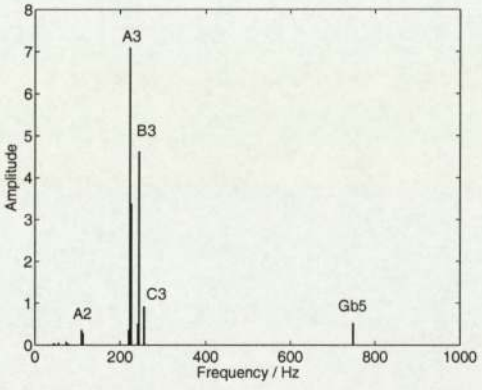

Figure 3.28: f_0 of ICs extracted from 011nof37.wav (A3 piano) plotted in accordance with the magnitude.

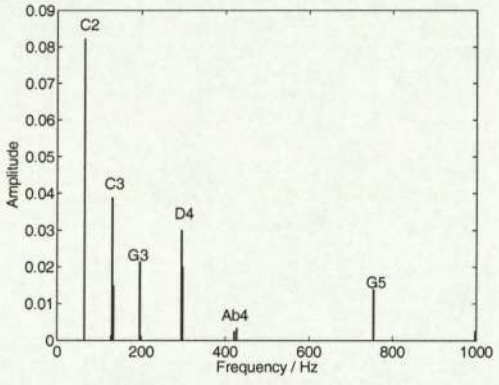

Figure 3.29: f_0 of ICs extracted from MamB25-0.wav (C3 & D5 piano) plotted in accordance with the magnitude.

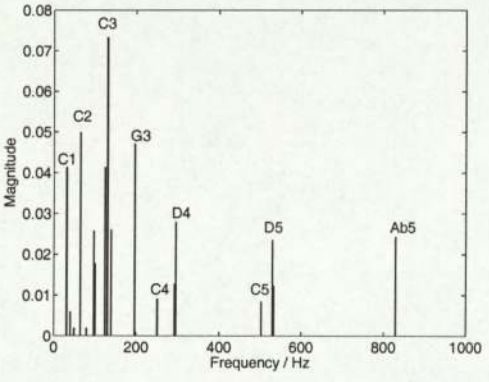

Figure 3.30: f_0 of ICs extracted from MamB25-3.wav (C3 & C5 piano) plotted in accordance with the magnitude.

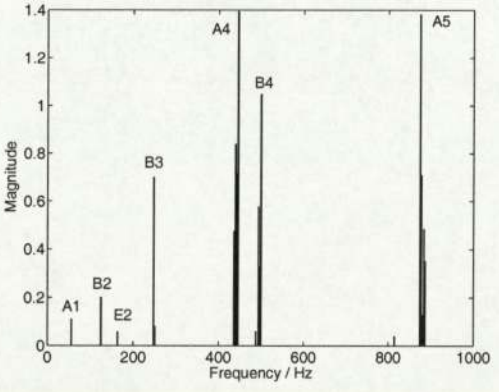

Figure 3.31: f_0 of ICs extracted from GavB02-1.wav (B4 & A5 violin) plotted in accordance with the magnitude.

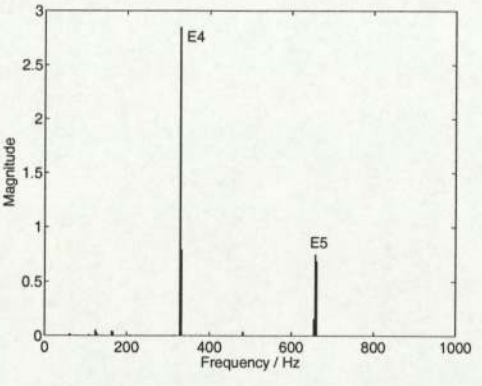

Figure 3.32: f_0 of ICs extracted from GavB08-1.wav (E4 & E5 violin) plotted in accordance with the magnitude.

Chapter 4

Timbre Analysis

In this chapter, we shall discuss a method of modelling a musical signal generated by different instruments and estimating the size of the harmonics for each note. First, additive synthesis of a pure sine wave and its harmonics for modelling a single tone (monophonic) and mixture of several tones (polyphonic) are described, followed by the estimation method for the parameters of the model. Then the estimation method is evaluated and results are discussed.

4.1 Harmonic Model (monophonic)

In a basic theory of signal processing, it is stated that any finite power periodic signal can be represented as a summation of sine waves and cosine waves. In Chapter 1, we have discussed that the colour of the sound created by the instrument can be analysed using the ratio of partials found in each note played, and in Chapter 2, a brief history of *additive synthesis*, which attempts to model the tonal structure as a summation of sine waves and cosine waves, was discussed. Davy and Godsill [14] developed a harmonic model based on such method, and the basic model of a single note for a short time interval is,

$$y(t) = \sum_{n=1}^{N} A_n \cos(2\pi n f_0 t) + B_n \sin(2\pi n f_0 t) + v(t) \qquad \text{for } t = 0, \ldots, T-1, \tag{4.1}$$

where f_0 is the fundamental frequency, N is the number of partials present, A_n and B_n are the amplitude of these partials, and $v(t)$ is the noise component. This model agrees with the FFT given by the note played by the instruments in Figure 1.8 and 1.7, where there are peaks at the harmonics of the fundamental frequency. However, in the above model it is assumed that the amplitude of each partial is constant throughout the interval $[0,\ T-1]$. For longer intervals, this may not be true, so they generalised the model,

$$y(t) = \sum_{n=1}^{N} A_{n,t} \cos(2\pi n f_0 t) + B_{n,t} \sin(2\pi n f_0 t) + v(t), \tag{4.2}$$

so the amplitudes $A_{n,t}$ and $B_{n,t}$ can now depend on time. In this model, it is important to model the true frequency of the fundamental, and to cater for low frequency

variation in amplitudes (e.g. vibrato). Thus smooth basis function ϕ_i for $i = 1, \ldots, I$, were introduced to represent the amplitude $A_{n,t}$ and $B_{n,t}$;

$$A_{n,t} = \sum_{i=1}^{I} A_{n,i}\phi_{i,t}, \quad \text{and} \quad B_{n,t} = \sum_{i=1}^{I} B_{n,i}\phi_{i,t}.$$

The basis function can be any sufficiently smooth interpolation function. Here we shall use half over-lapping raised cosine functions (Hanning windows). It is important to choose the window size large enough to eliminate the unwanted low frequency components, but small enough, so it can include the audible frequencies played by the instruments. Also, by introducing basis functions, the number of parameters in the model becomes much lower as I would typically be much smaller than T. For a monophonic signal, the model becomes;

$$y(t) = \sum_{n=1}^{N}\sum_{i=1}^{I} \phi_{i,t}\left\{A_{n,t}\cos(2\pi n f_0 t) + B_{n,t}\sin(2\pi n f_0 t)\right\} + v(t). \tag{4.3}$$

