Real-Time Spectral Modelling of Audio for Creative Sound Transformation

Jeremy John Wells

A thesis submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy in Music Technology

THE UNIVERSITY of York

Department of Electronics

January 2006

ABSTRACT

The spectral analysis capability of the human auditory system is crucial to its ability to extract information from audio signals. Spectral analysis, processing and modelling are concerned with the decomposition of audio signals into simpler components which can have different positions in, and may vary over, frequency and time. Modern spectral models which combine sinusoids and other signal component types offer a powerful and flexible means of changing sounds in perceptually meaningful and acoustically plausible ways. However, whilst many of these offer real-time interaction during modification of, and resynthesis from, model data, real-time analysis for such models has received relatively little attention from researchers or designers. This thesis examines the possibilities for real-time spectral modelling on available hardware using a combination of Fourier and wavelet techniques.

Two specific areas of analysis are addressed. Firstly, single-frame high accuracy description of stationary and non-stationary sinusoids by the use of time-frequency reassignment data and the derivation of sinusoidality measures from such analysis is described and compared with an existing single frame approach. Secondly a complex B-spline wavelet analysis system for audio signals is devised, which offers estimation of component magnitude, frequency and bandwidth, for use with parametric equalisers at resynthesis.

These novel methods are then combined in a frame-by-frame "sinusoidal plus residual" spectral analysis, modelling and resynthesis system. This heterogeneous system performs all of its resynthesis in the time domain on a sample-by-sample basis whilst offering control over the mean instantaneous frequency of all of its components. In its current implementation the system executes at speeds very close to real-time. Whilst not all audio signal types are successfully modelled, the results obtained demonstrate that frame-by-frame spectral modelling, using techniques developed in this thesis, is possible for a range of sounds.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

To Gran, a wonderful person who is an inspiration to all of us.

ACKNOWLEDGEMENTS

There are many people without whose support and encouragement I would not have been able to undertake and complete this thesis. Firstly I would like to thank my supervisor, Dr. Damian Murphy, for his guidance, encouragement, time and patience as this research has progressed. In particular his recent 'out of hours' support is very much appreciated. My thesis advisor, Dr. John Szymanski, has also been a great help and someone who has always had the time, no matter how busy, to chew over the time-frequency cud. I am hugely indebted to the department as a whole for giving me this and so many other opportunities. My former teachers, now colleagues, in both the electronics and music departments have also offered a great deal of encouragement and illumination and I'm grateful for this and to be part of such a friendly, helpful and talented team. To everyone in the audio lab (past and present) thanks for making it such a pleasant and fun place to work and, now that I've moved out, for always letting me know when the coffee's on!

Much of what I have learned since I decided to do music and technology for a living has come from a long line of highly talented musicians, audio professionals and teachers that I have had the privilege to work with and/or learn from: Kevin Edge, Michael Howard, Tony Starr, Dave Fisher, Francis Rumsey and James Jackman. What I have learned from you has no doubt seeped into these pages and they will be all the better for it. As great friends and former housemates I also owe Kev and James a lot of great times and just as much fantastic music.

Thanks also to all those people whose friendship has kept me sane over the last few years and have generally been around for me: Andy and Anna, Suzanne, James and Liz, Merl, the Guildford and London posses and everyone at home as well as the purple people who have given me a lifetime's worth of bad jokes, incredible music and lots of lovely hot air to keep me warm.

Thanks too to Jasper for being such good company and reminding me at least twice a day that the great outdoors is a much better place to think than sat in front of a computer screen!

Finally, and most importantly, the people without whom none of the past years would have happened: my family. Without the love, support, encouragement, and practical help from my parents and my brother and sister none of this would exist. I will always be grateful for what you have done and continue to do for me. My only regret about the last five years is that they have been so far away from home.

DECLARATION

I hereby declare that this thesis is entirely my own work and all contributions from outside sources have been explicitly stated and referenced.

Although not forming the main body of work presented here, reference is made in the text to previously presented research carried out during the preliminary stages of work for this thesis. These publications are listed as follows:

1. J. Wells and D. Murphy, “Real-Time Partial Tracking in an Augmented Additive Synthesis System”, *Proceedings of the 5th International Conference on Digital Audio Effects (DAFx02)*, pp. 93-96.

2. J. Wells and D. Murphy, “Real-Time Spectral Expansion for Creative and Remedial Sound Transformation”, *Proceedings of the 6th International Conference on Digital Audio Effects (DAFx03)*, pp. 61-64.

1 OVERVIEW OF THESIS

1.1 Introduction

This chapter gives a general overview of the research and application context of this thesis. A hypothesis is stated from which practical aims and objectives for the work undertaken and reported here are described. An overview of the structure of this document, along with the reasoning behind aspects of content and style, is also given.

1.2 Spectral modelling for creative sound transformation

Ever since the advent of technology for capturing, storing and replaying acoustic signals, either as a direct physical analogue or as series of discrete numbers, musicians have sought to manipulate or create sound in such forms as part of compositional and creative processes. This has gone hand in hand with acoustic and psychoacoustic research into the nature of sound and music and human responses to it. In addition to this exploration into the nature of sound, its perception, and how both can be moulded through creative processes, there have been continuous improvements in the technologies and techniques available for such purposes. Music Technology, as this general field has become known, is therefore truly multidisciplinary (some would say transdisciplinary). It incorporates aspects of the traditional disciplines of music, engineering, physics, psychology and computer science and, of course, these disciplines themselves incorporate others such as mathematics, biology and chemistry. Whilst no discipline is truly isolated from all others, music technology is perhaps one of the least ‘compartmentalised’.

Both physiological and psychological investigations of the human auditory system have found that a key means by which we are able to extract information from audio signals is through *spectral analysis*. In other words we are able to describe a complex pattern of vibration in terms of the summation of simpler patterns of vibration which occur at different spectral positions (frequencies). Therefore if information is contained in the spectral ‘layout’ of sounds, a valid tool for communication via sound is one which offers access to this spectral map and the individual components it contains. Spectral modelling for audio aims to provide a meaningful description of sound in terms of constituent components which occupy particular positions in frequency and, commonly, time.

Many analogue electronic and computer based tools exist for the spectral description and processing of sound. Many of these, such as the phase vocoder, enable acoustically plausible, as well as fantastic, transformations of existing sounds such as the scaling of duration and pitch independently of each other. However, the phase vocoder model of sound as overlapping grains with magnitude and phase is not wholly compatible with the common conception of sound as the combination of spectral elements which are continuous in time and have time varying frequency, magnitude and bandwidth. Spectral modelling systems attempt to infer this latter conception of sound from spectral data in order to provide a more intuitive and flexible description which is more amenable to a wide variety of perceptually meaningful transformations.

1.3 Investigation of real-time spectral modelling for creative sound transformation

1.3.1 Motivation for work contained in this thesis

Whilst tools exist for modification of, and synthesis from, spectral models in real-time little attention has been paid to the *creation* of models in real-time. In fact there seems to be an acknowledgement, or assumption, in the literature on this subject that modelling of sound, be it physical or spectral, is an inherently non-real time process: spectral processing may be performed in real-time, or quasi real-time, (using the phase or channel vocoder for example) but deriving a model cannot. As far as spectral modelling is concerned, and in the opinion of the author, this is at odds with how we as humans perceive, and infer meaning from, sound. When we hear a time stretch of an audio signal, which we believe is a poor approximation to what we would expect from an acoustic source, we are basing this expectation on how we imagine the time stretch should actually sound. In one sense that ‘imagination’ is the product of a model of the behaviour of how such a sound should behave which we are able to construct as we hear it.

Tools exist within recording studios for capturing, generating, storing and processing sound. Tools for processing in many studios, particularly before the advent of computer based hard disk recording of digital audio, process audio in real-time; they have audio input and output connections but no, or only short-term temporary, facilities for storage. Where spectral modelling is used for audio processing it must currently be performed ‘offline’. Whilst this ‘rendering’ paradigm may be suited to the needs of some musicians it may well be, and probably is, constraining for others. A primary motivation for the work contained in this thesis is to question whether this constraint need exist.

Fourier analysis has been a ubiquitous tool for the spectral analysis and processing of audio signals with computers. Other methods such as Gabor and Walsh-Hadamard analysis and time-frequency energy distributions have been applied but far less often. However, since the late 1980s a new analysis technique, that of wavelets, has emerged which offers certain potential advantages over Fourier analysis. As, through considerable research effort, understanding of wavelets and their applications has grown so has interest in them for audio processing and modelling applications. Fourier has a longer history and so many of its associated techniques, and understanding of them, are more mature. Many audio systems use either a wavelet or a Fourier approach. It is this author's contention that no one analysis technique offers a panacea for all analysis and modelling problems and that combinations of such techniques should be investigated.

1.3.2 Statement of hypothesis

> *Wavelet and Fourier analysis methods can be combined to provide a meaningful and flexible parametric spectral modelling system that can be used as a real-time audio processor for monophonic sources. In particular, by high-accuracy modelling of their parameters (including those of non-stationarity), sinusoids can be identified and tracked from frame to frame and complex wavelets can be used to produce a model of the residual which is time-variant and can be synthesized in the time domain.*

This hypothesis is examined in three ways:

1. Through the development and testing of an algorithm for estimating the mean amplitude and frequency, as well as the linear frequency change and exponential amplitude change, that a spectral magnitude peak exhibits in a single analysis frame and determining whether the behaviour of its parameters are consistent with that of a stable sinusoid.

2. The adaptation of B-spline wavelets to a partially decimated complex transform technique and investigation of this transform's analysis properties for different audio signal types.

3. The combination of these two novel analysis methods in a frame-by-frame spectral modelling system for monophonic audio sources and their potential application in a real-time system.

1.3.3 Thesis structure

No discipline knows more than all disciplines. [Taddei, cited in Nowotny]

The remainder of this thesis is concerned with the investigation of the hypothesis given in the previous section. In order to do this the work undertaken is first placed in the broad context of sound, music and the application of technology to both of these. Since music technology is multidisciplinary a readership from a wide range of backgrounds and with differing knowledge of the areas covered is anticipated. For this reason an extensive overview of the motivation for computer based creative sound transformations is given along with an extensive coverage of the existing literature in this field. Initially a ‘wide angle’ view is taken of the subject area followed by a more detailed ‘zoom view’ of the specific areas of time-frequency analysis and spectral modelling. This is followed by a description and evaluation of the novel methods investigated in order to test the hypothesis. Finally the results and conclusions are summarised and directions for future work are considered.

Chapter 2 summarises methods of describing and representing sound and music. An overview of the technologies employed to test the hypothesis is given and how the work and knowledge presented in this thesis might contribute to the field of digital music research. Key scientific and technological aspects of this thesis and their musical and cultural imperatives are introduced and explained.

Chapter 3 describes methods for generating, analysing, modelling and processing digital audio signals. It is in this chapter that the underlying analysis and modelling paradigms which underpin the novel work in later chapters are explained and mathematically described. Spectral modelling and Fourier and wavelet analysis are the subject of detailed study and are also placed in a general context of audio signal processing and modelling methods.

Chapter 4 describes novel techniques for identifying sinusoids and estimating their parameters, stationary and non-stationary, from reassigned Fourier analysis data. These methods are compared to existing methods. This system is intended to function as a high-accuracy analysis system for frame-by-frame sinusoidal modelling.

Chapter 5 gives a detailed overview of a new complex wavelet system which uses B-spline wavelets as an approximation to Gabor grains to provide a ‘constant-Q’ time-frequency analysis of audio signals. This system aims to provide data for a model of spectral components which can be used to control parametric equalisers in real-time. This system offers control over the amount of decimation that is performed on data and so the trade-off between ‘over completeness’ and computational cost is examined.

Chapter 6 puts the work of the previous two chapters in the practical context of a spectral analysis and resynthesis system and examines the possibility for real-time spectral modelling of sound using such a system. An overview of the specific system is given followed by examination of its performance, both in terms of the quality of the resynthesized output and the computational cost of producing it.

Finally chapter 7 summarises key results and conclusions from the previous three chapters and considers them in relation to the hypothesis. Directions for future work which might further expand on the knowledge presented in the thesis and improve upon the techniques and system described are given.

1.3.4 Contribution to knowledge in spectral analysis and modelling

The author believes that work described within this thesis contributes to knowledge of techniques for spectral analysis and modelling and their applicability to real-time implementations. The key novel areas studied and reported are:

- The possibilities for high accuracy estimation of sinusoidal parameters using reassigned short-time Fourier data and an iterative method for reducing the influence of intra-frame frequency and amplitude change upon the estimation of each other.

- The use of ‘estimated parameter behaviour’ as a test of sinusoidality on a frame-by-frame basis and how it compares with other tests which do not assume stationarity in their sinusoidal models.

- The application of B-spline wavelets for audio signal modelling in general and estimation of component centre frequency, magnitude and bandwidth in particular.

- The potential for a heterogeneous real-time spectral modelling system which uses a combination of Fourier and wavelet analysis and whose output is synthesized wholly in the time domain.

1.4 Summary

An overview of the work investigated for this thesis as well its context and motivation has been presented in this chapter. A hypothesis, whose testing underpins all of the work presented in the following pages has been stated. The structure of this document and an overview of how it presents this work and the literature describing the existing research on which it builds has been given. The contributions to knowledge in the field of spectral analysis and processing in general, and of real-time spectral modelling in particular, have been summarised.

2 REPRESENTATIONS OF SOUND AND MUSIC

If this word 'music' is sacred and reserved for eighteenth- and nineteenth-century instruments, we can substitute a more meaningful term: organisation of sound. [Cage, 1961]

2.1 Introduction

This thesis deals with applications of quasi real-time analysis, modification and resynthesis of audio. The modification of audio can be seen as an attempt to organise its structure to create some kind of meaningful perceptual effect in those listening to it. A common modern definition of music is 'organised sound' and so with this definition in mind the broad purpose of the modification of audio using the tools described in this thesis is the creation of music. Therefore this thesis is fundamentally concerned both with sound as a physical and perceptual phenomenon and music as the organisation (or re-organisation) of sound to produce or contribute to "a pattern of sounds made by musical instruments, singing or computers, or a combination of these, intended to give pleasure to people listening to it" [Cambridge Advanced Learner's Dictionary].

This chapter reviews current theories on sound and music as they relate to the contents of this thesis. The first part of the chapter discusses the production, transmission, and manipulation of sound itself and audio signals and data which are used to represent it. The second part deals with issues related to the creation (writing and performing) of music and the manipulation of audio signals and data within electronic and computer based systems for the composition or performance of music.

2.2 Sound and its representation

2.2.1 Sound and sensing sound

If a tree falls in the woods and there is no-one there to hear it
does it make a sound? [unattributed]

The above question (which may, or may not, be a zen koan) illustrates the dual nature of sound in that it is both an acoustic and psychoacoustic phenomenon. For the purposes of this discussion the answer is 'no' since the definition of sound used here is that of "an acoustic event or events perceived by the human auditory system". The acoustic event may be the falling of a tree, the clapping of hands or the movement of a loudspeaker diaphragm. To be acoustic an event must produce variations in the pressure of the matter surrounding the object (usually air) which then propagate as a wave.

As these variations in pressure emanate outwards from an object the intensity of these variations will reduce since the power generated by the source is spread over a greater area. Under certain conditions sources can be considered to occupy a single point (if they are very small for example) from which sound waves will propagate equally in all directions. The intensity of a sound wave for a given distance x from a point source is the power of the source distributed over the surface area of a sphere whose radius is this distance. Therefore the intensity I (in Watts/m^2) of the sound at a distance x from a point source of power W is given by:

$$I = \frac{W}{4\pi x^2} \qquad (2.1)$$

This means that the intensity of sound waves propagating in the free field varies in inverse proportion to the square of the distance travelled. The sound intensity will also be further reduced if energy is dissipated in the medium through which it travels, however this loss of energy will almost certainly be frequency dependent.

As sound propagates through the volume surrounding the source any objects in its path will move in sympathy with the variations in fluid pressure to a greater or lesser extent. Objects with a low mass and a large surface area in the plane of propagation will move more than objects with high mass and a low surface area. The ear canal begins at the outer ear and ends at a lightweight, thin and taut membrane which is the ear drum. This moves in sympathy with variations in the pressure of air (or water) in the ear canal and these movements are passed on via the sympathetic movement of a chain of three bones, called the ossicles, to the oval window. It is via the oval window that the movements of these bones are transferred to the fluid in the cochlea of the inner ear. As the fluid in the cochlea moves a travelling wave motion is set up in the basilar membrane which runs its length. As the basilar membrane moves hairs within the Organ of Corti, which sits on top of membrane, also move. It is their movement which excites associated nerve fibres and it is these nerve fibres which carry auditory information to the brain. The extent of the movement of the hairs determines how many nerve fibres are excited by the stimulus and so there must be a certain level of sound energy present to cause auditory information to be sent to the brain. This means that below a certain threshold of intensity, variations in pressure at the opening of the ear canal will not cause sufficient stimulation of the auditory nerve fibres and there will be no sense of sound in the listener [Pickles, 1988]. Since the intensity of sound diminishes as it propagates there will be a distance between source and listener (for a given source power) beyond which the sound cannot be perceived. Perhaps a more precise form of the question would be "if a tree falls in the woods and there is no-one there to hear it, is sound sensed?" In this thesis the distinction is made between sound as an acoustic phenomenon and the sense of sound via the human auditory system as a psychoacoustic phenomenon. Sound is generated by the relative movement of object and fluid, it propagates through that fluid as variations in pressure, it is conducted via bone and fluid in the middle and inner ear where it causes movement of hairs which trigger electrical impulses in the associated nerve fibres which transmit the 'sense' of the sound stimulus to the central nervous system. This distinction between acoustic and psychoacoustic sound, between how sound is produced and how it is sensed is an important one when considering different approaches to sound modelling that are discussed in the next chapter.

2.2.2 Limits of auditory perception

In the previous section the basic properties of sound and the means of its perception in humans were introduced along with a limitation to this perception: an intensity threshold below which sound cannot be perceived. Sound intensity is the power (rate of flow of energy) per unit of area and this power manifests itself as variations in air pressure. Since sound is produced in an atmosphere which exerts its own continuous pressure (atmospheric pressure) sound power causes deviations from this pressure and it is variations in the net pressure on an object in a soundfield which cause it to move. Like pressure, the amplitude of pressure variations, is measured in Newtons/m^2 (also known as Pascals). Since this net pressure can vary rapidly between positive and negative values it is usually measured by taking the root mean square of all (or regularly sampled) instantaneous values over a given time interval. The intensity of a sound at a given point is given by the square of the deviation from atmospheric pressure. Therefore for a source of given power, as the distance from this source increases so the sound pressure level P decreases according to:

$$P \propto \frac{1}{x} \quad (2.2)$$

The minimum pressure deviation that can be detected by the human auditory system is approximately 20 μPa and this level is referred to as the threshold of hearing. The upper limits of human hearing in terms of sound pressure level (SPL) are the threshold of feeling and threshold of pain. The threshold of feeling is often quoted in the literature as being approximately 20 Pa which is a pressure level 10^6 times that of the threshold of hearing. Taking the threshold of feeling as the upper limit of comfortable listening and the threshold of hearing as the lower limit we have a range of SPLs, expressed in decibels, of 120 dB. The decibel is defined as

$$10\log\left(\frac{W_1}{W_2}\right) \qquad (2.3)$$

where W_1 and W_2 are power values. Since sound power is proportional to the square of the pressure amplitude the decibel is also defined as

$$20\log\left(\frac{P_1}{P_2}\right) \qquad (2.4)$$

Where P_1 and P_2 are pressure variation amplitudes. The decibel is a common way of representing sound pressure levels with the threshold of hearing being the reference (zero) point. So the threshold of hearing is at 0 dB SPL and the threshold of feeling is 120 dB SPL. It should be noted that the threshold figure of 0 dB SPL, 20 μPa, is only an approximation of the threshold of hearing in humans and the actual threshold varies from person to person and with frequency. In fact the actual threshold of hearing in healthy humans is closer to 10 μPa at frequencies where the ear is most sensitive [IS0226, 2003].

One aspect of the perception of sound that the dB SPL measure does not take account of is the *rate* at which the instantaneous pressure fluctuates about atmospheric pressure. As sound is the variation in net pressure so the frequency of sound is the rate at which that variation fluctuates between negative and positive, a single cycle being the time taken for one negative and one positive excursion. The period of the sound is the duration of the cycle and its frequency is the number of cycles which occur in a single second (measured in Hz). It is widely held that a young adult human can perceive variations in air pressure which have a frequency of between 16 Hz and 20 kHz (provided the sound is of sufficient intensity) via the auditory system [Moore, 1997]. More recent research suggests that the physiology of the human brain is altered when exposed to sounds which are higher in frequency than the upper limit of 20 kHz [Oohashi et al, 1991]. It is certainly the case that very intense variations in pressure below 16 Hz can be experienced as vibration by the touch senses.

2.2.3 Discrimination between audio components

Meaningful variations in pressure, or any other measurable physical property of an object or a medium, can often be broken down into combinations of simpler variations. Indeed, the identification and description of such patterns is one of the main areas of investigation within this thesis. Within audio the purpose of this ‘decomposition’ of sounds (or the signals that represent them) is often to identify perceptually meaningful elements, or groups of elements, which can be modified to effect a perceptually meaningful transformation of that sound. Another application is the efficient storage of audio as data by producing an invertible representation of the audio which is as sparse as possible. These two applications are combined in entropy based lossy compression of audio signals where the representation is made sparse by the removal, or reduction in resolution, of components that are considered less perceptible than other components [Watkinson, 1999]. Very often components that would be quite audible if heard in temporal or spectral isolation from each other cannot be perceived when heard in combination with other, proximate and more dominant, components. To return to our original example “if a tree falls in the woods yet the sound it makes is inaudible due to the sound made by a larger tree falling at the same time, does it make a sound?”. This section provides a brief overview of how the human auditory system is, and is not, able to discriminate between different audio components.

A pattern of variation around a central point or value (i.e. an oscillation) which has a straightforward mathematical description and many physical manifestations in the world around us is described by the sine function:

$$x(t) = A\sin(2\pi ft + \phi) + X \qquad (2.5)$$

where $x(t)$ is the instantaneous value of the function at a point in time t, A is the overall amplitude of the variation, f is the number of oscillations per second, ϕ is the point during the cycle reached at a given time instant (the phase offset) and X is the offset of the mean value of the function from zero. When measuring air pressure in a sound field, X would be the average atmospheric pressure although this is commonly omitted in the literature . The cosine function, which is the sine function shifted by $\pi/2$ radians can be described by the following:

$$x(t) = A\cos(2\pi ft + \phi) + X = A\sin\left(2\pi ft + \pi/2 + \phi\right) + X \qquad (2.6)$$

In this thesis sine and cosine functions are defined as being these functions with no phase offset ϕ whereas sine or cosine functions with a non-zero phase offset are defined as being 'sinusoidal'. Single instances of such functions are often referred to as a 'simple tone', or just a 'tone' and this is the definition of tone which is used in this thesis. Sounds comprised of a combination of tones are often referred to as complex tones but use of this label is avoided in this thesis to avoid confusion with complex numbers.

The location of the region (or regions) of excitation along the basilar membrane depends on the spectrum of the sound causing the excitation. The spectrum of a sound (or of any pattern of vibration) can be described by the amplitude and phase of individual sinusoidal vibrations that, when combined, will produce the same pattern of vibration, although certain types of sound would require an infinite number of sinusoids. Therefore the basilar membrane acts as a mechanical spectrum analyser [Pickles, 1988]. A single, time-invariant sinusoidal function will produce the most localised region of excitation on the membrane. The width of the region of excitation is also determined by the intensity of the sound at the ear so a sinusoidal function that is relatively low in level will produce a narrower region of excitation than a sinusoidal function that is higher in level [Plack, 2005]. The centre point of the excitation depends on the frequency of the vibration. The region of excitation caused by a sinusoidal vibration is known as a critical band, its width is the critical bandwidth and its shape is the excitation envelope. Numerous studies have attempted to determine how the critical bandwidth varies with centre frequency for tones at the same level. The more recent studies have used a measure known as the equivalent rectangular bandwidth (ERB) which is the width of a rectangular filter which has the same peak level and which passes the same total power for a white noise input[1]. This relationship is described by:

$$ERB = 24.7\left(\left(4.37\times10^{-3} f\right)+1\right) \qquad (2.7)$$

[1] White noise, discussed in more detail in the next chapter, is noise whose magnitude is independent of frequency.

where the ERB and frequency (*f*) are given in Hz [Glasberg and Moore, 1990]. A plot of frequency versus ERB derived from this equation is shown in figure 2.1.

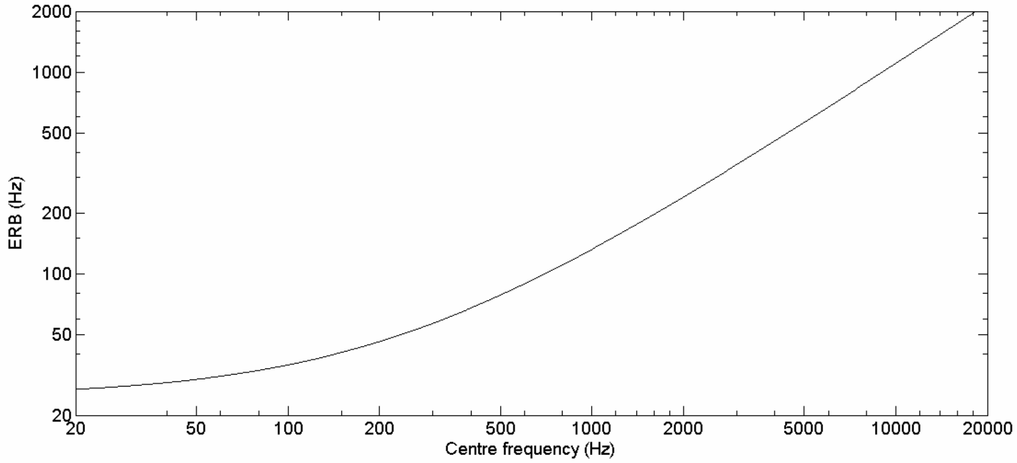


Figure 2.1: Relationship between centre frequency and equivalent rectangular bandwidth (ERB) of auditory filter.

If a sound is comprised of two sinusoids of similar intensity, which are very close together (i.e. much less than one critical bandwidth apart) then they will be fused together by the basilar membrane and will be perceived as a single tone. If they are close in frequency (up to approximately 15 Hz difference between the two) then this single tone will appear to undulate in intensity as the individual excitations caused by the two components will constructively and then destructively interfere with each other as their relative phase constantly varies over time. Above 15 Hz the amplitude modulations can no longer be explicitly perceived and the sense is of a 'rough', single component sound. As the frequency difference between the two components increases this sense of roughness continues but at the point where the combination of the two excitation envelopes yields two distinct peaks then two distinct components are heard. The critical bandwidth is defined as the frequency difference between two components at the point when the sensation caused by the combination of these two tones changes from 'rough' to 'smooth' [Watkinson, 1999].

So far the perception of two tones that are similar in frequency and intensity has been discussed. Where one tone is significantly higher in intensity than the other then the dominant tone will tend to mask the second tone. Masking is the decreased audibility of one component of a sound in the presence of another component [Watkinson, 1999]. The masking level is defined as the amount by which the threshold of audibility for one sound component is raised by the presence of the masking sound component [American Standards

Association, 1960]. Since the excitation envelope of the basilar membrane is not symmetrical neither is the masking pattern created by a component. The masking effect of a component decreases more slowly with increasing (logarithm of) frequency than it does with decreasing frequency, a situation known as the 'upward spread of masking' [Plack, 2005]. For example for a narrowband masking stimulus of level 80 dB SPL centred at 410 Hz the masking level of audibility of a pure tone at 800 Hz is approximately 45 dB whereas at 200 Hz this level is less than 10 dB [Egan and Hake, JASA, 1950].

The critical bandwidth for a given frequency and intensity is not the same as the resolution of the frequency discrimination of the auditory system. Frequency discrimination is the ability to differentiate between two tones of the same level. This is measured as the ability to discriminate between two successive tones with different frequencies (a measure known as the difference limen, or just noticeable difference, for frequency) or as the ability to discriminate between two tones, one unmodulated and the other slightly modulated (known as the frequency modulation detection limen). Experimentation has shown that the just perceptible difference in frequency between two tones is approximately 30 times smaller than the critical bandwidth. Although some models of the auditory system assume that the basilar membrane acts as a bank of filters with fixed frequencies, there is no evidence that this is actually the case and it is more appropriate, where possible, to view this system as performing a continuous frequency analysis with finite output resolution [Moore, 1997] [Watkinson, 1997]. This is a primary motivation for modelling of sound with time varying oscillators and noise filters in this thesis. A general overview of frequency analysis systems is given in the next chapter of this thesis.

It is not just proximity in frequency than can cause one component of a sound to obscure another. Temporal, or non-simultaneous, masking occurs when the audibility of a component is diminished due to the presence of a second component that is present either just before or just after, but not during the first component. Forward (pre-) masking, where the masking component precedes the masked component, should not be confused with auditory fatigue, which is the shift in hearing threshold after a relatively loud stimulus has been applied for some time and then removed, and auditory adaptation which is a change in the perceived intensity of a tone which has been continuously presented. Backward (post-) masking, where the masked component precedes the masking component, is not currently well understood and in listening tests has produced variation in results according to the experience of the test subjects. Indeed highly practiced subjects demonstrate negligible backward masking [Moore, 1997] and recent research has suggested that this may be related to learning difficulties since people with language difficulties often exhibit “significant elevated thresholds for this [masking] paradigm” [Roth et al, 2001].

Pre-masking occurs for maskers which are usually of just a few hundred milliseconds duration and the effect is limited to signals which occur within about 200 ms of the masking component reduces. The amount of masking increases as the masking component increases in intensity and/or the time interval between the cessation of this component and the masked component. A short duration 4 kHz tone will have its threshold of audibility raised by 40 dB 17.5 ms after the cessation of a broad band masking component with a spectrum level[2] of 50 dB. The same tone will have its threshold raised by just under 10 dB 37.5 ms after the cessation of a masker with a spectrum level of 0 dB.

This section has outlined the limitations in the ability of the ear to discriminate between sound components which occur close to each other in either time or frequency. As will be discussed in the next chapter this ‘blurring’ of multiple components into a single perceived component, or the obscuring of one component in the presence of multiple components is a facet of any time-frequency analysis system.

2.2.4 Analogies of audio

Since sound is of finite intensity which diminishes with distance and the human auditory system has finite sensitivity the communication of sound over long distances, or through dense materials, requires technological intervention. To allow sound to propagate over long distances it must be converted to a form of energy which can propagate over such distances and/or be amplified. Transduction is the process whereby energy is converted from one form to the other. The production and sensing of sound has a number of transduction stages as the energy is converted from mechanical (sound production) to acoustic (sound propagation) to mechanical (transmission through the middle and inner ear) to electrical (nerve impulses sent to the brain). Inserting additional transducers into this signal chain can extend its useful length.

[2] This is defined as the level of sound measured in a 1 Hz wide band (the intensity density) and expressed in dB relative to 20 μ Pascals (0 dB SPL).

If a diaphragm connected to an electro-acoustic transducer is placed in the soundfield then the movement of diaphragm caused by the variations in pressure will cause a varying electrical current to be produced whose varying property (or properties) is an analogue of the acoustic signal. The output of this transducer is an electrical signal which represents the movement of the diaphragm which in turn represents the acoustic signal (i.e. the sound). This arrangement of a moving diaphragm with transducer which produces an electrical analogue of the acoustic signal is known as a microphone. The inverse arrangement, where an electrical signal produces movement of a diaphragm which produces an acoustic signal, is called a loudspeaker. It is common for the time-varying voltage of a signal to represent the time-varying amplitude of the acoustic signal [Templaars, 1996].

Since sound is time-varying pressure it is, by its very nature, transient. In other words it has no physical permanence and it cannot be recalled by subsequent analysis of the media that sound or its analogues it pass through. As well as spatially dislocating sound from its source it is often desirable to temporally dislocate it so that it can be heard at a different time or repeated. This requires a permanent change to be made in a material which will represent the entire sequence of variations which comprises the time-varying signal. In other words the signal must be applied to a material that has memory. Since sound can be considered as variations in pressure over time, a spatial dimension in which to store these variations is required. A magnetic material is one which can respond to, and produce, magnetic fields. Magnetic tape consists of a number of particles of a magnetic material covering a long thin strip along which variations in an applied magnetic field can be stored. This storage is possible since the reorientation of the particles caused by the presence of the field is partially retained when the field is removed. The field is introduced to and removed from successive parts of the tape by moving it past the source of the field which is an electromagnet whose applied voltage is an analogue of the sound signal to be stored. Thus variations of the signal in time are stored as variations in the magnetic field along the tape. In order to reproduce what is stored on the tape it must be passed close to a similar electromagnet so that the varying magnetic field caused by the movement of the tape causes an alternating current to be produced in the coil of the magnet which is a time varying analogue of the original sound signal which, after amplification, can be converted into an acoustic signal again by a loudspeaker. Since the position in time of the audio signal is directly related to its position along the length of the strip, magnetic tape is referred to as a linear access medium [Jorgensen, 1995].

Another common method of storing audio signals is to etch a groove into a medium, usually vinyl plastic, whose shape is an analogue of the original sound signal. In order to ‘remember’ the medium should be made from a solid material (i.e. one that retains its shape). This single groove is recorded in a spiral from outside towards the inside of a circular disk of the recording medium. The information is read from the groove via a stylus connected to a mechanical to electrical transduction mechanism (usually a coil moving in relation to a magnet) which travels through the groove as it rotates on a turntable. This is a non-linear access medium since different positions along the groove can be accessed by moving the stylus across the spiral rather than through it. Different parts of the audio signal can be accessed much more quickly with a non-linear, as opposed to linear, access medium [Davies, 1996]. A third method is to record sound signals as the variations in the area and/or luminance of a photographic image and this is commonly how soundtracks to motion pictures are stored with the varying intensity of light showing through the ‘sound image’, placed lengthways alongside the still images for the video, being the analogue of the amplitude variations of the sound [Hartstone and Spath, 1996].

2.2.5 Channel bandwidth and dynamic range

A channel is a path through which a signal can pass or in which it can be stored. It may be a volume of air, a length of copper cable or a groove cut into a vinyl disk. Since the signal is transferred/stored as the analogue of one, or more, of the physical characteristics of the channel, the faithfulness of this analogue to the original signal is directly related to such characteristics. As examples these characteristics may be the elasticity of the material into which the signal is etched, the coercivity of the magnetic particles whose orientation represents the signal or the resistance of the copper cable. It is also the interaction between medium and transducer that defines the channel so channel quality might also be dependent on the velocity of the stylus in the record groove or of the tape that moves past the electromagnet. As well as its *physical* characteristics a channel, or any system, may be described in terms of the difference between a signal input to it and the altered signal at its output. When these properties change over the time the channel or system is described as time-variant, when they do not change over time it is described as time-invariant [Lynn and Fuerst, 1994].

Two important properties of a channel are its dynamic range and bandwidth or, a single property that combines these two, its frequency response. A linear system is one in which changes in the input level cause a proportional change in the output level. The dynamic range of a channel is the ratio of the intensity of the noise produced by the channel when no signal is present to the greatest signal intensity that can be recorded and reproduced from the channel without non-linear distortion of the signal. Since no channel offers absolutely perfect linearity, reproduction of any signal at any level will always introduce some distortion to the signal so measures of maximum level are normally taken once the level of distortion has exceeded a specified threshold. For example for magnetic tape the maximum output level (MOL) is determined as the level at which the third harmonic distortion reaches 3%[3]. Noise is measured as the output level from the channel when no signal is present in it. There are two other terms used in audio which are related to dynamic range: headroom and signal to noise ratio. The headroom of a channel is the difference in level between the MOL and the nominal operating level, which is the "design target signal level of audio circuits" [Convention Industry Council, 2004]. The signal to noise ratio (SNR) is the difference between the noise level and the nominal operating level.

The frequency response of a channel is a comparison between a signal at the input to, and output from, a channel as a function of frequency. The magnitude frequency response is the difference between the modulus of the input and output signals. A channel with a flat magnitude response is one where the difference in modulus between input and output is the same for all frequencies of input signal. Just as the dynamic range is bounded by the linearity of the channel within acceptable limits so the bandwidth of the channel is bounded by the limits within which the frequency response is acceptably flat. The phase frequency response is the phase difference between the input and output signals. A channel with a linear phase response is one in which sinusoids of any frequency each take the same amount of time to propagate through it, thus the phase shift introduced into the sinusoid as it travels through the channel is linearly related to the frequency of the channel. The rate of phase shift at a particular frequency is known as the group delay. If the group delay is the same for all frequencies then the channel has linear phase. If the group delay varies with frequency then the channel has non-linear phase and the linear combination of components at different frequencies will be different between the input and output since they are no longer temporally aligned. Thus a channel can have a flat magnitude frequency response but may

[3] Third harmonic distortion is usually measured by recording a single sinusoid at 1 kHz on to the tape and then comparing the signal level at 1 kHz and 3 kHz (the third harmonic) reproduced from the tape with the latter expressed as a percentage of the former. This is a useful measure of 'saturation' or 'clipping' which occurs when changes in the signal amplitude can no longer be represented by proportional changes in the medium.

noticeably alter a signal which passes through it if it is has a non flat phase response [Blauert and Laws, 1978].

Therefore a channel, like any system, can introduce noise and non-linear distortion into a signal as well as frequency dependent changes in level and phase. It may also introduce an overall delay and change in level between its input and output. An important design concept for the majority of audio recording, transmission and reproduction equipment is that of ‘fidelity’ to the original signal. In fact consumer audio equipment is often referred to as ‘hi-fi’, a shortening of ‘high fidelity’. This fidelity to the original signal is determined by the dynamic range, linearity and frequency response of the channels through which the signal passes.

2.2.6 Digital representations of signals

In the previous section the quality of a channel was related to its dynamic range and its frequency response. Where either of these extends beyond the limits of human auditory perception there is redundancy within the channel and where they are within these limits the signal may be audibly degraded. Clearly if the dynamic range is worse than that of the auditory system but the bandwidth is greater, or *vice versa*, then it is advantageous to perform a transformation on the signal so that useful signal information can be stored in the ‘out of human bandwidth’ part of the frequency range in the channel which, upon inverse transform, will yield a signal with a greater dynamic range. By sampling and quantising a continuous signal it is transformed into discrete data. This data can then be rearranged so that it best maximises the capabilities of the channel. This section describes the principles of sampling, quantisation and channel coding.

When a signal which is continuous in amplitude and in time is sampled it is measured at regularly spaced instants in time giving a finite series of continuous values. Measuring the signal in this way is the equivalent of modulating it with a train of pulses. Since a pulse is constructed from the linear combination of harmonically related sinusoids the signal is effectively multiplied by each of these sinusoids. If the original signal has a bandwidth from 0 to 20 kHz then the frequency spectrum of the modulated signal consists of this spectrum superimposed with shifted and reflected versions centred at the frequencies of the sinusoids which make up the pulse. These sinusoids (and, therefore, the shifted spectra) are $Fs = \frac{1}{Ts}$ apart, where *Ts* is the time interval between successive pulses and *Fs* is the sampling rate. To prevent the original spectrum (the base band) overlapping with its shifted versions its upper limit must not extend more than half way from 0 Hz to *Fs*. If there is an overlap then signal components which are greater than $\frac{Fs}{2}$ will be reflected and appear in the base band spectrum at a different frequency. For example if a signal which contains a sinusoidal component at 25 kHz is sampled at 40 kHz this component will appear as an ‘alias’, a sinusoid at 15 kHz, in the base band. In order to prevent aliasing from occurring the analogue signal to be sampled must not contain any components of a frequency greater than $\frac{Fs}{2}$. This upper spectral limit is known as the Nyquist or Shannon frequency and requires the analogue signal to be low-pass filtered (known as anti-alias filtering). This is shown in figure 2.2.

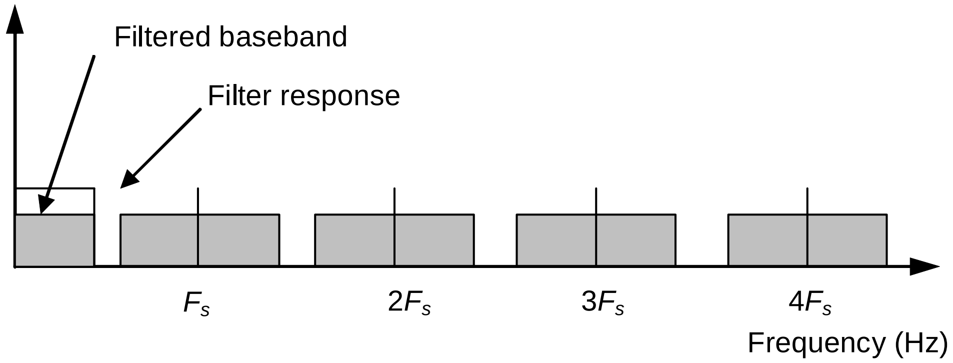


Figure 2.2: Anti-aliasing in sampled systems to prevent pulse modulation sidebands encroaching into the audio baseband. After [Watkinson, 1994].

The compact disc (CD), which was the first widely available technology for distributing digital audio, has a sampling rate of 44.1 kHz giving a Nyquist frequency of 22.05 kHz which is approximately 10 % higher than the 20 kHz taken as the upper limit of hearing in this thesis. The purpose of this margin is to allow sufficient stop band rejection for the anti-alias filter (which needs to be higher than 90 dB to completely eliminate aliasing) whilst maintaining a flat frequency response over the 20 Hz to 20 kHz range. Many professional systems sample at 48 kHz and the initial reason for this was so that the record/playback speed of audio could be varied by +/- 10% (a feature offered by most professional analogue tape machines) without any audible impact on the frequency response of the recorder and without the need for variable sample rate conversion which was an extremely computationally expensive process at the advent of commercial digital audio. Currently much audio-for-video is recorded at 48 kHz partly because this gives an integer number of samples per SMPTE/EBU timecode sub-frame at the EBU rate of 25 frames per second. Recently higher sample rates have become available in higher density formats such as Super Audio CD (SACD) and various flavours of digital versatile disc (DVD) such as –V (video) and –A (audio). The case for higher sampling rates, aside from the financial aspirations of equipment manufacturers, has not been clearly made but, as stated earlier in this chapter, there may be mechanisms by which audio components at a frequency higher than 20 kHz might be detected by humans, if not by the auditory system specifically[4]. Also the pre-response of the brick-wall digital anti-aliasing filters employed in over-sampling converters may be audible to listeners who are less susceptible to backwards masking [Robert Stuart, 2004]. Often a lower sampling rate is used to increase the recording density of a channel, such as the 32 kHz 'long-play- mode available on some R-DAT recorders.

Having sampled a continuous signal at regular intervals it still remains to describe its amplitudes with numbers of finite resolution. This process is known as quantising since the use of finite precision requires that a continuous value be approximated to the closest within a series of discrete values. The number of discrete values available determines the resolution of the quantiser. In a binary system the number of different values is 2^n where n is the number of binary digits (bits) available. For a sixteen bit linear quantisation system, as used for CD, the number of available values is 65 536. Since these values represent the amplitude of the signal, the range of values that can be represented is

$$20\log_{10}\left(65536\right) = 96.3\text{dB} \qquad (2.8)$$

which is the theoretical dynamic range of the quantiser[5]. Since a continuously varying input to the quantiser is output as a representation that moves in steps between discrete values the quantisation process is not linear (i.e. not even a 'linear' quantiser can be perfectly linear without an infinite number of quantisation levels). Where an input signal is high enough in level to exercise a large number of bits in the quantiser then the step distance is insignificant and the quantiser is close to linear but where the input signal is relatively low in level and only uses 1 or 2 bits then this step size is significant. In such situations the quantiser is highly non-linear and its output is a highly distorted version of the input. The distortion is in the form of additional components whose frequencies are integer multiples of those of the

[4] Another possible reason for a much higher Nyquist frequency than 20 kHz is inter-modulation of components above 20 kHz causing distortion components below this limit [Robert Stuart, 2004].

[5] This is actually a simplification of the calculation of the dynamic range. The correct figure is given by equation $(2.8) + 20\log_{10}\left(\sqrt{6}/2\right) = 98.1\text{ dB}$ in the 16 bit case. The reader is directed to [Watkinson, 1994] for further details of this calculation.

components in the input signal. In other words additional components have been added to the signal which are correlated with those of the original signal. Such distortion is highly objectionable since it is fused with the original signal altering its character. This effect can be ameliorated by adding wide band noise to the input signal prior to quantisation. This has the effect of 'blurring' the steps in the quantiser and de-correlating the components added to the signal by the quantiser from the input signal. The cost of this process is that it reduces the dynamic range of the quantiser. For example the use of noise with a rectangular probability distribution at sufficient level to render a system linear at all signal levels reduces the signal to noise ratio by 3 dB [Watkinson, 1994].

Clearly for a given number of bits per unit time (or space) the bandwidth and dynamic range of the signal represented by the data can be determined independent of the bandwidth and dynamic range of the analogue channel[6]. If an analogue channel can accommodate a 16 bit 48 kHz digital signal then it can accommodate a 24 bit 32 kHz signal which has a greater theoretical dynamic range but a lower bandwidth. This is of obvious interest to audio designers since it enables direct negotiation between these two quality criteria. So what effect do the physical characteristics of the channel have on the digital data it contains? The integrity of this data depends on the ability of the digital to analogue converter, or decoder, to differentiate between an analogue signal that represents a 0 and that which represents a 1. Both the bandwidth and dynamic range of the channel can affect this ability. Since frequency is the reciprocal of time a relatively poor high frequency response in the channel will increase the time taken for a transition to a high or low analogue level or *vice versa*. Many channel codes denote one binary value with two transitions within a given time period (e.g. low to high and back to low again) and the other binary value with just a single transition. If the transition from one state to the other occurs too slowly then the signal in the channel will not make it as far as 'high' for fast transitions. The extent to which the signal can move from high to low in the time available for such a transition is known as the 'eye height'. Too low an eye height will introduce bit errors. Noise in the analogue channel, whose signal represents the state of each bit, can also introduce bit errors as the additional components may take the signal below or above the threshold for the correct bit to be identified.

[6] In fact this is not quite true. Care must be taken to ensure that the binary code recorded in the channel is matched to the capabilities of the channel and adjusting the sample rate or word length may have an effect on this. A channel coding scheme is usually devised to best match the raw audio data to the channel. An example is eight to fourteen modulation (EFM) code for CD where 8 bit chunks of data are converted to fourteen bit representations, via table look-up, primarily to avoid long strings of zeros which are of very low frequency and offer no mechanism for detecting bit transitions.

An analogue channel will introduce bit errors and the average rate of these errors is dependent upon the dynamic range of the channel. An analogue channel with a relatively high dynamic range will introduce relatively few errors but the possibility of an error occurring is not eliminated. For this reason additional ‘error correction’ data is generated from the audio data. The purpose of this additional data is for the reconstruction of audio data which may be lost in the channel. A simple example is the addition of parity bits to indicate whether the sum of a sequence of audio bits is odd or even. By arranging data into a square matrix and adding a parity bit for each row and column if any single bit out of the matrix is in error then this error can be detected and corrected. Such a simple system is not particularly robust (it would fail if more than one bit was in error) and CD uses a much more sophisticated system known as cross-interleaved-reed-solomon-coding (CIRC) which generates about 30 % redundancy, i.e. for every 16 bit word there are an additional 4.8 error correction bits. Such a system will reduce the capacity of the channel but is essential for the ‘perfect reproduction’ attribute that is so often associated with digital audio by the layman to be realised. It should be remembered that no error correction system can eliminate errors but they can be made extremely rare. It is interesting to note that computer disk drives which are commonly used to store extremely high resolution digital audio (e.g. 192 kHz and 24 bit digital words) have an analogue channel with a signal to noise ratio of only 20-30 dB. It is the extremely high bandwidth of this analogue channel which allows such vast amounts of high resolution digital audio to be delivered free of errors in real-time.

There is considerable design flexibility in digital systems. The bandwidth and dynamic range of the digital signal can be negotiated independently of those attributes of the analogue channel carrying it or storing it and there is the capability to correct errors which enables perfect transfers of audio data without loss of quality. This has made digital storage and transmission ubiquitous for audio media. As will be discussed in the next chapter, the availability of general purpose computer systems for audio processing and generating operations, has also been a tremendous benefit of digital audio. It is digital audio data, in the general format described in this section, that the signal analysis and modelling methods described in this thesis operate on.

2.3 Organisations of sound

> The core of music as culture is organised and meaningful sound. Its character can best be grasped by contrast with other media and their forms of signification. Musical sound is alogogenic [i.e. not conducive to be expressed in words], unrelated to language, nonartifact, having no physical existence, and non representational. It is self-referential, aural abstraction. This bare core must be the start of any sociocultural understanding of music. [Born, 1995]

So far the nature of sound, human perception of sound and the transmission, storage and reproduction of sound signals have been considered. A spectral model of sound must take account of its nature and human perception of it. Audio data are required to produce a spectral model which then requires synthesis and reproduction to be heard. A brief overview of these areas has been given as preparation for the more specific discussions of spectral modelling that follow this chapter. The rest of this chapter deals with the deliberate intervention of human beings in the production and organisation of sound and the signals and data which represent sound in order to create music.

2.3.1 Definitions of music

The western postmodern understanding of 'what music is' is perhaps best reflected in this statement about the distinction between music and non-music:

> The border between music and noise is always culturally defined - which implies that, even within a single society, this border does not always pass through the same place; in short, there is rarely a consensus ... By all accounts there is no *single* and *intercultural* universal concept defining what music might be. [Nattiez trans. Abbate, 1990].

The point here is that even the broad dictionary definition of music given at the beginning of this chapter may not find agreement amongst all people from all cultures. An example of this is the lack of distinction between music and dance in some cultures. Assumptions about the nature of music and musical sounds which may be fixed in one society may be invalid in another. Whilst what is known as the western European art music tradition may have been concerned with notions of music produced from a small, fixed palette of largely harmonic instruments (i.e. instruments whose modes of vibration are approximately integer multiples of a fundamental mode) elsewhere there is a preference for *in*harmonicity such as in the *campesino* culture of Northern Potosi in Bolivia [Cross, 2003] or in the gamelan music of Java and Bali. Whilst it is important to acknowledge these different definitions and conceptions of music there must also be a stated definition of certain concepts as these apply to work in this thesis, indeed it is a crucial part of the engineering design process. The viewpoint presented here is therefore of music as a solely auditory phenomenon which may be the result of an acoustic event, an electroacoustic one or a combination of both. The original source of the sounds, or components of sound, heard may be physical or synthetic (i.e. energy generated in a non-acoustic domain and then converted into acoustic energy). Components of music may be grouped into particular types of features such as those related to time or frequency such as rhythm (time), pitch (frequency) and timbre (time and frequency). Many sound modelling systems assume harmonicity. This is avoided in this thesis in the light of what has been discussed in this section.

2.3.2 Parameters of music

The French word timbre can be literally translated as ‘stamp’. The American National Standards Institute (ANSI) definition of timbre (in English) to describe sound is “that attribute of auditory sensation in terms of which a listener can judge that two sounds similarly presented and having the same loudness and pitch are dissimilar” [ANSI, 1960]. Whilst timbre is dependent upon the frequency and level ratios of any stable sinusoids present in a sound it is also dependent upon the temporal evolution of frequency and level for each sinusoid and of the level, bandwidth and centre frequency of components that are not localised in frequency [Risset and Wessel, 1998]. Organisations of sound in the western European tradition before the advent of electronic music can be seen as arrangements of notes, a note being an individual sounding of an object. The sounding object, if designed for the purpose of producing music, is referred to as a musical instrument. Confining our discussion to a monophonic instrument (i.e. one that is capable of playing only one note at a time) the attributes of a note are its pitch, loudness and timbre as well as how it links to other notes that come before and after it. This latter attribute is known as articulation and a simple distinction here might be that between ‘detached’ (with a temporal gap between notes) or ‘non detached’ (the onset of the next note occurs at the same instant as the termination of the previous note). Whilst all other attributes may be determined entirely by the player of the instrument the timbre is largely associated with the instrument itself i.e. it is the auditory ‘signature’ of the source of the sound. Some instruments allow a great deal of control over timbre, for example the way that a piano note is struck on a keyboard can vary the timbre (as well as the intensity) significantly. Other instruments do not offer such control, an example being the pipe organ which shares the same interface layout with the piano but whose keys are little more than switches turning the air supply to a resonant pipe on and off.

Representations of music in this tradition are usually in the form of a ‘score’. A score is a set of written instructions for performing music. A score must be rendered by instrument players in order to be sensed by the human auditory system. Often a score will contain very precise instructions such as ‘crotchet = 120’ (play at a speed of 120 quarter notes per minute) but they may be more vague such as ‘adagio’ which is defined as “At ease [from Italian] (not so slow as largo but slower than andante” [Kennedy, 1980]. In addition to the language of the score (i.e. the system of notating pitch, timing and articulation) there are the notions of performance practice and interpretation. The former is a set of conventions to be observed when playing a certain kind of music, or music written in a particular era and the latter is an additional input to the control of timing, articulation etc. on the part of the performer to better match the rendered sound to the needs of the audience, their own artistic aspirations or the intentions of the composer of the score. In addition to the score there are also traditions of improvisation and the oral dissemination of musical instructions where a performer learns how to play a piece by hearing someone else play it.

2.3.3 Music and audio technology

The advent of recording systems has allowed the performance, an instantaneous and unique event to be captured and replayed. In addition a number of separate performances can be edited together to give the *impression* of a single, continuous and spontaneous event. This modification of captured performances can be taken further to the manipulation of signals as part of the compositional process. A new approach to composition using recording machines, *musique concrète,* began with the work of the French composer Pierre Schaeffer who coined the term in 1948 which was originally designed to differentiate it from synthetic electronic music produced using simple tone generators and other such means [Kennedy, 1980]. *Musique concrète* constructed music from the recordings of ‘concrete’ objects (i.e. recordings of real, physical objects made using a microphone). Schaeffer’s celebrated first piece in this idiom *Etude aux Chemins de Fer* (‘Railway Studies’) was created solely from recordings of steam locomotives [Born, 1995]. With such a piece the recording medium becomes the score and this is the only means of rendering the performance – there is no set of instructions that a human being can follow with an acoustic instrument since the music must emanate from the concrete source of the original sound material. Current technology enables recordings to be made and stored digitally on random access media such as computer disk drives. Fast random access devices enable the assembly of musical compositions at performance time rather than having to be rendered in non-real time by splicing pieces of tape together or by assemble editing[7]. This offers the option of extemporaneous composition using concrete sounds or a set of playback instructions can be devised at the composition stage to be executed at performance time. Such a set of instructions, particularly if intended for interpretation by a computer system rather than by a human operator of the computer playback system is called an edit decision list (EDL) which, when used as a tool of musical creation, can be considered to be a score.

Abstract electronic music uses signals which are synthetically generated (i.e. are not the result of acoustic to electric transduction) and so do not have a concrete source. The first electronic synthesizer was probably the ‘Telharmonium’ of Thaddeus Cahill, patented in 1897 and first demonstrated to the public as a complete system in 1906. It used a system of dynamos with shafts with differing gear ratios to produce alternating currents of different frequencies. These signals were rendered acoustically via horns connected to modified piano sound boards and were later distributed to listener’s homes via the telephone network with a horn connected to the telephone receiver to render the acoustic signal audible [Williston, 2000]. With the invention of the triode vacuum tube of De Forest in 1907 the amplification of electrical signals was possible meaning that audio signals (synthetically generated, transmitted or recorded) could be amplified prior to transduction enabling them to be reproduced via loudspeakers at sufficient level to be audible.

Other innovations in the field of electronics such as filters and non-linear devices, in addition to flexibly manipulated media such as magnetic tape, allowed signals to be processed and the use of such devices for the modification of audio became a means of expression in the composition of electronic music. The strict definition of *musique concrète* is recorded sound that has not been modified electronically but the boundary between concrete and abstract electronic music has become less clearly marked over time and the term is considered archaic by some [Kennedy, 1980]. There is little agreement on precise definitions of the terms electroacoustic and electronic music. Electroacoustic has a precise engineering definition which is an adjective describing any process which involves the transduction from electrical energy to acoustic energy or *vice versa*. Thus electroacoustic music is any form of music that requires loudspeakers for its performance and this is the definition which is adhered to in this

[7] Assemble editing is the process of compiling a composite signal from different recording signals (or different portions of the same signal) by recording the wanted signal excerpts in the required order contiguously on one system from the playback output of another. This is as opposed to ‘cut and splice’ editing where the media storing the original signals is physically manipulated to produce the desired contiguous sequence.

thesis. Electronic music was a term applied to electroacoustic music to that was not *musique concrète* but the terms electronic and electroacoustic have become interchangeable for many people, hence the use of the adjective ‘abstract’ thus far to refer to electronic music using wholly synthetic sounds[8].

2.3.4 Music and computer technology

Whilst the application of electronic devices to analogue signals offers the potential to alter sounds in a variety of meaningful ways, systems which allow manipulation of digital data, be it a general purpose computer or dedicated digital audio processing device, enable audio to be modified by any process that can be described by discrete mathematics. For example, two audio signals represented by such data can be mixed together by addition or a signal can have its level altered by multiplication. These two processes can be accomplished in the analogue domain using a bus bar and an amplifier respectively but the processing of audio as data “allows tremendous opportunities which were denied to analogue signals” [Watkinson, 1994]. In addition, rather than requiring a specific type of electronic device for a particular processing task, a general purpose *hardware* digital device such as a central processing unit (CPU) or digital signal processor (DSP) can be adapted to a different audio signal processing task if given a new set of instructions (known as *software*). The practical outcome of this thesis is a software algorithm for spectral modification of audio which can run on general purpose computer hardware.

With such tools available for the manipulation of audio the composer is able to exert much greater control over the timbre of sounds than is possible when providing a set of instructions for players of acoustic instruments. The time evolution of individual spectral components, the relationship between those components in the frequency domain and their spatial dimensions and position are examples of how this control can be exerted on synthetic or recorded sounds through the use of a computer. The composer Dennis Smalley observes that:

> Prior to the electroacoustic period music always involved identifiable sources. The listener could spontaneously link both to a sounding body and a human, physical cause. Gesture not only activated the source but could, through breathing and bow control, and techniques of touch, maintain and continue the sounding of the vibratory system. Traditionally, therefore, there is an inherent, culturally imbedded, stable *source bonding* in music. Source bonding is the term I use to encapsulate the natural tendency to relate sounds to supposed sources and causes, and to relate sounds to each other because they appear to have shared or associated origins. [Smalley, 1994]

[8] See [Landy et al] for a more detailed discussion of the etymology of these terms.

There are a myriad of processing/synthesis paradigms that can be assumed when composing music in this way such as additive, subtractive, physical modelling and spectral modelling synthesis. These are discussed further in the following chapter.

2.3.5 Implementation of computer audio processing

The EDL as a form of score has already been discussed and this leads to the discussion of languages for interacting with computer hardware and defining computer software. Computer languages can be categorised by their level, a low-level language being one which can be acted on by computer hardware with relatively little or no translation and a high-level language being closer in terms of language (or, possibly, visual representation) to a form that a human being could easily recognise and interact with. A high level language will require translation into a set of machine instructions (also known as code) which can be loaded directly onto the target computer hardware and executed (or run). Machine code is the lowest level computer language, the instructions it contains cannot be further broken down into a larger set of simpler instructions. This code consists of binary numbers which represent types of instruction that the processor can execute, numbers or memory locations where data can be stored and read from. A specific task, such as calculating the cosine of an angle, can be described as a series of instructions for the processor to carry out at run-time. Usually such a series of instructions will be designed specifically for the processor to be as short as possible (in order to reduce execution time) and/or to use as little memory as possible. Entering instructions in this way gives exact control over how a processor performs a particular task but is counter-intuitive (instructions are represented by binary codes which have no representative meaning) and time-consuming (tasks are programmed as a series of single instructions rather than as groups of instructions).

An assembly language is one level removed from machine code, it still specifies single instructions for a particular processor but it uses mnemonics to represent each instruction giving them representative meaning. An ‘assembler’ program is required to convert these mnemonics to machine code that can be executed. Above this are general-purpose programming languages such as C which offer a set of commands for describing tasks which may, and often do, require multiple machine instructions in order to be executed. Another feature of such a language are control-flow constructions for decision making (using the `if` and `else` statements), recursion with testing (`while`, `for` and `do`) and so on [Kernighan and Ritchie, 1988]. A compiler converts this language to machine code. There is usually an intermediate stage where ‘objects’ are produced which contain machine code along with references to other existing machine code that must be included in the final executable program in order for it to run. The descriptors high-level and low-level are relative terms, C is a higher level language than assembler but is lower in level than languages that offer, for example single commands for file reading and writing. However the use of libraries (such as the ‘math’ and ‘standard IO’ libraries for C) which are readily-specified instructions for performing certain tasks is straightforward since a single set of C instructions (a program) can be specified in more than one file.

Generally the higher in level the language the more abstracted it is from the hardware that it runs on. This means that the programmer has less and less control over how their instructions are carried out and so is less able to ensure that the instructions are carried out in the most efficient way for the hardware that they are performed on. This can be a serious disadvantage when writing performance critical code such as that which must be repeatedly completed within a specific time, such as a real-time algorithm for processing audio data. In the same way a composer writing a fast passage for a particular acoustic instrument must ensure that it is written in such a way that it can be played by a human being at the prescribed tempo. The definition of an algorithm used in this thesis is that of a fixed set of instructions which will perform a task and terminate with a pre-determined *type* of outcome. Algorithms often consist of a large number of instructions which would be prohibitively costly in terms of time to specify in machine code or assembler. However an algorithm will often itself use algorithms that have already been defined for a particular piece of hardware. For example the Fast Fourier Transform (FFT) can be described using a few lines of C code (using the maths library) [Press et al, 1992] but a much faster FFT, written specifically or optimised for the hardware on which it is running, can be used by accessing a specialised signal processing library. An example of such a library is the Fastest Fourier Transform in the West (FFTW), a library of FFT functions which can be called from a C program and which can be optimised by calling a ‘trial’ function which evaluates which is the fastest FFT algorithm that can be run on a given processor [Frigo, 1999]. The MATLAB scientific programming environment, which uses a specialised high level language, uses the FFTW and so provides a fast implementation of this intensive process within a high level language.

Other such libraries are specifically designed for a particular processor, or group of processors, such as Intel’s Integrated Performance Primitives for its Pentium, Itanium and

Xeon processors[9]. The FFT routines in these libraries are reported to run faster than the FFTW on their target processors [Gonzalez and Lopez, 2001]. Libraries can either be statically or dynamically linked to C code. Static linking refers to the inclusion of the required code from the library at compile time, dynamic linking to its inclusion at run time. Dynamic linking allows new versions of libraries to be used with an executable (or other dynamically linked library, or DLL) written in C without the need for recompilation [Microsoft, 2006].

With relatively few keywords (words which belong to the core language) and the ability to combine these to produce complicated sets of instructions C is an intuitive language to use. An example of C's suitability to audio processing is the fact that the symbols for array referencing are very similar to those for representing discrete time signals in signal processing, where a value in square brackets indicates the sample number (e.g. $x[n]$ represents the *n*th sample of the signal *x*, or the *n*th variable in the array *x*). The flexibility in using existing, often highly optimised, routines for specific operations within new code offers compiled code that is close to optimum efficiency despite not being written in assembler. These factors combine to make C an attractive language for development. However, it is important to consider the context of a processing algorithm when evaluating the efficacy of a programming language with which to implement it.

Whilst library functions obviate the need to 'reinvent the wheel' when dealing with common signal processing routines such as the FFT, there is still a requirement for an environment in which a new audio processing algorithm can be used. Any audio processing algorithm will need audio to process and it would be costly to include audio file opening and closing, audio playback, mixing and so on. The digital audio workstation (DAW), a computer based system for recording, editing, mixing and processing audio, offers such functionality and processing algorithms can be hosted by many of these systems. The term 'plug-in' refers to a computer program that can interact with another (the host) to provide a specific function. Very often such plug-ins are dynamically linked libraries meaning that they can be installed independently of the host and so do not require the host software to be reinstalled every time a new plug-in is required. Thus a hierarchy of functionality exists: there is a host program which provides a user interface and functions such as audio playback, below this there is the plug-in which may be used to perform specific audio processing tasks and below this there

[9] Pentium, Itanium and Xeon are trademarks of the Intel Corporation.

are library routines which the plug-in can use to perform specific tasks such as calculating the FFT of some data or to display a graphical user interface. Plug-ins must conform to a specified format in order to function correctly within a host program since both the host and the plug-in must know what data, and in what format, each requires from the other and when it is required, and/or how to request it via function calls. There are many audio plug-in ‘standards’. Some are platform/OS specific such as the Apple’s Audio Units for their Macintosh computers or DirectX plug-ins for Microsoft Windows. A popular cross-platform specification is the Steinberg Virtual Studio Technology (VST) plug-in which is supported by many DAW applications such as Steinberg’s own *Cubase*, *Nuendo* and *Wavelab* and those by third parties such as Cakewalk’s *Sonar* and Plogue’s *Bidule*. The software development kit for VST plug-ins is provided free of charge and essentially supplies a C++ ‘wrapper’ into which C code describing the real-time process can be inserted [Steinberg, 1999].

C++ was developed as an extension of C at Bell labs in the 1980s. It was intended to allow large software projects to be more easily managed, implemented and maintained. The language retains all of the keywords from C whilst adding new ones that offer new facilities such as classes and function and operator overloading. A class can consist of data and functions which can be used to perform a particular programming task. Classes allow aspects of code to be compartmentalised making it easier to design and write large scale code. Function and operator overloading allow functions to be defined according to the variables they take [Stroustrup, 1997]. For example a function with the same name may be defined to behave differently when the input arguments are integers to when they are floating point numbers. In C a different function definition would require a different function name. In the context of VST plug-ins, C++ provides a means for inheriting the basic functionality which all plug-ins require in a framework in which the additional functionality required specific to the particular plug-in can be added quickly and easily.

Whilst C++ and plug-in technologies offer a much faster design and implementation cycle than that of purpose built hardware they are not purposefully designed as research tools for the development of audio processing algorithms. The MATLAB environment mentioned previously offers a number of advantages over the C++/plug-in approach. Algorithms can be implemented much more quickly since there is a wide range of functions within the environment for performing common mathematical tasks many of which can operate on vectors and matrices [Mathworks, 2006]. A wide variety of tools for graphically representing data and transferring it between formats (such as audio files) exist. As is discussed in chapter 6, functions written in MATLAB's 'm' file language do not execute as fast as their C or C++ equivalents but interfaces between MATLAB and DLLs written in C exist for use where optimisation for speed is required. The spectral processing algorithm presented in chapter 6 is written in a combination of the MATLAB and C languages. However it is envisaged that any real-time processing tool which arises from the research presented in this thesis would be implemented as a VST plug-in.

Programming languages are not the only way that composers can interact with a computer. The type of DAW host application previously discussed, which will usually provide a graphical user interface possibly augmented by a specially designed hardware control surface, are much more common means of interaction than the languages used for creating code. Very high level languages, such as *CSound*, also exist specifically for computer music tasks. *CSound* 'compiles' audio files from an 'orchestra' file, which specifies the instruments and/or processing functions (known as opcodes) to be used and how they are connected, and a 'score' file which determines how and when the instruments are to be played (and/or how and when processes are to be applied). A number of opcodes are provided within *CSound* ranging from oscillators to phase vocoders and new opcodes can be written in C [Boulanger, 2000].

2.3.6 Real-time versus rendering

Users can interact with computers to produce music in real-time as they might with an acoustic instrument – a gesture is performed or an instruction is issued and the result is immediately audible. Computer music also offers a second means of working, as does any means of music creation which involves manipulation of the medium on which instructions or audio signals reside (such as a piece of magnetic tape or a piano roll), which is non-real time. Instructions and signals are recorded and then audible output is rendered. The rendering process may or may not happen in real-time, for example a player piano will render its output in real-time but a computer system will usually render output to a file which can only be played out once rendering is complete. Non-real time interaction may be the result of choice (i.e. the composer does not want to extemporise) or of processing limitations. For a computer to carry out a task in real-time the number of instructions per unit time required to perform it must not exceed the number of instructions per unit time it is capable of executing. The execution speed of processors has approximately doubled every year as predicted by Moore's

Law [Moore, 1965] and so the number of audio processing algorithms that can be executed in real-time has also increased. Spectral morphing of a few seconds of audio by time varying interpolation between two sets of FFT data which a few years ago would have had to be rendered over a number of hours [Wishart, 1988] can now be easily performed in real-time [Fitz and Haken, 2003].

Whilst real-time FFT and time domain filter based processing can now be executed in real-time many processes which use these analysis techniques are still performed 'offline' (i.e. not in real-time) since they require the analysis of whole sounds, or sequences of sounds, rather than just successive portions of sound as they are acquired in real-time. A real-time system allows live performance with that system thus removing one of the three acousmatic dislocations[10] in electroacoustic music as identified in [Emmerson, 1994]. An example of this is the spectral modelling of sound which is discussed in detail in the next chapter. The primary objective of this thesis is to investigate how spectral modelling might be achieved in real-time.

2.4 Summary

In this chapter some basic definitions of key terms and concepts that will be used throughout this thesis have been introduced and a brief overview of the use of computer systems for the manipulation of audio as part of a creative musical process has been given. The coverage is by no means exhaustive and is necessarily selective but a context for the work described in this thesis within music, technology and the cultures that surround them has been described. The first part of this chapter described the nature of sound, its perception and the numerical format in which it is stored and operated on in computer systems. The second part surveyed the means and motivation for creative transformation as musical expression. Spectral modelling is a mature field of audio signal processing which offers many creative transformation tools, however it is commonly considered to be an 'offline' process which restricts its applications and appeal. This thesis investigates whether musicians who wish to use spectral modelling tools need be constrained in such a way.

[10] Acousmatic sound is defined as sound which whose physical cause is not observable by the listener [Landy et al]. According to Emmerson "the truly fundamental acousmatic dislocations occurred in the half century to 1910:
Dislocation 1: Time (recording)
Dislocation 2: Space (telecommunications (telephone, radio), recording)
Dislocation 3: Mechanical causality (electronic synthesis, telecommunications, recording)
There were 'pre-historic' versions of these dislocations – the western score itself may be seen as one – but I refer here to the physical wave trace in the first instance" [Emmerson, 1994]

A recent organisational initiative in this country has been the Digital Music Research UK Roadmap, launched in December 2005 by the Digital Music Research Network (DMRN) which is funded by the Engineering and Physical Sciences Research Council (EPSRC) [DMRN]. This roadmap identifies strands in current research in this ‘transdisciplinary’ field and attempts to identify and stimulate its long-term aims. Although the roadmap was published as work for this thesis was nearing completion it does fall into two of the broad research ‘goals’ identified. When discussing the area of musical innovation it is noted that “developments are also expected within the area of live computer algorithms” and, in the area of producing fertile environments for creativity “objectives must include … [the development] of human-computer interfaces better suited to supporting creative activities in composition and performances” [Myatt, 2005]. It is hoped that the work presented in the following chapters of this thesis contributes in some way to these goals.

3 MODELS OF SOUND

> The main goal of musical signal processing is to provide musicians with representations that let them modify natural and synthetic sounds in perceptually relevant ways. This desideratum explains the existence of techniques for synthesis, often supported by associated analysis methods. [Cavaliere and Piccialli, 1997]

3.1 Introduction

This chapter deals in detail with the context of this thesis. Modelling, synthesis and processing of sound are explained and in particular current techniques and theory relating to spectral analysis, processing and modelling of audio are presented. Although not a research focus of this thesis some discussion of physical modelling and its relative merits compared with spectral modelling are presented in order to place the latter in the overall context of sound modelling.

For simplicity many of the equations here refer to phase expressed in radians and angular frequency which is the frequency (in Hz) multiplied by 2π. However, where numerical examples are given they are expressed in Hz since this measure has a more tangible physical meaning. Later chapters present results relating to frequency in Hz. Phase is always referred to in radians. The terms amplitude and magnitude are synonymous with each other in the literature although there is a tendency to use amplitude to describe the time domain magnitude of oscillation and magnitude for the frequency domain equivalent. This latter distinction is the one adhered to here. Where time domain plots are shown the vertical axis is unnamed as is standard practice in the literature. This axis ultimately refers to amplitude of pressure variations in the corresponding acoustic signal. The letter j is used to refer to both $\sqrt{-1}$ and scale, or level, of multiresolution/wavelet analysis. It is clear which of these it represents from the context in which it is used.

3.2 What is a sound model?

A mathematical model is a description of the behaviour of a system or event in terms of numbers and the operations that can be applied to them. Mathematical models often exist to allow the simulation of a physical system since, for example, it is much cheaper to have a mathematical model of a prototype aircraft crash than the real thing. Models are also useful for imagining systems or events that could not physically happen, such as how life forms on this planet might have adapted differently under different gravitational conditions.

A sound model is one which describes a system for creating, conducting, transforming or sensing sound. For example models exist to explain how sound emanating from a point source propagates through air (the inverse square law described in the previous chapter) and how changes in the intensity of a sound produce changes in the perceived loudness of that sound. A distinction is made in this thesis between a process and a model. A process is an operation on a set of data which takes no account of what the data represents and does not attempt to infer anything about the data, such as an underlying structure, from the outcome of the operation. A model attempts to derive a meaningful structure from the data on which it operates (it may even assume, *a priori*, an underlying structure) and may well use this assumed structure to determine how it subsequently operates on the same data. The Fourier transform of data from the time to the frequency domain is an example of a process, the inference of the underlying processes that might have produced the original time domain data is an example of a model. Models are useful since they allow us to better understand and describe real-life systems that currently exist and, by extrapolation, to imagine those which do not.

3.3 Methods of modelling sound

3.3.1 Imagining acoustics and mechanics: physical modelling

Physical modelling (PM) of sound is concerned with the mathematical description of the physical systems which create sound or modify sound, such as the string of a guitar or an enclosed space in which sound reverberates. That is not to say that PM is only concerned with that which can physically exist, in fact one of the primary goals is to develop instruments that could not physically exist (such as instruments which can expand and contract whilst remaining in tune) or that would be prohibitively expensive to create or recreate such as a reverberator like the original cathedral building in Coventry, which was destroyed during the second world war.

Two important forms of PM are what is known as classical modelling using a lumped (homogeneous) model and waveguide modelling that uses a distributed (heterogeneous) model. Lumped models describe a sound generator or modifier in terms of the interconnection of masses, springs, dampers and non-linear devices and this way of modelling is often referred to as the ‘mass/spring’ paradigm. Distributed models tend to model the propagation of vibrational waves through objects rather than how each tiny part of the object responds to a force applied to it by the movement of its neighbour. The relationship between the displacement of a mass from its equilibrium position (*x*), its velocity $\left(\frac{dx}{dt}\right)$, its acceleration $\left(\frac{d^2x}{dt^2}\right)$, its mass (*m*), the stiffness (*K*) of the spring it is connected to and the mechanical resistance (*R*) to the movement of the mass in fluid surrounding it is given by Hooke's Law and Newton's Second Law of Motion as the following second order differential equation.

$$\left(\frac{d^2x}{dt^2}\right)+\frac{R}{m}\left(\frac{dx}{dt}\right)+\frac{K}{m}x=0 \qquad (3.1)$$

Whereas the behaviour of a spring connected to a mass as a resonator can be modelled by a second order filter, the delay caused by ideal propagation through a series of masses is modelled by a delay line (known in this context as a waveguide). Since waves travel at a finite speed through physical objects the time taken to travel through a particular material is dependent upon the distance travelled through it. In this way spatial separation between parts of a PM instrument are modelled by time delays although losses at key points due to absorption and dispersion are still modelled as filters. In fact the two types of model can be, and often are, combined [Smith, 1996]. The mathematical basis for waveguides is given by the solution to the wave equation in one dimension:

$$\frac{d^2y}{dx^2}=\left(\frac{1}{c^2}\right)\frac{d^2y}{dt^2} \qquad (3.2)$$

where, for a vibrating string, *y* is the displacement of the string from its equilibrium position, *x* is the distance along the string, *t* is time and *c* is the speed of wave movement along the string, a constant value given by:

$$c=\frac{K}{\varepsilon} \quad (3.3)$$

Where *K* is the string tension and ε is the mass density of the string. Essentially (3.2) means that the curvature of the string is proportional to the acceleration of the string in the *y* direction and inversely proportional to the square of the speed of wave propagation in the *x* direction. The solution to (3) by D'Alembert is:

$$y(x,t) = y^{+}\left(t + \frac{x}{c}\right) + y^{-}\left(t - \frac{x}{c}\right) \qquad (3.4)$$

Where y^{+} and y^{-} represent two waves travelling in opposite directions along the string where *y* is an arbitrary twice-differentiable function. Therefore the displacement of the string in the *y* direction at point *x* and time *t* is given by the sum of these two waves. This displacement can be sampled at any point along the string, at any time, by sampling at a particular point in a delay line at a particular sample number. This analysis of wave propagation is not confined to strings. For example, it can be adapted to acoustic tubes and extended to two dimensions to model membranes and to three dimensions to simulate rooms [Murphy, 2000].

Both types of PM require something to be 'done' to the instrument to make it sound. An initial condition is often specified such as a displacement of a mass or the initial shape of a plucked string will be loaded into a waveguide. Different physical acts such as striking or bowing a string can be realised by causing a time varying velocity, displacement or acceleration to be applied to a part, or parts, of a physical model, breath can be simulated by injecting broad band noise into the system and so on. The interaction between the 'exciter' (the cause of vibration) and the 'resonator' (the filter applied to the vibration) in PM is an important aspect of how the instrument functions. In many acoustic instruments the resonator feeds back to the exciter causing changes in the excitation pattern which is injected into the resonator and this behaviour can be easily replicated in PM.

PM offers intuitive models firstly because the parameters of the model are, by the definition of PM, closely related to the parameters of a real physical object (even if the instrument cannot physically exist its structure is still based on that of a real, vibrating system). For example the velocity of a physical movement (such as a hand moving a mouse) can be mapped to the relative bow-to-string speed of a bowing action which is the excitation for a PM instrument. This relative speed can then be mapped to the magnitude of the force exerted on the mass (or masses) which the bow is in direct contact with. This gives some connection between the gesture that generates the sound with the physical causality that might be associated with that sound [Howard and Rimell, 2003]. Since PM is the only form of modelling that directly describes the source of a sound it is the only paradigm that offers such immediate control of physical gesture type inputs to the system.

A recent study at the Helsinki University of Technology (HUT) made a comparison between different synthesis systems. The tasks which PM was best suited to were identified as:

- simulation and analysis of physical instruments
  - copy synthesis
  - study of the instrument physics [sic]
  - creation of physically unrealizable instruments of existing instrument families
- applications requiring control of high fidelity[1]

[Tolonen et al, 1998]

A disadvantage of PM is that it usually requires prior analysis of an existing system in order to produce a model, a model cannot be readily derived just from audio data for example. A principal feature of PM is that the model will often change drastically from one instrument to another so each instrument will require considerable investment in design before it can be realised but perhaps will more readily offer, over other sound models, the nuances of sound production and subjective 'character' that acoustic instruments possess.

3.3.2 Deconstructing what we hear: an overview of spectral modelling

Whereas PM is concerned with sound as it is generated at its source so spectral modelling (SM) is concerned with describing sound as it arrives at the receiver (e.g. ear or microphone). The components of a PM system are imagined versions of physical components of a sound source, the components of an SM system are generators or filters of simple sound components that can be combined to reproduce the input sound. SM can be seen as an extension of two well known synthesis techniques: additive and subtractive. Additive synthesis creates more complicated sounds by the addition of simple oscillations which are usually sinusoidal. Many additive synthesizers offer control over the relative amplitude of different components and possibly their relative tuning as well. Often these two parameters can be controlled over time. Subtractive synthesis offers pre-defined wave shapes (such as pulse or sawtooth) and noise sources which can be combined and filtered to produce the desired output. Here a spectrally rich sound is started with and portions of it are removed by filtering. So-called 'sample and synthesis' systems allow users to load audio data for use as the starting waveform before using time-variant filters and amplifiers to alter the sound produced.

There is an acoustic precedent for additive synthesis which is the pipe organ. A keyboard (sometimes pedal board for feet as well) instrument which produces sound by blowing air

[1] Meaning applications requiring high fidelity of control of the instrument's performance parameters, not to be confused with 'Hi-Fi' reproduction systems discussed in the previous chapter.

through pipes of different lengths (some simple flues, others containing some sort of vibrating object such as a metal reed). Air pressure is provided by a wind chest into which air is pumped from bellows. By a system of stops the operator/performer can select pipes which have a different pitch (either an integer multiple or a rational number representing a common musical interval such as the octave plus a fifth, known as a quint) and/or timbre and these stops can be used in combination to provide combined sounds of varying timbre. With a large number of stops a wide variety of different timbres and dynamics can be produced although the operator has no control over the timbre of each individual stop – this is fixed.

The output of time-varying additive synthesis can be described by (assuming that the function is centred around zero, i.e. there is no offset as there is in equation (2.5) of the previous chapter):

$$x(t) = \sum_{p} A_p(t)\sin(2\pi f_p(t)t + \phi_p) \qquad (3.5)$$

where *p* is the ‘partial’ number (a partial being a single sinusoidal component of the output sound). The lowest partial is also known as the fundamental. Note that for time-varying synthesis the amplitude and frequency of each partial is an independent function of time. It is common in additive synthesizers for the upper partials to have a frequency which is an integer multiple of the frequency of the fundamental (e.g. the Kawai K5000 synthesizer) but at least one exists (Synergy GDS) where the frequency ratios need not be integers or rational numbers representing a common interval [Bellingham and Gorges, 1998], [Vail, 2000]. This latter type of additive synthesizer can be seen as a significant move away from the fixed harmonic additive synthesis of pipe and electric organs which opened up new areas of timbre and tuning system design [Carlos, 1986]. A sinusoidal analysis and resynthesis system which tracks the slowly changing parameters of individual partials from frame to frame is the McAulay and Quatieri (MQ) system. This system links partials across frames and uses cubic phase interpolation at synthesis between frames to produce smoothly changing partial frequencies [McAulay and Quatieri, 1986].

The theory that any periodic function, no matter how complicated, could be decomposed into the superposition of a number of sinusoidal functions of different amplitudes and different, but harmonically related, frequencies was first stated by Fourier in1807 [Robinson, 1982]. This theory has had a tremendous impact on many aspects of science and other disciplines and it is an important result for additive synthesis since it implies that any periodic waveform can be recreated *exactly* by a sinusoidal additive synthesizer. In other words any audio signal, provided it is periodic (i.e. it repeats itself at regular intervals), can be *modelled* as a sum of functions that are perfectly localised in frequency. Since sinusoids cause the narrowest areas of excitation on the basilar membrane, and can therefore be seen as the simplest spectral component of sound, this model can be seen as valid in terms of perception and complete in terms of the range of complex functions it can represent. The previously cited HUT study gives the following assessment of sinusoidal additive synthesis:

> The parameters are *fairly* intuitive in that frequencies and amplitudes are easy [for the user] to comprehend. The behaviour of the parameters is *good* as the method is linear. Perceptibility and physicality of the parameters is *poor*.[2] [Tolonen et al. 1998]

The Fourier sinusoidal additive model is both effective and intuitive for deterministic signal components. A deterministic function is one whose output at a given point in time can be predicted from the parameters of the function. For example, provided we know the start phase and frequency and amplitude trajectories of a sinusoid we can predict what its value will be at any instant in time; there is no uncertainty involved in the process. In contrast a stochastic process is one whose instantaneous value cannot be perfectly predicted. An example of a stochastic process is white noise which can be produced by a random number generator. Since it is a random process the next signal value cannot be predicted from the previous one or from any initial conditions. However a random process can still be described in terms of its spectral content and its probability density function. For a random number generator to produce white noise it must not base its output on knowledge of previous output although it may have a probability density function that is not rectangular (i.e. all numbers have an equal probability of occurring). A single dice has a rectangular probability distribution since each of its six numbers has an equal probability of occurring. The addition of two die has a triangular probability function. As more die are added the distribution tends to a normal distribution (Gaussian function). White noise has a flat power spectral density which describes how a signals power varies with frequency. Other 'colours' of noise include pink noise where the power spectral density as a function of frequency falls at 6 dB per

[2] In [Tolonen et al, 1998] a number of different criteria are evaluated with one of three descriptors: poor, fair or good. The following criteria are used to evaluate the synthesis parameters. Intuitivity is how well the parameters relate to a musical attribute or timbral quality. Perceptibility is how well parameter changes are mapped to noticeable changes in timbre. Physicality describes how well a synthetic instrument's parameters correspond to those which a player of a physical instrument might be familiar with. The behaviour of a parameter is related to how linear the parameters are since this describes how well changes in parameters can be related to changes in output.

octave, giving equal power in frequency bands of equal width since each successive band has double the width of the previous band. Since coloured noise does not have a flat magnitude spectrum it is not perfectly random since the current value will be determined to some extent by recent values.

