Towards Real-Time Non-Stationary Sinusoidal Modelling of Kick and Bass Sounds for Audio Analysis and Modification

John Stuart Murray

PhD

University of York

Physics, Engineering and Technology

September 2022

Abstract

Sinusoidal Modelling is a powerful and flexible parametric method for analysing and processing audio signals. These signals have an underlying structure that modern spectral models aim to exploit by separating the signal into sinusoidal, transient, and noise components. Each of these can then be modelled in a manner most appropriate to that component's inherent structure. The accuracy of the estimated parameters is directly related to the quality of the model's representation of the signal, and the assumptions made about its underlying structure. For sinusoidal models, these assumptions generally affect the non-stationary estimates related to amplitude and frequency modulations, and the type of amplitude change curve. This is especially true when using a single analysis frame in a non-overlapping framework, where biased estimates can result in discontinuities at frame boundaries. It is therefore desirable for such a model to distinguish between the shape of different amplitude changes and adapt the estimation of this accordingly.

Intra-frame amplitude change can be interpreted as a change in the windowing function applied to a stationary sinusoid, which can be estimated from the derivative of the phase with respect to frequency at magnitude peaks in the DFT spectrum. A method for measuring monotonic linear amplitude change from single-frame estimates using the first-order derivative of the phase with respect to frequency (approximated by the first-order difference) is presented, along with a method of distinguishing between linear and exponential amplitude change. An adaption of the popular matching pursuit algorithm for refining model parameters in a segmented framework has been investigated using a dictionary comprised of sinusoids with parameters varying slightly from model estimates, based on Modelled Pursuit (MoP).

Modelling of the residual signal using a segmented undecimated Wavelet Transform (segUWT) is presented. A generalisation for both the forward and inverse transforms, for delay compensations and overlap extensions for different lengths of Wavelets and the number of decomposition levels in an Overlap Save (OLS) implementation for dealing with convolution block-based artefacts is presented. This shift invariant implementation of the DWT is a popular tool for de-noising and shows promising results for the separation of transients from noise.

Contents

List of Tables

List of Figures

Acknowledgements

There are many people who have played a major role in the completion and undertaking of this thesis, many too numerous to mention, but thanks to everyone who has positively impacted me over this time, as well as everyone within the field who has spared their precious time to discuss matters pertaining to the content within, you hopefully know who you are. Most notably, I would like to thank Dr Jeremy Wells for his inspiration, guidance, encouragement, time and understanding during my part-time undertaking of my PhD at York. Jez's research and thesis on real-time spectral modelling was the inspiration behind this undertaking. He initially started as my supervisor at the Electrical Engineering department, we then moved over to the Music department before I moved back to Electrical Engineering after his retirement. I would not advise a part-time PhD to anyone after undertaking one myself. A PhD is not made easier when trying to conduct ones research outside of work hours. The process was not made any easier by starting out in England, moving to Belgium and finishing in South Africa. There were many unfortunate leave of absences taken due to a devastating loss in the family, followed by marriage and the birth of my first son in Belgium. Moving back to South Africa with the intention of working on the PhD full-time was further hampered by my employer not being able to pay me the money owed which would have made that possible, and by the birth of my second son two months before a lengthy Covid lockdown being the final hurdle. But even with death, births and moves between countries and continents, Jez stuck by me and for that I am eternally grateful.

Arriving in South Africa without any funds after just purchasing a small farm with all of our savings, and needing to find immediate work was not easy but a sincere debt of gratitude is owed to Michael Kelly at Xperi as well as Douglas Castro, Francisco Cresp and everyone else at NeuralDSP for making the completion of this thesis possible.

I would also like to thank my initial thesis advisor Dr John Szymanski, who also retired over the course of my studies, for his encouragement and guidance. A huge thank you is also owed to Frank Stevens and Helena Daffern for taking over as my supervisors after Jez's retirement and my move back to the Electrical Engineering department. I am hugely indebted to Frank in particular for his role in taking over my official supervision. Thank you for all your time, encouragement, and detailed feedback.

Last but not least I would like to thank my family, especially Tine for her love and support. My sons Caz and Milan for their endless high priority non-maskable interrupts and inspiration. One can never tire of your smiles and the sound of your laughter. Isimo, you are sorely missed, your lust for life will continue to inspire me, you are forever close to my heart, thank you for everything.

Declaration of Authorship

I declare that this thesis is a presentation of original work and I am the sole author. This work has not previously been presented for an award at this, or any other, University. All sources are acknowledged as references.

Chapter 1

Overview of the Thesis

They use a language which you see.
It is made out of sound, it is sound, but you see it.

Terence McKenna [1]

1.1 Introduction

This chapter gives an overview of the research goals and activities of this thesis. The motivations with an intended practical application are explained and put in context. The aims and objectives for the work are described, some of these aims and the research activities involved inevitably shifted and changed over the duration of working on the thesis. However, the purpose and significance have remained the same. The motivations, thesis contributions and an overview of the structure of the thesis are described in the following Sections.

1.2 Motivations

Dancing and music are behavioral foundations which evolved from primate and shemanic rituals where synchronous group vocalizations provided "an expressive system that communicates emotions and enhances group integration". [2]

Humans have a genetic disposition to engage in collective musical behaviours where expressive activities such as dancing and drumming serve a variety of purposes in group integration. Comparison of chimanzee ritualizations indicates that this behaviour represents a pre-adaptation for shamanism and reflects a "basic neuropsychological structures and social psychological functions of hominids" [2]. Outdoor electronic dance music festivals can be likened to modern shamanic rituals where the effect of music and dancing forms a function for social bonding and emotional communication as well as having an effect on altering states of consciousness through the positive effect this activity has on emotions and personal healing.

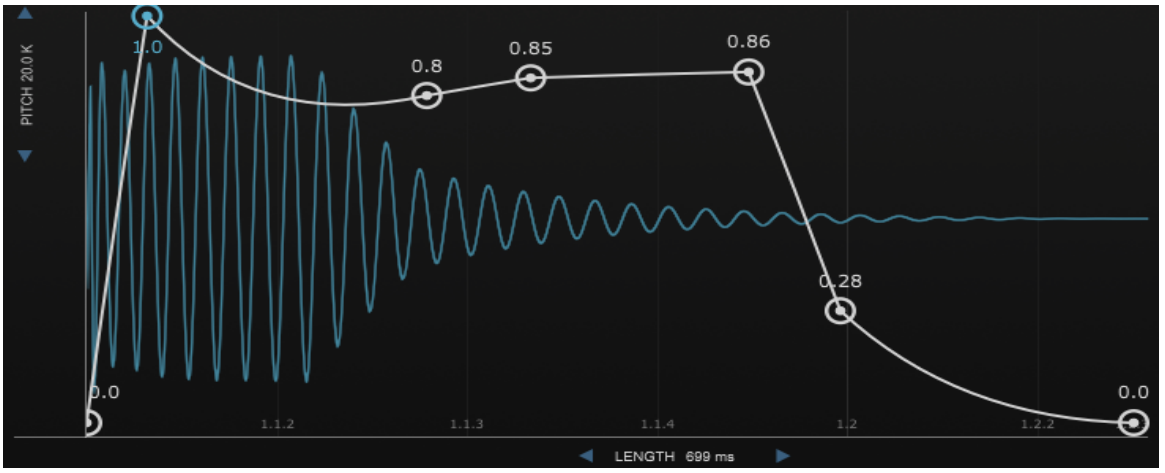
Electronic Dance Music (EDM) is a broad term for multiple genres of music with an emphasis on dancing. These genres have evolved since the 1960's through the influence and technological advancements in music technology and the development of synthesizers, digital samplers, rhythm machines and Digital Audio Workstations (DAWs) [3]. Different electronic dance music genres will have varying structures and flow, there are variations of tempo, melody and percussive elements within each style and sub-genre, but the fundamental element in all electronic dance music which is conventionally used to convey a rhythmic foundation is: the 'Kick and Bass'[4–6]. Recent neurobiological research on bass sounds and their stimulation shows that brain activity becomes synchronised with the frequency of beats, and that lower frequencies are more successful at synchronising large pools of neurons in the brain. The kick and bass provide the low frequency content ('low end') in music which causes our brains to synchronise with the rhythm of a song, and this in turn creates a motorical response "to drive people to move to the beat" [6].