Now, the unknown parameters to determine the model in Equation 4.3 are; the fundamental frequency f_0, and the amplitudes $\theta = \{A_{1,1}, B_{1,1}, \ldots, A_{N,I}, B_{N,I}\}$. More precisely;

$$\theta[2(i+n)-1] = A_{n,i} \tag{4.4}$$
$$\theta[2(i+n)] = B_{n,i}. \tag{4.5}$$

With the assumption that I and N are already specified, the model is written as;

$$y = D\theta + v, \tag{4.6}$$

where $y = \{y_0, \ldots, y_{T-1}\}^T$, $v = \{v_0, \ldots, v_{T-1}\}^T$, the matrix D contains the Gabor atoms stacked in columns [14], such that;

$$D[t+1, 2(i+n)-1] = \phi_{i,t}\cos(2\pi n f_0) \tag{4.7}$$
$$D[t+1, 2(i+n)] = \phi_{i,t}\sin(2\pi n f_0). \tag{4.8}$$

The background noise v also includes components created by instruments which do not fit in the harmonic model. For example, when a person plays a wind instruments, it may also emit air sounds, or aspiration noise. Here we assume the noise to be an autoregressive (AR) model of order M;

$$v(t) = \sum_{m=1}^{M} \alpha_m v_{t-m} + \epsilon_t, \tag{4.9}$$

where ϵ_t is a zero mean Gaussian white noise of variance $\sigma\epsilon^2$. Given the linear model in Equation 4.3, and the assumption of i.i.d. Gaussian excitation for the AR process, the likelihood function will be [14]:

$$p(y|\theta, f_0, N, \alpha, \sigma_\epsilon^2) = \frac{1}{(2\pi\sigma_\epsilon^2)^{T/2}}\exp\left[-\frac{1}{2\sigma_\epsilon^2}(y - D\theta)^T A^T A(y - D\theta)\right]. \tag{4.10}$$

A is a $T \times T$-dimensional matrix constructed by stacking the AR coefficients α in rows:

$$A = \begin{bmatrix} 1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ -\alpha_1 & 1 & 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & & & \vdots \\ -\alpha_M & \cdots & -\alpha_1 & 1 & 0 & \cdots & 0 \\ 0 & \ddots & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -\alpha_M & \cdots & -\alpha_1 & 1 \end{bmatrix}$$

Let us use S to denote the $2NI$-dimensional square matrix

$$S = D^T A^T A D, \tag{4.11}$$

and the distribution of θ conditional on the other parameters is defined by;

$$p(\theta | f_0, N, \alpha, \sigma_\epsilon^2, y) = N(\mu, \sigma_\epsilon^2 S), \tag{4.12}$$

where μ is the $2NI$-dimensional vector,

$$\mu = S D^T A^T A y. \tag{4.13}$$

4.2 Harmonic model (polyphonic)

In the section above, we have discussed the method of estimating fundamentals and related parameters in the harmonic model given in Equation 4.3. However, in most cases in the musical signal, several notes are recorded at the same time (polyphony). For the modelling of polyphonic signals with K notes present at a time, the Equation 4.3 can quite easily be extended to a mixture of monophonic models;

$$y(t) = \sum_{k=1}^{K} \sum_{n=1}^{N} \sum_{i=1}^{I} \phi_{i,t} \left\{ A_{k,n,t} \cos(2\pi n f_{0,k} t) + B_{k,n,t} \sin(2\pi n f_{0,k} t) \right\} + v(t). \tag{4.14}$$

The matrix D from Equation 4.7 then should be of the form;

$$\begin{aligned} D[t+1,\ 2R_{k-1}(I+1) + 2(N_k i + n) - 1] &= \phi_{i,t} \cos(2\pi n f_0) \\ D[t+1,\ 2R_{k-1}(I+1) + 2(N_k i + n)] &= \phi_{i,t} \sin(2\pi n f_0). \end{aligned} \tag{4.15}$$

where $R_k = \sum_{j=1}^{k} N_j$. Also θ from Equation 4.4 should be;

$$\theta[2R_{k-1}(I+1) + 2(i+n) - 1] = A_{n,i} \tag{4.16}$$

$$\theta[2R_{k-1}(I+1) + 2(i+n)] = B_{n,i}. \tag{4.17}$$

4.3 Estimation of Parameters

In [14], all the parameters defined in Section 4.1 are estimated from Markov Chain sampling. However, given the number of parameters which have to be sampled in their model, this method requires many samples to be made for the optimisation process, and thus the simulation can run for hours, or sometimes days. Markov chain sampling, however, is a very straight forward technique, which we will be using as the core algorithm for the estimation of the parameters defined in Section 4.1 together with other methods for optimisation of parameters. The sampling method is called a Markov chain if each sample $x(t+1)$ is generated from the last sample $x(t)$ by taking a step determined by a stochastic transition. The aim is to sample from a Markov chain whose stationary distribution is our posterior distribution. To do this, we generate a new state from an old state by generation of a candidate state from a proposal distribution, and then we decide whether to accept or reject the state. If the new state is accepted, then the state now becomes a candidate state, otherwise we keep the previous state [29]. There are several other sampling methods available, but here we will be using the Metropolis-Hasting algorithm because of its simplicity, and to make use of the fact we can easily build the probability density distribution $p(f_0|y)$, proportional to the spectrum of y, for the proposal distribution of f_0. First in this section, we shall discuss using Markov chain sampling for estimation of f_0. Later in this section, the method of using other f_0 estimators, such as Yin fundamental frequency estimator [15] will be discussed.

4.3.1 Markov Chain Sampling and Estimation

As mentioned earlier, Markov chain sampling is a computationaly expensive process as it involves generating many samples to get a good estimate. Thus generating samples for each parameter defined in the model as suggested in [14] seemed computationally too demanding. We suggest the following algorithm which uses Markov chain sampling with maximum likelihood estimation by Yule-Walker equation for monophonic recordings to minimise the number of samples needed to achieve a fair result given the computational time needed.

1. Build the probability density distribution $p(f_0|y)$ proportional to the spectrum of y. This now becomes our proposal distribution.

2. Generate the candidate state f^* from the proposal distribution.

3. Accept the first candidate state and let $f^l = f^*$ where l is the number of accepted state, and f^l denotes the lth accepted state.

4. Sample u from $\mathcal{U}[0, 1]$.

   (a) **If** $u < \lambda_1$
   Generate the new candidate state f^* from proposal distribution $p(f_0|y)$.

   (b) **Else if** $u < \lambda_1 + \lambda_2$
   Sample k from $\mathcal{U}[0, 1]$.

      i. **If** $k > 0.5$
      Generate a new candidate state f^* from $\mathcal{N}(2f^l, \sigma_f^2)$.

ii. **Else**
Generate a new candidate state f^* from $\mathcal{N}(\frac{f^l}{2}, \sigma_f^2)$.

(c) **Else**
Generate a new candidate state f^* from $\mathcal{N}(f^l, \sigma_f^2)$

where λ_1 and λ_2 are parameters to switch between sampling methods.

5. Update matrix D.

6. Estimate matrix A and θ (see later for more details).

7. **If** $\eta(f^*) \geq \eta(f^l)$, accept the candidate state;
let $l = l + 1$, and $f^l = f^*$.
Else reject the candidate state.