Therefore stochastic (or ‘noisy’) processes can be described but their instantaneous value can never be perfectly predicted. Since noise exists in a continuous frequency range it theoretically requires a sinusoid at every frequency within that range (i.e. an infinite number of them) to reproduce it. Even in a discrete system the modelling of noise with sinusoids is expensive and counter-intuitive.

3.3.3 Spectral Modelling Synthesis

> The first and principal difference between various sounds experienced by our ear, is that between *noises* and *musical tones*. [Helmholtz, 1954]

The Spectral Modelling Synthesis (SMS[3]) system models the spectrum with both deterministic functions (slowly varying sinusoids) and stochastic processes (time varying filters applied to a white noise source) [Serra, 1989]. SMS uses sinusoidal amplitude and frequency tracks and filter coefficients which are inferred from short time Fourier analysis (discussed later in this chapter). A further extension to spectral modelling is the separation of transient events. Transient, or suddenly changing, components usually occur at note onsets often as the result of non-linear excitation or the presence of very highly-damped (so very short-lived) sinusoidal oscillations which appear as broad band components due to their short duration and fast changing parameters. Again, broad band components are not well modelled by sinusoids since the latter are perfectly localised in frequency. Parametric lossy audio coding attempts to reduce signal bandwidth by describing audio as sinusoids, transients and noise. Bandwidth is reduced since filter coefficients and sinusoidal data can be sent at a much lower sample rate than the original time domain audio data. Transient data is usually transform encoded (i.e. it is not modelled but simply transformed into the frequency domain) and thresholding is used to reduce the amount of transformed data [Levine, 1998]. Another spectral modelling system is the reassigned bandwith-enhanced additive model [Fitz and Haken, 2002]. This is a homogeneous model which attempts to model all signal types with sinusoids of varying bandwidth. The bandwidth of sinusoids is increased where required (i.e.

[3] In this thesis the acronym SMS refers specifically to the spectral modelling system described by Serra [Serra, 1989]. The acronym SM refers to spectral modelling in general, just as PM refers to physical modelling in general. Therefore SMS is a subset of SM.

for modelling of noisy processes) by adding low pass filtered noise to the amplitude estimates. The reassignment technique (discussed in section 3.10) is used to produce accurate estimates of sinusoidal mean instantaneous frequency (frequency reassignment) and for sharpening transients (time reassignment).

These extended systems perform better in terms of generality (since the systems are suited to a wider range of sounds) in the HUT study. The main uses of SM are described as:

- simulation and analysis of existing sounds
  - copy synthesis (audio coding)
  - study of sound phenomena
  - pitch shifting, time scale modification

[Tolonen et al. 1998]

Changing the duration of a signal can be achieved by resampling audio at a different rate to that at which it was recorded. For example, if an audio sequence recorded at 48 kHz is replayed at 24 kHz, the duration of the signal will be doubled. However, when time scaling is performed in this way there is also a reciprocal shift in pitch. Since each individual oscillation now takes twice as long to occur frequency (and hence the overall pitch) is halved. This relationship between frequency and time is fixed since they are the reciprocals of each other. The terms time-scaling and pitch-shifting usually refer to time-scaling independent of pitch and pitch-shifting independent of time and these can only be achieved by analysis-modification-resynthesis of the signal[4]. Since SM involves the description of how the spectral characteristics of sound components vary over time the model is clearly amenable to time and pitch scaling/shifting independently of each other. The quality of the resultant scaled or shifted sound is dependent upon the quality/appropriateness of the model and the scaling/shifting algorithm employed.

SM is the main subject of this thesis and much of this chapter is devoted to common spectral analysis, modelling, and transformation techniques. Before this other sound models are briefly considered.

3.3.4 Other sound models

There are many different taxonomies of synthesis and sound modelling. Serra makes the distinction between PM, SM and ‘abstract’ models which he defines as “models, such as

[4] The term ‘pitch-scaling’ here can be confusing since, as discussed in the previous chapter, pitch is defined as an attribute of perception related to frequency. Here the ‘physical pitch’ of a combination tone is defined as the frequency of simple tone that would produce the same perceived pitch in the listener.

frequency modulation [which] attempt to provide musically useful parameters in an abstract formula" [Serra, 1997]. Smith adopts this taxonomy but adds a fourth group, 'processed recordings', and refers to abstract techniques as algorithms rather than models which is more in keeping with the definition of sound model given earlier in this chapter [Smith, 1991].

Modulation synthesis produces sound by modulating the amplitude or frequency of a waveform (the carrier) with the amplitude of a second waveform (the modulator). To distinguish this form of modulation from vibrato (slow frequency modulation) and tremolo (slow amplitude modulation) modulation synthesis produces spectral rather than temporal changes to the carrier by using a modulator whose frequency is 20 Hz or higher. Ring modulation (RM) results from the multiplication of two signals and is amplitude modulation (AM) where the modulator is bipolar (i.e. it oscillates between positive and negative values). The term AM usually refers to the specific case of modulation by a unipolar signal (i.e. one whose output values are either wholly non-negative or non-positive). The output of RM with two sinusoidal signals with frequency f_m and f_c is a complex tone with partials at the frequencies $(f_m - f_c)$ and $(f_m + f_c)$. The output of AM is a complex tone with partials at the frequencies f_c, $(f_m - f_c)$ and $(f_m + f_c)$. Frequency modulation (FM) synthesis maps the amplitude of the modulator to the *frequency* of the carrier. The output of FM between two sinusoids is a spectrum with many more partials than for AM since there are spectral components at the sum and difference of f_c and f_m but also at the sum and difference of f_c and integer multiples of f_m. Both forms of synthesis allow spectrally rich sounds to be generated from two simple waveforms. Waveforms other than sinusoids can be used but the resultant output, particularly where both waveforms are non-sinusoidal, is difficult to predict. The original motivation for the use of FM for sound synthesis is that "in natural sounds the frequency components of the spectrum are dynamic, or time variant. The energy of the components often evolves in complicated ways; in particular during the attack and decay portions of the sound" [Chowning, 1973]. FM became hugely popular during the 1980s through commercial exploitation of this technology by the Yamaha Corporation realised in its DX range of synthesizers. FM is a computationally cheap means of producing synthetic sounds with rich spectra which can be easily time varied. However the synthesis parameters are not directly related to physical sound production (as for PM) or to the linear combination of spectral components (as for SM) and it is for this reason that the 'abstract' classification is used. The HUT study observes:

> The FM synthesis parameters are strong offenders in the criteria of intuitivity, physicality, and behaviour, as modulation parameters do not correspond to musical parameters or parameters of musical instruments at all, and because the method is highly non-linear. [Tolonen et al. 1998]

Despite these limitations accurate yet simple models of the sounds made by real physical instruments have been produced. A study of small Chinese and Tibetan bells uses a simple combination of AM and FM instruments to model the sound they produce and whose output sound is indistinguishable from the acoustic original [Horner et al., 1997].

3.4 A proposed taxonomy of sound synthesis and modelling

With so many different attributes of a synthesis or modelling system to consider, developing a functioning, clear-cut classification can be difficult, if not impossible. For a given situation one classification may be more appropriate than the other. It is the belief of this author that for many situations a split into two categories is the most appropriate and intuitive; models, processes and techniques based on the description of the vibrating body creating the sound and the environment in which it sounds and those based on the decomposition of a sound into simpler components and where these components occur in time and frequency. As intimated in the previous chapter these categories generally correspond to an acoustic or psychoacoustic conception of sound. These two areas are covered by the commonly used terms 'physical modelling' and 'spectral modelling' however, as stated previously, not all processes or methods explicitly create a sound model. For example, a process which takes the phase values from one input sound, the phase values from a second sound and uses these to produce a single output sound which represents a combination of the two input sounds, does not explicitly generate a model of the two inputs, but transforms them into another domain (the Fourier domain, which is itself based on a model of a periodic signal as a sum of stationary sinusoids) within which they are combined. So approaches to sound analysis, synthesis or processing are either concerned with how sound is produced or how sound is heard.

Included in spectral approaches are subtractive (including that based on audio samples) and additive synthesis as well as FM and AM since these are both concerned with generating a desirable spectral output rather describing the behaviour of vibrating physical object. Whilst the distinction between sound and sound generator is straightforward there are, inevitably, some scenarios in which the distinction is not clear. For example, some vocoders attempt to impart the slowly changing spectral envelopes of one sound on to the same spectral regions of a second sound. A classic application of this process is to produce a 'talking' instrument where the formants are extracted from a speech signal and applied to an instrumental signal. Extracting the formants from a speech signal is the equivalent of separating the exciter (the vocal folds in this case) from the resonator (the vocal tract) so it could be argued that this is a process based on the sound generator/modifier rather than the sound itself. However this is considered by this author to be a spectral process which also, in this particular example, has an interpretation in terms of the sound source. This classification is based on the fact that the process is spectrally conceived and the design/operation parameters of the process are not based on a consideration of the 'physical causes' of the sound.

3.5 Fourier analysis of time series

3.5.1 The Fourier series and transform

Fourier's theorem [Robinson, 1982] states that a periodic function (i.e. a function whose patterns of successive values repeat at the same intervals) can be represented as the weighted sum of an infinite series of cosine and sine functions, which are integer multiples of the fundamental frequency of the function ω, plus a constant term:

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos(n\omega t) + \sum_{n=1}^{\infty} b_n \sin(n\omega t) \qquad (3.6)$$

$$\equiv A_0 + \sum_{n=1}^{\infty} A_n \sin(n\omega t + \phi_n) \text{ where } A_n = \sqrt{a_n^2 + b_n^2}, \phi_n = \arctan\left(\frac{a_n}{b_n}\right) \qquad (3.7)$$

The set of sine and cosine functions $\{1, \cos\omega t, \sin\omega t, \cos 2\omega t, \sin 2\omega t, ..., \cos n\omega t, \sin n\omega t\}$, where $n \in \mathbb{Z}$ and ω is the radial frequency, are orthogonal to each other over the interval $-\pi$ to π. A set of real functions $\phi_n(x)$ that are piecewise-continuous[5] on an interval $x_1 \le x \le x_2$ are defined as orthogonal over that interval if

$$\int_{x_1}^{x_2} \phi_n(x)\phi_m(x)dx = 0, (n \ne m) \qquad (3.8)$$

Which can be easily shown for cosine and sine terms:

[5] A piecewise continuous function is continuous on all but a finite number of points. [www.mathworld.com]

$$\int_{-\pi}^{\pi} \cos(n\omega t)dt = \begin{cases} 0, (n \neq 0) \\ 2\pi, (n = 0) \end{cases} \quad (3.9)$$

$$\int_{-\pi}^{\pi} \sin(n\omega t)dt = 0 \qquad (3.10)$$

$$\int_{-\pi}^{\pi} \sin(n\omega t)\cos(m\omega t)dt = \begin{cases} \pi\delta_{mn}, (m, n \neq 0) \\ 2\pi, (m = 0, n = 1) \end{cases} \quad (3.11)$$

$$\int_{-\pi}^{\pi} \cos(n\omega t)\cos(m\omega t)dt = \pi\delta_{mn} \qquad (3.12)$$

$$\int_{-\pi}^{\pi} \sin(n\omega t)\sin(m\omega t)dt = \pi\delta_{mn} \qquad (3.13)$$

Where δ_{mn} is the kroenecker delta function defined as:

$$\delta_{mn} = \begin{cases} 1, m = n \\ 0, m \neq n \end{cases} (3.14) \qquad \text{[James et al, 1999]}$$

The coefficients in (3.6) for a periodic function with period $2\pi/\omega$ are given by:

$$a_n = \frac{\omega}{\pi} \int_{0}^{\frac{2\pi}{\omega}} f(t)\cos(n\omega t)dt, n \in \mathbb{Z}-* \quad (3.15)$$

$$b_n = \frac{\omega}{\pi} \int_{0}^{\frac{2\pi}{\omega}} f(t)\sin(n\omega t)dt, n \in \mathbb{Z}+ \qquad (3.16)$$

where $\mathbb{Z}-*$ is the set of non-negative integers and $\mathbb{Z}+$ is the set of positive integers

(3.6), (3.15) and (3.16) can also be expressed in complex form [Weisstein, 2006]:

$$f(t) = \sum_{n=-\infty}^{\infty} A_n e^{jn\omega t} \qquad (3.17)$$

$$A_n = \frac{\omega}{2\pi} \int_0^{\frac{2\pi}{\omega}} f(t) e^{-jn\omega t} \quad (3.18)$$

$$A_n = \begin{cases} \frac{1}{2}(a_n + jb_n), (n < 0) \\ \frac{1}{2} a_0, (n = 0) \\ \frac{1}{2}(a_n - jb_n), (n > 0) \end{cases} \qquad (3.19)$$

It can be seen that from (3.6) that values for a_n and b_n occur at integer multiples of the fundamental frequency giving a line spectrum. This is the case when the function being analysed is strictly periodic and the period is $2\pi/\omega$. Figure 3.1 illustrates a continuous period function in time and its corresponding line spectrum.

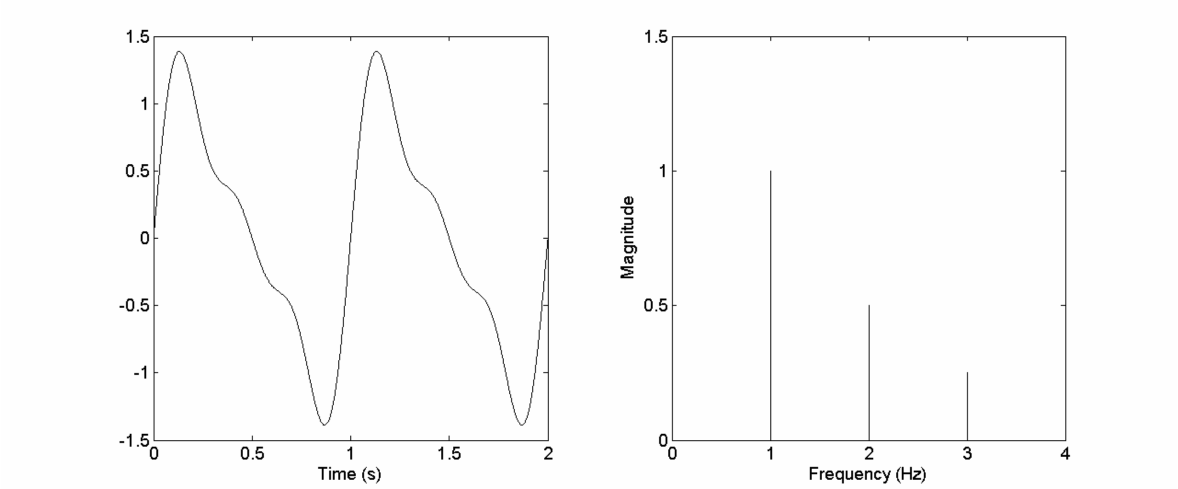


Figure 3.1: A periodic function in time and its corresponding line spectrum.

In order to decompose an aperiodic signal the discrete summation of harmonically related sine and cosine functions becomes an integral as the period of the signal tends to infinity and the difference in frequency between each component tends to zero. The Fourier transform generalises the Fourier series for periodic and aperiodic signals. Adapting the complex exponential form of the series, (3.17) and (3.18), to the continuous case the Fourier transform is given by:

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{j\omega t} d\omega \qquad (3.20)$$

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}dt \qquad (3.21)$$

Equation (3.20) represents the Fourier Transform and (3.21) represents the inverse Fourier Transform where the series of coefficients A_n has been replaced by the function of frequency $F(\omega)$.

3.5.2 The discrete Fourier transform

As discussed in the previous chapter when a continuous signal is sampled its spectrum becomes periodic which is not the case for the continuous signals considered thus far in this section. The discrete Fourier transform (DFT) is a discrete-time version of the Fourier Series and is given by:

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]e^{\frac{j2\pi kn}{N}} \qquad (3.22)$$

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-\frac{j2\pi kn}{N}} \qquad (3.23)$$

As implied by (3.22) and (3.23) the DFT is invertible meaning that a time series can be completely recovered from its DFT by the inverse DFT (IDFT) within the limits of the numerical resolution of whatever processor is used to perform the calculations. The DFT is computationally expensive since to find the spectrum for all values of *k* requires N^2 complex 'multiply and add' operations. However, for particular lengths of input sequence a number of repeated (hence redundant) calculations can be removed leading to much faster algorithms for computing the DFT. These are known as fast Fourier transforms (FFT) and were first introduced into DSP literature in the 1960s [Cooley and Tukey, 1965]. The FFT requires $N\log_2(N)$ calculations making it $\frac{N}{\log_2(N)}$ times faster than the DFT[6].

As already seen, when using Fourier analysis a signal can be decomposed into a set of weighted sines and cosines or as single set of weighted sinusoidal functions each with its own independent phase offset. The DFT of a sequence of *N* real numbers produces a set of *N*

[6] It should be noted that this expression is based on the number of complex multiplications required and how this relates to relative computational efficiency may vary according to the programming language and hardware being used [Lynn and Fuerst, 1994].

complex numbers, apparently doubling the amount of data required to represent the signal in the transform domain, but there is redundancy within the complex data in the form of complex conjugate symmetry. The spectrum is reflected as shown in figure 3.2 around the point $\left(N/2\right)+1$. Therefore the number of useful separate spectral regions (known as analysis 'bins') is $\left(N/2\right)+1$.

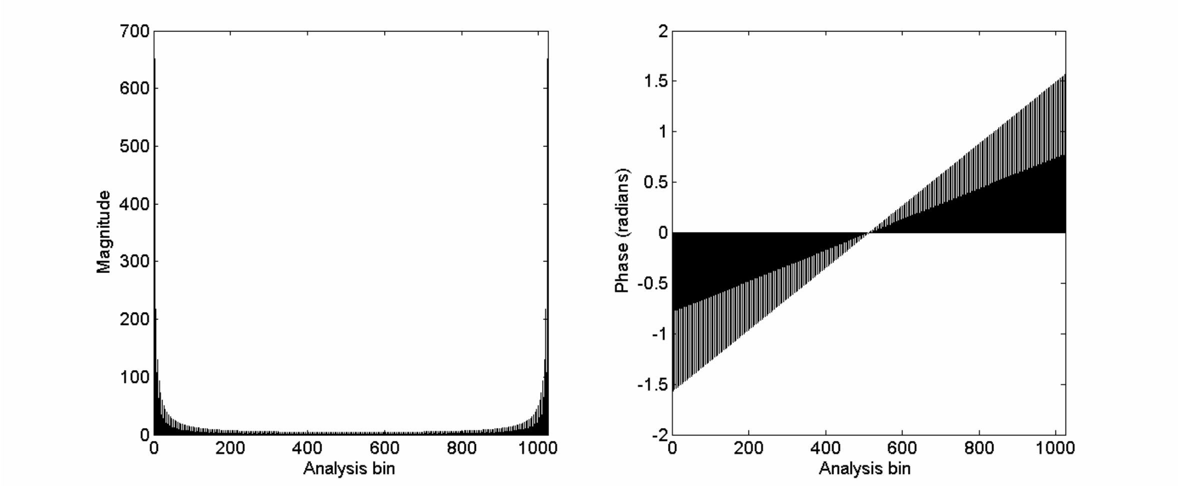


Figure 3.2: The magnitude (left) and phase (right) of the DFT of a real valued sequence showing complex conjugate symmetry.

Clearly, as the value of *N* increases there are more frequency bins in the output analysis. The number of bins is related to the frequency resolution of the analysis by the rate at which the time series was sampled. For a fixed number of samples, the lower the sample rate the higher the resolution (within the Nyquist limit) and *vice versa*. The longer the duration, in time, of a time series then the greater the resolution of the analysis will be. Any signal, continuous or discrete, is subject to the uncertainty principle, also known as the time-bandwidth product theorem which is defined as:

$$TB \geq \frac{1}{2} \qquad (3.24)$$ where:

$$\begin{aligned} T^2 &= \int \left(t - \langle t \rangle\right)^2 |f(t)|^2 dt \\ B^2 &= \int \left(\omega - \langle \omega \rangle\right)^2 |F(\omega)|^2 d\omega \end{aligned} \qquad (3.25)$$

where $\langle t \rangle$ and $\langle \omega \rangle$ are the mean values of t and ω and $|f(t)|^2$ and $|F(\omega)|^2$ are the energy at an instant in time and at a point in the spectrum respectively. This principle can be informally summarised as “a signal component cannot simultaneously have a narrow bandwidth in frequency and short duration in time” [Cohen, 1994].

3.5.3 The short-time Fourier transform for the time-frequency analysis of time series

The practical implication of the uncertainty principle for the DFT is that analysis of a shorter time series will yield an analysis spectrum with lower resolution. An assumption inherent in Fourier analysis, as can be seen from the previous equations, is that the sinusoidal functions are stationary, since the arguments of those functions are linear. Little, if any, meaning is conveyed in a sound signal comprised of components whose parameters do not change over time. Taking the DFT of an entire non-stationary audio signal does not lose any of the information about how the signal changes over time (since the process is invertible) but it is encoded in the relative phases of stationary functions and deriving a single non-stationary sinusoidal function from the superposition of a number of functions with different frequencies and relative phases is certainly non-trivial if not impossible.

It is useful to consider the following discussion of stationarity and non-stationarity in signals:

> Since non-stationarity is a negative property, its simplest definition refers to the corresponding property: that of *stationarity*. Although this concept is theoretically well-defined only in a stochastic framework, intuition attaches to it some sense for both deterministic and stochastic signals: loosely speaking, a signal is considered to be stationary as soon as its relevant properties remain relevant the same throughout all time. In a deterministic context, these relevant properties are mostly related to the instantaneous behaviour of the signal regarding amplitude and frequency … In accordance with what intuition suggests, a deterministic signal can then be considered as stationary if it consists in a superposition of components such that their instantaneous behaviour … does not depend on time. In such a case, Fourier analysis is sufficient for a satisfactory and meaningful description. [Flandrin, 1989]

In order to depict the non-stationarity in a signal as a smaller number of stationary functions the signal is divided up into shorter sub-signals, known as frames, and a DFT analysis is performed on each one. This analysis method more readily shows non-stationarity by showing a series of snapshots during which the signal is assumed to be stationary. As each snapshot corresponds to a different point in time it can be seen how the frequency content of a signal changes over time, therefore it is a form of ‘time-frequency’ analysis. This method of time-frequency analysis is known as the short-time Fourier transform (STFT) and it forms the basis of many spectral processing tools for audio and music composition. This thesis investigates how non-stationarity in both deterministic and stochastic components of a single analysis frame can be modelled in a real-time system.

The STFT of a continuous real signal can be mathematically expressed as:

$$F(t,\omega)=\int_{-\infty}^{\infty} f(\tau-t)\gamma(\tau)e^{-j\omega t}d\tau \qquad (3.26)$$

where $\gamma(t)$ is a windowing function applied to the signal. For a discrete signal it is:

$$X[rI_a,k]=\sum_{n=-N/2}^{N/2-1} x[rI_a+n]\gamma[n]e^{-\frac{j2\pi kn}{N}} \qquad (3.27)$$

where rI_a is the position of the analysis window, *r* is the window number (i.e. the index of each sampling of the spectrum) and *I* is the step size of the analysis window. I_a is given by the window length (*N*) divided by the window overlap which is the number windows that have non-zero values at any one sample. γ usually has compact support so the windowing process localises the signal in time around the centre of the window. The degree of localisation is dependent upon the width and shape of the window. Due to the uncertainty principle a signal cannot simultaneously be highly localised in time and frequency and this is negotiated in the choice of window length and shape. The most straightforward window shape is the rectangular window which is defined as:

$$\gamma_{\text{rect}}(t)=\begin{cases}1,\left(-\frac{N}{2}\le t\le\frac{N}{2}\right)\\ 0,(\text{elsewhere})\end{cases} \qquad (3.28)$$

The shape of this window in the frequency domain is given by:

$$\Gamma(\omega)=\int_{-N/2}^{N/2} e^{-j\omega t}dt=\left[\frac{e^{-j\omega t}}{-j\omega}\right]_{-N/2}^{N/2}=\frac{e^{j\omega\frac{N}{2}}}{j\omega}-\frac{e^{-j\omega\frac{N}{2}}}{j\omega}=\frac{\sin\left(\omega\frac{N}{2}\right)}{\frac{\omega}{2}} \qquad (3.29)$$

which is the sinc function (discussed further in 3.8.6) [Harris, 1978]. Figure 3.3 shows a 128 sample rectangular window and the magnitude of its DFT. The DFT has been zero-padded and shifted so that centre of the window appears at bin zero. Zero padding is explained later on this section.

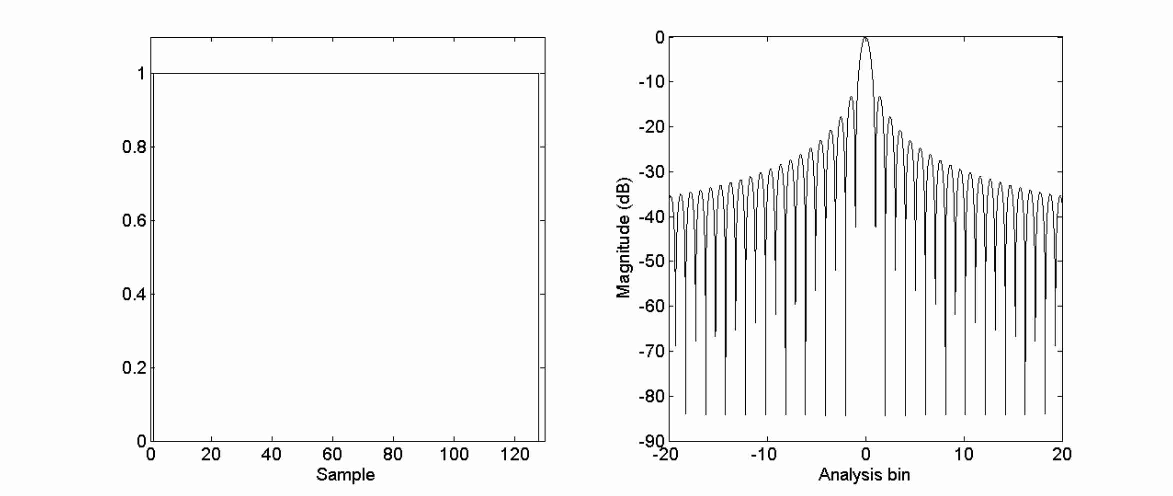


Figure 3.3: Rectangular window and its DFT magnitude.

If modulated by a sinusoid this window shape is shifted so that the centre of its main lobe lies at the frequency of the sinusoid. It can be seen from the spectral plot of the window function that a single sinusoid will not produce a single vertical line in the spectrum but is spread in frequency. A single line will only be produced by a single sinusoid where the rectangular window is used and the window length is an exact multiple of period of the sinusoid. The spectrum of the rectangular window has a relatively narrow main (centre) lobe but the magnitude of the side lobes decays relatively slowly. This is the main trade-off in window design; that between main lobe width (often referred to as the ‘thickness’ of the spectral line representing a sinusoid) and the rate at which the amplitude (or magnitude) of the side lobes decays.

The window function that is used for STFT analysis in this thesis is the Hann window, named after Julius von Hann and also known as the von Hann, Hanning and raised cosine window. It is defined as follows in the time and frequency domains respectively:

$$\gamma_{\mathrm{Hann}}(t) = \frac{1}{2} + \frac{1}{2}\cos\left(\frac{t}{N}\right), \left(|t| \le \frac{N}{2}\right) \quad (3.30)$$

$$\Gamma_{Hann}(\omega) = \frac{\sin\left(\frac{N\omega}{2}\right)}{N\omega\left(1 - \frac{N^2\omega^2}{4\pi^2}\right)} \quad (3.31)$$

Its time and spectral magnitudes are shown in figure 3.4 (as for figure 3.3). It can be seen that, compared to the rectangular window, the main lobe is wider but the side lobes decay at a greater rate [Nuttall, 1981].

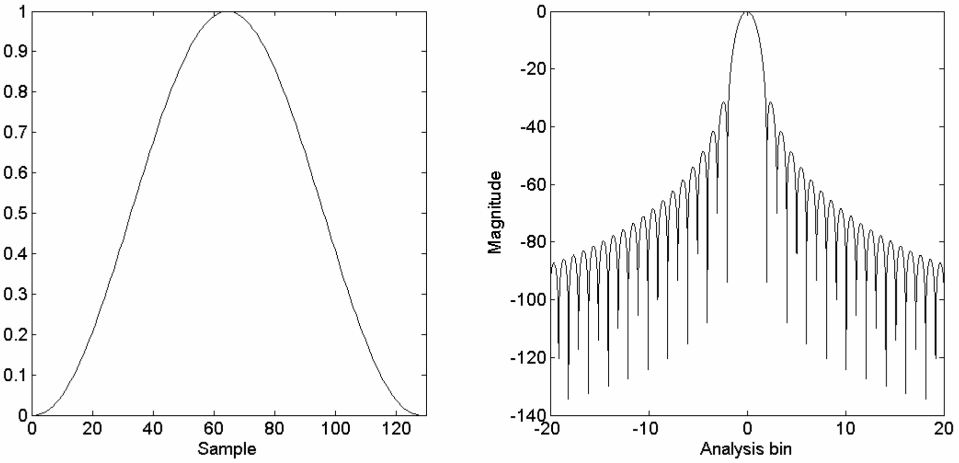


Figure 3.4: Hann window and its DFT magnitude.

Windows applied in this way will have a linear, rather than a constant, phase characteristic around sinusoidal peaks. This effect can be ameliorated by a technique known as zero-phase windowing. Here an odd number of samples is windowed and then the window is circularly shifted so that the centre of the window is shifted to the beginning of the time series. This has the effect of keeping the phase constant around a stationary sinusoidal peak.

Whilst increasing the length of the windowed time series increases the spectral *resolution* due to the trade-off between frequency and time greater spectral *definition* can be achieved with the same length window by augmenting the input to the DFT with zeros (zero-padding). The output of the DFT is a more finely sampled version of the output of a non-zero padded DFT, the overall shape is still the same and the resolution of the analysis is no greater (i.e. the main lobe and side lobes are still of the same level and spread over the same range of frequencies) but the spectrum can be seen in more detail since more data points are available per unit frequency.

3.6 The phase vocoder

The term ‘vocoder’ is from ‘voice coder’, so called since an early application of a such a system was to reduce the bandwidth of a speech signal by separating the excitation part of the signal (generated by the vocal folds) from the resonator part (filtering by the shape of the vocal tract). The resonator part of the signal is separated by applying a bank of band pass filters to the signal and then low pass filtering the output of each of these to produce a set of amplitude envelopes for each band, known as spectral envelopes. Since the vocal tract changes shape much more slowly than the rate at which the vocal folds vibrate but it is the vocal tract that produces different phonemes, the much lower bandwidth spectral envelopes of the resonator can be transmitted/stored and the speech can then be synthesized using a new excitation signal modified by a bank of filters controlled by these spectral envelopes. The vocoder has also been used extensively in electroacoustic music to produce a hybrid between two audio signals by taking the excitation part of one signal and filtering it using the spectral envelopes of a second. A well known example is Wendy Carlos's use of a ten band vocoder developed by Robert Moog for an electronic realisation of Beethoven's Ninth Symphony in the 1971 film *A Clockwork Orange*.

Whereas the filter outputs of a 'channel' vocoder are real for a real signal, the phase vocoder uses complex filters to provide phase as well as magnitude at the output of the filters [Dolson, 1986]. Since the frequency of a single sinusoid is the first derivative of phase this allows the frequency of the underlying sinusoid to be estimated, although this assumes that the output from a single filter is due to a single sinusoid. If the output of the filter is due to more than one sinusoid then the phase and magnitude values are still interpreted by the system as if they were due to only a single sinusoid. An important difference between the channel and phase vocoder is that with the latter the output, without modification, is identical to the input [Moorer, 1978].

A phase vocoder for digital signals can be implemented with the STFT, the magnitude and the phase for a given analysis bin (*k*) at a given sample (*rI*), simply being the modulus and argument of the complex output of (3.27). In order to obtain the frequency of the underlying sinusoid the difference in the unwrapped phase is taken between adjacent analysis frames. Then this phase increment, expressed as a fraction of the phase for one whole period, is added to the bin number:

$$\Delta\theta_k(rI) = \arctan\left(\frac{\sin(\theta_k(rI) - \theta_k((r-1)I)}{\cos(\theta_k(rI) - \theta_k((r-1)I)}\right) \quad (3.32) \text{ [Moorer, 1978]}$$

$$\omega_k = \left(k + \frac{\Delta\theta_k}{\pi}\right)B \qquad (3.33)$$

where B is the width of the analysis bin:

$$B = \frac{2\pi F_s}{N} \qquad (3.34)$$

and F_s is the sample rate.

3.6.1 Overlapping windows and over-sampling the spectrum

Although the non zero-padded DFT does not itself produce an increase in the data required to represent a signal, the STFT with overlapping windows does. If no modification is to be performed on the data in the frequency domain then non-overlapped rectangular windows will be adequate. This is the critical sampling case for the STFT, if there is any overlap between windows then there will be redundancy in the transform data and if there is any gap between the windows then the signal cannot be perfectly reconstructed. However, the purpose of the phase vocoder in the context of this thesis is the modification of audio, and it is generally necessary to over-sample the frequency spectrum, giving a redundant representation, in order to perform high-quality modifications. There are a number of reasons for this.

For fast side lobe attenuation the window should not introduce discontinuities at the edges and therefore should be tapered. Without overlap tapered windows will introduce amplitude modulation into the reconstructed signal so overlapping is required. Also, because of the spreading in frequency caused by windowing the signal a single sinusoid will appear in more than one bin. Considering (3.33), with no frame overlapping a value can be obtained in the correct range for the peak bin but for the two adjacent bins the deviation value could be in the range of ± 1.5 and this range increases with increasing distance from the peak. Such deviations lead to phase offsets in the range of $\pm 3\pi$ which will produce incorrect deviation measurements (for example frequency offsets of 0.5, 1.0 and 1.5 will each give a phase offset of π). Such incorrect deviation measurements will lead to alias frequencies appearing in our analysis. With an overlap factor of 4 frequency deviations of 0.5B, 1.0B and 1.5B will give phase offsets of $\pi/4$, $\pi/2$ and $3\pi/4$ (since the phase only has a quarter of the time to increment) which are unambiguous. Overlapping windows *over-sample* the spectrum and so offer control over aliasing frequencies around sinusoidal peaks.

The greater the overlap employed in the time domain, the greater the range of bins either side of a sinusoidal peak that give a correct estimate of the frequency of that sinusoid. Table 3.1 gives estimates for the peak bin and its eight closest neighbours for a stationary sinusoid of 1 kHz with for overlaps of 1x and 4x. It can be seen that there is agreement for the frequency estimate across a greater number of bins where there is a greater overlap.

	1 x overlap		**4 x overlap**	
bin	Magnitude	frequency estimate (Hz)	magnitude	frequency estimate (Hz)
peak - 4	0.4	827.7	0.4	827.7
peak – 3	0.9	870.8	0.9	827.7
peak – 2	3.0	913.9	3.0	827.7
peak – 1	43.7	956.9	43.7	1000.0
peak	123.9	1000.0	123.9	1000.0
peak + 1	85.0	1043.1	85.0	1000.0
peak + 2	6.8	1086.1	6.8	1000.0
peak + 3	1.4	1129.2	1.4	1172.3
peak + 4	0.5	1172.3	0.5	1172.3

Table 3.1**:** Frequency estimates close to peak for different overlaps (figures given to 1 decimal point).

3.6.2 Time and pitch scaling with the phase vocoder

Synthesis of a time-series from STFT data is achieved by the following equation:

$$x[rI_s + n] = \frac{1}{N}\sum_{k=0}^{N-1} X[rI_a, k]e^{\frac{jkn}{N}} \qquad (3.35)$$

where I_s is the synthesis hop size. Perfect reconstruction is achieved when $I_s = I_a$, time scaling is performed when $I_s \neq I_a$. When time scaling is performed the phase of the STFT data is interpolated by a phase propagation formula to ensure that the phase increment for the new hop size is consistent with the instantaneous frequency of its respective bin. Rearranging (30) and introducing a time-scaling factor *T*, where for $T > 1$ the resynthesized sound is longer than the original, for $T < 1$ it is shorter than the original and for $T = 1$ the durations are identical, gives:

$$\Delta\theta_{k(synthesis)} = T(\omega_k - k) \qquad (3.36)$$

$$T = \frac{I_S}{I_a} \qquad (3.37)$$

Pitch-scaling can be achieved by time-scaling with the above method and then resampling the output. For example, in order to shift the overall pitch of a time series up by an octave (i.e. double it) it is first stretched using STFT analysis and resynthesis with *T* = 2 and then the modified time series is decimated (every other sample is removed from the sequence) to give a final series that is the same length as the input but which plays at twice the pitch due to the decimation [Laroche and Dolson, 1999].

One problem with this method is that where a component is not a stationary sinusoid the estimated instantaneous frequency for each bin around a peak will not be exactly the same and so phase errors result which results in an un-focussed quasi-reverberant sound which is referred to as being ‘phasey’ [Laroche and Dolson, 1997]. Another cause of this effect is sinusoidal components which are slowly varying in frequency crossing into adjacent bins between analysis frames and so causing a lack of continuity in the phase progression (since phase differences between frames are taken between the same bin in each frame). A third contribution to this problem is that, as stated previously, the correct evaluation of the instantaneous frequency for a bin surrounding a peak depends on the distance of the bin from the peak bin and the degree of over-sampling of the spectrum. Although the windows used for such applications are designed to focus energy in the main lobe[7] there may still be noticeable energy contribution outside these bins and, since the instantaneous frequency is not correctly evaluated for these bins, the modified phase trajectory will be incorrect. The reverberant sound produced by lack of phase coherence between bins “is simply an artefact

[7] For example, the peak side lobe level of the 4 term Blackman-Harris window is 92 dB below the peak level of the main lobe, although the trade-off is a main lobe which is twice as wide as that of the Hann window requiring double the overlap to correctly resolve frequencies for bins within the main lobe [Nuttall, 1981].

caused by beating between adjacent channels as their relative phases change" [Puckette, 1995].

3.6.3 Cross synthesis with the phase vocoder

A simple channel vocoder effect can be obtained by combining the phase values from the STFT of one input signal with the magnitude values of the STFT of a second input, provided the window length is sufficient that successive magnitude values to do not track the oscillations of the underlying sinusoids but just their amplitudes. In this situation the magnitude input is the equivalent of the slowly varying control of the amplitude which is applied to the sinusoids whose frequency is determined by the phase input so the choice of window length is important: too short and the magnitudes of lower bins will follow the oscillations of lower frequencies, too long and changes in the amplitude envelopes will be smoothed out and not tracked correctly. At a sample rate of 44.1 kHz a window size of 2005 samples will smooth amplitude changes that occur faster than 20 Hz.

The above method is often effective for producing a useful hybrid output when one of the inputs is pitched and homophonic. However, where both inputs have pitched components and the spectral peaks of these components do not overlap then large magnitudes will be combined with a phase which may be influenced by a distant bin. This is likely to produce a loud alias frequency which is not common to either input sounds. A method for overcoming this and providing good separation between spectral content and spectral envelope in signals to be cross-synthesized in this way, known as frequency shaping has recently been proposed and implemented as the *Shapee* process[8]. This method divides the spectrums of both sounds in to 'shaping regions' which are the width of the main lobe (4 bins for the Hann window although the user may override this shaping region width). For each shaping region an overall magnitude weighting is calculated:

$$S(r)=\frac{\sum_{n=0}^{w} M_e(rw+n)}{\sum_{n=0}^{w} M_f(rw+n)}, r=0,1,2,...,\frac{\frac{N}{2}-1}{w} \qquad (3.38)$$

[8] A real-time VST plug-in implementation of *Shapee* has been produced by this author as part of initial investigations into sound hybridisation methods and is available for download free of charge at www.jezwells.org.

Where w is the width, in bins, of the shaping regions, N is the number of analysis bins, $M_e(k)$ and $M_f(k)$ are the magnitude of the *i*'th bin of the spectral envelope reference signal and the frequency reference signal respectively. The output spectrum is then defined by:

$$M_{output} = M_f(k)S(\tfrac{k}{w}) \quad (3.39)$$

$$\theta_{output}(k) = \theta_{frequency}(k) \quad (3.40)$$

The designer of this algorithm states that "the effect of scaling $M_f(k)$ by this ratio is to extract the spectral envelope from $M_e(k)$ while maintaining the localised frequency information inherent to $M_f(k)$" [Penrose, 2001]. What *Shapee* does demonstrate very clearly is the importance of the relative magnitudes around a sinusoidal peak as well as the phase values for reconstructing components of the correct frequency.

Often when modifying STFT data, particularly when combining the magnitude and phase of two completely separate sounds in this way, discontinuities will be introduced at the edges of the window where a change in phase has introduced a circular shift of the time domain waveform. If these discontinuities do not exactly match up with those from a preceding or succeeding frame at the overlap-add stage of the inverse STFT then an audible click is likely to result. To remedy this the output frames of the STFT can be windowed prior to overlap-add in order to remove theses discontinuities. A property of the Hann and Hamming windows is that if an overlap of $\frac{N}{l} \geq 2$ is used then, in addition to the overlapping windows summing to a constant, the power of that window, up to $\frac{N}{l}-1$, will also sum to a constant [Puckette, 1995]. This means that, with sufficient overlap, this additional windowing may be applied without generating amplitude modulation at the output.

It is not possible to cover every sound modification and combination process that makes use of phase and magnitude data from the STFT here but time and pitch scaling and traditional vocoding/hybridisation are the purpose of many of these.

3.6.4 Applying a spectral model to phase vocoder sound modification

A spectral model can be used within a phase vocoder process even if the output signal is not to be synthesized with sinusoidal oscillators and/or filtered noise. Here spectral modelling is applied to find which parts of the Fourier spectrum are due to stable sinusoids and which are due to stochastic or transient signal components. One process for pitch scaling only operates on those parts of the spectrum which are due to stable sinusoids since it is these components and not noise or transients that impart a sense of pitch to the listener [Laroche and Dolson, 1999].

Correct identification of stable sinusoids is one of the areas of investigation of this thesis. Current methods vary in their levels of sophistication. The Laroche-Dolson method simply defines a peak in the magnitude spectrum which is higher than that of its four nearest neighbours to be due to a sinusoid.

> This criterion is both simple and cost-effective, but might fail to distinguish local maxima due to the presence of a sinusoid from local maxima corresponding to the lobes of the Fourier transform of the analysis window. Any more refined scheme can be used instead, although it was found in practice that this very basic technique yields good results. [Laroche and Dolson, 1999]

One of the goals of this thesis is to develop and evaluate such 'more refined' schemes for achieving improved sinusoidal identification.

The modulation property of the DFT means that multiplication in the time domain (such as when a window is applied to time series) is equivalent to convolution in the frequency domain. Therefore, as already seen, a stable, windowed sinusoid in the time domain will appear as an impulse in the spectral domain convolved with the spectral shape of the window function itself. When a function is convolved with an impulse it is shifted so that its centre is at the same position on the frequency axis as the impulse. Therefore, providing the magnitude response of the windowing function is known, the shape of a sinusoidal function in the frequency domain when analysed with that particular window can be predicted. Clearly the magnitude response is at its maximum for the bin in which the sinusoid resides so one of the most straightforward ways of identifying a sinusoid is by searching for maxima in the magnitude response, and this is commonly the first step in sinusoidal identification (as for the Laroche-Dolson method). The relative magnitudes of surrounding bins can be used as a measure of how close a peak in the spectrum is to that of a stationary sinusoid. When analysing peaks it is common to consider those bins either side of the peak up to local minima on either side as a spectral region. The Motion Picture Engineering Group (MPEG) layer I and II lossy coder/decoders (or codecs) use a simple measure that considers the relative magnitude of the peak bin and close neighbours to determine whether a peak represents a sinusoid (or sinusoids very close together in frequency) [ISO/IEC, 1992]. A peak is considered to be a sinusoid if the following condition is met, where *X*(*k*) and *X*(*k*+*i*) are magnitude components of a 1024 sample analysis frame:

$$X(k) - X(k+i) \geq 7 \text{ dB} \qquad (3.41)$$

For MPEG layer 1, *i* is chosen as follows:

If $2 < k < 63$ then i = -2 and 2.

If $63 \leq k \leq 127$ then i = -3, -2, 2 and 3.

If $127 \leq k < 250$ then i = -6, -3, -2, 2, 3 and 6.

The different ranges are a crude method of accounting for increasing critical bandwidth with frequency. Values for *k* only span approximately half of the spectrum due to bandwidth constraints in low bit rate coding systems such as the MPEG system. Sinusoidal peaks rarely occur in isolation above 10 kHz in audio signals with more than one monophonic instrument and so these are transform encoded.

3.6.5 Modelling non-stationarity

One of the assumptions of the STFT is that the signal being analysed is stationary for the duration of each analysis frame. Many signals, such as those with vibrato or tremolo for example, have continuously varying frequency and/or amplitude and sinusoids which exhibit such behaviour have different magnitude responses to those which are stationary. Modulation has the effect of flattening the main lobe of the analysed sinusoid. Figure 3.5 shows the main lobes for a stationary sinusoid and one whose frequency is increasing linearly at approximately 172 Hz (the width of four analysis bins) and whose amplitude is falling exponentially at 2 dB per frame for a 1024 sample, 32 times zero padded DFT (sinusoid sampled at 44.1 kHz). As well as assisting with the identification of non-stationary sinusoids, knowledge of this change in shape of the spectral peak is important for estimating the amplitude of such signal components.

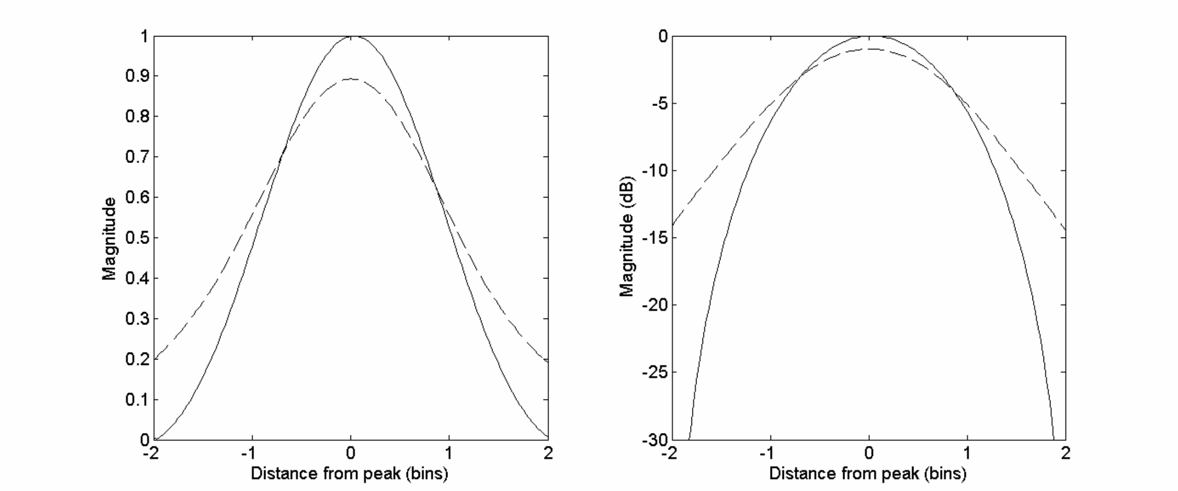


Figure 3.5: The effect of amplitude and frequency non-stationarity on window spectral magnitude

Given knowledge of the analysis window a 'sinusoidality' measure can be applied to a peak in the spectrum and its surrounding bins. This is a measure of the correlation between the actual bin magnitudes surrounding the peak and the window function shifted to the estimated frequency:

$$\Gamma_{peak} = \left| \sum_{f_{peak}+B}^{f_{peak}+B} H(f).W(f) \right| \qquad (3.42)$$

H(*f*) is the measured and normalised DFT and *W*(*f*) is the shifted and normalised window function in the frequency domain. *B* is the half bandwidth over which the correlation is measured (often the bandwidth of the main lobe of the window function) and so the number of points considered in the correlation measure depends on the degree of zero padding used in the DFT. Amplitude and frequency modulation effects can be accounted for by suppressing the modulation or by estimating it and adapting the window function W(*f*) accordingly. In order to suppress frequency modulation it must first be detected. A method for doing this is to estimate the fundamental frequency of a sound and track how this varies over time. This variation is then taken as being the same for all partials of the sound and is suppressed by time-varying re-sampling of the signal [Peeters and Rodet, 1998].

A method, known as phase distortion analysis (PDA), for estimating frequency and amplitude modulation analyses differences in phase between bins around peak in the DFT spectrum [Masri, 1996]. For a zero-phase symmetric window function modulated by a sinusoid the phase is shifted by π at the edge of each lobe. When the amplitude changes during the window then the main lobe is widened and the phase is no longer constant in the main lobe. For increasing amounts of exponential amplitude change, ΔA (measured in dB per frame), during a frame the phase difference between bins either side of the peak bin increases. For increasing linear frequency change, Δf (measured in bins per frame[9]), the phase difference between the peak bin and each adjacent neighbour increases. However beyond a certain linear frequency change the phase difference falls and therefore a unique frequency modulation value cannot be inferred from the PDA measure. It is suggested that the modulation value is assumed to be in the range of the first monotonically increasing part of the figure shown.

Since the phase difference between the peak and either adjacent neighbour is the same for frequency modulation, and the same magnitude but different sign for amplitude modulation the two can be separated and independent measures obtained:

$$\Delta f_{measure} = \Delta\phi_{peak-1} + \Delta\phi_{peak+1} \quad (3.43)$$

$$\Delta A_{measure} = \Delta\phi_{peak-1} - \Delta\phi_{peak+1} \quad (3.44)$$

where $\Delta\phi_{bin}$ is the difference in phase between the peak bin and *bin*, $\Delta f_{measure}$ is the Δf measure and $\Delta A_{measure}$ is the ΔA measure. These measures are related to the actual ΔA and Δf values by functions which are dependent upon the type of window. For a Gaussian window these functions can be analytically derived from phase and magnitude distortion measures around a peak [Peeters and Rodet, 1999]. However the Gaussian window, which must be truncated for short-time analysis has a wide main lobe (compared to the best least suqares fit Hann window) and is therefore not favoured for sinusoidal analysis [Marchand, 2000].

PDA has been used to estimate modulation parameters for sinusoids and to identify non-stationary sinusoids within a single analysis frame [Lagrange et al, 2002]. Once ΔA and Δf have been estimated a sinusoid with these modulations is synthesized and windowed and the DFT of this single windowed sinusoid is taken. The magnitude of this DFT around the peak is compared with that of the peak from which the ΔA and Δf estimates were taken using (3.42) thus extending this technique to non-stationary sinusoids. PDA, and its

[9] This means that the change in frequency is expressed as bin widths. For a 44.1 kHz audio signal with a window length of 1024 samples this width is approximately 43 Hz. The important thing to note is that for different window lengths the same phase distortion measure will be different.

application to the identification of non-stationary sinusoids is dealt with in detail and extended to time reassignment data in the next chapter of this thesis and so will not be discussed further here.

An alternative method has been proposed for estimating FM based on Fresnel integrals [Master, 2002]. Considering the following equation for a discrete sinusoid with linearly increasing frequency:

$$x[n] = \sin(\alpha n^2 + fn + \phi) \qquad (3.45)$$

it can be seen that the argument of the sine function is a quadratic. The integrals $\int \sin(x^2)dx$ and $\int \cos(x^2)dx$ are undefined suggesting that a function defining the shape of a windowed convolved with a sinusoid such as that in (3.45) cannot be found analytically. Fresnel integrals offer an accurate large limits approximation to the integral of sine and cosine functions with respect to a squared term.

$$C(u) \equiv \int_0^u \cos\left(\frac{\pi x^2}{2}\right)dx \approx \frac{1}{2} + \frac{1}{\pi u}\sin\left(\frac{\pi u^2}{2}\right) \qquad (3.46)$$

$$S(u) \equiv \int_0^u \sin\left(\frac{\pi x^2}{2}\right)dx \approx \frac{1}{2} - \frac{1}{\pi u}\cos\left(\frac{\pi u^2}{2}\right) \qquad (3.47)$$ [Gautschi, 1965]

Using these results the following equation for estimating α from the DFT of the Hann windowed signal is derived:

$$\hat{\alpha} \approx \frac{-j\Gamma_{Hann}(0)}{2}\left(\left(\frac{K}{2\pi}\right)^2 \frac{\Delta^2\Gamma_{Hann}(k)}{\Delta k^2}\right) \qquad (3.48)$$

where k is the bin corresponding to the peak in the spectrum due to the sinusoid being measured and K is the size of the (possibly zero-padded) window. Since the second order difference in this equation corresponds to a second order derivative in the continuous case K must be large for the approximation to hold [Master and Liu, 2003]. When the size of α and N (the length of the data used in each window, independent of zero-padding) is small α cannot be accurately estimated as an imaginary part appears although α should be wholly real. However "this imaginary part mimics the ratio of the real part to the correct α value, allowing us to solve the equation

$$x_1 + x_2 \Im\{\hat{\alpha}\} \approx \frac{\Re\{\hat{\alpha}\}}{\alpha} \qquad (3.49)$$

as a least squares problem to obtain optimal coefficients x_1 and x_2" [Masters and Liu, 2003]. The authors claim that this estimation method is robust to ΔA. However, since large amplitude changes drastically alter the time domain window shape and (3.48) is specifically formulated for the Hann window this author disagrees with this statement and experiments show that Δf estimates are significantly affected for large ΔA. PDA and the time reassignment equivalent proposed in this thesis are adopted because they offer estimates of both Δf and ΔA. Chapter 4 examines how the interaction between Δf and ΔA can be accounted for.

3.7 Gabor and Walsh-Hadamard analysis

So far only Fourier analysis has been considered for audio signals but, whilst it may be the most popular and best understood method for time-frequency analysis, the STFT is far from being the only method available. This section discusses these alternative analysis techniques and how the STFT fits into the overall context of time-frequency analysis. The material in this section also serves as a useful link between concepts introduced in previous sections on the STFT and those in subsequent sections on wavelets.

The STFT belongs to a class of time-frequency analysis methods known as atomic decompositions. These analysis methods compare a signal with an elementary signal component, known as an atom, which can be shifted in time and can have different frequencies, sequencies or scales [Auger et al, 1996]. For example the STFT atom is a window function modulated by sine and cosine functions at different frequencies shifted to different positions of the signal. The signal is compared to the modulated window at these different frequencies and positions. The correlation between the signal and the atom at each analysis frequency for a given point in time is given by the inner, or dot, product

$$c_{m,p} = \left\langle f(t), \psi_{m,p} \right\rangle = \int_{-\infty}^{\infty} f(m-t)\psi *_p (t)dt \qquad (3.50)$$

for a continuous signal and:

$$c_{m,p} = \left\langle x, \psi_{m,p} \right\rangle = \sum_n x[m-n]\psi *_p [n] \qquad (3.51)$$

for discrete signals where $\psi_{m,p}$ is the set of functions ψ_p shifted to the position m [Qian, 2002]. The asterisk in these equations denotes the complex conjugate.

Of course, any combinations of atoms may be used for specific analysis tasks. An example of this is a constant-Q analysis used for monophonic transcription [Brown, 1990]. Here sinusoids which are logarithmically spaced in frequency, and are windowed by Hamming functions of length inversely proportional to frequency, form the time-frequency atoms. This particular analysis system is designed so that fundamental frequencies corresponding to notes in the equal tempered scale (twelve geometrically spaced frequency intervals per ocatve) can be individually resolved. This requires long window lengths at low frequencies (greater than 6000 samples at a sampling rate of 32 kHz). For this analysis each time-frequency atom must be pre-computed and stored in memory and the transform is non-invertible due to insufficient sampling of the spectrum at higher frequencies, however the system is successful, in many monophonic, cases in isolating and describing individual note events.

3.7.1 Walsh-Hadamard analysis

Walsh functions are an orthogonal set of functions which take the value of 1 or -1. The sequency of the function is the number of zero crossings (transitions between 1 and -1) within a given time interval. Zero crossings may only occur at “fixed intervals of a unit time step” [www.mathworld.com]. Walsh analysis may be applied to continuous signals but it is highly suited to digital applications due to its binary structure. Walsh functions, in sequency (as opposed to natural) order, can be derived from the Hadamard matrix (hence the alternative name Walsh-Hadamard functions). A Hadamard matrix is a square matrix whose individual elements are either 1 or -1 and whose rows are orthogonal to each other (i.e. for an n by n matrix the inner product of a row with itself is n and the inner product of a row with any other row is 0) [Weisstein, 2006b]. The 1 by 1 Hadamard matrix is given by:

$$H_1 = |1| \qquad (3.52)$$

The 2n by 2n Hadamard matrix is given in terms of the n by n Hadamard matrix as:

$$H_{n+1} = \begin{vmatrix} H_n & H_n \\ -H_n & H_n \end{vmatrix} \qquad (3.53)$$

For a discrete series of length *N* the set of Walsh functions is the *N* by *N* Hadamard matrix. Whereas Fourier analysis can be used to efficiently describe a signal composed of highly localised frequency components Walsh functions are more suited to broadband signal components and as such have been used in applications such as speech coding to efficiently describe broadband components with the narrow band components described by Fourier coefficients [Spanias and Loizou, 1992]. Walsh analysis can also be performed by full binary tree decomposition with the Haar wavelet which is discussed later on in this section.

3.7.2 Gabor analysis and biorthogonality

In 1946 Gabor proposed representing signals jointly in time and frequency, in what he referred to as 'information diagrams', rather than simply as a function of either time or frequency. His diagrams represented continuous signals as collections of a finite number of information 'quanta'. These quanta are Gaussian functions modulated by sinusoids, although this definition can be relaxed and a Gabor representation can be based on any normalised window [Auger et al, 1996]. The form of the Gaussian function used is:

$$\gamma(t) = \sqrt[4]{\frac{\alpha}{\pi}} e^{-\alpha t^2 / 2} \qquad (3.54)$$

where α is a parameter that controls the variance of the function (it is inversely proportional to the time domain variance and proportional to the frequency domain variance). The time bandwidth product is:

$$TB = \frac{1}{\sqrt{\alpha}} \frac{\sqrt{\alpha}}{2} = \frac{1}{2} \qquad (3.55)$$

Therefore, recalling (3.24), this function is optimally localised in terms of the combined time and frequency domains. The Gabor expansion of a signal *f(t)* is:

$$f(t) = \sum_{m=0}^{M} \sum_{p=0}^{P} c_{m,p} \sqrt[4]{\frac{\alpha}{\pi}} e^{-\alpha(t-mT)^2 / 2} e^{jn\Omega t} \qquad \text{(3.56) (adapted from [Qian, 2002])}$$

where t and Ω are the distances between sampling steps in the time and the frequency domains respectively. For an expansion to definitely be possible the critical sampling density condition, $T\Omega \leq 2\pi$, must be met. Note that (3.56) is given in terms of terms of reconstructing the signal from the Gabor coefficients ($c_{m,p}$) and Gabor's work does not contain a practical method for deriving these coefficients [Qian, 2002]. If any signal can be completely reconstructed by multiplying the coefficients c_n from (3.50) or (3.51), by the same set of analysis functions, as in (3.57) where *A* is a scalar which is 1 for the orthogonal case, then that set of functions $\{\psi_n\}_{n\in Z}$ is an orthogonal or over-complete basis[10].

$$f(t) = \frac{1}{A}\sum_{n} c_n \psi_n \qquad (3.57)$$

It has been seen that any signal can be recovered from its STFT coefficients provided the windowing does not introduce undulations in the time domain envelope of the signal. Whilst the Gaussian window is optimally localised in the joint time-frequency domain it cannot be overlap-added to sum to a constant as the window functions discussed earlier, such as the Hann and Hamming windows, can. Because of this it is necessary to find a set of dual functions $\left\{\overset{\circ}{\gamma}_n\right\}_{n\in Z}$ such that

$$f(t) = \sum_{m=0}^{M}\sum_{p=0}^{P}\left\langle f(t), \overset{\circ}{\gamma}(t)_{m,p}\right\rangle \sqrt[4]{\frac{\alpha}{\pi}}\, e^{-\alpha(t-mT)^2 / 2}\, e^{jn\Omega t} \qquad (3.58)$$

In this case $\{\gamma_n\}_{n\in Z}$ and $\left\{\overset{\circ}{\gamma}_n\right\}_{n\in Z}$ are said to form a bi-orthogonal or over-complete bi-orthogonal basis. A valid dual function for the critically sampled Gabor expansion is:

$$\overset{\circ}{\gamma}(t) = \left(\frac{4\pi}{\alpha}\right)^{-\frac{1}{4}}\left(\frac{K_0}{\pi}\right)^{-\frac{3}{2}} e^{\frac{\alpha t^2}{2}} \sum_{n>\left|(t/T)-(1/2)\right|} (-1)^n e^{-\pi\left(n+\frac{1}{2}\right)^2} \qquad (3.59)$$

“where K_0 = 1.85407468… is the complete elliptical integral for the modulus $\frac{1}{\sqrt{2}}$”. [Bastiaans, 1980]. The functions described by (3.54) and (3.59) for *T*=1 and $\alpha = \pi$ and with no modulation are shown in figures 3.6 and 3.7. Both their time domain shape and the magnitude of the Fourier transform are shown. It can be seen that whilst the Gabor function is well localised in time and frequency the dual function has poorer localisation in both domains.

[10] An orthogonal basis is a critically sampled representation of a function (i.e. it contains the minimum amount of data capable of representing any function), an over-complete basis can represent any function and is redundant.

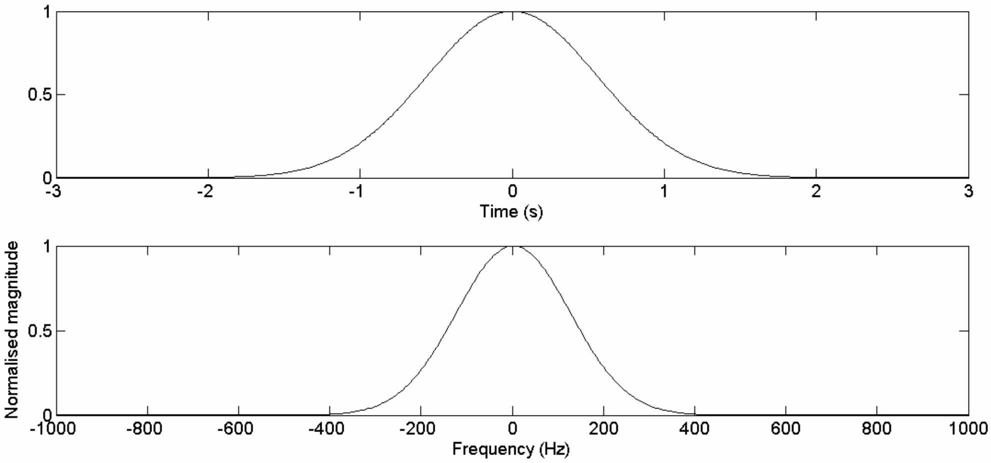


Figure 3.6: Gaussian function and its normalised magnitude spectrum.

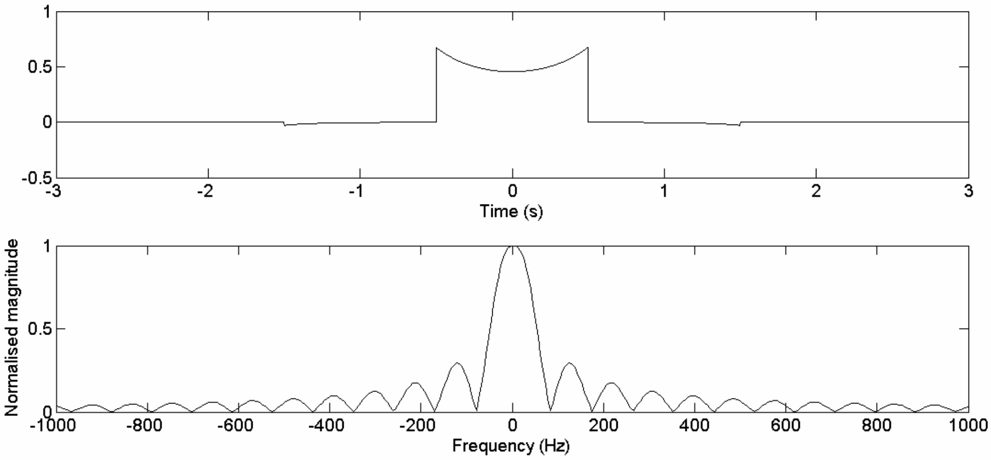


Figure 3.7: Dual of Gaussian function in figure 3.6 and its normalised magnitude spectrum.

The Gabor interpretation of sound as quanta of information centred at points in frequency and time has led to the field of granular synthesis, a large group of spectral processing and modelling methods, where a musical signal or sound is seen as being constructed from ‘grains’ of simpler sound structures localised in time and frequency [Cavaliere and Piccialli, 1997]. In this sense the STFT is implicitly a granular approach because it uses an infinite sinusoid (localisation in frequency) to modulate a window (localisation in time). The contribution of Gabor in 1946 and Bastiaans in 1980 has been to demonstrate that any signal can be represented by quanta optimally localised in time and frequency as a modulated Gaussian (Gabor grain) and that a dual function can be derived to determine the magnitude of the quantum at each point on a sampled time and frequency grid.

It is the orthogonal STFT using the Hann window that is used in this thesis for sinusoidal analysis and, although Gabor analysis in the form described here is not used, Gaussian atoms are used for residual analysis in the form of wavelets which are discussed in the following section. The discussion presented here is intended to demonstrate the link between the time-frequency analysis of wavelets and the STFT.

3.8 Wavelets

Wavelets are atoms which are *shifted* and *dilated* (as opposed to modulated which is the case for the atomic decompositions discussed so far) so that their inner product with the function, or series, being analysed can be found.

The continuous wavelet transform (CWT) is given by:

$$CWT(f(u,s)) = \int_{-\infty}^{\infty} f(t)\frac{1}{\sqrt{s}}\psi^{*}\left(\frac{t-u}{s}\right)dt \qquad (3.60)$$

where $f(t)$ is the analysed function, u is the centre of the analysis atom, s is the scale of the atom and $\psi(t)$ is the function describing the analysing wavelet [Mallat, 1999]. The scale of a wavelet is inversely proportional to its frequency[11]. At low scales the frequency of the wavelet is relatively high and its duration is relatively short, at high scales the duration is relatively long but the frequency is relatively low. This is a fundamental difference between wavelets and the atoms considered so far (which retain the same length regardless of their frequency). This is because of the dilation of the wavelet at different scales (the wavelet is usually defined at the lowest scale and then progressively dilated, although it could be defined at the highest scale and then seen as being progressively compressed).

Since, through dilation, the duration of the wavelet is extended but its centre frequency is lowered through a direct inverse relationship, the bandwidth of the wavelet varies linearly with the centre frequency. This linear relationship between bandwidth and centre frequency means that such analysis with shifted and scaled functions is 'constant Q' analysis[12] whereas

[11] Only certain wavelets have a constant instantaneous frequency, many exhibit a range of instantaneous frequencies over their duration. Reference is often made to the 'main' frequency (related to the number of zero-crossings) or centre frequency (the centre of the Fourier transform) of a wavelet.

[12] Bandwidth is a measure of the localisation of a signal component in frequency. This is often taken as the difference between the frequencies either side of a peak at which the magnitude response is 3 dB lower than that of the peak. Q stands for 'quality factor' and is the bandwidth divided by the centre frequency. Parametric equalisers in audio equipment such as mixing consoles provide control over the level of signal cut and boost at

the STFT provides constant bandwidth analysis. As discussed in the last chapter, the ERB of the auditory filter is approximately ‘constant Q’ from 100 Hz upwards. This leads to a dyadic division of the time-frequency plane as shown in figure 3.8 where, although the area which each wavelet at different scales occupies remains the same (as for the STFT at different frequencies), the time duration is halved and the bandwidth is doubled at each higher octave (unlike the STFT), changing the shape of the time-frequency ‘tile’.

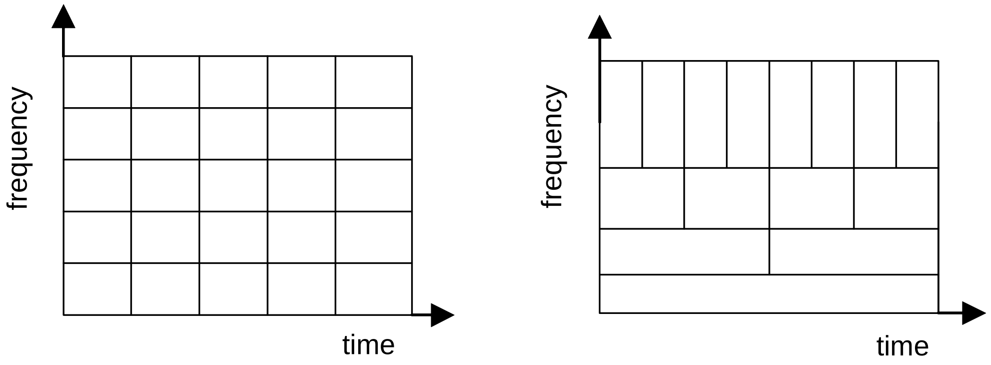


Figure 3.8: Uniform (left) and dyadic (right) time-frequency tiling.

This complementary change in time and frequency resolution is a key feature of wavelet analysis since long duration, frequency localised events can be depicted with, ideally, a single wavelet at a high scale and a short duration, high bandwidth, transient event can also be depicted, again ideally, by a single wavelet at a lower scale. Scaling can be likened to ‘zooming in’ to a very small part of a signal in order to determine the nature of sample by sample variations or ‘zooming out’ to look at trends in the signal over a larger part of its duration. By contrast STFT analysis uses the same ‘zoom’ level to look at both fast fluctuations and slower moving trends in the signal.

If the CWT is applied to identify certain components within a signal then the analysing wavelet can be of any required shape, such as the shape (in the time domain) of the components being searched for. If invertibility (perfect reconstruction) is required then the wavelet must satisfy two conditions. Firstly it must be admissible such that:

$$\int_{-\infty}^{\infty} \frac{|\Psi(\omega)|^2}{|\omega|} d\omega < \infty \qquad (3.61)$$

the peak frequency, the position of the centre frequency and the Q of the filter. Varying the centre frequency whilst maintaining the same Q causes the bandwidth to vary so such systems are ‘constant Q’.

where $\Psi(\omega)$ is the Fourier transform of $\psi(t)$ which implies that the $\Psi(0) = 0$ which means that the wavelet has zero mean in the time domain and is band pass in the frequency domain [Calderon, Grossman, Morlet cited in Mallat, 1999]. For complex wavelets an additional condition is that the wavelet must be progressive meaning that its Fourier transform must be zero for $\omega < 0$ (i.e. the wavelet should be analytic) [Grossman et al, 1989]. An intuitive explanation for these conditions is:

> Literally the term "wavelet" means *little wave*. In the most general context, a wavelet is a function that satisfies the main (time domain) conditions:
>
> 1. It has a small concentrated burst of energy in the time domain; and
>
> 2. it exhibits some oscillation in time
>
> The first condition makes the wavelet "little" in the sense that it is well localised in time, whereas the second condition makes it "wavy" and hence a *wavelet*. … Since a non-zero function with a zero-mean necessarily has some oscillation, the oscillation requirement is met. [Teolis, 1998]

3.8.1 The undecimated discrete wavelet transform

A common analysing wavelet for the CWT is the complex Morlet wavelet which is a modulated Gaussian function of which one form is given as:

$$\psi_{0,1}(t) = e^{-\frac{t^2}{2}} e^{j\omega t} \qquad (3.62)$$

Note that this analysing function differs from the Gabor function in (3.56) due to the dilation inherent in (3.60). In order to compute the wavelet coefficients for this, or any other kind of wavelet, a discrete implementation must be found. The first step is to produce a discrete version of (3.60), the discrete wavelet transform (DWT):

$$DWT(u,s) = \frac{1}{\sqrt{s}} \sum_n x[n] \psi^* \left[\frac{n-u}{s}\right] \qquad (3.63)$$

with $n,u,s \in \mathbb{Z}$. Note that in this form the discrete wavelet sequence $\psi[n]$ will require interpolation or zero-padding under dilation since non-integer sample values will result from this operation. An efficient implementation of (3.63) which uses zero-padding is the so-called 'algorithme à trous' (algorithm with holes). This algorithm was originally implemented in order to approximate the Morlet wavelet. This algorithm finds the wavelet coefficients at a given scale for each sample by convolving the signal with a filter whose coefficients represent the shape of the analysing wavelet. At scale 1 the signal is simply convolved with the wavelet filter, at the next scale the filter is dilated by the insertion of a zero between every sample before convolution with the signal and then at the next scale this dilated filter is dilated again, doubling its length by the insertion of zeros, and so on at each scale.

The advantage of this approach is that the number of non-zero filter coefficients remains the same at each scale, reducing the computational complexity. Since a filter dilated with holes is not a good approximation to a sampled version of the wavelet at a given scale use is made of a ‘pre-integrating’, low pass filter to perform interpolation on the signal at each scale before convolution with the dilated wavelet. Successive convolution of the signal with this filter at each scale (it is itself dilated) and then with the dilated (with holes) wavelet filter at a given scale is equivalent to the convolution of the signal with the actual wavelet at that scale [Dutilleux, 1988]. This iterative convolution with the dilated low pass filter for each scale and then convolution with the wavelet filter at the scale being analysed is common to both forms of discrete wavelet analysis: the decimated and undecimated discrete wavelet transforms.

From (3.63) it can be seen that a discrete wavelet transform implies a discretisation of both scale and time. When using a sampled signal with a digital filter the time quantisation of the output will be the same as that at the input to the filter, hence with the ‘algorithme à trous’ the spacing of the resultant wavelet coefficients in time is the same as that of the input sequence. Also implicit in this algorithm, as described, is the sampling of the scale axis by the dilation of the wavelet filter. To maintain the same number of non-zero coefficients at each scale being analysed the scale should double for each analysis so in this case (3.63) is modified so that $s = 2^p, p \in \mathbb{Z} - *$.

The search for wavelets for the analysis of discrete signals that offer a useful time-scale representation of the signal, are amenable to efficient computation and offer perfect reconstruction has led to the development of what is known as multiresolution analysis (MRA) theory. An MRA describes a signal as linear combinations of nested vector spaces of different resolutions. The finest resolution vector subspace in an MRA should contain all of the square integrable functions (this set includes all functions that describe audio signals). These functions have, by definition finite energy with a norm given by:

$$\| f \| \triangleq \left(\int_{-\infty}^{\infty} | f(t)|^2 \, dt \right)^{\frac{1}{2}} < \infty \qquad \text{(3.64)} \quad \text{where } f(t) \in \mathbf{L}^2(\mathbb{R}) .$$

For any continuous function, $f(t)$, that satisfies (61) and a set of vector subspaces $V_j, j \in \mathbb{Z}$, in an MRA the following relations apply for $j, k \in \mathbb{Z}$:

$$f(t) \in V_j \Leftrightarrow f(t - 2^j k) \in V_j \quad \text{(3.65)}$$

$$V_{j+1} \subset V_j \qquad \text{(3.66)}$$

$$V_j \to \mathbf{L}^2(\mathbb{R}) \text{ as } j \to -\infty \text{ and } V_j \to \{0\} \text{ as } j \to \infty \qquad \text{(3.67)}$$

$$f(t) \in V_j \Rightarrow D_2 f\left(\frac{t}{2}\right) \in V_{j+1} \quad \text{(3.68)}$$

These relations specify, in order, invariance to translation proportional to the scale 2^j, that any vector subspace is able to span the lower resolution spaces, that the coarsest resolution subspace contains only $\{0\}$ and the finest resolution subspace contains all possible functions, and that there is dilation (by a power of 2) invariance [Mallat, 1999]. For V_j a 'scaling' function, ϕ, is chosen such that its integer translations form an orthonormal basis for one of the subspaces. A basis is the smallest set of vectors that can span a vector space. In order to span a vector space a set of vectors must be capable of forming any possible vector in the subspace by linear combination. An orthonormal set of vectors is one which is orthogonal and has unit norm. Considering these definitions and (3.68) the scaling function can be described with the following equation, known as the 'dilation', 'two scale' or 'refinement' equation:

$$\frac{1}{\sqrt{2}} \phi\left(\frac{t}{2}\right) = \sum_{n=-\infty}^{\infty} h[n] \phi(t-n) \qquad \text{(3.69)}$$

where $h[n]$ can be viewed as a discrete filter which can be determined by:

$$h[n] = \left\langle \frac{1}{\sqrt{2}} \phi\left(\frac{t}{2}\right), \phi(t-n) \right\rangle \qquad \text{(3.70)} \quad \text{[Mallat, 1999].}$$

Each of the subspaces V_j provides an approximation to the function being analysed. In addition to these approximations a set of vector subspaces W_j which are complementary to V_j and orthogonal to V_{j+1} for any $j \in \mathbb{Z}$ can be defined:

$$V_{j-1} = V_j \oplus W_j \qquad (3.71)$$

$$V_j \perp W_j \qquad (3.72)$$

Therefore the subspace W_j contains those functions required, yet not available in V_j, in order to span V_{j-1}. These functions represent the difference in resolution between the two approximation subspaces V_j and V_{j-1}. This difference represents the additional detail provided at the higher resolution approximation. For this reason the subspace V_j is seen as that which provides the approximation to a function at level *j* and W_j provides the detail of that function at this level. For the subspaces W_j a function can be defined, as for the scaling function, such that:

$$\frac{1}{\sqrt{2}}\psi\left(\frac{t}{2}\right) = \sum_{n=-\infty}^{\infty} g[n]\phi(t-n) \qquad (3.73)$$

where:

$$g[n] = \left\langle \frac{1}{\sqrt{2}}\psi\left(\frac{t}{2}\right), \phi(t-n) \right\rangle \qquad (3.74)$$

This function is known as the wavelet function. In some literature, e.g. [Debnath, 2002], the wavelet function is known as the 'mother wavelet' and the scaling function is known as the 'father wavelet'. It can be shown that for an orthogonal MRA[13]:

$$\left|H(w)\right|^2 + \left|G(w)\right|^2 = \left|H(w)\right|^2 + \left|H(w+\pi)\right|^2 = 1 \qquad (3.75)$$ [Qian, 2002]

and, from taking the inverse Fourier transform:

[13] This is without loss of generality. The specific case in (3.75) is where the Fourier transform of the scaling function is 1 at DC $(\omega = 0)$ [Qian, 2002].

$$g[n] = (-1)^{1-n} h[1-n] \quad (72) \qquad\qquad (3.76) \text{ [Mallat, 1999]}$$

Therefore $g[n]$ and $h[n]$ are quadrature mirror filters (QMF). These filters can be used to find the approximation and detail coefficients for an MRA which completely describe a discrete signal at a given level of decomposition, from the approximation coefficients at the next level:

$$a_{j+1}[m] = \sum_{n=-\infty}^{\infty} h[n-2m] a_j[n] \qquad (3.77)$$

$$d_{j+1}[m] = \sum_{n=-\infty}^{\infty} g[n-2m] a_j[n] \qquad (3.78)$$

and where $a_j[m]$ and $d_j[m]$ are the approximation and detail coefficients at level j and at sample m. This leads to fast, recursive algorithms for computing the undecimated (‘à trous’) and decimated (critically sampled) wavelet transform. For the orthogonal wavelet transform the signal can be perfectly reconstructed by simply reversing the high and low pass filters in time:

$$a_j[p] = \sum_{n=-\infty}^{\infty} h[m-2n] a_{j+1}[n] + \sum_{n=-\infty}^{\infty} g[m-2n] d_{j+1}[n] \quad (3.79) \text{ [Mallat, 1999]}$$

The wavelet analysis algorithm described in chapter 5 combines the decimated and undecimated wavelet transforms. This allows a trade off between computational cost and redundancy/invariance.

3.8.2 The decimated wavelet transform

The decimated wavelet transform, when calculated using the type of recursive algorithms described by (3.77), (3.78) and (3.79), is known as the fast wavelet transform (FWT). It differs from the undecimated transform in that the filter dilation process is replaced by decimation (down-sampling) of the outputs of the filters. This decimation reduces the computational cost of the wavelet transform and the amount of data in the output. The output of the FWT is time-scale data with coefficients on a dyadic grid. At the lowest scale (highest octave) the sampling in the time domain is at its most dense but the sampling in the frequency domain at its least dense. For each increase (doubling) in scale the bandwidth of the equivalent band pass filter is halved. For a real wavelet analysing a real signal the size of the output can be identical to the size of the input.

In a digital audio system the initial sampled sequence that represents the original signal can be seen as the highest resolution approximation to that continuous signal in the MRA. As discussed in the previous chapter, provided an audio signal is sampled at a sufficient rate the 'detail' signal, which is the complement of the 'approximation' signal that the sampled version represents, only exists outside the frequency range of interest (the part of the spectrum that can be perceived by the human auditory system).

In order for the complete execution of such wavelet transforms to be possible on a computer system the wavelet should have compact support in time since without this there will be infinite coefficients or truncation at all scales. In order for there to be a finite number of non-zero filter coefficients the wavelet function ψ should have compact support. Considering (3.73) it can be seen that this requirement implies that the scaling function should also have compact support. Using (3.76) to derive either the low-pass or high-pass filter coefficients from the other implies that if one filter has a finite number of non-zero coefficients then so will the other. The earliest known wavelet and scaling function with compact support is the Haar wavelet [Qian, 2002][14]. The continuous Haar wavelet is defined as:

$$\psi(t)=\begin{cases} 1, & 0 \le t < \frac{1}{2} \\ -1, & \frac{1}{2} \le t < 1 \\ 0, & \text{elsewhere} \end{cases} \qquad (3.80)$$

and its scaling function is defined as

$$\phi(t)=\begin{cases} 1, & 0 \le t < 1 \\ 0, & \text{elsewhere} \end{cases} \qquad (3.81)$$

The low and high pass filters associated with these functions are given by the Hadamard matrix:

[14] The term wavelet was not used by Haar and was not adopted until much later on in the development of this field.

$$H_2 = \begin{vmatrix} 1 & 1 \\ -1 & 1 \end{vmatrix} \qquad (3.82)$$

The Haar wavelet offers an intuitive insight into the relationship between MRAs and wavelets since it performs piecewise constant approximation to the function being analysed and it is computationally cheap since it has time domain support of just two coefficients. However, the Haar wavelet is not a good choice for approximating smooth functions since it only has one vanishing moment. It has been demonstrated that for an orthogonal wavelet to have p vanishing moments that it should have a support size larger than or equal to $2p$-1 [Daubechies, 2002]. Orthogonal wavelets that have support $2p$-1 are known as Daubechies wavelets (they are often referred to as “db$x$” wavelets where x is the number of vanishing moments[15]). The db1 wavelet is in fact the Haar wavelet. These wavelets are designed so that the high pass filter yields a response that is as close to zero as possible for the given time support [Press et al, 1992]]. Having as many detail coefficients as close to zero as possible, with signal energy concentrated in just a few coefficients, makes a critically sampled wavelet transform amenable to data reduction since the transform itself does not add any redundancy to the data and a number of coefficients in the transform domain can be removed with a relatively small impact on the reconstructed data.