The combination of the kick drum and bass is the foundation of any dance song's rhythm and harmony. How these two elements are combined is incredibly important for creating this foundation. Careful attention to the relationship between these two components and the effect they have on one another in the frequency domain is crucial for creating a professional sounding mix. Achieving this balance as well as the balance of these elements and other melodic lead and percussive sounds in the overall mix can be very challenging. A golden rule in music production is that the kick and bass must not conflict with one another. Overlapping frequencies, phase issues and the duration of bass sounds overlapping with the kick drum are some of the potential problems a producer needs to pay particular attention to in order to ensure a clean well produced kick and bass. The ability to shape the kick and bass so they co-exist and work together is a skill which can take years to master using numerous techniques for creating low-end separation such as equalisation, dynamic equalisation, multi-band compression, side chain compression, stereo imaging, transient designing, balancing and arrangement.

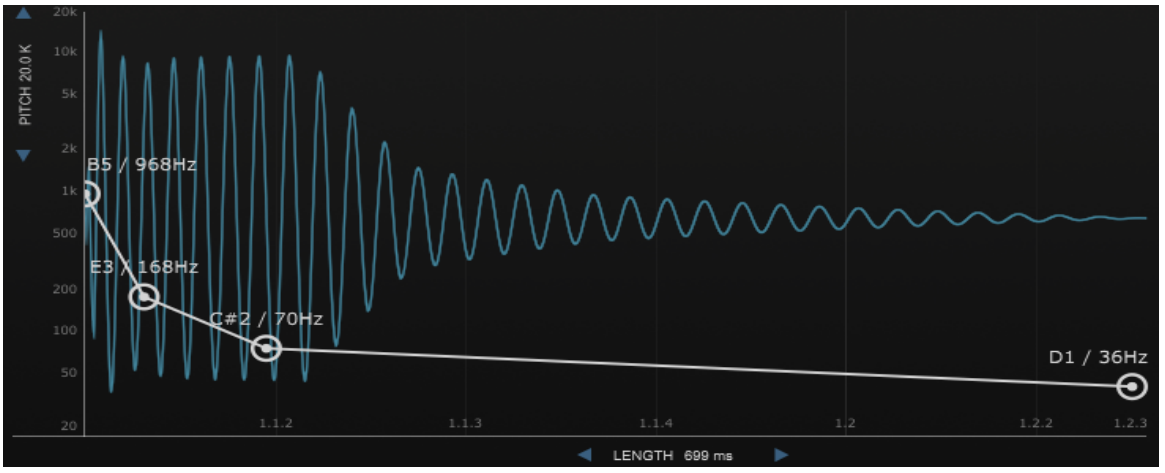
> The importance of bass in modern music can not be overstated. Low-end frequency content (i.e. basslines, kick drums and the like) is essential to any genre that falls under the umbrella of club music. The push and pull of a track's low-end elements are what can drive a song's momentum, and, when utilized correctly, a track's bass should not just be heard, but also felt. In turn, the absence of this low-range physicality can work against you: a techno track without a pulsating kick drum is hardly a techno track at all; a hip hop beat without a substantial low-end thud can sound frail and lifeless; and a drum and bass track without the bass is, well, only half of what it could be.
> Low-end frequency content is not something that is easily controlled or manipulated. Especially for beginning producers, arriving at a satisfactory bass sound during the course of a mix can be a painstaking process informed by many stages of trial and error. How does one create bass weight without overpowering the rest of the mix? How does one craft warm, full-bodied bass lines without muddying up the sonic spectrum? The truth is, there are no universally accepted answers to these questions. Rather, over time each producer is left to formulate their own approaches and develop their own techniques for effectively utilizing bass and low-end in their work.
>
> ---
>
> *Glenn Jackson [7]*

Given the importance of the 'Kick and Bass' in dance music and how challenging it can be to produce, the modelling of already well produced kick and bass lines with the ability to manipulate and resynthesise these components from model parameters, while maintaining the quality, is the main motivation behind the work conducted in this thesis.

Sinusoidal Modelling offers a powerful and flexible parametric method for analysing and processing kick and bass sounds. These models separate the signal into the three most general representations of sinusoids, transients and noise so that each component can be modelled separately incorporating the individual properties of each of these components into the overall model. Each of these can then be modelled in a manner most appropriate to that component's inherent structure. The accuracy of the estimated parameters is directly related to the quality of the model's representation of the analysed signal, and the assumptions made about a signal's underlying structure. For sinusoidal models, these assumptions generally affect the non-stationary estimates related to amplitude and frequency modulations. These assumptions work best when they fit the signals underlying structure, but are not always correct as the manner in which audio amplitude changes, can take on numerous shapes and forms.



(A) Amplitude Envelope



(B) Pitch Envelope

FIGURE 1.1: Kick drum synthesis shaped by (A) Amplitude and (B) Pitch Envelopes

Figure 1.1 shows a kick drum synthesised using the Kick2 plugin [8]. Figure 1.1a displays the amplitude envelope used to shape the initial attack (transient) and decay of the synthesised sound. The Attack, Decay, Sustain and Release (ADSR) parts of an amplitude envelope which are used to shape the volume of a sound over time are described in Section 2.1.5. The pitch envelope which controls the frequency over time is displayed in Figure 1.1b. It is not always the case, but a common technique for synthesizing a kick drum is with a single sinusoid with a fast decaying pitch over the attack portion of the sound. Other examples of amplitude and pitch envelopes applied to other Kick, Bass and Snares, synthesised using the Kick2 VST plugin are shown in the Appendix A. There are many forms of kick and bass synthesizers available, however in this thesis the use of the Kick2 VST plugin is mostly used because of the user interface's feature of displaying the sound and the effect changes to parameters have on the resulting waveform.

Synthesised sounds, electronic music and audio created or edited on hardware or software can have the amplitude of the signal shaped in numerous ways. The user can select to fade a sound in or out with a variety of amplitude curves, the two most common being linear or exponential. It is therefore desirable for a model to distinguish between the shape of different amplitude changes and adapt the estimation of this accordingly.

1.3 Aims and objectives

This thesis sets out to analyse and model kick and bass lines from electronic dance songs. Electronic music producers generally start working on the kick and bass-line of a song first, often using a kick and bass line from another song as a reference. Being able to model, manipulate and re-synthesise different kick and bass lines from referenced songs can be of interest to an electronic music producer. Synthesised kick and bass lines can be shaped and sculpted in many ways. They are characterised by the way the frequencies and amplitude evolve over time and so it is of interest to model these non-stationary characteristics.