8. Back to step 4 and repeat.

Steps 4(b)i and 4(b)ii allow us to perform an octave jump between frequencies to handle the case where the fundamental frequency f_0 should be an octave higher or lower than its current value. This process is important to handle the strong overlap of partials for notes on octave apart generated by instruments. In step 7, the energy or cost function E is calculated as the negative log likelihood of the model; this is given by Equation 4.10 so $p(y|\theta, f_0, N, \alpha, \sigma_\epsilon^2) \propto \exp(-E(y|\theta, f_0, N, \alpha, \sigma_\epsilon^2))$. The test always accepts candidate states with low energy, but only accepts candidate states of higher energy with probability $\exp(E(y|\theta^l, f^l, \alpha^l, N, \sigma_\epsilon^2) - E(y|\theta^*, f^*, \alpha^*, N, \sigma_\epsilon^2))$. So step 7 can be re-written as [29];

E1. If $E(y|\theta^*, f^*, \alpha^*, N, \sigma_\epsilon^2) \leq E(y|\theta^l, f^l, \alpha^l, N, \sigma_\epsilon^2)$, then accept f^*.

E2. If $E(y|\theta^*, f^*, \alpha^*, N, \sigma_\epsilon^2) > E(y|\theta^l, f^l, \alpha^l, N, \sigma_\epsilon^2)$, then accept f^* with probability;

$$\exp(E(y|\theta^l, f^l, \alpha^l, N, \sigma_\epsilon^2) - E(y|\theta^*, f^*, \alpha^*, N, \sigma_\epsilon^2)) \tag{4.18}$$

For the estimation of matrix A and θ we make use of Equations 4.11 to 4.13.

1. Let A be an identity matrix of size $T \times T$.

2. Update S and μ using Equations 4.11 to 4.13.

3. Let $\theta^* = \mu$.

4. Re-calculate noise v according to Equation 4.6.

5. Fit AR model to the noise v by maximum likelihood, and re-build matrix A.

6. Back to step 2 and repeat until sum-squared error between θ^j and θ^{j+1} converges, where j is the iteration number.

Estimation of the coefficients α of the noise model in step 5 was performed using the Yule-Walker method. Recall the AR process of Equation 4.9, by multiplying both sides by $v(t-\tau)$, taking expectation values, and normalising, the autocorrelation coefficients can be found by solving the set of linear equations;

$$\rho_\tau = \sum_{m=1}^{M} \alpha_m \rho_{\tau-m}.$$

Then by solving the system, AR coefficients can be determined.

4.3.2 YIN fundamental frequency estimator

The fundamental frequency (f_0) of a signal is the lowest frequency component, which relates to the other partials. For a periodic signal, f_0 relates to the periodicity of the signal, and is the inverse of its period. As discussed briefly in Section 2.4, there exists many methods of f_0 estimation, and using the autocorrelation method would be the simplest method available where it measures the 'similarity' between two windowed waveform with time lag. However, a problem arises when it is applied to the complex waveform as it would produce many maxima between 'true' maxima. To solve such problem, some form of peak detection algorithm is needed.

To overcome this problem, the YIN f_0 estimator was developed on similar principles to the autocorrelation algorithm [15]. It is based on the difference function, which attempts to minimise the difference between the waveform and time shifted waveform:

$$d_t = (\tau) \sum_{j=1}^{W} (x_j - x_{j+\tau}),$$

where $d_t(\tau)$ is the difference function of lag τ. So the function starts at 1 and it remains high until the difference in the function drops to its dip. Thus by searching for the values of τ for which the function is zero, the period of the waveform is found. As for autocorrelation method, several dips may be found at subharmonics. This has been overcome by first, introducing the 'cumulative mean normalised difference function':

$$d'_t(\tau) = \begin{cases} 1, & \text{if } \tau = 0, \\ d_t(\tau)/[(1/\tau)\sum_{j=1^\tau} d_t(j)] & \text{otherwise,} \end{cases}$$

and introducing an absolute threshold and choose the smallest value of τ that gives a minimum of $d_t(\tau)$ deeper than the threshold.

With this f_0 estimator, the process of parameter estimation is shortened:

1. Estimate f_0 using Yin estimator.
2. Update matrix D.
3. Estimate matrix A and θ (see the algorithm in Section 4.3.1 for detail).

4.4 Experimental Framework

A collection of short segments of piano, violin, clarinet, and flute playing a single tone and chords are taken from RWC Music Research Database [17] for the analysis. The list of details on each samples can be found in Appendix A. The samples are sampled at 44100Hz sampling frequency, and 3300/sample size. Hanning windows of size 512 samples with 50% overlapping were used as a basis function.

4.5 Results and Discussion

We are expecting to see the results with the highest amplitude at the fundamental frequency, and different sizes of partials depending on the instrument. It is hoped that we could observe some characteristic features in how the amplitude parameters θ are organised within the segments. The amplitude parameters θ shows the change in power ratio of each partials with time, which relates to the timbre of the instrument.

4.5.1 Results using Markov chain sampling for f_0 estimation

For the estimation process using Markov chain sampling, the number of iterations were set to 50, and the average acceptance ratio of the Markov chain sampling was $\approx 14\%$. The estimates made for f_0 had an accuracy of 56% accuracy with $\pm$ 5% precision and if those mis-estimated by an octave are included, the accuracy then becomes 76%. The resulting θ were re-organised and plotted as a density map in Figures 4.1 and 4.2 with estimated fundamental frequency shown next to the corresponding plots. The x-direction denotes the ith basis function (equivalent to time segments), and y-direction denotes the partials, in which the intensity of the partials are calculated by

$$\varphi_n = (\theta_{2n-1}^2 + \theta_{2n}^2)^{1/2} \quad \text{for } n = \{1, \ldots, N\}\,. \tag{4.19}$$

The results obtained for piano notes are shown as Sample 1 to 10 in Figure 4.1. Most results obtained for piano did not show the results as expected. The estimation of the f_0 was proven to be very unsuccessful. For Samples 5, 9, 10, f_0 were estimated with octave error, and showed the brightest line at the 1st harmonic f_1. However, for Sample 8, f_0 was correctly estimated, and a bright line at the f_0 was recorded. For results obtained for violin shown as Sample 11 to 20, f_0 estimations did not work so well for lower notes played on the G and D strings. However, for notes played on higher strings (A and E), the f_0 estimation showed better results, thus a bright line at f_0 was observed. The results obtained for clarinet again showed better f_0 estimation at lower frequencies, and the same for the results obtained from flute notes. Overall, f_0 estimation worked better at higher frequencies. This is because of the nature of the model, that is based on the periodicity of the data. When a note is produced at lower f_0, the partials are also found at lower frequencies. This reduces the periodicity of the signal captured within the windowed segments, thus the model used, which is based on the periodicity of the data cannot model the signal well anymore. Even with fairly good estimation of f_0, the density plot did not show consistent characteristics for each instrument.