The Daubechies wavelets are not the only wavelets with compact support but they are popular since they offer a sparse representation of many commonly encountered functions for low support. Two other varieties of compactly supported wavelet developed by Daubechies are symmlets and coiflets. Symmlets are designed to be more symmetrical in the time domain by choosing roots of the polynomial in the frequency domain to have as close to linear phase as possible. Coiflets are designed in a similar fashion to Daubechies wavelets but are designed to have a specified number of vanishing moments for the scaling function as well as the wavelet function. Their support is $3p$-1 [Daubechies, 1992]. There are many other types of filter for the orthogonal FWT. The interested reader is directed to [Misiti et al, 2000] or [Mallat, 1999] for further information.

[15] In this context in the literature the support size corresponds to the support width, so a support size of 3 actually requires 4 non-zero filter coefficients, a support size of 4 requires 5 coefficients and so on.

3.8.3 Spline wavelets

What follows is a brief overview of splines and spline wavelets. Much of this section is based on [Unser, 1999] and [Unser et al, 1993] and for a more detailed treatment the reader is referred to these.

Where an MRA is invertible but the resynthesis filters $\overset{\circ}{h}[n]$ and $\overset{\circ}{g}[n]$ are not the time reverse of $h[n]$ and $g[n]$ then it is biorthogonal rather than orthogonal. In the biorthogonal case the two filters need not satisfy the power complementarity condition (3.75). As for Gabor atoms, relaxing the orthogonality requirement allows optimisation of the analysis or synthesis filters for a particular task. For example, B-spline wavelets can be used in a biorthogonal MRA, either as the synthesis or the analysis functions (although it is not the case that splines offer the only functions suitable for a biorthogonal MRA). A B-spline curve through a set of points consists of the linear combination of shifted B-spline basis curves of a given order (the order of the B-spline). A zeroth order spline curve is constructed from a series of constant functions at the height of each data point. A first order spline curve is constructed from a series of straight lines that join each data point. A second order spline curve is constructed from a series of quadratic functions that span three data points and so on with each ‘piece’ of the curve having its own weighting coefficient, meaning that a function can be described by:

$$f(x) = \sum_{k \in Z} c(k)\beta^{m}(x-k) \qquad (3.83)$$

where β^{m} is the B-spline basis curve of order m. Figure 3.9 shows first order (linear) and third order (cubic) spline approximations to a coarsely sampled sine function.

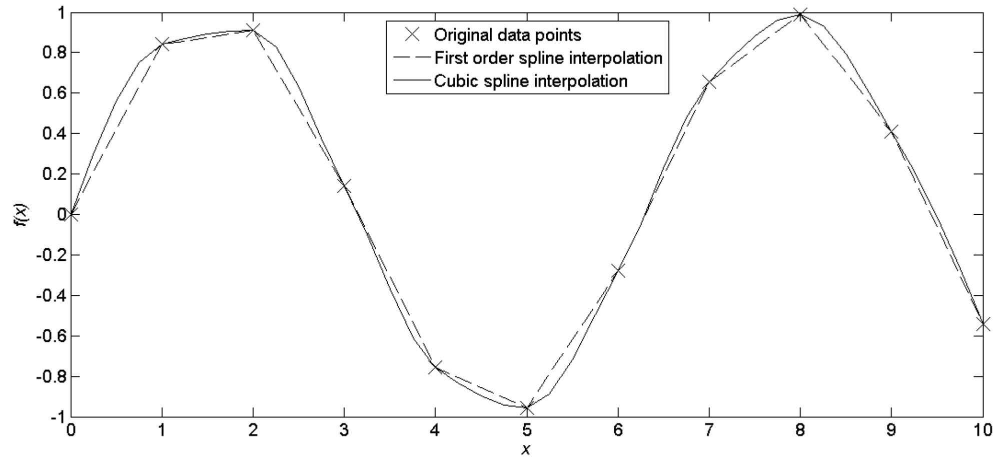


Figure 3.9: Spline approximations to a coarsely sampled sine function.

The B-spline can be used as a piecewise approximation to an arbitrary function and such a representation is useful since "each spline is unambiguously characterised by its sequence of B-spline coefficients $c(k)$, which has the convenient structure of a discrete signal, even though the underlying model is continuous" [Unser, 1999]. They are useful for this purpose since they are locally simple but can be combined to describe functions for which no other underlying model can be found. The zeroth order B-spline is given by

$$\beta^0(x) = \begin{cases} 1, & -\frac{1}{2} < x < \frac{1}{2} \\ \frac{1}{2}, & |x| = \frac{1}{2} \\ 0, & \text{elsewhere} \end{cases} \qquad (3.84)$$

which is similar to the continuous Haar wavelet. Higher order B-splines can be derived from the zeroth by convolution:

$$\beta^m(x) = \underbrace{\beta^m(x) * \beta^m(x) * \ldots * \beta^m(x)}_{m+1 \text{ times}} \qquad (3.85)$$

An equivalent equation to (3.85) is:

$$\beta^m(x) = \frac{1}{m!} \sum_{k=0}^{m+1} \binom{m+1}{k} (-1)^k \left(x - k + \frac{m+1}{k} \right)_+^m \qquad (3.86)$$

where

$$(x)_+^m = \begin{cases} x^m, x \ge 0 \\ 0, x < 0 \end{cases} \qquad (3.87)\ \text{[Unser, 1999]}$$

The constant (zeroth order) B-spline basis leads to a model which is not continuous (since it is piecewise constant). The first order basis offers a continuous (piecewise linear) underlying model but it is not smooth since the first derivative is not continuous. The second order (quadratic) basis is continuous and smooth but its rate of change of curvature (second derivative) changes in a piecewise constant fashion. The third order (cubic) basis exhibits the ‘minimum curvature property” since the second derivative is continuous and so for many applications the cubic B-spline is considered the most appropriate underlying continuous piecewise function.

As the order of the B-spline basis tends to infinity so the wavelet function tends to a modulated Gaussian function making such functions ideal for approximating the Morlet wavelet using a fast algorithm. The error in approximating the Morlet wavelet with a cubic B-spline is less than 3% and the product of its time and frequency variance is within 2% of the optimum limit imposed by the uncertainty principle (3.24 and 3.25) [Unser et al, 1992]. Since the B-spline basis is not orthogonal dual scaling and wavelet functions must be determined for the invertibility of its wavelet transform. These functions are known as the dual spline functions, or D-splines. A biorthogonal wavelet transform using B-spline functions for analysis uses D-spline functions for synthesis and *vice versa*. The low and high pass filter coefficients for analysis and synthesis with different types of spline wavelet are given by:

$$h[n] = (p[n])_{\uparrow 2} * u_2^m[n] * (p[n])^{-1} \qquad (3.88)$$

$$g[n] = \left(q[n]_{\uparrow 2}\right) * \left((-1)^m u_2^m[n]\right) * \left((-1)^m b^{2m+1}[n]\right) * \delta_{-1}[n] * \left(p[n]\right)^{-1} \qquad (3.89)^{16}$$

$$\overset{\circ}{h}[n] = \frac{1}{2}\left(\left(p[n] * b^{2m+1}[n]\right)^{-1}\right)_{\uparrow 2} * p[n] * b^{2m+1}[n] * u_2^m[n] \qquad (3.90)$$

$$\overset{\circ}{g}[n] = \frac{1}{2}\left(\left(q[n] * b^{2m+1}[n]\right)^{-1}\right)_{\uparrow 2} * p[n] * \left((-1)^m u_2^m[n]\right) * \delta_1[n] \qquad (3.91)$$

[16] This is corrected from equation (3.31) in [Unser et al, 1993]. As it appears in the paper it is inconsistent with equations (3.6) and (A.5).

where $\delta_r[n]$ is the unit impulse at sample *r*, $u_2^k[n]$ is the binomial kernel of order *k* given by:

$$u_2^m[n] = \begin{cases} \dfrac{1}{2^m}\begin{pmatrix} m+1 \\ n+\dfrac{m+1}{2} \end{pmatrix}, & |n| \le \dfrac{m+1}{2} \\ 0, \text{ otherwise} \end{cases} \qquad (3.92)$$

and $b^m[n]$ is the discrete B-spline sampled from the continuous *m*th order B-spline function at the integers:

$$b^m[n] = \beta^m(t)\big|_{t=m} \qquad (3.93)$$

$\left(x[n]\right)_{\uparrow 2}$ represents up-sampling by a factor of two by insertion of zeros and $\left(x[n]\right)_{\downarrow 2}$ represents down-sampling by a factor of two (decimation) by removal of odd-numbered samples. The filters $p[n]$ and $q[n]$ are determined by the type of spline wavelet. Where the B-spline wavelet and scaling functions are required these are the unit impulse at $n = 0$:

$$p[n] = q[n] = \delta_0 \qquad (3.94)$$

These filters for the *k*th order D-spline filters are given by:

$$p[n] = \left(b^{2m+1}[n]\right)^{-1} \qquad (3.95)$$

$$q[n] = \left(b^{2m+1}[n] * \left(\left(-1\right)^m b^{2m+1}[n] * b^{2m+1}[n]\right)_{\downarrow 2}\right)^{-1} \qquad (3.96)$$

The Battle-Lemarie [Mallat, 1999] orthogonal spline (referred to as O-spline) filters are given by:

$$p[n] = \left(b^{2m+1}[n]\right)^{-\frac{1}{2}} \qquad (3.97)$$

$$q[n] = \left(b^{2m+1}[n] * \left(\left(-1\right)^m b^{2m+1}[n] * b^{2m+1}[n]\right)_{\downarrow 2}\right)^{-\frac{1}{2}} \qquad (3.98)$$

A fourth set of B-spline filters can be used to calculate the cardinal spline (or C-spline) wavelet transform. The C-spline wavelet tends to the sinc function $\left(\frac{\sin(x)}{x}\right)$ as the order of the spline tends to infinity. The filters are given by:

$$p[n] = (b^m[n])^{-1} \qquad (3.99)$$

$$q[n] = \left(\left(b^m[n] * (-1)^m u_2^m[n] * (-1)^m b[n]^{2m+1}\right)_{\downarrow 2}\right)^{-1} \qquad (3.100)$$

A number of these solutions specify the inverse filter, defined as:

$$\left(x[n]\right)^{-1} * x[n] = \delta_0[n] \qquad (3.101)$$

The inverse filter, where it exists, can be found from the inverse DFT of the following:

$$\left(X[k]\right)_{inverse} = \frac{1}{X[k]} \qquad (3.102)$$

Where $X[k]$ is the DFT of the sequence $x[n]$. Clearly where $X[k]=0$ for any k the inverse does not exist. As has been intuitively expressed “a linear operator [such as a filter] cannot raise the dead – it only recovers 0 from 0” [Strang and Nguyen, 1996]. Also, where it does exist the inverse may not have a finite impulse response (FIR), even if sequence from which it derived is finite. As discussed earlier this is not desirable since this leads to wavelet and scaling filters without compact support. However provided the coefficients decay towards 0 as the distance from the middle coefficient of the filter increases then the infinite impulse response (IIR) can be truncated to provide an FIR filter which is an approximation to it. The approximation error is determined by the rate of decay and the number of coefficients in the truncated filter. It is also possible to use a cascade of FIR and IIR filters to compute the coefficients of filters without compact support. Where a filter does not have an inverse, because $X[k]=0$ for some *k*, then the original input to the filter can be recovered if the DFT of the input contains no energy for these values of *k*.

3.8.4 Complex wavelet transforms

Complex wavelet transforms for real input data can be implemented by performing two separate wavelet transforms on the same input sequence. One transform represents the real part of the complex transform and the other represents the imaginary part. The relationship between filters for the real and imaginary parts varies according to implementation. Often the difference between the filters for each transform is that underlying wavelet and scaling functions are shifted in phase by $\frac{\pi}{2}$. Alternatively the analytic version of the input signal can be found and the same filters applied to the real and imaginary parts of this signal. The analytic version of a real signal is complex and can be found by setting the negative part of the spectrum, which for a real signal is the complex conjugate of the positive part of the spectrum, to zero and multiplying the positive spectrum by two to preserve the signal energy:

$$F_{analytic}(\omega) = \begin{cases} 2F(\omega), & \omega>0 \\ F(\omega), & \omega=0 \\ 0, & \omega<0 \end{cases} \qquad (3.103)$$

A complex FWT of a real input has 100% redundancy and this redundancy can provide approximate shift invariance. So called 'dual tree' wavelets have been developed to offer the least shift variance within the FWT framework [Kingsbury, 2001]. Shift (or translation) invariance is a desirable property in time-frequency and time-scale analysis. When a signal representation is shift invariant shifts in the input signal produce a corresponding shift in, but not a modification of, the representation. The CWT and the STFT are shift-invariant transforms but the sampling of the translation parameter (decimation in time by a factor of two for each lower octave) produces a shift *variant* representation in the FWT [Mallatt, 1999]. The mechanism of this shift invariance can be viewed in either the time or frequency domains. In terms of frequency, decimation produces alias components within the analysis sub bands since no compactly supported filter can have perfect stop band rejection. In terms of time, the wavelet coefficients will vary depending on which samples are removed at each decimation stage which is directly affected by the position (the relative shift) of the input data [Bradley, 2003]. Thus the relative size of the coefficients in the wavelet domain will vary with relative shift even though energy is preserved across scales in the orthogonal case. These terms are cancelled at the reconstruction of the signal (provided there is *no modification* to the data between analysis and synthesis) but they are present in the data in the wavelet domain. The design of the filters for the dual-tree complex wavelet transform considers this problem in the frequency domain and attempts to minimise aliasing within each sub-band for a given FIR filter length and a redundancy of 100% [Kingsbury, 1999]. This approach to the design of complex wavelets is in contrast with the previously discussed approach of designing wavelet functions to be identical (or as similar as possible) but separated in phase by $\frac{\pi}{2}$ which assumes the same relationship between real and imaginary parts of the transform as for Fourier analysis.

3.8.5 Complex wavelets for audio

The discussion of wavelets presented in this chapter has covered three separate types of wavelet transform: the continuous, undecimated and decimated wavelet transforms. The last two types are suitable for the computer analysis of audio. As discussed the real valued decimated wavelet transform makes low demands on memory, requires relatively few computations and can offer sparse representations of data. However it does not describe the time-frequency plane in terms of components with phase and so it is not ideal for inferring descriptions of audio components. It is also shift variant meaning that numerical descriptions of a component varies with the position of the component in time. The undecimated wavelet transform is more expensive in terms of computation and memory since the length of the filters grows exponentially at each decomposition level (although they effectively remain the same length for the à trous implementation) and the number of components at each level remains the same since there is no decimation. However it is shift invariant and it offers a redundant representation of the data which makes it an attractive analysis tool. The decimated complex wavelet transform comes somewhere between the decimated and undecimated real wavelet transforms in terms of redundancy (100%), memory requirements and computational cost. It has been pointed out that little work has been published on the use of complex wavelets for audio processing [Wolfe and Godsill, 2003] and it is certainly the experience of this author that there is little published work in this area. As the authors point out:

> One reason for the success of real-valued wavelets to date has been their tendency to provide a sparse representation for many types of data-images being a primary example. However aside from a small number of papers over the last decade, audio applications of wavelets have seen relatively few major successes in comparison with traditional Fourier approaches … [the inherent redundancy of the dual-tree FWT] plays a role in ensuring that a complex wavelet transform exhibits a degree of translation invariance, a key property for tasks involving some degree of pattern recognition, such as auditory feature extraction. [Wolfe and Godsill, 2003]

However the claim that *no* published work, regarding processing with complex wavelets, predates this paper overlooks work published in 1988 regarding the application of the 'à trous' algorithm to audio analysis with complex wavelets [Kronland-Martinet, 1988] and perhaps more besides[17]. Chapter 5 of this thesis describes a complex wavelet analysis system based on the B-spline wavelets described in the previous section. The data from this transform is then used to infer values for the magnitude, centre frequency and bandwidth of underlying components.

3.8.6 Initialisation of the discrete wavelet transform

The orthogonal and biorthogonal wavelet transforms offer, by definition, perfect reconstruction of the original time series. Using MRA terminology the original time series is considered to be the projection of an underlying continuous function onto the sub-space V_0. The resolution of the subspace is determined by the sampling period T_s (the reciprocal of the sampling frequency). As $T_s \to 0$, n in $V_n \to -\infty$. For a given sampled time series T_s cannot be made smaller and so the approximation at scale 0 is taken as being the time series:

$$a_{j=0}(x[n]) \equiv x[n] \qquad (3.104)$$

Using (3.104) will allow analysis and perfect reconstruction but it may not provide intuitive analysis data since this is not the projection of the underlying continuous function on the V_0 sub-space. For a sampled sequence the underlying band limited continuous function can be found by convolving the samples with a sinc function:

$$f(t) = g(t) * \frac{\sin(\pi F_s t)}{\pi F_s t}$$ (3.105) [adapted from Smith, 2005] where:

$$g(t) = \begin{cases} x[n], t = nT_s \\ 0, \text{elsewhere} \end{cases} \qquad (3.106)$$

The sinc function, a sampled sequence and the sinc function convolved with the sequence are shown in the figures 3.10 and 3.11.

[17] It may be that the authors are referring specifically to dual-tree complex wavelets but this is not explicitly stated.

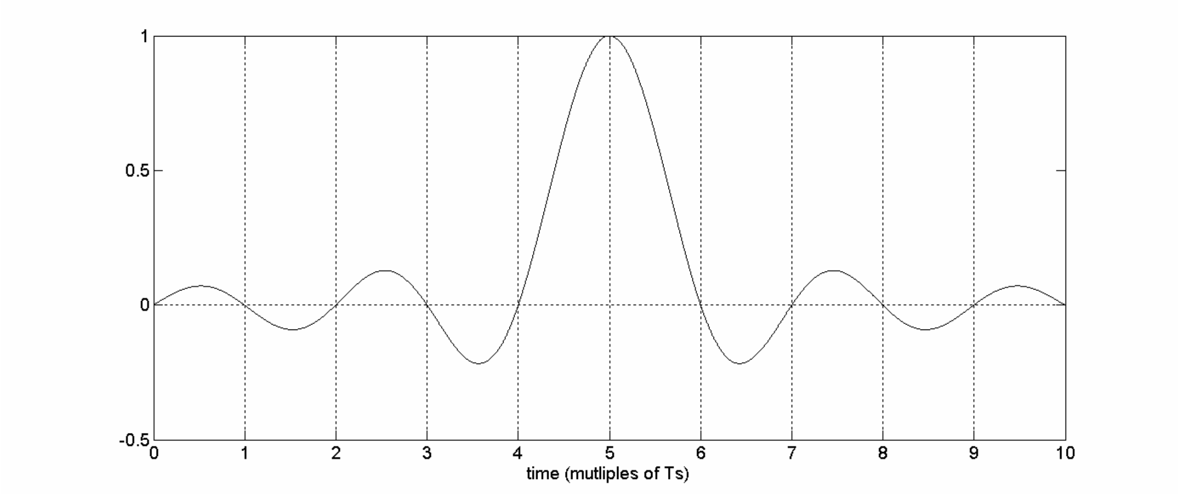


Figure 3.10: Sinc function.

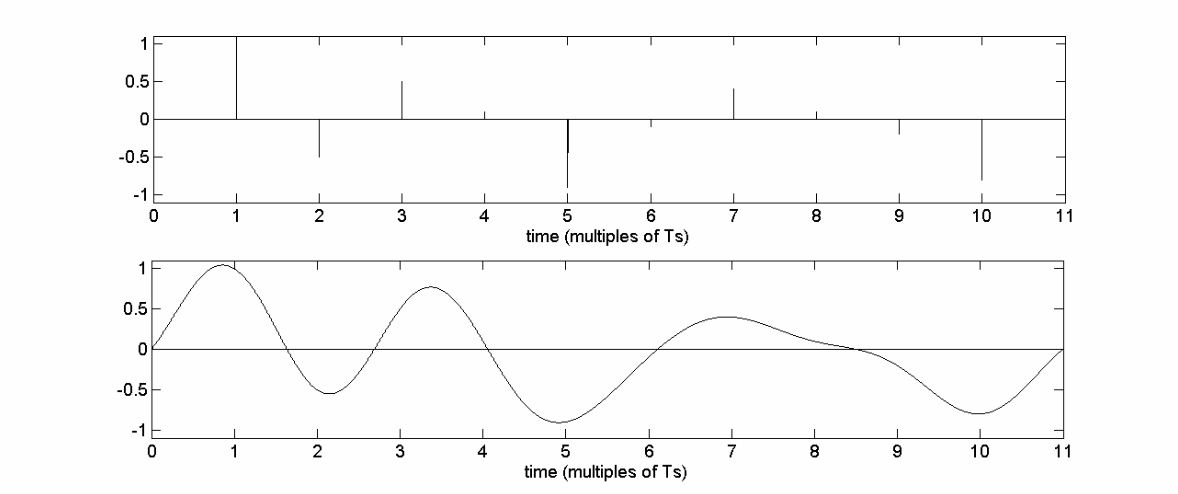


Figure 3.11: Convolution of a sequence of pulses with a sinc function.

Since the result of convolution of a discrete series with the sinc function represents the ideal underlying band-limited continuous function (which in an audio processing context, would be the continuous signal after the anti-aliasing filter has been applied) then projection of this signal onto V_0 is given by the convolution of the time series with a filter (α in the equation below). This filter is the inner product of the sinc function and the dual scaling function:

$$a_{j=0,k}\left(x[n]\right)=\sum_k x[n]\alpha_{k-n} \qquad (3.107) \qquad \text{where}$$

$$\alpha_p=\left\langle \text{sinc}, \overset{\circ}{\phi}_{j=0,-p}\right\rangle = \int_{-\infty}^{\infty} \text{sinc}(t)\overset{\circ}{\phi}(t-p)dt \qquad (3.108)\ \text{[Abry and Flandrin, 1994].}$$

As discussed in the previous section the cardinal (‘C’) spline tends to the sinc function as its order tends to infinity. This similarity between splines and the sinc function explains the application of both in interpolation and sample rate conversion. They are both smoothing functions which vanish at all integers other than the origin. In fact compactly supported (‘B’) splines are attractive in this regard since they offer interpolation at low computational cost:

> This is precisely why splines are so much more computationally efficient than the traditional sinc-based approach. Because sinc(x) decays like $\frac{1}{|x|}$ computing a signal value at a particular non-integer location with an error of less than 1% [-40 dB] will require of the order of 100 operations in each direction, while B-splines provide an exact computation with just a few non-zero terms ($n+1$ to be precise). [Unser, 1999].

This similarity between splines and the sinc function suggests the use of the B-spline as the initialisation filter when computing the wavelet transform with such functions. This gives the V_0 projection as:

$$a_{j=0}(x[n]) = p * b^n * x[n] \qquad (3.109)$$

where p is the filter specific to the type of spline as discussed in section 3.8.3. At reconstruction the final approximation coefficients must be convolved with the inverse filter to obtain the original sampled sequence.

$$x[n] = (p[n] * b^k[n])^{-1} * a_{j=1} \qquad (3.110)$$

As stated in a previous section this initialisation better conditions the input data for analysis. As an example an impulse in the original time series should produce the wavelet shape in the detail sequences at increasing dilations for increasing detail level. This is the case when the initialisation has been performed but is often not the case without it. In the same way if the original time series consists of the shape of the dilated wavelet at a given level then the wavelet analysis should produce an impulse in the detail level corresponding to this dilation, at other detail levels all of the coefficients should be zero. Again this is the case when the initialisation is performed but without it energy may be spread into other detail levels. In chapter 5 the importance of proper initialisation for estimating frequency is discussed.

3.8.7 The frequency-splitting trick and wavelet packets

The decimated wavelet transform described in section 3.8.2 uses a recursive ‘filter and decimate by two’ algorithm to find the detail coefficients at all scales as well as the approximation coefficients at the highest scale (which is dependent upon the number of decomposition levels in the analysis). Firstly the discrete signal itself (or the initialised wavelet sequence) is high and low pass filtered, splitting it into two bands with the output of each filter decimated. The high frequency band contains the signal details at the first scale and the low frequency band contains the signal approximations at this decomposition level. Next the approximation coefficients are split into bands using the same filters and decimated giving the details and approximations at this decomposition level. Then the approximations at this decomposition level are split and decimated to give the two bands at the next decomposition level and so on.

In the process just described it is the lowest band out of the two that is split each time and, since the output of the filters is decimated by two, the number of coefficients at each decomposition level is halved. The splitting of the lowest band at each decomposition level is what yields the ‘constant-Q’ property of wavelet analysis. This is often a desirable property. However, it need not be the lowest band that is split. Similar to (3.71) we also have:

$$W_{j-1} = V_j \oplus W_j \qquad (3.111)$$

which implies that the highest band may also be split. In fact both bands may be split at each decomposition level to give what is known as the *full binary tree* [Mallat, 1999]. For orthonormal filters any route from the bottom of the tree to the end of one of the branches, for any given level of decomposition, forms an orthonormal basis and so the entire tree forms a library of bases, known as wavelet packets, with which a signal can be analysed. Wavelet packets have been described as “particular linear combinations or superpositions of wavelets” [Wickerhauser, 1994]. If the full binary tree is known then the ‘best basis’ for describing a signal with a particular wavelet can be derived from this tree by assigning a cost function to each route from the base to the top of the tree. For example a cost function, C_{basis}, used for denoising of audio signals is the Shannon second order entropy:

$$C_{basis} = -\sum_{n=1}^{N} \alpha_n^2 \log_2 \alpha_n^2 \qquad (3.112)$$

where α_n is the series of coefficients for the basis atoms ω_n^2 which comprise the signal:

$$f(t) = \sum_{n=1}^{N} \alpha_n \omega_n(t) \qquad (3.113) \qquad \text{[Berger et al, 1994].}$$

The cost function attempts to identify the sparsest representation of the signal offered by the basis library with a small number of large coefficients (that are assumed to represent the coherent part of the signal). These are complemented by a large number of small coefficients that are assumed to represent the unwanted, noisy part of the signal which can be reduced in level (soft thresholding) or set to zero (hard thresholding) to reduce the level of this ‘assumed noise’ component of the re-synthesized signal. Such thresholding can be carried out on data derived from fixed rather than best bases such as the STFT or the decimated or undecimated wavelet transforms. The band splitting technique of [Daubechies, 1992] that leads to the full binary tree is used to estimate the spectral width of sound components in chapter 5 of this thesis.

3.9 Time-frequency energy distributions

So far atomic decompositions of signals, specifically time series, have been considered. These attempt to describe how a one dimensional series’ spectral content changes over time by finding the inner product of the time series with atoms of differing frequencies (or scales) and with differing shifts in time. The exception to this is the DFT which does not shift atoms and so provides frequency, rather than time-frequency, analysis of time series.

A second type of time-frequency analysis method attempts to describe how the energy in a signal varies as two variables, time and frequency, vary [Auger et al, 1994]. The first such distribution to be described was the Wigner-Ville distribution (WVD). This is defined as:

$$WVD_{t,\omega} = \int_{-\infty}^{\infty} f\left(t + \frac{\tau}{2}\right) f^*\left(t - \frac{\tau}{2}\right) e^{-j\tau\omega} d\tau \qquad (3.114) \text{ or}$$

$$WVD_{t,\omega} = \int_{-\infty}^{\infty} F\left(\omega - \frac{\theta}{2}\right) F^*\left(\omega + \frac{\theta}{2}\right) e^{-j\theta t} d\theta \qquad (3.115)$$

where $f(t)$ and $F(\omega)$ are the signal and its spectrum respectively [Cohen, 1994]. The WVD is similar to the Fourier transform of the autocorrelation function, where the autocorrelation is given by:

$$A(t)=\int_{-\infty}^{\infty} f*(\tau)f(\tau+t)d\tau \qquad (3.116) \qquad \text{[Cohen, 1994]}$$

It is classified as a bi-linear (or quadratic, as opposed to linear) distribution because the signal appears twice in the integral, albeit at different points in time (or frequency). This aspect of the distribution, the product of the signal at two different points (except when τ, $\theta=0$), imparts two important properties of the distribution, one desirable the other undesirable: good time-frequency resolution compared to the STFT and the presence of cross, or interference terms. The first property offers high resolution of mono-component signals, at any time instant the conditional mean frequency of the WVD (over all frequencies) is equal to the first derivative of the phase of the signal which gives the instantaneous frequency[18]. Also, at any frequency the conditional mean frequency of the WVD (over all time) is equal to the group delay of the signal at that frequency [Qian, 2002]. The interference terms appear in the WVD when there is more than one sinusoidal component in the signal, they are a direct result of the quadratic nature of the distribution. Considering a two component signal:

$$f(t)=u(t)+v(t) \qquad (3.117)$$

$$\begin{aligned} WVD_{t,\omega}(f(t)) &= \int_{-\infty}^{\infty} f\left(t+\frac{\tau}{2}\right)f^*\left(t-\frac{\tau}{2}\right)e^{-j\tau\omega}d\tau \\ &= \int_{-\infty}^{\infty}\left(u\left(t+\frac{\tau}{2}\right)+v\left(t+\frac{\tau}{2}\right)\right)\left(u^*\left(t-\frac{\tau}{2}\right)+v^*\left(t-\frac{\tau}{2}\right)\right)e^{-j\tau\omega}d\tau \qquad (3.118) \\ &= WVD_{t,w}(u(t))+WVD_{t,w}(v(t))+WVD_{t,w}(u(t),v(t))+WVD_{t,w}(v(t),u(t)) \end{aligned}$$

where:

$$WVD_{t,w}(u(t),v(t))=\int_{-\infty}^{\infty} u\left(t+\frac{\tau}{2}\right)v^*\left(t-\frac{\tau}{2}\right)e^{-j\tau\omega}d\tau \qquad (3.119) \text{ and}$$

$$WVD_{t,w}(v(t),u(t))=\int_{-\infty}^{\infty} v\left(t+\frac{\tau}{2}\right)u^*\left(t-\frac{\tau}{2}\right)e^{-j\tau\omega}d\tau \qquad (3.120)$$

[18] The conditional mean of a signal or function is the mean under a given condition. In this context the conditional mean frequency is the mean frequency at a given instant in time and the conditional mean time is the mean time at a given frequency.

are the cross-Wigner distributions of the two signals. The cross-Wigner distribution can be complex but (117) and (118) are equal to the conjugates of each other and so are both real. This gives, from (116):

$$WVD_{t,\varpi}\left(f(t)\right)=WVD_{t,w}\left(u(t)\right)+WVD_{t,w}\left(v(t)\right)+2\Re\left(WVD_{t,w}\left(u(t),v(t)\right)\right) \quad (3.121)$$

To differentiate it from the cross-Wigner distribution the single signal distribution described in (3.114) and (3.115) is sometimes termed the auto-Wigner distribution. Since the auto-Wigner distribution and its conjugate are also equal the auto-Wigner distribution is real. This is an important difference between this analysis method and the STFT since it implies that it contains no phase information.

For a dual component signal, where each component is a sinusoid of a fixed frequency and each component has equal amplitude, the two auto terms will appear in the WVD with equal amplitude and at the respective frequencies of the sinusoids. The cross term will appear midway between the two auto terms in frequency and with twice the amplitude. It will also exhibit bipolar oscillation at a rate proportional to the distance between the two auto terms. It is clear that if the signal contains a third component which occurs at this same midpoint frequency then it will be obscured by the behaviour of the cross term. This is in addition to the fact that a third component will generate even more cross terms. The presence of oscillating cross terms which have zero average energy can also cause the distribution to be negative in places which is counterintuitive since negative energy does not have a straightforward physical manifestation. Considering a general *N*-component signal the number of cross terms will be $\left(N^2-N\right)\!\big/_2$ which, for $N>3$, will generate more cross terms in the distribution than auto terms [Qian, 2002]. Despite the fact that they can be reduced by applying the WVD to the analytic signal, since this avoids all cross terms associated with negative frequencies, the abundance and magnitude of these oscillating interference terms makes the useful application of the WVD to the analysis of multi-component signals very limited. This has led to the development of the smoothed WVD (SWVD) which applies low pass filtering to the distribution in both the time and frequency directions:

$$SWVD_{t,w}\left(f(t)\right)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\phi_{x,y}WVD_{t-x,\varpi-y}\left(f(t)\right)dxdy \qquad (3.122)$$

where: $\phi_{t,\omega} = e^{-\alpha t^2 - \beta\omega^2}$, $\alpha,\beta>0$ (3.123) [Qian, p.163]

Since the parameters α and β control the spreading of the Gaussian function in each direction, the amount of smoothing can be controlled. Since the cross terms oscillate and have zero average energy the low pass filtering can reduce the magnitude of these terms. However this filtering also reduces the resolution of the wanted auto terms, with sudden changes of amplitude in time and frequency also being smoothed. When $\alpha\beta \geq 1$ the distribution can no longer be negative at any point and when $\alpha\beta = 1$ then the SWVD is identical to the square of the STFT [Qian, 2002]. The square of the STFT is known as the spectrogram and this is also an energy distribution since the energy of a signal is either the square of its magnitude in the time or frequency domains. As for the WVD, the spectrogram is real valued and so has no phase (although the phase spectrum associated with the Fourier spectrum from which it is derived can be known). Therefore the spectrogram is a smoothed WVD. In fact "all time frequency representations can be obtained from:

$$C_{t,\omega} = \frac{1}{4\pi^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f^*\left(u-\frac{\tau}{2}\right) f\left(u+\frac{\tau}{2}\right)\phi(\theta,\tau)e^{-j(\theta t+\tau\omega-\theta u)}du\,d\tau\,d\theta \quad (3.124)$$

where $\phi(\theta,\tau)$ is a two dimensional function called the kernel … [which] determines the distribution and its properties" [Cohen, 1994]. For the WVD this kernel is 1. For the STFT it is:

$$\int_{-\infty}^{\infty} h^*\left(u-\frac{\tau}{2}\right) h\left(u+\frac{\tau}{2}\right)e^{-j\theta u}du \quad (3.125)$$

There are numerous kernels which have been specified which offer time-frequency distributions with different properties such as the Choi-Williams, Zhao-Atlas-Marks ('cone') and modal distributions [Cohen, 1994]. The form presented in (122) is known as 'Cohen's class' of time frequency distributions. Examples of musical applications of distributions derived from this class can be found in [Pielemeier et al, 1996].

Practicable computation of the discrete time WVD requires the using of a finite-length time domain window, as for the STFT and referred to as $\gamma[n]$ in the following equation, to remove the need for an infinite summation. The use of such a finite length window reduces the frequency resolution, but not the time resolution, giving what is referred to as the pseudo WVD (PWVD) which is given by:

$$PWVD_{n,\omega}\left(x[n]\right)=2\sum_{m=-\infty}^{\infty}\gamma[m]x[n+m]x^{*}[m-n]e^{-j2\omega m} \quad (3.126)\ [\text{Qian, 2002}]$$

The multiplication by two in the power term of the exponential is introduced to avoid $\frac{m}{2}$ terms in the bilinear part of the equation. This makes the period of the distribution πF_S rather than $2\pi F_S$ so if the bandwidth of the signal is greater than $\frac{F_s}{4}$ then aliasing will occur. This means that for a signal sampled near to the Nyquist limit the sample rate needs to be doubled which can be done by inserting zeros between existing samples and performing interpolation filtering. If the window is real and symmetric and it has length $2L-1$ and $u[n]$ is the real valued $x[n]$ up-sampled by a factor of two then the discrete pseudo WVD (DPWVD) is given by:

$$DPWVD_{n,\omega}\left(x[n]\right)=4\Re\left\{\sum_{m=0}^{2L-1}\gamma[m]u[2n+m]u[2n-m]e^{-j\frac{4\pi kn}{2L}}\right\}-2\gamma[0]\left(u[2n]\right)^{2} \quad (3.127)$$

[Qian, 2002].

In this form the DPWVD can be calculated using the FFT. Initial work undertaken for this thesis investigated whether the autocorrelation inherent in the WVD could be used, when compared with the spectrogram, to determine the sinusoidality of the component [Wells and Murphy, 2002]. Whilst promising results were obtained the smoothing required made such an approach unsuitable to real-time operation and it was subsequently abandoned in favour of the methods described in the following chapter.

3.10 Estimating the instantaneous frequency of signal components

> Instantaneous frequency is one of the most intuitive concepts, since we are surrounded by light of changing colour, by sounds of varying pitch, and by many other phenomena whose periodicity changes. The exact mathematical description and understanding of the concept of changing frequency is far from obvious and it is fair to say that it is not a settled question. [Cohen, 1994]

As already stated there is little musical value in a deterministic or stochastic process that is completely stationary throughout a sound, or a piece of music. If the parameters of a sound production process change over time then those parameters will be different at different instants. As has been discussed, there are no analysis methods that are capable of simultaneously giving arbitrary resolution in time and frequency. This means that, aside from the continuous WVD for a single component signal, the instantaneous frequency of a component cannot be precisely known from the output of any one of the algorithms discussed thus far. For example an FFT of a 1024 sample sequence, sampled at 44.1 kHz, has 513 useful analysis bins spanning the frequency range from 0 Hz to 22.05 kHz. This means that if there is a peak in a bin due to a stationary sinusoid, the frequency of that component lies in a range covering 43 Hz. In order to reduce this range further analysis is required. An example of a method for estimating the frequency of a sinusoidal component from STFT data was discussed in section 3.6. This section outlines some additional methods. For discrete analysis, if the spectrum is sampled at every sample (i.e. the hop size is one), then an estimate for the instantaneous frequency is provided for each sample. Where the hop size is greater than one then instantaneous frequency for intermediate samples must be interpolated, either using the nearest estimates on either side of the current sample or a method such as PDA (discussed in section 3.6.5).

Two approaches to estimating the frequency of sinusoids using magnitude data are those of parabolic and triangular interpolation. These rely on knowledge of the magnitude spectrum of the window at and around the peak bin to determine the precise location of a spectral peak between bins. Parabolic interpolation takes advantage of the fact that the magnitude response of most analysis windows when expressed in decibels is close in shape to that of a parabola. The following equation is used to obtain a frequency estimate using this method. Figure 3.12 illustrates this.

$$\omega_{sinusoid} = B\left(n + \frac{1}{2}\frac{M_{n-1} - M_{n+1}}{M_{n-1} - 2M_n + M_{n+1}}\right) \qquad (3.128)$$

Where B is the bin width in radians, n is the peak bin and M is the magnitude of a bin expressed in dB.

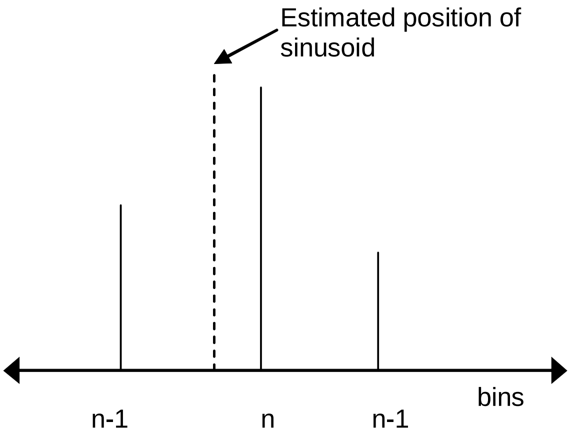


Figure 3.12: Parabolic interpolation.

Using an example DFT of 1024 frames, 44.1 kHz and a Hann window to analyse a 1 kHz sinusoid an estimate of 1000.6 kHz is obtained with this method. The accuracy can be improved by zero-padding the DFT and by using a specially designed window whose time domain function is the inverse transform of a function whose main lobe shape is as close as possible to that of a parabola (however this may adversely affect other aspects of window performance such as main and side lobe properties and the ability of overlapping windows to sum to 1).

The triangle algorithm is named after the shape of the main lobe of the window function in the frequency domain, although this is when the window is plotted with a linear rather than logarithmic magnitude scale. After a peak has been identified, two straight lines are drawn through the bin magnitudes and the frequency estimate is taken as the point at which these two lines (which form the two opposing slopes of the triangle) intersect. The slope of the lines is determined by calculating the best fit with the least squared error [Keiler and Marchand, 2002].

Another method which uses DFT magnitudes is the derivative algorithm [Desainte-Catherine and Marchand, 2000]. However this requires the computation of two DFTs for frequency estimation – one DFT is of the sampled signal, as for the other methods discussed so far, the second is of the derivative of the signal. For a sampled signal the closest approximation to the derivative of the signal is the first order difference:

$$y[n] = F_s\left(x[n] - x[n-1]\right) \qquad (3.129)$$

where $x[n]$is the sampled signal and $y[n]$ is the approximation of the derivative. A preliminary estimate of the frequency is then obtained from:

$$p = \frac{\Delta M_{peak}}{M_{peak}} \qquad (3.130)$$

where ΔM is the magnitude of the DFT of the difference data and M is the magnitude of the DFT of the original sampled data. Equation (3.129) is effectively a high pass filtering operation whose frequency dependent gain can be calculated. In order to match the gain of this filter to that of first order differentiation of the continuous signal the following scaling operation must be performed in order to obtain an improved estimate of the frequency .

$$f_{peak} = \left(\frac{F_S}{\pi}\right)\arcsin\left(\frac{p}{2F_S}\right) \qquad (3.131)$$

This method takes account of phase (even though the phase from both DFTs is not used) since the difference data actually forms an overlapping frame with the original data:

Data set for single frame of DFT: $x[0]$…………….$x[n$-1]

Next frame: $x[n]$………….$x[2n$–1]

Data set for single frame of difference DFT is calculated from: $x[$-1]………$x[n$-1]

Next frame: $x[n$-1]…………$x[2n$-1]

In fact, considering the time-shifting property of the DFT, this method is equivalent to the phase difference method described in section 3.6 with a hop size (distance between successive analysis frames) of 1 [Hainsworth and Macleod, 2003]. If the derivative method is employed with a hop size greater than 1 the mean instantaneous frequency is not evenly sampled since it is measured between two adjacent samples. For the phase difference method the frequency estimate is averaged over the hop distance from one frame to the next giving a frequency estimate across the whole length of an analysis hop.

One final method discussed here for estimation is that of frequency reassignment [Auger and Flandrin, 1995]. The general method of reassignment, as well as estimating frequency deviations from the centre of analysis bins, can also be applied to the position in time of spectral data. Time reassignment provides estimates of deviations from the centre of analysis frames. Reassignment frees the time-frequency representation from the grid structure imposed by the frame length and the hop size of the STFT. Once an analysis window has been chosen, two further windows are calculated – one that is ramped in the frequency domain (for frequency reassignment) and one that is ramped in time (for time reassignment). The frequency domain window can be calculated in the time domain by calculating the first order difference of the original window (as we do for the actual signal with the derivative method discussed previously). A 1024 point Hann window and its time and frequency ramped versions are shown in figure 3.13.

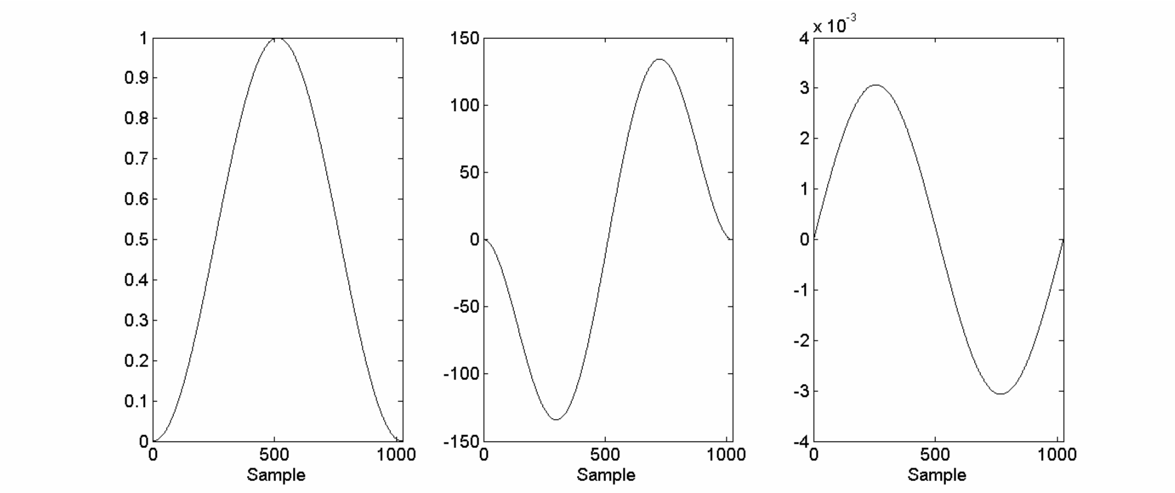


Figure 3.13: Hann window (left), its time ramped version (middle) and its frequency ramped version (right).

The estimate of frequency deviation (in radians) from the centre of an analysis bin is given by:

$$-B\Im\left\{\frac{DFT_{\text{frequency ramped window}}}{DFT_{\text{standard window}}}\right\} \qquad (3.132)$$

where *B* is the bin width (in radians) and *DFT* represents the complex value obtained for that bin by the DFT. The estimate of time deviation (in seconds) from the centre of an analysis frame is given by:

$$-\frac{1}{F_s}\Re\left\{\frac{DFT_{\text{time ramped window}}}{DFT_{\text{standard window}}}\right\} \qquad (3.133)$$

where F_s is the sample rate of the signal.

A recent study has demonstrated that equivalent ‘amplitude’ reassignment (subsequently referred to here as magnitude reassignment for consistency) can be used in place of the ‘phase’ reassignment where the frequency deviation and time deviation estimates are given by, respectively:

$$-B\left(\frac{F_s}{2\pi}\right)^2 \Im\left\{\frac{DFT_{\text{time ramped window}}}{DFT_{\text{standard window}}}\right\} \qquad (3.134)$$

$$-\frac{F_s}{4\pi^2}\Re\left\{\frac{DFT_{\text{frequency ramped window}}}{DFT_{\text{standard window}}}\right\} \qquad (3.135)$$

These two estimates will only be identical for the continuous STFT where a Gaussian window is used. For other window types noisy components will give rise to different estimates implying that comparison of the two sets of estimates may be useful in sinusoidal discrimination [Hainsworth and Macleod, 2003b]. This is investigated in the following chapter where novel methods for deriving estimates for ΔA and Δf and determining sinusoidality from reassignment data are described.

3.11 Amplitude correction

In section 3.6.5 it was seen how the window shape in the frequency domain affects the magnitude of the measured DFT spectrum for both stationary and non-stationary sinusoids and how this magnitude varies with distance from the centre of an analysis bin (see figure 3.5). This means that amplitude estimates for the underlying sinusoidal function will be incorrect unless the frequency of the sinusoid is at the centre of the bin from which the magnitude is measured. Knowledge of the window shape in the frequency domain and the deviation from the bin centre can be used to correct this error. It is straightforward to extend the parabolic interpolation discussed earlier to estimate the position of the parabolic peak on the magnitude as well as the frequency axis by the equation

$$Amp_{sinusoid} = M_2 - \frac{1}{8}\frac{(M_1 - M_3)^2}{(M_1 - 2M_2 + M_3)} \qquad (3.136)$$

This gives the amplitude of the sinusoid in dB relative to the amplitude of the peak bin. Alternatively, rather than taking the main lobe shape as being a parabola when its magnitude response is expressed in dB the power spectrum of the window can be calculated, by (3.29) or (3.31) for example and the amplitude can be corrected from this. However, a drawback of both of these methods is that they assume that the sinusoid is stationary. Amplitude estimation for non-stationary sinusoids is considered in chapter 4.

3.12 Summary

A wide range of sound analysis and modelling techniques have been surveyed as preparation for the following three chapters which describe novel work undertaken for this thesis. These chapters focus on sinusoidal identification and description using reassignment data, complex wavelet analysis and the development of a real-time spectral modelling system respectively. Whilst the coverage of the existing literature has been broad and extensive it is by no means exhaustive and represents a small fraction of all of the published work into analysis, processing and modelling of audio signals. For more information on the ideas and techniques presented in this chapter the reader is directed to the references given herein and listed in full at the end of the thesis.

4 REAL-TIME IDENTIFICATION AND DESCRIPTION OF STATIONARY AND NON-STATIONARY SINUSOIDS

4.1 Introduction

The spectral model adopted in this thesis distinguishes between stable sinusoids and other types of signal component. A stable sinusoid is defined here as a signal component that can be synthesized with minimal error using a single sinusoidal function with stationary or non-stationary amplitude (monotonic exponential) and frequency (monotonic linear) over the duration of at least one analysis frame. The analysis/transformation/synthesis algorithms that this thesis proposes are designed to be used as real-time processes. As a result of the uncertainty principle more than one sample of a signal must be acquired before a useful frequency analysis can be carried out (the DFT of a single sample is identical to the sample itself). There will therefore be latency between the input and output of such a system, giving a delayed output in a causal system[1]. In such a situation real-time clearly cannot mean instantaneous. Definitions of real-time vary in the literature but the definition adopted for this thesis is:

1. Quasi instantaneous: as close to instantaneous as is allowed by the frame size (window length) of the analysis algorithm.

2. Frame by frame: this is implied by the previous condition. Only the current and/or previous frames may be used in the model. Waiting for future frames is not possible.

3. Real-time execution: the execution time of the algorithm must be shorter than the time taken to replay the data it analyses or produces. Where there is redundancy in the analysis (e.g. due to the use of frame overlapping) this will increase the execution time but the real-time execution limit must not be exceeded.

This chapter describes the development and evaluation of an algorithm for identifying sinusoids (i.e. discriminating between sinusoids and non-sinusoids according to the definition given above) and describing them (i.e. deriving the parameters that can be used for their accurate resynthesis). Since no prior knowledge of the the spectrum is assumed and the system is intended to identify and describe individual sinusoidal components Fourier analysis

[1] As discussed in chapter 2 a DAW system can be seen as anti-causal on playback of data that it has already acquired. In fact many DAW systems compensate for processing delays by ‘looking ahead’ of the audio that is currently being played back to give the impression of instantaneous processing.

is used since it provides uniform sampling of frequency. The constant-Q approach, such as that discussed in section 3.7, does not provide this uniform sampling. Also, in order to provide the same number of divisions of the frequency axis, a much longer window length is required at the lowest frequency to accomodate the geometric spacing of those divisions. This mitigates against its use in a quasi real-time system.

4.2 Overview of sinusoidal identification

As seen in section 3.5.3 the DFT of a windowed stationary sinusoid is the DFT of the window function shifted from the zeroth analysis bin to the bin within which the sinusoid resides. Windows commonly used with the DFT for spectral analysis are characterised by a peak in the magnitude spectrum at the zeroth bin [Nuttall, 1981] and so a stationary sinusoid will be characterised by this peak but shifted in frequency. This is illustrated by figure 4.1 which shows the magnitude spectrum for a stationary sinusoid at 1 kHz at 44.1 kHz (all following examples assume this sample rate). Shown in the figure 4.2 is the spectrum for a highly non-stationary sinusoid demonstrating that, whilst the spectrum is not the same, it is still characterised by a peak, albeit in a different bin. For all figures in this chapter, unless otherwise stated, the frame length is 1025 samples zero-padded to 8192 samples for FFT analysis.

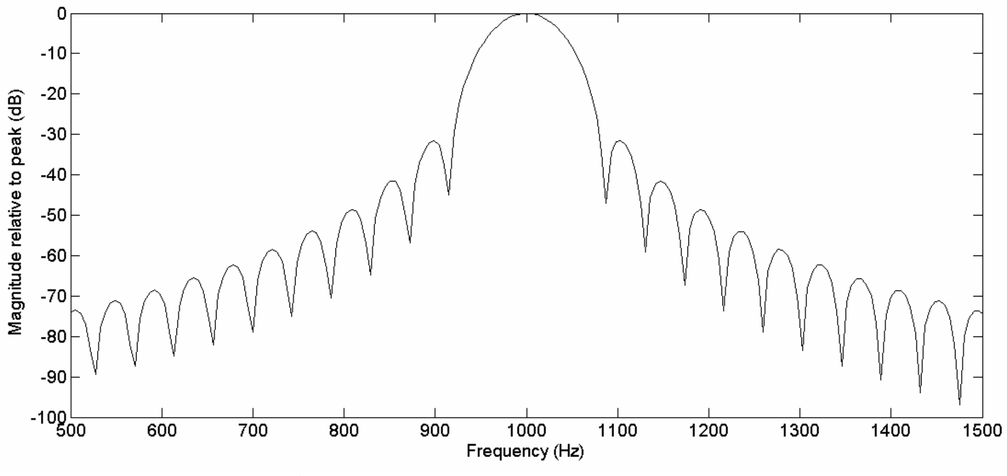


Figure 4.1: DFT magnitude of a stationary 1000 kHz Hann windowed sinusoid.

This suggests that the search for sinusoids should begin with the identification of local maxima in the magnitude spectrum. The simplest definition of a local maximum is a bin which has a higher magnitude than either of its two neighbours. Since the phase of a stationary sinusoid is constant across the main lobe for odd length zero-phase windows an alternative could be to search the phase spectrum for areas of flat phase. However for non-stationary sinusoids the phase is not constant across the main lobe so this is not a reliable method. Also, where the amplitude is, or is close to, stationary the position of the peak in the magnitude spectrum will indicate the centre of the range within which the actual frequency of the sinusoid lies. The sinusoidal components that are searched for in this model are of the form:

$$\text{sinusoid} = A(t)\sin\left(\int_0^t 2\pi f(\tau)d\tau + \phi\right) \qquad (4.1)$$

where $A(t)$ and $f(t)$ are linear and exponential functions respectively, as in [Masri, 1996]. An exponential function in this context is defined as $f(x) = ab^x$ and a linear function as $f(x) = cx + d$ where $a, b, c,$ and $d \in \mathbb{R}$ are constants. The difference between the lowest and highest value of $A(t)$ in a single frame is denoted ΔA and its mean $\overline{A}$. Likewise the difference and mean of $f(t)$ are denoted Δf and $\overline{f}$. It can be seen from (4.1) that when $A(t)$ is not constant, or close to constant, the average, or centre, value of $f(t)$ will not reside in the peak bin of the magnitude spectrum since $A(t)$ is not symmetrical. This can be seen in figure 4.2.

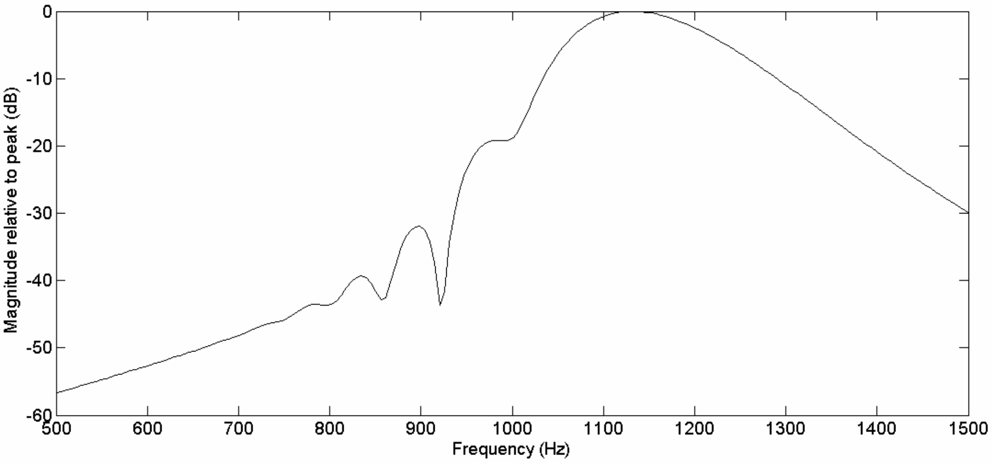


Figure 4.2: DFT magnitude of a non-stationary Hann windowed sinusoid. The amplitude of the sinusoid increases exponentially by 96 dB during the frame. The frequency increases linearly from 750 to 1250 Hz.

Clearly the centre value of $f(t)$ cannot be assumed to lie within the bin of the magnitude peak. Therefore in order to estimate this centre value both ΔA and Δf must be estimated first. However the amplitude weighted mean frequency $\overline{f}_{\text{amp}}$ (i.e. that obtained by reassignment or other method) for a component should lie within the range of frequencies covered by the (non zero-padded) analysis bin. For example in figure 4.2 the frequency estimate obtained by reassignment in the peak bin is 1.1472 kHz and the frequency range covered by this bin is 1.1197 kHz to 1.1628 kHz. This is consistent with a sinusoidal component whereas if $\overline{f}_{\mathrm{amp}}$ lay outside of this range this suggests bin contamination by an outlying component and the component should be considered non-sinusoidal and rejected [Desainte-Catherine and Marchand, 2000]. Once ΔA and Δf have been found and the centre frequency has been found it is possible to predict in which bin the magnitude peak would reside for a sinusoid with these parameters. If the predicted peak bin is not the same as the actual peak bin then this in an indication that the component is not sinusoidal. In addition to this indicator the behaviour of the time reassignment values around the peak can be analysed to determine whether they are the same as predicted for a sinusoid with the given analysis parameters. A method for doing this is proposed in this chapter. The comparison of phase and amplitude reassignment for the sinusoidal discrimination is also assessed for non-stationary sinusoids.

Just as Flandrin (cited in section 3.5.3) describes stationarity in terms of non-stationarity so here it is proposed that peaks due to sinusoids are identified in a frame by the rejection of other peaks due to non-sinusoids. This is achieved by estimating the parameters of the sinusoid and then determining whether the behaviours of the peak match those predicted for such a sinusoid. Once non-sinusoidal peaks have been rejected all other peaks are assumed to be sinusoids. The reassignment method is used to produce the estimate of $\overline{f}_{\mathrm{amp}}$ (i.e. that which does not yet take ΔA and Δf into account). This is because reassignment can produce estimates of $\overline{f}_{\mathrm{amp}}$ from a single frame whereas the phase difference and derivative algorithms require more than a single frame of data. The parabolic interpolation and triangle methods have not been used since these have been shown not to perform as well for stationary sinusoids as these three methods [Keiler and Marchand, 2002]. Also these two methods do not discriminate against bin contaminants since they take no account of phase. Since reassignment analysis is used the phase distortion analysis (PDA) technique [Masri, 1996] is adapted for this form of analysis in this thesis.

4.3 Reassignment distortion analysis

Phase distortion analysis uses phase deviations either side of magnitude peaks in the DFT spectrum ((3.43) and (3.44)). For frequency and time reassignment these deviations are embedded in the frequency and time offset estimates respectively ((3.132) and (3.133)). Figures 4.3 to 4.8 show the phase and frequency and time reassignments across the main lobe for both a stationary sinusoid and for a non-stationary sinusoid. It can be seen that non-stationarity produces perturbations of phase and reassignment measures in and around the main lobe. Both time and frequency reassignment values exhibit similar perturbations to these as well. Time reassignment is also a measure of ΔA since it analyses the energy distribution in time; an increase in amplitude during a frame shifts energy towards the end of the frame, a decrease shifts it towards the start.

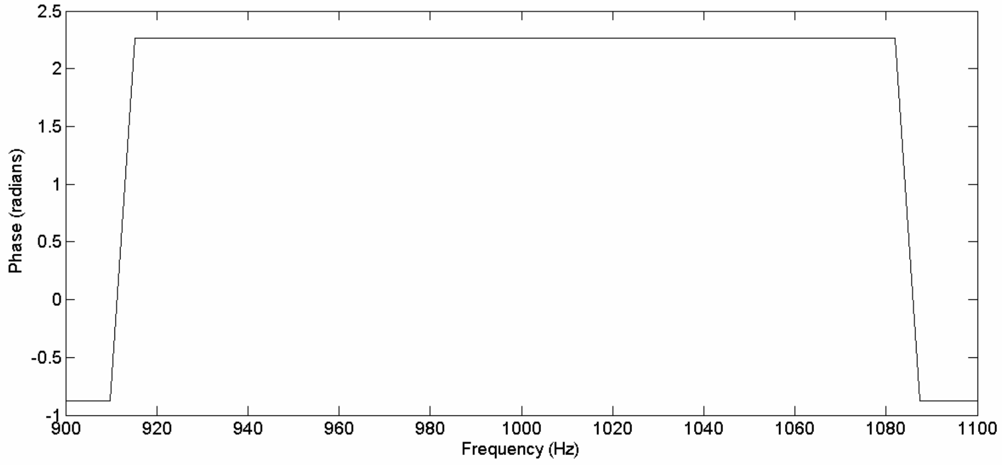


Figure 4.3: DFT phase of a stationary Hann windowed sinusoid.

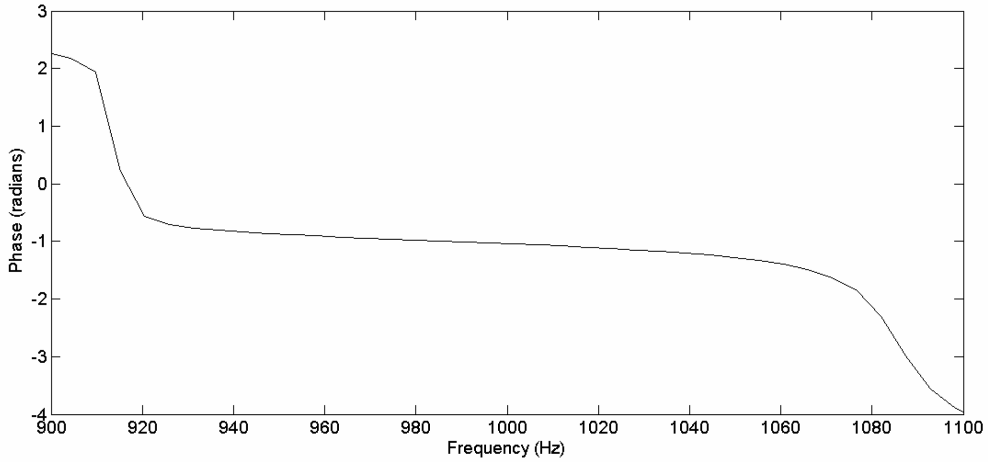


Figure 4.4: DFT phase of a non-stationary Hann windowed sinusoid. The amplitude of the sinusoid increases exponentially by 6 dB during the frame. The frequency increases linearly from 995 Hz to 1005 Hz.

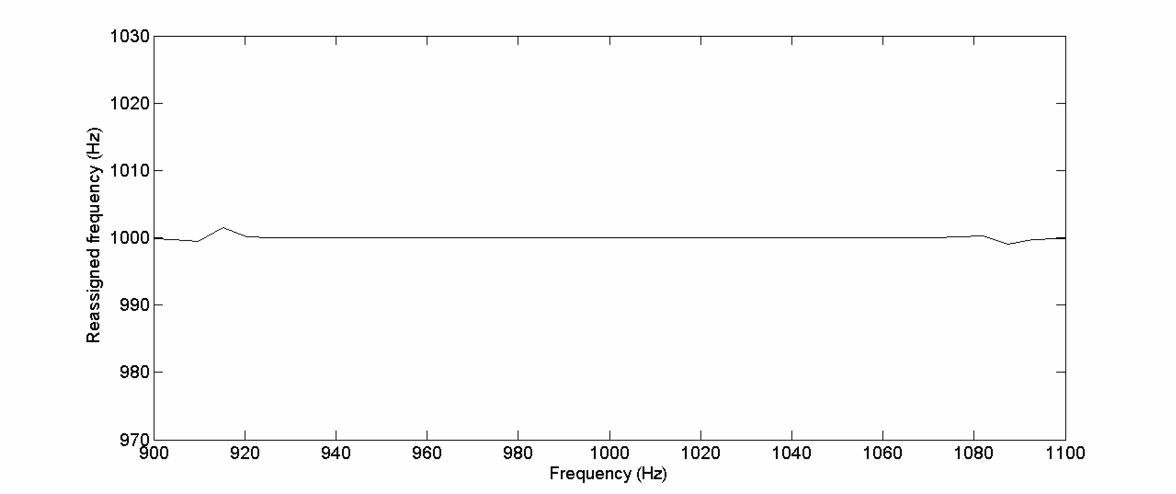


Figure 4.5: Frequency reassigned DFT of a stationary Hann windowed sinusoid.

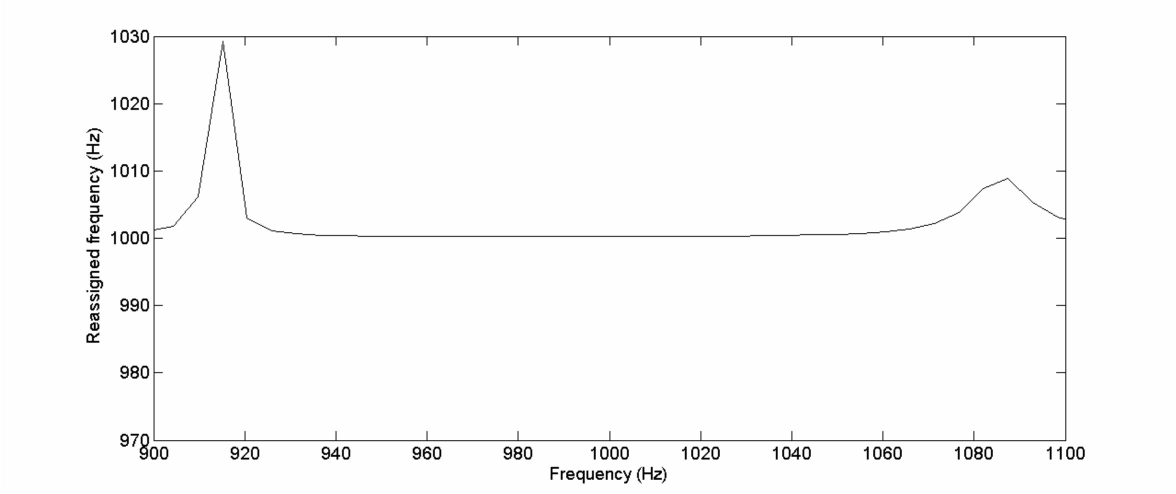


Figure 4.6: Frequency reassigned DFT of a non-stationary Hann windowed sinusoid. The amplitude of the sinusoid increases exponentially by 6 dB during the frame. The frequency increases linearly from 995 Hz to 1005 Hz.

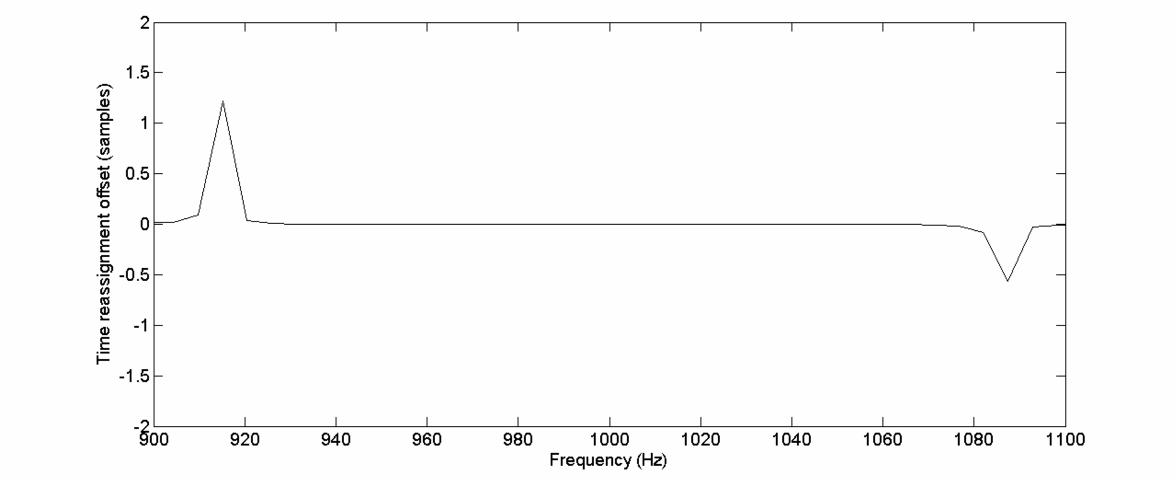


Figure 4.7: Time reassigned DFT of a stationary Hann windowed sinusoid.

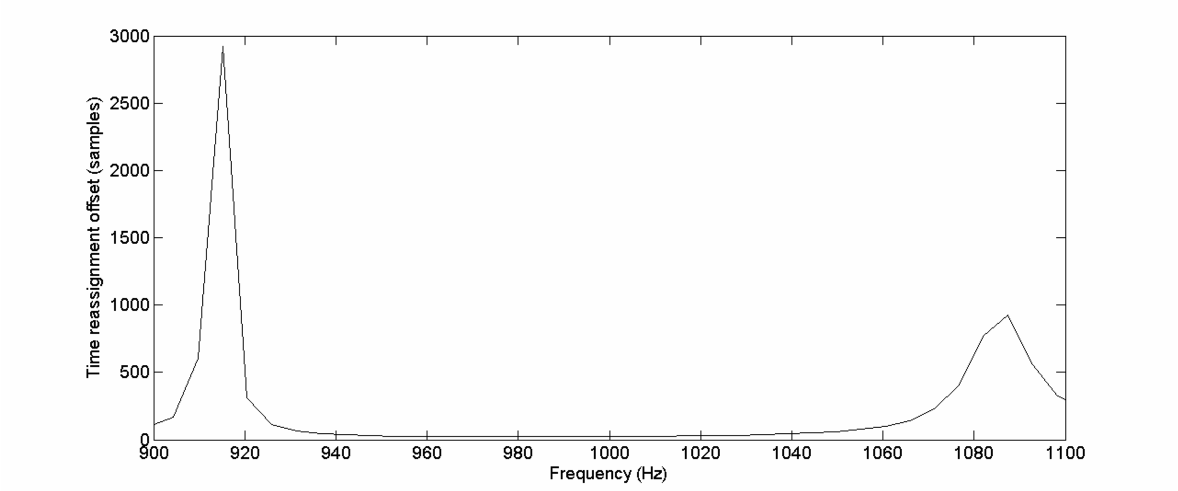


Figure 4.8: Time reassigned DFT of a non-stationary Hann windowed sinusoid. The amplitude of the sinusoid increases exponentially by 6 dB during the frame. The frequency increases linearly from 995 Hz to 1005 Hz.

Figures 4.9 to 4.11 show how the time reassignment offsets close to the centre of the peak behave for various values and combinations of ΔA and Δf. PDA is based on the fact that there is a relationship between Δf and the difference in phase across the peak and that there is a relationship between ΔA and the combined differences between the peak phase and the phases either side (as described by equations 3.43 and 3.44). RDA is also based on these relationships (with regard to time reassignment offset instead of phase) but with measures for ΔA and Δf exchanged. PDA effectively models the phase as a first order polynomial:

$$y = mx + c \qquad (4.2)$$

where *y* represents the phase value, *x* the bin number, *m* the value from which Δf is derived and *c* the value from which ΔA is derived. This thesis proposes modelling time reassignment data as a second order polynomial and using ‘goodness’ of fit across the main lobe as a measure of non-sinusoidality. The benefits of second, as opposed to first, order polynomial are explained later in this section.

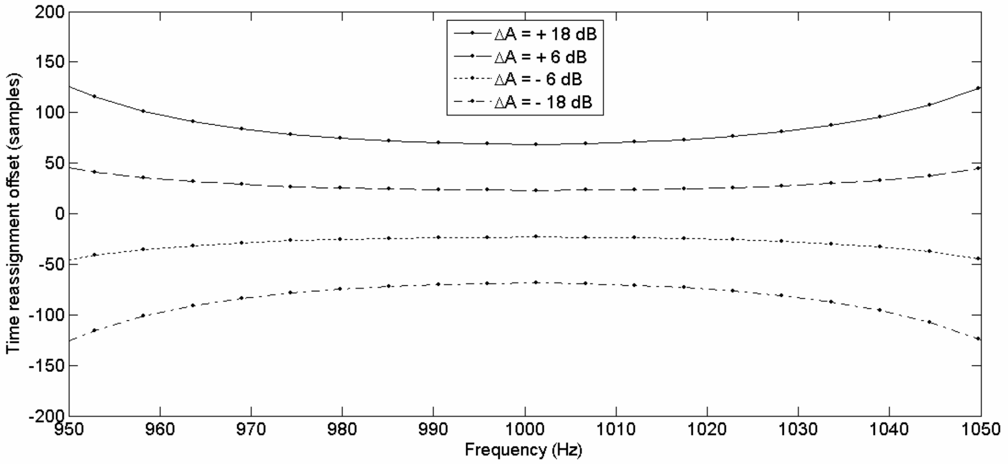


Figure 4.9: Time reassigned DFT of a non-stationary Hann windowed sinusoid for various values of ΔA. The frequency is stationary at 1 kHz.

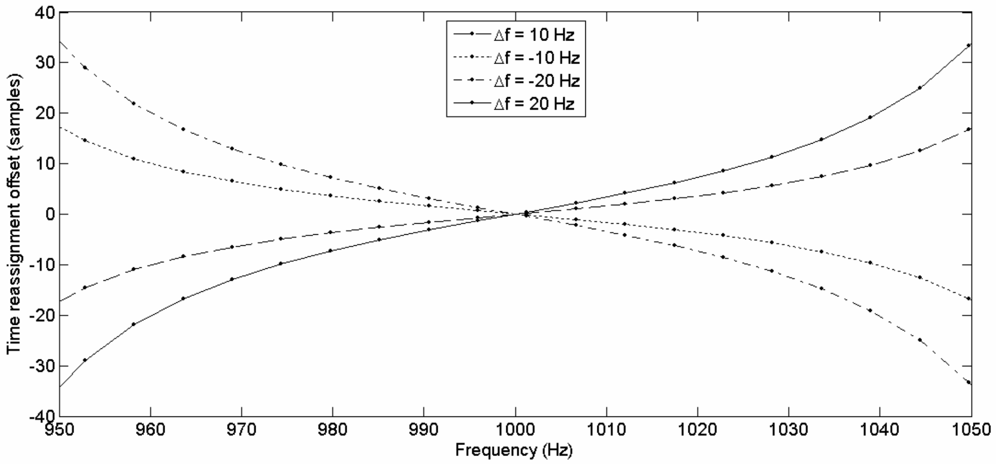


Figure 4.10: Time reassigned DFT of a non-stationary Hann windowed sinusoid for various values of Δf. The amplitude is stationary.

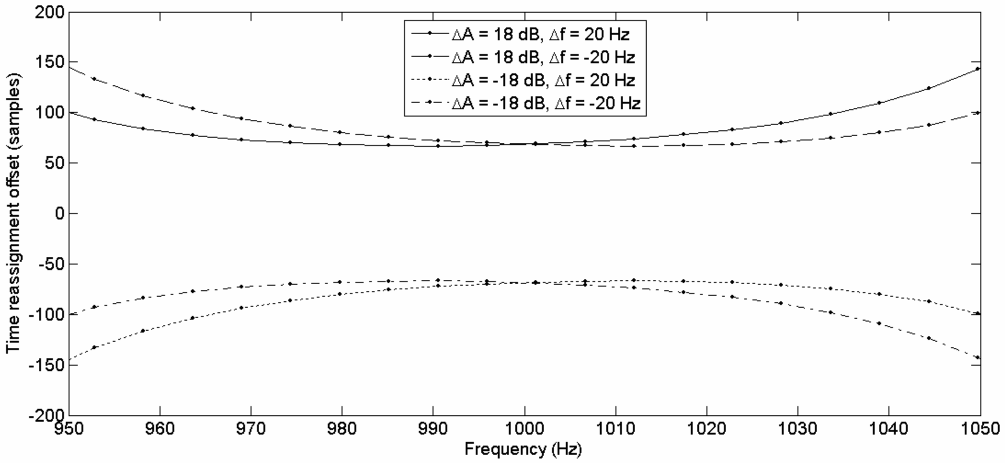


Figure 4.11: Time reassigned DFT of a non-stationary Hann windowed sinusoid for various combinations of values for Δf and ΔA.

4.3.1 Estimating ΔA and Δf from RDA data

Just as with PDA, RDA does provide corresponding values that are unique for all values of ΔA although this is not the case for Δf . Figure 4.12 shows the RDA measure *c*, in (4.2), for different values of ΔA when using the Hann window. It should be noted that the relationship between ΔA and *c*, although linear for relatively small values of $|\Delta A|$, is non-linear across the whole range of values of amplitude change that could occur in a 16 bit (96 dB SNR) or higher system. Much work that uses PDA estimates of ΔA assumes that this relationship is linear [Masri, 1996], [Lagrange et al, 2002].

Figure 4.13 shows the RDA measure *m*, in (4.2), for different values of Δf . It can be clearly seen that there is not a unique relationship between these two quantities. As with PDA the relationship between *m* and Δf depends on the size of the window used. Extending the results presented here to arbitrary sample rates and frame lengths is discussed in section 4.8. As for PDA, the exact relationships shown in figures 4.12 and 4.13 vary according to the window chosen. Also for small values of Δf and ΔA, *m* and *c* are largely independent of each other but this is not the case for larger values. Figures 4.14 and 4.15 show how the relationships shown in figures 4.12 and 4.13 are dependent upon each other.

Although in the presence of no amplitude change the relationship between ΔA and c is smooth this is not the case when there is also a change in frequency. Likewise, but more markedly, the relationship between Δf and m is not smooth where there is amplitude change. As figure 4.16 shows the effect of a large amplitude change (100 dB) upon this relationship is dramatic and this effect is frequency related. This can be ameliorated by fitting the straight line to data more closely localised to the peak (i.e. by increasing the zero-padding factor) or by increasing the order of the polynomial.

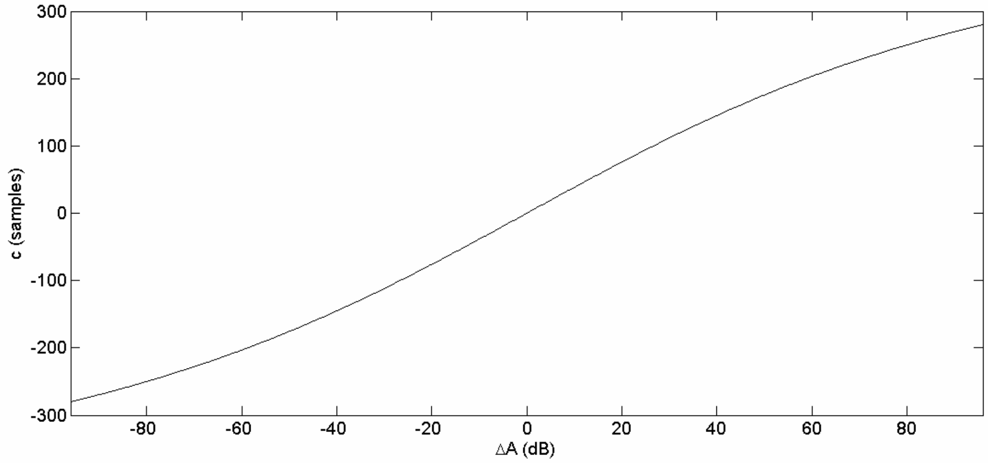


Figure 4.12: RDA measure, c in (2), against the exponential change in amplitude for a single Hann windowed sinusoid. The frequency is stationary at 1 kHz.

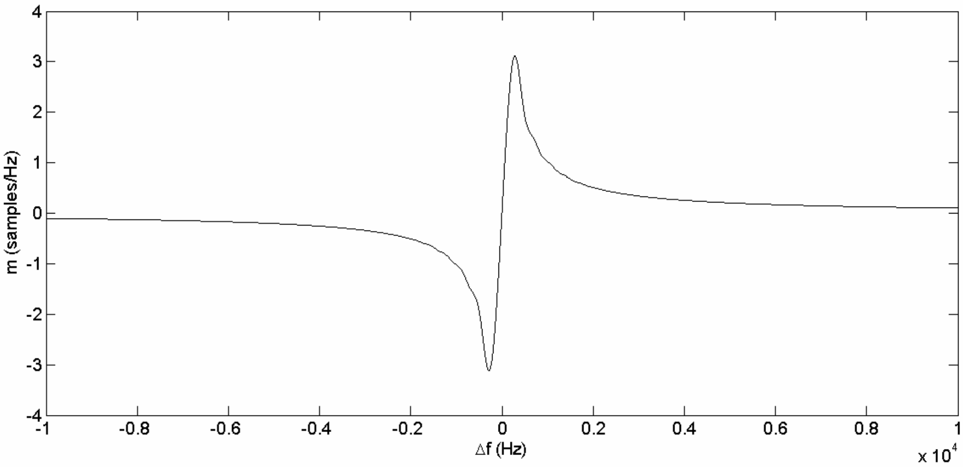


Figure 4.13: RDA measure, m in (4.2), against the linear change in frequency for a single Hann windowed sinusoid. The mean frequency for each measurement is 10 kHz. The amplitude is stationary.

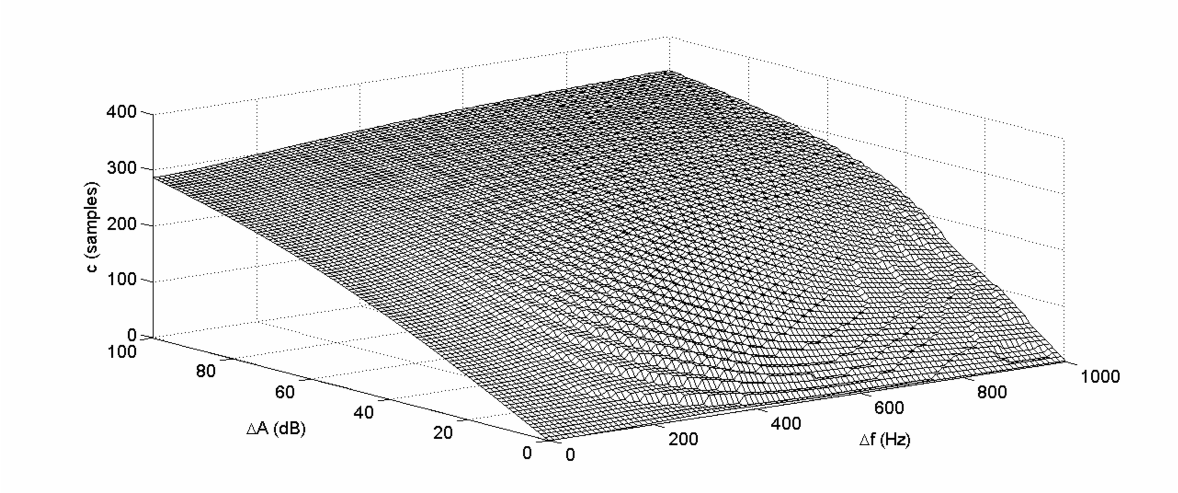


Figure 4.14: Relationship between RDA measure, c in (4.2), and ΔA versus the linear change in frequency for a single Hann windowed sinusoid. The mean frequency for each measurement is 10 kHz.