The two most common curve shapes used to adjust and shape the amplitude of kick and bass sounds are linear and exponential curves, and so distinguishing between these two types of curves and incorporating that into a sinusoidal model allows for a more flexible and accurate model.

Spectral Modelling Synthesis, the Phase Vocoder and common implementations of sinusoidal models are presented in Chapter 2, including the concept of Windowing and the effect this has on Transients and on the frequency spectrum. In general spectral models incorporate windowing and are therefore required to implement an 'Overlap Add' (OLA) framework, where frames of windowed audio data are overlapped and added to ensure unity gain. The overlapping of analysis frames, and the effect this has to a model are presented in 2.4.5. This process can improve the accuracy of a model but at a greater computational expense, determined by the amount of overlap and the resulting number of audio samples overlapping in successive frames. This thesis is inspired by previous work on single-frame spectral analysis which examined the possibilities of real-time sinusoidal modelling and estimation of exponential amplitude and linear frequency modulations from phase distortion [9, 10]. A non-overlapping single frame analysis framework requires estimating sinusoidal components and the non-stationarities to a high degree of accuracy to avoid discontinuities at frame boundaries.

This thesis aims to implement a sinusoidal model using a non-overlapping analysis/synthesis framework. Estimation of amplitude and frequency modulations are taken from single frame analysis methods which do not require any information from previous frames for calculation of monotonic amplitude and frequency modulation within a frame.

A non-overlapping single frame based method which assumes exponential amplitude change can not accurately approximate linear modulations within a frame. Biases in amplitude modulation can result in discontinuities at frame boundaries. Therefore, it is highly desirable for such a model to be able to distinguish between linear and exponential amplitude change within a single analysis frame. An initial aim and objective of this thesis therefore sets to explore distinguishing between Linear and Exponential Amplitude change within a single analysis frame, from examination of the phase and magnitude information and the differences these two envelope shapes have on the Fourier data.

The research presented in this thesis, reviews a wide range of algorithms related to modelling musical signals. Sinusoidal models, atomic decomposition, adaptive models relying on iterative refinement, and wavelets which can be likened to filter banks and multiresolution analysis are some of the methods examined. Some of these methods have been adopted and then adapted in novel ways with the intent of improving the accuracy of modelling kick and bass sounds using a sinusoidal model in conjunction with some of these adapted methods. Matching Pursuit (MP) [11] is an iterative approximation algorithm that decomposes any signal into a linear expansion of waveforms from a "redundant dictionary of functions". Guided Matching Pursuit (GMP) [12, 13] and Modelled Pursuit (MoP) [14] are methods adapted from MP for specific purposes. GMP for the use in source seperation, while MoP has been used for modal decomposition of room impulse responses (IRs) by examining the results returned from a single DFT of the entire IR.

The examination of the use of a dictionary comprised of sinusoids with slightly varying parameters from model estimates is evaluated in a segmented audio framework with an initial objective of running in real-time. A similar method to MoP has been adapted for use in a segmented framework using the non-causal methods derived of describing and discrimination between both exponential and linear monotonic intra-frame amplitude change. Parallelization of spectral modelling using MoP in real-time was initially investigated given recent work on synthesising thousands of sinusoids on a Graphics Processing Unit (GPU) [15], and other GPU audio advances. However, the focus of this work shifted due to a hypothesis of how best to extract and model transient components which are quite prominent at the start of kick and bass sounds. Parallelization of the system as a whole is important for achieving real-time performance, but is left as a future research directive.

Modeling monotonic amplitude change from non-causal measurements requires zero-phase windowing to remove unwanted shifting distortions, which results in a flat phase response across the main lobe of a stationary sinusoidal peak. The windowing process can unfortunately negatively impact transient details due to any smearing of these components in time from the windowing process. However, it is possible to extract sinusoidal components with monotonic amplitude change from single frame analysis. The remaining signal after the subtraction of all decomposed monotonic quasi-stationary sinusoidal components, contains the remaining short time components such as transients; higher order amplitude modulated components, and noise. These signals appear as broadband components in the frequency spectrum which are not well modelled by quasi-stationary sinusoidal components. The alternative hypothesis presented conversely investigates modelling these broadband components; transients and quasi-stationary sinusoids together, as an overcomplete decomposition of quasi-stationary sinsoidal components. This approach uses rectangular windowing to avoid any smearing of transient data.

The Undecimated Wavelet Transform (UWT), also known as the Stationary Wavelet Transform (SWT), 'algorithme à trous', and other names; Matlab [16] refers to it as the 'Nondecimated Discrete Stationary Wavelet Transforms (SWTs)' [17]. This transform has the desirable property of shift invariance [18]. For this reason the method is a popular choice in signal de-noising [19], as well as detecting discontinuities which is interesting with regards to transient detection [20]. Implementing the forward and inverse transforms in a segmented audio framework suffers from block end artifacts due to the nature of the implementation via convolution. Overcoming the block end artifacts and implementing this in a segmented audio framework for the use of analysing the signal residual after sinusoidal modelling has therefore been investigated. A generalisation of calculations for the different amounts of delay compensations and differing amount of overlap extensions for different lengths of filters at each level of the decomposition in an Overlap Save (OLS) implementation for dealing with convolution block-based artefacts is given, for both the forward and inverse transforms.

The use of the segmented Undecimated Wavelet Transform (SegUWT) is then explored for the separation of transient components from the residual signal by de-noising.

1.4 Thesis Contributions

This thesis contributes the following:

- Non-causal estimates of linear intra-frame amplitude change have been derived from the analytical derivation of the effect linear AM has on a Hanning widow applied to a stationary sinusoid and the effect this has on the first order difference to the phase information.
- A method for discriminating between linear and exponential amplitude change is derived by examining the effect this has to the magnitude spectrum when amplitude changes of equal energy displacement between the two curve types are applied.
- Non-causal estimates of exponential and linear amplitude change with amplitude type discrimination have been applied in an adapted version of MoP using a Hanning window and zero-phase padding for modelling both linear and exponential monotonic amplitude change, whilst leaving transients and other broadband components in the residual signal for further analysis and possible extraction using the SegUWT.
- MoP is shown to improve a model's accuracy by increasing the dictionary to contain a number of quasi-stationary sinusoidal atoms with a slightly random normal distribution of parameters with varying values from those of the initial atom estimated directly from the DFT.
- Modelling of short transient signals such as percussive instruments is also investigated using an over-complete decomposition from a single analysis frame.
- Residual modelling using the segmented Undecimated Wavelet Transform (SegUWT) is presented. A general form for calculating the delays introduced at each level due to the oversampling of filter coefficients at each decomposition depth is given. Generalised rules for dealing with block-end effects at frame boundaries are derived for dealing with different filter lengths and decomposition depths for both forward and inverse transforms.
- The use of de-noising to separate transient components remaining in residual signal from noise is explored. This requires further investigation on how best to achieve the optimal level of separation through the combination of wavelet filters, the filter order, decomposition depth, and is also left as a future research direction.

1.5 Thesis Organisation

Chapter 2 introduces the reader to some of the topics and methods used, including key concepts mentioned within the thesis for reference. Sinusoidal Modelling and different approaches to modelling non-stationary sinusoids, along with multi-resolution approaches and Wavelets are discussed.