In Figure 4.3, the original signal of Sample 1 (piano A2 at 110Hz) and the recovered

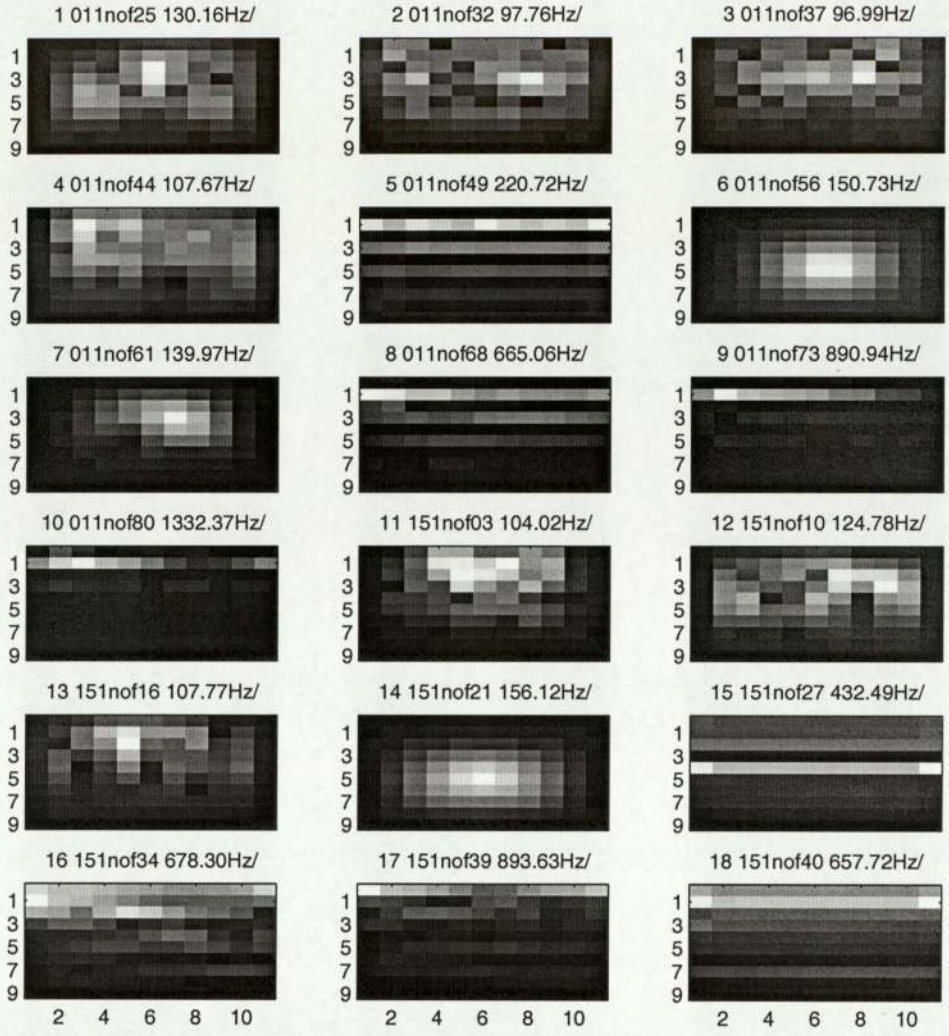

Figure 4.1: Graphical view of estimated amplitude of each partials obtained by Markov chain sampling for samples 1 to 18.

signal from the estimated parameters are plotted on the same graph, and in Figure 4.4, spectra of these signals can be found. From those graphs, especially in Figure 4.3, it can be seen that the signal recovered by the estimated parameter fits very well to the original signal, although as shown in Figure 4.1, the f_0 estimation did not work well and the density plot of the harmonics did not show what was expected. Also in Figure 4.3, a transient in the first 0.01s with a different structure to the main note is observed. This is due to the hammer striking a string when a piano is played. The model fits this transient well, but this leads to inaccurate estimation of parameters. In Figure 4.4, it was shown that the estimated signal had many peaks that match with the original up to around 1500Hz, this is because of the fact that the estimated f_0 was at 130.16Hz, and parameters were estimated for the first 10 harmonics.

Similarly in Figures 4.5, the signal and recovered signal of Sample 13 (violin E4 330Hz on D string) are shown, and 4.6 shows their spectrum. f_0 was estimated as 107.77Hz, and it showed matching peaks up to slightly above 1000Hz on Figure 4.6. Again, the recovered signal showed good fit to the original even though the density plot shown in

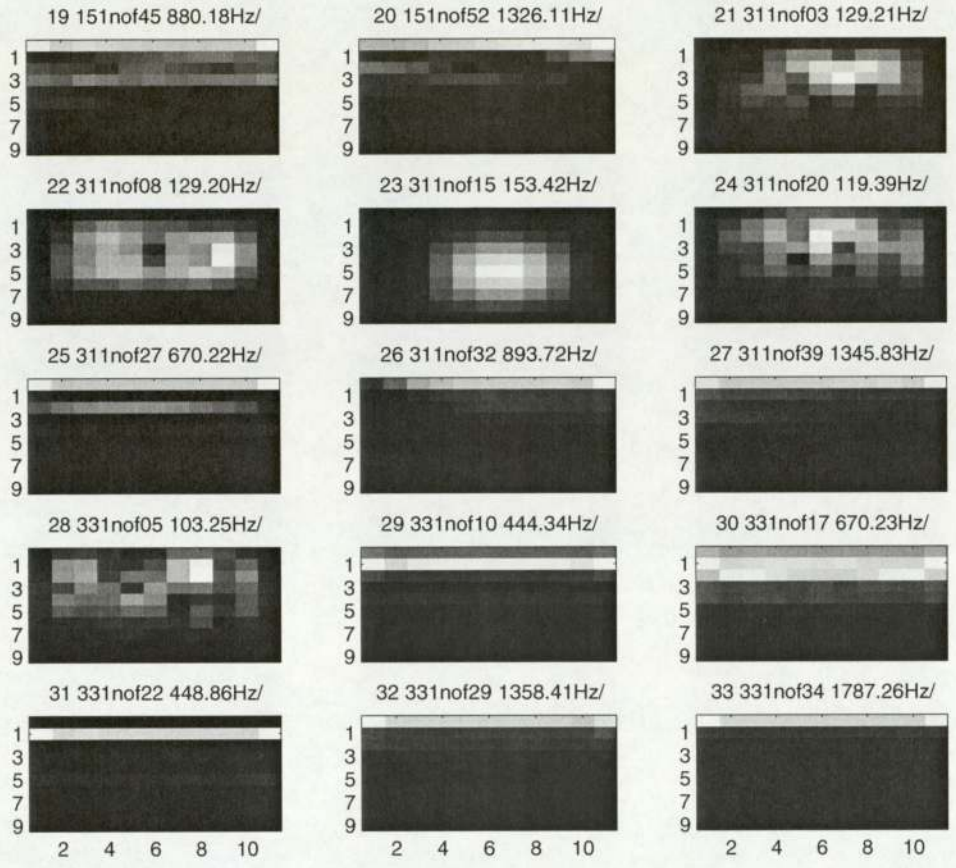

Figure 4.2: Graphical view of estimated amplitude of each partials obtained by Markov chain sampling for samples 19 to 33.