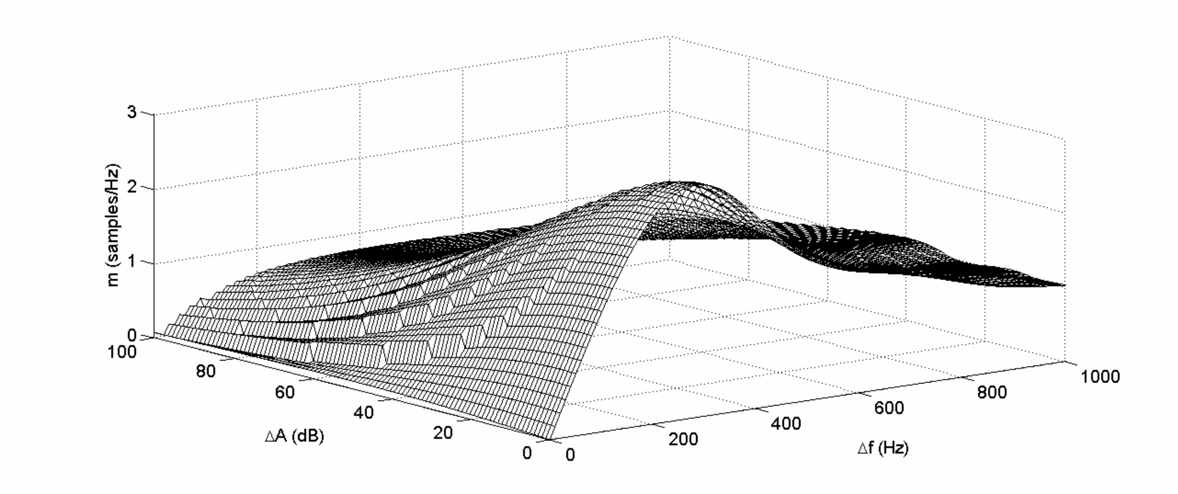


Figure 4.15: Relationship between RDA measure, *c* in (4.2), and Δf versus the exponential change in amplitude for a single Hann windowed sinusoid.

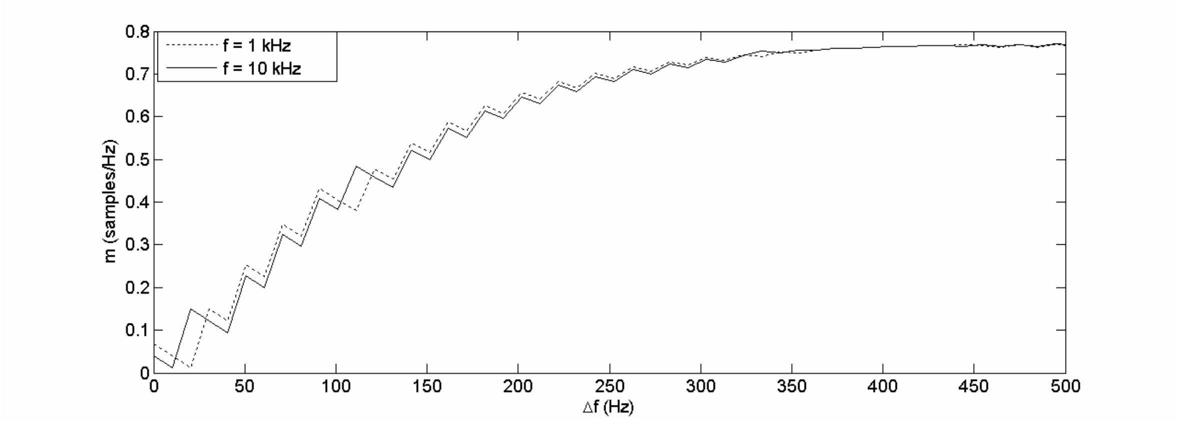


Figure 4.16: Relationship between RDA measure, *m* in (4.2) and Δf for a 100 dB exponential change in amplitude for a single Hann windowed sinusoid at a frequency of 1 kHz and 10 kHz.

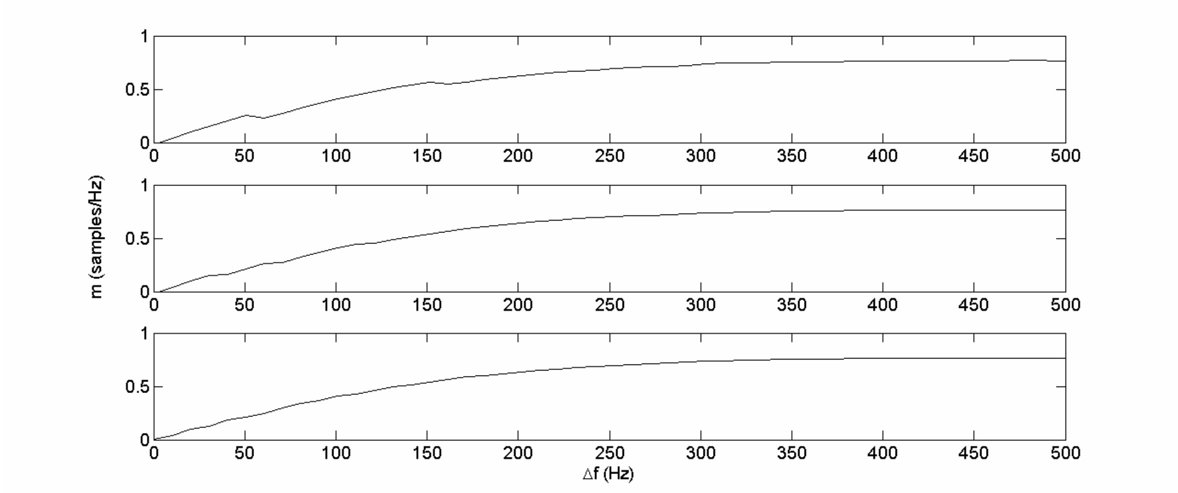


Figure 4.17: Relationship between RDA measure, m in (2) and Δf for a 100 dB exponential change in amplitude for a single Hann windowed sinusoid at a frequency of 10 kHz. The window length is 1025 samples, the zero-phase DFT size is 16,384 samples (top), 32,768 samples (middle) and 65,536 samples (bottom).

Figure 4.17 shows the effect of increasing the zero-padding factor. This improvement suggests a relationship between *m*, Δf and the distance of $\overline{f}_{\mathrm{amp}}$ from the centre of the peak bin. Such an improvement comes at a significant cost. Assuming that the FFT calculation speed is directly related to the number of complex multiplications (*n*log(*n*) for a radix-2 FFT [Lynn and Fuerst, 1994] then increasing the size from 8192 to 16384 samples more than doubles the computational cost and increasing it from 8192 to 65,536 samples increases the cost by a factor of almost 10. For a real-time process these increases in cost are likely to be prohibitively high for many applications and platforms. For this reason, and because a zero-padding factor of 8 is commonly encountered in the current literature [Masri, 1996] [Lagrange et al, 2002], this is the factor used in the presentation of results in this chapter. For certain applications, and as faster processing and larger memory becomes available, it may be that a higher zero-padding factor is desirable. Considerable smoothing can be achieved by modelling the data with a second, rather than first, order polynomial:

$$y = px^2 + mx + c \qquad (4.3)$$

The smoothing effect of this is shown in figure 4.18 and it can be seen that it provides a much smoother relationship between *m* and Δf than that obtained when using a zero-padding factor of 64. This is at the cost of a smaller range of values of Δf over which *m* is monotonically increasing. Modelling with a second order polynomial has a negligible effect on the relationship between *c* and ΔA but, as can be seen from comparing figures 4.15 and

4.19, that between m and Δf is considerably smoother reducing frequency dependent errors in the estimation of frequency change in the presence of amplitude change.

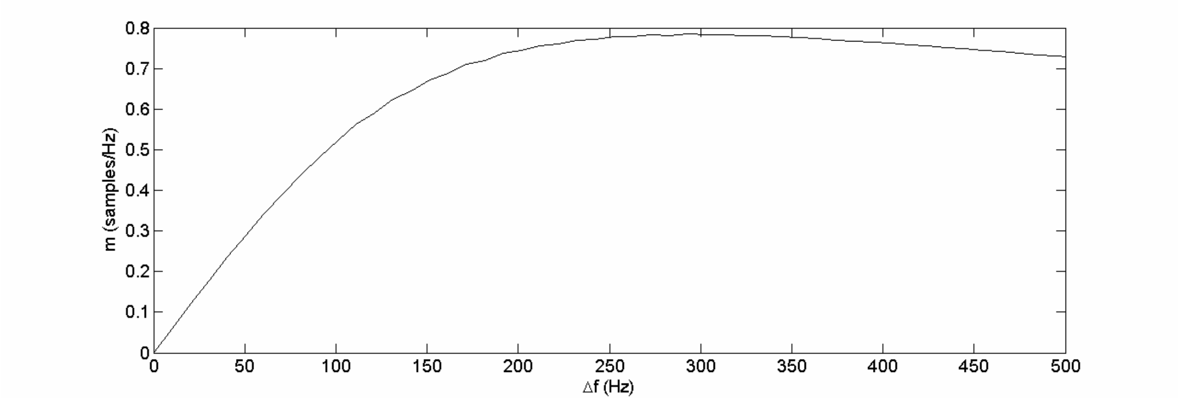


Figure 4.18: Relationship between RDA measure, m in (4.3) and Δf for a 100 dB exponential change in amplitude for a single Hann windowed sinusoid at a frequency of 10 kHz.

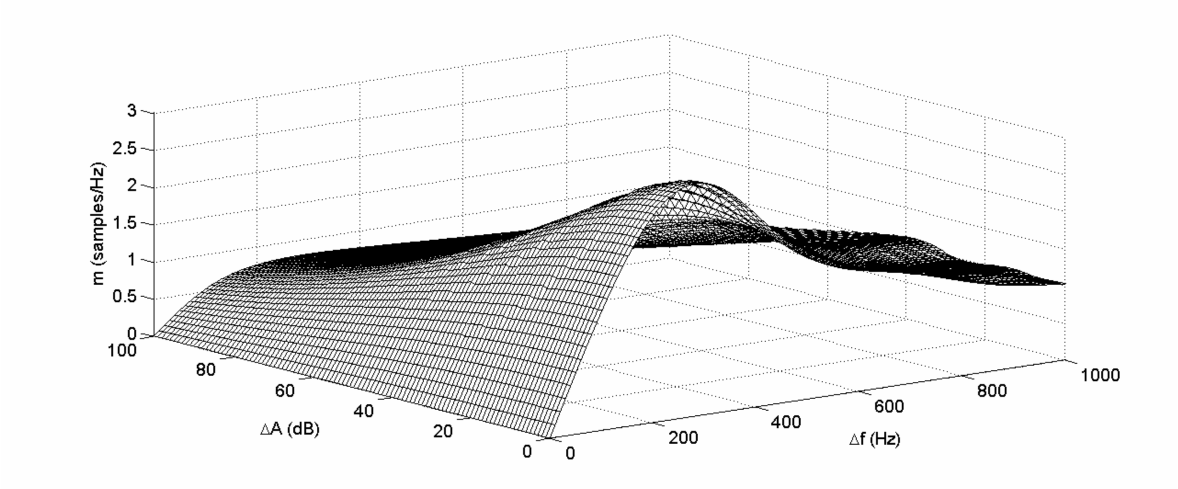


Figure 4.19: Relationship between RDA measure, m in (4.3), and Δf versus the exponential change in amplitude for a single Hann windowed sinusoid.

4.3.2 Interdependence of ΔA and Δf estimates

The magnitude of Δf has some effect on the relationship between ΔA and c with a 21% increase in the size of c for an amplitude change of 90 dB when Δf changes from 0 to 1000 Hz. However the effect of changes in ΔA on the relationship between Δf and m is much greater. For a change in amplitude of 90 dB m is 71 % lower than it is for no amplitude change with a frequency change of 250 Hz. This indicates that if a sound model is to incorporate sinusoids that rapidly change in both frequency and amplitude within a single frame then the assumption that c and m are largely immune to changes in Δf and ΔA respectively is no longer valid.

Proposed here is a new iterative method which uses two dimensional (2D) array lookup to produce much more accurate estimates of both Δf and ΔA in situations where the magnitudes of both are high. This gives improved estimates for these two quantities over those obtained when the distortion analysis measures are assumed to be independent of each other (as is the case for all applications of PDA to date). The two 2D arrays that are used contain data generated by synthesizing sinusoids for different combinations of different values of ΔA and Δf. Modestly sized (100 by 100 element) arrays are used since, with second order polynomial modelling, the functions are relatively smooth. Linear interpolation is used to extrapolate between elements. The range of values is 0 – 96 dB for ΔA and 0 – 260 Hz Δf. The range for Δf is chosen approximately to be range where, for increasing $|\Delta f|$, $|m|$ is also increasing. The range for ΔA is chosen as the largest range of amplitudes that can be represented in a 16 bit system.

The steps of the iterative algorithm are as follows:

1. Estimate ΔA from the amplitude change array assuming that Δf is 0 Hz since changes in Δf have a smaller effect on *c* than those in ΔA have on *m*.

2. Estimate Δf from the frequency change array assuming that ΔA is the value derived in the previous step.

3. Estimate ΔA from the amplitude change array assuming that Δf is the value derived in the previous step.

4. Repeat steps 2 and 3 until the algorithm is terminated.

The termination point may be determined by the processing power available (particularly in a real-time context), the required accuracy of estimates or the number of iterations before the final estimates of ΔA and Δf are no longer improved by repeated steps but begin to oscillate either side of their correct values. Figure 4.20 shows the relationship between Δf and its estimated value $\Delta f_{\text{estimate}}$ for different values of ΔA assuming that $\Delta A = 0$ This figure clearly illustrates the adverse impact that amplitude change has upon frequency change estimates using this one dimensional interpolation technique. As Δf increases the estimates begin to 'undershoot' where ΔA is non-zero due to the estimation process not taking account of how amplitude changes reduce the height of the curve in figure 4.19. Figure 4.21 shows the results obtained when one dimensional interpolation is used to estimate ΔA and then this estimate is used to perform two dimensional interpolation to estimate Δf . Here the estimates remain close to the actual value for a greater range of Δf (approximately 0 to 100 Hz) but above this the estimates begin to 'overshoot' the actual value. This is due to estimates of ΔA being too low for higher values of Δf , as expected from figure 4.14. Increasing the number of iterations beyond 3 does not appear to improve the accuracy of the method. However, taking the mean of $\Delta f_{\text{estimate}}$ values for 3 and 4 iterations does give slightly improved performance over the three iterations case.

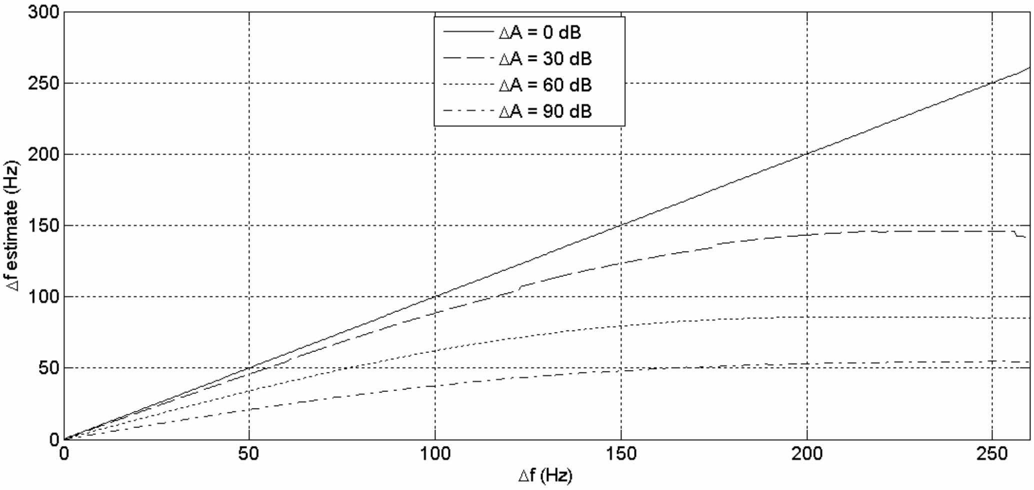


Figure 4.20: Relationship between Δf and $\Delta f_{\text{estimate}}$ for different values of ΔA without using an estimate of ΔA to improve value obtained for $\Delta f_{\text{estimate}}$.

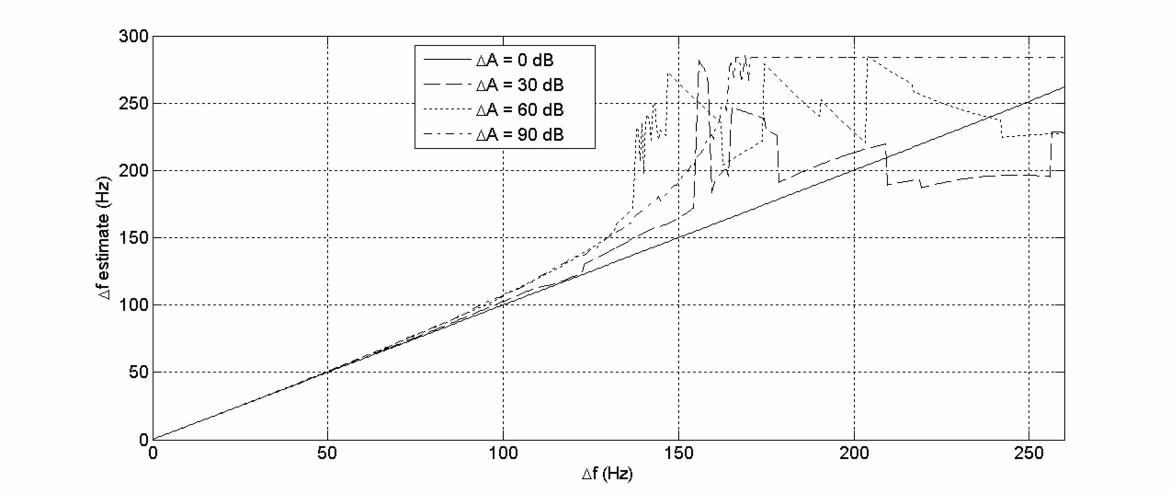


Figure 4.21: Relationship between Δf and $\Delta f_{\text{estimate}}$ for different values of ΔA, first estimating ΔA using 1D interpolation and then Δf using 2D interpolation from this estimate.

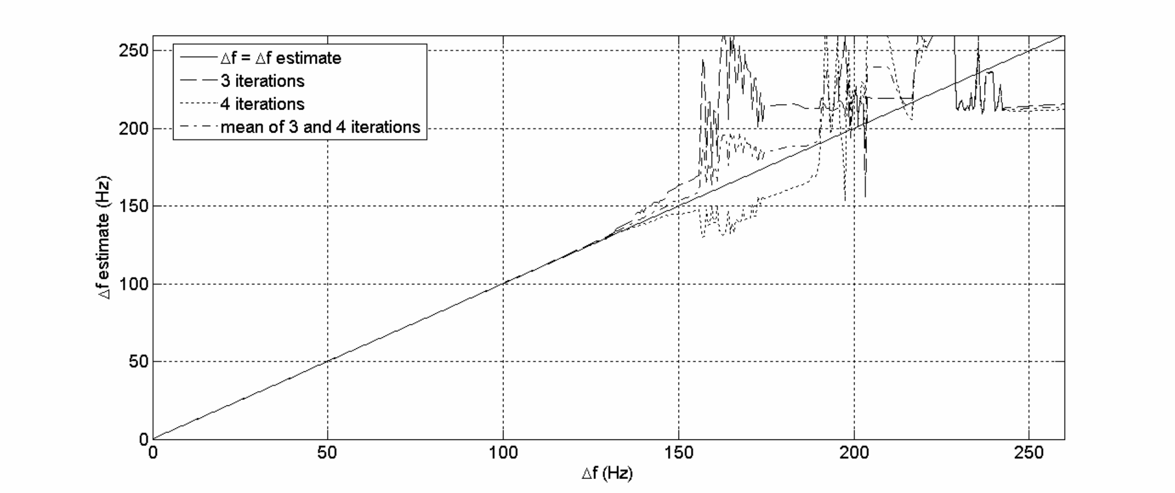


Figure 4.22: Estimation of Δf using three and four iterations and the mean of these estimates.

For some of these figures the estimated values appear to be 'hard limited' (for example the ΔA = 60 dB case above Δf = 200 Hz in figure 6e). This is due to the constraining of *m, c,* ΔA and Δf values at some points during the iteration process. Figures 4.23 to 4.25 show the percentage error for this method (taking the average of 3 and 4 iterations) and that which occurs when measures of ΔA and Δf are assumed to be independent as they have been in previous studies. These figures clearly illustrate the improvement in Δf estimation that this new method provides in the presence of amplitude change. For changes in frequency of up to 150 Hz the error is kept very small even for very large changes in amplitude. Above this range the error remains within 20% although the error performance is erratic in this region.

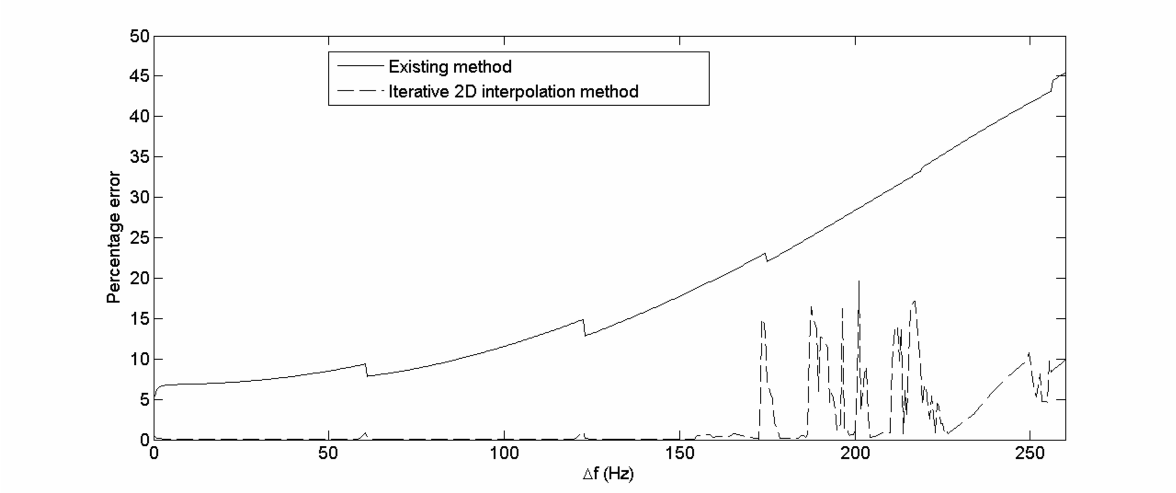


Figure 4.23: Percentage error versus Δf for both estimation methods where ΔA = 30 dB.

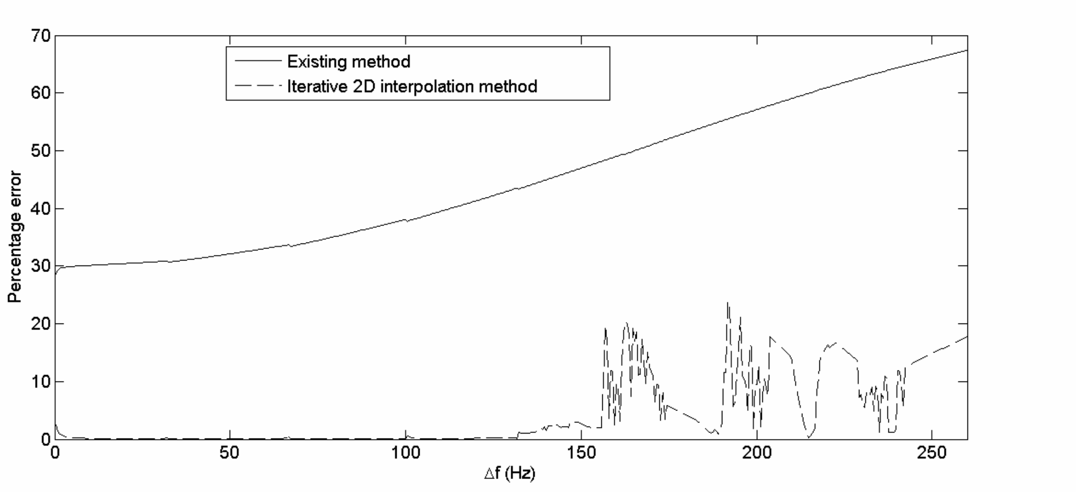


Figure 4.24: Percentage error versus Δf for both estimation methods where ΔA = 60 dB.

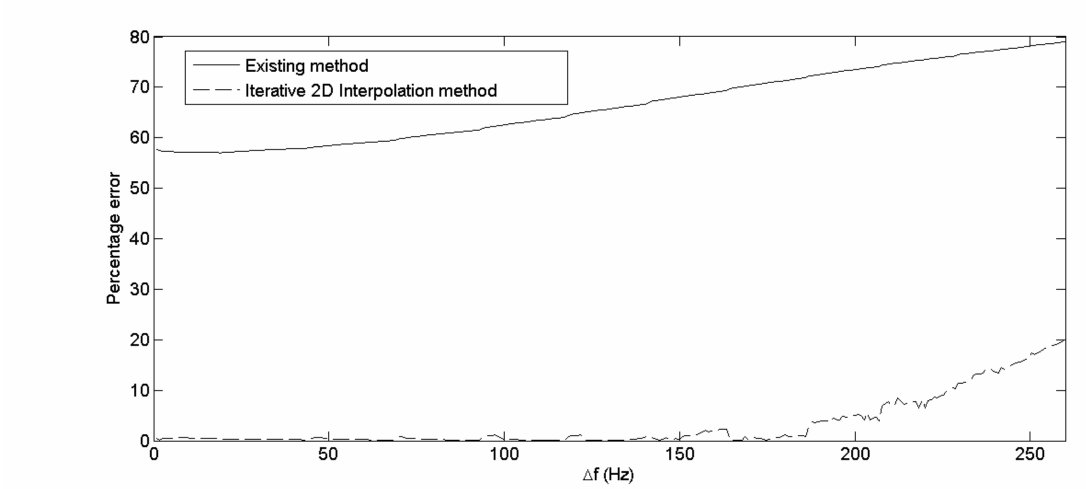


Figure 4.25: Percentage error versus Δf for both estimation methods where ΔA = 90 dB.

Figures 4.26 and 4.27 compare the iterative and existing methods when estimating ΔA. Figure 4.26 shows estimates of ΔA when this quantity is assumed to be independent of Δf. Figure 4.27 demonstrates the improvement for four iterations with the mean of the last two iterations taken. As is the case when estimating Δf some estimates exhibit 'hard limiting' where values in the estimation process have been constrained.

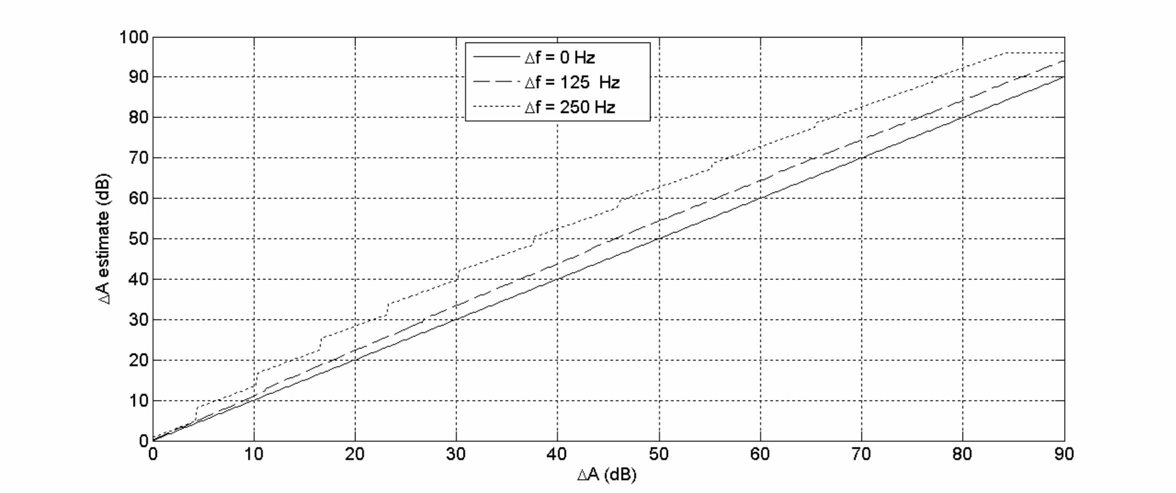


Figure 4.26: Relationship between ΔA and $\Delta A_{\text{estimate}}$ for different values of Δf without using an estimate of ΔA to improve the value obtained for the $\Delta A_{\text{estimate}}$.

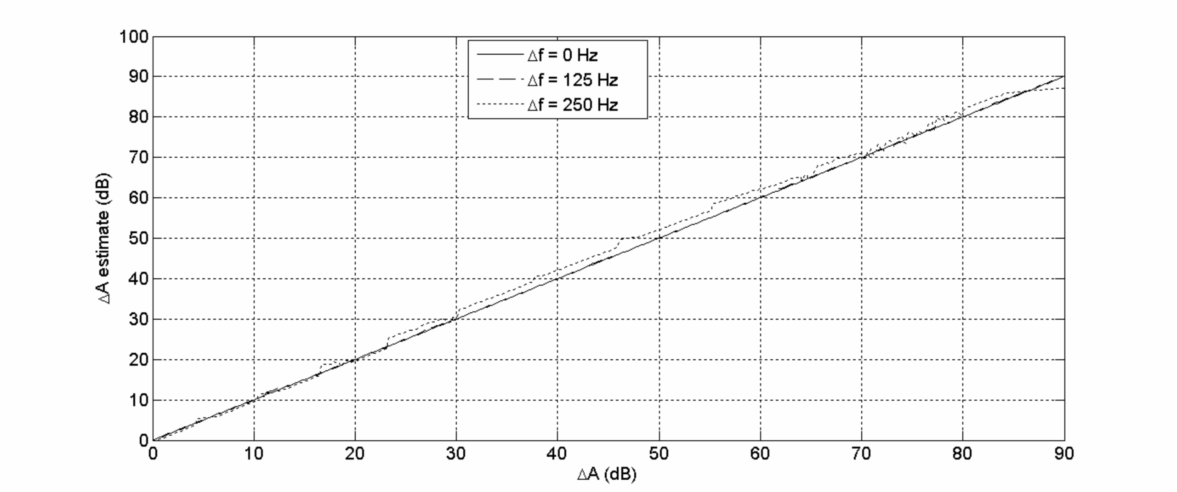


Figure 4.27: Relationship between ΔA and $\Delta A_{\text{estimate}}$ where the latter is the mean of estimates obtained from 3 and 4 iterations.

Figures 4.28 and 4.29 show the percentage error for this new estimation method compared to the existing method for two different values of Δf. The percentage error is logarithmically scaled in these plots due to the large difference in magnitude between the error for the two different methods. In fact where there is no frequency change the existing method performs better then the new method proposed here, however in this case the maximum error is negligible (< 0.004 %). Where there is a frequency change the 2D interpolation clearly out-performs the existing method.

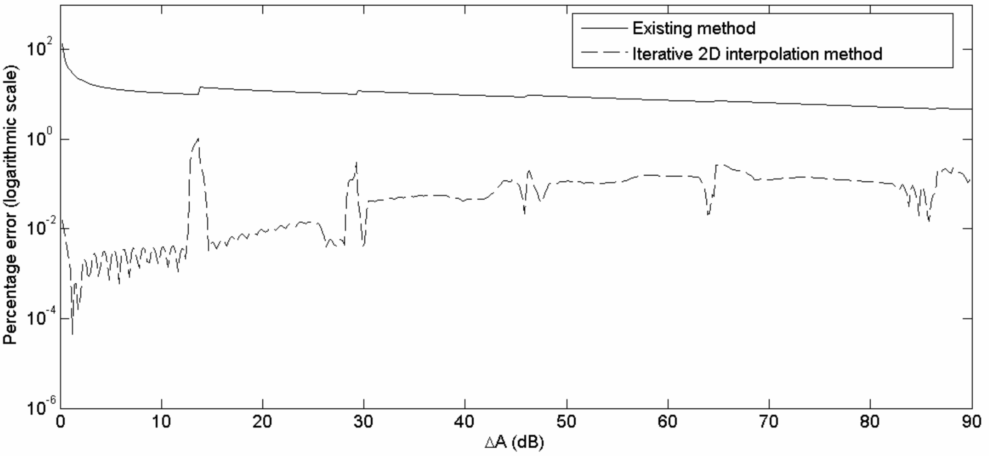


Figure 4.28: Percentage error versus ΔA for both estimation methods where Δf = 125 Hz.

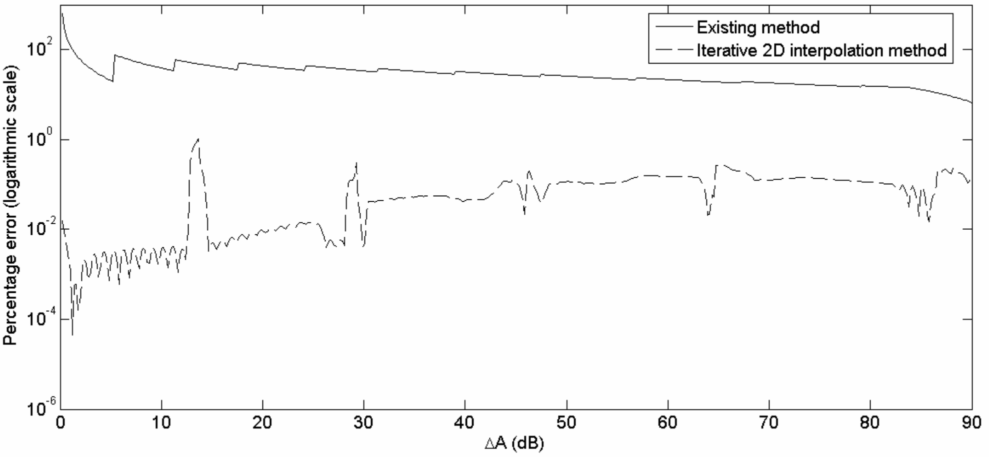


Figure 4.29: Percentage error versus ΔA for both estimation methods where Δf = 250 Hz.

4.3.3 Summary of algorithm

By means of a summary of the proposed RDA method pseudo code of the algorithm used to produce the improved estimates shown in the figures in section 4.3.2 is given:

1. Perform zero-phase and zero-padded windowing on odd-length frame of input data with the basic and time-ramped window functions.

2. Perform FFT on both sets of windowed data.

3. Identify potential sinusoidal peaks

For each peak in turn:

i. Fit second order polynomial to time reassignment data and obtain *m* and *c*.

ii. Assuming Δf is 0 Hz estimate ΔA from *c* (1D linear interpolation) constraining the value of *c* beforehand (i.e. If *c* is higher than the maximum value stored in the 1D wavetable set it to the maximum value and if it is lower set it to the minimum). Constrain $|\Delta A|$ so that it lies within the range 0 – 96 dB.

iii. Use both ΔA and (constrained) *m* to produce an estimate of Δf (2D linear interpolation). Constrain $|\Delta f|$ so that it lies within the range 0 – 260 Hz.

iv. Use both Δf and (constrained) *c* to produce an updated estimate of ΔA. Constrain $|\Delta A|$.

v. Use both ΔA and (constrained) *m* to produce an updated estimate of Δf. Constrain $|\Delta f|$.

vi. Repeat steps iv and v twice.

Take the mean of the last two estimates of ΔA and Δf and use these as the output estimates of these two parameters.

4.4 Improving frequency and amplitude estimates for non-stationary sinusoids

4.4.1 Frequency estimation

Like most FFT based frequency estimators the frequency reassignment technique gives an estimate of the mean instantaneous frequency of a component during the analysis frame. Where the amplitude is constant during the frame then this estimate will be the same as the actual mean instantaneous frequency of sinusoid. Where the amplitude is not constant this estimate will be weighted by the amplitude function, where there is no frequency change during the frame this amplitude weighting will not adversely affect the mean frequency estimate. However, in the presence of amplitude and frequency change the mean frequency estimate will be biased by the amplitude function. This biased frequency estimate is referred to here as $\overline{f}_{\text{amp}}$ (the amplitude weighted mean) whereas, in order to fully separate the amplitude and frequency functions in our analysis, knowledge of the non-weighted mean instantaneous frequency, referred to here as $\overline{f}$, is required. Other non-stationary sinusoidal modelling systems simply use $\overline{f}_{\text{amp}}$, e.g. [Lagrange et al, 2002], but in a system which includes large amplitude changes, such as the 0-96 dB range adopted in this thesis, this will produce large errors in the model. For example, a 260 Hz frequency change combined with a 96 dB amplitude change can produce an error in the estimate of $\overline{f}$ of up to 74 Hz.

In this section a method is proposed to correct this bias and produce a more accurate estimate of $\overline{f}$ from estimates of $\overline{f}_{\text{amp}}$, A, ΔA and Δf. Taking the continuous case of a non-stationary sinusoid, as in (4.1), with the known parameters f, A, ΔA and Δf where ΔA is given in dB and is assumed to be exponential and Δf is given in Hz and is assumed to be linear, the sinusoid has the following amplitude and frequency functions:

$$f(t) = \overline{f} + \frac{\Delta f t}{2} \qquad (4.4)$$

$$A(t) = A\left(10^{\left(\frac{\Delta A}{40}\right)t}\right) \qquad (4.5)$$

where t is in the range -1 to 1 for a single frame. This range for t is chosen since it greatly simplifies the following integration. The mean amplitude of an analysed signal is found by integrating the amplitude weighted window function across the entire frame. Likewise the mean product of the amplitude, window and frequency functions is found by integrating that product across the same frame. The amplitude weighted mean instantaneous frequency is then given by the ratio of these two integrals. It can be shown that for a Hann-windowed function the amplitude weighted mean instantaneous frequency is given by:

$$\overline{f}_{\text{amp}} = \frac{\frac{1}{2}\int_{-1}^{1} A\left(10^{at}\right)\left(f+\frac{\Delta\overline{ft}}{2}\right)\left(\frac{1}{2}+\frac{1}{2}\cos\left(\pi t\right)\right)dt}{\frac{1}{2}\int_{-1}^{1} A\left(10^{at}\right)\left(\frac{1}{2}+\frac{1}{2}\cos\left(\pi t\right)\right)dt} \qquad (4.6)$$

$$= \frac{\left(\frac{\overline{f}}{\ln(a)}\left(a-\frac{1}{a}\right)\right)+\left(\frac{\Delta f}{\ln(a)^2}\left(a\left(\ln(a)-1\right)+\frac{1}{a}\left(\ln(a)+1\right)\right)\right)+\left(\frac{\overline{f}\ln(a)}{\pi^2+\ln(a)^2}\left(\frac{1}{a}-a\right)\right)}{\left(a-\frac{1}{a}\right)\left(\frac{1}{\ln(a)}-\frac{\ln(a)}{\pi^2+\ln(a)^2}\right)}$$

$$+\frac{\left(\frac{\Delta f}{\left(\pi^2+\ln(a)^2\right)^2}\right)\left(\frac{1}{a}\left(\pi^2-\pi^2\ln(a)-\ln(a)^2-\ln(a)^3\right)-a\left(\pi^2+\pi^2\ln(a)-\ln(a)^2+\ln(a)^3\right)\right)}{\left(a-\frac{1}{a}\right)\left(\frac{1}{\ln(a)}-\frac{\ln(a)}{\pi^2+\ln(a)^2}\right)}$$

(4.7)

where:

$$a = 10^{\left(\frac{\Delta A}{40}\right)t} \qquad (4.8)$$

Rearranging to find $\overline{f}$ gives:

$$\overline{f} = \overline{f}_{\text{amp}} - \frac{\left(\frac{\Delta f}{\ln(a)^2}\left(a\left(\ln(a)-1\right)+\frac{1}{a}\left(\ln(a)+1\right)\right)\right)}{\left(a-\frac{1}{a}\right)\left(\frac{1}{\ln(a)}-\frac{\ln(a)}{\pi^2+\ln(a)^2}\right)}$$

$$-\frac{\left(\frac{\Delta f}{\left(\pi^2+\ln(a)^2\right)^2}\right)\left(\frac{1}{a}\left(\pi^2-\pi^2\ln(a)-\ln(a)^2-\ln(a)^3\right)-a\left(\pi^2+\pi^2\ln(a)-\ln(a)^2+\ln(a)^3\right)\right)}{\left(a-\frac{1}{a}\right)\left(\frac{1}{\ln(a)}-\frac{\ln(a)}{\pi^2+\ln(a)^2}\right)}$$

(4.9)

Using this formula to improve the estimate of $\overline{f}$, rather than simply assuming $\overline{f} \approx \overline{f}_{\text{amp}}$, gives a significant improvement in the model accuracy as shown in figures 4.30 and 4.31.

These figures show the magnitude of the $\bar{f}$ estimation error for different values of ΔA and Δf with and without this bias correction. For both figures estimates of ΔA and Δf obtained using the methods described in previous sections, rather than the actual values used to synthesize the sinusoids, were used in the bias correction.

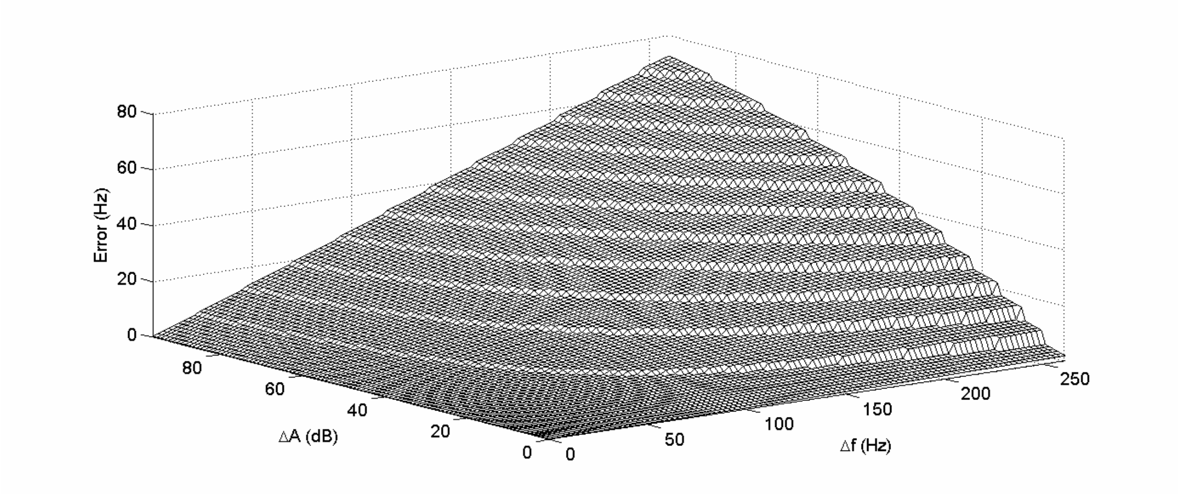


Figure 4.30: $\bar{f}$ estimation error without bias correction for a sinusoid with a true non-amplitude weighted mean frequency of 10 kHz.

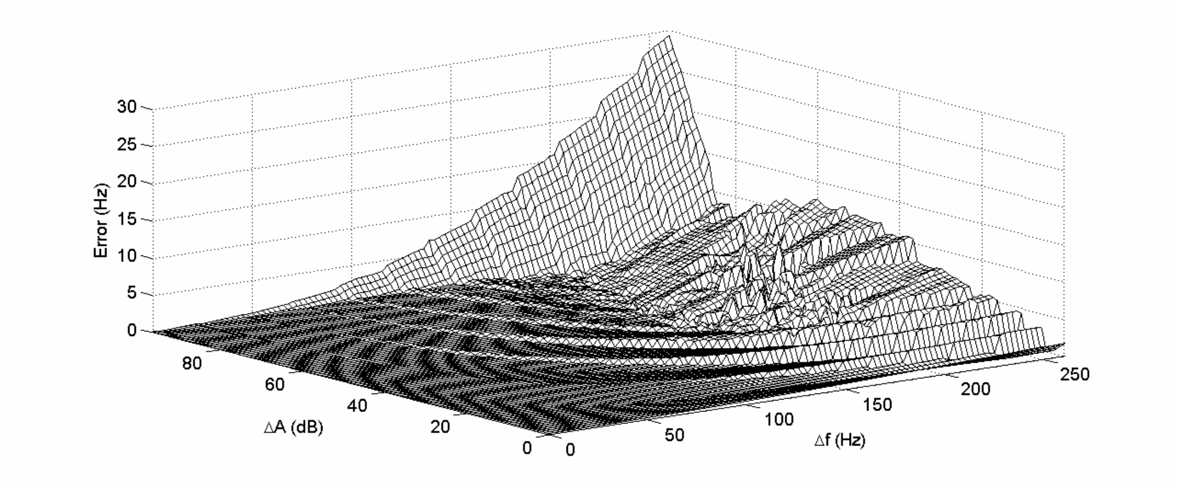


Figure 4.31: $\bar{f}$ estimation error with bias correction for a sinusoid with a true non-amplitude weighted mean frequency of 10 kHz.

4.4.2 Amplitude estimation

Amplitude estimation is also affected by non-stationarity: frequency change within a frame causes greater spreading of signal energy across bins around the peak, lowering the peak magnitude and amplitude change causes the signal to be more localised in time thus widening the main lobe. In addition to this the peak magnitude varies with the difference between $\overline{f}$ and the actual centre frequency of the bin in which the peak appears. For stationary sinusoids, knowledge of the magnitude of the window function in the frequency domain allows amplitude estimation errors caused by deviation from the centre of the analysis bin to be corrected (see section 3.11). Since no analytical solution for the Hann window of a sinusoid with non-stationary frequency is known it has been proposed to calculate the magnitude spectrum of the window via FFT [Lagrange et al, 2002]. From this an ‘amplitude factor’ is derived which is multiplied by the initial amplitude estimate. Using such an approach requires an additional FFT to be calculated for every peak in the magnitude spectrum of each frame which is likely to be prohibitively expensive in a real-time context.

Two new approaches are proposed here which do not require additional FFTs to be computed: estimation of the amplitude correction factor by two dimensional wavetable lookup (as used to estimate ΔA and Δf) and by modelling the relationship between the amplitude correction factor, ΔA and Δf , with two polynomials. Figure 4.32 shows the relationship between the normalised amplitude (i.e. the amplitude is 1 for stationary amplitude and frequency), ΔA and Δf for a sinusoid whose frequency is the exact centre of an analysis bin (10.02 kHz in this case). The values for the amplitude correction wavetable are simply derived by inverting the amplitude values. Again a 100 x 100 array is used to store values and linear interpolation is used. Figure 4.33 shows the percentage error in amplitude estimation when using this array to correct FFT derived values for non-stationary sinusoids. A 150 x 150 array is used to exercise the interpolation and the centre frequency of a different bin is used (1.001 kHz).

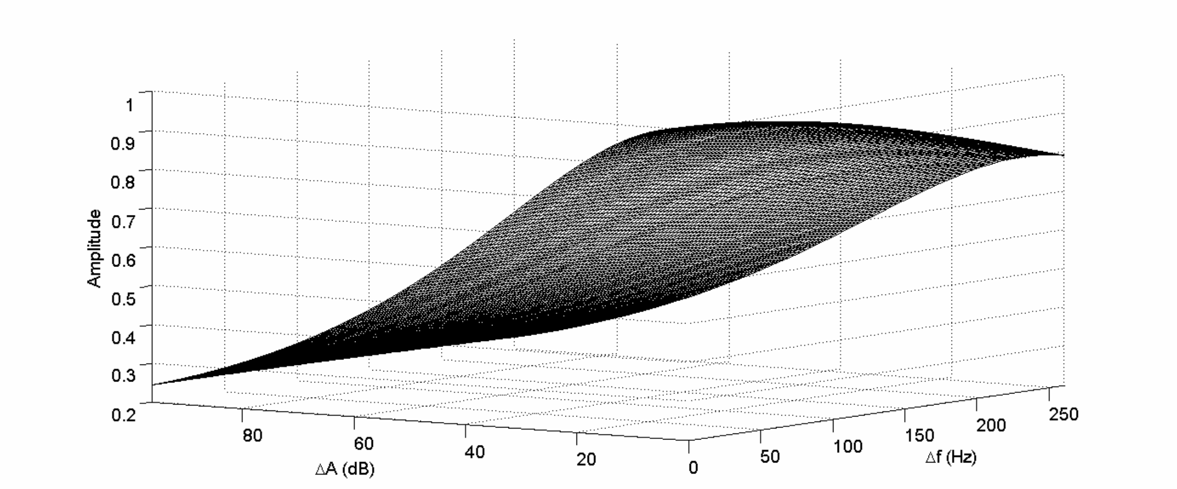


Figure 4.32: Normalised amplitude as a function of ΔA and Δf for a sinusoid with mean frequency 10.02 kHz.

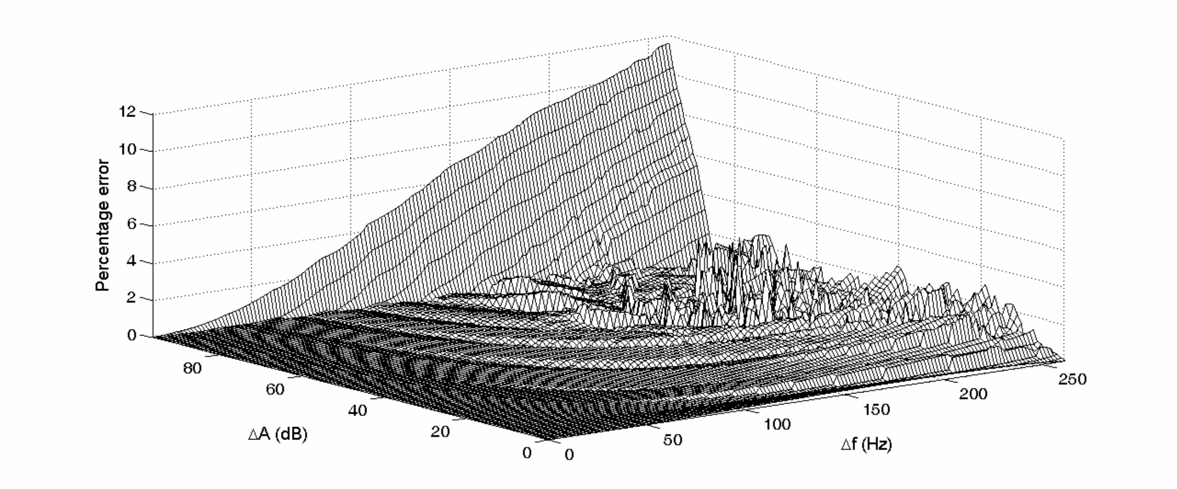


Figure 4.33: Percentage error in amplitude estimation for a sinusoids with mean frequency 1.001 kHz.

The second method investigated is to find the amplitude correction factor by multiplying two quartic polynomials. These polynomials are fitted to the data from figure 4.32: normalised amplitude against ΔA (for Δf =0) and against Δf (for ΔA=0). Figures 4.34 and 4.35 show this data and the quartics that produce the best least squares fit.

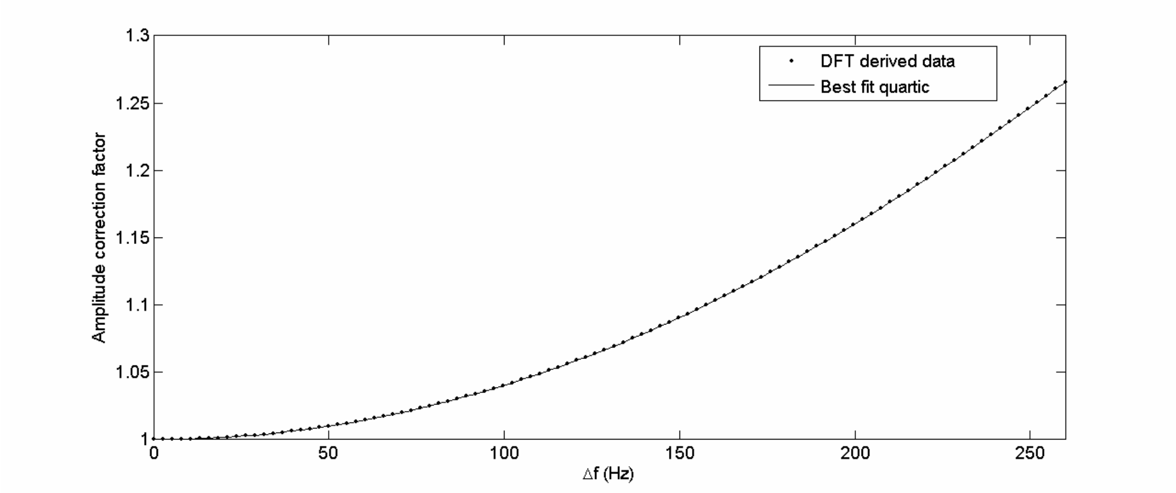


Figure 4.34: Amplitude correction factor against Δf derived from 8192 point DFT of a 1025 point Hann-windowed sinusoid and quartic polynomial which provides the best fit to this data.

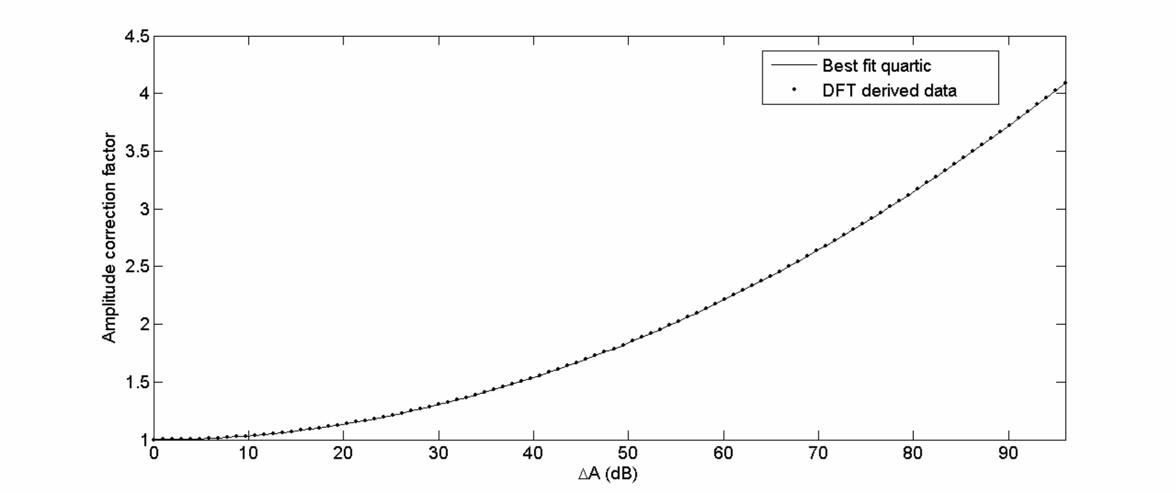


Figure 4.35: Amplitude correction factor against ΔA derived from 8192 point DFT of a 1025 point Hann-windowed sinusoid and quartic polynomial which provides the best fit to this data.

The non-stationary (ns) amplitude correction factor is given by:

$$a_{\mathrm{ns}} = f\left(\Delta A\right) g\left(\Delta f\right) \qquad (4.10)$$

where $f(x)$ and $g(x)$ are the quartic functions.

The percentage error in the amplitude estimates produced by this method is shown against ΔA and Δf in figure 4.36. As for figure 4.33, a 150 x 150 array of values and a sinusoid at 1.001 kHz was used in the error test. Clearly the two dimensional interpolation performs better in terms of error, however it may be a useful alternative in situations where memory is scarce or on systems where memory lookup is a relatively expensive operation. It is the former array lookup and interpolation approach which is used in the sinusoidal modelling system described in chapter 6 but this latter approach is included here for comparison.

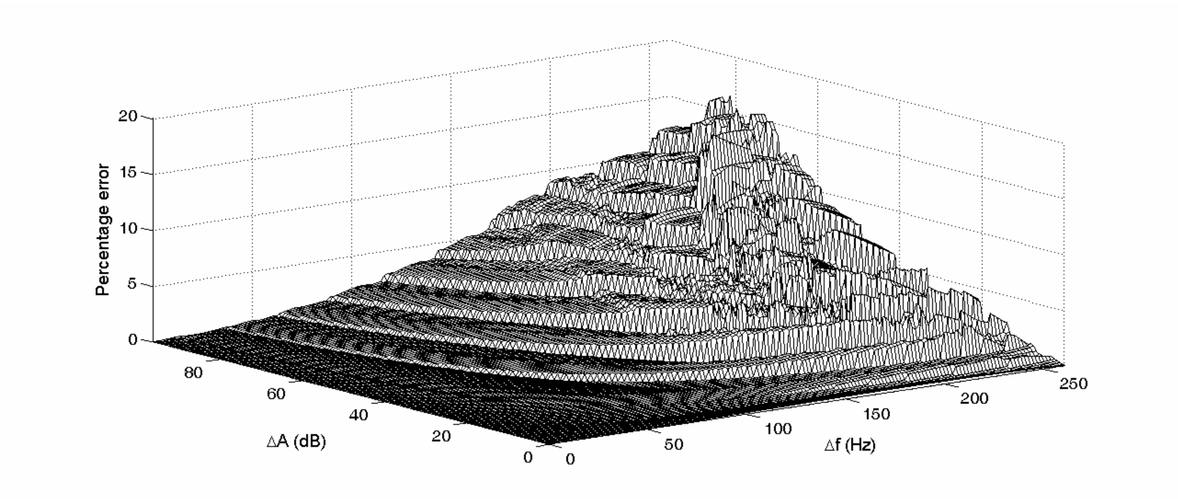


Figure 4.36: Normalised estimated amplitude of stationary sinusoid with and without correction for window shape.

Whilst these methods take account of the effect of amplitude and frequency change on amplitude estimates they do not incorporate correction for the main lobe shape. For a stationary Hann-windowed sinusoid the amplitude factor as a function of distance of the frequency from the centre of the peak bin is given as:

$$a_{\text{window}} = \frac{\pi d(1-d^2)}{\sin(\pi d)} \qquad (4.11)$$

where *d* is the distance from the centre of the bin as the proportion of the width of one bin. Equation (4.11) can be easily derived from (3.31). The advantage of using this correction factor for a stationary sinusoid is shown in figure 4.37.

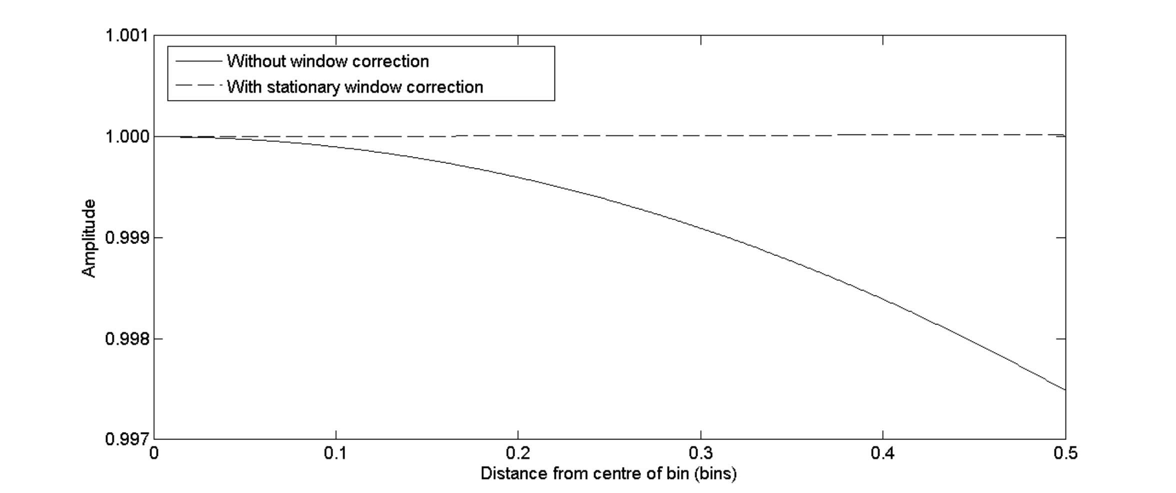


Figure 4.37: Normalised estimated amplitude of stationary sinusoid with and without correction for window shape.

Where there is flattening of the main lobe due to non-stationarity then this correction will no longer produce a constant amplitude across an analysis bin and, in cases of extreme flattening, it will actually make the amplitude estimate worse. A simple but effective modification to the stationary window correction is proposed here. This uses the non-stationary amplitude correction factor, derived using either of the methods described earlier in this section, as a measure of the lobe flattening. This modified correction for the window is given by:

$$a_{\text{ns window}} \approx \left(\frac{\pi d(1-d^2)}{\sin(\pi d)} \right)^{\frac{1}{a_{\text{ns}}}} \qquad (4.12)$$

In the presence of non-stationarity this produces an improved window correction as shown in figure 4.38 for a sinusoid whose amplitude changes by 96 dB and whose frequency is stationary. It can be seen that the stationary window correction produces an over-correction, resulting in a greater error in the amplitude estimate than when no correction is used. The proposed non-stationary window correction performs best. It should be pointed out that where there is a high zero padding factor, as is the case here, the change in amplitude without correction for window shape does not exceed the just noticeable difference for intensity which is not less than 0.3 dB [Riesz cited in Moore, 1997]. However the non-stationary correction described here is relevant in situations where lower zero-padding factors are used.

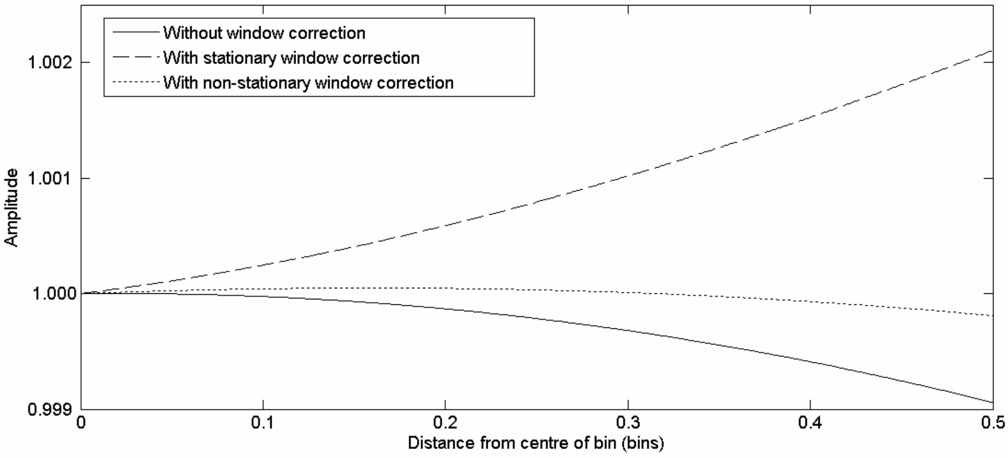


Figure 4.38: Normalised estimated amplitude of sinusoid with stationary frequency whose amplitude changes by 96 dB with and without correction for window shape.

Combining amplitude correction for non-stationarity and window shape gives the overall amplitude correction factor:

$$a_{\text{combined}} = a_{\text{ns window}} a_{\text{ns}} \qquad (4.13)$$

4.5 Predicted variance of data from polynomial fit as a measure of sinusoidality

4.5.1 Overview of method

A common method for determining whether or not a spectral component is a sinusoid is its behaviour across frames. If, using an analysis technique whose basis functions are sinusoidal, the estimated parameters of a component remain stationary, or close to stationary, over a number of frames then it is likely that such a component is a sinusoid. For example, the McAulay-Quatieri sinusoidal representation of speech links components between frames which give the smoothest frequency trajectory [McAulay and Quatieri, 1986]. A real-time system cannot be anti-causal, it cannot look forwards in time to determine whether the evolution of parameters of a given candidate sinusoidal component will match up with those of a component in the next frame. Causal analysis is possible but increases the delay between an analysed event and its manifestation in the synthesized output. For example waiting a frame before making a decision on whether a component in frame *k* is a sinusoid (by finding a component in frame *k*+1 whose parameters link with this component) means that it cannot be recognised as a sinusoid, and synthesized as such, until frame *k*+1 has been analysed. This would either double the latency of the system or result in the sinusoidal component being synthesized as part of the residual signal in its onset frame and then as part of the sinusoidal signal thereafter meaning that it is misclassified for part of its duration.

As stated in the introduction to this chapter the definition of what constitutes a stable sinusoid has been modified for the real-time analysis context of this thesis. Here a sinusoidal component is one which behaves as a sinusoid would during the current analysis frame given the estimated parameters discussed sections 4.3 and 4.4. If it does not then it is rejected as a sinusoidal candidate and is retained for further analysis as part of the residual signal. Some use can be made of causality; a component that might have been rejected on the basis of its behaviour in the current frame might still be included in the sinusoidal signal if links well with a sinusoidal component from the previous frame. This 'current frame behaviour' approach is already employed in sinusoidal analysis systems to a lesser or greater extent. For example, any sinusoidal identification algorithm that begins by searching for peaks in the spectrum is basing this part of its analysis on the expected behaviour of the magnitude response for a sinusoidal component. Other examples of more detailed analysis of the magnitude and phase behaviour around spectral peaks were discussed in section 3.6.4 of the previous chapter and are two specific examples are given in [Peeters and Rodet, 1998] and [Lagrange et al, 2002]. The prohibiting factor with the former approach is that the whole signal must be acquired (for normalisation of the fundamental frequency) and a disadvantage with the latter approach is that an FFT is required to test each sinusoidal candidate in each frame, a requirement that would be likely to be prohibitively expensive in a real-time system such as the one proposed here.

What is required is a method which can predict the behaviour of DFT data for a sinusoid given the parameters A, f, ΔA and Δf, without simulation via FFT, and compare it to the component under investigation. Since use has been made of fitting phase data (in the PDA case), or time reassignment offset data (as in the RDA case presented here), the 'goodness of fit' of the actual data to the polynomial derived from it is an obvious candidate for investigation. Whilst the energy weighted reassignment variance has previously been investigated as a measure of sinusoidality, this approach uses supervised machine learning for discrimination and does not take account of non-stationarity [Hainsworth et al, 2001].

When the data is modelled as a second order polynomial the variance of the data from a parabola varies with Δf (with $\Delta A = 0$) as shown in figure 4.39. Since three points can always be fitted perfectly to a parabola the polynomial fitting and variance measure uses two points either side of the peak. The relationship between Δf and *m* is also given in this figure as a reference. It can be seen that the points of zero variance coincide with the stationary point (around Δf = 250 Hz) and the points of inflection. Figure 4.40 shows how the variance and *c* vary with ΔA (with $\Delta f = 0$). It is clear from these figures that both ΔA and Δf affect the variance therefore the way that these two sinusoidal parameters interact to produce different variance values should be examined. Figure 4.41 illustrates how the variance changes as a function of ΔA and Δf .

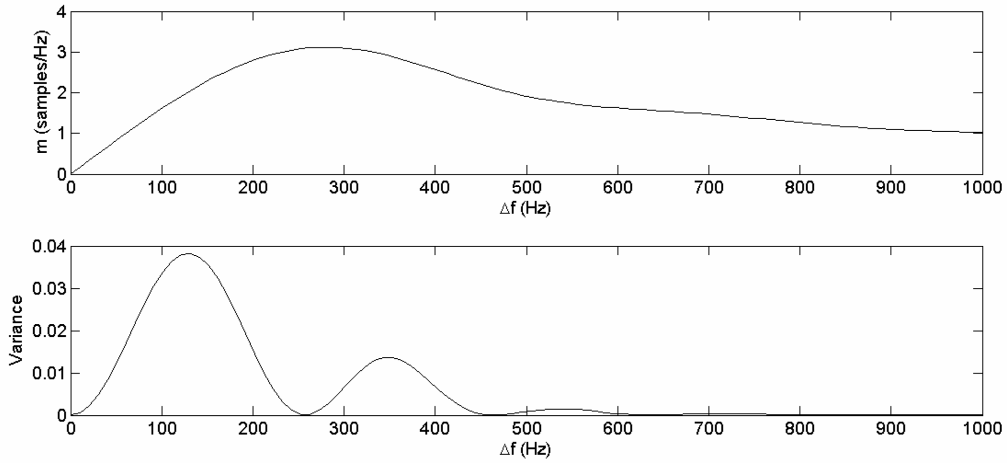


Figure 4.39: RDA measure, *m* (top), and variance (bottom) versus Δf for a single Hann windowed sinusoid. The mean frequency for each measurement is 10 kHz. The amplitude is stationary.

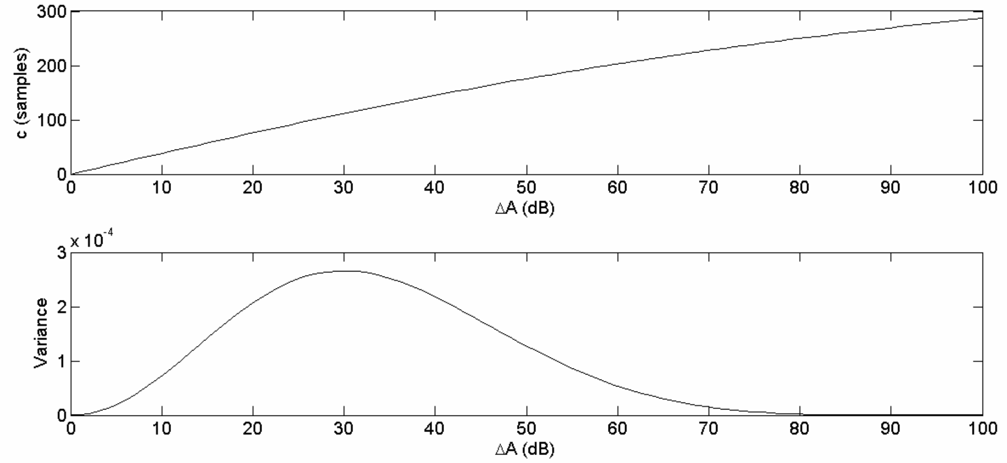


Figure 4.40: RDA measure, *c* (top), and variance (bottom) versus ΔA for a single Hann windowed sinusoid. The frequency for each measurement is stationary at 10 kHz.

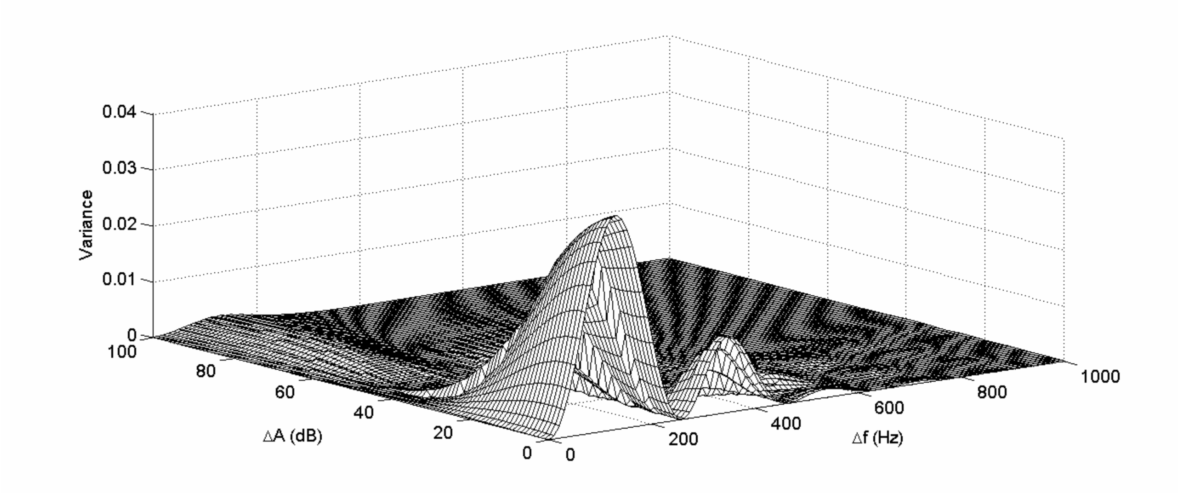


Figure 4.41: Variance as a function of ΔA and Δf for a single Hann windowed sinusoid.

The newly proposed sinusoidality test is straightforward: extract the sinusoidal parameters as described previously and then ‘look-up’ the expected variance for the given values of ΔA and Δf in a two dimensional wavetable. If the expected variance differs from the actual variance by more than a specified threshold then the component is considered not to be a sinusoid; i.e. if:

$$\left|\sigma^2_{\text{expected}} - \sigma^2_{\text{measured}}\right| = v > V \qquad (4.14)$$

where *v* denotes the modulus of the variance difference and *V* denotes the variance difference threshold (VDT), then the component is not a sinusoid. As for the two-dimensional look-up described in the previous section an array of 100 by 100 elements is used with ranges of 0 – 96 dB and 0 – 260 Hz for ΔA and Δf respectively.

4.5.2 Performance for simple tones

The first test of such a method is to determine the highest *v* for sinusoids whose ΔA and Δf values span this range but do not coincide with those values used to produce the two-dimensional array for the expected variance (so that the interpolation is properly tested) and for a different value of $\overline{f}$. This is done here by measuring *v* for a different sized 2D array with values ranging from 0 to 96 dB and from 0 to 260 Hz. The variation of *v* with ΔA and Δf is shown in figures 4.42 and 4.43 for two different values of $\overline{f}$. When estimating the expected variance the estimates of ΔA and Δf made by the algorithm described in section 4.3 are used, rather than the actual known values.

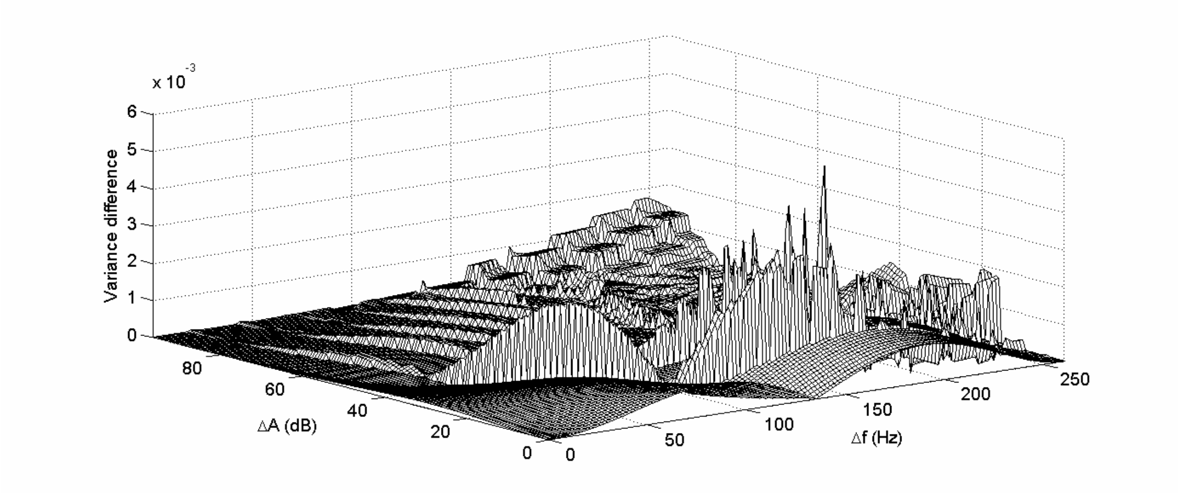


Figure 4.42: Variance difference (4.14) as a function of ΔA and Δf for a single Hann windowed sinusoid $\left(\overline{f} = 1\ \text{kHz}\right)$.

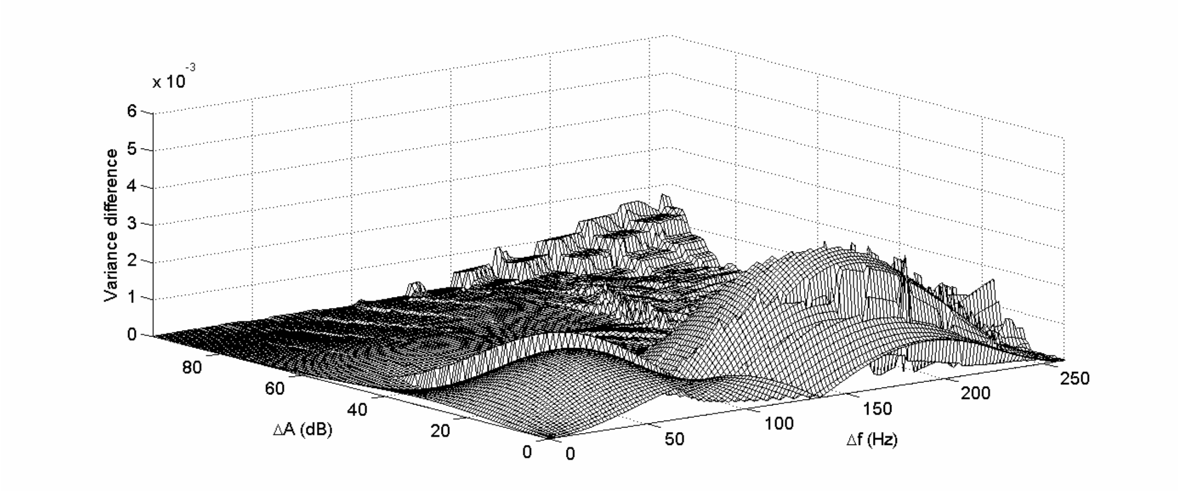


Figure 4.43: Variance difference (4.14) as a function of ΔA and Δf for a single Hann windowed sinusoid $\left(\overline{f} = 8\ \text{kHz}\right)$.

Significant differences can be seen between these two figures, demonstrating that the variance is a function of $\overline{f}$ as well as ΔA and Δf. As is the case for the relationship between m and Δf, the variance fluctuations are smoothed by increasing the zero-padding factor. Figures 4.44 and 4.45 show how the relationship between variance and frequency for a non-stationary sinusoid changes with the zero-padding factor.

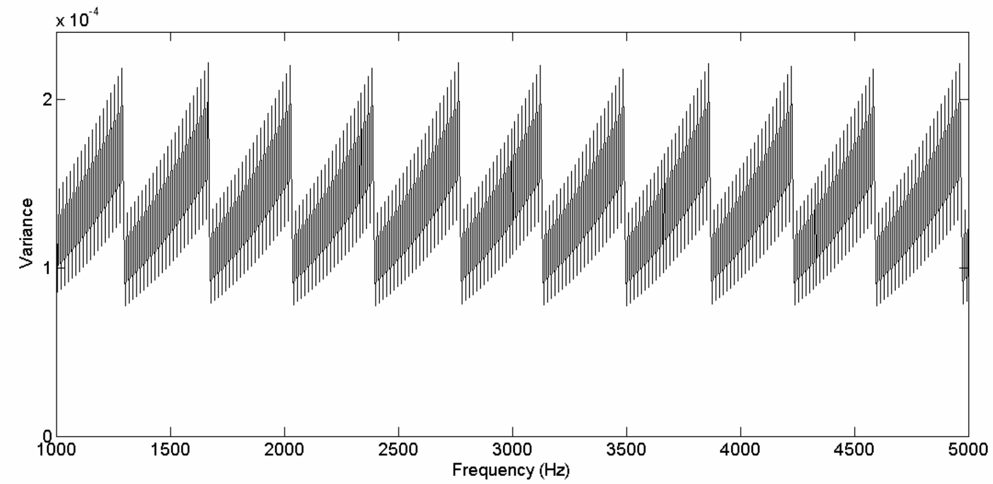


Figure 4.44: Relationship between variance and frequency for a non-stationary Hann-windowed sinusoid (ΔA = 48 dB, Δf = 130 Hz). The window length is 1025 samples, the zero-phase DFT size is 8192 samples.

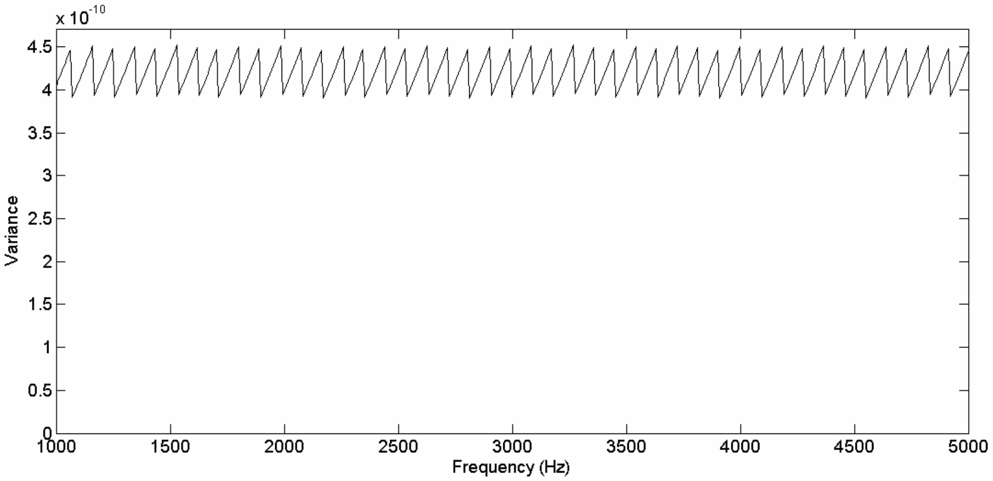


Figure 4.45: Relationship between variance and frequency for a non-stationary Hann-windowed sinusoid (ΔA = 48 dB, Δf = 130 Hz). The window length is 1025 samples, the zero-phase DFT size is 65,536 samples.

In both cases the variance is affected by fluctuations at two different rates (although this harder to see in figure 4.45). The rate of the faster fluctuations is related to the width of the analysis bin. The rate of the slower fluctuations is related to the zero-padding factor. An analytic method of predicting the variance behaviour has not been found and is the subject for future work. Despite this a comparison of the actual and predicted variance for a peak may assist in the identification of non-sinusoids by setting the variance difference threshold, V in (3), so that it is higher than v for all possible sinusoids, e.g. those with ΔA in the range 0-96 dB and those with Δf in the range 0-260 Hz. Repeated tests with sinusoids of different $\overline{f}$

across this range of values did not produce a value of *v* higher than 0.006 and this is used as the value of *V*. This method is tested as follows:

1. An FFT is taken of a zero-phase zero-padded windowed frame (the length of the audio data is 1025 samples and the zero-padded length is 8912 samples).

2. The magnitude spectrum of the FFT is searched for peaks. These are taken as being bins in which the magnitude is higher than that of the four closest neighbours on either side.

3. The reassigned frequency $\left(\bar{f}_{\text{amp}}\right)$ for each peak bin is compared with the bin number to see whether the bin is contaminated by a component from an outlying bin or whether it is due to a component within the actual frequency range of the bin. If the bin is contaminated then the component in this bin is rejected as a candidate sinusoid. As discussed in section 4.4.1 the range of frequency values takes account of the effects of amplitude and frequency change on the reliability of the estimate.

4. Finally, RDA is applied to the remaining candidates and the actual and expected variances are compared as described above. Where the variance difference *v* is higher than the threshold *V* that component is rejected as a candidate sinusoid.

When this test is run with a single stationary sinusoid at 1 kHz only one candidate remains at stage 4 and it has an estimated frequency of 1000.0 Hz. The variance difference is 5.27×10^{-23} which is well below the proposed threshold of 0.006. 507 peaks are identified at step 2 but 506 are rejected as being due to contamination at step 3. Figure 4.46 shows the variance difference, *v*, against the estimated frequency for each component that is still considered a candidate after step 3 for a single frame comprising a stationary 1 kHz sinusoid and a full band white Gaussian noise signal. Both of these signals have been normalised to have the same root mean square (RMS) value. The figure shows the threshold as a dotted line and a log scale has been used for the vertical axis to accommodate the large range of *v*. The component due to the sinusoid has been circled. Here 173 peaks are identified of which 83 are rejected as contaminated. At step 4 10 peaks are retained as sinusoidal candidates and 73 are rejected. Although the peak in the frequency region of the sinusoid does not have the lowest variance difference it is still comfortably below the threshold. Further analysis of performance in noise is deferred until a comparison of methods in section 4.7.