Chapter 3 describes analytically a method for distinguishing between linear and exponential amplitude change, as well as deriving a method for providing accurate estimates of monotonic linear amplitude change from Fourier data.

Chapter 4 introduces MP and MoP which is then extended and used in a non-causal system using a Hanning window and zero-phase padding for improving parameter estimates. MoP is also adapted to use causal intra-frame parameter estimates using a rectangular window in a novel manner for modelling transients as a number of sinusoidal components from non-overlapping frames in an over-complete decomposition. Modelling of transient signals using an over-complete decomposition is then explored and shown to accurately model and successfully alter certain types of percussive sounds.

Chapter 5 derives a generalised method for performing the segmented Undecimated Wavelet Transform (SegUWT) in real-time while avoiding block-end artifacts for both the forward and inverse transforms. The popular application of using this shift-invariant implementation of the wavelet transform for de-noising, is then explored for separating transient components from noise in the residual signal.

Chapter 6 reviews the work presented from the previews chapters, puts this in a practical context and provides analytical results of the presented methods. Two implementations, a segmented and a single frame system, are presented incorporating the methods discussed in previous chapters.

Chapter 7 concludes the thesis and suggests possible future research directions.

Chapter 2

Signal Models and Relevant Technologies

> There is geometry in the humming of the strings, there is music in the spacing of the spheres. The stars in the heavens sing a music, if only we had ears to hear.
>
> *Pythagoras* [21, 22]

In ancient Greece, Pythagoras discovered the relationship of musical pitch to the vibration of a single stringed instrument; identifying that the pitch of a musical note is inversely proportional to the length of the string. Pythagoras investigated the ratios of the lengths of strings in relation to the pitch produced and is attributed for noticing that the pitch of a string played at half its length produces a pitch exactly twice that of the original frequency (a measure of repeated cycles over a unit of time). The doubling of frequency of a musical pitch is called an octave, and musical tones which are integer multiples of a fundamental frequency, form what is known as a harmonics.

Harmonies appear everywhere in natural phenomena and Pythagoras extended his discovery of the ratios between musical pitch from not just instruments, but to include all objects in motion including the orbit of planets. Interestingly enough, planets do orbit in harmony. "Mars takes approximately twice as long to orbit as Earth, turned into a musical chord that means Mars and Earth play an octave. Venus orbits three times faster than Mars, which means they play a fifth plus one octave." [23]

The frequency of gravitational waves which are ripples in the fabric of space itself have recently been discovered by the merging of black holes, and can indeed be listened to by transforming the gravitational waves into sound waves [24].

Mathematics and music have an undeniable relationship with one another. Rhythm is related to counting, while harmony is related to complex temporal relationship between notes. This chapter introduces the relevant concepts, theories and technologies related to the work presented in this thesis which aims to mathematically model musical signals. Examples of these musical signals are initially introduced in Section 2.1. These form the basis of the signals presented for analysis and representation and as such form the foundation from which the work in this thesis is based on. The following sections introduce the concepts and mathematics used to represent, analyse, model and re-synthesise these signals.

Section 2.3 introduces common forms of signal representations, which are expanded in the following subsection. Sinusoidal models and improving parameter estimations from information provided from the DFT are presented in Sections 2.4.2 and 2.7, before more intensive techniques of improving parameter estimations are presented in Section 2.8. Finally, Wavelets and the DWT is introduced, providing the foundation for the method of modelling the residual in Chapter 5.

2.1 Musical Signals

When investigating musical instruments, especially in the context of sinusoidal modelling, it is important to understand the properties of a wave, which can be defined as any periodic disturbance that propagates through a medium such a sound waves through air. Frequency refers to the number of recurring cycles a periodic wave from repeats itself over a unit of time, measured in Hertz (Hz). Pitch is related to frequency, but rather than describing the physical properties of a sound wave, pitch describes how high or low a sound is perceived by the listener. Pitch is perceived as fixed ratios of frequencies. An instrument producing a number of harmonically related frequencies is relatively perceived as a single pitch, typically determined by the fundamental frequency of which the harmonics are all multiples of. An octave describes the eighth notes in a musical scale, as well as the interval between one musical pitch and another [25]. Musical signals have specific acoustic properties not shared by speech or other types of audio signals. These properties may include harmony, rhythm, timbre and melody. Musical signals in general (at least those produced by physical instruments) produce a vibration which exhibits some form of a stable frequency and pitch over a period of time.

The main signals of interest in this thesis are kick and bass sounds, but stringed instruments and acoustic environments are initially introduced: Stringed instruments for their importance in understanding how sound can be produced from the vibration caused by standing waves, as well as the rate at which they decay; acoustic environments as another example of how sounds in nature can exhibit an exponentially decreasing rate in amplitude. Kick and Bass signals are then explained in more detail. The kick drum is one of the loudest sounds in an electronic dance song and together with the bass form the low end of the mix, meaning the sound sits at the lower part of the frequency spectrum.