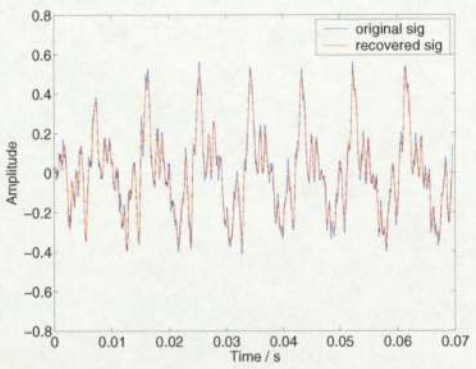

Figure 4.3: Sound wave of Piano A2, 110Hz, and signal recovered from the estimated parameters.

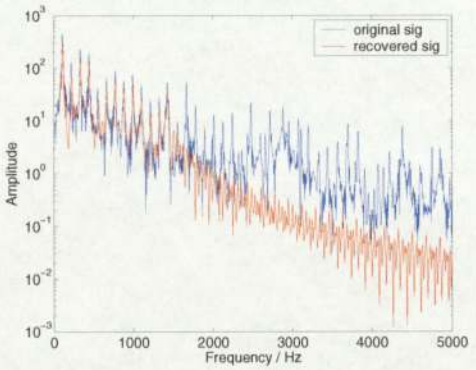

Figure 4.4: Spectrum of Piano A2, 110Hz, and signal recovered from the estimated parameters.

Figure 4.1 was not what was expected.

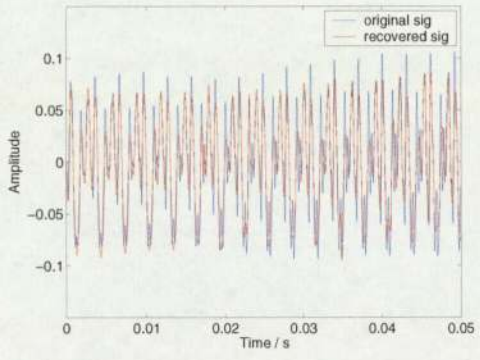

Figure 4.5: Sound wave of Violin E4, 330Hz, and signal recovered from the estimated parameters.

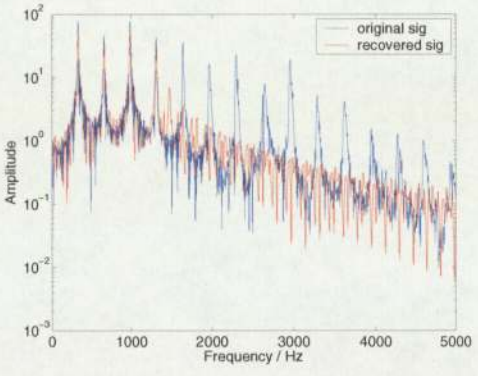

Figure 4.6: Spectrum of Violin E4, 330Hz, and signal recovered from the estimated parameters.

4.5.2 Results using YIN f_0 estimator

f_0 was estimated and using the algorithm explained in Section 4.3.2, other parameters are obtained. By using Equation 4.19, intensity of the partials are calculated and plotted in Figures 4.7 and 4.8.

YIN f_0 estimator was proven to be an effective f_0 estimator with its estimation time considerably faster than using Markov chain sampling. With its estimation of f_0, the density plot of the harmonic model seemed almost as expected for more samples. However, for Sample 1, the density plot did not show clear lines at the harmonics. From the plot of the original signal shown in Figure 4.3, it can be seen that this is caused by the transient in the original signal. The phenomenon observed on Sample 2 on Figure 4.7, which can also be observed a little on Sample 1 and on Sample 21 on Figure 4.8 is caused by the window size that is too small to capture whole period of wave. This can be proven on the plot shown in Figures 4.9, in which the window size of 1024 with overlap of 512 is used.

The density plots of piano did not show much characteristics apart from showing some partials existing. This may be because of its inharmonicity of the partials, so it becomes impossible to monitor the correct amplitude at each harmonics by modelling the tone as simply summation of multiples of f_0. However, results produced by violin samples have shown many strong lines up to few partials. There were very clear characteristics observed for the notes played by clarinet. This is because of the construction of clarinet, as briefly discussed in Introduction. Again, it showed clear lines around f_0 in the results produced by flute notes, but no clear characteristics were observed.

Overall results shown more harmonics being observed at the notes played at lower frequency. This is because attenuation of sound wave being greater at high frequency and many partials produced were attenuated for the notes played at higher f_0.

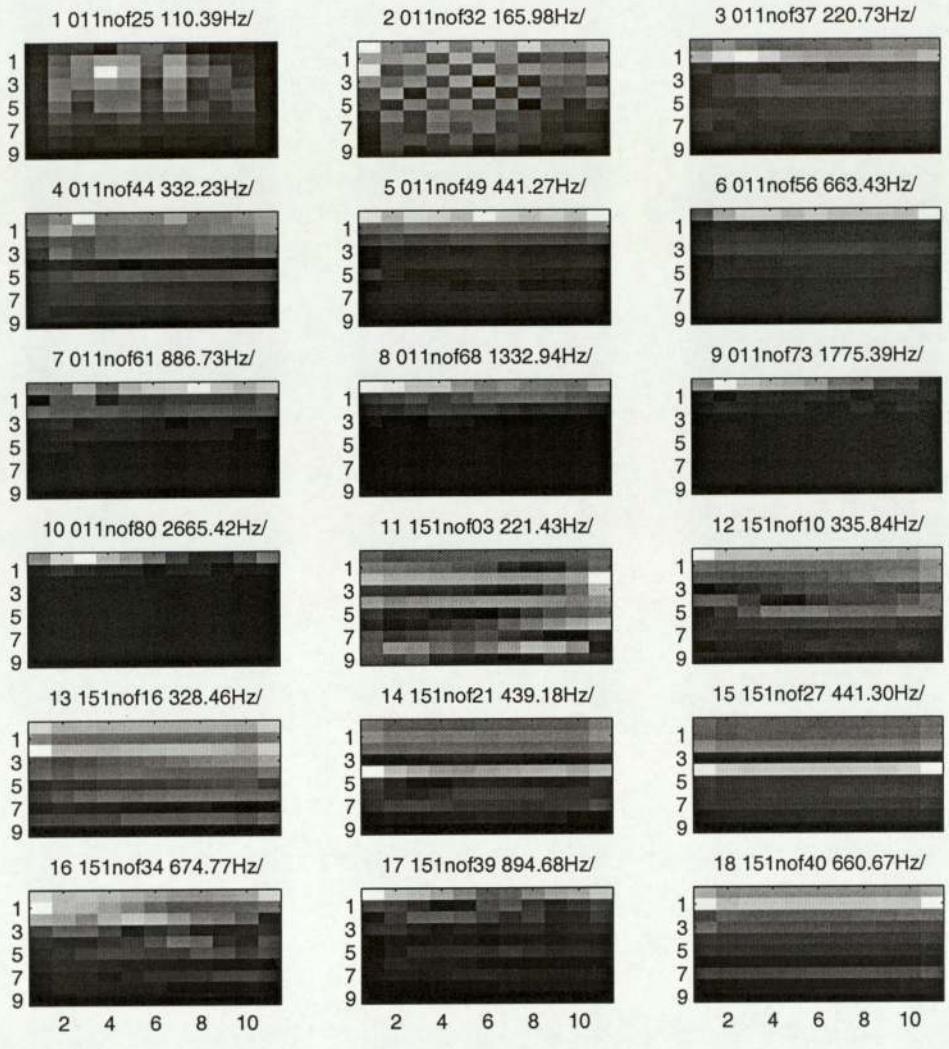

Figure 4.7: Graphical view of estimated amplitude of each partials obtained by YIN f_0 estimator for samples 1 to 18.