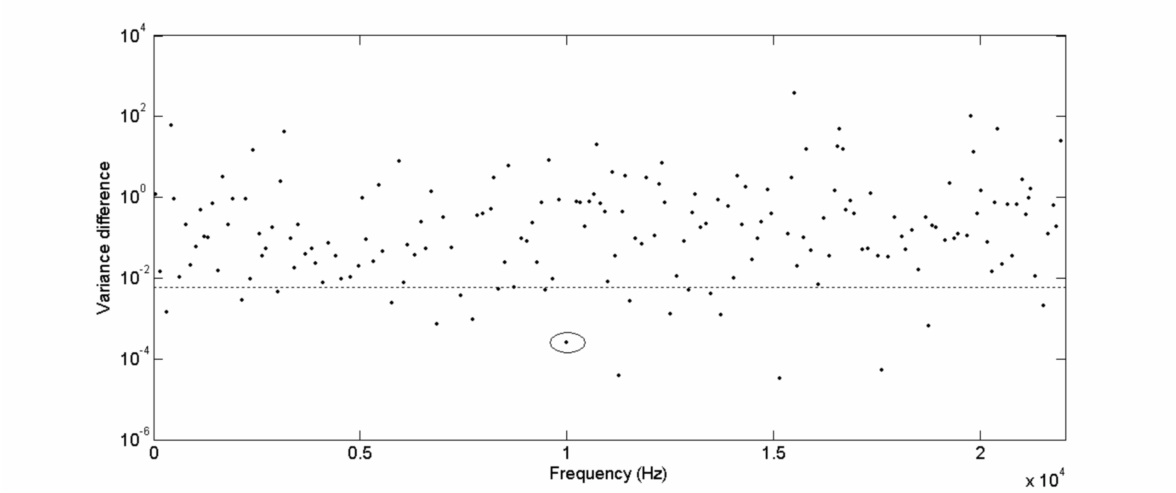


Figure 4.46: Variance difference versus estimated frequency of sinusoidal candidate components (RMS level of sinusoid and that of the broad band noise are the same).

4.5.3 Performance for combination tones

Additive interference caused in multi-component signals is another important aspect of performance; how do clusters of sinusoidal components affect their individual categorisation? To examine this seven stationary sinusoids of the same level are combined in a single frame and this method is considered for linear frequency spacings of 50,100, 200, 500 and 1000 Hz between each sinusoid (the frequency of the lowest sinusoid is 1 kHz). Results for this test are presented in table 4.1.

	50 Hz	100 Hz	200 Hz	500 Hz	1000 Hz
Peaks identified at step 2	0	499	485	484	488
Peaks rejected at step 3	0	492	478	477	481
Peaks below threshold (*V*) at step 4	0	3	7	7	7
Peaks above threshold (*V*) at step 4	7	4	0	0	0
Mean of variance difference (*v*) at step 4	4.67	0.0541	1.96×10^{-5}	8.28×10^{-7}	8.33×10^{-8}

Table 4.1: Performance of variance difference method for a seven sinusoid composite signal with different frequency spacings.

For spacings of 200, 500 and 1000 Hz the method performs correctly in that it retains the sinusoidal peaks identified at step 3. However at a spacing of 100 Hz only 3 peaks identified at step 3 are retained at step 4 demonstrating that for closely clustered sinusoids the proposed method may well erroneously reject sinusoidal peaks that have been correctly identified by previous stages. At this spacing the mean of the variance difference for each of the seven peaks is much higher than the threshold value. At a spacing of 50 Hz, whilst there are 7 candidates the variance difference for each is above the threshold and so all are rejected as sinusoids indicating that the system will give rise to false negatives at very close spacings of such components. As discussed in the previous chapter, for a spacing this narrow a frame length of 1025 samples is too small to achieve adequate separation of components (50 Hz is slightly higher than the width of one non-zero padded analysis bin yet the main lobe of a Hann window is 4 bins wide). Figure 4.47 illustrates how the mean variance difference of peaks retained after step 3 varies with the spacing of these seven sinusoids. The mean is of the lowest 7, or less if less peaks remain after step 3, values of *v*.

For comparison figure 4.48 shows the relationship between these two parameters when the time reassignment offsets are modelled by a first, as opposed to a second order, polynomial. Here *V* is set such that *v* does not exceed it for any pure sinusoids within the same range of ΔA and Δf values (0-96 dB and 0-260 Hz). In this case *V* is set to 2.3. It can be seen that using a second order polynomial allows narrower spacing between components than using a first order polynomial.

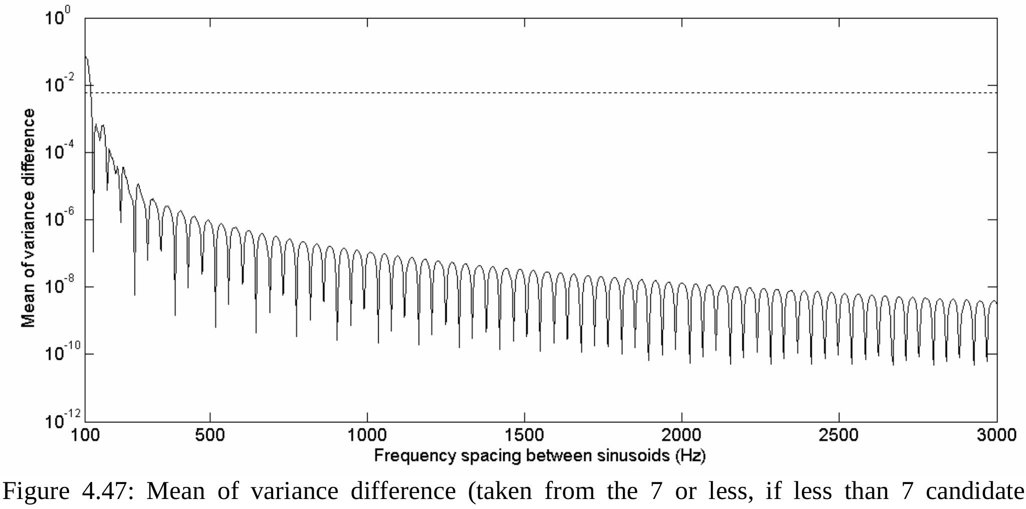


Figure 4.47: Mean of variance difference (taken from the 7 or less, if less than 7 candidates are available, components with the lowest variance difference). A second order polynomial has been used to model the time reassignment offset data.

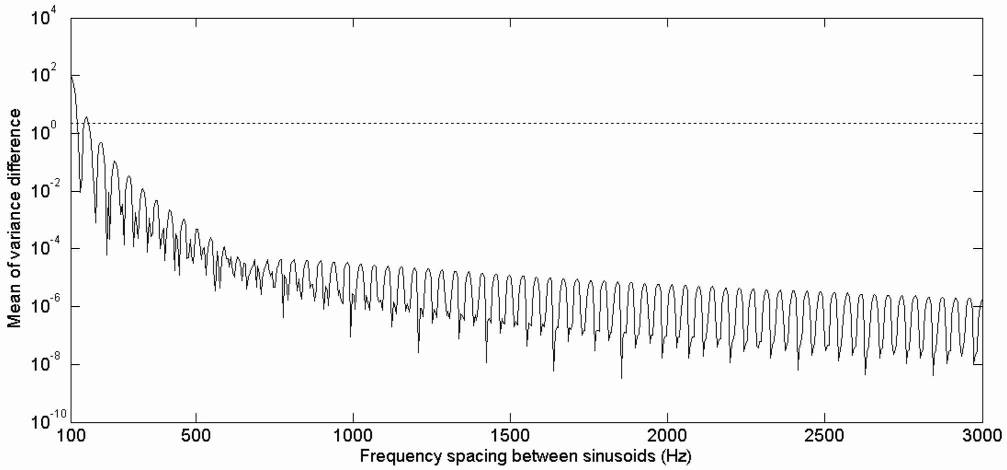


Figure 4.48: Mean of variance difference (taken from the 7 or less, if less than 7 candidates are available, components with the lowest variance difference). A first order polynomial has been used to model the time reassignment offset data.

Figures 4.49 and 4.50 illustrate the performance for non-stationary sinusoids when using a second-order polynomial. For figure 4.49 a seven sinusoid cluster is used with each sinusoid having the parameters ΔA=96 dB and Δf = 0 Hz (grey line) and ΔA= 0 dB and Δf = 0 Hz (black line). In figure 4.50 each of seven sinusoids has the parameters ΔA=96 dB and Δf = 260 Hz. Further analysis and discussion of this measure of sinusoidality is given later in this chapter, by means of comparison with the performance of an alternative measure described in the next section.

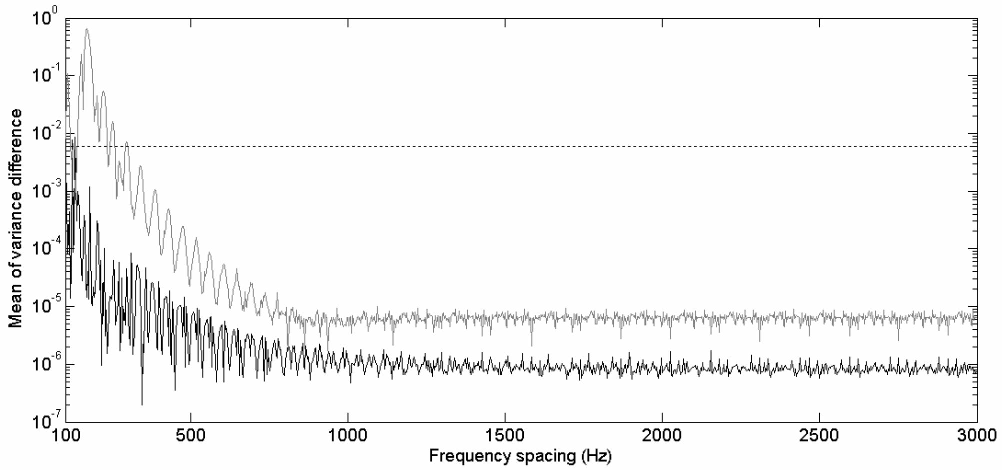


Figure 4.49: Mean of variance difference (taken from the 7 or less, if less than 7 candidates are available, components with the lowest variance difference). A second order polynomial has been used to model the time reassignment offset data. The grey (upper) line represents sinusoids whose amplitude changes by 96 dB during the frame but whose frequency remains stationary. The black line represents sinusoids whose frequency changes by 260 Hz during the frame but whose amplitude remains stationary.

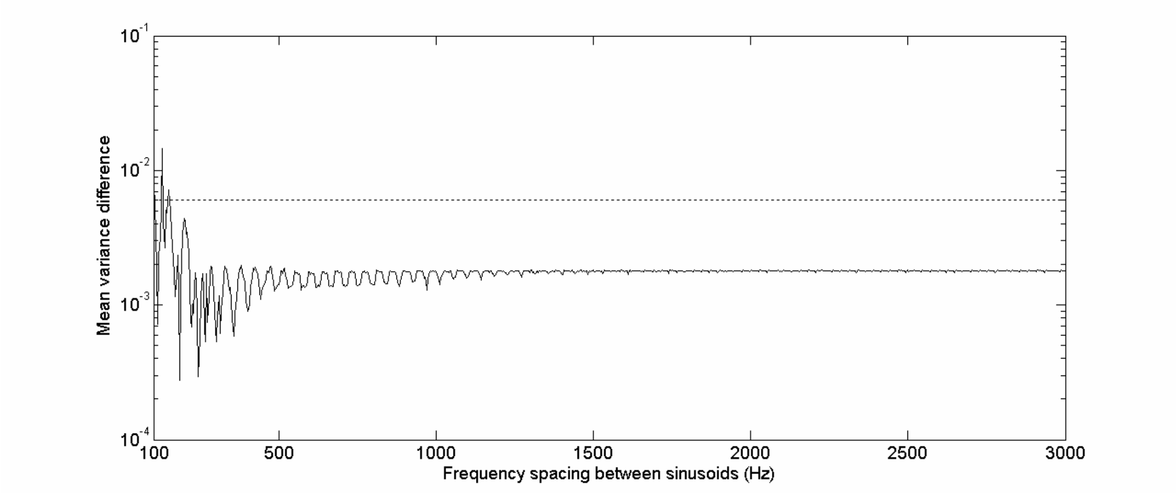


Figure 4.50: Mean of variance difference (taken from the 7 or less, if less than 7 candidates are available, components with the lowest variance difference). A second order polynomial has been used to model the time reassignment offset data. The amplitude of the sinusoids changes by 96 dB and the frequency by 260 Hz throughout the frame.

4.6 Differences in phase and magnitude reassignment estimates of parameters as a measure of sinusoidality

As discussed in section 3.10, the reassignment described in [Auger and Flandrin, 1995] and commonly encountered in the literature uses partial derivatives of the phase of the STFT with respect to frequency (for time reassignment) and time (for frequency reassignment). Alternative reassignment measures have been proposed which use the partial derivatives of the magnitude of the STFT with respect to frequency (for frequency reassignment) and time (for time reassignment). The authors refer to this alternative method of reassignment as amplitude reassignment (referred to here as magnitude reassignment) and to the traditional method as phase reassignment [Hainsworth and Macleod, 2003b]. The authors observe that frequency estimates from each reassignment method only concur in the presence of a stable sinusoid. This section investigates whether the differences between magnitude and phase reassignment estimates can be used as an indicator of the sinusoidality of a component.

Equations (3.134) and (3.135) given in the previous chapter are quoted in the literature for the continuous case. In practice the scalings applied to the imaginary and real parts of the equations have had to be adapted empirically for the discrete case of an 8 times zero-padded FFT of a 1025 point window. The differences between magnitude and phase reassignment of frequency were considered first. As for the consideration of the expected time reassignment offset variance in section 4.5, if the differences in phase and magnitude reassignment can be predicted for stationary and non-stationary sinusoids then these can be compared with the observed differences for a component in order to determine whether or not that component is a sinusoid. Figures 4.51 and 4.52 show the modulus of the frequency reassignment difference (FRD) against ΔA and Δf for a sinusoid with $\overline{f}$ of 1 kHz and 10 kHz respectively. The functions in each of these figures are characterised by ridges whose position is dependent upon the value of $\overline{f}$. Figures 4.53 and 4.54 show the modulus of the time reassignment difference (TRD) against ΔA and Δf, again for a sinusoid with $\overline{f}$ of 1 kHz and 10 kHz respectively. These functions are smoother than those for FRD and there is less difference between them for different values of $\overline{f}$. For these reasons the TRD, as opposed to the FRD, is investigated here as a measure of sinusoidality.

As for the variance test described in the previous section the proposed sinusoidality test uses a 100 x 100 element array from which the expected TRD can be estimated using 2D linear interpolation. If the expected TRD differs from the actual TRD by more than a specified threshold then the component is considered to not be a sinusoid; i.e. if:

$$\left|TRD_{\text{expected}} - TRD_{\text{measured}}\right| = u > U \qquad (4.15)$$

where *u* denotes the modulus of the TRD and *U* denotes the threshold, then the component is not a sinusoid. Figures 4.55 and 4.56 show how *u* varies with ΔA and Δf for two different values of $\overline{f}$. As for figures 4.42 and 4.43 in section 4.5.2, 150 x 150 points are sampled to test interpolation of the TRD_{expected} array and estimates of ΔA and Δf are produced by the algorithm described in section 4.3. Repeated tests at different values of $\overline{f}$ do not yield a value of *u* higher than 8.0 and this is the threshold value, *U*, that is adopted in subsequent tests of this method.

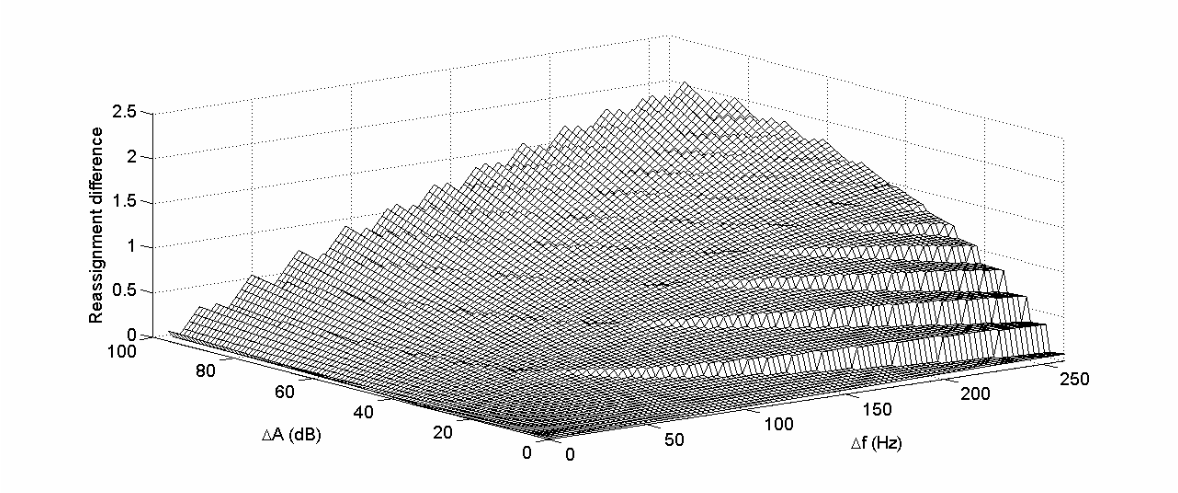


Figure 4.51: Modulus of the difference between phase and amplitude reassignment offsets for frequency for a sinusoid of 1 kHz with the non-stationary parameters ΔA and Δf .

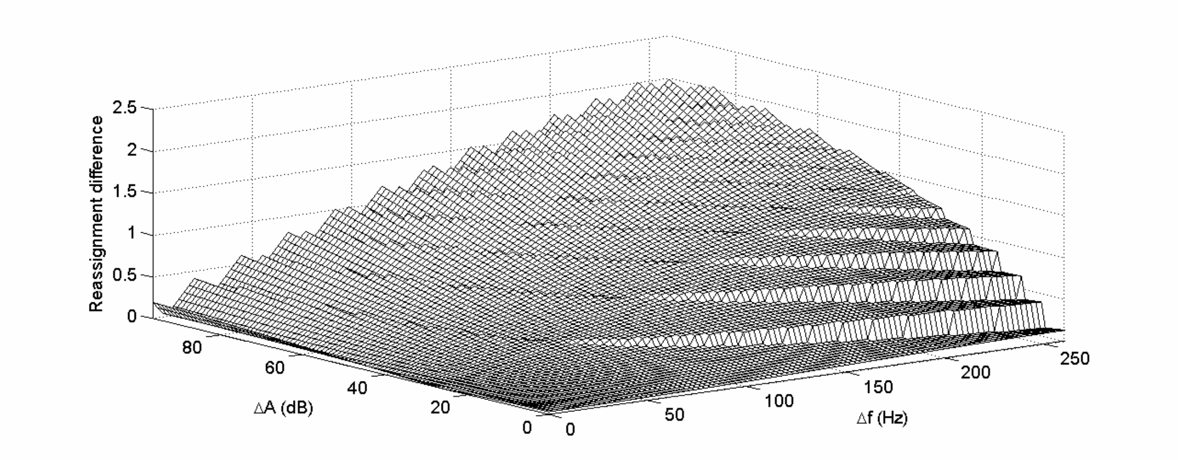


Figure 4.52: Modulus of the difference between phase and amplitude reassignment offsets for frequency for a sinusoid of 10 kHz with the non-stationary parameters ΔA and Δf .

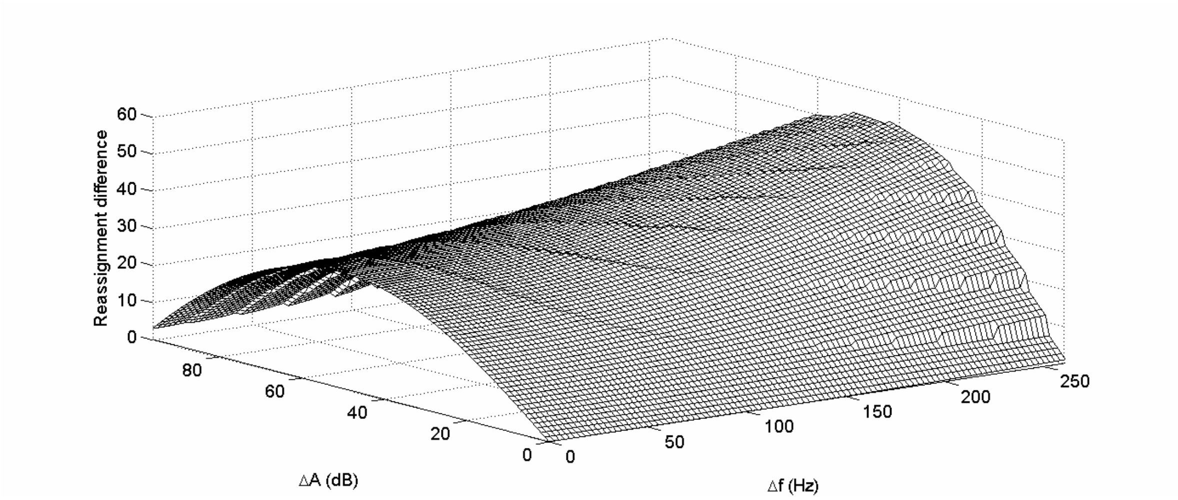


Figure 4.53: Modulus of the difference between phase and amplitude reassignment offsets for time for a sinusoid of 1 kHz with the non-stationary parameters ΔA and Δf .

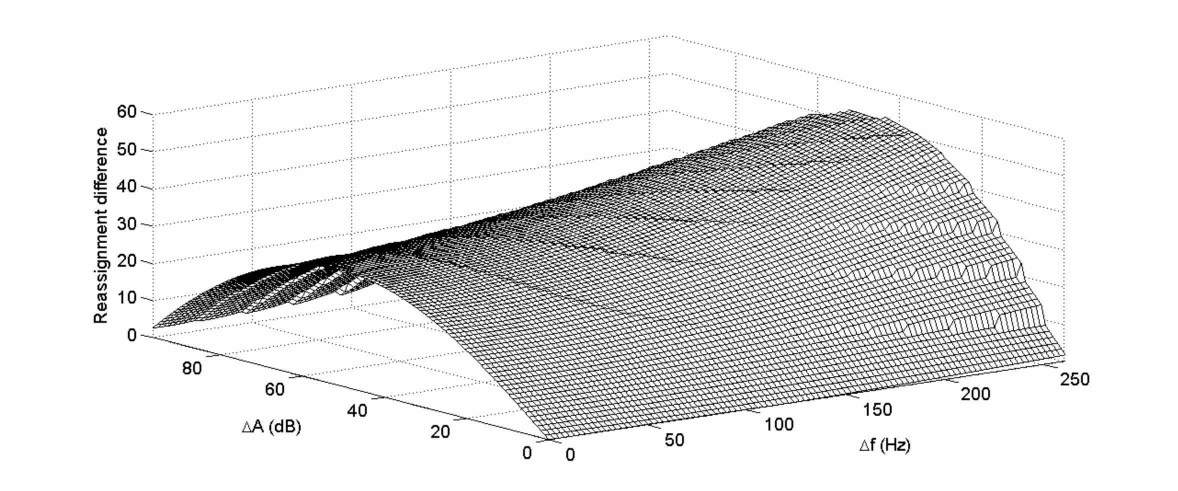


Figure 4.54: Modulus of the difference between phase and amplitude reassignment offsets for time for a sinusoid of 10 kHz with the non-stationary parameters ΔA and Δf .

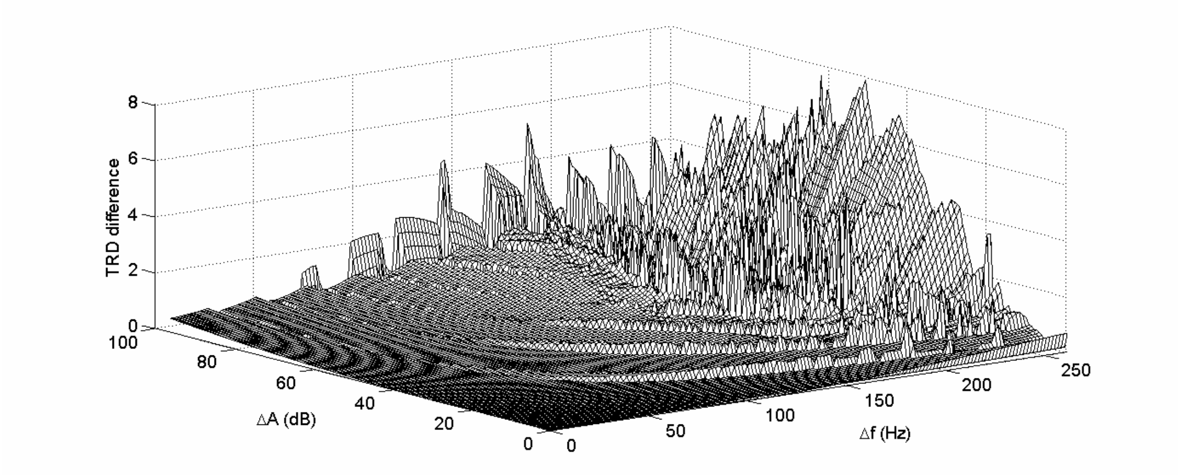


Figure 4.55: Modulus of the difference between expected and actual TRD for a sinusoid of 1 kHz with the non-stationary parameters ΔA and Δf .

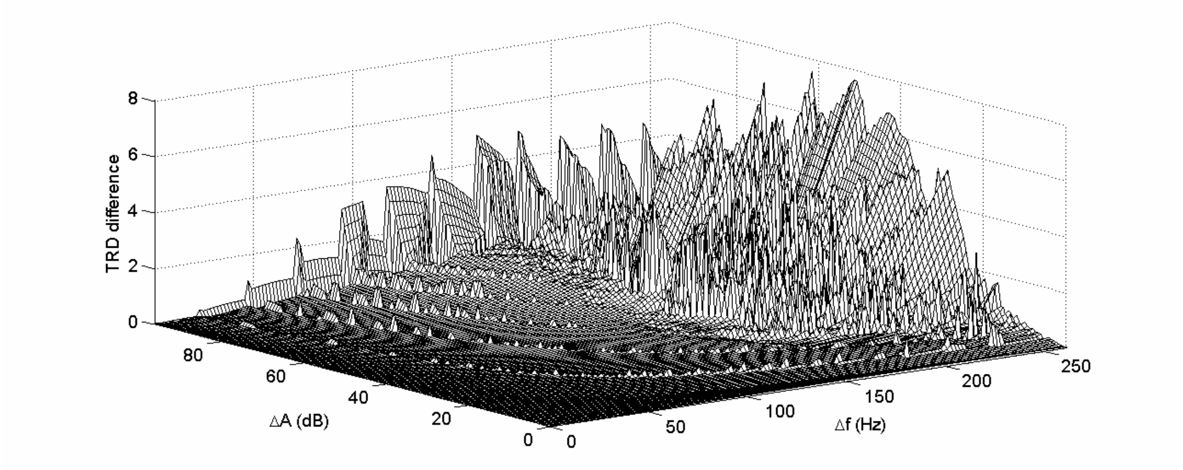


Figure 4.56: Modulus of the difference between expected and actual TRD for a sinusoid of 10 kHz with the non-stationary parameters ΔA and Δf .

This measure of sinusoidality has been tested in the same four-step algorithm described previously. Figure 4.57 shows *u* for those peaks retained after step 3 of this algorithm for the identical signal frame used to produce figure 4.46. The peak due to the sinusoid has been circled and the threshold value, *U*, is shown as the dotted line. As for the variance difference, the performance of the TRD difference sinusoidality measure is tested for sinusoidal clusters with different frequency spacings between individual components. Table 4.2 gives data equivalent to that in table 4.1 (section 4.5.3) for this method. It can be seen from this table that he cluster performance is worse than that for the variance difference method since all peaks are rejected at step 4 for a spacing of 100 Hz. For comparison figures 4.58 to 4.60 present equivalent data to that in figures 4.47 and 4.49.

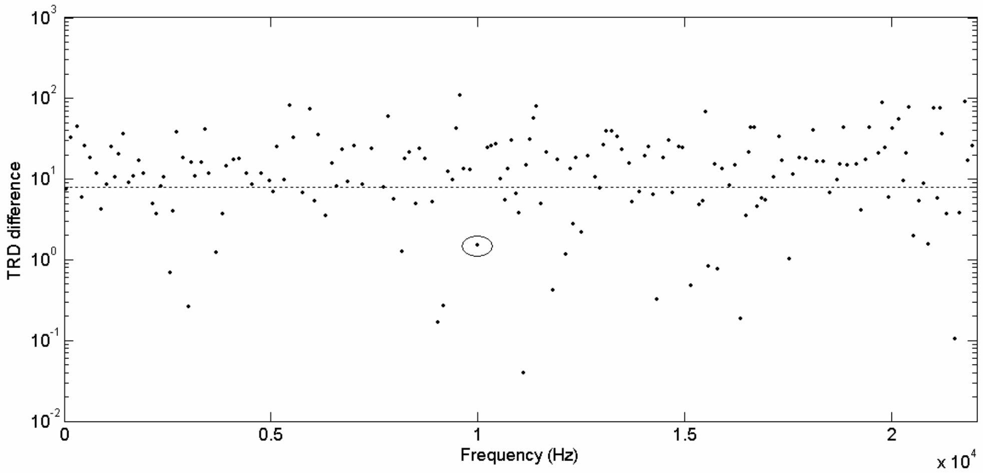


Figure 4.57: TRD difference versus estimated frequency of sinusoidal candidate components (RMS level of sinusoid and that of the broad band noise are the same).

	50 Hz	100 Hz	200 Hz	500 Hz	1000 Hz
Peaks identified at step 2	0	499	485	484	488
Peaks rejected at step 3	0	492	478	477	481
Peaks below threshold (*V*) at step 4	0	0	7	7	7
Peaks above threshold (*V*) at step 4	7	7	0	0	0
Mean of variance difference (*v*) at step 4	25.2	19.39	2.75	0.572	0.114

Table 4.2: Performance of TRD difference method for a seven sinusoid composite signal with different frequency spacings.

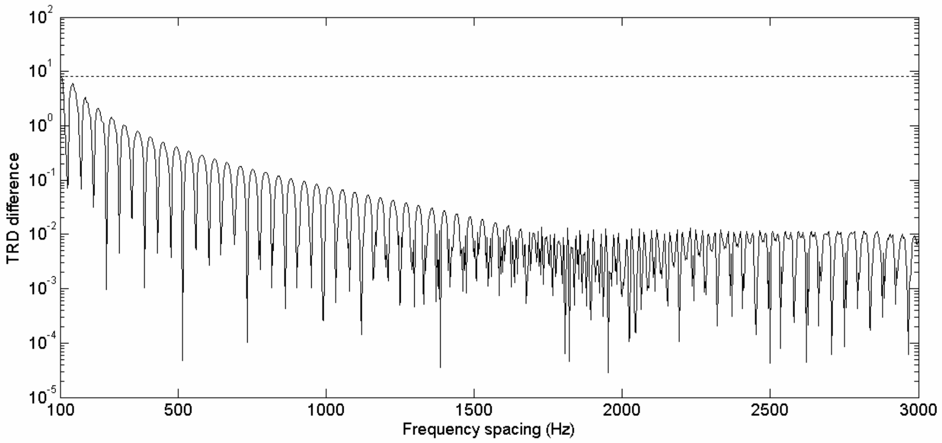


Figure 4.58: Mean of TRD difference (taken from the 7 or less, if less than 7 candidates are available, components with the lowest variance difference).

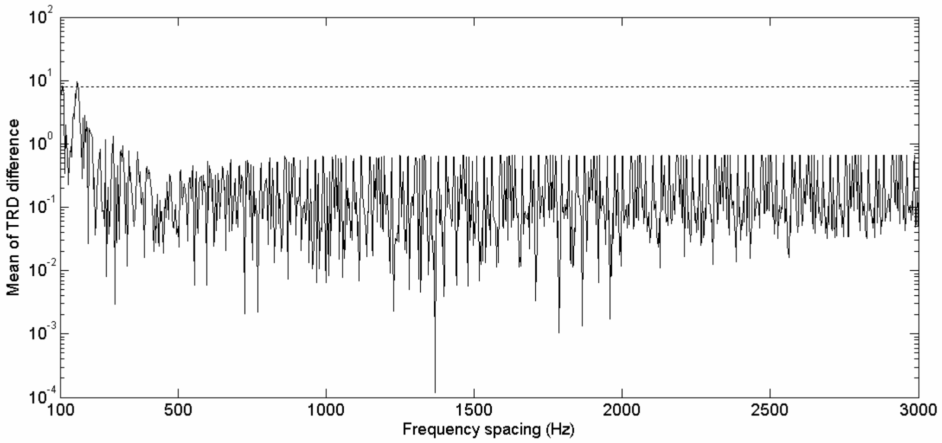


Figure 4.59: Mean of TRD difference (taken from the 7 or less, if less than 7 candidates are available, components with the lowest variance difference). The amplitude of the sinusoid is stationary, the frequency changes by 260 Hz during the frame.

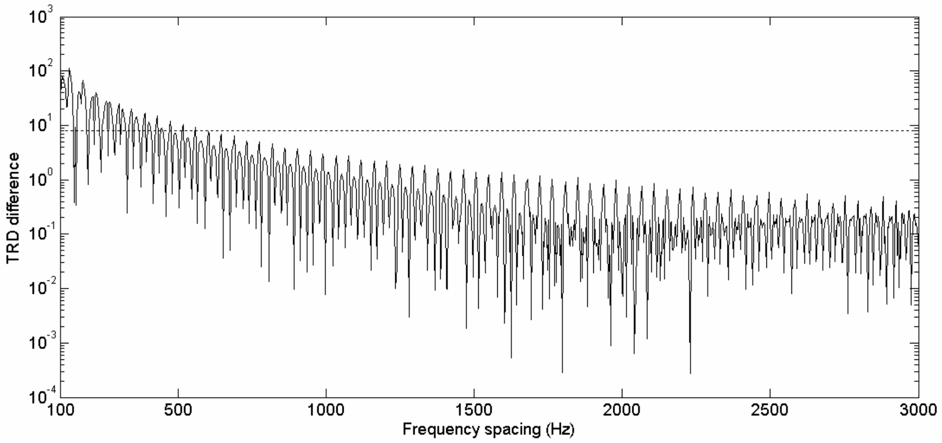


Figure 4.60: Mean of TRD difference (taken from the 7 or less, if less than 7 candidates are available, components with the lowest variance difference). The frequency of the sinusoid is stationary, the amplitude changes by 96 dB during the frame.

4.7 Comparison of methods for the identification of non-stationary sinusoids

This section compares the two methods for non-stationary sinusoidal identification described in the previous two sections with the correlation method described in section 3.6.5 and [Lagrange et al, 2002]. Although, as has been discussed, the correlation method comes at high computational expense making it an unlikely candidate for real-time applications it is used for comparison here since it does offer sinusoidal identification on a ‘single frame’ basis.

The correlation method has previously been evaluated for relatively modest frequency and amplitude changes on a ‘highest score’ basis: for each frame a certain proportion (10%) of the spectral peaks with the highest correlation measure are considered to be sinusoidal [Lagrange et al, 2002], [Lagrange, 2004]. The methods proposed previously in this chapter have used a different approach: any spectral peak whose tested parameter (either variance or TRD difference) is within the range that might be expected for a single sinusoid, not in the presence of noise, whose ΔA and Δf values do not exceed 96 dB and 260 Hz respectively, is classified as a sinusoid. The correlation method is adapted here to match this approach. Once these non-stationary parameters have been estimated then $\overline{A}$ and $\overline{f}$ are estimated, a sinusoid with these estimated parameters is synthesized whose spectrum can be compared with that of the actual spectral peak. Where the sinusoid is highly non-stationary, particularly where there is a large amplitude change, the correlation measure can be significantly reduced. To prevent this a two dimensional array (100 x 100 elements) of ‘expected correlation’ is used to normalise the correlation measure. Therefore the correlation measure used here is:

$$\Gamma_{peak} = \frac{\left| \sum_{f_{peak}+B}^{f_{peak}+B} \frac{H(f)}{\left|H(f)\right|}.W(f) \right|_{\text{actual}}}{\left| \sum_{f_{peak}+B}^{f_{peak}+B} \frac{H(f)}{\left|H(f)\right|}.W(f) \right|_{\text{expected}}} \qquad (4.16)$$

where B is the number of zero-padded bins taken either side of the peak (32 in this case), $H(f)$ is the actual spectrum and $W(f)$ is the spectrum of the sinusoid synthesized from the estimates with an amplitude $\left(\overline{A}\right)$ of 1. A plot of the array of expected values is shown in figure 4.61.

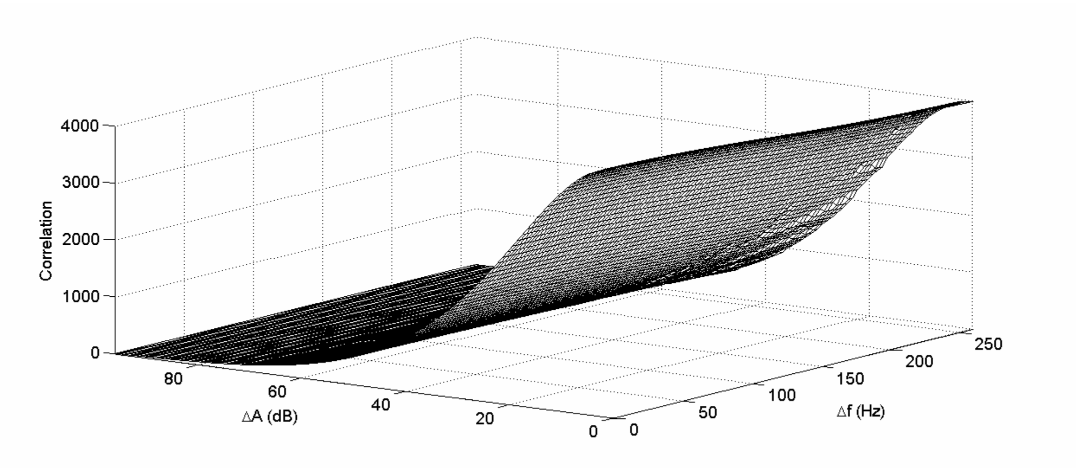


Figure 4.61: Correlation versus ΔA and Δf for a sinusoid with mean frequency 10 kHz.

This array of values was then tested with a sinusoid with a different mean frequency and the minimum and maximum values of Γ_{peak} taken as being the limits within which a sinusoid would reside. Figure 4.62 shows the values of Γ_{peak} obtained. The range of values is 0.989 to 1.434.

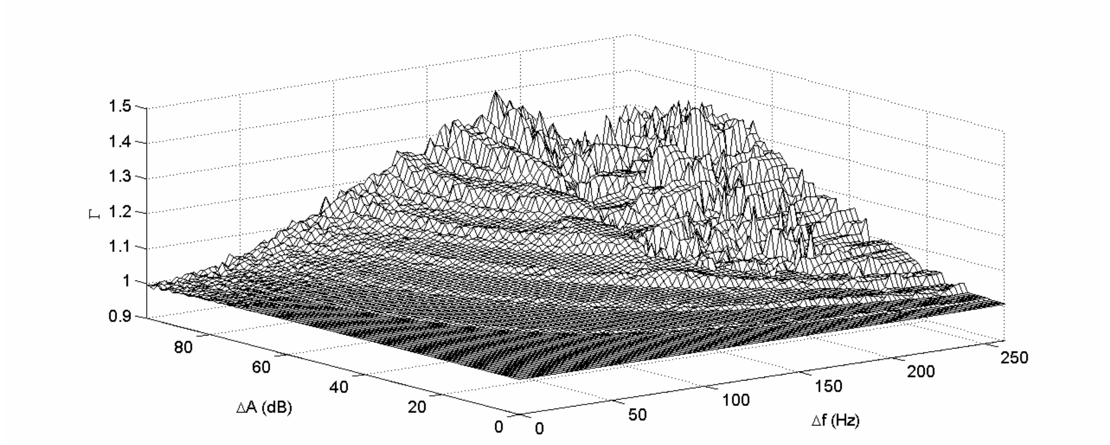


Figure 4.62: Range of correlation values versus ΔA and Δf for a sinusoid with mean frequency 1

In terms of the cost per candidate, the variance difference method requires 15 multiply operations and the TRD method requires 2 complex divides and 2 multiply operations. The correlation method requires an FFT to be performed (106 496 complex multiply and adds for a 8192 point FFT) in addition to 64 multiply and add and 2 divide operations (for the correlation calculation given by (41.6). Clearly the former methods which are proposed here are much cheaper than the correlation test. All three methods require one interpolated 2D lookup operation per candidate.

Each of these three methods of sinusoidal identification is tested with stationary and non-stationary sinusoids embedded in different levels of broad band Gaussian white noise. The four step procedure described in section 4.5.2 is used for each method. The first three steps are identical (identify magnitude peaks and reject those due to bin contamination) and the fourth step uses whichever method is being tested (variance difference, TRD difference or normalised correlation) to reject further peaks. Each time the test is run the percentage of peaks rejected is recorded along with whether any of these is the sinusoidal peak. Each test is run 100 times at each noise level and with different values of ΔA and Δf . Figure 4.63 shows the mean percentage of peaks rejected for each test. Figure 4.64 shows the mean percentage of times that the actual sinusoidal peak is correctly retained.

The variance method clearly performs best where the relative noise level is low (-60 and -40 dB) since there are less false negatives and this method rejects more peaks than the others whatever the noise level. However where the relative noise is higher the variance method performs less well than the others, particularly where ΔA and Δf are high.

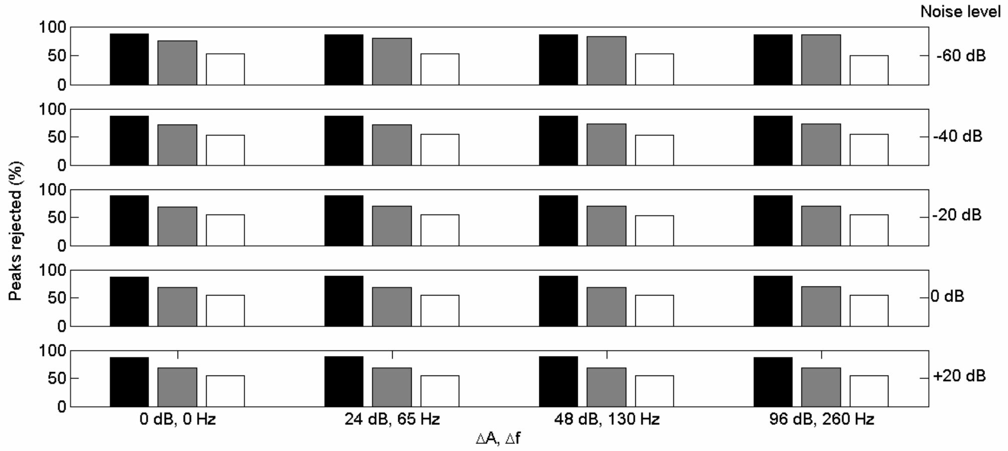


Figure 4.63: Non-contaminated peak rejection for variance difference (black), TRD difference (grey) and correlation (white) measures. At each relative noise level 100 frames were created each containing a single sinusoid (randomly selected mean frequency). The noise level refers to the RMS level of the noise relative to the RMS level of the sinusoid.

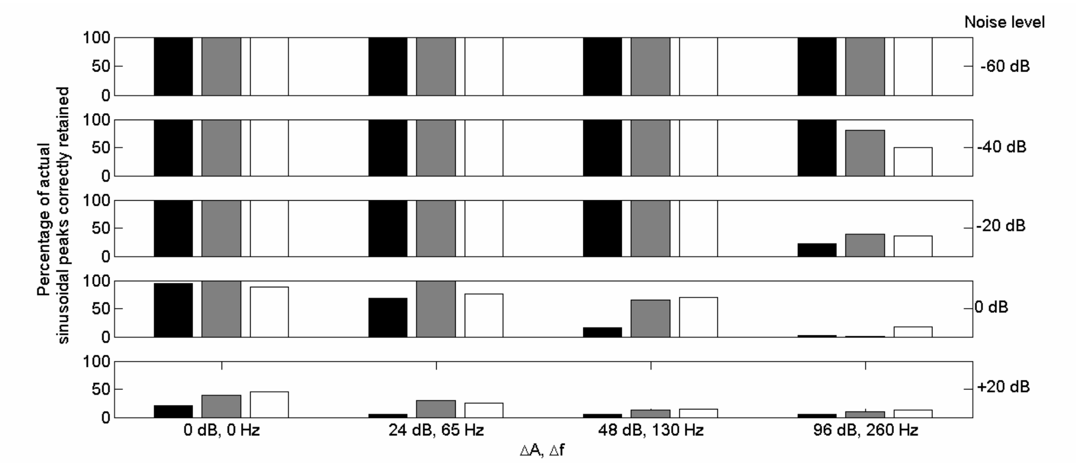


Figure 4.64: Percentage of correctly retained peaks for variance difference (black), TRD difference (grey) and correlation (white) measures. At each relative noise level 100 frames were created each containing a single sinusoid (randomly selected mean frequency). The noise level refers to the RMS level of the noise relative to the RMS level of the sinusoid.

The variance method consistently rejects around 87% of peaks, the TRD difference method has a mean rejection across these tests of 72% but this performance drops from 81% at -60 dB noise to 69% at +20 dB noise. The correlation method only rejects 54% of peaks but its relative performance in terms of false negatives is better than the other two methods at 0 dB and +20 dB. For real-time applications where it is not possible to implement the correlation method a straightforward deduction from these results is that the variance method works best where the stochastic component is relatively low in level and the TRD difference method works best where it is relatively high in level. This suggests that an adaptive analysis system which selects its sinusoidality test method for each frame on the basis of the level of coherence in the time domain input would perform best. Development of such a test is a subject for future investigation. It should also be remembered that the variance method performs better where there is close clustering of sinusoids, as demonstrated in the previous section. It is for this reason that the variance method is employed in the analysis-resynthesis system described in chapter 6.

Considering the general application of such tests of sinusoidality, the specifics of the implementation and application must also be considered. A low rate of rejection for non-sinusoidal peaks will lead to the synthesis algorithm using sinusoids to synthesize noise which will be computationally inefficient, counter-intuitive and possibly reduce the quality of creative transformations. For example, where pitch shifting is to be performed on the sinusoidal part only the appropriation of noise components into this sinusoidal part of the signal may well produce a perceptually undesirable result. A high rate of false negatives, which would leave sinusoidal components in the residual part of the analysis, may compromise synthesis of this residual by causing excessive narrowing of the filters used to shape the noise. In addition, if on-setting and off-setting sinusoids (e.g. those with a high degree of amplitude change) are not classified as such then their synthesized counterparts will start late and finish early. The thresholds that have been adopted here for each of these methods have been determined in order to include all sinusoids with non-stationary parameters in the ranges previously stated. These thresholds could be adjusted by the user or adapted to the signal itself and, again, this would be application and implementation specific.

4.8 Extension to arbitrary frame lengths and sample rates

For clarity the techniques in this chapter have been presented in the context of a sample rate of 44.1 kHz and a frame length of 1025 samples. Extension to the general case of arbitrary sample rate is achieved by the operation of scaling the *f*, Δf and ΔA estimates by the ratio of the sample rate to 44.1 kHz.

Where different frame lengths are used the scaling operation is more complex. For PDA the ΔA estimates remain the same regardless of frame length since the phase distortion measurements are a function of the change in amplitude in dB per frame. The Δf estimates (in Hz) require scaling by the ratio of frame lengths since they are function of ‘bins per frame’ [Masri, 1996]. For RDA the *c* measure must be scaled by the ratio of frame lengths and the *m* measure by the square of this ratio. This is because the measured time reassignment offset for a given non-stationarity is scaled by the frame length and *c* is directly related to this. RDA uses frequency in Hz rather than bins for the *x* axis of its polynomial and since *m* is a measure of ‘*y* by *x*’ this scaling must be applied twice to this value. The variance must also be scaled by the square of the frame length ratio since it is the square of sum of errors in the estimation of *y*. Once the actual Δf has been obtained it must itself be scaled once by the frame length ratio since it is measured in Hz per frame rather than bins per frame.

As an example of this scaling the following steps adapt the analysis presented in this chapter to the frame length = 513 samples case:

1. Obtain *m*, *c* and variance estimates for each component.

2. Multiply *c* by $^{1025}\!/_{513}$.

3. Multiply *m* and variance by $\left({}^{1025}\!/_{513}\right)^2$

4. Reject non-sinusoids and obtain Δf and ΔA estimates

5. Multiply Δf estimate by ${}^{1025}\!/_{513}$.

4.9 Conclusions

This chapter has described methods, using reassignment data, for the identification of sinusoidal components in a single DFT frame and for their description in terms of mean amplitude, mean frequency, linear frequency change and exponential amplitude change. A modification of the phase distortion analysis of Masri, reassignment distortion analysis, has been presented which uses a higher order polynomial to produce more reliable, and more frequency independent, estimates of ΔA and Δf. The range of values of ΔA and Δf considered in previous literature with this method has been extended here and the interaction between the reassignment distortion measures for ΔA and Δf (which also occurs in phase distortion) has been taken account of in a 2D interpolation system which uses array look-up to produce greatly improved estimates where there is a high degree of non-stationarity. An analytic solution to removing the bias in $\overline{f}$ in the presence of amplitude and frequency change has been presented as has an approach to correcting the bias in estimates of $\overline{A}$ with an adaptation of the analytic solution to amplitude correction for window shape to improve its performance in the presence of non-stationarity. Two methods for using time reassignment data for determining whether or not a non-contaminated spectral peak is due to a sinusoid have also been proposed, tested and compared to an existing method for single frame identification of sinusoids. The variance difference and TRD methods proposed certainly offer comparable performance to the existing correlation method without the need to synthesize a sinusoid and perform an FFT for each magnitude peak considered. The variance difference out-performs the TRD method where there is low noise or close clustering of sinusoids but is inferior in the presence of relatively high levels of noise.

At the time of writing this thesis new work has been published that offers an analytic solution to the derivation of the four sinusoidal parameters for a Gaussian window and an adaptation of this for commonly encountered orthogonal windows such as the Hann and Hamming [Abe and Smith, 2005]. This author has verified the method for the Gaussian window, which is similar to that presented in [Peeters and Rodet, 1999] however the Hann adapted method produces very poor estimates for all parameters compared to the method proposed in this thesis. This may be due to the fact that the multiple regression analysis used to derive values for the adaptation from Gaussian to Hann window was only used over modest ranges of sinusoidal parameter values (maximum ΔA of 2 dB and Δf of 23 Hz per frame). Therefore this method is not suitable, in its present form, to analysis of the range of sinusoidal signals discussed in this chapter. Useful further work in this area of spectral analysis would be to perform new multiple regression analysis for this adapted method over a much greater parameter range so that a useful comparison between the adapted method and the one described and investigated in this chapter can be made.

5 MODELLING AN AUDIO SIGNAL AS NOISE FILTERED BY A BANK OF TIME VARIANT PARAMETRIC EQUALISERS

5.1 Introduction

This thesis investigates how time-frequency and time-scale analysis techniques can be used to produce real-time spectral models of audio signals. The generic spectral model used is the ‘sinusoids + noise’ (or ‘deterministic + residual’) approach that began, and is most commonly associated, with SMS [Serra, 1989]. The previous chapter detailed methods for using data from a single frame of a reassigned DFT to identify and numerically describe the sinusoidal components it contains. Once this deterministic part of the frame is known the residual part, which is assumed to be noisy or stochastic since it contains no components that are well modelled by non-stationary sinusoids, can be derived by subtracting the sinusoidal part from the whole signal. This subtraction can be performed in either the time or frequency domains and is discussed in the next chapter.

This chapter describes how the residual signal can be modelled as a broad band noise source which is shaped by a bank of time-variant equalisers. All three equaliser parameters are time-variant: centre frequency, gain and quality factor (Q). The parameters for these filters are derived from the complex wavelet transform. To reduce aliasing and shift variance but retain low computational cost a combination of the undecimated and decimated wavelet transform, the partially decimated wavelet transform, is described and evaluated. The ‘frequency splitting’ technique [Daubechies, 1992] is used to determine the Q of each filter. The intention is that this wavelet-orientated approach offers an intuitive and efficient means of describing ‘non-sinusoidal’ components such as those produced by long term random and impulsive processes. However, there are some disadvantages to the short-time wavelet analysis proposed here and these are also discussed along with possible solutions.

Although in this context the analysis system described here is intended for frame by frame residual modelling it is hoped that the use of complex B-splines will also have general applications to spectral modelling of audio signals. Therefore the modelling of spectral components using this system is described in a general context and additional aspects of the system not implemented in the frame by frame spectral modelling system described in the next chapter are also surveyed.

Early sections of this chapter describe the partially decimated complex wavelet transform (PDCWT) and how it is used to produce gain and centre frequency trajectories for each of the filters. The use of the splitting trick to derive the Q trajectory is then described. Later sections discuss how these parameters are then matched to those of the real IIR filters which are used to filter the noise source at the synthesis stage.

5.2 The complex B-spline wavelet transform

Spline wavelets are described in detail in section 3.83 of this thesis. They are used here for three reasons. The first is that with increasing spline order the wavelet function tends to the modulated Gaussian of the Morlet wavelet which has constant instantaneous frequency [Flandrin et al, 1995] and has optimal localisation in time and frequency. The second reason is that the time/frequency localisation of the time-scale atoms can be controlled via the order of the spline; a higher order spline is less localised in time but better localised in frequency than a lower order spline and *vice versa*, offering the user control over this important aspect of time-scale analysis. Finally, they are compactly supported meaning that they can be implemented using FIR filters within the wavelet transform without truncation errors. Although these wavelets are not orthogonal this is ameliorated by over-sampling applied in this case and the magnitude correction for frequency described later in this chapter.

5.2.1 Wavelet and scaling functions

The low pass coefficients for the scaling function are given by (3.88), which in the case of the B-spline simplifies to the binomial kernel:

$$h[n] = u_2^m[n]. \quad (5.1)$$

To accommodate both odd and even length spline orders the following causal definition of the binomial kernel (adapted from [Chui and Wang, 1992]) is used here, rather than the symmetric even-only formulation given by (3.92):

$$u_2^m[n] = \begin{cases} \dfrac{1}{2^m}\begin{pmatrix} m+1 \\ n \end{pmatrix}, & 0 \le m \le n+1 \\ 0, & \text{otherwise} \end{cases} \qquad (5.2)$$

The high pass coefficients for the wavelet function are given by (3.89) which, for causal B-splines, simplifies to:

$$g[n] = u_2^m[n] * \left((-1)^m u_2^m[n] \right) * \left((-1)^m b^{2m+1}[n] \right) \qquad (5.3)$$

Where b^{2m+1} is the B-spline of order $2m + 1$. An explicit formula for the B-spline that does not require repeated convolution of step functions is given by (3.86). When investigating high-order splines and evaluating (3.86) with double precision floating point numbers the equation was found to produce incorrect results for large, positive values of t. The cause of this error has not been investigated further but it is thought that it may be due to small numerical differences in the magnitude of terms within the summation that should cancel. These are then magnified by the high exponent in the power function giving the observed error. Although such high order splines are not employed in this thesis for general interest a straightforward solution is given here. This exploits the symmetry of (3.85) and reformulates (3.86) as:

$$\beta^m(x) = \frac{1}{m!} \sum_{k=0}^{m+1} \binom{n+1}{k} (-1)^k \left(-|x| - k + \frac{m+1}{k} \right)_+^m \qquad (5.4)$$

Figure 5.1 shows the difference between the output of the original and modified equations. The range of values for x has been constrained so that the context of the error can be seen alongside the rest of the function. However for values of $x > 10$ the error grows rapidly in size giving a value for the spline function, which should be very close to 0 at this point, in excess of 6×10^4 at $x = 20$.

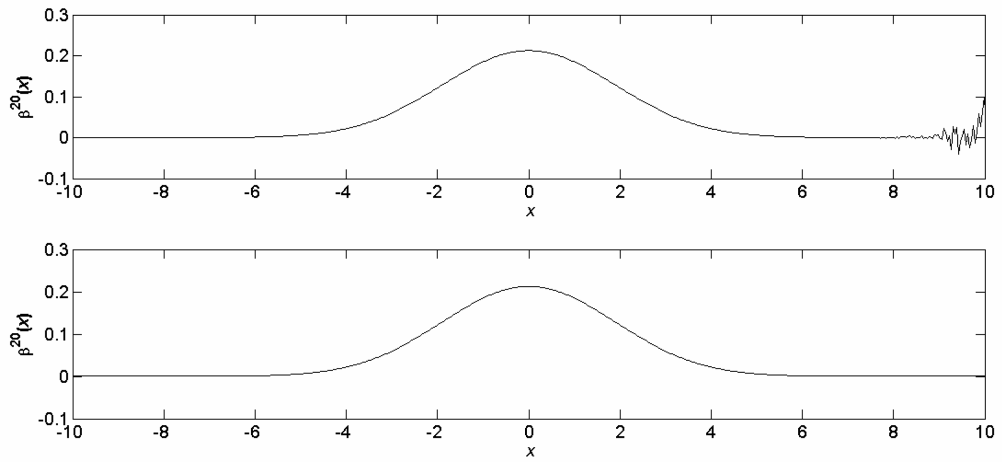


Figure 5.1: Twentieth order B-spline as calculated by equation (3.86) (top) and the modified equation presented here (bottom).

The real and imaginary parts of the wavelet coefficients are found by performing a real B-spline wavelet analysis on the real and imaginary parts of the analytic signal. The discrete analytic signal can be determined in the discrete Fourier domain by:

$$X_{\text{analytic}}\left(k\right)=\begin{cases} X(k), k=0, {}^{N}\!\!\diagup_{2} \\ 2X(k), 1\le k\le\left({}^{N}\!\!\diagup_{2}\right)-1 \\ 0,\left({}^{N}\!\!\diagup_{2}\right)+1\le k\le N-1 \end{cases} \qquad (5.5)$$

There are two reasons for adopting this approach here. Firstly the binomial kernel which forms the low pass filter of the spline wavelet is only defined for integers. Therefore the low pass filter cannot be shifted forward by half a sample which is how complex wavelets are commonly produced [Selesnick, 2001]. Secondly for the specific application described in this thesis the data is already available in the Fourier domain and so the analytic signal can be readily derived. For applications where a time domain implementation is required that uses two different scaling functions to produce the real and imaginary parts of the transform one approach could be to perform the phase shift in the Fourier domain and then take the inverse Fourier transform. However, in the case of the binomial kernel this does not produce an FIR filter and it is the compact support, and the support is very compact at low spline orders, which is one of the advantages of the B-spline analysis.

It has been shown that this method of deriving the analytic signal fails for signals of the form $a,b,a,b,a,b,...$ where a and b are non-zero and an alternative method, the ‘eHilbert’ transform has been proposed [Elfataoui and Mirchandani, 2004]. This method is not adopted here since such signals correspond to continuous signals at the Nyquist limit. Such signals will have been extinguished by the anti-aliasing filter if they originate in the analogue domain and occur outside the range of frequencies that can be resolved by the human auditory system in the specific case of $F_s = 44.1$ kHz adopted in this thesis. However it has been found here that the imaginary output of the discrete Hilbert transform is not perfectly shift invariant and introduces a high frequency artefact which appears in the complex analysis. This is discussed later on in this chapter.

The scaling and wavelet functions of the zeroth, first, third and twentieth order splines are shown in figure 5.2 below. Figure 5.3 shows the corresponding wavelet functions. The underlying continuous functions are shown, rather than the discrete filter coefficients. It can be seen that the zeroth spline is in fact the Haar wavelet which has the most compact support. As the order of the spline increases so the support of scaling function and the wavelet widens. Figures 5.4 and 5.5 show the effect upon the time localisation of an impulse that this widening of the time support has at scales 1 and 5 of an undecimated complex wavelet analysis. The time localisation is reduced for each wavelet analysis at the higher scale as is expected for constant Q analysis. At both scales the zeroth-order spline wavelet analysis has better time localisation but the shape of the magnitude curve at scale 5 is not smooth and gives the impression that there are three impulses rather than one in the analysed frame. It can be seen that the shape of twentieth-order magnitude curve is the shape of the Gaussian function which high order B-spline scaling functions tend to. Although B-spline wavelets have been described in general in this section, the rest of the chapter deals with the specific example of the cubic B-spline wavelet. This particular order offers a good compromise between support width and time/frequency resolution optimisation: "our numerical results tend to support the conclusion the localisation performance of the cubic spline wavelet transform should be sufficient for most practical applications" [Unser et al, 1992].

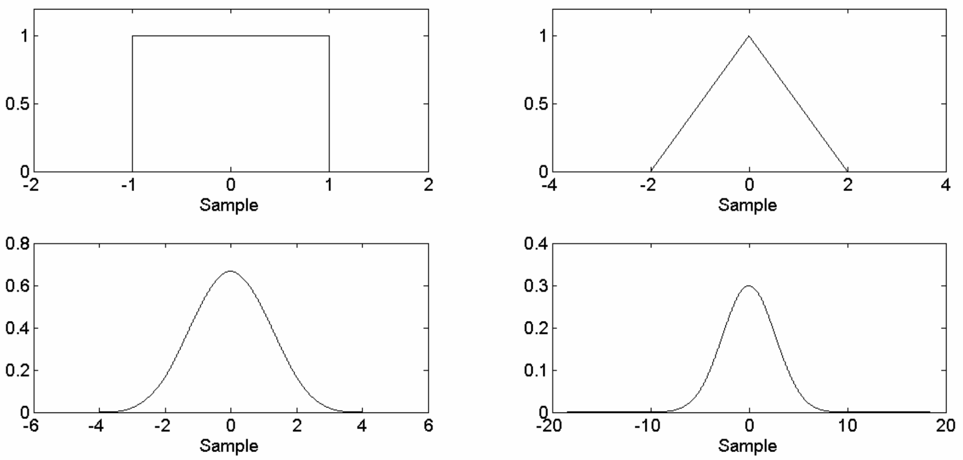


Figure 5.2: Scaling functions for the zeroth (top left), first (top right), third/cubic (bottom left) and twentieth order (bottom right) spline wavelet analysis at scale 1.

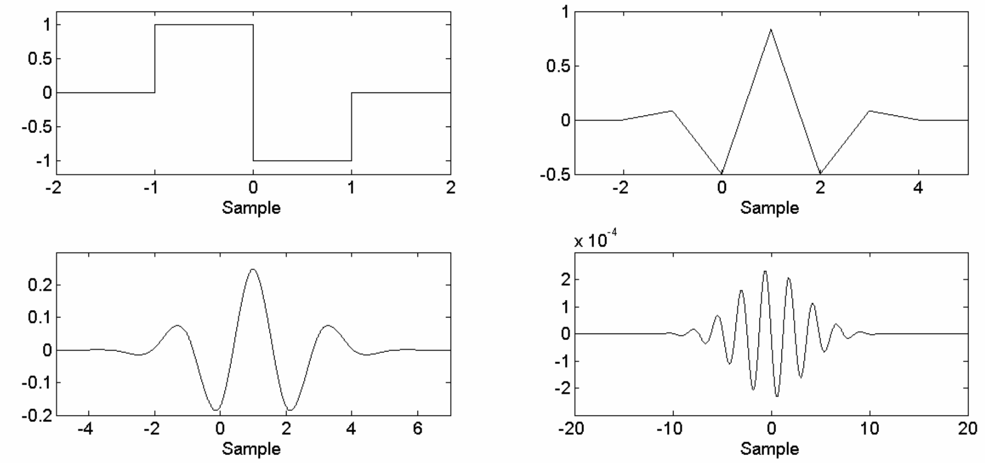


Figure 5.3: Wavelet functions for the zeroth (top left), first (top right), third/cubic (bottom left) and twentieth order (bottom right) spline wavelet analysis at scale 1.

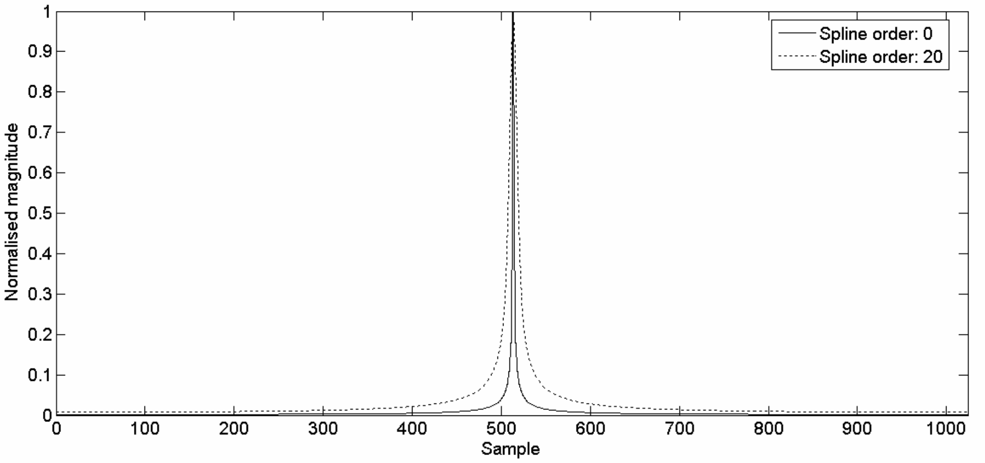


Figure 5.4: Normalised magnitudes at scale 1 of the undecimated complex B-spline (order 0 and 20) wavelet transforms of a 1025 sample frame containing an impulse at sample 513.

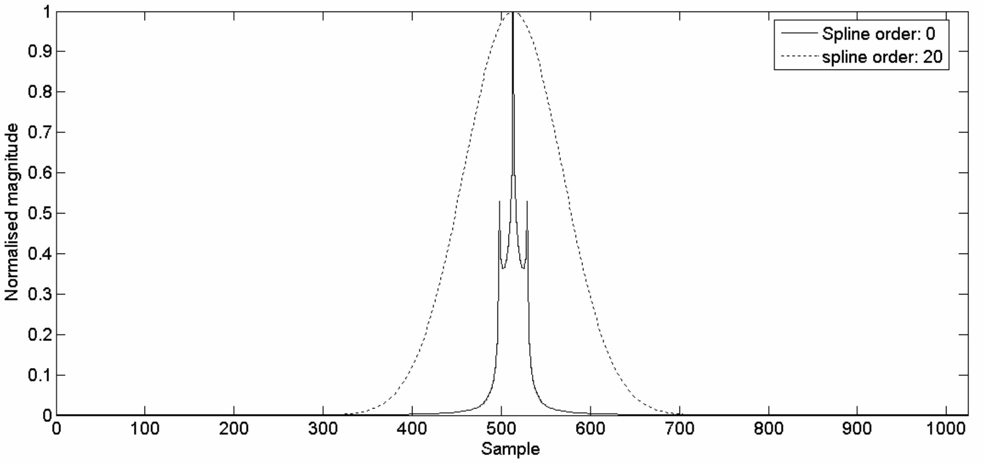


Figure 5.5: Normalised magnitudes at scale 5 of the undecimated complex B-spline (order 0 and 20) wavelet transforms of a 1025 sample frame containing an impulse at sample 513.

5.2.2 Mean instantaneous frequency estimation

As is the case for the STFT the mean instantaneous frequency of a spectral component can be estimated using the complex wavelet transform. Reassignment is used for frequency estimation in the STFT in the previous chapter since this method can be performed using only the data in a single frame. Unlike the STFT the wavelet transform provides more than one coefficient at each scale (apart from the highest scale of a critically sampled wavelet transform). This implies that the mean instantaneous frequency can be estimated from the first order difference of the phase between consecutive coefficients in a given scale within a single frame:

$$\overline{f}_{j,k,k+1} = \left(\phi_{\text{detail}\, j,k+1} - \phi_{\text{detail}\, j,k}\right)\frac{F_s}{2\pi} \qquad (5.6)$$

for an undecimated transform and

$$\overline{f}_{j,k,k+1} = \left(\phi_{\text{detail}\, j,k+1} - \phi_{\text{detail}\, j,k}\right)\frac{F_s}{2^{j-1}\left(2\pi\right)} = \frac{F_s}{2^j \pi} \qquad (5.7)$$

for a decimated transform, where *j* is the scale, *k* is the coefficient index at that scale and ϕ is the phase of the coefficient. Figure 5.6 shows the estimated frequency for sinusoids in the range 20 Hz to 20 kHz. The frequency estimates are derived using the undecimated complex cubic B-spline transform, each estimate is produced by finding the phase difference between the two middle coefficients (*k* = 512,513) for a 1025 sample frame at scale 1. Figure 5.7 shows the frequency estimates produced by the decimated complex cubic B-spline transform.

It is clear from this figure that the decimated transform does not sample the phase at a high enough rate to prevent aliasing within the frequency band covered by scale 1 $\left(F_s/4 \text{ to } F_s/2\right)$ although at frequencies below this band the phase rotates slowly enough for the correct frequency to be derived. At this scale the effective sampling rate is $F_s/2$ whereas for the undecimated transform it is F_s. A straightforward solution to this problem is to not decimate the output detail coefficients at each scale. Whilst this doubles the number of coefficients produced at each scale it does not increase the computational burden since the detail coefficients are not used in further iterations of the decimated algorithm, it is only the approximation coefficients that are used recursively. This prevents aliasing at scale 1 however aliasing still occurs at higher scales since the number of detail coefficients at each scale is reduced by decimation of the approximation coefficients at the previous scale. There are two possible solutions to this problem. The first is to modify any negative frequency estimates via the following:

$$\overline{f}_{\text{corrected}} = \begin{cases} \dfrac{F_s}{2^j} + \overline{f}_{\text{estimated}}, & \overline{f}_{\text{estimated}} < 0 \\ \overline{f}_{\text{estimated}}, & \overline{f}_{\text{estimated}} \geq 0 \end{cases} \quad (5.8)$$

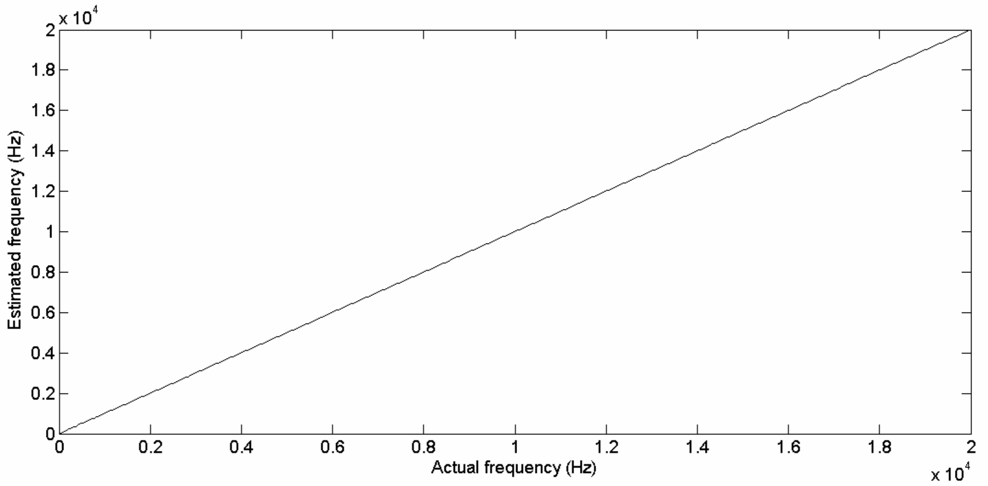


Figure 5.6: Frequency estimates derived from phase data from scale 1 of the undecimated complex cubic B-spline wavelet transform of a 1025 sample frame. This is also the output of the decimated transform if the detail coefficients that are output at each scale are not decimated.

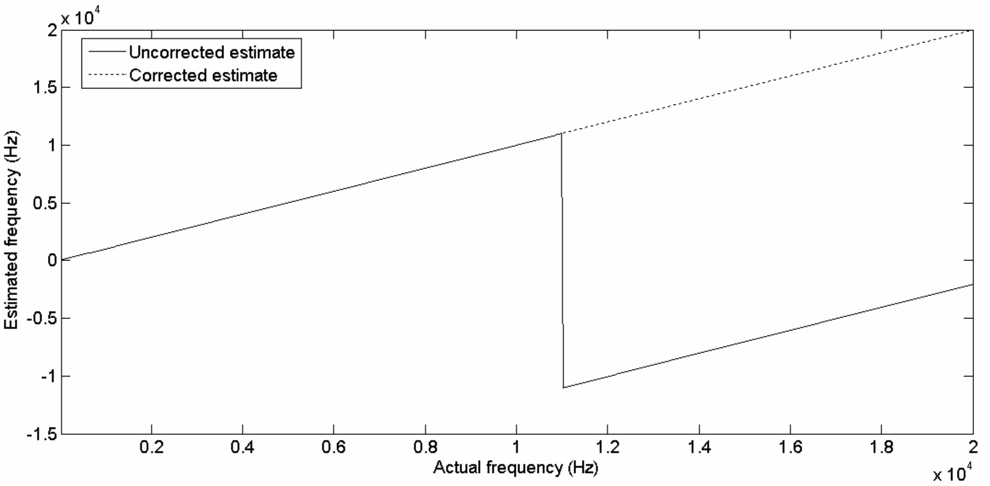


Figure 5.7: Corrected and uncorrected frequency estimates derived from phase data from scale 2 of the decimated (critically sampled) complex cubic B-spline wavelet transform of a 1025 sample frame. The detail coefficients at each scale have not been decimated.

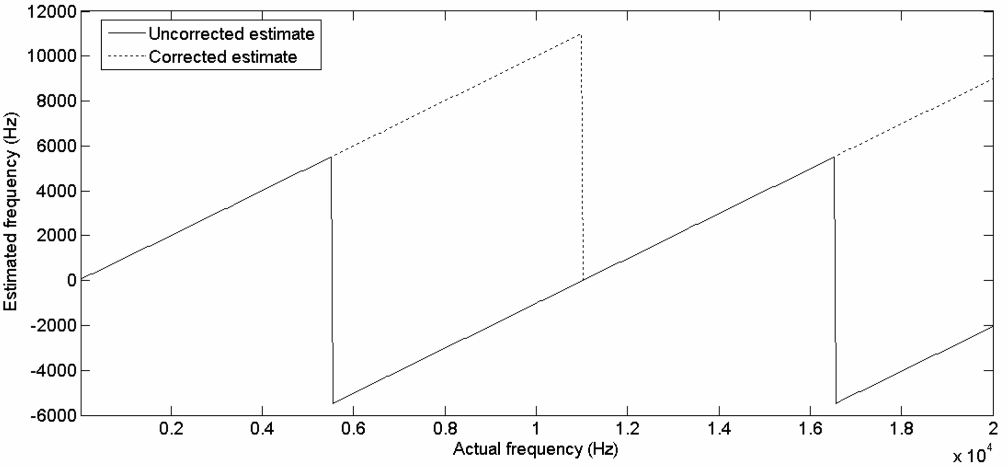


Figure 5.8: Corrected and uncorrected frequency estimates derived from phase data from scale 3 of the decimated (critically sampled) complex cubic B-spline wavelet transform of a 1025 sample frame. The detail coefficients at each scale have not been decimated.

However, (5.8) is only effective at the next lowest scale. At higher scales there is not a unique relationship between $\overline{f}_{\text{corrected}}$ and $\overline{f}_{\text{estimated}}$. Whilst non-decimation of the output detail coefficients and use of (5.8) will produce a correct frequency estimate for a component which falls within the frequency range of the scale and that of the scale above, lower frequency components can appear as alias frequencies within this range. Figure 5.8 illustrates this for a decimated transform at scale 2. Therefore for the decimated transform only the highest scale is guaranteed to produce a correct $\overline{f}$ estimate for a spectral component with any mean frequency in the range 20 Hz to 20 kHz. The second solution is to sample the phase at a higher rate as the undecimated transform does. The undecimated transform theoretically offers non-aliased frequency estimation for any spectral component at all scales (although in practice this is limited at lower scales).

5.2.3 Shift invariance

An additional advantage of the undecimated transform is perfect shift invariance. It has been demonstrated that orthogonal dual tree wavelets (complex wavelets implemented using two ‘trees’ of real wavelets with fractional sample shifts between them) offer ‘approximate shift invariance’ [Kingsbury, 2001]. The desirability of shift invariance in an audio analysis system, as opposed to one for audio data compression, is obvious. As discussed in section 3.84 there are both time domain and frequency domain implications of shift variance. For an analysis system it is desirable that identical components should have identical magnitude wherever they occur in time and that identical components should have identical $\overline{f}$ for all scales.

A straightforward test of shift invariance in the time domain is to measure the maximum deviation of coefficient values from the mean coefficient value for a unit impulse at each position in an input sequence. The ratio of the mean and the deviance, along with the upper aliasing limit, for each scale is given in decibels for the real and complex, decimated and undecimated transforms in table 5.1. The variance is tested across the centre of the input sequence, since for overlapping frames only wavelet coefficients in the middle of the frame are used for subsequent synthesis (the middle 512 values for a 1025 sample frame and an overlap factor of 2). The values for the complex transforms were calculated as the modulus of the transform coefficients. The values for the decimated transform were calculated with non-decimation of the detail coefficients and (5.8); a standard decimated algorithm would not be shift invariant at scale 1. The lack of perfect shift invariance at lower scales of complex transform is due to fluctuations in the imaginary component introduced by the Hilbert transform. Figure 5.9 shows the detail coefficients at each sample position for both types of transform at scale 2.

Scale	Undecimated			Decimated		
	Real (dB)	Complex (dB)	Non-alias limit (*Fs*)	Real (dB)	Complex (dB)	Non-alias limit (*Fs*)
1	$-\infty$	$-\infty$	1.0000	$-\infty$	$-\infty$	1.0000
2	$-\infty$	$-\infty$	1.0000	-6.7	-22.2	0.5000
3	$-\infty$	$-\infty$	1.0000	-7.5	-14.6	0.2500
4	$-\infty$	$-\infty$	1.0000	-0.6	-22.5	0.1250
5	$-\infty$	$-\infty$	1.0000	-0.6	-20.4	0.0625
6	$-\infty$	$-\infty$	1.0000	-0.6	-19.4	0.0313
7	$-\infty$	-107.8	1.0000	-0.0	-18.8	0.0156
8	$-\infty$	-65.5	1.0000	-0.0	-18.4	0.0078
9	$-\infty$	-49.0	1.0000	-0.0	-17.4	0.0039
10	$-\infty$	-55.6	1.0000	-0.0	-14.9	0.0020

Table 5.1: Comparison of undecimated and decimated wavelet transforms. Shift invariance relative to mean coefficient level (dB) and aliasing limit (multiple of sampling rate).

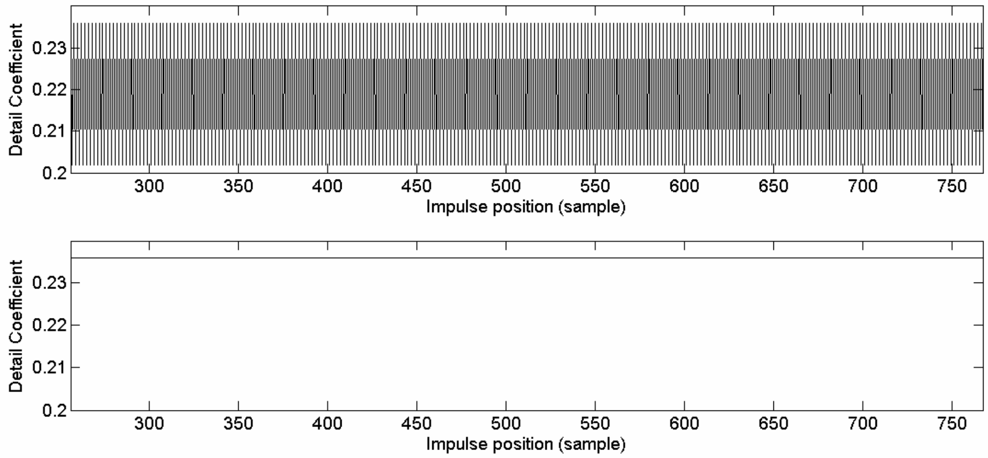


Figure 5.9: Shift variance in scale 2 detail coefficients for the decimated (top) and undecimated (bottom) complex cubic B-spline wavelet transforms.

5.2.4 The partially decimated wavelet transform

It is clear that the undecimated transforms outperform the decimated in terms of shift invariance as well as mean frequency estimation but the computational cost is high. The undecimated transform is implemented as shown in figure 5 (adapted from [Shensa, 1992]). This algorithm is numerically and computationally equivalent to the à trous algorithm but offers a more straightforward and intuitive implementation. For full dyadic decomposition, circular convolution (i.e. the length of the sequence to be filtered does not increase at successive scales), filters of length L_{LPF} and L_{HPF} and a signal of length N which is a power of two the à Trous algorithm requires $\left(L_{LPF}+L_{HPF}\right)N\log_2 N$ multiply and add operations whereas the decimated transform (Mallat algorithm) requires only $2\left(L_{LPF}+L_{HPF}\right)N$. For a 10 scale transform the undecimated transform is 5 times as computationally expensive as its decimated version.

In order to offer some mediation between these two extremes the partially decimated wavelet transform is proposed here. The principle is straightforward: the algorithm begins by filtering the signal and inserting holes into the filter until a given decomposition level (scale) is reached, at which point the filter remains the same and the output is decimated for subsequent iterations. The only other wavelet analysis that combines decimated and undecimated transforms in this way is the over complete DWT (OCDWT) described in [Bradley, 2003]. However, this system begins with decimation and then at higher scales switches to filter dilation. This order is reversed in the system proposed here since this way shift variance is reduced at all scales.

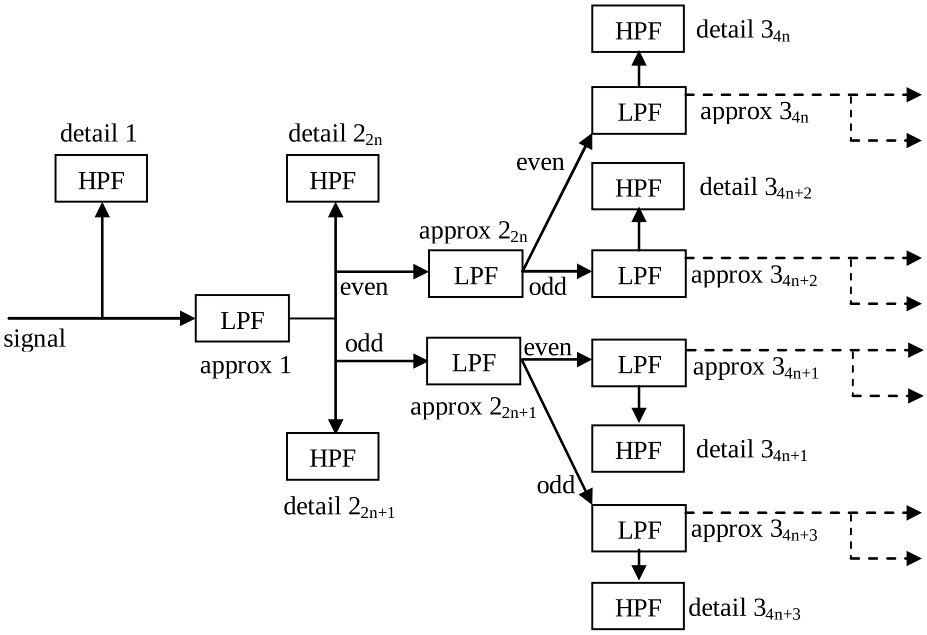


Figure 5.10: Alternative implementation of the undecimated discrete wavelet transform (adapted from [Shensa, 1992]).