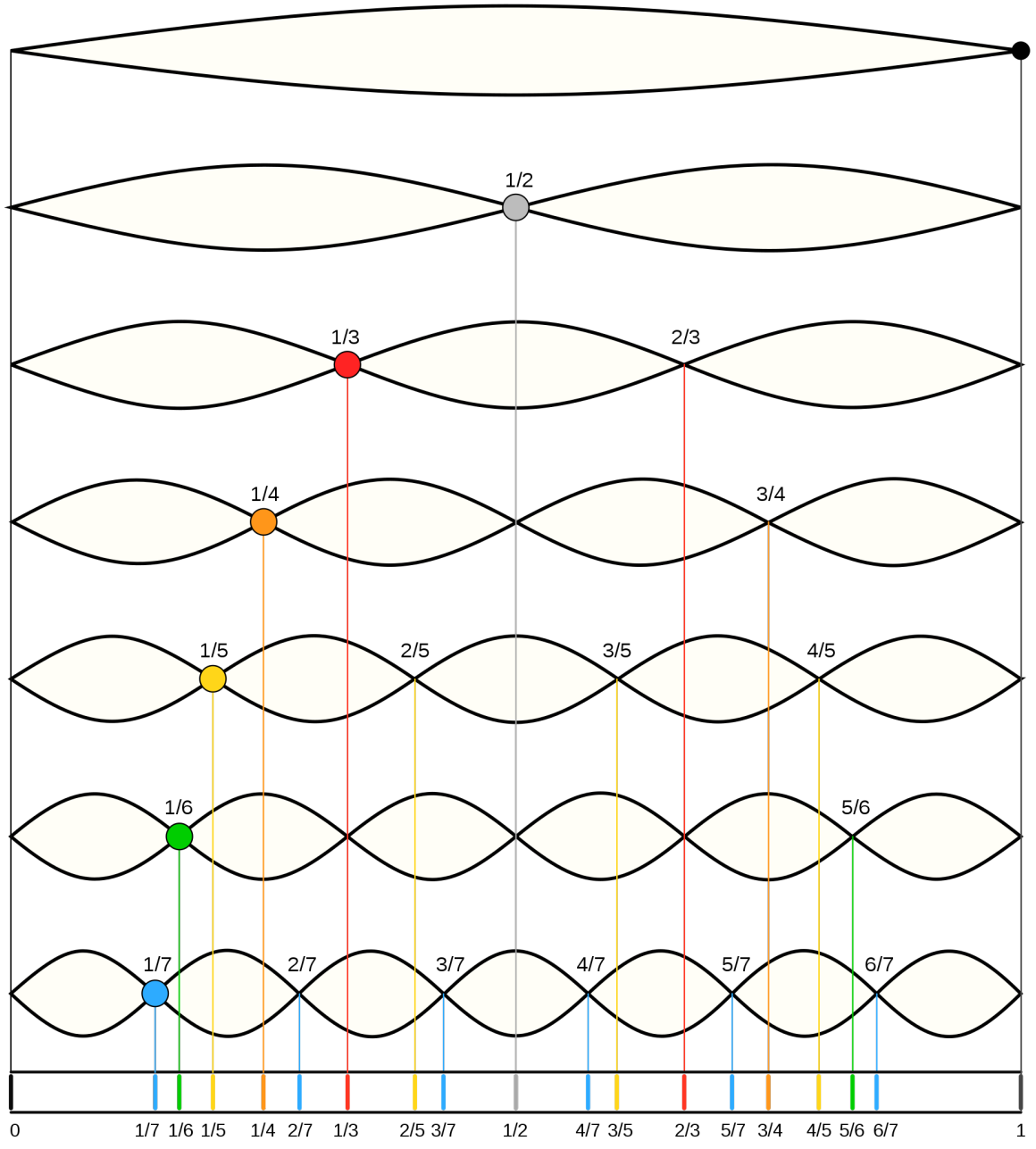


FIGURE 2.1: The nodes of a vibrating string are harmonics

Smith in [26] identifies the importance of exponentials and goes on to describe all momentary excited oscillations that are linear and time-invariant as examples of signals which decay exponentially. Musical examples of these include the vibrations of tuning forks and plucked string instruments. Another example of exponential decay can be found in how the reverberant sound of a room decreases over time.

2.1.1 Plucked String Instruments

Plucked string instruments such as the guitar or bass guitar have certain characteristics shown in Figures 5.4 and 5.5. These include a sudden excitation or burst of energy, followed by vibrations, known as standing waves, which travel along the string of the instrument, these oscillations of the air around the body of the instrument causes the sound heard. The vibration of a string with a fixed length creates a number of standing waves which are harmonically related. The length of the string controls the length of the standing waves and is responsible for the fundamental frequency of the sound produced and the associated harmonics; which are multiples of the first harmonic. The plucking of a stringed instrument converts the oscillations into a combination of sine and cosine waves which decay at exponential rates, with higher frequency components decaying faster than low frequency components.

Figure 2.1 [27] shows Harmonic waveforms, the fundamental lowest frequency waveform is at the top and the harmonics related to integer multiples of the fundamental are shown, increasing in frequency as you move down the figure. This shows the related harmonics as multiples of the first harmonic. The second wave down is half the length of the first, doubling the frequency which results in two periods of the wave in the time of a single period of the fundamental.

Plucked string instruments have been researched extensively in Physical Modelling (PM) [28–33] which is concerned with the mathematical modelling of the physical attributes of the systems which creates the sound.

2.1.1.1 Acoustic Environments and Reverberation

Exponential decay is also used in the modelling of acoustic environments. The calculations for the reverberation time of a room are based on equations “which assume a diffuse sound field and an exponential energy decay” [34, 35].

In [36] Moorer experimented with exponentially decaying noise to model late reverb reflections. An exponential decay of white noise was found to satisfy both time and frequency domain criteria for a natural sounding reverb. In practise, higher frequencies decay faster than lower frequencies but exponential decay is not only found in stringed instruments, but also in natural acoustic environments.

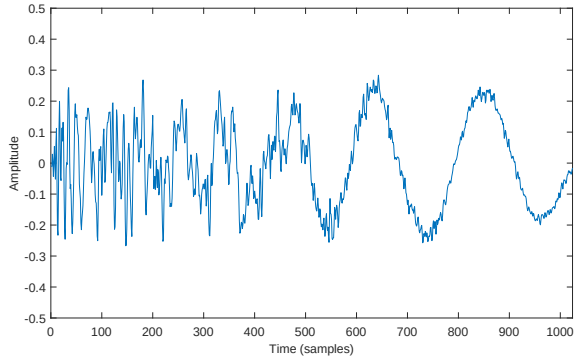
Exponential decay sounds natural due to the manner in which the human ear processes sound which is on a logarithmic scale. The exponential decay of plucked stringed instruments and the decay of impulses or transients in acoustic environments is mentioned here as the assumption of exponential amplitude change is incorporated into different sinusoidal models such as the Exponentially Dampened Sinusoidal Model (EDSM).

2.1.2 Electronic Kick Drums

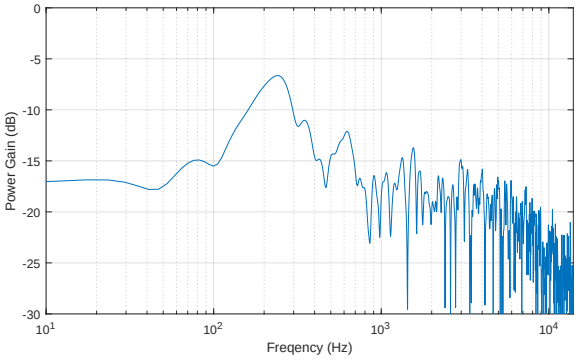
Figure 2.2 shows some examples of kick drums and their log-magnitude spectrums. Further examples of kick drums and the resulting time domain waveforms, using the Kick2 VST plugin [8], can be found in A.1.1. Examining the waveform of a kick drum shows that it is more compact at the beginning, and the waveform becomes more spaced out towards the end. This indicates that the pitch of the kick drum is dropping from the begging part, known as the attack / transient, which is rich in harmonic frequencies. The amplitude of the kick drum also drops from a very sharp attack to slower release. This fast attack part of the sound full of rich harmonics, and the drop in amplitude and drop in pitch is what gives the kick drum its sonic character. Kick drums can take on a sinusoidal shape, sometimes with added distortion as in Figure 2.2a.

A kick drum is characterised by a short burst of energy where the attack portion (known as the transient) and sustain of this burst of energy controls how the sound is perceived. Kick drums are transient instruments, meaning that when you look at the waveform of a single drum hit, there is a high amount of energy concentrated at the beginning of the waveform which quickly decays. Even though kick drums are characterised as short bursts of energy, they also include a fundamental frequency.

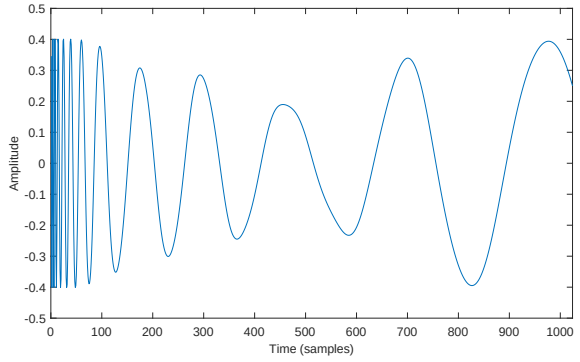
Physical kick drums have two drumheads, which doubles the vibrating mass of the instrument, and allows it to sustain longer. The kick drum is also the largest drum in a kit, meaning that it creates lower frequencies than the other drums. When the kick pedal (mallet) strikes the stretched surface of the kick drum, it creates a high-frequency sound that drops in pitch and amplitude over a short period of time.