4.5.3 Harmonic density plot on polyphonic tone

We have performed harmonic estimation on the polyphonic tones to observe how well the model would work for the polyphonic tones providing correct f_0 can be estimated. f_0 was manually searched from the spectrum of the signal, and estimation of θ and A was performed using maximum likelihood. The result of this process to Sample 34 (a chord of E4 and B4 played by violin), are shown in Figure 4.10 together with frequency spectrum of the chord shown in Figure 4.11.

There are many peaks in spectrum where harmonics of two notes share the same frequency. This makes the modelling of harmonic density of polyphonic tones using the model shown in Equation 4.14 very difficult.

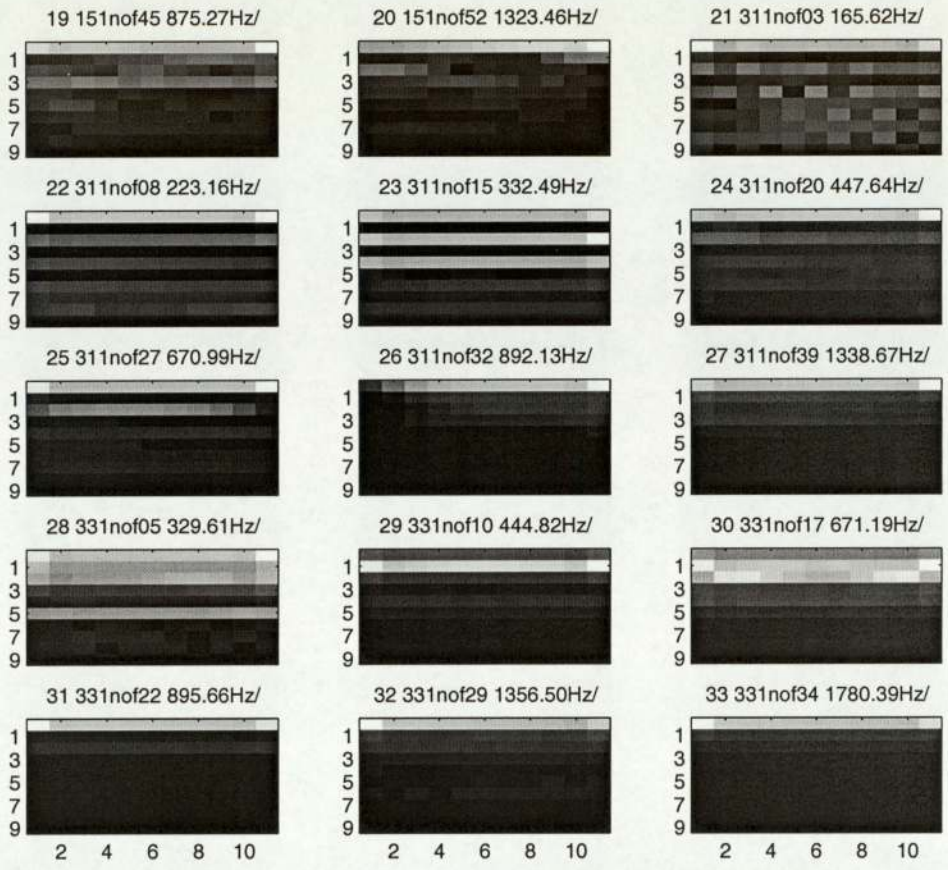

Figure 4.8: Graphical view of estimated amplitude of each partials obtained by YIN f_0 estimator for samples 19 to 33.

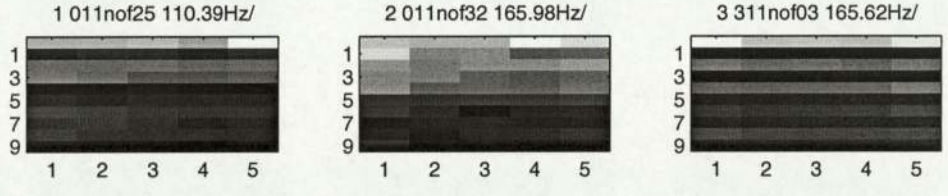

Figure 4.9: Graphical view of estimated amplitude of each partials obtained by using large window - From left, Sample1 011nof25 at 110.39Hz, Sample2 011nof32 at 165.98Hz, Sample21 311nof03 at 165.62Hz.

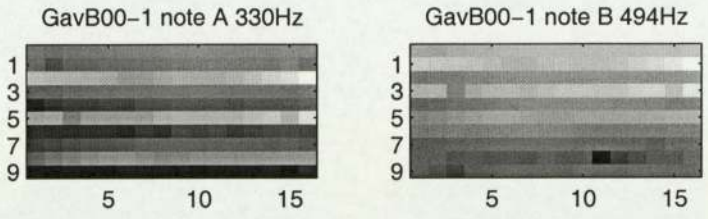

Figure 4.10: Graphical view of estimated amplitude of each partials obtained for GavB00-1, Sample 34, Violin chord E4 & B4, 330Hz & 494Hz.

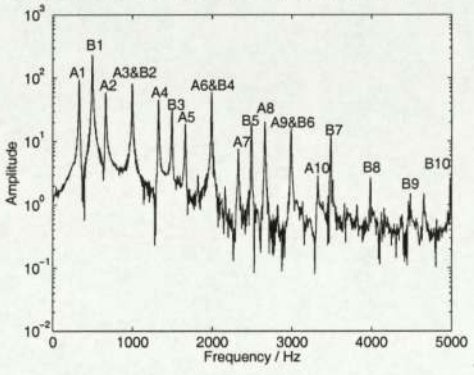

Figure 4.11: Frequency spectrum of GavB00-1, Sample 34, Violin chord E4 & B4, 330Hz & 494Hz

4.5.4 Conclusion of Harmonic Analysis

It was not possible to monitor the harmonic characteristics of most instruments. The sound production of string instruments are triggered by the oscillation of the string. For piano, it is done by hammering 3 strings that are tuned around the desired frequency. The harmonic density plots obtained for piano were very unsatisfactory. This is due to high complexity in the physical construction and the sound production of the piano. Because of its system construction, anharmonicity of piano is very well known and is necessary for the better sound production that gives the characteristics of different piano tones [5] [1]. This anharmonicity of course will be increased if the piano has not been tuned for a long time.

Another type of string instrument is a violin. The sound is produced either by plucking a string, or bowing a string. The results obtained by correctly estimated f_0 using YIN estimator showed many bright lines at lower harmonics. This suggests the contribution of many low harmonics for the production of its tone. Tonal quality of violin could easily differ by the bowing point when it is played. It can also be differed by the finger damping. However, because of its physical characteristics, inharmonicity of the violin is considerably smaller compared with the piano.