Figure 5.11 illustrates the partially decimated wavelet transform where increasing filter subscripts *n* indicate dilation by insertion of 2^{n-1} zeros. Treating the number of multiply and add operations in the decimated part of the transform as a geometric series the cost, *C*, of the partially decimated transform can be approximated by:

$$C \approx \begin{cases} N\left(L_{LPF}+L_{HPF}\right)\left(u+\left(1-2^{-d}\right)\right), & d>0 \\ Nu\left(L_{LPF}+L_{HPF}\right) & , \; d=0 \end{cases} \qquad (5.9)$$

where *u* is the number of undecimated scales and *d* is the number of decimated scales. Figure 5.12 shows the computational cost factor for transforms with $(u+d=)$ 10 scales, where the cost of the decimated transform $(u=1, d=9)$ is 1 and the undecimated transform is equivalent to $u=10, d=0$. Table 7.2 lists the shift variances and alias limits at each scale for two cubic spline transforms each with different levels of decimation. Figure 5.13 shows the magnitude of the complex cubic spline transform at scale 8 for an impulse at the centre of the analysis frame. Control of the amount of decimation within the transform offers a choice between the computational cost and the resolution of the wavelet analysis. It should be noted that even at full decimation the complex transform is less shift variant than its real counterpart although it is twice as expensive.

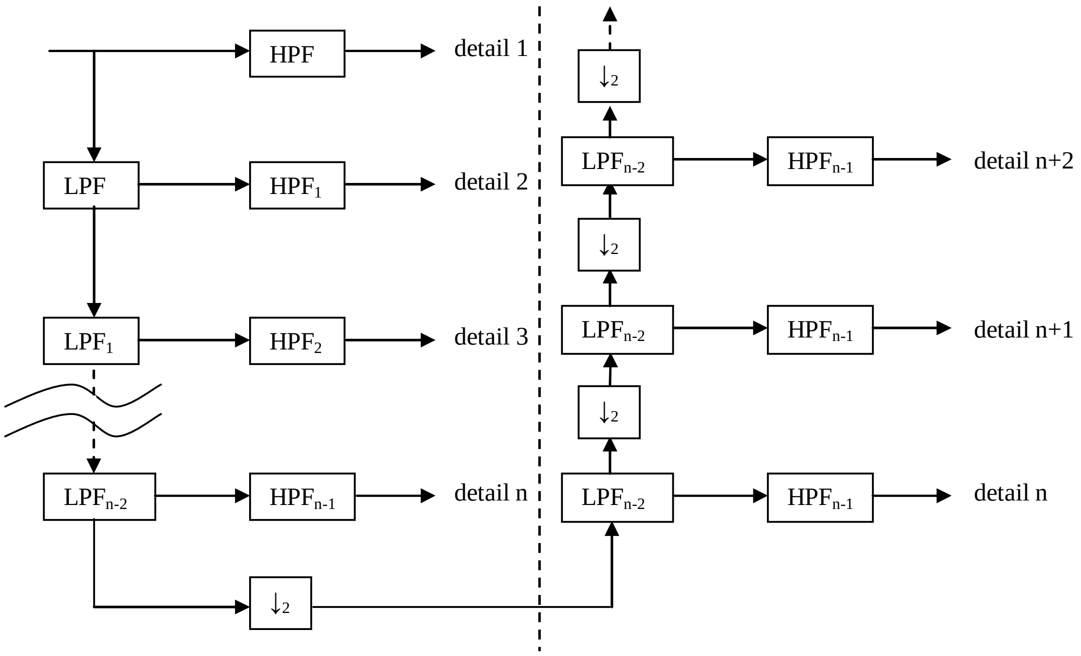


Figure 5.11: The partially decimated discrete wavelet transform.

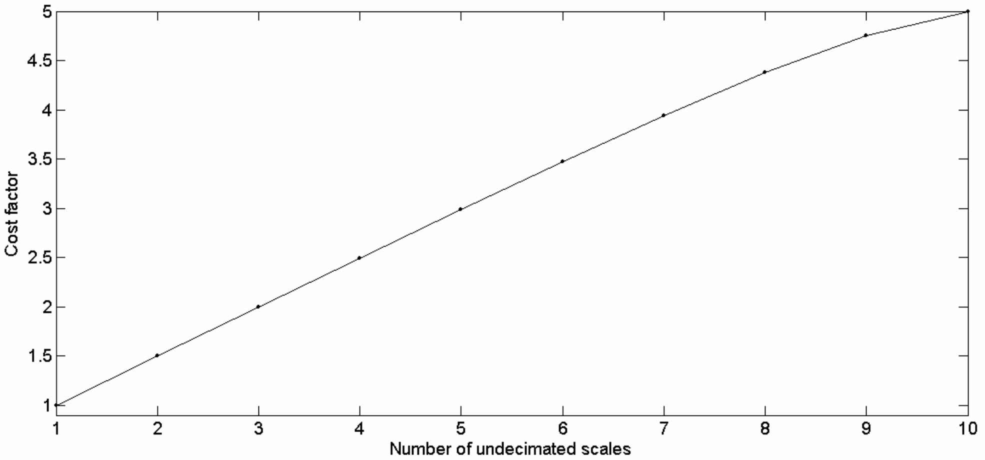


Figure 5.12: Computational cost of partially decimated 10 scale discrete wavelet transforms (relative to the cost of the decimated transform).

Scale	2 undecimated, 8 decimated			7 undecimated, 3 decimated		
	Real (dB)	Complex (dB)	Non-alias limit (*Fs*)	Real (dB)	Complex (dB)	Range(Hz)
1	$-\infty$	$-\infty$	1.0000	$-\infty$	$-\infty$	1.0000
2	$-\infty$	$-\infty$	1.0000	$-\infty$	$-\infty$	1.0000
3	$-\infty$	$-\infty$	1.0000	$-\infty$	$-\infty$	1.0000
4	-20.3	-41.6	0.5000	$-\infty$	$-\infty$	1.0000
5	-17.6	-38.5	0.2500	$-\infty$	$-\infty$	1.0000
6	-16.7	-37.4	0.1250	$-\infty$	$-\infty$	1.0000
7	-16.3	-36.9	0.0625	$-\infty$	-107.8	1.0000
8	-10.7	-30.1	0.0313	-68.3	-62.1	0.5000
9	-20.0	-27.8	0.0156	-65.2	-44.2	0.2500
10	-17.7	-28.6	0.0078	-64.1	-56.5	0.1250

Table 5.2: Comparison of wavelet transforms with different levels of decimation. Shift invariance relative to mean coefficient level (dB) and aliasing limit (multiple of sampling rate).

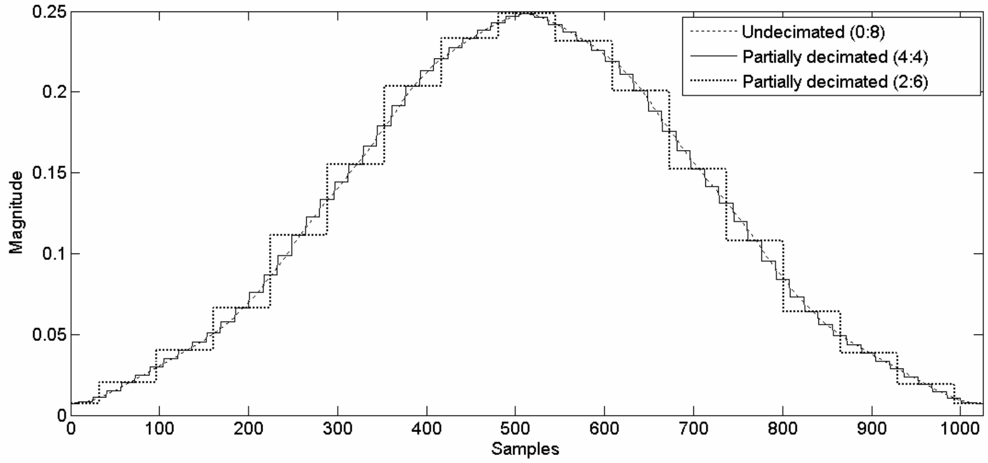


Figure 5.13: Magnitude response of the complex cubic spline wavelet transform with differing numbers of decimated scales. The figures in brackets in the legend indicate the ratio of decimated to undecimated levels. The input is a single impulse at the centre of the frame.

5.3 Short-time wavelet analysis

5.3.1 Magnitude estimation errors

Although the use of wavelets described in this chapter is also applicable to non frame based, non real-time signal modelling the specific application for this thesis necessarily divides the signal into small sections to enable quasi real–time analysis. In the short-time Fourier case the time support of all the filters is the same and related to the frame length by the zero padding factor. In the short-time dyadic wavelet case the time support of the filters doubles at each scale. For the m^{th} order spline wavelet the time support, or length, of the low and high pass filters in samples, at scale 1 is given by:

$$L_{\text{LPF}} = m + 2 \quad (5.10)$$

$$L_{\text{HPF}} = 3m + 2 \quad (5.11)$$

and, as a result of the convolutions and dilations in the transform, the wavelet length at scale j is given by:

$$L_{\text{wavelet}} = \left(2^{j-1} - 1\right) L_{\text{LPF}} + 2^{j-1} L_{\text{HPF}} - 1 \quad (5.12)$$

For a single impulse in the centre half of the frame (i.e the central 512 samples for a 1025 sample frame) the peak coefficient level will be the same at all scales. At lower scales for a full band component that lasts the whole frame, the coefficient levels near the centre of the frame will double at each scale since the time support of filter has been doubled. However this relationship breaks down at higher scales and/or near the frame boundaries since the local support of the filters extends beyond the frame (i.e. where the signal level is zero). This is shown in figure 5.14 for three different scales. At each scale a sinusoid the duration of the frame, whose frequency is the peak frequency of the wavelet filter at that scale, is analysed. At the highest scale the magnitude varies throughout the frame but as the scale is reduced this variation is increasingly localised at the frame boundaries. At scale 10 the length of the cubic spline wavelet is almost eight times the length of the frame, at scale 7 it is almost the length of one frame and at scale 5 it is just under a quarter of the length.

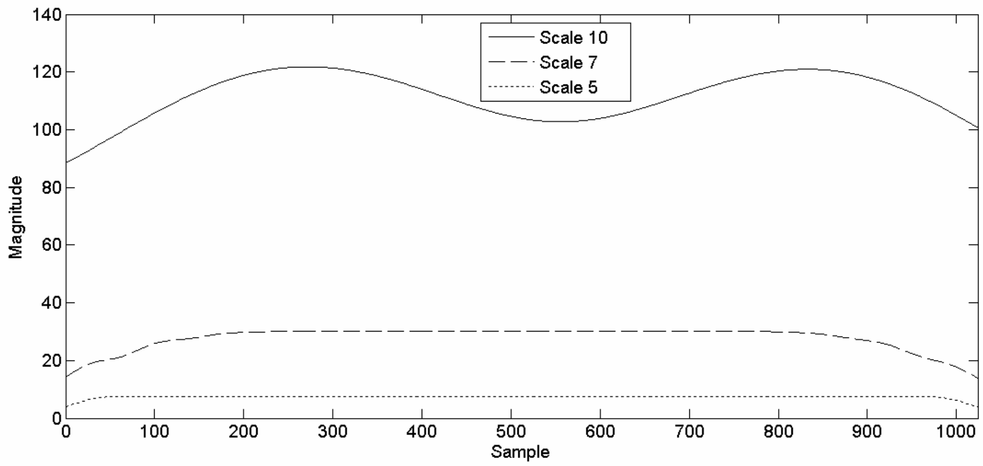


Figure 5.14: Magnitude variations for a single sinusoidal component during a single frame.

If the frames overlap then variations near frame boundaries can be ignored; the greater the overlap factor, the more samples that can ignored near the boundaries. For example, with an overlap factor of 4 and a frame size of 1025 the wavelet coefficients of concern correspond to the middle 257 samples of the frame. However for the highest scales (such as scale 10 in the figure) there is significant coefficient variation throughout the frame and increasing the overlap factor will not eliminate this. The only way to reduce the length of the filters is to reduce the number of scales and, hence, the actual number of filters too. For example a 1025 sample frame could be modelled with 6 wavelet band pass filters derived from the detail coefficients at the first 6 scales and a low pass filter derived from the approximation coefficients at scale 6. Whilst this would reduce the flexibility of the model it would serve to reduce these effects and, possibly, the computational complexity since less filter operations would be required.

However, the high and low pass compact B-spline filters are not the power complements of each other (i.e. their power spectra do not add to constant at all frequencies). Their power spectra are shown in figure 5.15. If, at the final level of decomposition, the approximation (i.e. the remainder of the signal) is to be modelled by a fixed filter then that filter should be the power complement of the final detail filter to ensure that energy is not lost and a hole in the spectrum is not formed. Such a filter can be devised in the Fourier domain by:

$$LPF_{\text{power complement}}\left(f\right)=\sqrt{1-\left(HPF\left(f\right)^{2}\right)} \qquad (5.13)$$

where $HPF\left(f\right)$ is the magnitude spectrum of the high pass wavelet filter given by:

$$HPF(f) = 2\cos^{m+1}\left(\pi\left(f - \tfrac{1}{2}\right)\right)\text{sinc}^{2(\text{m}+1)}\left(f - \tfrac{1}{2}\right) \qquad (5.14)$$

where *m* is the spline order [Unser et al, 1993]. The magnitude spectrum of the power complement filter is also shown in figure 5.15. This power complement filter has a narrow transition band and its derivative at the Nyquist frequency is not zero. A good approximation to such a filter requires a large number of taps. Figure 5.56 shows the performance in matching the desired magnitude response for FIR filters with different numbers of taps. Performance is presented as the ratio, in dB, of error energy for each bin to the total energy in the spectrum measured across an 8192 point DFT. A smaller number of taps would be required for an IIR filter with equivalent performance however such a filter is not compatible with the à trous algorithm. Since the power complement filter requires a large number of taps the original problem of filter support extending beyond the windowed frame is not solved. This problem is inherent in short-time wavelet analysis, however further investigations into the possibility of a solution will be the subject of future work in this area. In the wavelet part of the real-time analysis and resynthesis described in the following chapter the magnitude scaling at higher scales is adjusted to compensate for the support of the filters extending beyond the frame boundaries. At these higher scales the magnitude scaler, rather than halving at each successively higher scale (to compensate for the doubling in the support length of the filter) has been determined empirically.

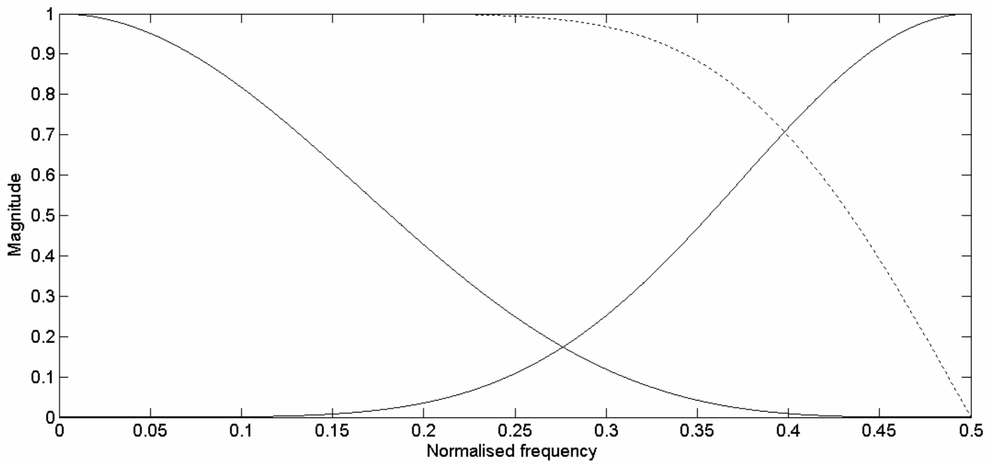


Figure 5.15: Magnitude of cubic B-spline high and low pass filter (solid line) and power complement to high pass filter (dotted line). Frequency axis is in units of $\frac{Fs}{2^j}$ where *j* is scale.

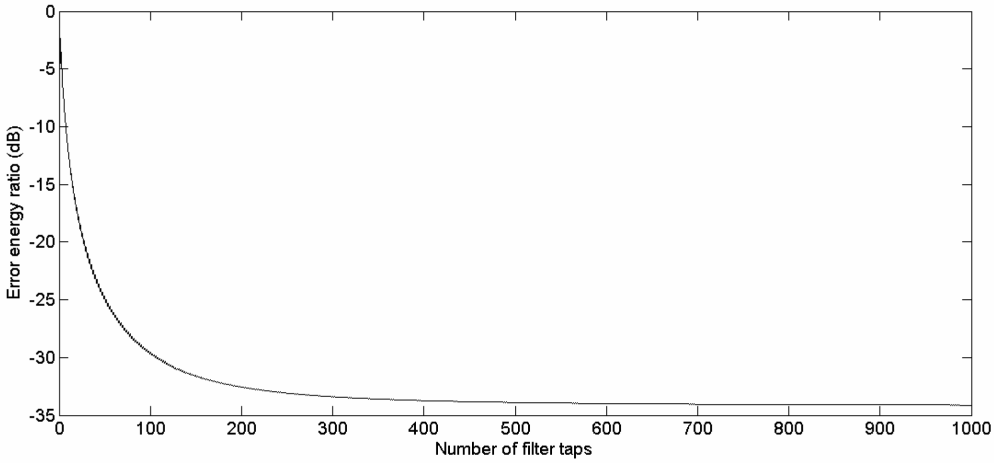


Figure 5.16: Error energy versus number of FIR filter taps for the power complement to the high pass wavelet filter.

5.3.2 Frequency estimation errors

The process of dividing a signal into separate frames introduces spectral artefacts into the analysis. The rectangular window can introduce abrupt changes into the signal that did not previously exist which results in a spreading of energy in the spectral domain. Section 3.5.3 discussed how different window functions applied to signal frames can reduce such effects (albeit at the expense of widening the main lobe of the filter). It is not possible to separately specify a window function for the wavelet transform since its support would need to be different at different scales. However the scaling function can be seen as analogous to the Fourier window, especially for the B-spline wavelets since the scaling function is approximately a non-modulated Gaussian and the wavelet is a modulated Gaussian, implying that the window function is a Gaussian and the underlying basis functions are infinite sinusoids. In the context of a combined Fourier/wavelet analysis system where the Fourier analysis occurs first and uses a window function, as is proposed here, the wavelet analysis will inherit this windowing in the residual signal.

The interaction of this window with the shape of the wavelet and scaling functions impacts both negatively and positively on the analysis. Impulsive components introduced by the rectangular window generate a long term, high frequency ringing effect in the imaginary part of the Hilbert transformed signal which has the effect of smearing energy in time at the lowest scale (this effect is also discussed in section 5.4.1). For spectrally remote components this has an adverse effect on the phase and, hence, the frequency estimation. Figure 5.6 in section 5.2.2 was produced from Hann, rather than rectangular windowed, frames. Figure

5.17 shows the estimated frequencies at scale one for the rectangular window. It can be seen that these estimates are seriously corrupted below about 6 kHz. Although this is out of the local band for that scale it could lead to aliasing of ‘out of scale’ components, although at such a distance from the centre frequency of this wavelet filter they would be unlikely to be high in level due to the attenuation by the filter. Figure 5.6 shows that this effect is completely eliminated by the use of a tapered window.

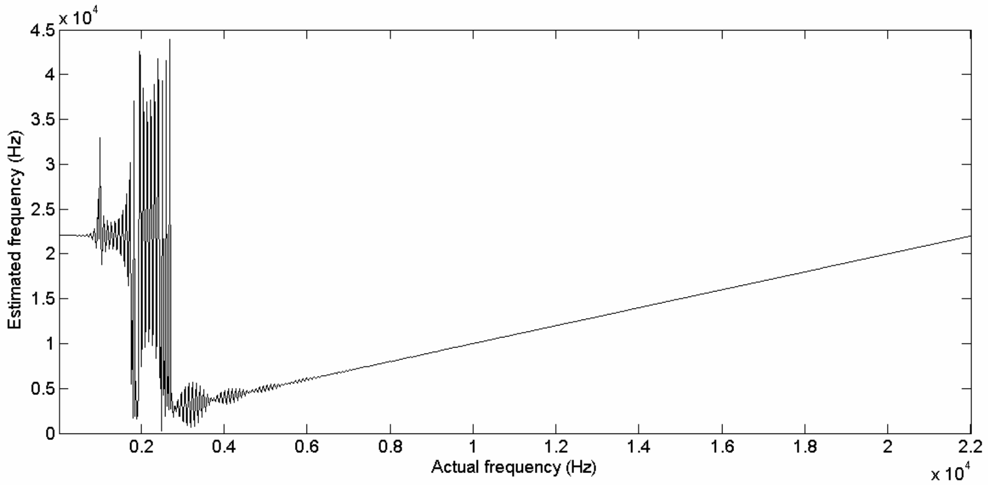


Figure 5.17: Frequency estimates derived from phase data from scale 1 of the complex cubic B-spline wavelet transform of a 1025 sample frame where a rectangular, rather than a Hann window (as used for figure 5.6) has been applied to the input frame.

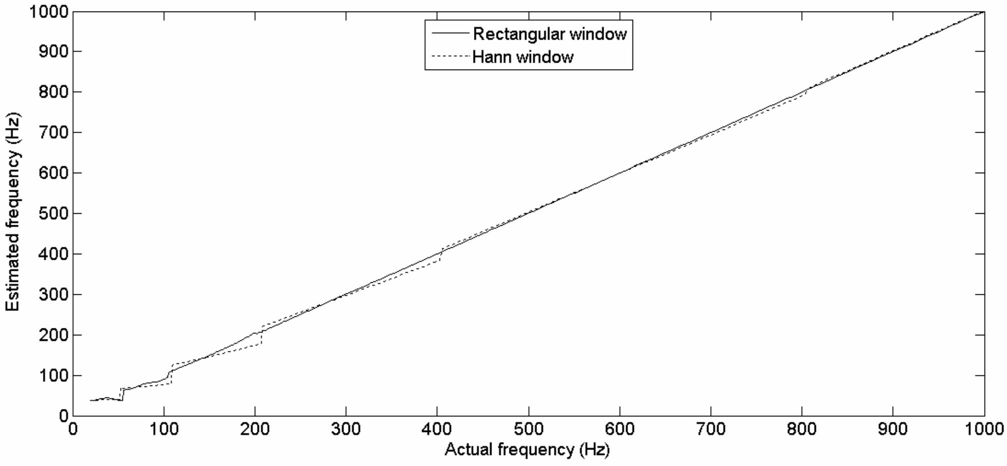


Figure 5.18: Frequency estimates derived from phase data from the peak scale of the undecimated complex cubic B-spline wavelet transform of a 1025 sample frame for two different window types.

An adverse effect of Hann ‘pre-windowing’ is poorer estimation of the mean instantaneous frequency at higher scales. Figure 5.18 demonstrates the frequency estimation in the range 20 Hz to 1 kHz for the peak scale for the rectangular and Hann windows. The estimates from the

Hann windowed frame improve for decreasing scale and for greater proximity to the centre frequency of the wavelet filter (at which points the estimates are correct). As the frequency increases above the centre of the wavelet filter it is underestimated and as the frequency decreases below the centre of the wavelet filter it is overestimated. The linear nature of the relationship between the actual frequency of the component and the estimate suggests that a rotation of the line about the centre frequency of the wavelet filter at lower scales would correct the frequency estimates.

Figure 5.19 shows the error between the actual and estimated frequencies across the audio frequency range for an estimate taken at the centre of the analysis frame (between samples 512 and 513 of 1025) and three quarters of the way through (samples 768 and 769). A correction algorithm has been found which models the error at each scale as a first order polynomial $f(x)$. The input to the polynomial is the deviation of the estimate from the centre frequency of the wavelet and the output is the predicted error. The variation of the slope of this polynomial with the distance from the centre of the frame, in samples, from which the estimate is taken is itself modelled as a second order polynomial $g(y)$. The corrected frequency estimate is given by:

$$f_{\mathrm{correct}} = f_{\mathrm{est}} - 2^{2(j-1)} g(y) f\left(F_s f_0 2^{j-1} - f_{\mathrm{est}}\right) \tag{5.15}$$

where y is the number of samples from the centre of the frame, j is the scale and $F_s f_0$ is the centre frequency of the wavelet at scale 1 (a further explanation of this is given in the next section). Figure 5.19 shows the error for the two estimates corrected by this method. This method improves the estimates down to 50 Hz (centre estimate) and 70 Hz (off-centre estimate). Below these frequencies the original estimates are better than the corrected ones. This method is expensive to implement in a real-time system and has not been used in the system described in chapter 6. As an alternative to performing this correction procedure *after* the wavelet analysis, the residual signal could be ‘un-windowed’ *prior* to analysis by dividing its real and imaginary parts in the time domain by the original window function. However, in practice this is not successful due to the presence of artefacts introduced into the imaginary part of signal by the Hilbert transform. For this reason the centre frequency and bandwidths of the lower frequency equalisers that are significantly affected by this estimation error are fixed at the centre of the wavelet analysis band in the current implementation of the system described in the next chapter.

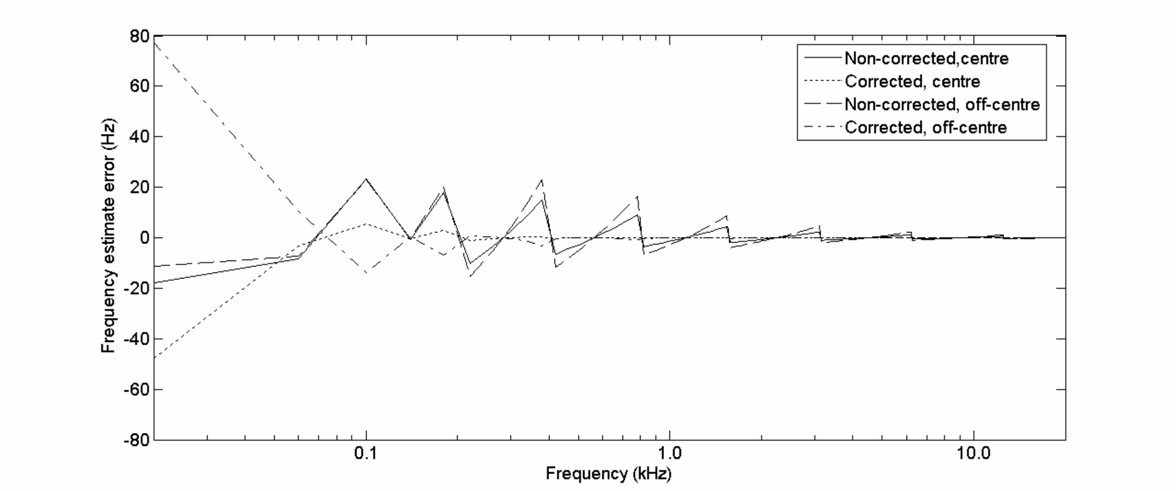


Figure 5.19: Frequency estimation error for corrected and uncorrected frequency estimates taken at the centre and three-quarters through a 1025 sample Hann windowed analysis frame for the complex cubic spline wavelet transform.

5.3.3 Computational cost for linear convolution

In the previous section the costs of the decimated, undecimated and partially decimated wavelet transforms were compared for circular convolution (periodic extension at the boundary, where the length of the filtered sequence does not increase at each filtering stage). Circular convolution is not desirable for time-scale analysis since the purpose is to describe where events occur in (linear) time. Where circular convolution is employed a component which occurs near the end of a frame may appear near the beginning of the frame in the analysis, this is especially likely at higher scales where the filter response is longer and so more likely to wrap around. For short-time wavelet analysis circular time, as opposed to linear time, within frames will make matching of components at frame boundaries difficult. As discussed in section 5.3.1, for wavelet *analysis* of short overlapping frames it is most important to capture events that occur near the centre of the frame accurately rather than obtain the most compact representation possible which, for example, might be the primary objective for a data compression application. For this reason linear convolution is employed at each scale with coefficients falling outside the region of interest for the frame (e.g. for an overlap factor of 4 this would be the region of width $\frac{N}{4}$ samples across the centre of the frame) being discarded from the analysis only when it is complete. Truncation of the filtered sequence at each scale would distort the analysis at the centre of the frame for higher scales. Linear convolution increases the cost of the transform since the number of samples to filtered at each scale is increased. In a real-time context computational cost is an important consideration. This section derives expressions for the cost of convolution for different types of wavelet transform described in this chapter.

When a sequence of length *S* is convolved with a filter of length *L* the output length *O* is given by:

$$O = S + L - 1 \qquad (5.16)$$

For the undecimated transform the input sequence at one scale is the approximation of the previous scale which is achieved by convolution with the dilated low pass filter. Therefore, for the undecimated transform, the sequence length N_j before low pass filtering at scale *j* is given by:

$$N_j = \left(N + \left(L_{LPF} - 1\right) \sum_{n=1}^{j-1} 2^{n-1} \right) = N + \left(L_{LPF} - 1\right)\left(2^{j-1} - 1\right) \qquad (5.17)$$

Where *N* is the frame length. This gives a total cost for the transform of:

$$C = \left(L_{LPF} + L_{HPF}\right) \sum_{j=1}^{J} N_j = \left(L_{LPF} + L_{HPF}\right)\left(NJ + \left((L_{LPF} - 1)(2^J - 1 - J)\right)\right) \qquad (5.18)$$

Where *J* is the number of scales. For the decimated transform the filter output is decimated at each scale and so the sequence length at scale *j*, before filtering and decimation, is given by:

$$N_j = \left\lceil N 2^{-(j-1)} + \left(L_{LPF} - 1\right)\left(1 - 2^{-(j-1)}\right) \right\rceil = \left\lceil 2^{-(j-1)} \left(N - L_{LPF} + 1\right) + L_{LPF} - 1 \right\rceil \quad (5.19)$$

Allowing for rounding up of numbers of coefficients when an odd length sequence is decimated the approximate total cost is given by:

$$C = \left(L_{LPF} + L_{HPF}\right) \sum_{j=1}^{J} N_j \approx \left(L_{LPF} + L_{HPF}\right)\left(J\left(L_{LPF} - 1\right) + \left(N - L_{LPF} + 1\right)\left(2 - 2^{-(J-1)}\right)\right) \ (5.20).$$

Figure 5.20 compares the cumulative computational cost at different scales of the decimated and undecimated complex cubic B-spline transforms, using circular and linear convolution for a 1025 sample input sequence. For comparison the cost of 1024 and 8192 point FFTs are also plotted (FFT cost estimation is described in section 3.5.2).

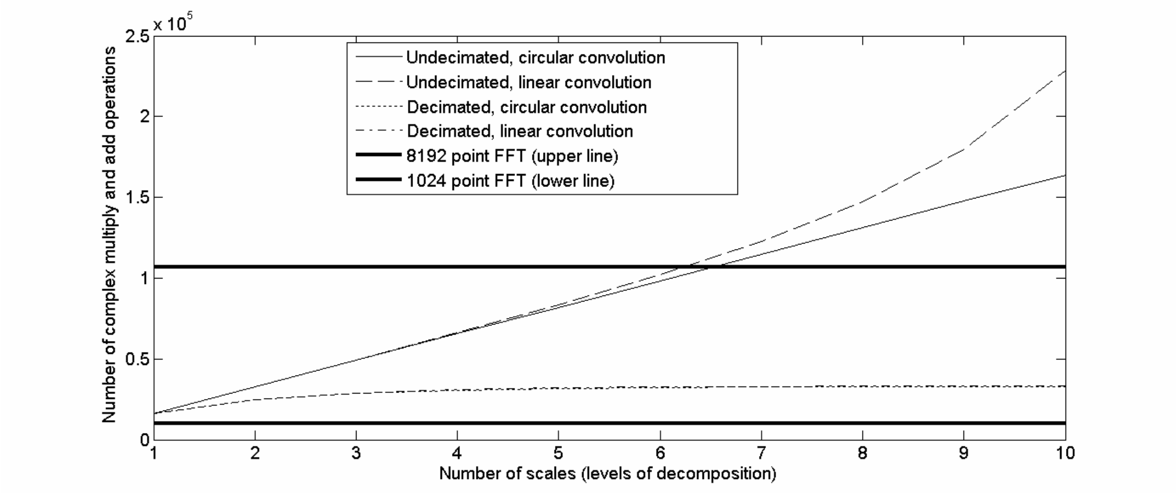


Figure 5.20: Computational cost of various types of complex cubic spline wavelet transform for an input length of 1025 samples. Example FFT costs are included for comparison, the cost of generating the analytic signal is not considered.

Equations (5.17) – (5.20) can be combined to calculate the cost of the partially decimated transform described in section 5.4. The cost of calculating the undecimated scale coefficients can be calculated directly from (5.18) where *N* is the length of the input sequence and *J* = *U* is the number of undecimated scales. The cost of calculating the subsequent decimated coefficients is given by a modified version of (5.20):

$$C = \left(L_{LPF} + L_{HPF}\right)\sum_{d=1}^{D} N_d = \left(L_{LPF} + L_{HPF}\right)\left(D\left(L_{dLPF} - 1\right) + \left(N_{undec} - L_{dLPF} + 1\right)\left(2 - 2^{-(D-1)}\right)\right) \quad (5.21)$$

where *D* is the number of decimated scales and N_{undec} is the length of the final approximation sequence output from the undecimated part of the transform, given by (5.17) where $j = U + 1$, which is then halved (since this sequence is decimated before the next filtering stage). L_{dLPF} is the length of the dilated LPF and is given by:

$$L_{dLPF} = \left(L_{LPF} - 1\right)2^{U-1} + 1 \qquad (5.22)$$

where *U* is the number of undecimated scales. The reason that the combined HPF and LPF lengths before the summation in (5.21) are not the lengths of the dilated filters is because this part of the calculation happens ‘À Trous’. Figure 7.9 shows the cost of a ten scale partially decimated complex cubic spline transform implemented with linear convolution for different numbers of undecimated scales.

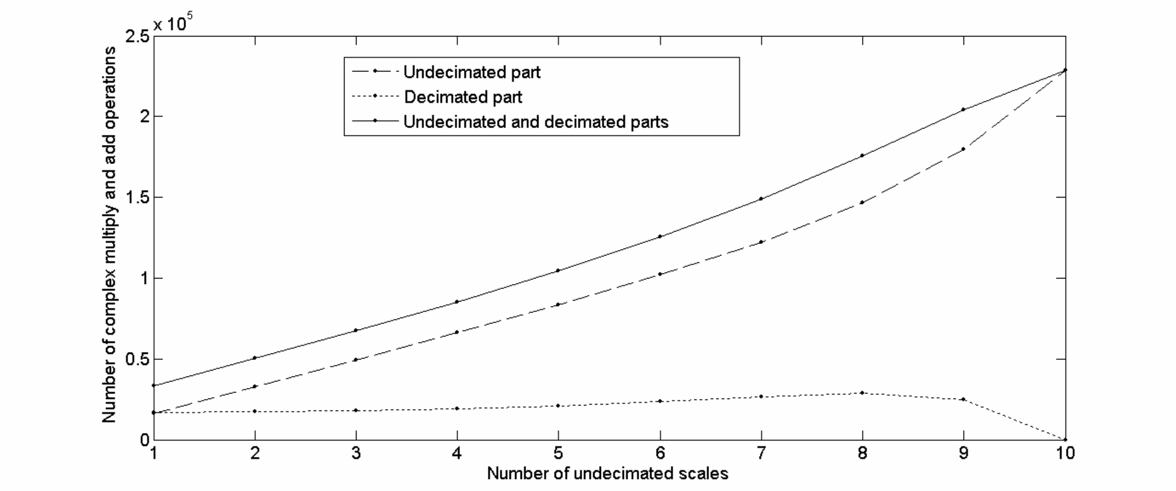


Figure 5.21: Computational cost of the partially decimated complex cubic spline wavelet transform. The number of undecimated scales is given. The number of decimated scales is the difference between the total number of scales (10 in this case) and the number of undecimated scales. The input is 1025 samples long. The cost of generating the analytic signal is not considered.

5.4 Component modelling

5.4.1 Wavelet filter behaviour

It is proposed here to use the wavelet transform to produce data for controlling filters with continuously variable centre frequency. Despite having control of the time/frequency localisation of the analysing filters via the spline order, the centre frequencies of these filters are fixed. The frequency estimation process described in section 5.3 offers a means of estimating the mean instantaneous frequency of the underlying components which excite these filters. However the estimated magnitudes of these underlying components will be biased by the frequency response of the fixed wavelet filters and this must be accounted for if the underlying components are to be correctly and intuitively described. This was discussed for Fourier analysis of sinusoids in section 3.11.

As the order of the spline increases so the resulting wavelet tends to a modulated Gaussian (the Morlet wavelet) and its Fourier transform is approximated by (and tends to with increasing *m*):

$$\Psi(f) \approx 2\left(C(f)\right)^{m+1} \quad (5.23)$$

where *n* is the order of the spline and

$$C(f) \approx \frac{\sin^2(2\pi f)}{4\pi^3 f\left(f-\tfrac{1}{2}\right)^2} \qquad (5.24)^1$$

This function is plotted in figure 5.22. The value of $C(f)$ is found by determining the value of *f* at which its first derivative is zero and the frequency at which this occurs is the centre frequency of the wavelet. This is found to be (rounded to four decimal places):

$$f_0 = 0.4092 \qquad (5.25)$$

[Unser et al, 1992]. Therefore the centre frequency of the wavelet at a given scale *j* for a sampled signal is approximated by:

$$f_c = \frac{f_0 F_S}{2^{j-1}} \qquad (5.26)$$

but this will only hold if the input sequence has been properly initialised (as discussed in section 3.8.6). For spline wavelets it has been proposed that this initialisation is performed by convolving the input sequence with the B-spline of order *m* sampled at the integers, b^m (equation 3.109) [Unser et al., 1992]. However this approach has been found to be non-ideal here, since the behaviour of the wavelet filter at low scales deviates from that predicted by (5.23), and an improvement to this initialisation is proposed.

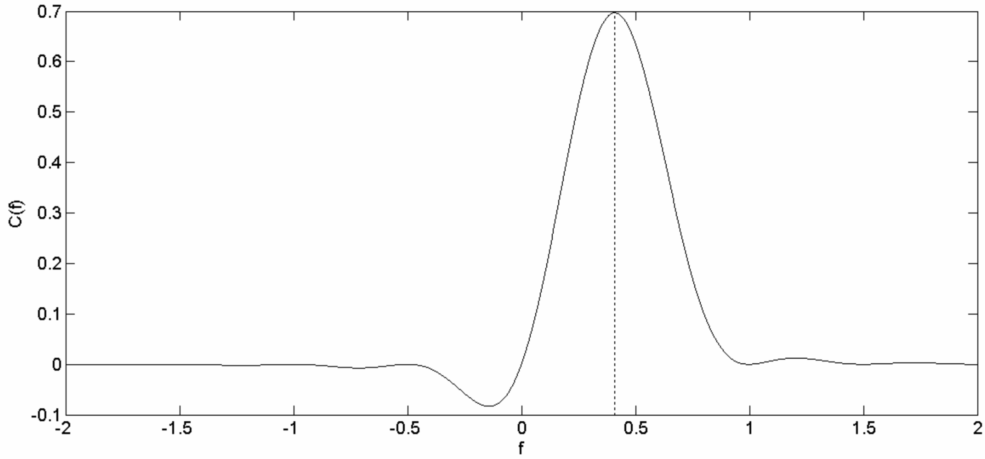


Figure 5.22: The function $C(f)$ (equation (5.20)). The peak value $C(f_0)$ is shown with a dotted line.

[1] This is corrected from [Unser et al, 1992]. Equation (4.2) quoted in this paper is inconsistent with the rest of the paper (figure 2) and the quoted value for $C(f_0)$.

The Fourier transform of the continuous B-spline of order *m* is given by:

$$F(\omega) = \operatorname{sinc}^{m+1}\left(\frac{\omega}{2}\right), \ f(x) = \beta^m \qquad (5.27)$$

which is a straightforward result since the spline of order *m* is found by $m+1$ convolutions of the zeroth order spline, whose Fourier transform is the sinc function. Figure 5.23 demonstrates that the DFT of b^m is quite different at high frequencies from the continuous function $F(\omega)$ sampled over the same number of points. This results in the wavelet filter at scale 1 applying a markedly different gain near the Nyquist limit. The difference between the expected and actual filter characteristics at scale 1 when (3.109) is used to initialise the input sequence is shown in figure 5.24. The most straightforward solution is to apply the initialisation in the Fourier domain and this is the approach adopted in the system described in chapter 6 since the data has already been transformed for the sinusoidal analysis. Otherwise the Fourier inverse of the quartic sinc function can be applied in the time domain, however this produces a longer filter than b^m which has only 3 coefficients for *m* = 3.

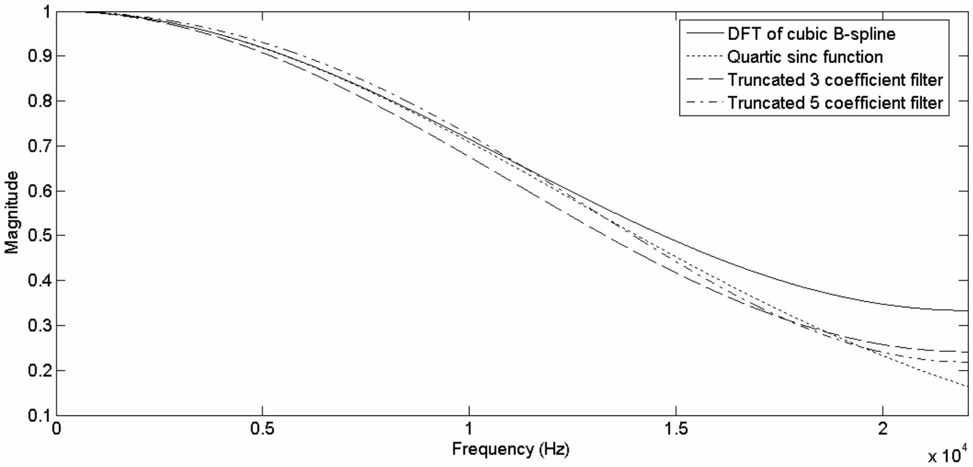


Figure 5.23: The Fourier transform of the cubic B-spline (quartic sinc function) and the 8192 point DFT of the spline sampled at the integers. Both are presented in the form of a low pass digital filter where $F_s = 44.1$ kHz. Two truncated filters derived from the inverse DFT of the quartic sinc function are also shown.

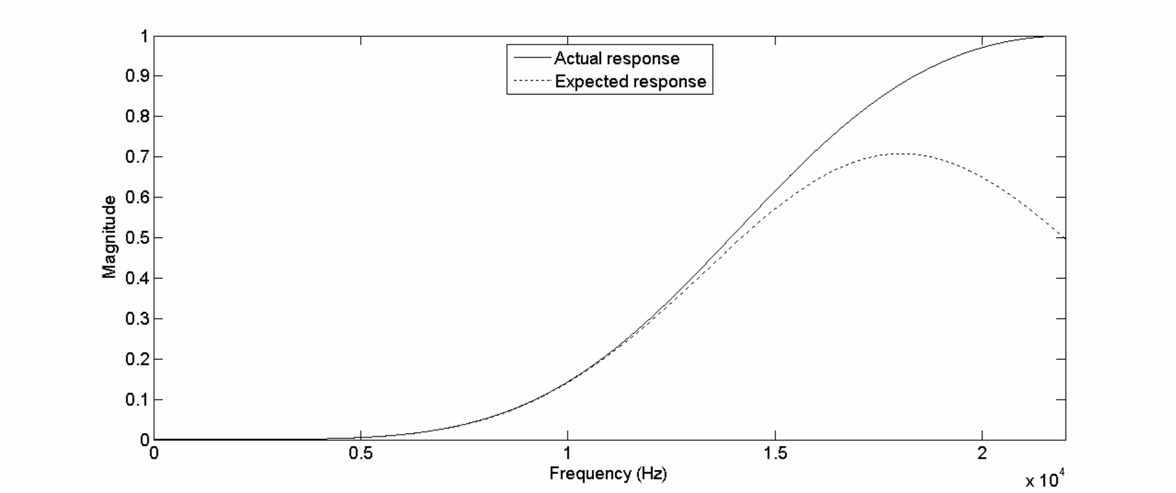


Figure 5.24: Normalised wavelet frequency response at scale 1. The actual response is that derived via DFT of the impulse response, the expected is that calculated with equation (5.19).

Truncation of the time domain sequence produced by the inverse Fourier transform produces a filter response which is closer to the ideal at high frequencies but at the expense of an ideal response at lower frequencies. As expected, increasing the number of coefficients improves the approximation of the truncated filter to the ideal one. Figure 5.23 shows the frequency responses of such truncated and normalised filters, one with three coefficients and the other with five. The coefficients are given in Table 5.3.

Filter	Filter coefficients
b^3	0.1667, 0.6667,0.1667
Truncated 3 coefficient	0.1897, 0.6207,0.1897
Truncated 5 coefficient	-0.0150, 0.1953, 0.6393, 0.1953, -0.0150

Table 5.3: Coefficients for wavelet initialisation filters.

When the initialisation is performed in the Fourier domain the actual frequency response of the wavelet at scale 1 is visually indistinguishable from the expected response in figure 5.24.

With this improved initialisation the peak magnitude of the frequency response doubles at each increment in scale since the bandwidth is halved (although this relationship breaks down due to truncation effects at the highest scales as described in section 5.3.1). In the time domain the peak magnitude of the complex impulse response remains the same and the time support is doubled. This is demonstrated for the first 5 scales in figures 5.25 and 5.26. This is, of course, the expected result given the uncertainty principle but it can be seen from this figure that complex spline wavelets present this result in a very straightforward and intuitive fashion. This, coupled with the possibility of fast implementation, makes them an obvious choice for investigating constant-Q time-frequency modelling. One anomalous result is the lower peak magnitude and the different 'skirt' shape in the response at scale 1. The cause of this is temporal spreading of energy at high frequencies in the imaginary part of signal by the Hilbert transform.

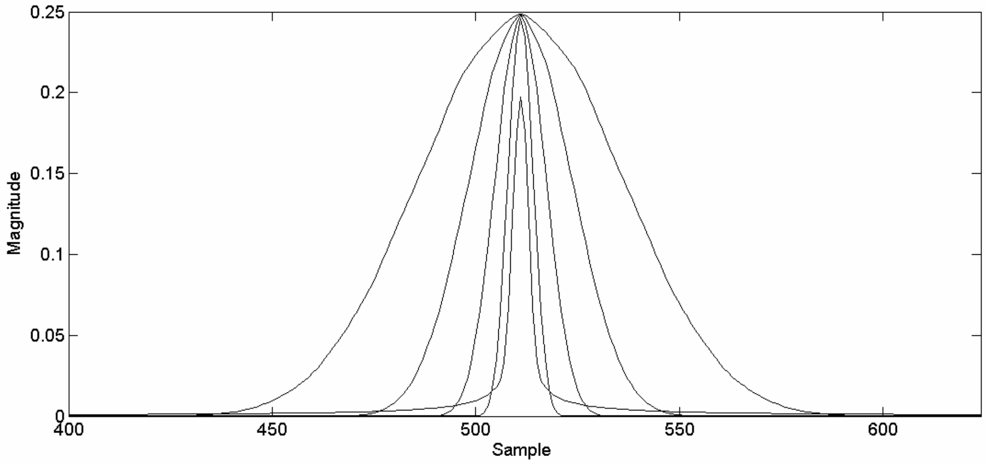


Figure 5.25: The time domain magnitude response of the complex cubic B-spline for an impulse at the centre of a 1025 frame. The first scale has the narrowest response, the second scale the next narrowest response and so on.

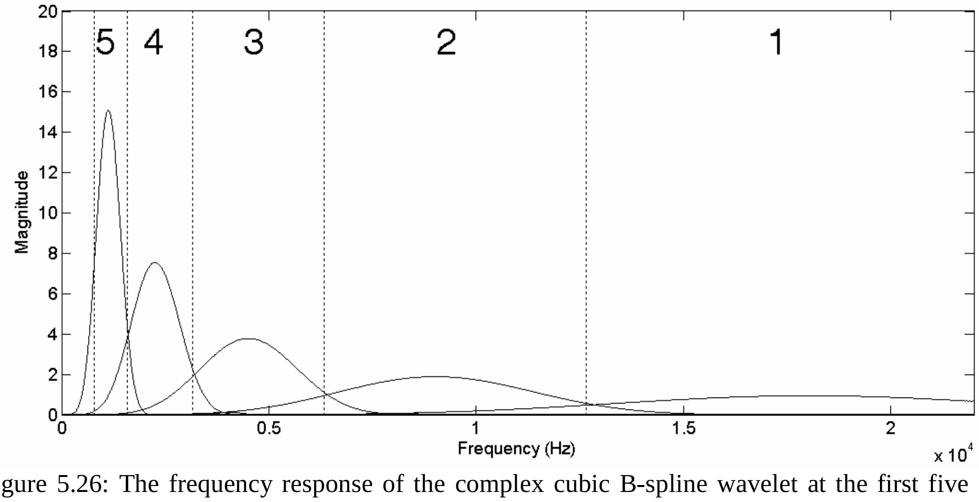


Figure 5.26: The frequency response of the complex cubic B-spline wavelet at the first five scales calculated from the 8192 point zero-padded DFT of the above time domain responses. The scale transition points are marked with dotted lines. The numbers indicate the scales.

5.4.2 Magnitude correction

With knowledge of the wavelet filter shapes and the mean instantaneous frequency of the analysed component it is possible to correct the magnitude estimate. At scale *j* the magnitude correction factor, *m*, is given by:

$$m = \frac{\left(C\left(f_0\right)\right)^{n+1}}{\left(C\left(f\right)\right)^{n+1}} = \frac{\sin^{2n+2}\left(2\pi f_0\right) f^{\,n+1}\left(f - \tfrac{1}{2}\right)^{2n+2}}{\sin^{2n+2}\left(2\pi f\right) f_0^{\,n+1}\left(f_0 - \tfrac{1}{2}\right)^{2n+2}} \qquad (5.28)$$

where f_0 is defined in (5.25) and

$$f = \frac{f_{\text{est}} 2^{j-1}}{F_s} \qquad (5.29)$$

where f_{est} is the mean instantaneous frequency estimate in Hz.

5.4.3 Filter trajectories

If magnitude correction is performed, where the mean instantaneous centre frequencies of more than one filter coincide it is only necessary for one of those filters to be used to model the underlying component. This is analogous to the situation in the Fourier case where only the corrected magnitude from a peak bin for a sinusoid is used to estimate the magnitude of that sinusoid. This is because the correction restores energy to the estimate that has been smeared into other bins and the magnitudes of those other bins are ignored. The strategy here is that in the wavelet case the filters are restricted to bands and if the estimated centre frequency strays outside of that band the filter is switched off (its magnitude is set to zero). The underlying component is modelled solely by the filter 'local' to the band in which the component lies. This is slightly different to sinusoidal tracking from Fourier data, where tracks can 'roam' across bins until reaching a frame where there is no destination for the track to move to, at which point it dies. Here, where there are many fewer frequency bands in the analysis, the filter centre frequency tracks are constrained to their local bands. This is considered by this author to be a more intuitive approach for wide band residual modelling.

The frequency bands for each scale are defined as follows. The upper limit of the band for scale 1 is the Nyquist limit. The lower limit of the band for the highest scale is 0 Hz. The transition point between the bands for scales 1 and 2 is the lower -6 dB point of the wavelet filter for scale 1. Likewise the transition point between scales 2 and 3 is the lower -6dB point of the scale 2 filter, and so on. The -6 dB point is defined for the value of *f* at which:

$$\left(C(f)\right)^{n+1} = \frac{\left(C(f_0)\right)^{n+1}}{2} \Rightarrow \frac{\sin(2\pi f)}{(f-\tfrac{1}{2})\sqrt{f}} = \frac{\sin(2\pi f_0)}{2(f_0-\tfrac{1}{2})\sqrt{f_0}} \qquad (5.30)$$

This equation is transcendental and therefore no analytic solution has been found. However, using an iterative method *f* can be found within a very small margin of error. This method uses knowledge of the overall shape of the function to obtain an initial guess for *f* which is then improved iteratively by minimising the error in the calculation of (5.30). For the cubic spline case this method gives (rounded to 4 decimal places)

$$f = 0.2829 \qquad (5.31)$$

where the error in the estimation of equation (5.30) from this value is of the order of 10^{-16}. The wavelet band transition points given by this value of *f* for the first five scales are indicated by dashed lines in figure 5.26.

5.4.4 Estimation of component bandwidth

The frequency ‘splitting’ technique of [Daubechies, 1992] and its use in the wavelet packets of Coifman and Wickerhauser [Wickerhauser, 1994] was described in section 3.8.7. Splitting of the detail coefficients in this way at each scale splits the scales into lower and upper frequency bands. The exact nature of the frequency bands is, of course, dependent upon the filters used in the wavelet analysis. The splitting of a complex filter band into two further complex bands offers a means of estimating the width of the underlying signal component analysed by the original filter. At one extreme, the instantaneous mean centre frequencies of the scale filter (derived as described in section 5.2.2) and the two split filters will coincide for an impulse in the frequency domain and, at the other, their centre frequencies will be the same as those of the fixed filters for an impulse in the time domain. Therefore the proximity of the derived centre frequencies for the two complex split filters can be used to estimate the ‘narrowness’ of the underlying component.

For the undecimated transform the split at each scale is achieved by filtering of the detail coefficients at that scale. The filters are obtained by dilation of the high and low pass filters used to derive the approximation and detail coefficients by a factor of two. For the decimated transform the split can be achieved by convolution of the decimated detail coefficients with the existing filters. However this would produce fewer split than scale coefficients meaning that there could not be a one-to-one mapping of a scale coefficient to its lower and upper split coefficients. Therefore in the split implementation described in this thesis the filters are dilated and the scale coefficients left undecimated whether the split is occurring for a decimated or undecimated scale in the partially decimated transform. Where splitting is used in this way the number of detail coefficients at each scale *does* affect the computational cost of the analysis. Therefore the non-decimation of output detail coefficients to produce higher time resolution and to reduce aliasing for the decimated transform described in section 5.3 will increase the computational burden.

Splitting at a given scale is achieved by convolution of the detail signal with the low and high pass wavelet filters dilated by a factor of two from those used to generate the approximation and detail coefficients at that scale. Dilation of a filter's impulse response in the time domain is equivalent to an equal contraction of its response in the frequency domain. Therefore the frequency response of the split filters is given by:

$$\Psi_{\text{lower}}\left(\omega\right) = \Psi_{\text{scale}}\left(\omega\right) HPF_{\text{scale}}(2\omega) \quad (5.32) \text{ and}$$

$$\Psi_{\text{upper}}\left(\omega\right) = \Psi_{\text{scale}}\left(\omega\right) LPF_{\text{scale}}(2\omega) \quad (5.33)$$

The Fourier transform of the m^{th} order binomial kernel, from which the coefficients of the LPF are derived (5.1), is given by:

$$U_2^m\left(f\right) = 2\cos^{m+1}(\pi f) \qquad (5.34) \text{ [Unser et al, 1993]}$$

and that of the m^{th} order B-spline is given by (5.27). Modulation at the Nyquist limit $\left(\frac{F_S}{2}\right)$, achieved by multiplication of a discrete sequence by $(-1)^{k\in\mathbb{Z}}$ in the time domain, is equivalent to reflection about $\left(\frac{F_S}{4}\right)$ in the frequency domain. Therefore the Fourier transforms of the continuous equivalents of the low and high pass filter sequences given by (5.2) and (5.3) after dilation are given by:

$$LPF(f) = 2\cos^{m+1}(2\pi f) \qquad (5.35)$$

$$HPF(f) = 2\cos^{m+1}\left(\pi\left(2f - \tfrac{1}{2}\right)\right)\mathrm{sinc}^{2(m+1)}\left(2f - \tfrac{1}{2}\right) \quad (5.36)$$

The magnitude responses of the wavelet and these splitting filters at scale 2 are shown in figure 5.27. Figure 5.28 shows the magnitude response of the resultant filters after convolution with the wavelet filter. Figure 5.29 shows the underlying continuous time domain functions of the split wavelets. The perhaps counter-intuitive result that the upper split wavelet is produced by convolution with the LPF and the lower split by convolution with the HPF is explained by the fact that it is the reflected parts of the filters' frequency responses (i.e. their responses above $\frac{F_s}{4}$) that coincide with the region where the response of the wavelet filter is greatest. As would be expected of the dilation and convolution operations of the splitting operations the split wavelets have greater time support but are more localised in frequency than the parent wavelet.

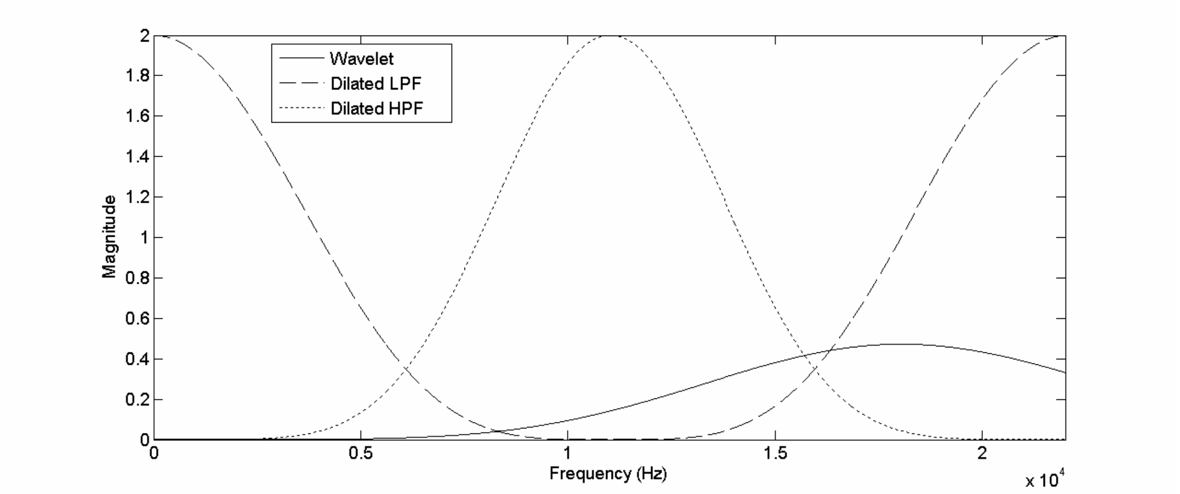


Figure 5.27: Cubic spline wavelet and splitting filters at scale 1.

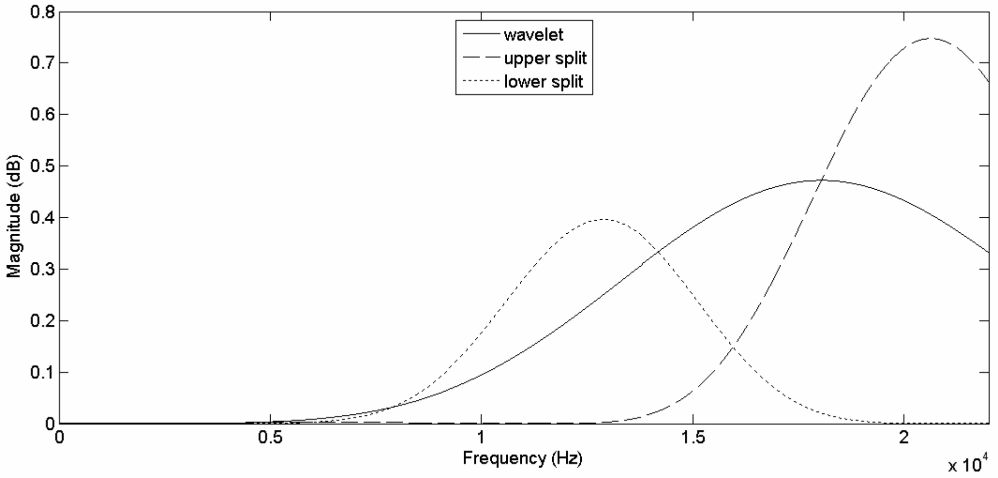


Figure 5.28: Magnitude responses of the real cubic spline wavelet and lower and upper split wavelets at scale 1.

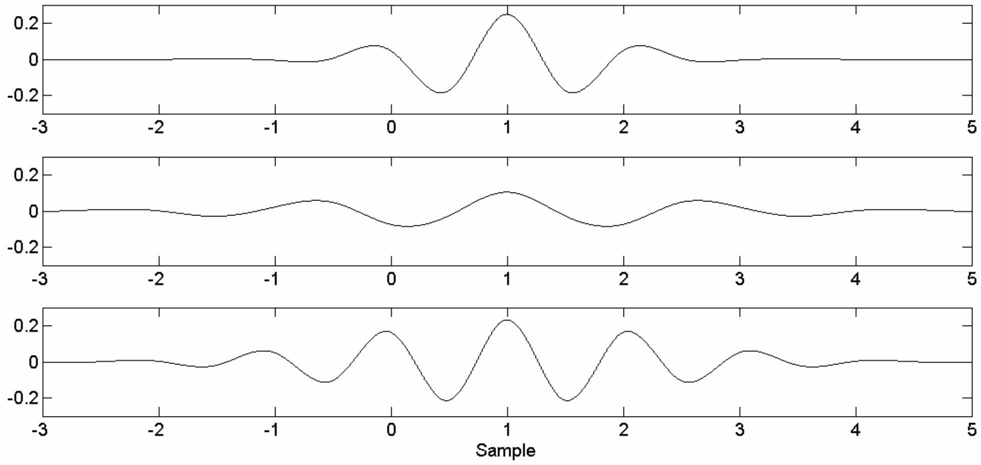


Figure 5.29: Cubic spline wavelet (top) , lower split wavelet (middle) and upper split (bottom) functions at scale 1.

The computational cost of the partially decimated split wavelet transform where circular convolution is employed is double that of its non-split counterpart. As discussed in section 5.3.3 linear convolution is used in this implementation of wavelet analysis. The cost of the split transform is the cost of the un-split transform, given by equations (5.17) to (5.22), plus the cost of filtering that produces the splits at each scale. The split at each scale is achieved by high pass filtering of the detail coefficients at that scale followed by high and low pass filtering with filters which are dilated by a factor of two from those used to produce the approximations and details at that scale. For the undecimated transform sequence length, Ns_j (the *s* indicates 'split'), before the high and low pass split filtering is given by:

$$Ns_j = \left(N + \left(L_{LPF} - 1\right) \sum_{n=1}^{j-1} 2^{n-1} \right) = N_j + 2^j \left(L_{HPF} - 1\right) \quad (5.37)$$

where N_j is the input length to the splitting stage which is calculated using (5.17). Therefore the combined cost of all the splitting stages is given by:

$$\begin{aligned} Cs &= \left(L_{LPF} + L_{HPF}\right) \sum_{j=1}^{J} Ns_j = \left(L_{LPF} + L_{HPF}\right) \sum_{j=1}^{J} N_j + 2^{j-1} \left(L_{HPF} - 1\right) \\ &= \left(L_{LPF} + L_{HPF}\right) \left(NJ + \left((L_{LPF} - 1)(2^J - 1 - J)\right) + \left(L_{HPF} - 1\right) \sum_{j=1}^{J} 2^{j-1} \right) \quad (5.38) \\ &= \left(L_{LPF} + L_{HPF}\right) \left(NJ + \left((L_{LPF} - 1)(2^J - 1 - J)\right) + \left(L_{HPF} - 1\right)\left(2^J - 1\right) \right) \end{aligned}$$

And the total cost of the transform is given by adding the result of (5.14) and (5.24). For the decimated transform the sequence prior to splitting is the detail sequence at that scale. The length of this is given by length of the convolution of (5.15) with the HPF:

$$Ns_j = \left\lceil 2^{-(j-1)} \left(N - L_{LPF} + 1\right) + L_{LPF} + L_{HPF} - 2 \right\rceil \qquad (5.39)$$

and the total cost of the splitting stage of the decimated transform is given by adapting (5.16):

$$C = \left(L_{LPF} + L_{HPF}\right) \left(J \left(L_{LPF} - 1\right) + \left(N - L_{LPF} + 1\right)\left(2 - 2^{-(J-1)}\right) + J \left(L_{HPF} - 1\right) \right) \quad (5.40)$$

The total cost of the split decimated transform is given by adding (5.20) and (5.40). The cost of the splitting stage of the partially decimated transform can be calculated for the undecimated levels by the addition of (5.20) and (5.40), as above. The cost of splitting at the decimated levels is given by a modification of (5.40)

$$C = \left(L_{LPF} + L_{HPF}\right) \left(J \left(L_{dLPF} - 1\right) + \left(N_{undec} - L_{dLPF} + 1\right)\left(2 - 2^{-(J-1)}\right) + J \left(L_{dHPF} - 1\right) \right) \quad (5.41)$$

where N_{undec} is defined as for (5.21), L_{dLPF} is given by (5.22) and L_{dHPF} is given by:

$$L_{dHPF} = \left(L_{HPF} - 1\right) 2^{U-1} + 1 \qquad (5.42)$$

Finally the total cost of the partially decimated split wavelet transform is given by adding (5.21) and (5.41). Figure 5.30 shows the computational cost of the split and un-split, decimated and undecimated complex cubic spline wavelet transforms for linear convolution.

It can be seen that the cost of the undecimated split transform grows rapidly with increasing number of scales and even for just three scales it is close to the 8192 point FFT in this respect. Figure 5.31 illustrates the cost of a ten scale partially decimated real cubic spline wavelet transform with different ratios of undecimated to decimated scales. Only two undecimated scales are available if the cost is kept under that of an 8192 point FFT.

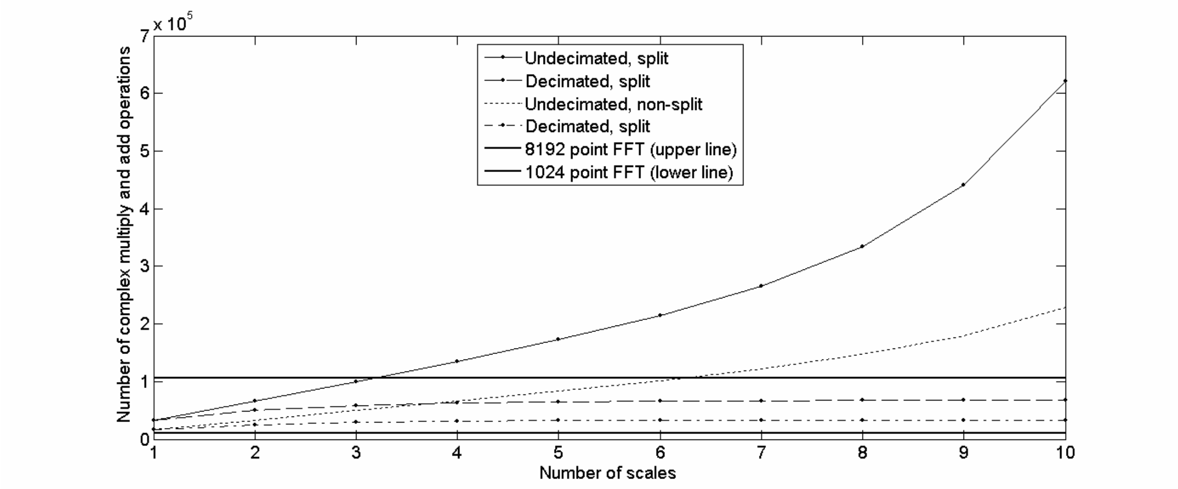


Figure 5.30: Computational cost of various types of complex cubic spline wavelet transform for an input length of 1025 samples. Linear convolution is used for the wavelet filtering. Example FFT costs are included for comparison, the cost of generating the analytic signal is not considered.

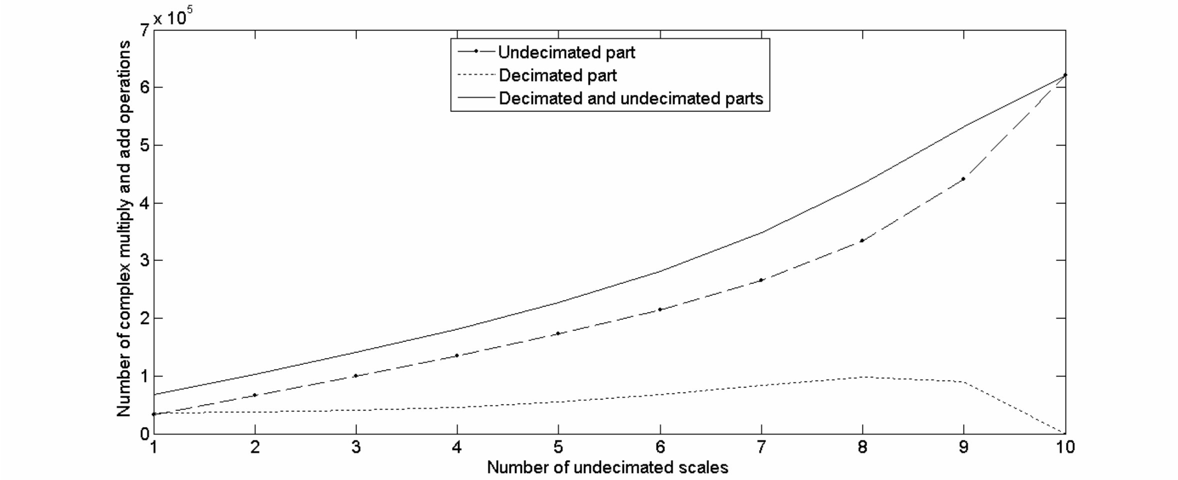


Figure 5.31: Computational cost of the partially decimated split real cubic spline wavelet transform. The number of undecimated scales is given. The number of decimated scales is given by the difference between the total number of scales (10 in this case) and the number of undecimated scales. The input is 1025 samples long. The cost of generating the analytic signal is not considered.

The purpose of split transform is to use the two additional mean instantaneous frequency estimates from each of the ten scales to estimate the width of the component in each of their bands. For example, where the underlying component is a sinusoid the measured width from an ideal analysis would be 0 Hz whereas for an impulse it would be the width of the entire band. For an impulse the peak frequency of the two wavelet split filters would be the frequency for which the split functions shown in figure 5.28 peak. These frequencies can be identified by finding the value at which the first derivative is zero (within the frequency region of the peaks so as to avoid solving for a side lobe as opposed to the main lobe). This can be done for B-spline wavelets of any order by considering the zeroth order case. The peak frequency of the upper wavelet is given by solving:

$$\begin{aligned}&\frac{2\sin\left(2\pi f_{\text{peak}}\right)}{\pi^2 f\left(2f-1\right)^2}\cdot\\&\left(\frac{4\cos^2\left(2\pi f_{\text{peak}}\right)\left(2\pi f^2-\pi f\right)-\sin\left(2\pi f_{\text{peak}}\right)\cos\left(2\pi f_{\text{peak}}\right)\left(6f-1\right)}{\pi f\left(2f-1\right)}-2\sin^2\left(2\pi f_{\text{peak}}\right)\right) \quad (5.43)\\&=0\end{aligned}$$

and the peak frequency of the lower wavelet is given by solving:

$$\begin{aligned}&\frac{2\sin\left(2\pi f_{\text{peak}}\right)}{\pi^3 f_{\text{peak}}\left(2f_{\text{peak}}-1\right)^2}\cdot\\&\left(\begin{array}{l}2\sin\left(2\pi f_{\text{peak}}\right)\operatorname{sinc}\left(2f_{\text{peak}}-\tfrac{1}{2}\right)\left(\begin{array}{c}\dfrac{\pi\cos\left(\pi\left(2f_{\text{peak}}-\tfrac{1}{2}\right)\right)-\sin\left(\pi\left(2f_{\text{peak}}-\tfrac{1}{2}\right)\right)}{\pi f_{\text{peak}}{}^2}\\-2\sin\left(\pi\left(2f_{\text{peak}}-\tfrac{1}{2}\right)\right)\cos\left(\pi\left(2f_{\text{peak}}-\tfrac{1}{2}\right)\right)\operatorname{sinc}\left(2f_{\text{peak}}-\tfrac{1}{2}\right)\end{array}\right)\\+\left(\dfrac{\cos\left(\pi\left(2f_{\text{peak}}-\tfrac{1}{2}\right)\right)\sin^2\left(2f_{\text{peak}}-\tfrac{1}{2}\right)}{f_{\text{peak}}\left(2f_{\text{peak}}-1\right)}\left(\begin{array}{l}4\pi f_{\text{peak}}\cos\left(2\pi f_{\text{peak}}\right)\left(2f_{\text{peak}}-1\right)\\-\sin\left(2\pi f_{\text{peak}}\right)\left(6f_{\text{peak}}-1\right)\end{array}\right)\right)\end{array}\right)\\&=0\end{aligned}$$

(5.44).

These are both transcendental and so no analytic solution exists. However f_{peak} can be estimated with very small error using an iterative method similar to that used to solve (5.30). This gives values for f_{peak} (expressed as multiples of $F_s/2^{j-1}$ where *j* is the scale) of 0.2919 and 0.4678 for the lower and upper split filters respectively, to four decimal places. Therefore the maximum difference (i.e. that due to an impulse) between split filters at scale *j* is given by:

$$\Delta f = \frac{0.1760 F_s}{2^{j-1}} \qquad (5.45)$$

Figure 5.32 illustrates how differences between frequency estimates at a single scale occur where a component has spectral breadth. The frequency estimates at scale 1 for the wavelet and its splits are shown for a sinusoid and for a single impulse which occurs in the middle of the frame (sample 512). There is a clearly visible difference in estimates for the impulse whereas, at the same scaling of the vertical axis, there is no difference in estimates for a stationary sinusoid.

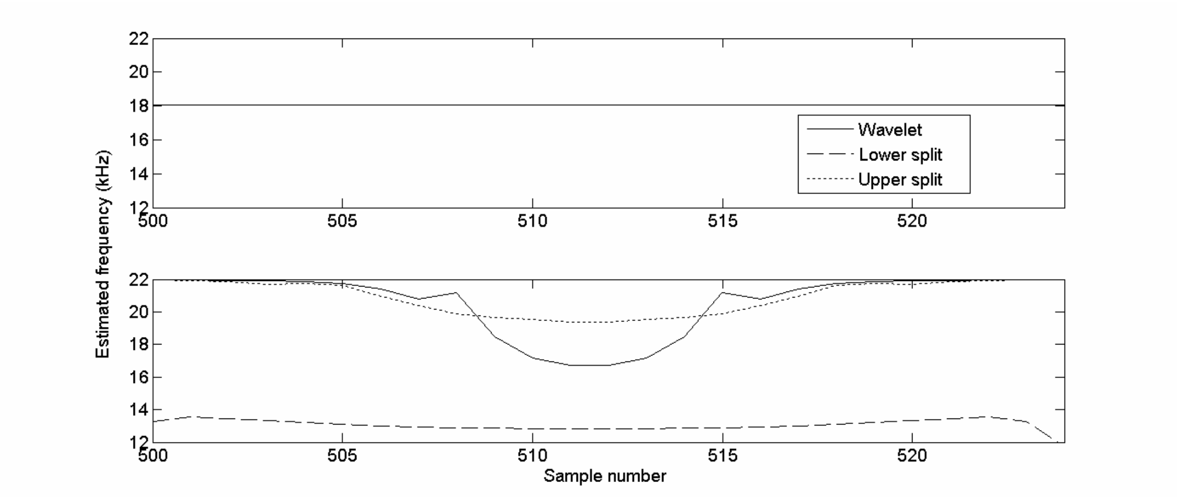


Figure 5.32: Frequency estimates at scale 1 for a sinusoid (top) and an impulse at the centre of the frame (bottom). The frequency of the sinusoid is the centre frequency of the wavelet (18.045 kHz).

5.5 Parametric synthesis equalisers

The purpose of the analysis described thus far in this chapter is to provide control data for a bank of parametric equalisers which are applied to a white noise source. Having obtained an estimate for component width in the previous section this value is used to control the width of the *synthesis equaliser*. *Parametric equalisers* are common tools in music studios. Although the term equaliser is something of a misnomer, since these devices are usually used to 'un-equalise' a signal for creative effect in a studio context, the term is retained here since it is common terminology in audio processing for music applications. The term parametric in this context implies that the user is offered control of centre frequency, gain and bandwidth (or Q). Bandwidth and Q are related by:

$$Q = \frac{f_{\text{centre}}}{f_{\text{high cut off}} - f_{\text{low cut off}}} \qquad (5.46) \text{ [Roads, 1996]}$$

where f_{centre} is the centre frequency of the equaliser, $f_{\text{high cutoff}}$ and $f_{\text{low cutoff}}$ are the high and low 'cut-off' frequencies respectively and their difference is the bandwidth. Definitions of what constitutes cut-off vary in the literature but it is usually defined as the point at which the modulus of the equaliser gain reaches some fraction of the modulus of the centre gain (which, when given in dB, is positive for a peak equaliser and negative for a notch equaliser). Different definitions of cut-off are surveyed in [Bristow-Johnson, 2004]. The most complete specification of equaliser bandwidth is to specify the cut-off frequencies *and* the cut-off gain relative to the centre gain such as is given in an implementation in [Moorer, 1983]. In the context of this thesis the bandwidth is given by the difference between estimated frequency of the upper and lower wavelet splits. The gain at the cut-off points can then be adapted to suit the equalisers employed at the synthesis stage and can be offered as a user-adjustable parameter.

The design of digital parametric equalisers is not a research focus of this thesis but a brief overview of current practice follows. Further detail may be found in the references cited. Many digital equalisers are derived from a second-order analogue prototype filter with the transfer function:

$$H_{\text{analogue}}(s) = \frac{s^2 + as + \omega_0^2}{s^2 + bs + \omega_0^2} \qquad (5.47)$$

where ω_0 is the centre frequency and a and b are coefficients that are defined by the desired shape and centre frequency gain of the filter. The gain of the filter at 0 and ∞ is 1 (0 dB) and is $\frac{a}{b}$ at the centre of the peak (or notch) [White, 1986]. A digital version of this filter can be obtained via the bilinear substitution:

$$s = \frac{1 - z^{-1}}{1 + z^{-1}} \qquad (5.48).$$

Substituting this into (5.47) and 'warping' the analogue frequency gives[2]:

[2] Warping is required to compensate for the nonlinear mapping from the *s* plane to the *z* plane [Rorabaugh, 1999].

$$H(z)=\frac{1+\gamma\sqrt{G}-2\cos\left(\Omega_0\right)z^{-1}+\left(1-\gamma\sqrt{G}\right)z^{-2}}{\left(1+\gamma/\sqrt{G}\right)-2\cos\left(\Omega_0\right)z^{-1}+\left(1-\gamma/\sqrt{G}\right)z^{-2}} \quad (5.49)$$

where

$$\gamma=\frac{\sin\left(\Omega_0\right)}{2Q} \quad (5.50)$$

and where G is the gain (expressed in linear terms as opposed to dB), Q is the filter Q, and Ω_0 is the angular frequency expressed divided by the sample rate [Bristow-Johnson, 1994], [Bristow-Johnson]. The cut off is defined as the point at which the gain is half the centre gain (this time expressed in dB). This gives a computationally cheap digital equaliser however it does not perfectly emulate the analogue prototype described by (5.47). Because the bilinear transform maps the Nyquist frequency onto $s=\infty$ the digital equaliser is forced to unity gain at this frequency. A digital equaliser without this constraint has more recently been developed and is described in [Orfanidis, 1997]. Here the gain of the analogue prototype at ∞ can be set to a value other than 1. This value is set to the frequency of the target analogue filter as it crosses the Nyquist limit. This gain (G_1) is given by:

$$G_1=\sqrt{\frac{G_0^2\left(\omega_0-\pi^2\right)+G^2\pi^2\left(\Delta\omega\right)^2\left(G_B^2-G_0^2\right)/\left(G^2-G_B^2\right)}{\left(\omega_0-\pi^2\right)+\pi^2\left(\Delta\omega\right)^2\left(G_B^2-G_0^2\right)/\left(G^2-G_B^2\right)}} \quad (5.51)$$

where G_0 is the gain at 0 Hz (set to unity for this application), G_B is the gain at the lower and upper cut-off frequencies and $\Delta\omega$ is the bandwidth in radians per sample. The transfer function of the filter is given by:

$$H(z)=\frac{\left(G_1+G_0W^2+B\right)-2\left(G_1-G_0W^2\right)z^{-1}+\left(G_1+G_0W^2-B\right)z^{-2}}{\left(1+W^2+A\right)-2\left(1-W^2\right)z^{-1}+\left(1+W^2-A\right)z^{-2}} \quad (5.52)$$

where:

$$W^2=\sqrt{\frac{G^2-G_1^2}{G^2-G_0^2}}\Omega_0^2 \quad (5.53)$$

$$A = \sqrt{\frac{C + D}{\left|G^2 - G_B^2\right|}} \qquad (5.54)$$

$$B = \sqrt{\frac{G^2 C + G_B^2 D}{\left|G^2 - G_B^2\right|}} \qquad (5.55)$$

where:

$$C = \left(\Delta\Omega^2\right)\left|G_B^2 - G_1^2\right| - 2W^2\left(\left|G_B^2 - G_0 G_1\right| - \sqrt{\left(G_B^2 - G_0^2\right)\left(G_B^2 - G_1^2\right)}\right) \qquad (5.56)$$

$$D = 2W^2\left(\left|G^2 - G_0 G_1\right| - \sqrt{\left(G^2 - G_0^2\right)\left(G^2 - G_1^2\right)}\right) \qquad (5.57)$$

Clearly the computational cost of calculating the coefficients of the Orfanidis equaliser is higher than that described by (5.49) and (5.50). However, the advantage of the Orfanidis filter is that its behaviour is much closer to that of the analogue prototype, particularly close to the Nyquist limit. This is demonstrated in figure 5.33 where magnitude frequency plots are shown for the analogue filter and the two digital derivations from it. Here the centre frequency is 15 kHz, the gain at this point is 12 dB, the bandwidth is 5 kHz (the cut-off is defined here as 6 dB below centre frequency gain) giving a Q of 3. Despite the improvement this is still not a perfect fit to the analogue prototype since the first derivative of the Orfanidis filter at the Nyquist limit is zero, which is not the case for the analogue filter. Therefore, for centre frequencies and/or wide bandwidths close to the Nyquist limit the behaviour is different to that of the filter at lower frequencies. However this situation is an improvement over the alternative case where the magnitude response is forced to unity at this point, resulting in unwanted attenuation of high frequency components.

Figure 5.34 shows a bank of ten Orfanidis equalisers each centred an octave apart at the spline wavelet centre frequencies for each scale, each with a bandwidth equal to the upper-minus-the-lower split frequency, as would be measured between the wavelet splits for an impulse. The gain of the summed responses is also shown in the figure. The gain at the cut-off points is set here to –3dB since the equalisers are intended for use with stochastic signals with incoherent phase. The amplitude of the ripple in the magnitude response is approximately 6 dB and there is a gradual lift in the highest octave due to the non-uniform behaviour of the equaliser close to the Nyquist limit. The upper octave lift can be countered by scaling the bandwidth of this equaliser although this sharpens the peak producing a higher amplitude local ripple. A reasonable compromise between the two has been found to be a scaling factor of 0.9 for the upper octave bandwidth. The amplitude of the ripple can be reduced by increasing the bandwidth or by raising the cut-off gain relative to the peak gain. Setting the bandwidths of each equaliser so that upper and lower bands intersect at each other's cut-off frequency, given by:

$$f_{\text{high cut-off}} = \sqrt{2}\, f_{\text{centre}} = \frac{f_{\text{centre}}^{2}}{f_{\text{low cut-off}}} \qquad (5.42)$$

and increasing the cut-off gain to be 1 dB below the peak gain reduces the ripple amplitude to less than 1 dB but this is at the cost of increased interaction between synthesis bands and too broad a response for narrow band components such as sinusoids. At the time of writing this thesis new work relating biquadratic digital filters to high-order Butterworth, Chebyshev and elliptic analogue prototype filters has been published [Orfanidis, 2005]. Such filters offer flatter pass bands and sharper transitions than the filters described in this section and therefore may be more appropriate candidates for residual synthesis equalisation.