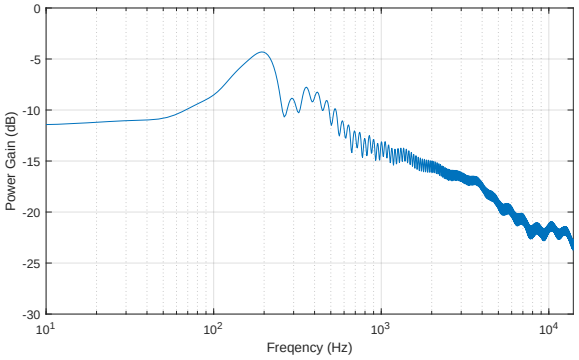
(A) Kick Drum (Key: B3) with fundamental frequency of 246 Hz (@48 kHz)



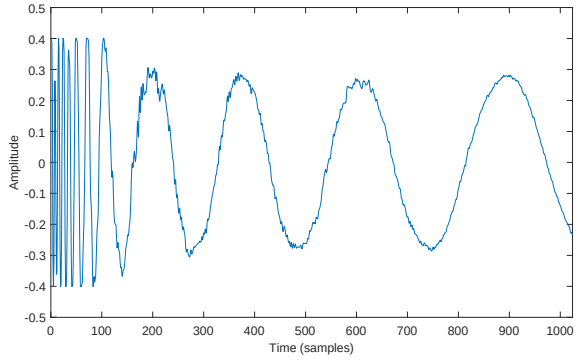
(B) Log Magnitude Response of (A)



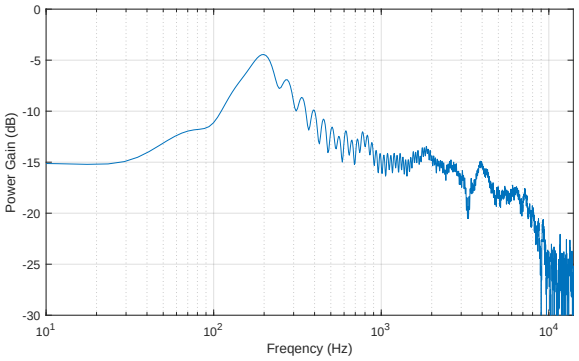
(C) Kick Drum (Key: G3) with fundamental frequency of 196 Hz (@48 kHz)



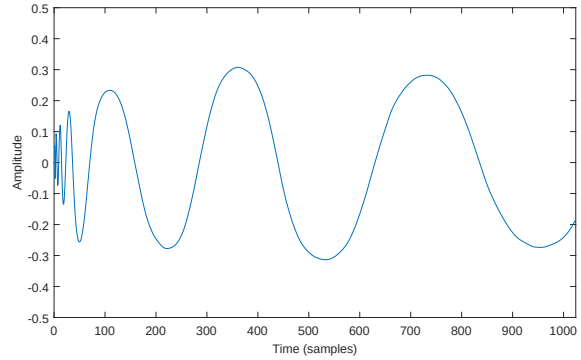
(D) Log Magnitude Response of (C)



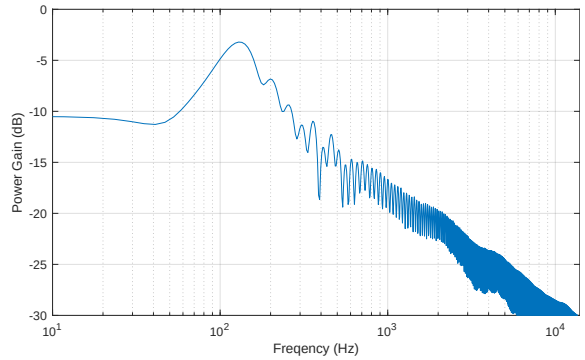
(E) Kick Drum (Key: C3) with fundamental frequency of 130 Hz (@48 kHz)



(F) Log Magnitude Response of (E)



(G) Kick Drum (Key: G3) with fundamental frequency of 196 Hz (@48 kHz)



(H) Log Magnitude Response of (G)

FIGURE 2.2: Examples of kick drums (1024 samples @48 kHz) containing a fast attack and part of the more gradual release, along with the log magnitude response.

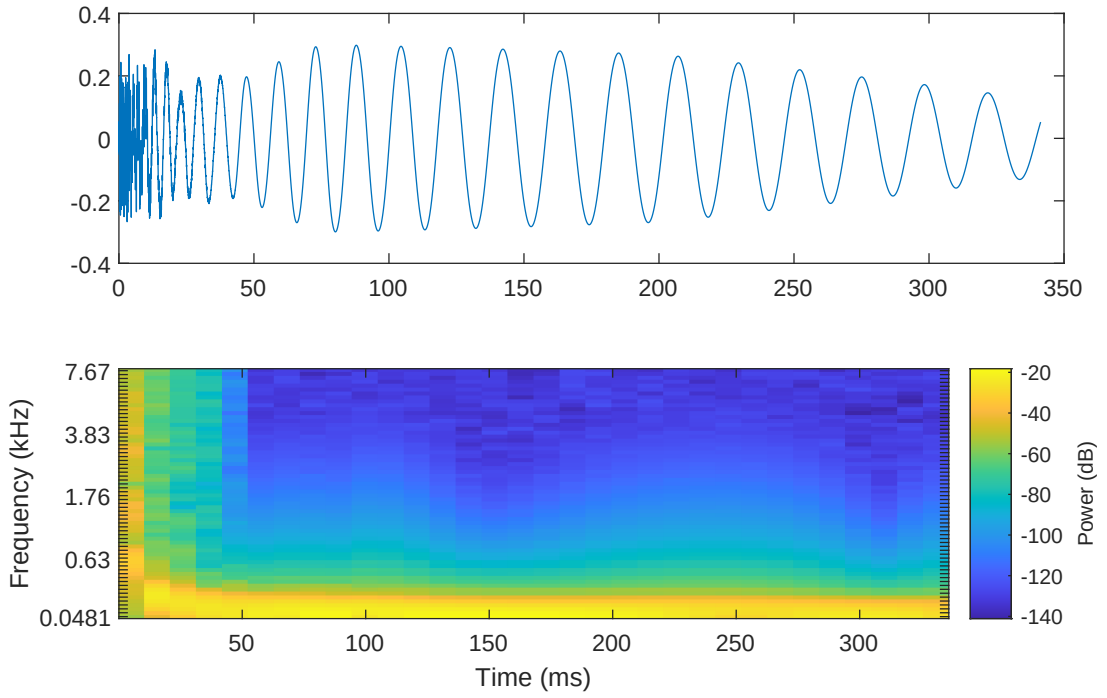


FIGURE 2.3: Spectrogram of Kick Drum in Figure 2.2a (16385 samples @48 kHz)