The density plot shown for the clarinet showed clear properties of woodwind instruments with a closed cylindrical air column. As discussed in Section 1, when flute is played, a skillful player can alter the strength of even and odd harmonics to produce

tones that gives different impression of the note. Even its physical property produces a tone whose harmonics are precisely at integer multiples, the model was not capable of detecting such characteristic.

In the estimation process using Markov chain sampling, for some cases, f_0 were estimated at where it was not expected. The graph shown in 4.12 reveals that in fact, the error calculated using likelihood function from Equation 4.10 were smaller in most cases than when parameters were estimated using the algorithm based on Markov chain sampling. This means that even if the right f_0 is sampled in the process that employs Markov chain sampling, it would not be accepted as a new state after sampling the f_0 that has been accepted as a final result for this experiment. Also from Figures 4.3, 4.4, 4.5 and 4.5, the recovered signal by estimated parameters shows a good fit to the original data. The model had a great flexibility to model the complexity of tones from musical instruments, thus monitoring characteristics only from amplitude parameters were found impossible.

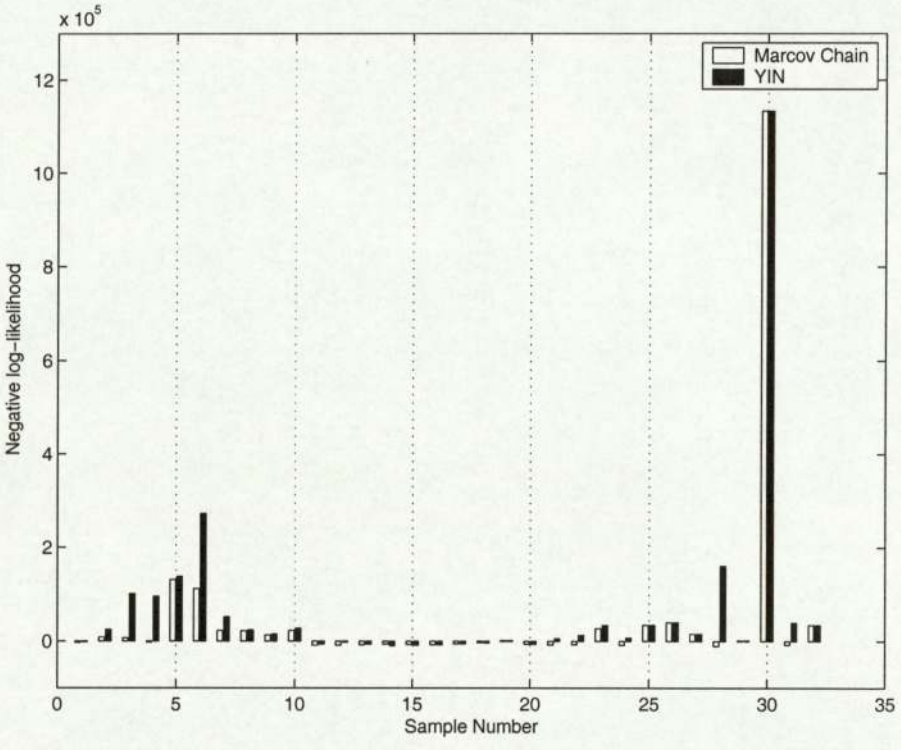

Figure 4.12: Negative log-likelihood (error) calculated for the recovered signal estimated using Markov chain sampling and YIN f_0 estimator

Chapter 5

Conclusion

First, we have discussed a method to analyse the complexity of musical audio signals using Singular Value Decomposition. The complexity of data increased as the window size of the embedding matrix increases and captures the new structure of the data. Monitoring the change in complexity, we have found that the window size of $m = 200$ is large enough to capture all the complexity needed but it is not too large, so it starts capturing unwanted low-frequency components such as non-linearity of the data.

An embedding matrix of suitable window size was constructed after the complexity analysis of matrix, and ICA was performed on the data to extract underlying sources of data. However, this method proved to be unsuccessful, and showed that embedding matrices and ICA are inappropriate for feature extraction from musical signals.

We then investigated an alternative method of feature extraction. A harmonic model based on *additive synthesis* was used to model musical signals. We have used two different approaches to the harmonic analysis of instruments tones. For the estimation of f_0, Markov chain sampling and the YIN f_0 estimator was employed. The optimisation of other parameters within the harmonic model was performed, and different relative intensity of the harmonics are shown as a density plot. For the estimation process that uses Markov chain sampling, f_0 estimation was found to be inaccurate because of the flexibility of the parameters within the model. On the other hand, YIN f_0 estimator was found to be very efficient, and produced clear density plots for the relative intensity of harmonics. However, very consistent characteristics of each instruments are observed from the results.

Although the timbre of notes differs within the same instrument depending on the pitch and how it is played, some similarities have been found within the intensity map of each instruments. The model may be extended to cater for the anharmonicity of the data and the pitch variation in notes played (e.g vibrato) to give a better estimation of the relative intensity of different harmonics. Also many more datasets should be tested for the analysis of pitch dependency of the relative intensity of harmonics. Only once this is completed estimation of the intensity of harmonics is reliable, and it may be possible to build a timbre classifier.

Bibliography

[1] J. Backus. *The Acoustical Foundations of Music.* W.W. Nortin & Company, Inc., 1968.

[2] H. B. Barlow. Sensory mechanisms, the reduction of redundancy, and intelligence. In *The Mechanisation of Thought Processes*, pages 525–559, 1959.

[3] H. B. Barlow. The coding of sensory messages. In *Current Problems in Animal Behaviour*, pages 331–360, 1961.

[4] A. J. Bell and T. J. Sejnowski. An information-mazimization approach to blind separation and blind deconvolution. In *Neural Computation*, volume 7, pages 1129–1159, 1995.

[5] H. Benade. *Fundamentals of Musical Acoustics.* Oxford University Press, 1976.

[6] D. S. Broomhead and G. P. King. On the qualitative analysis of experimental dynamical systems. In *Nonlinear Phenomena and Chaos*, Sorben Sorker Malverln Physics Series, pages 113–144. 1985.

[7] J. C. Brown. Frequency ratios of spectral components of musical sounds. In *Journal for Acoustical Society of America*, pages 1210–1218, 1995.

[8] J. C. Brown. Computer identification of musical instruments using pattern recognition with cepstral coefficients as features. In *Journal for Acoustical Society of America*, pages 1933–1941, 1999.

[9] J. C. Brown. Feature dependence in the automatic identification of musical woodwind instruments. In *Journal for Acoustical Society of America*, pages 1933–1941, 2000.

[10] M. A. Casey. Separation of Mixed Audio Sources by Independent Subspace Analysis. In *International Computer Music Conference (ICMC)*, Aug 2001.

[11] E. C. Cherry. Some experiments on the recognition of speech, with one and two ears. In *Journal for Acoustical Society of America*, pages 975–979, 1953.

[12] P. Comon. Independent component analysis, a new concept? In *Signal Processing*, volume 36, pages 287–314, 1994.

[13] P. Comon, C. Jutten, and J. Herault. Blind separation of sources, part ii: Problem statement. In *Signal Processing*, volume 24, pages 11–20, 1991.

[14] M. Davy and S. J. Godsill. Bayesian harmonic models for musical signal analysis. In J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. David, D. Hackerman, A. F. M. Smith, and M. West, editors, *BAYESIAN STATISTICS 7*. Oxford University Press, 2003.