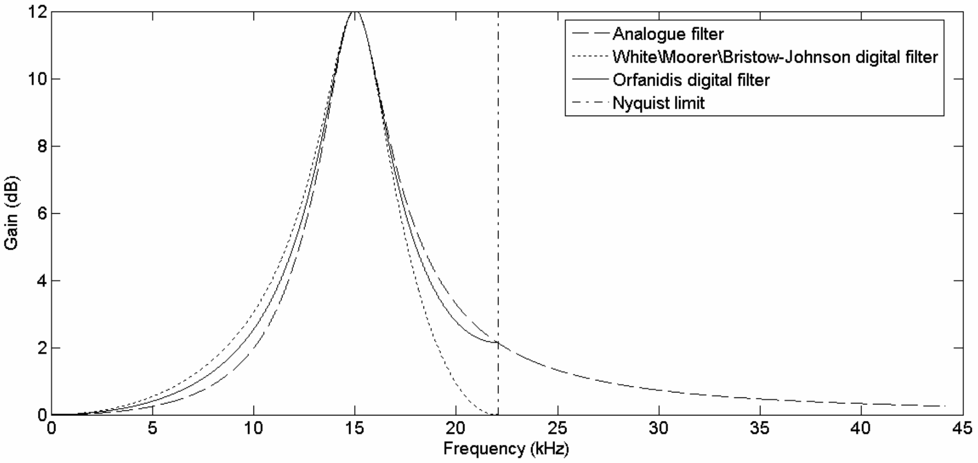


Figure 5.33: Magnitude response for a second order analogue filter and two digital derivations (after [Orfanidis, JAES, 1997]). The *Fs* = 44.1 kHz case for the digital filters is shown.

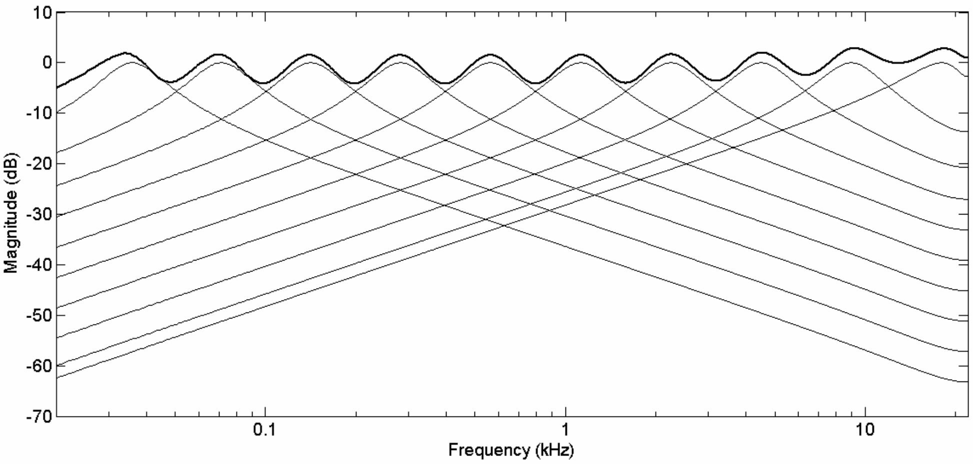


Figure 5.34: Magnitude versus log frequency for a bank of ten octave spaced Orfanidis parametric equalisers. The centre frequencies are those of the complex wavelet analysis filters for an impulse and their -3 dB points are the measured difference between the complex split wavelets.

The use of the measured difference between split frequencies in figure 5.34 has been given as an example and there is no direct link between this measurement (derived from symmetric FIR filters) and the bandwidth of the IIR filters used here. In the current implementation of this analysis synthesis system a user defined linear scaling is applied to the measured split difference and this, along with the control over cut-off gain, determines the relationship between the difference between the split analysis filters and the shape of the synthesis filters.

The estimated magnitudes are derived from analysis filters which have a fixed bandwidth. Therefore these magnitudes must be scaled to match the bandwidth of the synthesis equalisers so that they pass the correct amount of energy. The energy per unit gain passed by the equalisers described here increases approximately linearly with bandwidth. Energy is proportional to the square of the magnitude therefore the equaliser gain is scaled in inverse proportion to the square root of the bandwidth in order to pass the same energy.

5.6 Conclusions

The previous chapter of this thesis described methods for the frame-by-frame identification and description of non-stationary sinusoids in an audio signal. Once these have been synthesized and subtracted from the original signal a residual signal ideally comprising only broad band signal components remains. This chapter has described a system for modelling an audio signal with cubic B-spline wavelets. These wavelets have been shown in the literature to have excellent time-frequency localisation properties as well as offering good edge detection due to their compact support. There are far fewer analysis filters than for a Fourier analysis of the same length of frame but they are constant-Q, having good time resolution at low scales and good frequency resolution at high scales. Therefore it is proposed to utilise this analysis method on the residual signal in chapter 6. This is the ‘subtractive synthesis’ complement to the ‘additive synthesis’ of the chapter 4 in the heterogeneous spectral modelling system described in the following chapter.

Although the B-spline transform offers an efficient implementation of Morlet wavelet analysis it is costly in its complex undecimated form compared to the Fourier transform. This cost is increased by the use of linear, as opposed to circular, convolution as is desirable for short-time wavelet analysis. A partially decimated transform has been described which offers mediation between low computational cost and undesirable analysis artefacts such as frequency aliasing and shift variance. Comparisons of the cost of various types of transform described in this chapter have been presented. Whilst the undecimated transform offers the best representation, the redundancy in the complex decimated transform with the option to only decimate some levels offers good shift invariance and frequency estimation properties as well as adaptability to the processing constraints of the host system.

Finally, an overview of computationally cheap IIR parametric equalisers for re-synthesis of the residual has been given along with a discussion of how the analysis parameters relate to those of the equalisers. Two types of existing digital equaliser have been presented along with their relative merits. User adjustable parameters, in addition to those automatically controlled by the analysis, have been suggested.

Non-real time partial tracking for offline or real-time oscillator bank resynthesis has been the subject of a great deal of research throughout the last thirty years. Chapter 4 of this thesis has built on and extended knowledge in the specific area of non-stationary identification and description of sinusoids for frame-by-frame tracking. The general application of wavelets to audio residual modelling for control of equalisers presented in this chapter is entirely novel and is, therefore, less mature. In particular future work is needed to better relate the wavelet split frequency difference to the underlying component bandwidth. For example, with a better understanding of how these two are related the magnitude correction of section 5.4.2 can take account of the component width in addition to the measured magnitude. In addition to such further research, investigations into audio processing areas outside of the specifics of short-time residual modelling may well yield promising applications of such an approach to analysis and synthesis of audio.

Much of the new work presented here has related to the extension of the B-spline wavelet transform to time-frequency analysis and analysis of its behaviour for simple short-term signals such as stationary sinusoids and impulses. The analysis and resynthesis system described in the next chapter offers an opportunity to see how it performs on ‘real world’ signals.

6 A FRAME BY FRAME SPECTRAL MODELLING SYSTEM

6.1 Introduction

The previous two chapters of this thesis have described new techniques for using Fourier and wavelet analysis to produce non-stationary sinusoidal and noise based models of sound. This chapter puts these new techniques into a practical, application-based context by describing a frame by frame spectral modelling system which uses the sinusoidal identification and description methods described in chapter 4 for the deterministic part of the signal and the complex wavelet modelling described in chapter 5 for the residual part. The proposed system uses time-variant sinusoidal oscillators for synthesis of the deterministic part of the signal and a noise source filtered by time-variant parametric equalisers for synthesis of the residual. This 'additive plus subtractive' synthesis method offers a model which is familiar to many musicians since it is rooted in the well known and understood audio studio techniques for the direct generation and modification of audio spectra: additive synthesis and parametric equalisation. Since this model offers direct access to the mean instantaneous frequency of all its spectral components shifting the pitch of any or all of those components, independently of each other if required, is a trivial matter. This type of model also lends itself to other creative sound transformations such as cross synthesis and control over higher level sound attributes such as 'noisiness' and spectral spread.

A primary motivation for this research has been to investigate the possibility of quasi real-time spectral modelling (as opposed to processing) of audio. Quasi-causality is addressed by the 'frame by frame' method for separating sinusoidal components from the residual. The quality of sinusoidal/residual separation for a number of different sounds, of both acoustic and synthetic origin, is evaluated. To assess how the execution time constraint can be negotiated in a real-time system the computational cost of different algorithms within the system are discussed and quality-cost tradeoffs described. All of the files used to produce the MATLAB implementation of this frame by frame spectral modelling system are provided in Appendix A which is a CD-ROM. Results for this implementation and system testing have been performed on a PC using a 1.6 GHz Pentium M processor with 1 GB of RAM.

6.2 System design goals

The primary goal of this spectral modelling system is indicated by its name: *reSynth*. That goal is the frame by frame output of high-fidelity audio imitations of input sound signals. These imitations are generated by digital additive and subtractive synthesis using the model parameters derived from time-frequency/scale analysis data. The model is important since it is this which provides the parameters for sound transformation and the fidelity of imitations is important since this demonstrates the suitability of the model to the input sound and the quality of the analysis techniques employed. Even though the analysis happens on a frame by frame basis the system is designed so that synthesis occurs on a sample by sample basis so that the temporal location of events is as indifferent to frame boundary locations as possible.

A secondary goal is that, in computational terms, the system is efficient and offers some mediation between resynthesis quality and computational cost so that it can be used in a real-time context such as a traditional recording studio or live performance. Since the processing power and amount of RAM within computer systems is constantly increasing a system which satisfies the causal but not the 'execution in time' constraint for real-time processing on currently available hardware still offers the potential for a future real-time implementation.

As is the case for many other spectral modelling systems, such as SMS, *reSynth* is designed for use with monophonic (in the 'single note' sense) input sounds with a minimum spacing between individual sinusoidal components determined by the frequency resolution of the analysis. The system is designed for use with CD quality audio since for many applications this offers sufficient fidelity and output sounds are ready for distribution in a widely accepted digital audio format without further processing or format conversion.

Taking the lower frequency limit of human hearing to be 20 Hz which corresponds to a period of 50 ms an obvious choice of frame length in a spectral analysis system for processing audio sampled at 44.1 kHz is 2205 samples. However this introduces a clearly perceptible delay between input and output which does not provide a quasi real-time processing experience. Much shorter frame lengths offer much lower latency but at the cost of increased minimum spacing between sinusoidal components. It has been proposed that the maximum acceptable time gap between an input gesture and a computer system's response to it is as low as 10 ms [Wessel and Wright, 2002] however this gives a minimum spacing between components of over 400 Hz which would be insufficient for a wide range of sounds with closely spaced partials, particularly those with a low fundamental frequency. The compromise chosen here, and adopted for the results presented in the previous two chapters, is a frame length of 1025 samples (23 ms) which gives a minimum spacing of just over 100 Hz. There is a perceptible delay between gesture and outcome at this latency which manifests itself as a 'softness' or 'spongeyness' but not as a dislocation sufficient to seriously disrupt the relationship between cause and effect for an inexperienced user[1]. The modelling techniques presented in this thesis can be easily adapted to different frame lengths, although the effectiveness of the model for different sounds will vary with frame length.

In fact the processing delay is determined by both the analysis and synthesis frame lengths. For ‘transform followed by inverse transform’ processing the delay between input and output is the analysis frame length, since the output frames overlap, so the output and input frame lengths are equal regardless of the overlap factor. In a system such as *reSynth* where there is no overlapping of synthesis frames their length is a function of the analysis frame length and the overlap:

$$N_{\text{synthesis}} = \left\lfloor \frac{N_{\text{analysis}}}{O} \right\rfloor \qquad (6.2)$$

where $N_{\text{synthesis}}$ and N_{analysis} are the lengths, in samples, of the frames and O is the overlap factor. This reduces the delay between input and output since, where the overlap factor is greater than 1, the synthesis frame (which can be output immediately after the current input frame has been analysed) does not begin at sample 1 of the corresponding analysis frame but somewhere nearer the centre. For example, with a frame length of 1025 and an overlap of 2, the synthesis frame will be 512 samples long with its first sample corresponding to sample number 256 in the analysis frame and its last sample corresponding to sample 768 in the analysis frame. Thus the delay is reduced from 1025 samples to 769 samples. This example is illustrated in figure 6.1.

[1] Musical performers can adapt to latency between gesture and audio output. A common example of this is pipe organs which can have detached consoles far removed, 100 metres in some cases, from the sound production mechanisms. Inexperienced players will often involuntarily increase their playing speed since they feel that the latency is in their gesture not the sound production, whereas a performer who has *adapted* to the specific instrument can learn to play with a steady tempo despite a large time lag.

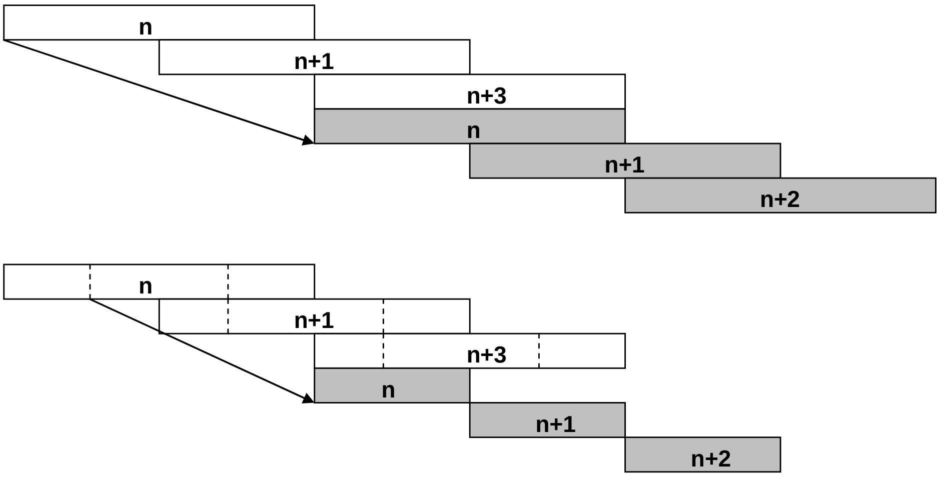


Figure 6.1: Temporal relationship between contents of overlapping analysis frames and their corresponding overlapping synthesis frames (top) and their non-overlapping synthesis frames (bottom). The analysis frames are shaded white, the synthesis frames grey. In the bottom example (the *reSynth* case) there is a smaller delay between input and output samples as shown by the arrows.

This system has been tested and developed in MATLAB [Mathworks, 2006]. The system is described by a single '.m' file with a number of sub-functions within it to aid readability. Whilst MATLAB offers excellent design and testing tools, routines written as m files do not execute as fast as identical routines written in a lower level language like C. In order to improve the execution speed of *reSynth* bottleneck sub-functions have been re-written in C as MEX (MATLAB executable) files however their 'm' counterparts have been retained in the code listing to aid understanding of how they work. MEX ('MATLAB executable') files are pre-compiled files (unlike text based m files) which can be written in C or FORTRAN. The compiled files are dynamic link libraries (DLL files). Some of the built-in functions in MATLAB (such as convolution) have also been reproduced as MEX files. The C code used to produce the MEX files is separately presented. These MEX files greatly reduce the execution time of *reSynth* for a given input. Many of these MEX files offer a hundred fold or greater increase in execution speed.

6.3 System overview

There are three stages in this audio process: analysis to produce model parameters, interaction with those parameters to produce a sound transformation and resynthesis to render the transformation as audio. The main parts of each of these stages are, in order:

Analysis:

- Zero-padded time and frequency reassigned FFT analysis of input.

- Parameter estimation and evaluation of sinusoidality of magnitude spectrum peaks by evaluation of behaviour of time and frequency reassignment data at and around peaks.

- Removal of sinusoidal peaks from magnitude spectrum.

- Linking of partial trajectories between current and previous frame.

- Production of time domain analytic residual signal from inverse transform of Hilbert transformed residual magnitude and phase spectra.

- Complex B-spline wavelet analysis of residual and estimation of parameters of underlying components.

Modification:

- Direct manipulation of one or more of: frequency, magnitude (sinusoids and residual components) and bandwidth (residual components).

Resynthesis:

- Individual sinusoidal oscillators produce starting, ending or continuing sinusoids depending upon the outcome of the linking process.

- Parametric equalisers filter a noise signal. The coefficients for a given equaliser are updated every time a new centre frequency and Q estimate are output from the analysis for the frequency/scale band that it covers.

- The outputs of each sinusoidal oscillator and parametric equaliser are added to produce the *reSynth* output.

The following sections describe each of these stages in more detail.

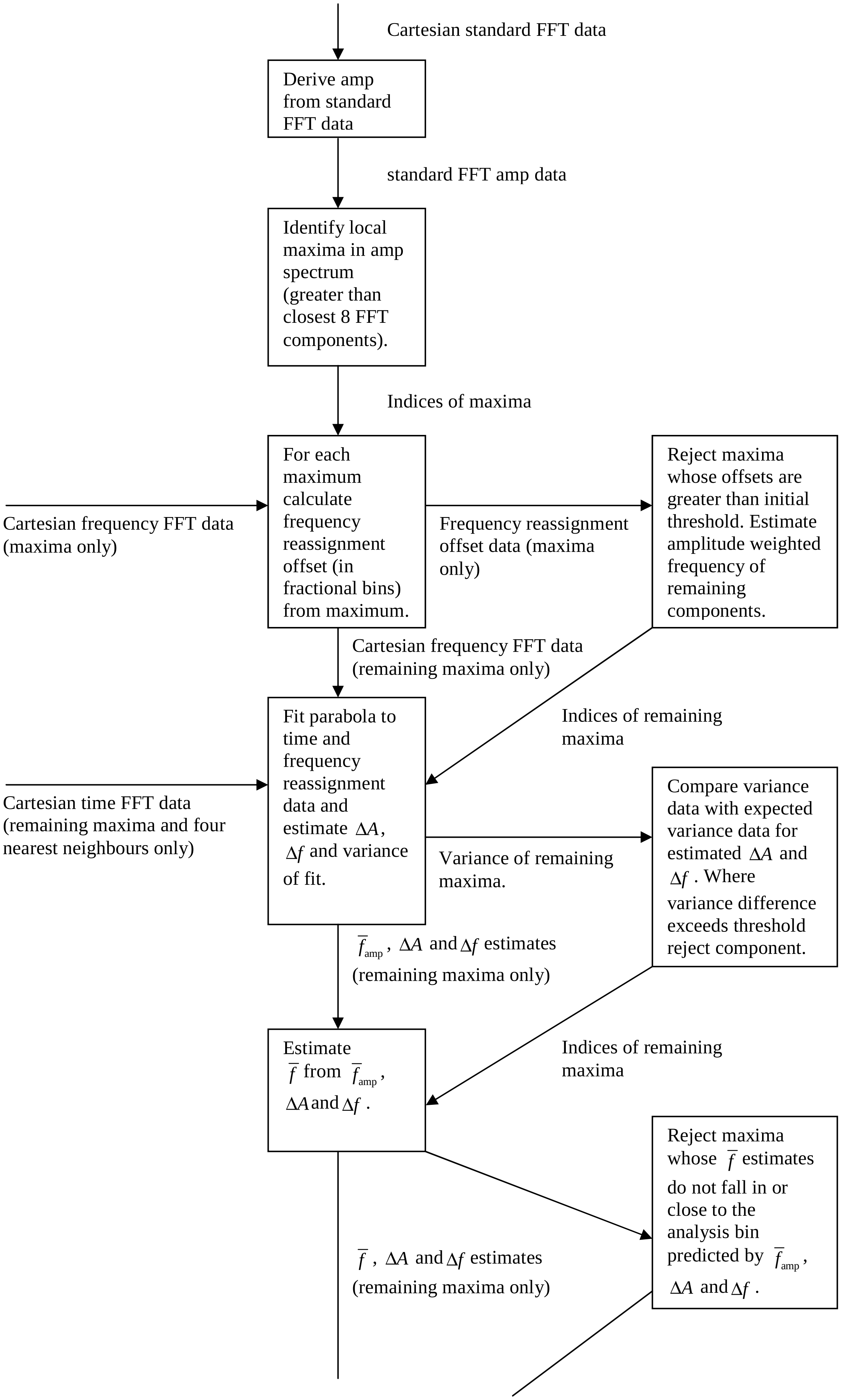
6.4 Sinusoidal identification and extraction

The sinusoidal identification and description methods employed in *reSynth* are described in detail in chapter 4. This section describes how this general method has been applied in this specific process which is designed to obtain the best model data for the least number of calculations. These are minimised by using tests of sinusoidality first which are simpler and will lead to the rejection of a relatively large number of components before further computational effort is expended on them by estimating parameters such as amplitude and frequency change.

The analysis begins with windowing of the data for computation of the reassigned FFT. Odd length zero-phase windows are used to provide consistent phase estimates and these are zero-padded from 1025 to 8192 samples to provide data sufficiently close to sinusoidal peaks for estimation of non-stationary parameters. A second reason for zero-padding is for greater consistency in the spectral subtraction process which is discussed later in this section. As stated previously, the Hann window is used since it offers an appropriate compromise between main lobe width (which partly determines minimum component spacing) and relative side lobe levels. Also it has side lobes whose peak level reduces monotonically with increasing distance from the peak. This means that side lobes cannot be mistaken for main lobes when using local maxima criteria to search for sinusoids. Once the three FFTs are complete (using the Hann and its time and frequency ramped versions) the analysis proceeds as shown in the diagram overleaf. The following subsections describe aspects of this analysis scheme.

6.4.1 Calculating amplitudes of standard FFT data

Phase is only used for new sinusoids (i.e. those which are born in the current frame). Since the zero-phase windowing provides an estimate of the phase at the centre of the frame, the start phase of a starting sinusoid is extrapolated from the centre to the beginning of the synthesis using its $\overline{f}$ and Δf estimates. Continuing and ending sinusoids inherit their start phase from the sinusoid in the previous frame to which they are linked. The start phase of these sinusoids is the end phase of the previous sinusoids’ plus the phase increment occurring over one sample period at the start frequency of the new frame. Therefore it is only necessary to calculate a component’s phase if it has been confirmed as a starting sinusoid, significantly reducing the number of arctangent calculations required in each frame. However the amplitude of every bin must be calculated since all bins are considered when searching for local maxima. This analysis system considers a bin whose amplitude is greater than its eight closest neighbours to be a local maximum. During transient portions of a sound there may be a number of short-lived (heavily damped) sinusoids whose relative phases contain important information about the temporal evolution of the sound. In this model this phase information is retained for single frame sinusoids but discarded if the sinusoid continues into subsequent frames since smooth continuation of the sinusoid across frame boundaries is more important.


Cartesian standard FFT data
Derive amp from standard FFT data
standard FFT amp data
Identify local maxima in amp spectrum (greater than closest 8 FFT components).
Indices of maxima
Cartesian frequency FFT data (maxima only)
For each maximum calculate frequency reassignment offset (in fractional bins) from maximum.
Frequency reassignment offset data (maxima only)
Reject maxima whose offsets are greater than initial threshold. Estimate amplitude weighted frequency of remaining components.
Cartesian frequency FFT data (remaining maxima only)
Indices of remaining maxima
Cartesian time FFT data (remaining maxima and four nearest neighbours only)
Fit parabola to time and frequency reassignment data and estimate ΔA, Δf and variance of fit.
Variance of remaining maxima.
Compare variance data with expected variance data for estimated ΔA and Δf. Where variance difference exceeds threshold reject component.
$\bar{f}_{\text{amp}}$, ΔA and Δf estimates (remaining maxima only)
Indices of remaining maxima
Estimate $\bar{f}$ from $\bar{f}_{\text{amp}}$, ΔA and Δf.
Reject maxima whose $\bar{f}$ estimates do not fall in or close to the analysis bin predicted by $\bar{f}_{\text{amp}}$, ΔA and Δf.
$\bar{f}$, ΔA and Δf estimates (remaining maxima only)

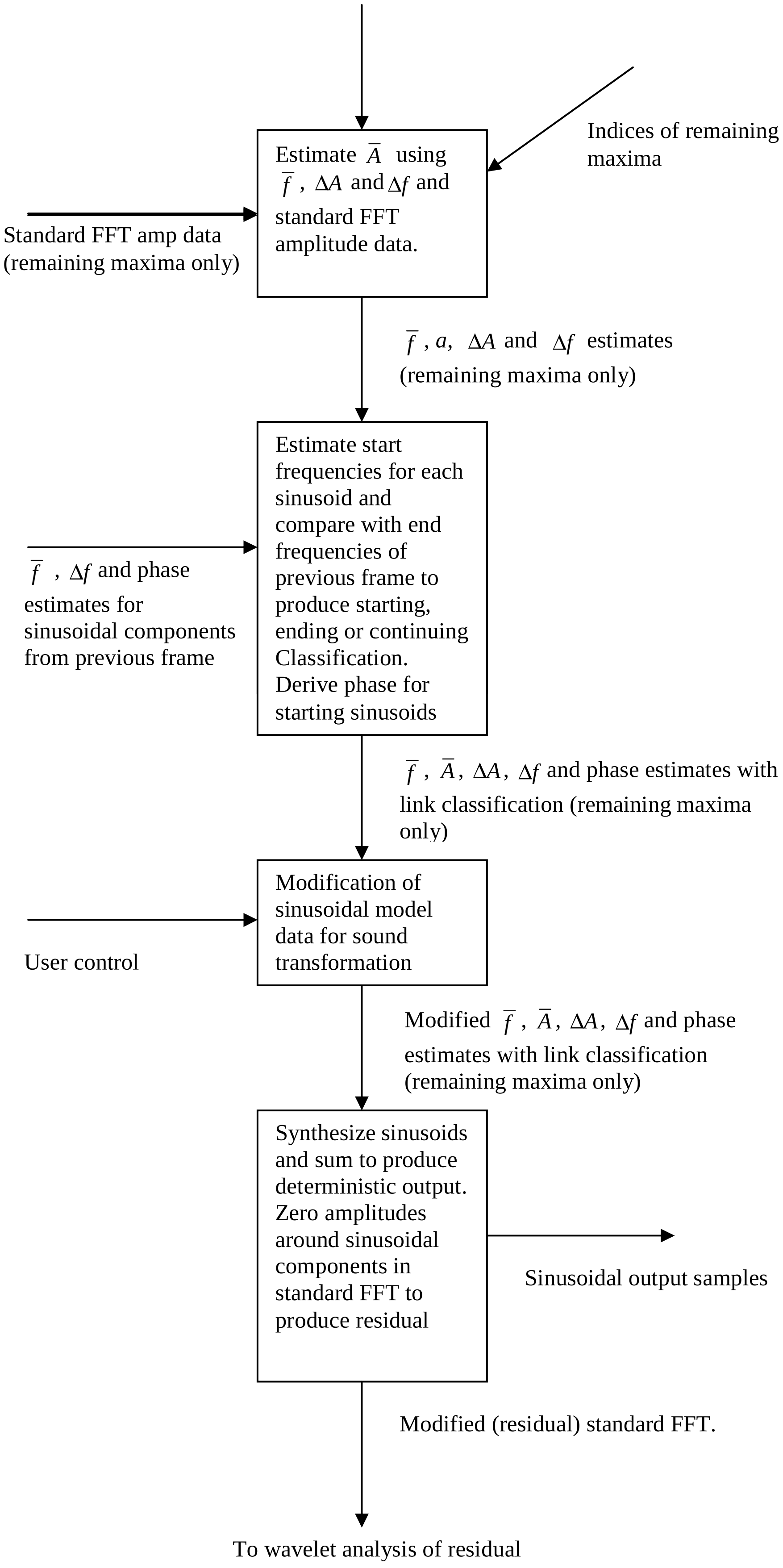

Indices of remaining maxima
Estimate $\bar{A}$ using $\bar{f}$, ΔA and Δf and standard FFT amplitude data.
Standard FFT amp data (remaining maxima only)
$\bar{f}$, a, ΔA and Δf estimates (remaining maxima only)
Estimate start frequencies for each sinusoid and compare with end frequencies of previous frame to produce starting, ending or continuing Classification. Derive phase for starting sinusoids
$\bar{f}$, Δf and phase estimates for sinusoidal components from previous frame
$\bar{f}$, $\bar{A}$, ΔA, Δf and phase estimates with link classification (remaining maxima only)
Modification of sinusoidal model data for sound transformation
User control
Modified $\bar{f}$, $\bar{A}$, ΔA, Δf and phase estimates with link classification (remaining maxima only)
Synthesize sinusoids and sum to produce deterministic output. Zero amplitudes around sinusoidal components in standard FFT to produce residual
Sinusoidal output samples
Modified (residual) standard FFT.
To wavelet analysis of residual

Figure 6.2: Sinusoidal analysis algorithm.

6.4.2 Estimation of amplitude weighted mean instantaneous frequency

As discussed in section 4.5 amplitude change within a frame, such as at the onset or offset of a sinusoid, combined with frequency change will produce a bias in the instantaneous frequency estimate which, in extreme cases, may place the amplitude peak as much as 74 Hz from the true instantaneous frequency. This biased parameter is referred to here as the amplitude weighted instantaneous frequency $\overline{f}_{\text{amp}}$. The bias introduced by the amplitude and frequency change can only be corrected once these parameters have themselves been estimated which is a relatively costly process. For this reason bias correction is deferred until as many components as possible have been rejected due to non-sinusoidality.

One of the criteria for rejection of a peak is whether the bin is contaminated by an outlying component (see section 4.6 and [Desainte-Catherine and Marchand, 2000]): if the estimated instantaneous frequency lies outside of the bin it is not due to a sinusoid within it. However the biasing effect of non-stationarities can produce frequency estimation errors of nearly 14 bins. Therefore, at this stage a bin is only rejected due to contamination if $\overline{f}_{\text{amp}}$ is a distance of 14 or more bins away from the centre frequency of the peak bin. Once the non amplitude weighted $\overline{f}$ has been estimated at a later stage then the relationship between this parameter and the centre frequency of the bin in which the peak resides is re-examined. The fractional bin offset is calculated (from equation 3.132) first and if this is greater than 14 the peak is rejected. If the peak is not rejected the full $\overline{f}_{\text{amp}}$ is calculated and stored.

6.4.3 Estimation of non-stationary parameters and variance of fit

A detailed discussion of the estimation of ΔA and Δf and the use of variance of fit of RDA data is given in sections 4.3 and 4.6. Estimates for ΔA, Δf and σ^2 are produced for each remaining sinusoidal candidate. Candidates whose σ^2 are not below the specified threshold are rejected.

6.4.4 Estimation of non-amplitude-weighted mean frequency and correction of amplitude

With the non-stationary estimates it is now possible to remove the biasing from $\overline{f}_{\text{amp}}$ to give the non-weighted mean $\overline{f}$ and to correct the amplitude estimate as described in section 4.5.

As a final test of sinusoidality the frequency correction (in bins) is compared to difference between the peak bin centre frequency and $\bar{f}_{\text{amp}}$. If

$$\left\lceil \frac{\left|\bar{f}_{\text{amp}} - \bar{f}\right|}{B} \right\rceil < \left| k_{\text{peak}} - \frac{\bar{f}_{\text{amp}}}{B} \right| \qquad (6.1)$$

where k_{peak} is the index of the peak bin, then the peak is rejected. This rejects any peaks that do not exhibit sufficient ΔA and Δf to account for the distance between $\bar{f}_{\text{amp}}$ and the peak bin and so are likely to be due to contamination by an outlying component.

6.4.5 Sinusoidal linking and classification

There are three types of sinusoid within *reSynth*: starting, continuing and ending. Sinusoids sorted into these three types are the result of sinusoidal linking across frames. In a non-real time analysis system (such as SMS) the linking process can be iteratively ‘tuned’ to provide the most appropriate representation for the type of sound being analysed. For example short lived partials can be eliminated and gaps in partial tracks can be interpolated if it is known that a partial continues up to and away from the frame in which the gap appears. In a system such as *reSynth* where the analysis and resynthesis happens on a frame by frame basis these retrospective improvements to the spectral model cannot be made. This is one of the main reasons for developing systems for testing the sinusoidality of a peak within a single frame and for producing very high accuracy estimates of amplitude, frequency and how they change within a single frame. The sinusoidality testing enables the right spectral components to be considered at the frame linking stage, the high accuracy estimates reduce the possibility of mis-linking of components between frames and of discontinuities at frame boundaries due to incorrect evolution of amplitude and frequency throughout the frame.

At this stage of *reSynth*’s analysis the system has decided which peaks in the magnitude spectrum are due to sinusoidal components. At the first analysis frame there are no sinusoids currently in existence since there are no previous frames so all sinusoids are classified as ‘starting’. Since the previous analysis provides estimates of exponential amplitude and linear frequency change the synthesized partials can evolve in this way during the synthesis frame. For example, after just one frame of analysis has been produced a frame of non-stationary sinusoids can be synthesized without requiring a second frame of analysis with which to interpolate amplitude, frequency and phase between. In order to avoid onset discontinuities starting sinusoids are forced to ‘ramp on’ regardless of their ΔA estimate. In *reSynth* this ramp on defaults to a 96 dB exponential increase in amplitude, the amplitude of the final sample of the synthesis frame being that predicted by combining the ΔA and amplitude estimates. For example a starting sinusoid that has estimates $\overline{A} = 1$ and $\Delta A =$ 48 dB is predicted to have an end of synthesis frame amplitude of approximately 5.5. In this case ΔA is set to 96 dB and $\overline{A}$ is modified so that the end amplitude is still 5.5. This ensures that, in a 16 bit system, there is no amplitude discontinuity at the start of the frame and the possibility of amplitude discontinuity for this sinusoid between the current frame and the next is minimised. The disadvantage of this fixed ramp is that it can delay the onset of sinusoidal component since a 96 dB exponential increase in amplitude focuses energy at the very end of the synthesis frame.

At the end of each frame the end frequency and phase of each starting and continuing sinusoid is stored for linking analysis in the next frame. In the subsequent frame the start frequency of all sinusoids are calculated from the estimates of $\overline{f}$ and Δf . For each start frequency value in the current frame the previous frame’s sinusoids are searched for an end frequency value which is close enough for the sinusoids to be linked. A difference threshold is set below which linking can occur. Because of the high accuracy of the iterative RDA technique this threshold can be set low enough that only a very small range of end frequencies has to be searched for each start frequency, often with no more than one or two potential candidates which greatly simplifies the linking process. Despite this simplification there is still the potential for a current frame sinusoid to be linked with that of a previous frame before the best (closest) link has been found. When a better link is found the current link must be broken and replaced. This breaking of the link leaves an unlinked sinusoid from the previous frame which may still be close enough to a current frame sinusoid for linking, therefore the linking process must be repeated to find the second best choice of link, if one exists, for this sinusoid and so the linking process is run twice. Figure 6.3 illustrates the decisions made in the linking process.

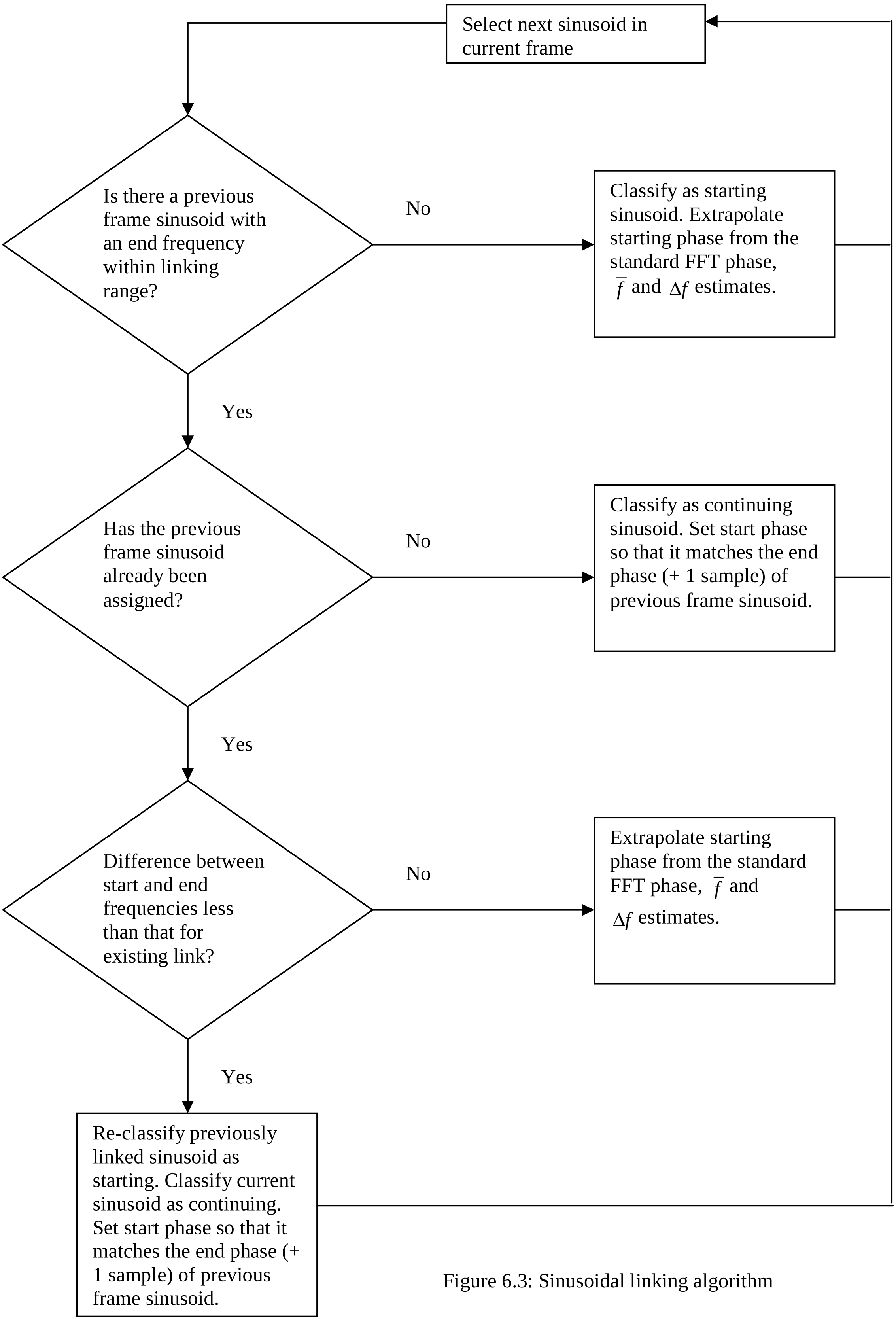

Select next sinusoid in current frame
Is there a previous frame sinusoid with an end frequency within linking range?
No
Classify as starting sinusoid. Extrapolate starting phase from the standard FFT phase, $\bar{f}$ and Δf estimates.
Yes
Has the previous frame sinusoid already been assigned?
No
Classify as continuing sinusoid. Set start phase so that it matches the end phase (+ 1 sample) of previous frame sinusoid.
Yes
Difference between start and end frequencies less than that for existing link?
No
Extrapolate starting phase from the standard FFT phase, $\bar{f}$ and Δf estimates.
Yes
Re-classify previously linked sinusoid as starting. Classify current sinusoid as continuing. Set start phase so that it matches the end phase (+ 1 sample) of previous frame sinusoid.

Figure 6.3: Sinusoidal linking algorithm

6.4.6 Modification of sinusoidal model data for sound transformation

Once the deterministic signal has been described in terms of combinations of sinusoids with non-stationary amplitude and frequency parameters many transformations become straightforward. Pitch shifting independent of time is achieved by multiplying each of the $\bar{f}$ and Δf estimates by the shift ratio (e.g. 2.0 for an upwards octave shift or 0.67 for a downward shift of a perfect fifth). Harmonisation can be achieved by producing additional sets of sinusoids, each of which represent an additional ‘voice’ in the harmony. The $\bar{f}$ and Δf values for sinusoids in each additional set are those of the original input sound multiplied by the pitch ratio for the given harmony. Since the phase of each sinusoid in the additional sets will increment at a different rate to that of the corresponding sinusoid in the input sound the end phase of each additional sinusoid must be stored to prevent discontinuities across synthesis frame boundaries.

Cross synthesis can be achieved in a number of ways such as combining the amplitude trajectories for a partial in one input sound with the frequency trajectories for a corresponding partial in a second sound. With sample by sample knowledge of amplitude and frequency trajectories amplitude effects such as frequency dependent compression/expansion of signals can be performed with great accuracy. This is a key advantage of having a parametric model of audio rather than just short term stationary magnitude and phase spectra: transformations can be intuitively and straightforwardly conceived making it much easier for non-technical users to create their own transformations.

6.4.7 Residual estimation by spectral subtraction

Serra’s SMS system uses time domain subtraction to produce the residual; the entire sinusoidal signal is synthesized and subtracted from the original input signal. An advantage of this approach is that having a time domain representation of the residual means that spectral analysis of the residual can be performed with optimised parameters, such as a shorter analysis frame, for what is assumed to be a stochastic signal. In a system such as *reSynth* which produces output from input on a frame by frame basis it is not possible to employ this approach. Unless there is no overlap between frames (only possible with a rectangular window) the synthesis and analysis frames will be of a different length and so short-time time domain subtraction is not available either. Therefore *reSynth* employs spectral subtraction to calculate the residual signal although the data is finally transformed back to the time domain in analytic form after Hilbert transformation in the Fourier domain.

Whereas SMS uses shorter windows to analyse the time domain residual, *reSynth* uses the constant Q B-spline split wavelet transform described in chapter 5 with the same analysis frame length as is used for the sinusoidal FFT prior to zero-padding. With a frame length of 1025 samples this offers ten frequency bands of analysis whose width and time resolution varies with centre frequency.

An assumption of SMS is “that the residual is fully described by its amplitude and its general frequency characteristics. It is unnecessary to keep either the instantaneous phase or the exact spectral shape information” [Amatriain et al, 2002]. Augmentations of the SMS model to include a third signal component type (transients) acknowledge that this assumption is not valid in some cases [Verma and Meng, 2000]. Whilst it is the case that for long term stationary noise the phase spectrum does not contain important information the case for short duration broad band (i.e. impulsive) components is that both the phase and magnitude are needed to retain perceptually relevant fast changing temporal detail. The spectral modelling technique used for the residual in *reSynth* is intended to be capable of capturing the temporal detail of transient components and the spectral resolution of longer term stochastic components. Since both the phase and magnitude of non-sinusoidal components remain intact after spectral subtraction the inherent timing information contained within these components is passed onto the complex wavelet analysis combining both transient and long term noise in the one model.

Time domain subtraction is a straightforward and, provided the instantaneous frequencies and amplitudes of the sinusoids are well predicted by the model, effective operation. Spectral subtraction is a more complex process since individual sinusoidal components are not represented by individual points in the Fourier domain. Finite length windowing smears components into multiple bins and non-stationarity exacerbates this: frequency change widens the main lobe and amplitude change narrows the main lobe but increases the level of side lobes, increasing the spread of energy to distant bins. A single sinusoid is represented by a single complex number in the Fourier domain only in a very specific situation: a rectangular analysis window is used, the analysed sinusoid has stationary amplitude and frequency and its frequency coincides exactly with the centre of an analysis bin (i.e. the length of the analysis window is an integer multiple of the sinusoidal period).

In preliminary investigative work undertaken for this thesis into the combination of Fourier and wavelet analysis a spectral subtraction technique was developed for use in a transform based thresholding process, *Wavethresh*. This technique used knowledge of the window power spectrum to predict the contribution made to adjacent bins made by a stationary sinusoid for a given deviation of the sinusoid's frequency from that of the bin centre [Wells & Murphy, 2003]. This was necessary since *Wavethresh* used a critically sampled (i.e. non zero-padded) FFT which produces large variations in energy localisation around a sinusoidal peak for different deviations of the mean frequency from that of the centre of the analysis bin. A zero-padding factor of 8 is used in *reSynth* and this over-sampling of the spectrum significantly reduces the variation in energy localisation. Figure 6.4 shows the relationship between the deviation from bin centre and the number of bins which would have to be zeroed to reduce the component energy by 48 dB for a Hann window for an eight times zero-padded and non zero-padded FFT. The non-zero padded FFT requires less bins to be zeroed to reduce the component to the required level. However there is significant variation in the number of bins that have be zeroed to achieve the desired degree of attenuation whereas this remains constant for the zero-padded case.

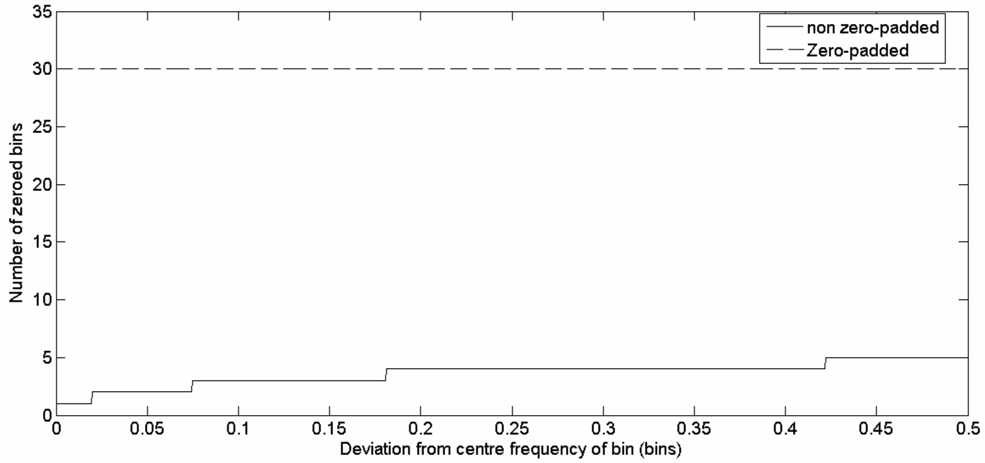


Figure 6.4: Number of bins zeroed to produce an attenuation of -48 dB for a stationary sinusoid versus distance from centre frequency of bin. The comparison is between non zero-padded and 8x zero-padded FFTs.

Since the spectral data is available in zero-padded form there are two approaches that can be taken to obtain a time domain version of the residual: decimation in the frequency domain or in the time domain. Figure 6.4 suggests that for a sufficiently zero-padded spectrum the spectral subtraction process can be performed very simply by setting sinusoidal peaks and adjacent bins to zero. Following inverse transformation decimation in time is performed by discarding samples beyond the time support of the analysis window. Since the spectral subtraction process can spread some of the remaining component energy outside the support of the analysis window this also helps to reduce the deterministic energy in the residual signal. The disadvantage of not decimating before transformation to the time domain is the increased cost of the IFFT. The time domain decimation method is used in *reSynth* since this greatly simplifies the spectral subtraction process and offers much greater consistency in the relationship between the number of bins that are zeroed and the attenuation of deterministic components.

Non-stationarity must also be accounted for in the spectral subtraction process. Frequency non-stationarity causes a widening of the main lobe but there is little change in the energy contained in distant bins. A very simple model of the relationship between width in the Fourier domain and amount of frequency change is employed in *reSynth*: the energy spread in the Fourier domain is approximately equal to the frequency change that occurs during the frame. For example a frequency change of two bins (just over 10 Hz for a 8192 point FFT of a 44.1 kHz sampled signal) spreads the energy into two additional bins over the stationary case, one either side of the peak bin). This is as if the stationary window spectrum were convolved with a rectangular pulse which is the width of the frequency change. Figure 6.5 illustrates the number of bins, actual and predicted, that need to be zeroed to produce an attenuation of 48 dB for a given frequency change.

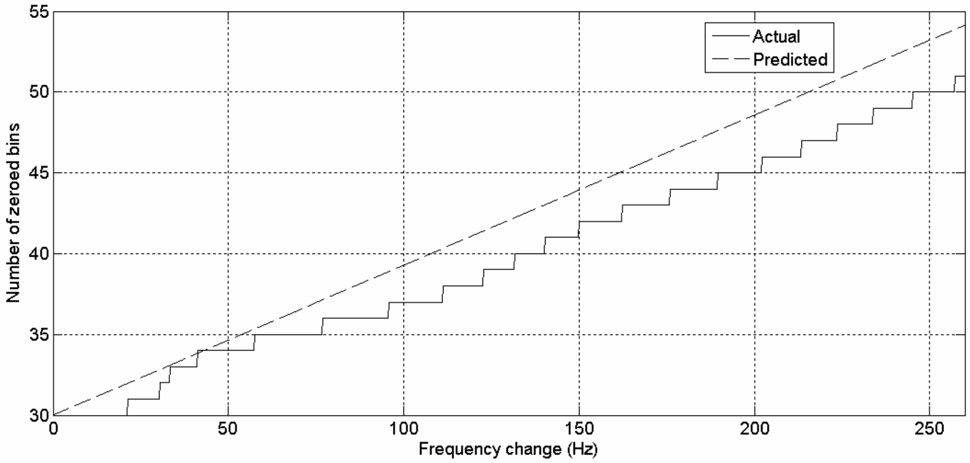


Figure 6.5: Number of bins zeroed to produce an attenuation of -48 dB for a non-stationary sinusoid versus amount of intra-frame frequency change.

Amplitude non-stationarity can produce a significant de-localisation in the Fourier domain of a sinusoidal component. This is due to the localisation in the time domain that is produced by the amplitude change; the greater the amplitude change, the more impulse-like the component becomes. The more impulsive a component becomes the less energy it contains compared to a stationary sinusoid with the same peak amplitude. A positive amplitude change localises energy at the end of the frame and negative change localises energy at the beginning of a frame. These are the parts of the frame that experience the greatest attenuation when a window is applied. The lower energy in a component with non-stationary amplitude combined with the attenuation introduced by the windowing process offsets the energy spreading in the Fourier domain: although zeroing a given number of bins produces less attenuation for a component with non stationary amplitude this loss of attenuation is compensated. This is illustrated in figure 6.6 which shows the attenuation produced by spectral zeroing of 30 bins and the attenuation produced by the amplitude non-stationarity. It can be seen that the combined attenuation actually falls with increasing amplitude change. For this reason the ΔA estimate for a sinusoidal component is not considered in the spectral subtraction process in *reSynth*.

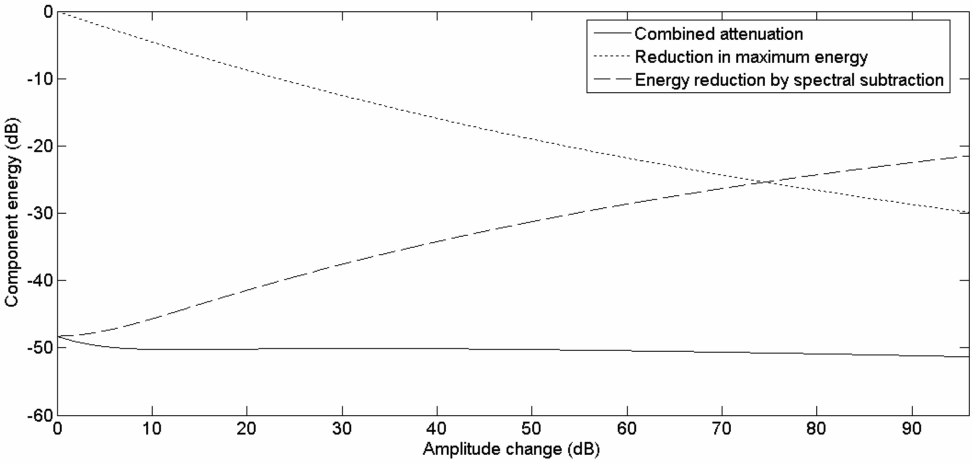


Figure 6.6: Combined energy reduction for a sinusoid with non-stationary amplitude.

6.5 Complex wavelet analysis of the residual signal

Much of the detail of the complex wavelet analysis employed in *reSynth* has been given in chapter 5. What follows in this section is a brief overview of its implementation.

After spectral subtraction, by zeroing bins containing sinusoidal components, a Fourier representation of the non-sinusoidal part of the signal (the residual) remains. As discussed in section 5.2 the complex wavelet analysis is performed by iteratively applying the same LPF and HPF separately to the real and imaginary parts of the analytic signal. The analytic signal is produced by Hilbert transformation in the Fourier domain. Section 5.4.1 noted that the finite initialisation filters proposed in [Unser et al, 1993] do not produce the expected behaviour at lower scales. Much improved initialisation of the input sequence to the wavelet analysis is achieved by multiplication with an m+1 order sinc function, which is the Fourier transform of the continuous mth order B-spline, in the Fourier domain. Since the residual signal is already in the Fourier domain the Hilbert transform and wavelet initialisation can be simply and cheaply accomplished.

The real and imaginary time domain signals are obtained by IFFT. This produces zero-phase 8192 point rather than chronological 1025 point sequences so they are reorganised and truncated to produce the time-domain analytic signal. The real and imaginary sequence are then separately analysed using the partially decimated transform to produce the control data for the parametric equalisers that are used to resynthesize the residual. The user has control over the number of decimated and undecimated levels in the analysis. The maximum total number of levels for a frame length of 1025 samples is 10. Reducing the number of undecimated levels reduces the computational cost of the wavelet analysis.

6.6 Synthesis

Additive and subtractive synthesis are two of the most common and well understood synthesis techniques available to the computer/electronic musician. Additive synthesis is used in *reSynth* to recreate the deterministic part of the input signal and subtractive synthesis is used to fashion the stochastic part from broad band noise. In *reSynth* both syntheses are performed on a sample by sample basis[2]. This offers flexibility but the computational cost is greater than synthesis by inverse transform [Freed et al, 1993]. At the time of writing *reSynth* is believed by the author to be the only heterogeneous spectral modelling system which performs all synthesis in the time domain. The following two sections describe how both parts of the signal are synthesized. The final output from *reSynth* is the addition of the output from each of sinusoidal oscillators and noise equalisers.

6.6.1 Sinusoidal synthesis

The sinusoidal synthesis is performed by generating the phase trajectories from the $\overline{f}$ and Δf estimates, generating the sinusoid from these and then applying the amplitude envelope derived from the $\overline{A}$ and ΔA estimates. The MATLAB implementation calculates each

[2] Strictly this is only true in the non-partially decimated case since the parameters of equalisers used to synthesize decimated scales are not updated every sample, although the noise which they process is produced on a sample by sample basis.

sinusoidal value from scratch rather than using a lookup table. Experimentation on the test PC has indicated that in both MATLAB and VST plug-ins written in C++ there is still a moderate speed advantage when using table lookup to produce sinusoid values. For example, within a VST plug-in DLL non-interpolating wavetable lookup takes approximately 75% of the execution time compared to direct sine calculation. However, MATLAB passes global variables to multi-function 'm' files extremely slowly compared to C which negates this advantage and actually makes a function which accesses such a global vector execute much more slowly. The sinusoidal synthesis sub-function has been translated to a MEX file, however the problem still remains that passing a lookup array to a MEX DLL within the context of a MATLAB main function is a relatively expensive operation. If *reSynth* were to be implemented in a single DLL file (such as a VST plug-in) then wavetable lookup would be the cheapest means of sinusoidal synthesis.

A key novel feature of *reSynth* is that it is able to produce tracks of sinusoids across frames despite the fact that it works on a frame-by-frame basis. The only other system known to the author that attempts streaming sinusoidal analysis is described in [Lazzarini et al, 2005]. However this system does not attempt to model non-stationarity and does not act on a single frame of data. Instead it uses a number of 'track points' across already acquired frames which introduces considerable delay (relative to that of *reSynth*). The partial tracking in *reSynth* is, to a certain extent, implicit in the high accuracy non-stationary sinusoidal modelling described in chapter 4. Fourier data produced by a long term sinusoid that exists across multiple frames will produce model data that produces agreement between the amplitude and frequency at the end of one analysis frame and the beginning of the next. If it does not then there will be a discontinuity that will manifest itself audibly as a click at the output. There is no amplitude or frequency interpolation across frame boundaries at the synthesis stage of *reSynth*. The only explicit 'fixing' of the model data is the derivation of the start of frame phase for a continuing or ending sinusoid from that of the previous frame sinusoid to which it is linked. This phase matching uses the information produced by the inter-frame linking process (described in section 6.4.5) to determine how phases in the current synthesis frame are determined. Therefore the phase trajectories are piecewise quadratic and the amplitude trajectories are piecewise exponential.

6.6.2 Residual synthesis

Synthesis of the residual part of the signal begins with the generation of a broad band noise signal. Since a wholly stochastic ('white') process has no knowledge of its previous behaviour there is no need to 'link' noise across frames therefore at each synthesis frame a new noise sequence of required number of samples is generated 'from scratch'. White noise with a Gaussian distribution is produced by two pseudo random sequences $U_1[n]$ and $U_1[n]$, each the length of the synthesis frame. These are combined to produce the zero-mean output noise sequence $G_1[n]$ as described in [DSP Primer, Rorabuagh, p.39]:

$$G_1[n] = \cos\left(2\pi U_2[n]\right)\sqrt{-2\sigma^2 \ln\left(U_1[n]\right)} \qquad (6.3)$$

where σ (with a default value of 0.5 in *reSynth*) is the standard deviation of the output sequence.

For a full level residual component the gain of the synthesis equaliser is set at 96 dB therefore the level of $G_1[n]$ is reduced by a corresponding amount so that unless it is actually boosted by one or more of the equalisers it remains below the noise floor of the 16 bit system. Where the output from the complex wavelet analysis is undecimated the frequency, magnitude and bandwidth of the equalisers are updated every sample. As discussed in section 5.7 this requires a large number of calculations for each equaliser at each sample, particularly where the Orfanidis design is used. Also, MATLAB is designed for high speed vector rather than scalar operations meaning that 'one at a time' processes (such as '`for`' loops) do not execute quickly in this environment. For these reasons the equalisation is performed by a MEX function. In order to reduce the number of calculations required at each equaliser update the gain is fixed and the amplitude of the input noise sequence is modulated by the estimated component magnitude. Where the output analysis is decimated then the equaliser coefficients are only updated when there is new analysis data. The current implementation of *reSynth* uses ten parametric equalisers. In order to compensate for the doubling of magnitude at each increase in scale in the analysis (see figure 5.26) scale dependent attenuation of the noise signal is performed at the input to each equaliser.

6.7 System performance

With an overview of system functionality in place, this section presents examples of how *reSynth* performs on examples of different types of sound, both synthetic and acoustic. The ability of the system to separate and describe different signal components is examined. Many of the examples discussed are provided in Appendix B of this thesis which is an audio CD. Where the system models a particular sound or component well then good quality pitch scaling independent of duration is easily achieved by multiplication of the sinusoidal and parametric equaliser centre frequencies by the scaling factor. Where the input sound is not well modelled then, of course, plausible pitch scaling is not successful either.

All of the signals used are mono, 44.1 kHz with 16 bit quantisation. Unless otherwise stated the analysis frame length is 1025 samples and the overlap factor is 2. Where there are time domain plots comparing input and output the inherent processing delay between input and output has been removed to make comparison between temporal evolution of features easier. This means that the sample numbers correspond between plots in the same figure, but the delay between input and output is not represented.

As has already been discussed, there are no other spectral modelling systems that attempt to work on a frame-by-frame basis and the vast majority of existing systems perform their analysis offline once the entire signal has been acquired. Since this is a novel system there are no others with which useful, direct comparison can be made. The following investigations and discussions therefore report on the real-time spectral modelling capability of a system which uses the novel signal analysis techniques described in previous chapters but they do not attempt a comparison with existing spectral modelling systems since their design goals are so different.

6.7.1 Sinusoids

6.7.1.1 Single sinusoid with abrupt onset and offset

Perhaps the most basic function that a sinusoidal plus residual modelling system can perform is to correctly resynthesize a stationary input sinusoid. Figure 6.7 shows the start of an input 1 kHz sinusoid and the start of the sinusoidal part of the output from *reSynth* with an overlap of 2 and 4. The input sinusoid starts and ends abruptly which poses a challenge to this system since at this point the behaviour of this ‘stationary’ sinusoid is actually highly non stationary. Since *reSynth* models sinusoidal onsets as an exponential function the onset of the sinusoid is smeared in time. Increasing the overlap reduces the onset time of the resynthesized sinusoid and this more abrupt onset is closer in shape to that of the input. Increasing the overlap factor in *reSynth* improves the time resolution of the resynthesized amplitude trajectories but at increased computational cost. For example, increasing the overlap by a factor of two increases the number of frames that have to be evaluated by the same amount.

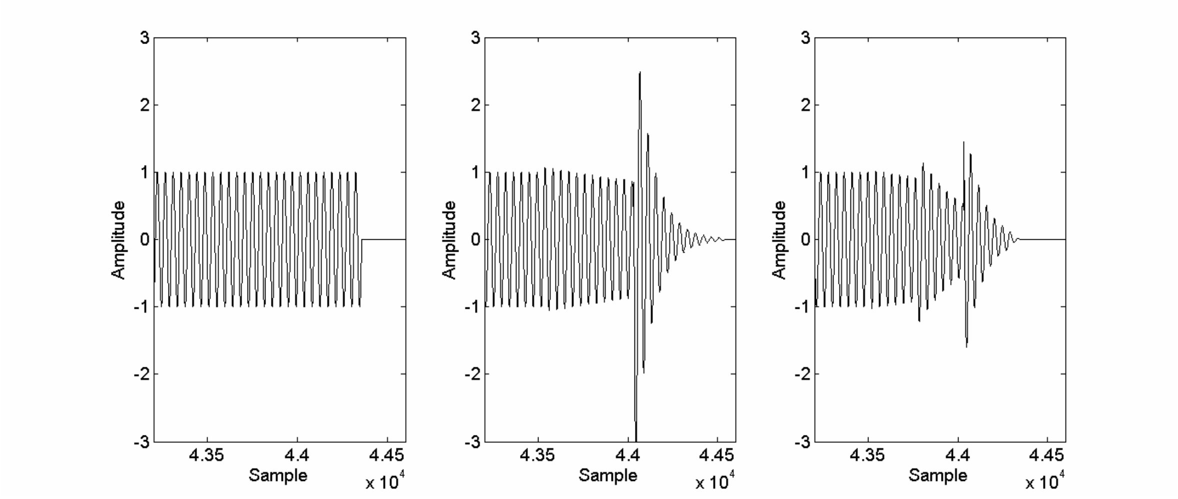


Figure 6.7: Onset of input sinusoid (left), resynthesized output, overlap factor of 2 (middle) and overlap factor of 4 (right).

The offset of the sinusoid is not synthesized as well as the onset. Figure 6.8 shows this part of the input and output signals. It can be seen that for a frame length of 1025 samples there are significant mismatches between the amplitudes at the end of one synthesis frame and the start of the next around the offset point. This is due to the start and end amplitudes of these frames being incorrectly estimated since the overall amplitude of the analysis frame is assumed to be due to a sinusoid that has undergone exponential, rather than abrupt, amplitude change. This problem does not manifest itself at the onset because the onset coincides with an analysis frame centre or boundary. Doubling the overlap does reduce the level of the amplitude discontinuities during the offset synthesis but there is still significant distortion of the offset which is audible as an increase in level of the click heard at the offset. A more satisfactory solution in this case is to reduce the analysis frame length. This reduces the error produced by the exponential extrapolation of the average frame amplitude to the start and end of the frames. The error is reduced since the extrapolation occurs over a shorter range. This is shown in figure 6.9. With a frame length of 513 and an overlap of 2 the amplitude 'overshoots' are reduced. Doubling the overlap retains more energy at the offset but the amplitude envelope is distorted. A 129 sample analysis frame provides an output that best matches the shape of the input at the offset stage. It is clear from this figure that, as expected, abrupt onsets and offsets of sinusoids are modelled much better when the frame length is lower than the default setting of 1025 samples.

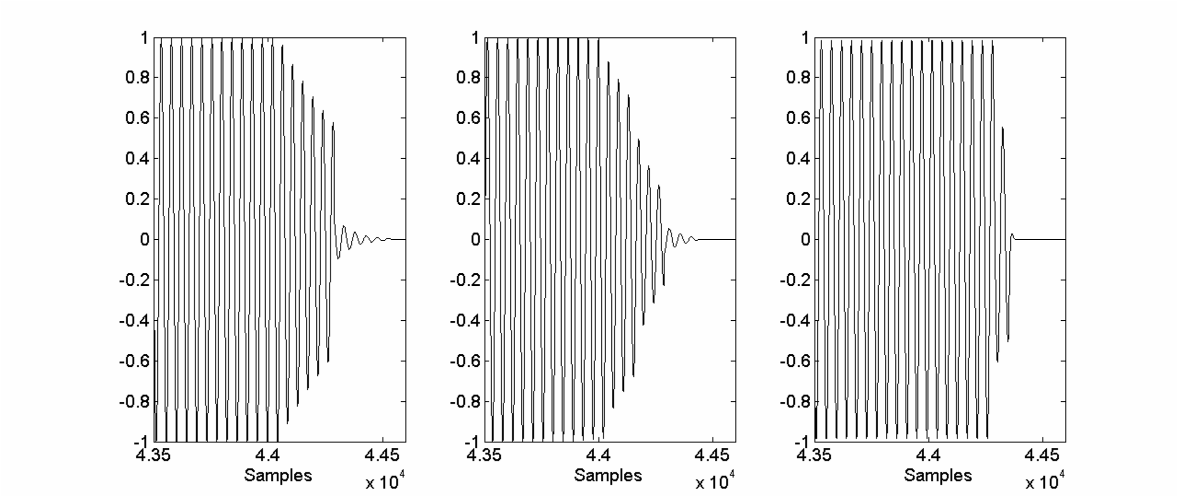


Figure 6.8: Offset of input sinusoid (left), resynthesized output, overlap factor of 2 (middle) and overlap factor of 4 (right).

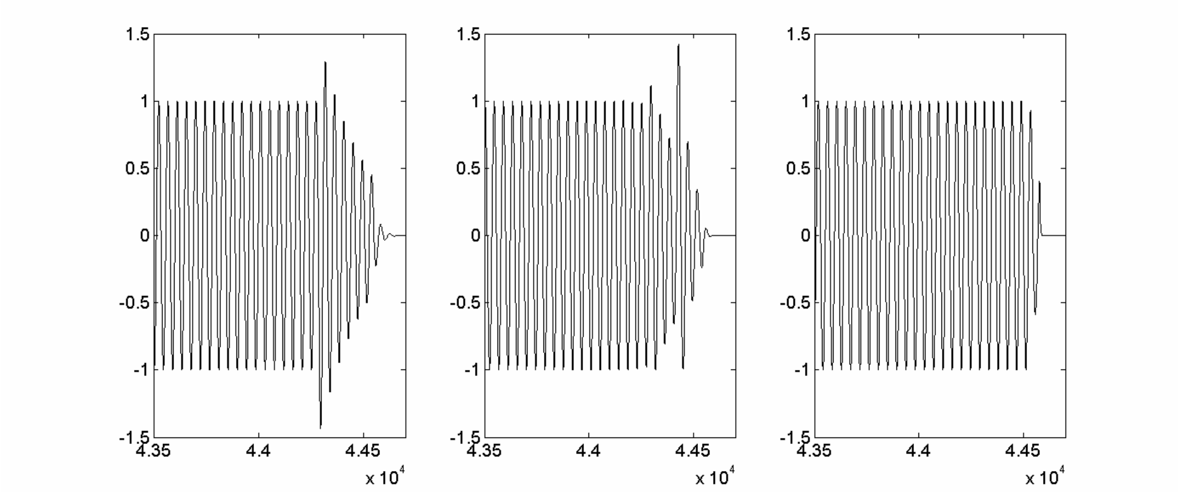


Figure 6.9: Offset of resynthesized output sinusoid with a frame length of 513 samples and overlap factor of 2 (left), frame length of 513 and overlap of 4 (middle) and frame length of 129 and overlap of 2 (right).

Where there are sudden onsets or offsets energy will be spread in the Fourier domain and vestiges of these transient regions will exist in the residual spectrum after spectral subtraction. This manifests itself in the noise output from the system. Figure 6.10 shows the onset in a windowed input frame before and after spectral subtraction and its manifestation as filtered noise at the output of *reSynth*. The localisation of the output from the noise synthesis is much better than for the exponential onset sinusoid. For low overlaps this causes an audible ‘double onset’ but at higher overlaps the two synthesis components become temporally fused.

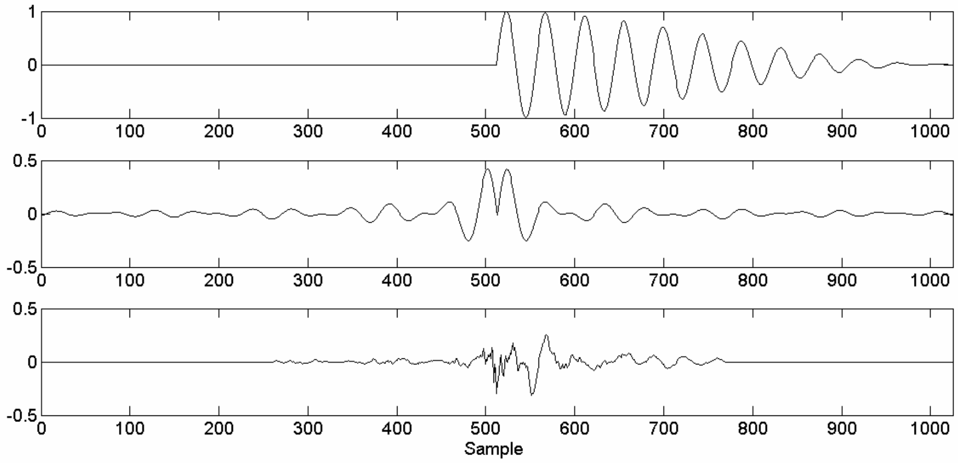


Figure 6.10: Windowed analysis frame with sinusoid onset at centre (top), residual analysis frame after spectral subtraction (middle) and noise synthesis of residual (bottom). The frame in the bottom figure is half the size of those above it but has been centrally aligned.

6.7.1.2 Combination of Sinusoids with vibrato

A synthetic signal comprising five harmonically related components with a slow and wide vibrato (2 Hz) is considered next. This signal tests the ability of *reSynth* to reproduce smooth variations in frequency and properly link sinusoids in a multi-component signal. The magnitude STFT of the input and the output is shown in figure 6.11 and the estimated sinusoidal trajectories are shown in figure 6.12. A logarithmic frequency scale is used so that the evolution of frequency trajectories can be clearly seen. There has been no smoothing of the trajectories, these are the exact linear segments derived from the RDA analysis.

A number of short duration sinusoids have been identified by *reSynth* above 5 kHz but these are of very low amplitude and are therefore inaudible in the final output. These are mis-identifications of side lobes since there is no noise in this signal. Whilst these are not desirable they are few in number compared to the number of magnitude peaks which are considered candidates at the start of the sinusoidal discrimination scheme employed here. In this example a total across all frames of 16,657 possible candidates are identified by the ‘local maximum’ test of which 15,631 (94 %) are rejected during the following stages of the discrimination methods described in this chapter and chapter 4. The main difference between the input and output is at the onset and offset of the harmonic signal. In the final frame of figure 6.12 a high number of spurious sinusoids can be seen since the offset of the signal does not coincide with an analysis frame boundary.

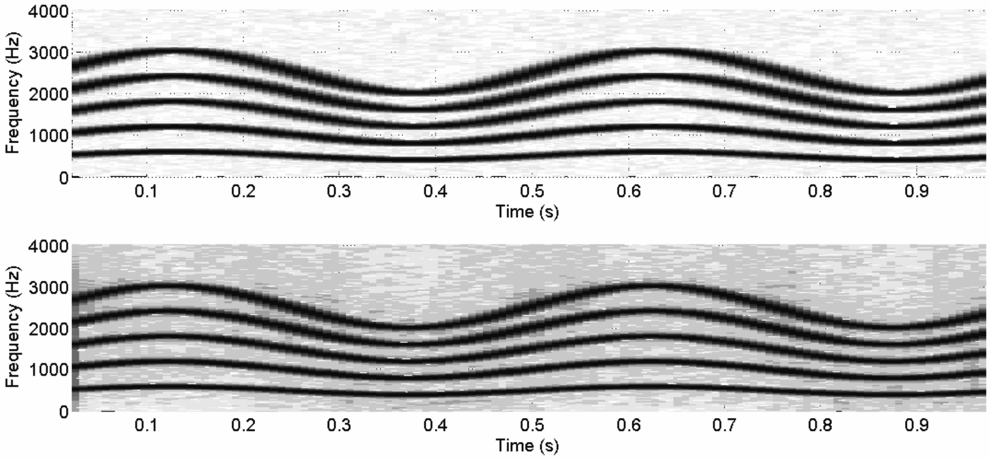


Figure 6.11: STFT of input (top) and output (bottom) signal. A Blackman-Harris 2048 point analysis window with an overlap factor of 6 is used.

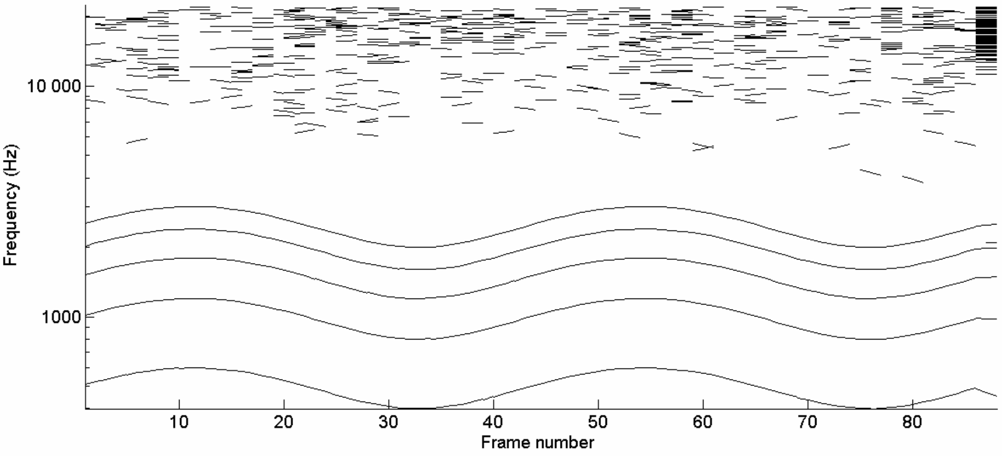


Figure 6.12: Frequency trajectories produced by sinusoidal analysis part of *reSynth*. The frame length is 1025 samples and the overlap factor is 2.

When the vibrato rate is increased to 12 Hz the sinusoidal discrimination and tracking breaks down for all but the fundamental. Figure 6.13 shows magnitude STFTs of the input and output for this signal. A shorter FFT length has been used for this figure to properly capture the fast vibrato. Figure 6.14 shows the partials identified by *reSynth*.

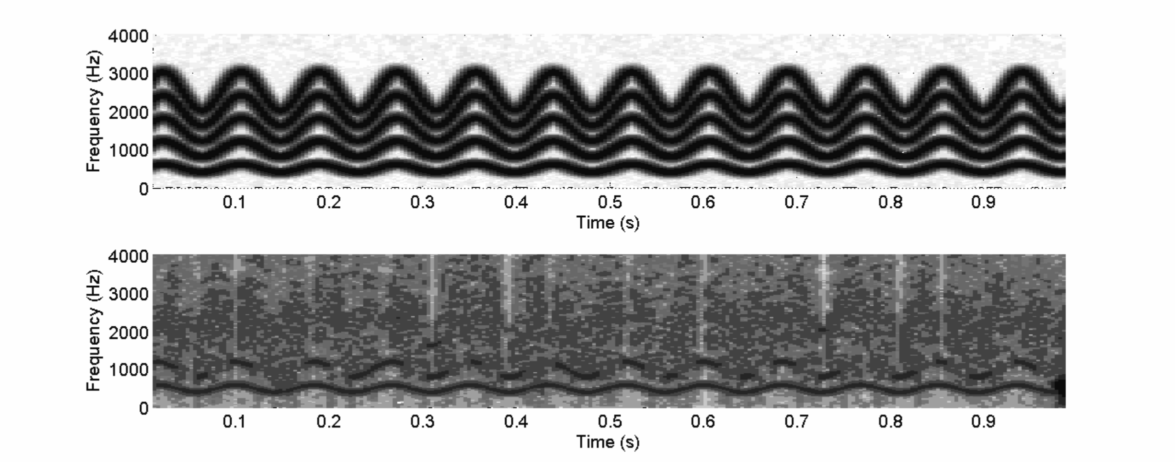


Figure 6.13: STFT of input (top) and output (bottom) signal. A Blackman-Harris 1024 point analysis window with an overlap factor of 6 is used.

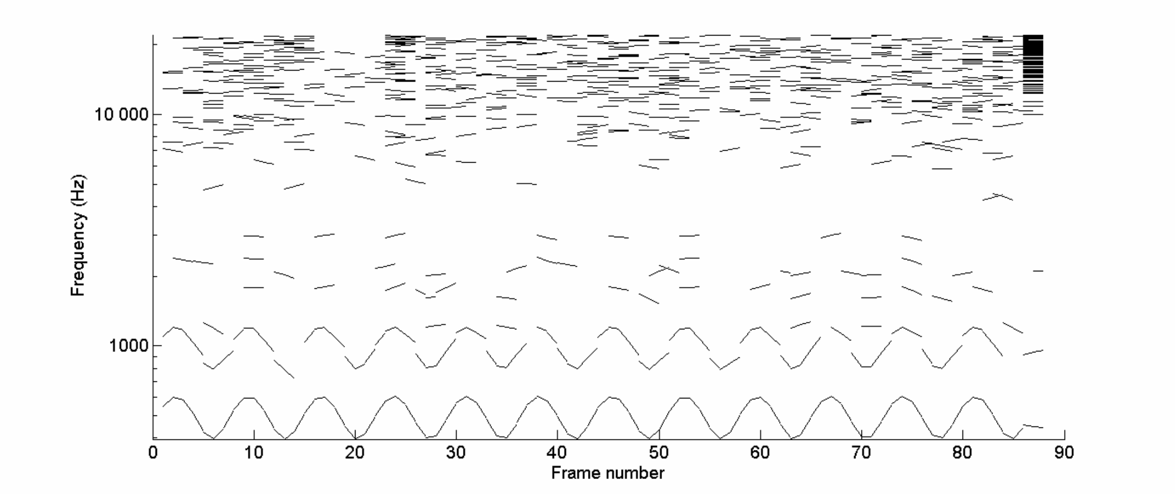


Figure 6.14: Frequency trajectories produced by sinusoidal analysis part of *reSynth*. The frame length is 1025 samples and the overlap factor is 2. The frequency axis is logarithmic.

It is clear from these figures that the sinusoidal discrimination and tracking is unable to properly model the fast and deep vibrato in the input signal. Where there are gaps in the sinusoidal trajectories or where the start and end frequencies are not within the linking range the resynthesized sinusoids are interrupted causing severe audible distortion. The performance is improved by raising the variance difference threshold (VDT) and by reducing the frame length. Figure 6.15 illustrates how these changes to the analysis parameters affect partial tracking. With a frame length of 1025 but the VDT raised from the default of 0.006 to 0.06 most sinusoids are correctly identified but the frame is too long to capture the fast frequency changes in the signal and the trajectories of the upper harmonics are corrupted.

Reducing the frame length to 513 aids both identification and parameter estimation but at the default VDT some gaps in the upper partial trajectories remain. With a shorter frame and a raised VDT the five harmonics are correctly identified, tracked and synthesized. The cost of raising the VDT threshold is that more signal components are incorrectly identified as being sinusoidal however this is offset by reducing the frame length. For the top plot in figure 6.13c a total of 14,902 local maxima are identified of which 73.7% are rejected. For the middle plot there are 16,530 maxima of which 85.0% are rejected and for the bottom plot 78.0% of 16,530 are rejected.

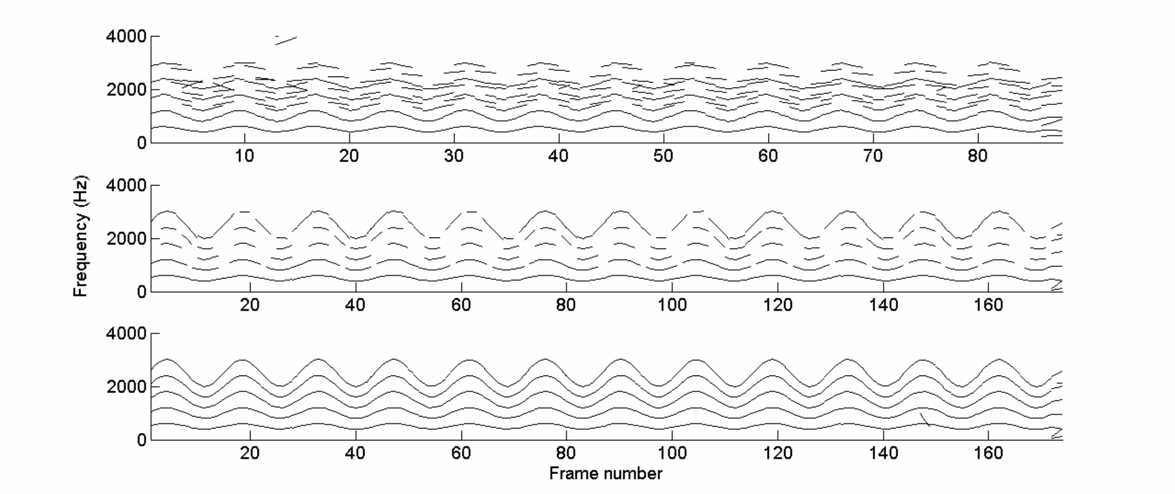


Figure 6.15: Frequency trajectories produced by sinusoidal analysis part of *reSynth*. The frame length is 1025 (top) and 513 (middle and bottom). The variance difference threshold is 0.06 (top and bottom) and the default setting of 0.006 (middle). The frequency axes are linear.

6.7.1.3 Residual analysis of mis-classified sinusoids

If *reSynth* fails to properly classify a sinusoid it will remain in the residual and the system will attempt to model it with filtered noise. This is shown in figure 6.16 for a 1 kHz sinusoid lasting 1 second. The figure shows the first 200 samples of the time domain input and output and the long term average magnitude spectrum of the entire output. A pitch which matches that of the input can be clearly discerned in the output although it has a ‘grainy’ rather than a ‘pure’ quality which is due to the stochastic nature of the input to the equaliser. The random variations in the output can be clearly see in the middle panel of the figure although variations in amplitude are smoothed somewhat by time domain smoothing effect of the equaliser which is highly recursive since its bandwith is narrow. Whilst such a situation is certainly not ideal this example does demonstrate the extent to which the residual part of the system can adapt to long term narrow band components. Since the analysis filters are spaced an octave apart where more than one sinusoid falls in a single analysis octave the bandwidth will extend to encompass both sinusoids and the components would not be well modelled by the system. They would appear as a single noise component whose bandwidth would correspond to the distance between the two.

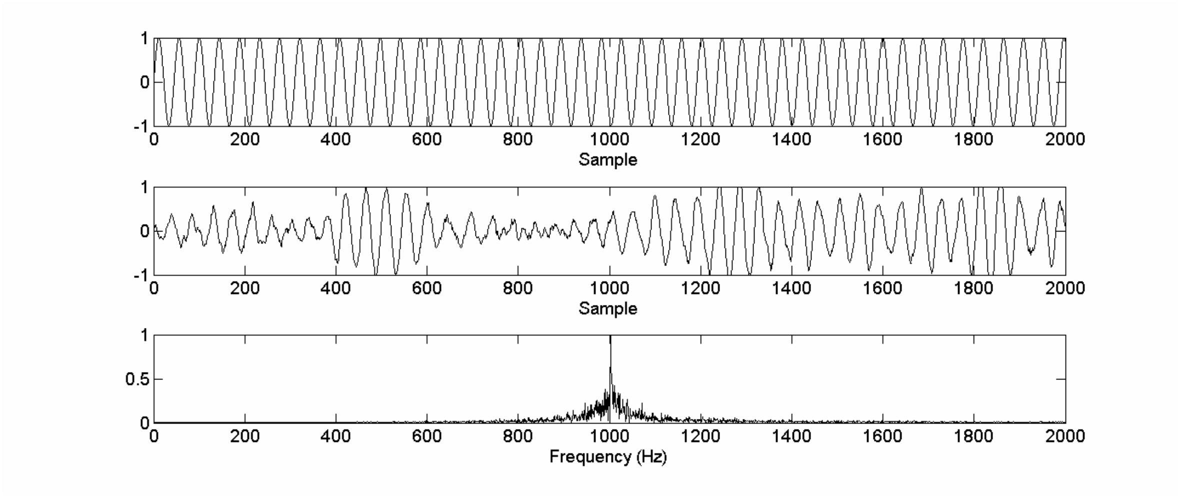


Figure 6.16: Input stationary sinusoid at 1 kHz (top), resynthesized as residual signal (middle) and the normalised long term average magnitude spectrum of the resynthesis (bottom).