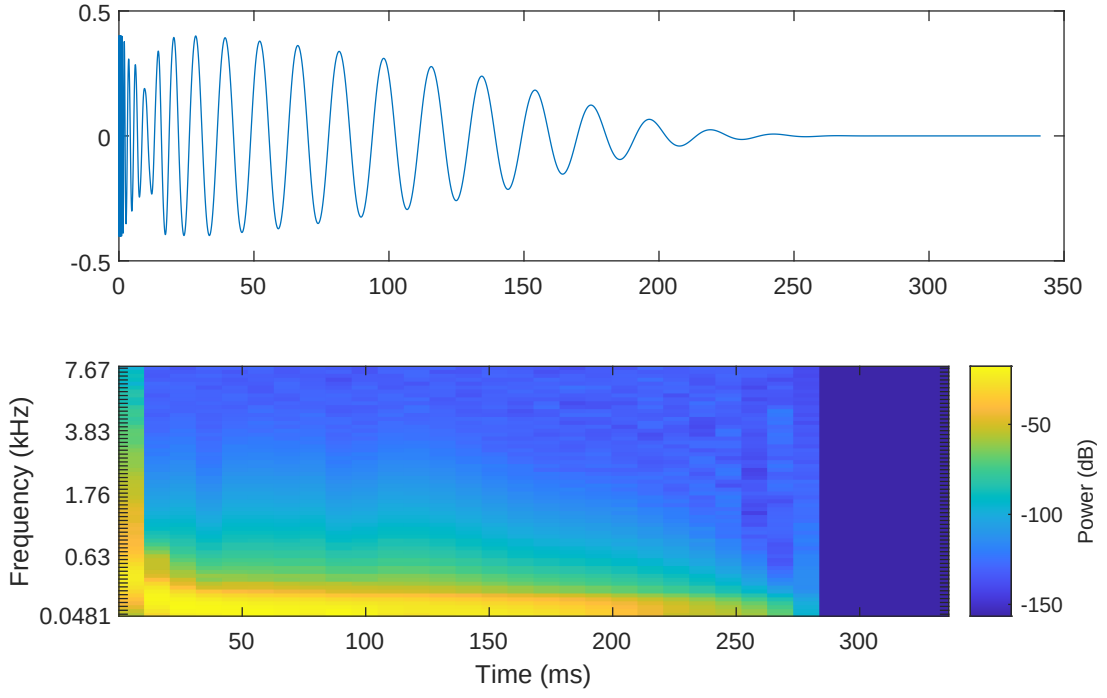


FIGURE 2.4: Spectrogram of Kick Drum in Figure 2.2c (16385 samples @48 kHz)

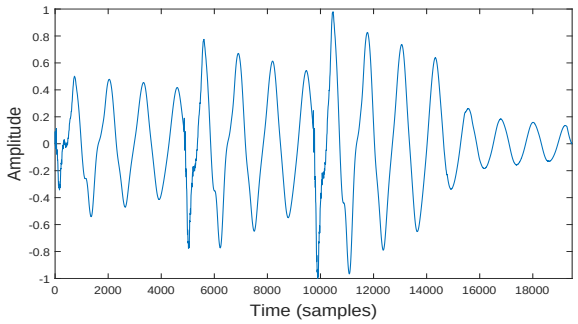
Figures 2.3 and 2.4 display the waveform in comparison to a spectrogram of the sound. Both spectrograms display a broad spectrum at the start of the sound, followed by a dominant sinusoidal frequency component over the remainder of the sound.

Adjusting the amplitude shape at the start and end of a recorded or synthesised kick drum has an crucial role in how the sound is perceived. Applying a more gradual attack envelope to a kick drum results in a longer attack, and the kick drum losing its punch as the transient part of the sound is faded in slowly, causing a loss in mid range frequencies and the sound to become more dull.

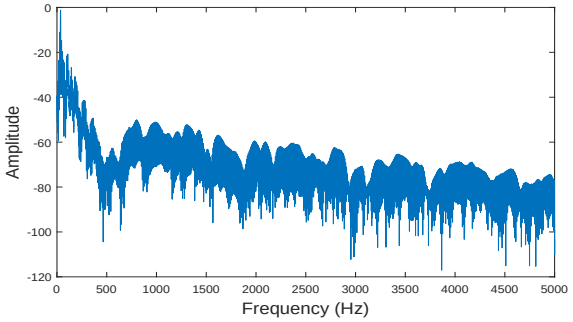
2.1.3 Bass

Bass synthesis often uses more harmonically rich waveforms in comparison to synthesizing kick drums. The bass synthesizer plugin Ultrabass [37] is shown in Figure 2.5. As can be seen from the list of waveforms available, that bass synthesis often uses more harmonically rich waveforms, and can often be created from wavetables or FM (Frequency Modulation) synthesizers.

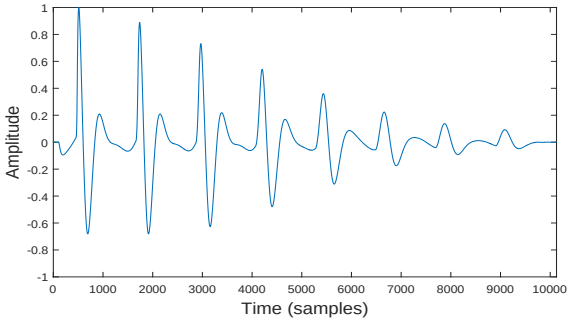
FIGURE 2.5: ultrabass VST plugin showing harmonically rich waveforms used for bass synthesis [37]



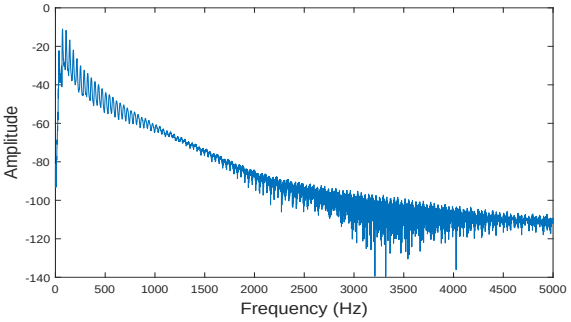
(A) Tight Bass Example



(B) Magnitude Response of Tight Bass Example from Figure 5.8a



(C) Single Bass note



(D) Magnitude Response of Single Bass note from Figure 5.6a

FIGURE 2.6: Single Bass note, and Bass line (@48 kHz) with magnitude spectrums

Some examples of Bass sounds and their magnitude spectrum are presented in Figure 2.6 and 5.8. Two examples of Bass synthesis using the Kick2 VST [8] are shown in Figures A.6 and A.7. These show a much richer magnitude spectrum in comparison to a kick drums. Sawtooth waveforms which contain both even and odd harmonics fill up the frequency spectrum with a rich complex sound that can be sculpted into a bass sound with some low pass filtering, and adjusting of the amplitude envelope. Filter envelopes are used to shape the frequency of the sound over time. While amplitude envelopes shape the overall volume of the bass over time. Square waves: presented in Figure 2.20, only produce odd harmonics can also be used for smoother bass tones. An example of a sawtooth waveform synthesised in the Key of G with 128 harmonics is shown in Figure 4.58d.

2.1.4 Kick and Bass

The Kick and Bass forms the low frequency spectrum of a song, know as the bottom end. These two components compliment each other, and should not overlap in frequencies or cause destructive interference (phasing) with one another.