[15] A. de Chéveigne and H. Kawahara. YIN, a fundamental frequency estimator for speech and music. In *Journal of Acoustic Society of America*, 2002.

[16] J. Foote. An overview of audio information retrieval. In *Multimedia Systems*, volume 7, pages 2–10, 1999.

[17] M. Goto, H Hashiguchi, T. Nishimura, and R. Oka. RMC music database: Music genre database and musical instrument sound database.

[18] M. Goto and Y. Muraoka. A sound source separation sysytem for percusion instruments. In *The Transactions of Institude of Electronics, Information and Communication Engineers D-II*, volume J77-D-II, pages 901–1011, 1994.

[19] P. Herrera and X. Serra. A proposal for the description of audio in the context of mepg-7. In *Proceedings of the CBMI'99 European Workshop on Content-Based Multimedia Indexing*, 1999.

[20] A. Hyvärinen. New approximations of differential entropy for independent component analysis and projection pursuit. In *Advances in Neural Information Processing Systems*, volume 10. MIT Press, 1998.

[21] A. Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis, 1999. Helsinki University of Technology.

[22] A. Hyvärinen and E. Oja. Independent component analysis: A tutorial, 1999. Helsinki University of Technology.

[23] C. J. James and D. Lowe. Extracting multisource brain activity from a single channel electromagnetic brain signals, 2001.

[24] C. Jutten and J. Herault. Blind separation of sources, part i: An adaptive algorithm based on neuromimetic architecture. In *Signal Processing*, volume 24, pages 1–10, 1991.

[25] R. Linsker. An application of the principle of maximum information preservation to linear systems. In *Advances in Neural Information Processing Systems*, volume 1, 1989.

[26] B. Logan. Mel frequency cepstral coefficients for music modeling, 2000.

[27] J. M. Martińez. MPEG-7 overview (version 8), July 2002. International Organisation for Standardisation, ISO/IEC JTC1/SC29/WG, Coding of Moving Pictures and Audio.

[28] B. C. J. Moore. *An Introduction to the Psychology of Hearing.* Academic Press, 1997.

[29] I. T. Nabney. *Netlab: Algorithm for Pattern Recognition.* Springer, 2002.

[30] G. Peeters, S. McAdams, and P. Herrera. Instrument sound description in the context of mpeg. In *Proceedings of International Comp. Music Conference (ICMC), Berlin, Germany*, 2000.

[31] L. Rabiner. On the use of autocorrelation analysis for pitch detection. In *IEEE Transaction on acoustics, speech, and signal processing*, volume 25, pages 24–33, 1977.

[32] L. Rabiner and B. H. Juang. *Fundamentals of Speech Recognition.* PTR Prentice-Hall, 1993.

[33] S. J. Roberts, W. Penny, and I. Rezek. Temporal and Spatial Complexity measures for EEG-based Brain-Computer Interfacing. In *Medical & Biological Engineering & Computing*, volume 37, pages 93–99, 1998.

[34] X. Rodet. Sinusoidal+residual models for musical sound signals analysis/synthesis. In *Proc. of the Fourth Meeting on the FWO Research Society on Foundations of Music Research: Time-Frequency Techniques and Music, Ghent, Belgium*, 1998.

[35] X. Serra. Musical sound modeling with sinusoids plus noise. In G.D. Poli, A. Picialli, S.T. Pope, and C. Roads, editors, *Musical Signal Processing.* Swets & Zeitlinger Publishers, 1997.

[36] M. Slaney. Lyon's cochlear model. Apple technical report #13, Apple Technology Group, 1988. available from the Apple Corporate Library.

[37] M. Slaney. A perceptual pitch detctor. In *International Conference on Acoustics Speech and Signal Processing*, volume 1, pages 357–360, 1990.

[38] M. Slaney. Auditory toolbox version2. Technical report, Interval Research Corporation, 1998.

[39] C. Uhle, C. Dittmar, and T. Sporer. Extraction of drum tracks from polyphonic music using independent subspace analysis. ICA 2003, Apr. 2003.

Appendix A

List of samples used

	FILE(.wav)	Instrument	Note(MIDI)	Frequency / Hz
1	011nof25	piano	A2(45)	110
2	011nof32	piano	E3(52)	165
3	011nof37	piano	A3(57)	220
4	011nof44	piano	E4(64)	330
5	011nof49	piano	A4(69)	440
6	011nof56	piano	E5(76)	659
7	011nof61	piano	A5(81)	880
8	011nof68	piano	E6(88)	1319
9	011nof73	piano	A6(93)	1760
10	011nof80	piano	E7(100)	2637
11	151nof03	violin	A3(57) on G	220
12	151nof10	violin	E4(64) on G	330
13	151nof16	violin	E4(64) on D	330
14	151nof21	violin	A4(69) on D	440
15	151nof27	violin	A4(69) on A	440
16	151nof34	violin	E5(76) on A	659
17	151nof39	violin	A5(81) on A	880
18	151nof40	violin	E5(76) on E	659
19	151nof45	violin	A5(81) on E	880
20	151nof52	violin	E6(88) on E	1319
21	311nof03	clarinet	E3(52)	220
23	311nof15	clarinet	E4(64)	330
24	311nof20	clarinet	A4(69)	440
25	311nof27	clarinet	E5(76)	659
26	311nof32	clarinet	A5(81)	880
27	311nof39	clarinet	E6(88)	1319
28	331nof05	flute	E4(64)	330
29	331nof10	flute	A4(69)	440
30	331nof17	flute	E5(76)	659
31	331nof22	flute	A5(81)	880
32	331nof29	flute	E6(88)	1319
33	331nof34	flute	A6(93)	1760

Table A.1: Single note samples used.

	FILE(.wav)	Instrument	Note(MIDI)	Frequency / Hz
34	GavB00-1	violin	E4(64) B4(71)	330 494
35	GavB02-1	violin	B4(71) A5(76)	494 659
36	GavB08-1	violin	E4(64) E5(76)	330 659
37	SicB01-1	piano & flute	n/a	n/a
38	SicB02-1	piano & flute	n/a	n/a
39	SicB02-2	piano & flute	n/a	n/a
40	MamB02-1	piano	E4(64) G5(79)	330 784
41	MamB09-1	piano	E4(64) G5(79)	330 784
42	MamB25-0	piano	C3(48) D5(74)	131 587
43	MamB25-1	piano	C3(48) D5(74)	131 587
27	MamB25-2	piano	C3(48) C5(72)	131 523
28	MamB25-3	piano	C3(48) C5(72)	131 523

Table A.2: Chord samples used.

- Each notes and frequencies stated are the notes and frequencies reported to be played, not actually recovered from the recording, thus it may differ in reality.
- For notes 11 to 20, 'on G', 'on D', 'on A' and on E' means the notes played on G string, D string, A string, and E string.
- Sample of the chords are taken from musical pieces.
  - Gav - Gavotte en Rondeau, J. S. Bach
  - Sic - Sicilienne, G. Fauré
  - Mam - Variations on 'Ah! Vous dirai-je, Maman', W. A. Mozart