6.7.2 Broad band component types

This section describes how the system performs for two important types of broad band component: impulses and long term stationary noise. Impulses are best analysed by filters with compact support which are able to follow sudden changes in a signal. In contrast stationary noise is best described by filters with longer support which are able to smooth the sample-by-sample fluctuations of a stochastic signal in order to derive its long term parameters. This system is better suited to the former since sinusoidal analysis is performed before that of the residual and it is inherently short-term since it divides the signal into short frames.

6.7.2.1 Impulses

Since the Fourier transform of an impulse contains no local maxima no sinusoids are identified. Therefore such a component occurring in isolation is synthesized purely as a residual part of the signal. Figure 6.17 shows the resynthesis of a series of single impulses each consisting of a single transition from 0 to +1 and back to 0 again. The output is not identical for each impulse since the input to the filters is noise and there is some smearing in time of the impulses due to the support of each of the analysis filters and synthesis equalisers being greater than 1. It can be seen that there may be occasions when the peak amplitude of the synthesized impulse exceeds that of the original impulse. This occurs when the impulse in the input signal coincides with a high noise level at the synthesis stage. In this case no clipping distortion would be generated since the overshoot is only one sample in duration although the spectrum of the distorted impulse would be altered slightly. Nevertheless it does highlight the possibility for occasional amplitude overshoots in resynthesis which may require some subsequent limiting or overall level reduction to prevent brief instances of distortion at the output of the system.

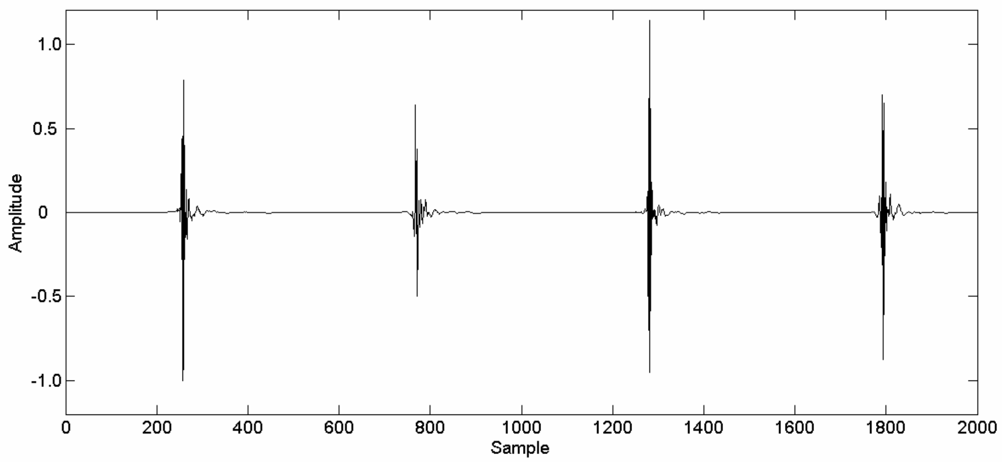


Figure 6.17: Resynthesized sequence of impulses.

6.7.2.2 Long term stationary noise

When a long term broad band component such as white noise is analysed, local maxima in each Fourier frame will occur. The position of these maxima will shift randomly from frame to frame enabling an algorithm which assesses stability/near-stationarity across a number of number of frames to reject such maxima as being due to sinusoids. Such knowledge is not available to a frame-by-frame system such as *reSynth*, requiring subsequent stages of sinusoidal identification within the current frame to correctly reject such peaks. The algorithm described in chapter 4 is only partially successful in this although it does offer a considerable improvement over one which classifies solely on the basis of whether a component is a local maxima. For a 3 second test signal comprising white noise of constant amplitude 44 700 local maxima are identified of which 73.2 % are rejected. Where the frame length is reduced to 513 samples only 40.0% of 44 688 are rejected.

Where peaks are not rejected they are synthesized by short term sinusoids which ramp on and off (since a link is rarely found between frames) and this can be heard as ‘birdy noises’ or ‘bubbling’ in the output. At the residual analysis and synthesis stage the bandwidth, frequency and magnitude estimates also vary in an unpredictable way. Since the support of the wavelet filters is lower than that of their splits the magnitude can change more rapidly than the bandwidth. Thus a sudden increase in magnitude where the estimated bandwidth is still low can cause a large amount of energy to be injected into a narrow part of the spectrum which itself causes similar audible effects to that of the sinusoidal part of the synthesis. This can be countered by smoothing the magnitudes at each scale but this reduces the temporal resolution of resynthesized impulsive components. Such smoothing is more successful where the magnitude is finely sampled (i.e. where there is little or no decimation in the wavelet analysis). Figure 6.18 shows the time domain input and output signals and figure 6.19 shows the long term average Fourier magnitude spectra of these for this white noise signal. Although some differences are evident in these plots in terms of total energy and consistency of time domain amplitude the fundamental audible difference is in the texture of the noise which cannot be discerned from this figure.

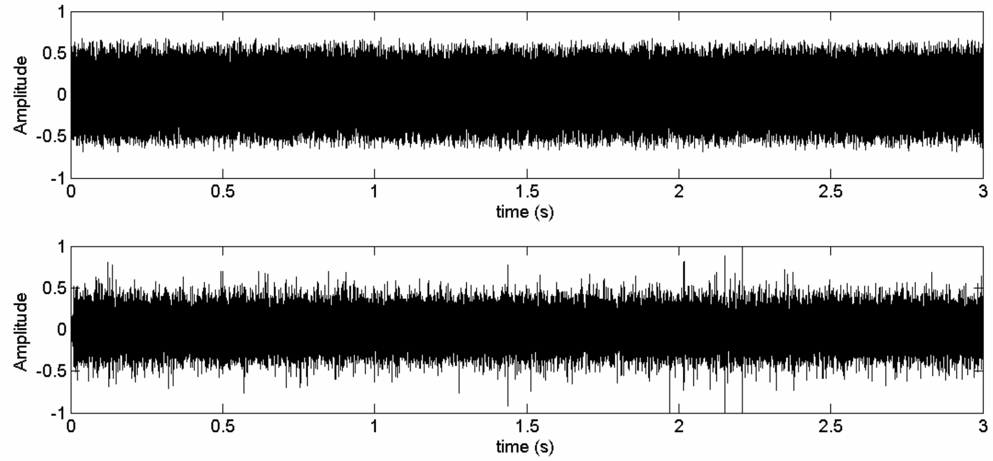


Figure 6.18: Time domain input and resynthesized output for a constant amplitude white noise input.

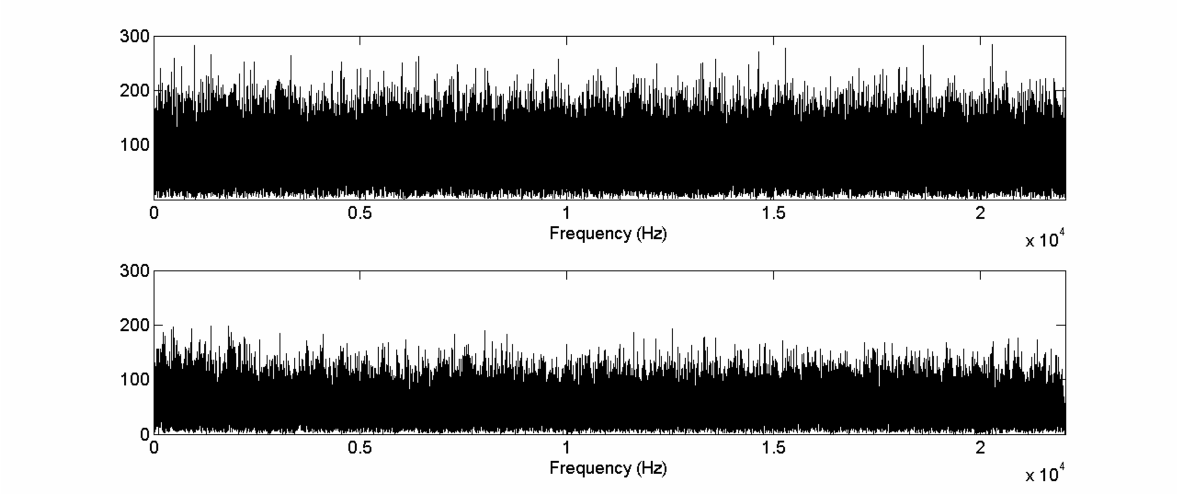


Figure 6.19: Long term average magnitude spectrum of input and resynthesized output from figure

The residual analysis/resynthesis system proposed and tested in this thesis is intended to offer an intuitive model with adaptability between long and short term broad band components. Whilst impulses are characterised with much better time resolution than would be possible with the equivalent Fourier analysis it is clear that the system does not perform as well with signals such as that just discussed. However the sinusoidality test employed does reduce the number of mis-identified peaks which reduces the cost of the sinusoidal synthesis. One possible solution to the lack of smoothness in magnitude estimates for stationary noise, whilst retaining the sharpness of transients, would be to perform some form of time domain transient analysis on each frame prior to residual analysis to determine the extent to which magnitudes should be smoothed. Where a transient is detected there would be little or no smoothing allowing the transient to be accurately followed in time. Where there are no sudden changes in signal energy smoothing would be applied that would produce a texture much closer to that of long term noise.

6.7.3 Acoustic signals

Having considered basic synthetic signal component types the performance of this system is now considered for examples of acoustic signals. Since the system is designed only for use with monophonic signals with partials which are greater than a given minimum spacing apart only these types of signals are considered. Short anechoic recordings of flute, violin, speaking voice and singing voice are surveyed. Different aspects and parameters of the system are discussed as relevant to the particular example.

6.7.3.1 Flute

The flute is a cylindrical metal pipe instrument which is open at both ends such that its modes of vibration contain all integer multiples of the fundamental frequency of vibration of the column of air that it contains. This vibration is produced as a result of interaction between the column of air and a jet of air produced by the performer [Rossing, 1990]. Whilst the majority of energy in the acoustic signal of a flute is due to harmonic vibration, noise in the form of turbulence around the mouthpiece and its resonant interaction with the pipe is also audible. In this example a single note G4, which has a fundamental of approximately 392 Hz, is played. Figure 6.20 shows the time domain waveforms of the input and resynthesized output and figure 6.21 the magnitude STFT of these signals. The upper frequency of the STFT plots is limited to 10 kHz so that lower partials can be clearly distinguished.

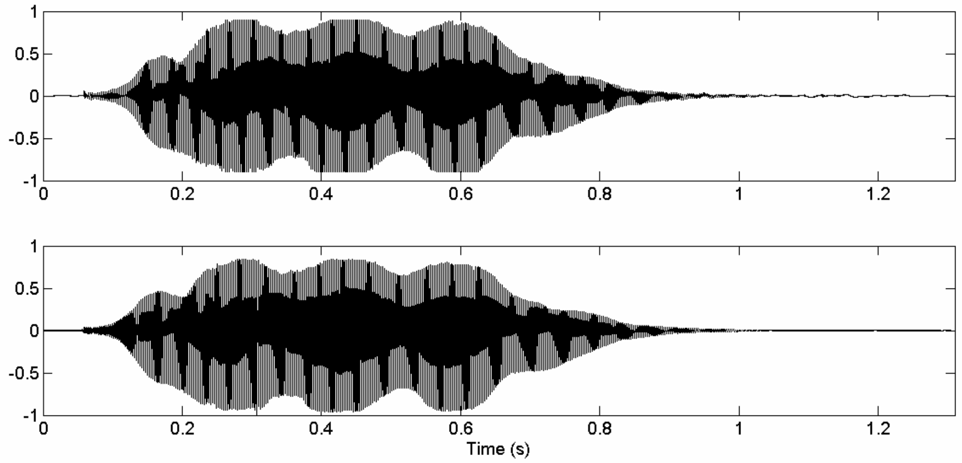


Figure 6.20: Time domain waveform of a flute note (top) and resynthesized output (bottom).

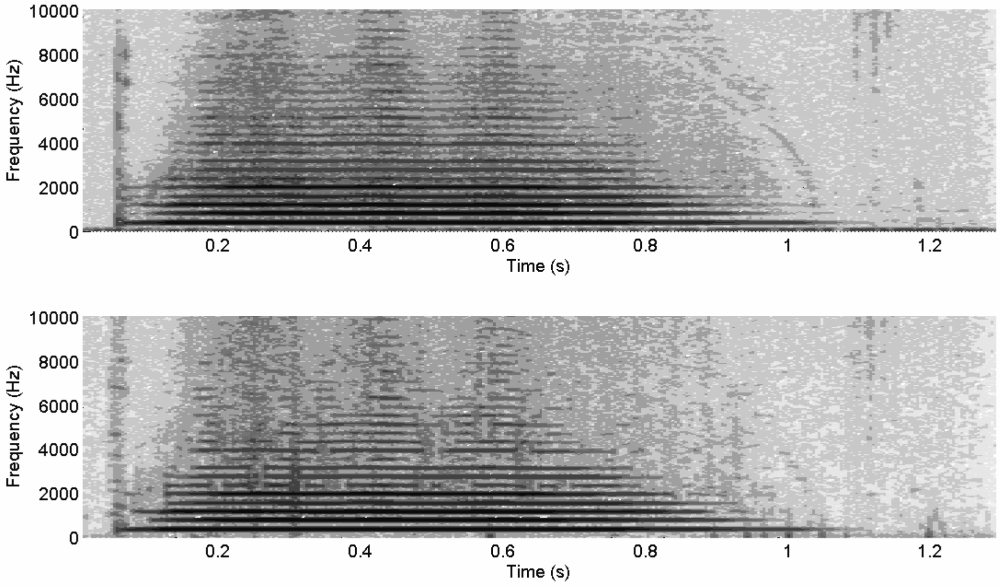


Figure 6.21: Magnitude STFTs of input (top) and synthesized output (bottom) signals shown in figure 6.20. A Blackman-Harris 1024 point analysis window with an overlap factor of 6 is used.

Whilst the output is audibly and visually distinguishable from the input the important temporal and spectral features are captured and the high amplitude, lower frequency harmonics are correctly linked as continuing sinusoids throughout the steady state portion of the note. Air turbulence components are plausibly modelled by the residual part of the system. It is at the onset and offset that differences between input and output can be seen and heard. During the offset some low level discontinuities can be heard where the trajectories of higher partials are broken up. At the onset the sinusoids ‘switch on’ almost instantaneously in the input and this can be heard as a distinct click which is fused with the start of the harmonics. At the default frame length and overlap this onset is smeared in time and the transient which is synthesized in the residual is too low in level and is temporally dislocated from the sinusoidal onsets. Reducing the overlap, which reduces the synthesis onset and offset times for sinusoids, does not restore the click which suggests that the onset is captured and smeared in the Fourier analysis rather than being introduced in the synthesis. In this case shortening the frame length to 513 samples does not improve this aspect either. Close-ups of the time domain waveform input and outputs at the onset are shown in figure 6.22.

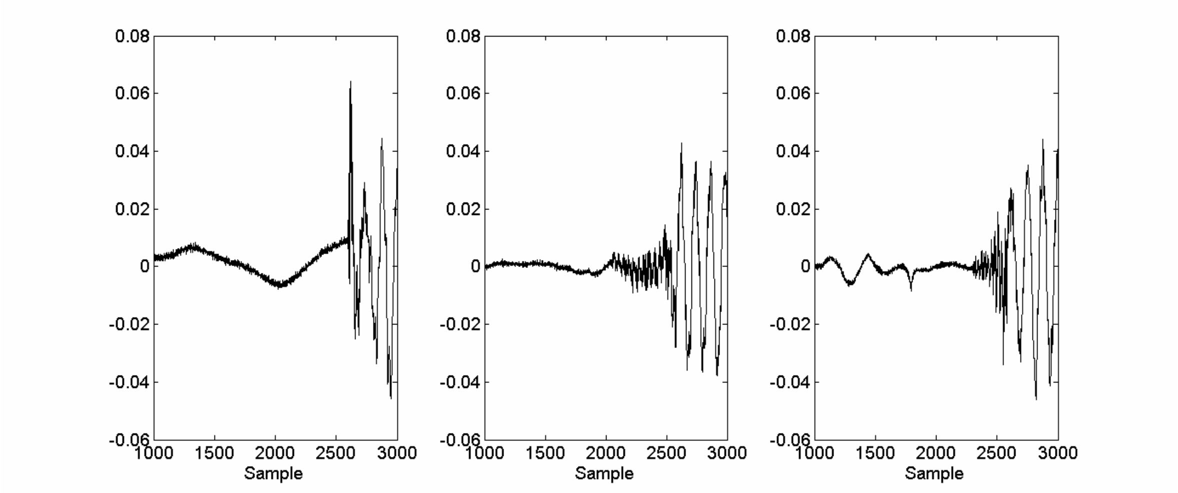


Figure 6.22: Time domain waveform of note onset for input (left), synthesized output for 1025 sample analysis frame (middle) and 513 sample analysis frame (right).

6.7.3.2 Violin

The violin is a stringed instrument. When it is bowed the string oscillates rapidly back and forth as it successively and rapidly held and released by the hair fibres of the bow as the bow is pulled across it. This causes the string to produce a sawtooth time domain waveform which is then filtered by the complex resonant structure of the instrument’s body. Thus a typical bowed violin note produces a large number of odd and even harmonics of relatively high amplitude giving it its bright tonal quality [Rossing, 1990]. The exact nature of interactions between bow and string which produce vibration are complex and can be chaotic giving rise to micro fluctuations in the parameters of the harmonics produced [Palumbi and Seno, 1998]. The note analysed here is G4. Figure 6.23 shows the time domain waveforms of the input and output and 6.24 the magnitude STFTs of these signals. Again, the upper frequency limit is 10 kHz so that harmonics can be clearly seen.

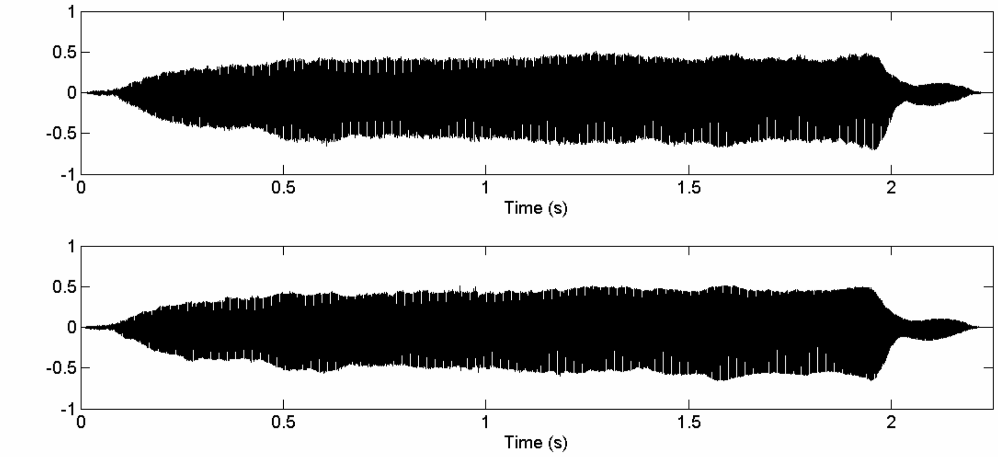


Figure 6.23: Time domain waveform of violin note (top) and resynthesized output (bottom).

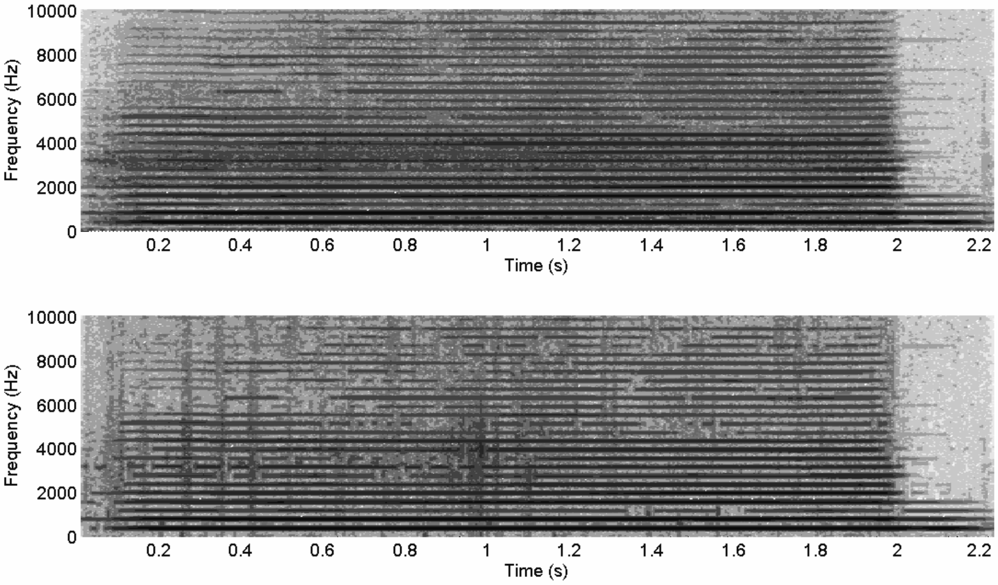


Figure 6.24: Magnitude STFTs of input (top) and synthesized output (bottom) signals shown in figure 6.23. A Blackman-Harris 1024 point analysis window with an overlap factor of 6 is used.

Here the resynthesis is not as successful as it is for the flute recording. There are some interruptions of the sinusoidal trajectories even for low harmonics which are at high amplitude. This is particularly noticeable at the note onset and for the third harmonic and is due to sinusoids being rejected as noise. Raising the VDT reduces the number of these discontinuities but at greater computational expense since this also causes more higher frequency noise components to be modelled by sinusoids. By raising the VDT from its default of 0.006 to 0.06 much of the 'bubbling' caused by sinusoids switching on and off is reduced and it is almost completely eliminated with a VDT of 0.6. This is partly due to missing links in trajectories being restored. It is also due to new sinusoids in the spectral vicinity of partial breaks apparently masking those breaks and creating a perception of a continuing sinusoid even where there is a break.

Raising the VDT has the effect of rendering the model almost wholly sinusoidal. At a VDT of 0.006 69.5% of peaks are rejected out of a total of 32 768, at 0.06 this figure is 49.8 % and at 0.6 only 33.2% are rejected. Where the VDT threshold is high the residual component contributes little to the output however it does contribute audible high frequency noise which does enhance the plausibility of the resynthesis. Whilst the identity of the synthesized output is clearly that of a violin there are still some audible disturbances such as small clicks, visible as the vertical striations in figure 6.24. In this specific case a higher VDT produces a more plausible output but at a higher computational cost due to the increased number of sinusoids that must be synthesized.

6.7.3.3 The human voice

Sound is produced by the vocal organ through the vibration of the vocal folds, which are driven by airflow from the lungs, within the resonant cavity of the vocal tract. The vibration of the vocal folds is a result of Bernoulli effect as air from the lungs passes between them. The folds open as a result of air pressure and then close again due to the drop in potential energy, and hence pressure, as air passes through the space between them at increasing velocity. The folds close more rapidly than they open producing a series of pulses producing a spectrally rich ‘buzzing’ excitation [Howard and Angus, 1996]. The spectral envelope of this excitation is then shaped by the vocal tract which acts as a time variant filter. In this way the timbre of the voice can be continuously varied to produce different vowel sounds. Voiced speech is produced when the vocal excitation is caused by the vocal folds closing and opening. Resonant peaks in the response of the vocal tract that combine to produce vowels are known as formants. Unvoiced speech is generated when the vocal folds are permanently open and are excited by air turbulence producing a broadband excitation. Consonants are produced by various forms of articulation, for example by the teeth and tongue. Plosives are produced by momentary blocking of the flow of air within the vocal tract and fricatives by constriction of air flow to produce turbulence. Therefore typical vocal sounds consist of short term stationary harmonic, as well as more stochastic, broadband components [Rossing, 1990].

The first vocal sound considered is an ‘ah’ vowel sung by an adult male at a pitch of G3 (fundamental of 196 Hz) with some vibrato. Time domain plots of the input and output signals for the default settings of overlap, frame length and VDT are shown in figure 6.25 and magnitude STFTs of these signals are shown in figure 6.26.

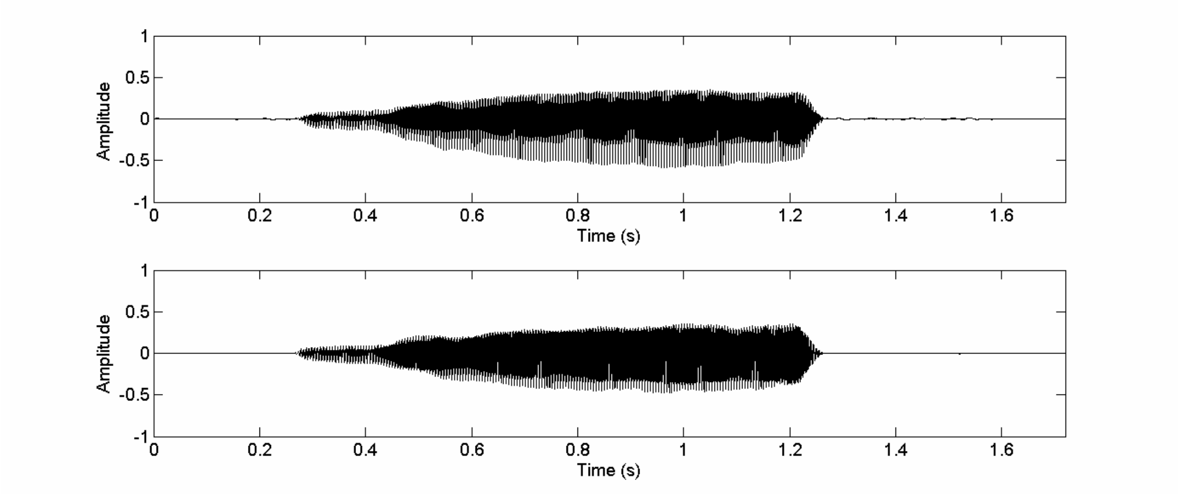


Figure 6.25: Time domain waveform of a sung male vocal ‘ah’ (top) and resynthesized output (bottom).

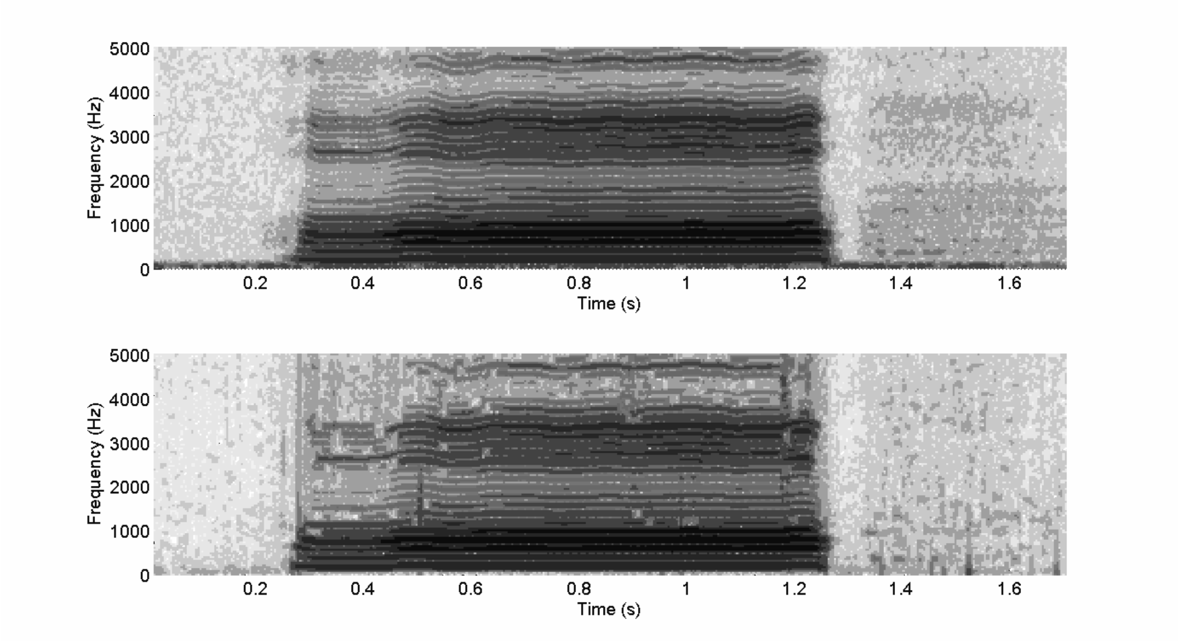


Figure 6.26: Magnitude STFTs of input (top) and synthesized output (bottom) signals shown in figure 6.25. A Blackman-Harris 1024 point analysis window with an overlap factor of 6 is used.

The overall temporal envelope of the input is well preserved and much of the harmonic structure is intact in the output but there are clear breaks in some of the partial trajectories. These occur mainly at the onset of the note and in the region between 4 and 5 kHz where partial amplitudes are lower due to a dip in the formant structure in this region. Also there is a low frequency component around 30 Hz which occurs throughout the input which is not represented in the output. This component is due to an extraneous rumbling noise which occurred during the recording. This is not classified as a sinusoid and rapid variations in the frequency and split estimates for the lowest equaliser spread energy from it into outlying parts of the spectrum rather than localising energy in this lowest band. Again, raising the VDT does fill in some gaps in the partial trajectories but at the cost of synthesizing more noise components with sinusoids. Reducing the analysis frame length to 513 samples produces a seriously corrupted output as the spacing between harmonics is then below the minimum spacing for the sinusoidality test (discussed in section 4.5.3).

The final acoustic audio example is the utterance “system output” spoken by an adult male. The fundamental frequency of vibration varies throughout this example, as usually happens in speech where pitch is varied to produce prosodic characteristics. The utterance contains fricatives in the ‘s’ of “system” and transients in the form of ‘t’ stop consonants in the word “output”. The time domain input and output is shown in figure 6.26 and the magnitude STFTs of these signals in figure 6.27.

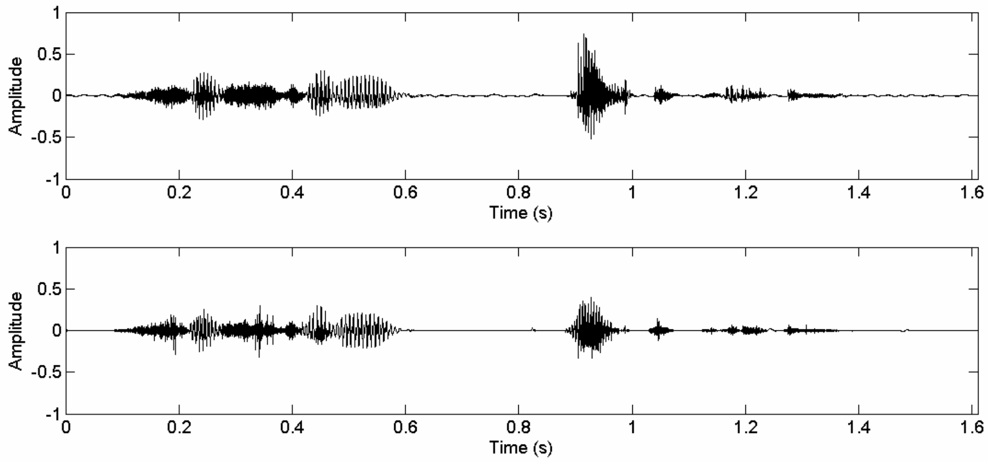


Figure 6.27: Time domain waveform of male utterance ‘system output’ (top) and resynthesized output (bottom).

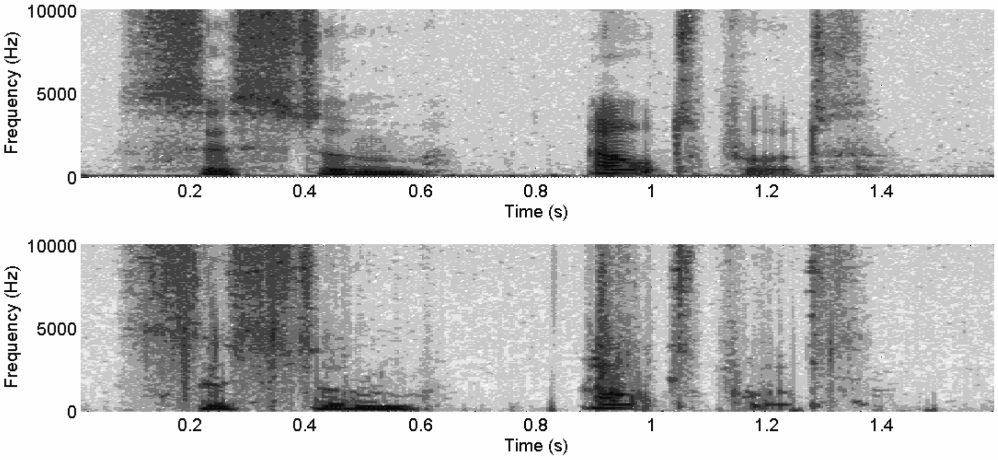


Figure 6.28: Magnitude STFTs of input (top) and synthesized output (bottom) signals shown in figure 6.27. A Blackman-Harris 1024 point analysis window with an overlap factor of 6 is used.

As can be seen from both the time domain and spectral plots there are some significant differences between the input and output and the intelligibility of the utterance is reduced in the output. The ‘s’ sounds are highly sibilant in the output and the ‘t’ sounds are smeared in time making them sound more like fricatives. Sinusoids are not properly assigned at the fast onset of voiced speech at the start of the word “output” and the residual contributes a burst of noise at this point which is not appropriate to this type of component. Raising the VDT to 0.6 improves the quality and intelligibility of the output but with some ‘birdy’ noise and clicks at sinusoidal offsets and onsets still audible. With the VDT at this level the residual contributes very little to the output and the model is essentially sinusoidal. Therefore, in its present form, the residual analysis and resynthesis system is not suited to the modelling of broad band speech components.

6.7.4 Computational performance

A prime motivation for the development and testing of algorithms described in this thesis is to enhance the possibilities for quasi real-time spectral modelling of audio signals. One aspect of this is improving the ability to make frame-by-frame decisions on how to model different components and how to link them between frames. The second important aspect is whether the methods used to make these decisions can execute quickly enough for real-time operation. Although *reSynth* has been implemented in MATLAB as an ‘offline’ (non real-time) process it is hoped that many of the algorithms it uses will form the basis of a real-time process in the future. This section discusses and compares the computational cost of different parts of the system. Since *reSynth* is a self contained function each time it is run there are a number of initialisation steps that need to be completed such as loading the two dimensional arrays for ΔA and Δf estimation and window generation. If *reSynth* were to be converted to a real-time processing environment such as a VST plug-in DLL these initialisation steps would occur only once when the plug-in is first launched in the host DAW. Therefore this section considers only those tasks which *reSynth* carries out on a frame by frame basis. Three of the example audio signal types previously discussed are considered for this analysis: the single stationary sinusoid since this is perhaps the most trivial signal that such a system can encounter, the long term white noise since, as discussed in section 6.7.2.2, this has proved to be a challenging signal type for the system to model and the flute recording, since this is an example of a real world signal which the system is able to model well. The data presented in this section has been generated using the MATLAB 'profiler' tool. Although this tool actually adds a processing overhead to functions it analyses, when running *reSynth* with the profiler on the execution time of the whole system is only increased by 0.9 %.

The profiling has been performed with the default settings of frame length (1025 samples) and overlap (2) and with the fastest wavelet analysis (1 undecimated and 9 decimated levels). The following stages in the analysis and resynthesis are profiled and are plotted as histograms, one for each audio example, with the duration of the audio plotted as a dotted line. These histograms are shown in figure 6.29:

1. Windowing of data and FFT for producing magnitude, phase and reassignment data.
2. Magnitude estimation, local maxima identification and initial frequency estimation.
3. Time reassignment data fitting, ΔA, Δf estimation and variance difference testing.
4. Amplitude correction and final frequency estimation.
5. Sinusoidal resynthesis.
6. Spectral subtraction, complex split wavelet analysis and estimation of EQ parameters.
7. Parametric equalisation.
8. Total computation for stages 1 to 7.

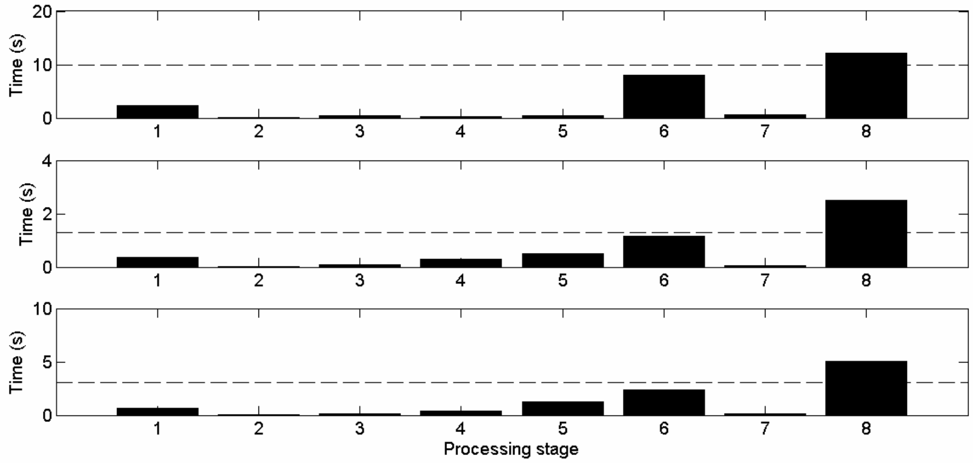


Figure 6.29: Computation time for different stages of *reSynth*. The inputs to the system are a 1 second stationary sinusoid (top), a flute recording (middle) and white noise. The dotted line indicates the duration of the audio being analysed.

In all cases the sinusoidal analysis and resynthesis is completed in less time than the duration of the audio signals indicating that on currently available hardware this part of the system can certainly be implemented in real-time. This is also the case for the subsequent complex wavelet analysis. However, this part of system does consume a considerable amount of the 'real-time' available for each of the example sounds. Approximately 20% of this time is spent on convolution operations, the rest on magnitude, bandwidth and frequency estimation of this data. The convolution for the wavelet analysis is performed by a MEX DLL which uses a convolution function optimised for the Intel family of processors [Intel, 2000]. The arctangent and subsequent phase unwrapping operations are also performed by a MEX function. The rest of the tasks within the main wavelet analysis functions are not implemented as DLLs since they use mainly vector operations. These results make it clear that implementation of all of the wavelet modelling tasks in a lower level language is necessary if such a combined sinusoidal and residual analysis system is to run in real-time on commonly available general purpose hardware. In the worst case example (the flute recording) the system is able to complete the analysis and resynthesis in just under twice the time duration of the actual signal.

6.8 Conclusions

This chapter has demonstrated a system for spectral modelling of audio that implements analysis methods developed in the previous two chapters. An overview of how such a system functions has been given and of its performance, both in terms of quality of audio output and execution time. The results obtained with the system in its present form demonstrate that a real-time partial tracking system that performs well on many, but at present certainly not all, types of sound can be realised. They also indicate that the wavelet analysis and equalised noise resynthesis method described in the previous chapter show good adaptation to and representation of impulses in both the time and frequency domains. As far as the author is aware, this is the first heterogeneous frame-by-frame spectral analysis system for wholly sample-by-sample time domain synthesis which has been devised. The potential inherent in the underlying model, which offers continuous (within the limits of a sampled audio system) control of the instantaneous frequency of *all* of its components with good time resolution beyond that of the homogeneous bandwidth enhanced system of [Fitz and Haken, 2002], is for highly flexible real-time and time variant pitch scaling of all those components. However, there are improvements to the current system which are required to improve fidelity and fusion of components for some types of sound.

The combination of complex Fourier and wavelet analysis shows promise but there is much scope for future work. In particular, additional analysis which assists in distinguishing between short and long term broad band components is needed to control smoothing of the equaliser gain in order to provide more plausible resyntheses of sounds such as speech fricatives, breathing and filtered noise generators used in widely available subtractive synthesizers. The sinusoidal part of the system would benefit from the inclusion of a hysteresis option in the partial identification. This would prevent long term partials from turning off during frames where they become corrupted by other components. If a partial was established (i.e. had been ‘continuing’ for a specified number of frames) then the system could be forced to wait for a number of frames before switching the partial off. The cost of this would be poorer time resolution at note offsets. The differences in partial tracking performance for the violin and flute examples suggest that a greater understanding of how the acoustics of musical instruments affects the sinusoidal detection algorithm is required. An area of future investigation of particular interest to the author is the examination of the relationship between high quality physical models of acoustic instruments and spectral models of the audio output they generate, identifying which aspects of acoustic systems affect the sinusoidality and ‘stochastic-ness’ of the output.

7 CONCLUSIONS

7.1 Introduction

This thesis has investigated the possibility of producing spectral modelling data in real-time using a combination of Fourier and wavelet methods. The prime motivating factor for this research has been the lack of real-time analysis tools available in existing spectral modelling systems, which limits the application of such tools in live and certain studio-based situations. The underlying question that runs through this investigation is: what can be inferred from a single frame of short-time analysis data that offers an intuitive spectral model of sound which can be realised in real-time and has good continuity, where it exists in the input sound, across frames without the need for overlapping (cross-faded) segments?

The work presented in previous chapters goes some way to answering this question and the key outcomes have been summarised and discussed at the end of each chapter. This final chapter draws together these answers and the new questions that they inevitably pose. The hypothesis is restated along with the methods employed to test it. The results and conclusions of each of the last three chapters are summarised and discussed and finally, future directions in which the work presented may be extended are considered.

7.2 Hypothesis

> *Wavelet and Fourier analysis methods can be combined to provide a meaningful and flexible parametric spectral modelling system that can be used as a real-time audio processor for monophonic sources. In particular, by high-accuracy modelling of their parameters (including those of non-stationarity), sinusoids can be identified and tracked from frame to frame and that complex wavelets can be used to produce a model of the residual which is time-variant and can be synthesized in the time domain.*

There have been three core questions to answer in the testing of this hypothesis:

1. Does there exist in a single frame of Fourier analysis data useful information for establishing which components are due to underlying stable sinusoids, how they evolve during the frame and, therefore, how they connect across the boundary between the previous and next frames?

2. Can a wavelet system be employed for the modelling of spectral components of audio signals, particularly those not well described by the behaviour of stable sinusoids?

3. Can a spectral modelling system exist which uses techniques developed in answer to the first two questions which can, firstly, operate on a frame-by-frame basis and, secondly, execute quickly enough to process data faster than it is acquired and, therefore, operate in real-time?

Chapter 4 examined how reassigned Fourier analysis data could be used to provide high accuracy estimates of a sinusoid’s mean frequency and amplitude and how their instantaneous values change over time. It also investigated how the interrelated behaviour of these parameters could indicate whether the component being analysed was a genuine stable sinusoid. Three different methods for evaluating sinusoidality were compared.

Chapter 5 described a method employing B-spline wavelets, since they closely approximate modulated windows with constant instantaneous frequency, for the analysis of audio signals. This system attempts to model components in terms of the control data for time-variant parametric equalisers, namely centre frequency, gain and bandwidth. To allow user-controlled mediation between ‘over-completeness’ and computational cost the analysis offers differing degrees of decimation.

Chapter 6 put the techniques and discoveries of the previous two chapters into a practical context by describing a frame-by-frame spectral modelling system. This system takes an input audio signal, produces a spectral model of it and resythesizes the input audio by using the model data to control time-variant sinusoidal oscillators and equalisers which are commonly encountered and understood tools for the creation and manipulation of audio signals. The quality of the system was investigated by testing with simple synthetic components and ‘real-world’ acoustic sounds and by profiling the time taken for key parts of the system to execute.

7.3 Summary of results and conclusions

7.3.1 Sinusoidal analysis

The work undertaken into non-stationary sinusoidal modelling and described in chapter 4 yielded the following discoveries:

- Phase distortion analysis can be adapted for use with reassignment data in order to obtain estimates for intra-frame frequency and amplitude change of individual sinusoidal components.

- Improved modelling of the distortion data can yield estimates which vary significantly less with frequency, particularly for extreme amplitude and frequency change.

- The influence of amplitude change on the estimation of frequency change, and vice versa, can be reduced by an iterative approach to their estimation using simple 2D array look-up and interpolation.

- The variance of the reassignment data to a least-squares polynomial fit can be used to indicate the sinusoidal 'behaviour' of the estimates obtained.

- As suggested in other work [Hainsworth and Macleod, 2003b], but not investigated, an alternative sinusoidality measure can be derived by comparing phase and magnitude reassignment data.

- The two tests of sinusoidality developed offer comparable performance to an existing correlation method for non-stationary sinusoids in terms of discrimination and far superior performance in terms of computational cost.

- The discrimination capability of all of the correlation methods tested diminishes with increasing non-stationarity and the presence of noise. Of the two reassignment measures proposed the variance difference method performs best for closely clustered sinusoids but the time reassignment difference method performs better where the signal contains a high level of noise.

7.3.2 Wavelet modelling of audio signals

The work presented in chapter 5 to develop a new, frame based, wavelet analysis system demonstrated that:

- Complex B-spline wavelets can accurately determine the frequency and magnitude of sinusoidal components provided the input frame is properly initialised in the frequency domain with a sinc function.

- That control over the amount of decimation performed within the transform offers mediation between shift invariance of magnitude estimates and the ranges within which aliasing of frequencies can occur at each scale.
- That knowledge of the frequency of a component can be used to correct magnitude estimates so that a single synthesis equaliser can be used for reconstruction.
- An additional ‘frequency splitting’ stage in the analysis offers a simple measure of the bandwidth of an underlying component. Estimates of bandwidth for isolated sinusoids and impulses are very precise.
- Estimates of magnitude and frequency are adversely affected by the window applied in preparation for prior Fourier analysis. The frequency estimates can be corrected by combining two polynomial functions that take the ‘distance from centre of frame’ and ‘deviation of original estimate from centre frequency of wavelet’, however this is a costly process in a real-time context. ‘Un-windowing’ the signal prior to wavelet analysis is not possible due to the amplification of artifacts introduced by the Hilbert transform although windowing effects can be reduced by increasing the analysis overlap, therefore reducing the length of the synthesis frame and constraining the analysis to a small region around the centre of the input frame.
- For short-time wavelet analysis the behaviour of the wavelet filters deviates from that predicted by the modulated Gaussian model as their support extends beyond the length of the analysis frame. The effects upon magnitude can be reduced by empirical methods and, again, by increasing the overlap.

7.3.3 Frame-by-frame spectral modelling

The frame-by-frame spectral modelling system described in chapter 6 has the following capabilities:

- The ability to produce non-stationary sinusoids which can be tracked across frame boundaries with minimal discontinuities for many types of signals and have piecewise quadratic phase and piecewise exponential amplitude.
- Where overlapping analysis frames are employed, the latency between input and output is less than the analysis frame length and, for a hop size of one sample, is half the length of a single analysis frame.

- The system can analyse, model and synthesize what it determines to be the sinusoidal part of the input signal within real-time using currently available general purpose computer hardware.
- The residual part of the signal can be analysed, modelled and resynthesized in the time domain using parametric equalisers applied to the output of a broadband noise source just within (95% of, for the worst case example) real-time .
- The residual part of the system is able to reproduce impulses with excellent time resolution using a single frame of analysis data. It can also adapt well, in terms of time and frequency resolution, to impulses or widely spaced sinusoids.
- Unlike other heterogeneous systems it generates all of its output in the time domain on a sample-by-sample basis.

The system currently has the following limitations:

- Where a sinusoid is misclassified as a noise component during a track the discontinuity is often audible. This can be a particular problem for signals where the distinction between noise and sinusoids is not clearly demarcated.
- The residual analysis and resynthesis system does not model stationary noise well and for certain types of sound, such as speech, there is considerable dissimilarity between residual components in their original and resynthesized forms. This also leads to poor perceptual fusion of different component types.
- The combined sinusoidal and residual analysis, modelling and resynthesis takes longer than real-time to execute on the test system although the execution time for signals tested, which range from a single sinusoid to broadband noise, does not exceed more than twice the duration of the input signal.

7.4 Consideration of hypothesis

This thesis has demonstrated that a frame-by-frame spectral modelling system that can produce an intuitive description and high quality resynthesis of certain types of input sound can be realised. The system, despite much of it being implemented in a high level language, can produce output within a time period which is close to the duration of the input signal. The quality of the sinusoidal model, which produces single, non-stationary, frame segments with only phase matching, and no interpolation, across frame boundaries is enhanced by the high accuracy parameter estimates achieved using the algorithms described in chapter 4. The description of sinusoids in the presence of, and their separation from, noise is not perfect and inaccuracies still exist in the estimation of extreme, combined frequency and amplitude change.

The equalised noise model proposed is successful for certain types of signal, adapting well to time and frequency domain impulses and adding a ‘liveliness’ to the flute and violin examples which would be missing, or costly to produce, in a sinusoid only model. However the sinusoidal and residual parts of the synthesis are not well fused for some signals with the residual model functioning poorly for some components and the sinusoidal only model performs better in these cases. As discussed in chapter 5, the wavelet analysis system and its application to time domain residual modelling and synthesis is entirely novel and, whilst it demonstrates potential in this area, requires further refinement before it will be suited to the wide range of sounds that a general spectral modelling system should expect to encounter.

On the evidence presented in this thesis there is considerable existing capability for real-time spectral modelling of many types of audio signals and many worthwhile avenues for exploration in order to extend the range of signals that can be successfully modelled by such a system and the quality of synthesis from such a model.

7.5 Further work

As well as discussion of results and the presentation of conclusions, each of the last three chapters has suggested ways in which the work described within them could be taken further. These are summarised in this section along with additional thoughts on how knowledge in each area might be improved.

Chapter 4 showed that increasing the order of the polynomial fitted to the reassignment data from first to second results in less variation of amplitude and frequency change estimates with distance of the component from the centre of the peak analysis bin. This begs the obvious question: can this variation be further reduced by the use of even higher order polynomials? However, whilst this may give better estimates it may also reduce the variance from the fit for all component types reducing the ability to discriminate between noise and sinusoids. Combing the time reassignment difference and variance methods may be the solution to obtaining more accurate estimates whilst retaining the discrimination capability. As suggested at the end of chapter 4, with so many methods beginning to appear for non-stationary parameter estimation and with available computational power continuing to grow, the long term future of frame based identification and description of sinusoids may lie in a meta-analysis of the output of all or many of these different techniques.

As discussed in chapter 5 the new work on biquadratic implementations of higher order digital parametric equalisers in [Orfanidis, 2005] will offer coverage of the audio spectrum that can be more accurately controlled than that of the overlapping equalisers employed in the *reSynth* system. In fact these higher order filters could act as 'equivalent rectangular bands' to those of the analysis filters. Also noted was the need for a greater understanding of the relationship between the centre frequencies of the split filters and the bandwidth of the underlying component that they are influenced by. Presently the magnitude correction does not consider the width of the underlying component. Within the band constraints of the filters this does not affect the correction to a significant degree and the magnitude correction will always improve rather than degrade the estimate. However this correction could possibly be improved. The difficulty here is that the integral of a Gaussian function is not analytically defined and so table look-up or another approach would need to be found.

Chapter 6 highlighted the need for a greater understanding of how sound production in acoustic instruments affects the behaviour of the sinusoids produced. This would inform the choice of sinusoidality thresholds for such instruments preventing partial tracks from switching on and off during steady-state portions of sound. Hysteresis in the partial tracking was also suggested to prevent such breaks occurring. Also, more accurate wavelet modelling, as outlined in the previous paragraph would aid the acoustic plausibility, and fusion with deterministic elements, of resynthesized components.

It is hoped that these will act as starting points not just for this author but for others working, or interested, in this area of sound modelling and transformation. The wide range of methods and applications covered will benefit most from a wide range of expertise and viewpoints. Recalling, and adapting, a quotation from the first chapter: "No one knows more than everyone".

APPENDIX A: CODE LISTING DATA CD

This appendix provides files containing text of the code for the system described in chapter 6 and used to generate the CD of audio examples (Appendix B) for this thesis. There is a single MATLAB 'm' file (resynth.m) and a number of C files from which MATLAB MEX functions have been compiled. The code is presented on a CD-ROM which accompanies this thesis to enable it to be viewed within a computer based development environment or a text editor if such an environment is not available. This CD-ROM can be found inside the back cover of the thesis. It is recommended that the MATLAB Editor/Debugger is used view the 'm' file and a C editing environment, such as that provided in Microsoft *Visual Studio* is used to view the C files. However, if no such software is available, this code can be viewed in a basic text editor or word processing application.

Wherever MEX files are used there is also an equivalent sub function within reSynth.m. Where these alternative functions exist they are preceded by the comments 'MATLAB version of function' and 'C version of function'. Both have identical functionality and only one is required to perform the required task.

In addition to these files there are four MATLAB data files (.mat) which contain the 2D arrays used for ΔA and Δf estimation, determining the expected variance and correcting mean amplitude estimates.

The list of files on the CD-ROM is as follows:

Resynth.m: main MATLAB program that implements the entire system.

CConv.c: MEX file to perform linear convolution.

CDestimate.c: MEX file to produce ΔA, Δf and σ^2 estimates from RDA measures.

CEQ.c: MEX file to perform parametric equalisation.

CFindPeaks.c: MEX file to search for local maxima in magnitude spectrum.

CPhaseUnwrap.c: MEX file that converts real and imaginary parts of complex wavelet output to phase and unwraps these values.

CSineGenerateCP.c: MEX file that synthesizes a single sinusoid given a value for the phase at the centre of the synthesis frame.

CSineGenerateSP.c: MEX file that synthesizes a single sinusoid given a value for the phase at the start of the synthesis frame.

CSpectralSubtract.c: MEX file that performs spectral subtraction of sinusoidal signal.

CTimeOffsetFit.c: MEX file that fits a second degree polynomial to time reassignment offset data around a magnitude peak to produce RDA measures.

daTable0_96_100.mat: data file containing the 2D array for ΔA estimation.

dfTable0_260_100.mat: data file containing the 2D array for Δf estimation.

chiSquaredTable0_96_260.mat: data file containing the 2D array of expected variance.

ampTable0_96_260.mat: data file containing 2D array of expected variance

APPENDIX B: AUDIO EXAMPLES CD

This appendix is in the form of an audio CD containing sound examples of the *reSynth* spectral modelling system described in chapter 6 of this thesis. Many of the examples included here are discussed in the chapter. A track listing with brief notes on each example is given below.

1. **Slow vibrato.** There are three items on this track: (i) input, (ii) resynthesized output with a frame length of 1025 samples, an overlap of 2 and a VDT of 0.006, and (iii) output (with the same analysis settings) but with the pitch of the output shifted down by a semitone. This example is discussed in section 6.7.1.2.

2. **Fast vibrato.** There are four items on this track: (i) input, (ii) output with a frame length of 1025 samples, an overlap of 2 and a VDT of 0.006, (iii) resynthesized output as for the previous item but with a frame length of 513 and (iv) resynthesized output with a frame length of 513 samples and a VDT of 0.06. This example is discussed in section 6.7.1.2.

3. **Impulse.** There are three items on this track: (i) input, (ii) output with a frame length of 1025 samples, an overlap of 2 and a VDT of 0.006, and (iii) output as for the previous item but pitch shifted up by an octave. Impulses are discussed in section 6.7.2.1

4. **Residual sinusoid.** There are two items on this track. The sinusoidal part of the system and the spectral subtraction have been disabled and the residual is modelling an input 1 kHz sinusoid. (i) is the output of the residual part of the system and (ii) is the output with the pitch shifted up by a perfect fifth. Residual analysis of mis-classified sinusoids is discussed in section 6.7.1.3.

5. **Flute.** There are five items on this track: (i) input, (ii) resynthesized output with a frame length of 1025 samples, an overlap of 2 and a VDT of 0.006, (iii) sinusoidal part of resynthesized output with the same settings, (iv) residual part of the output with the same settings, and (v) combined output, pitch shifted up by an octave. The performance of the system for this instrument is discussed in section 6.7.3.1.

6. **Violin**. There are four audio items on this track: (i) input, (ii) output with a frame length of 1025 samples, an overlap of 2 and a VDT of 0.006, (iii) output with the

same settings except the VDT is set to 0.06 and (iv) output with the same settings but with the VDT increased to 0.6. These items are discussed in section 6.7.3.2.

7. **Male singing.** There are three items on this track: (i) input, (ii) output with a frame length of 1025 samples, an overlap of 2 and a VDT of 0.006 and (iii) output with the same settings apart from the VDT which has been raised to 0.06. These items are discussed in section 6.7.3.3.
8. **Male speaking.** There are three items on this track: (i) input, (ii) output with a frame length of 1025 samples, an overlap of 2 and a VDT of 0.006 and (iii) output with the same settings but with the VDT increased to 0.06. These items are discussed in section 6.7.3.3.

LIST OF ABBREVIATIONS

Most of these abbreviations are in common use however some refer specifically to thesis, where this is the case it is denoted in brackets).

DFT: Discrete Fourier transform (sometimes referred to as the discrete-time Fourier series).

DWT: Discrete wavelet transform (decimated or undecimated).

DLL: Dynamically linked library.

FFT: Fast Fourier transform.

FWT: Fast wavelet transform (decimated, also known as Mallat, wavelet transform).

FRD: Frequency reassignment difference (introduced in this thesis).

HUT: Helsinki University of Technology

MATLAB: Matrix laboratory (scientific computing software produced by Mathworks Inc.)

MEX: MATLAB executable file (MATLAB function written in C or FORTRAN).

PDA: Phase distortion analysis.

PM: Used in this thesis to refer to physical modelling in general.

RDA: Reassignment distortion analysis (introduced in this thesis).

SM: Used in this thesis to refer to spectral modelling in general.

SMS: Spectral Modelling Synthesis (specific SM system devised by Xavier Serra).

SPL: Sound pressure level (dB SPL is referenced to $20\mu\text{Pa}$).

STFT: Short-time Fourier transform

TRD: Time reassignment difference (introduced in this thesis).

VDT: Variance difference threshold (introduced in this thesis).

VST: Steinberg’s Virtual Studio Technology, an audio processing and synthesis plug-in format

REFERENCES

[Amatriain et al, 2002] X. Amatriain, “Spectral Processing”, chapter in (ed. Zölzer), *DAFX – Digital Audio Effects*, John Wiley, Chichester.

[American Standards Association, 1960] American Standards Association, “Acoustical Terminology SI”, American Standards Association, New York.

[Abe and Smith, 2005] M. Abe and J. Smith, “AM/FM Rate Estimation for Time-Varying Sinusoidal Modeling”, *Proceedings of the 2005 IEEE Conference on Acoustics, Speech and Signal Processing*.

[Abry and Flandrin, 1994] P. Abry and P. Flandrin, “On the Initialization of the Discrete Wavelet Transform”, *IEEE Signal Processing Letters*, vol. 1, pp. 32-34.

[Auger et al, 1996] F. Auger et al, “Time-Frequency Toolbox Tutorial”, http://tftb.nongnu.org/

[Auger and Flandrin, 1995] F. Auger and P. Flandrin, “Improving the Readability of Time-Frequency and Time-Scale Representations by the Reassignment Method”, *IEEE Transactions on Signal Processing*, vol. 43, pp.1068-1089.

[Bastiaans, 1980] M. Bastiaans, “Gabor’s Expansion of a Signal into Gaussian Elementary Signals”, *Proceedings of the IEEE*, vol. 68, pp. 538-539.

[Bellingham and Gorges, 1998], D. Bellingham and P. Gorges, “Kawai K5000: Intoduction to Additive Synthesis, Advanced Sound Design, Tips and Tricks”, Wizoo, Cologne.

[Berger et al, 1994], J. Berger et al, “Removing Noise from Music Using Local Trigonometrical Bases and Wavelet Packets”, *Journal of the Audio Engineering Society*, vol. 42, pp. 808-818.

[Blauert and Laws, 1978] J. Blauert and P. Laws, “Group Delay Distortions in Electroacoustical Systems”, *Journal of the Acoustical Society of America*, vol. 63, pp. 1478-1483.

[Born, 1995] G. Born, “Rationalizing Culture. IRCAM, Boulez and the Institutionalization of the Musical Avant-Garde”, University of California Press, Berkeley.

[Boulanger, 2000] R. Boulanger (ed.), “The CSound Book”, The MIT Press, Cambridge.

[Bradley, 2003] A. Bradley, “Shift-Invariance in the Discrete Wavelet Transform”, *Proceedings of the 7th Conference on Digital Image Computing: Techniques and Applications*, pp. 29-38.

[Bristow-Johnson] R. Bristow-Johnson, “Cookbook Formulae for Audio EQ Biquad Filter Coefficients”, http://www.musicdsp.org/files/Audio-EQ-Cookbook.txt

[Bristow-Johnson, 1994] R. Bristow-Johnson, “The Equivalence of Various Methods of Computing Biquad Coefficients for Audio Parametric Equalizers”, *Presented at the 97th Audio Engineering Society Convention*, Preprint 3906.

[Brown, 1990] J. Brown, “Calculation of a Constant-Q Spectral Transform”, *Journal of the Acoustical Society of America*, Vol. 89, No. 1, pp. 425-434

[Cage, 1961] J. Cage, “Silence”, Wesleyan University Press, Conneticut.

[Cambridge Advanced Learner’s Dictionary] Unattributed, “Music”, *Cambridge Advanced Learner’s Dictionary,* http://dictionary.cambridge.org

[Carlos, 1986] W. Carlos, “Tuning At the Crossroads”, *Computer Music Journal*, Vol. 11, No. 1, pp. 29-43.

[Cavaliere and Piccialli, 1997] S. Cavaliere and A. Piccialli, “Granular Synthesis of Musical Signals”, *Musical Signal Processing*, Swets and Zeitlinger, Lisse.

[Chowning, 1973] J. Chowning, “The Synthesis of Complex Audio Spectra by Means of Frequency Modulation”, *Journal of the Audio Engineering Society*, vol. 21, pp. 526-534.

[Chui and Wang, 1992] C. Chui and J. Wang, “On Compactly Supported Spline Wavelets and a Duality Principle”, *Transactions of the American Mathematical Society*, vol. 330, pp. 903-915.

[Cohen, 1994] L. Cohen, “Time-Frequency Analysis”, Prentice Hall, New Jersey.

[Convention Industry Council, 2004] Unattributed, “APEX Industry Glossary”, http://glossary.conventionindustry.org

[Cooley and Tukey, 1965] J. Cooley and J. Tukey, "An Algorithm for the Machine Computation of the Complex Fourier Series", *Mathematics of Computation*, vol. 19, pp. 297-301.

[Cross, 2003] I. Cross, "Music as Biocultural Phenomenon", *Proceedings of the Cambridge Music Processing Colloquium 2003*, pp. 15-21.

[Daubechies, 1992] I. Daubechies, "Ten Lectures on Wavelets", Society for Industrial and Applied Mathematics, Philadelphia.

[Davies, 1996] G. Hartstone and T. Spath, "Disc Cutting" chapter in J. Borwick (ed.), *Sound Recording Practice*, Oxford University Press, Oxford.

[Debnath, 2002] L. Debnath, "Wavelet Transforms and Their Applications", Birkhäuser, Boston.

[Desainte-Catherine and Marchand, 2000] M. Desainte-Catherine and S. Marchand, "High Precision Fourier Analysis of Sounds Using Signal Derivatives", *Journal of the Audio Engineering Society*, vol. 48, pp. 654-667.

[Dolson, 1986] M. Dolson, "The Phase Vocoder: A Tutorial", *Computer Music Journal*, vol. 10, No. 4, pp. 14-27.

[DMRN] Digital Music Research Network, http://www.elec.qmul.ac.uk/dmrn/

[Dutilleux, 1988] P. Dutilleux, "An Implementation of the "Algorithme à Trous" to Compute the Wavelet Transform", chapter in J. Combes (ed.), *Wavelets: Time-Frequency Methods and Phase Space*, Springer Verlag, Heidelberg.

[Elfataoui and Mirchandani, 2004] M. Elfataoui and G. Mirchandani, "A Frequency Domain Method for Generation of Discrete-Time Analytic Signals", http://www.emba.uvm.edu/~mirchand/publications/ehilbert2.pdf

[Emmerson, 1994] S. Emmerson, "'Live' Versus 'Real-Time'", *Contemporary Music Review*, vol. 10, pp. 95-101.

[Fitz and Haken, 2002] K. Fitz and L. Haken, “On the Use of Time-Frequency Reassignment in Additive Sound Modeling”, *Journal of the Audio Engineering Society*, vol. 50, pp. 879-893.

[Fitz and Haken, 2003] K. Fitz and L. Haken, “Current Research in Real-Time Sound Morphing”, http://www.cerlsoundgroup.org/RealTimeMorph/

[Flandrin, 1989] P. Flandrin, “Some Aspects of Non-Stationary Signal Processing with Emphasis on Time-Frequency and Time-Scale Methods”, chapter in J. Combes (ed.), *Wavelets: Time-Frequency Methods and Phase Space*, Springer Verlag, Heidelberg.

[Flandrin et al, 1995] P. Flandrin et al, “Reassigned Scalograms and their Fast Algorithms”, *Proceedings of SPIE*, vol. 2569, pp. 152-163.

[Freed et al, 1993], “Performance, Control and Synthesis of Additive Synthesis on a Desktop Computer Using FFT^{-1}”, *Proceedings of the 1993 International Computer Music Conference (ICMC-93)*.

[Frigo, 1999] M. Frigo, “A Fast Fourier Transform Compiler”, *Proceedings of the 1999 ACM SIGPLAN Conference on Programming Language Design and Implementation*, http://www.fftw.org/pldi99.pdf

[Gautschi, 1965] W. Gautschi, “Error Function and Fresnel Integrals”, chapter in M. Abramowitz and I. Stegun (eds.), *Handbook of Mathematical Functions*, Dover Publications, New York.

[Glasberg and Moore, 1990] B. Glasberg and B. Moore, “Derivation of Auditory Filter Shapes from Notched-Noise Data”, *Hearing Research*, vol. 47, pp.103-138.

[Gonzalez and Lopez, 2001] A. Gonzalez and J. Lopez, “PC Based Real-Time Multichannel Convolver for Ambiophonic Reproduction”, *Proceedings of the AES 19th International Conference on Surround Sound Techniques*

[Grossman et al, 1989] A. Grossman et al, “Reading and Understanding Continuous Wavelet Transforms”, chapter in J. Combes (ed.), *Wavelets: Time-Frequency Methods and Phase Space*, Springer Verlag, Heidelberg.

[Hainsworth and Macleod, 2003] S. Hainsworth and M. Macleod, “On Sinusoidal Parameter Estimation”, *Proceedings of the 6th International Conference on Digital Audio Effects (DAFx03)*, pp. 151-156.

[Hainsworth and Macleod, 2003b] S. Hainsworth and M. Macleod, “Time-Frequency Reassignment: Measures and Uses”, *Proceedings of the Cambridge Music Processing Colloquium 2003*, pp. 36-39.

[Hainsworth et al, 2001] S. Hainsworth et al, “Analysis of Reassigned Spectrograms for Musical Transcription”, *Proceedings of the 2001 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*.

[Harris, 1978] F. Harris, “On the Use of Windows for Harmonic Analysis with Discrete Fourier Transform”, *Proceedings of the IEEE*, vol. 66, pp.51-83.

[Hartstone and Spath, 1996] G. Hartstone and T. Spath, “Film” chapter in J. Borwick (ed.), *Sound Recording Practice*, Oxford University Press, Oxford.

[Helmholtz trans. Ellis, 1954] H. Helmholtz (translated A. Ellis), “On the Sensation of Tone as a Physiological Basis for the Theory of Music”, Dover, New York.

[Horner et al, 1997] A. Horner et al, “Modelling Small Chinese and Tibetan Bells”, *Journal of the Audio Engineering Society*, vol. 45, pp.148-159.

[Howard and Angus, 2000] D. Howard and J. Angus, “Acoustics and Psychoacoustics”, Focal Press, Oxford.

[Howard and Rimell, 2003] D. Howard and S. Rimell, “CYMATIC: A Tactile Controlled Physical Modelling Instrument”, *Proceedings of the 6th International Conference on Digital Audio Effects (DAFx03)*, pp. 112-117.

[Intel, 2000] Unattributed, “Intel Signal Processing Library Reference Manual”, Intel Corporation, http://www.intel.com

[IS0226, 2003] International Standard 226:2003, “Acoustics. Normal equal-loudness-level contours”, International Standards Organization.

[ISO/IEC, 1992] ISO/IEC 11172, “Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s”, International Standards Organisation and International Electrotechnical Commission.

[James et al, 1999] G. James at al, “Advanced Modern Engineering Mathematics”, Pearson Education, Harlow.

[Jorgensen, 1995] F. Jorgensen, “The Complete Handbook of Magnetic Recording”, McGraw-Hill, New York.

[Keiler and Marchand, 2002] F. Keiler and S. Marchand, “Survey on Extraction of Sinusoids in Stationary Sounds”, *Proceedings of the 5th International Conference on Digital Audio Effects (DAFx02)*, pp. 51-58.

[Kennedy, 1980] M. Kennedy (after Scholes), “The Concise Oxford Dictionary of Music”, Oxford University Press, Oxford.

[Kernighan and Ritchie, 1988] B. Kernighan and D. Ritchie, “The C Programming Language”, Prentice Hall, New Jersey.

[Kingsbury, 1999] N. Kingsbury, "Shift invariant properties of the Dual-Tree Complex Wavelet Transform", *Proceedings of the 1999 IEEE Conference. on Acoustics, Speech and Signal Processing*.

[Kingsbury, 2001] N. Kingsbury, “Complex Wavelets for Shift Invariant Analysis and Filtering of Signals”, *Journal of Applied and Computational Harmonic Analysis*, vol. 10, pp. 234-253.

[Kronland-Martinet, 1988] R. Kronland Martinet, “The Wavelet Transform for Analysis, Synthesis and Processing of Speech and Music Sounds”, *Computer Music Journal*, vol. 12, No. 4, pp. 11-20.

[Lagrange, 2004] M. Lagrange, “Modélisation Sinusoïdale des Sons Polyphoniques”, PhD Thesis, University of Bordeaux, France.

[Lagrange et al, 2002] M. Lagrange et al, “Sinusoidal Parameter Estimation in a Non-Stationary Model”, *Proceedings of the 5th International Conference on Digital Audio Effects (DAFx02)*, pp. 59-64.

[Landy et al], L. Landy et al, “Electroacoustic Resource Site (Glossary)”, http://www.ears.dmu.ac.uk/rubriqueGlossary.php3

[Laroche and Dolson, 1997] J. Laroche and M. Dolson, “About This Phasiness Business”, *Proceedings of the 1997 International Computer Music Conference (ICMC-97),* pp.55-58.

[Laroche and Dolson, 1999] J. Laroche and M. Dolson, “New Phase Vocoder Techniques for Real-Time Pitch Shifting, Chorusing, Harmonizing, and Other Exotic Audio Modifications”, *Journal of the Audio Engineering Society*, vol. 47, pp. 928-936.

[Lazzarini et al, 2005] V. Lazzarini et al, “Alternative Analysis-Synthesis Approaches for Timescale, Frequency and Other Transformations of Musical Signals”, *Proceedings of the 8th International Conference on Digital Audio Effects (DAFx05)*, pp. 18-23.

[Levine, 1998] S. Levine, “Audio Representations for Data Compression and Compressed Domain Processing”, PhD Dissertation, Stanford University, USA.

[Lynn and Fuerst, 1994] P. Lynn and W. Fuerst, “Introductory Digital Signal Processing with Computer Applications”, John Wiley, Chichester.

[Mallat, 1999] S. Mallat, “A Wavelet Tour of Signal Processing”, Academic Press, San Diego.

[Marchand, 2000] S. Marchand, “Sound Models for Computer Music (Analysis, Transformation, Synthesis)”, PhD Thesis, University of Bordeaux, France.

[Masri, 1996] P. Masri, “Computer Modelling of Sound for Transformation and Synthesis of Musical Signals”, PhD Thesis, University of Bristol, UK.

[Master, 2002] A. Master, “Nonstationary Sinusoidal Model Frequency Parameter Estimation via Fresnel Integral Analysis”, Technical Report, Stanford University.

[Master and Liu, 2003] A. Master and Y. Liu, “Nonstationary Modeling with Efficient Estimation of Linear Frequency Chirp Parameters”, *Proceedings of the 1995 IEEE Conference on Acoustics, Speech and Signal Processing*, pp. 656-659.

[Mathworks, 2006] Unattributed, MATLAB user documentation, http://www.mathworks.com/access/helpdesk/help/helpdesk.html

[McAulay and Quatieri, 1986] R. McAulay and T. Quatieri, “Speech Analysis/Synthesis Based on a Sinusoidal Representation”, *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 34, pp. 744-754.

[Microsoft, 2006] Unattributed, “Dynamic Link Libraries”, http://msdn.microsoft.com/library/

[Misiti et al, 2000] M. Misiti, “Wavelet Toolbox”, http://www.mathworks.com/access/helpdesk/help/helpdesk.html

[Moorer, 1978] J. Moorer, “The Use of the Phase Vocoder in Computer Music Applications”, *Journal of the Audio Engineering Society*, Vol. 26, pp. 42-45.

[Moorer, 1983] J. Moorer, “The Manifold Joys of Conformal Mapping: Applications to Digital Filtering in the Studio”, *Journal of the Audio Engineering Society*, Vol. 31, pp. 826-841.

[Moore, 1997] B. Moore, “An Introduction to the Psychology of Hearing”, Academic Press, London.

[Moore, 1965] G. Moore, “Cramming More Components onto Integrated Circuits”, *Electronics*, Vol.38, pp.114-117.

[Murphy, 2000] D. Murphy, “Digital Waveguide Mesh Topologies in Room Acoustics Modelling”, DPhil Thesis, University of York, UK.

[Myatt, 2005] A. Myatt (coordinator), “Digital Music Research UK Roadmap”, http://music.york.ac.uk/dmrn/roadmap/

[Nattiez trans. Abbate, 1990] J. Nattiez (translated C. Abbate), “Music and Discourse: Towards a Semiology of Music”, Princeton University Press, Princeton.

[Nuttall, 1981] A. Nuttall, “Some Windows with Very Good Sidelobe Behavior”, *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 29, pp. 84-91.

[Orfanidis, 1997] S. Orfanidis, “Digital Parametric Equalizer Design with Prescribed Nyquist-Frequency Gain”, *Journal of the Audio Engineering Society*, vol. 45, pp. 444-455.

[Orfanidis, 2005] S. Orfanidis, “High-Order Digital Parametric Equalizer Design”, *Journal of the Audio Engineering Society*, vol. 53, pp. 1026-1046.

[Oohashi et al, 1991] T. Oohashi et al, “High-Frequency Sound Above the Audible Range Affects Brain Electric Activity and Sound Perception”, *Presented at the 91st Audio Engineering Society Convention*, Preprint 3207.

[Palumbi and Seno, 1998] M. Palumbi and L. Seno, “Metal String”, *Proceedings of the 1999 International Computer Music Conference (ICMC-1999).*

[Peeters and Rodet, 1998] G. Peeters and X. Rodet, “Signal Characterization in Terms of Sinusoidal and Non-Sinusoidal Components”, *Proceedings of the 1st International Conference on Digital Audio Effects (DAFx98).*

[Peeters and Rodet, 1999] G. Peeters and X. Rodet, “SINOLA: A New Analysis /Synthesis Method using Spectrum Peak Shape Distortion, Phase and Reassigned Spectrum”, *Proceedings of the 1999 International Computer Music Conference (ICMC-1999)*, pp. 153-156.

[Penrose, 2001] C. Penrose, “Frequency Shaping of Audio Signals”, *Proceedings of the 2001 International Computer Music Conference (ICMC-2001).*

[Pickles, 1988] J. Pickles, “An Introduction to the Physiology of Hearing”, Academic Press, London.

[Pielemeier et al, 1996] W. Pielemeier et al, “Time-Frequency Analysis of Musical Signals”, *Proceedings of the IEEE*, vol. 84, pp.1216.

[Plack, 2005] C. Plack, “The Sense of Hearing”, Lawrence Erlbaum Associates, New York.

[Press et al, 1992] W. Press, “Numerical Recipes in C”, Cambridge University Press, Cambridge.

[Puckette, 1995] M. Puckette, “Phase-Locked Vocoder”, *Proceedings of the 1995 IEEE Conference on Applications of Signal Processing to Audio and Acoustics*.

[Qian, 2002] S. Qian, “Time-Frequency and Wavelet Transforms”, Prentice Hall, New Jersey.

[Risset and Wessel, 1998] J. Risset and D. Wessel, "Exploration of Timbre by Analysis and Resynthesis", chapter in D. Deutsch (ed.), *The Psychology of Music*, Academic Press, London.

[Roads, 1996] C. Roads, "The Computer Music Tutorial", The MIT Press, Cambridge.

[Robinson, 1982] E. Robinson, "A Historical Perspective of Spectrum Estimation", *Proceedings of the IEEE*, vol. 70, pp. 885-907.

[Rorabaugh, 1999] C. Rorabaugh, "DSP Primer", McGraw Hill, New York.

[Rossing, 1990] T. Rossing, "The Science of Sound", Addison-Wesley, Boston.

[Roth et al, 2001] D. Roth et al, "Auditory Backward Masking and the Effect of Training in Normal Hearing Adults", *Journal of Basic and Clinical Physiology and Pharmacology*, vol.12, pp. 145-159.

[Robert Stuart, 2004] J. Robert Stuart, "Coding for High-Resolution Audio Systems", *Journal of the Audio Engineering Society*, vol. 52.

[Selesnick, 2001] I. Selesnick, "Hilbert Transform Pairs of Wavelet Bases", *IEEE Signal Processing Letters*, vol. 8, pp. 170-173.

[Serra, 1989] X. Serra, "A System for Sound Analysis/Transformation/Synthesis Based on a Deterministic plus Stochastic Decomposition", PhD Dissertation, Stanford University, USA.

[Serra, 1997] X. Serra, "Musical Sound Modeling with Sinusoids Plus Noise", *Musical Signal Processing*, Swets and Zeitlinger, Lisse.

[Shensa, 1992] M. Shensa, "The Discrete Wavelet Transform: Wedding the À Trous and Mallat Algorithms", *IEEE Transactions on Signal Processing*, vol. 40, pp. 2464-2482.

[Smalley, 1994] D. Smalley, "Defining timbre – Refining Timbre", *Contemporary Music Review*, vol. 10, pp. 35-48.

[Smith, 1991] J. Smith, "Viewpoints on the History of Digital Synthesis", *Proceedings of the 1991 International Computer Music Conference (ICMC-91)*, pp.1-10.

[Smith, 1996] J. Smith, “Physical Modeling Synthesis Update”, *Computer Music Journal*, vol. 20, No.2, pp. 44-56.

[Smith, 2005] J. Smith, “The Digital Audio Resampling Homepage”, http://www-ccrma.stanford.edu/~jos/resample/

[Spanias and Loizou, 1992] A. Spanias and P. Loizou, “Mixed Fourier/Walsh Transform Scheme for Speech Coding at 4.0 kbit/s”, *IEE Proceedings*, vol. 139, pp. 473-481.

[Steinberg, 1999] C. Steinberg, “Steinberg Virtual Studio Technology Plug-In Specification 2.0 Software Development Kit”, http://www.steinberg.de/Steinberg/Developers8b99.html

[Strang and Nguyen, 1996] G. Strang and T. Nguyen, “Wavelets and Filter Banks”, Wellesley-Cambridge Press, Wellesley.

[Stroustrup, 1997] B. Stroustrup, “The C++ Programming Language”, Pearson Educational, Indianapolis.

[Taddei, cited in Nowotny], F. Taddei cited in H. Nowotny (no date given), “The Potential of Transdisciplinarity”, http://www.interdisciplines.org/interdisciplinarity/papers/5/version/original

[Templaars, 1996] S. Templaars, “Signal Processing Speech and Music”, Swets and Zeitlinger, Lisse.

[Teolis, 1998] A. Teolis, “Computational Signal processing with Wavelets”, Birkhäuser, Boston.

[Tolonen et al, 1998] T. Tolonen, “Evaluation of Modern Synthesis Methods”, Report 48, Laboratory of Acoustics and Audio Signal Processing, Helsinki University of Technology, http://www.acoustics.hut.fi/publications/reports/sound_synth_report.pdf

[Unser, 1999] M. Unser, “Splines, A Perfect Fit for Signal and Image Processing”, *IEEE Signal Processing Magazine*, November, pp. 22-38.

[Unser et al, 1992] M. Unser et al, “On the Asymptotic Convergence of B-Spline Wavelets to Gabor Functions”, *IEEE Transactions on Information Theory*, vol. 38, pp. 864-872.

[Unser et al, 1993] M. Unser et al, “A Family of Polynomial Spline Wavelet Transforms”, *Signal Processing*, vol. 30, pp. 141-162.

[Vail, 2000] M. Vail, “Vintage Synthesizers, Pioneering Designers, Groundbreaking Instruments, Collecting Tips, Mutants of Technology”, Miller Freeman, San Francisco.

[Verma and Meng, 2000], “Extending Spectral Modeling Synthesis with Transient Modeling Synthesis”, *Computer Music Journal*, vol. 24, pp. 47-59.

[Watkinson, 1994] J. Watkinson, “The Art of Digital Audio”, Focal Press, Oxford.

[Watkinson, 1999] J. Watkinson, “MPEG-2”, Focal Press, Oxford.

[Weisstein, 2006] E. Weisstein, “The Fourier Transform”, *Mathworld*, http://mathworld.wolfram.com/FourierTransform.html

[Weisstein, 2006b] E. Weisstein, “Hadamard Matrix”, *Mathworld*, http://mathworld.wolfram.com/HadamardMatrix.html

[Wells and Murphy, 2002] J. Wells and D. Murphy, “Real-Time Partial Tracking in an Augmented Additive Synthesis System”, *Proceedings of the 5th International Conference on Digital Audio Effects (DAFx02)*, pp. 93-96.

[Wells & Murphy, 2003] J. Wells and D. Murphy, “Real-Time Spectral Expansion for Creative and Remedial Sound Transformation”, *Proceedings of the 6th International Conference on Digital Audio Effects (DAFx03)*, pp. 61-64.

[Wessel and Wright, 2002] D. Wessel and M. Wright, “Problems and Prospects for Intimate Musical Control of Computers”, *Computer Music Journal*, vol. 26, pp. 11-22.

[White, 1986] S. White, “Design of A Digital Biquadratic Peaking or Notch Filter for Digital Audio Equalization”, *Journal of the Audio Engineering Society*, vol. 34, pp. 479-483.

[Wickerhauser, 1994] M. Wickerhauser, “Adapted Wavelet Analysis from Theory to Software”, A K Peters, Wellesley.

[Williston, 2000] J. Williston, “Thaddeus Cahill’s Telharmonium”, http://www.synthemuseum.com/magazine/0102jw.html

[Wishart, 1988] T. Wishart, “The Composition of *Vox-5*”, *Computer Music Journal*, vol. 12, No.4, pp. 21-27.

[Wolfe and Godsill, 2003] P. Wolfe and S. Godsill, “Audio Signal Processing Using Complex Wavelets”, *Proceedings of the Cambridge Music Processing Colloquium 2003,* pp. 71-74.