"Sometimes the kick and bass play together rhythmically, while sometimes they never play at the same time. These scenarios are both fine, but what isn't okay is the low-frequency elements get in each other's way and distract from the rest of the song. No mix trick or plug-in can effectively fix this fundamental arranging mistake. The golden rule is, make sure the two complement each other in the arrangement. Acoustic drums fit naturally against bass guitar because their fundamentals and harmonics tend to complement each other. In a clean arrangement, the kick and bass naturally blend to create a full low end." [38]

2.1.5 Amplitude Envelope

The amplitude envelope is the change in a waveforms shape over time with regards to it energy. In music synthesis the amplitude envelope can be expressed by a sounds Attack, Decay, Sustain and Release (ADSR) parts as shown in Figure 2.7.

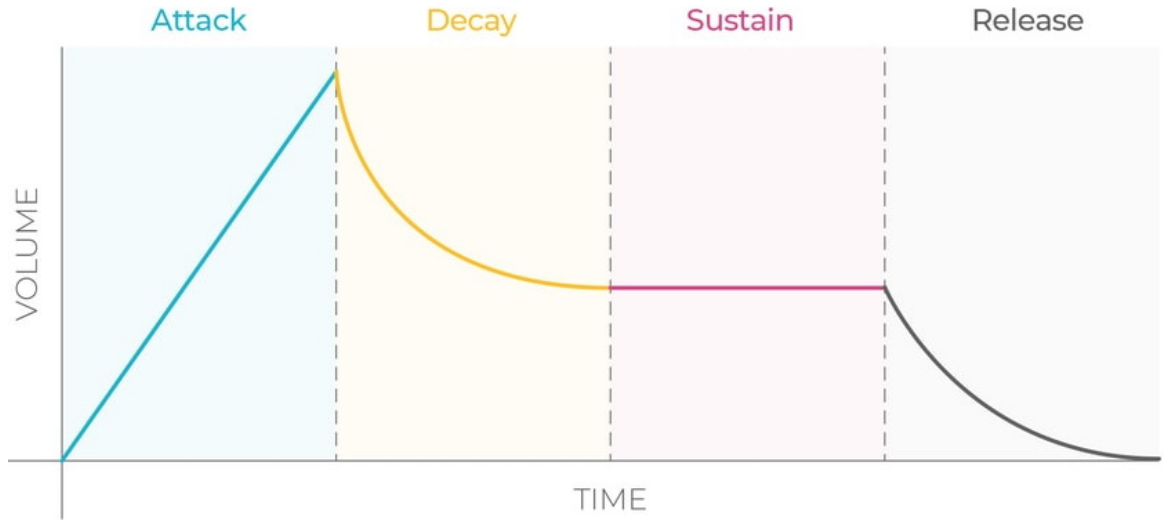


FIGURE 2.7: Attack Decay Sustain Release Envelopes [39]

The attack phase begins when a note/key is pressed, this is the start of the sound (note onset) and determines how quickly a sound reaches full volume. The decay stage determines the length it takes for the sound to decay from a peak level to the sustain level. The attack and decay stages of the sound for the transient portion of a sound. The sustain phase holds the sound at a certain level for some time duration, this is the stage of the sound which is most stationary. The release stage at the end starts as you release the note/key, and determines how quickly the sound takes to end.

2.1.5.1 Amplitude Curve Types

The amplitude envelopes of these sounds can be shaped in numerous ways. Linear, piecewise linear, and exponential being common options, but logarithmic, s-curve and other variations of these where you can adjust the slope of the curve are also available. As shown in Figure 2.7 it is quite common to have a linear attack and exponentially decaying decay and release stages, but this is not always the case. Exponential decaying sinusoids and linear interpolation of amplitude between successive frames are the two most common methods of modelling amplitude change. Figure 2.8 shows three examples of a user adjusting a fade in/out curve from logarithmic to exponential while adjusting the slopes of the curves, so a pure logarithmic or exponential slope is also not always the case in practise. An example of a linear crossfade is also presented in this Figure.

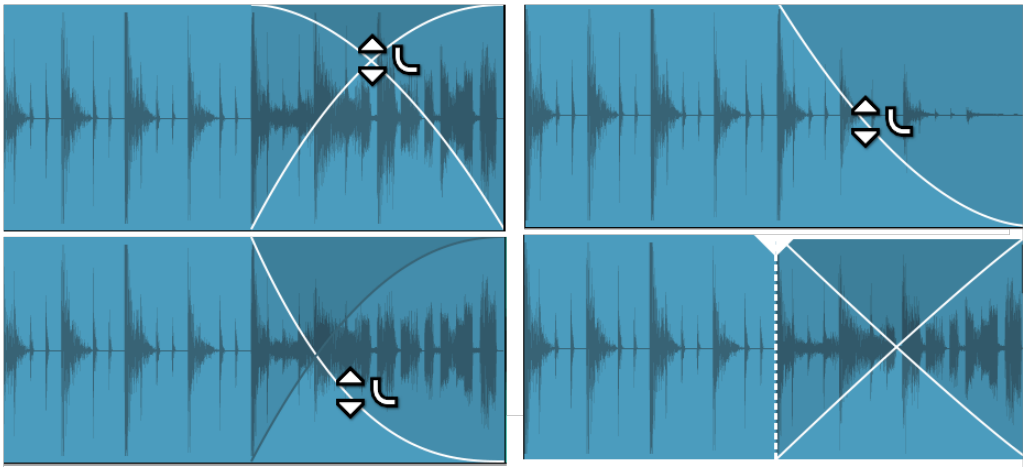


FIGURE 2.8: Manipulation of Amplitude Curves in Bitwig Studio [40]

2.1.6 Understanding the Frequencies

A classic 4 string bass guitar has a fundamental frequency range of 41 Hz (open E) to 311 Hz (high D# - 20th fret of the G-string), although many modern basses have 24 frets which extend this to 392 Hz (24th fret of the G-string). 6-string bass guitars extend this to 523 Hz, but generally, although style dependant, the fundamental bass frequency range falls between 40 and 200 Hz. Bass power ranges from 60 Hz to 150 Hz. while 80 Hz to 200 Hz adds to the fullness of a bass sound. 200 Hz to 500 Hz will create what is known as 'muddiness', meaning there is a lack of clarity. 500 Hz to 1 kHz creates punch. "Punch is just a cool word for dynamics. When people say a track needs more punch, they want the transients to sound more dynamic" [41]. 1 kHz to 5 kHz adds clarity to the attack of the bass. The sub bass sits between 20 Hz and 60 Hz, which is felt more than heard.

2.2 Sound Models

Sound Models are methods for mathematically modelling digital audio for musical synthesis and other transformations. In [42], Sound Models are grouped into four families used for sound generation: Abstract, Physical, Temporal and Spectral. An example of an Abstract Model is Frequency-Modulation (FM) Synthesis.

Abstract models do not have an analysis stage but are based purely on mathematical equations. Another example of an abstract model is 'Neurogranular synthesis', where networks of spiking neurons are used to control a granular synthesis engine [43, 44].

Physical models aim to capture the sonic properties of physical real-world instruments such as plucked string synthesis. These models provide parameters for controlling and altering the output of the mathematical models of drums, pipe organs or other instruments [28, 45–47].

Temporal models represent a sound as the amplitude of the wave over time. This is the most natural way of representing an audio signal, where a continuous signal is sampled and stored digitally as a discrete signal. Digital sampling is discussed in textbooks on the topic of discrete signal processing [48]. Temporal representations of audio are vital for storing, transporting and playing back audio signals from digital representation. A number of transformations can be applied to audio represented by this model, also referred to as the time domain representation, but the majority of these transformations are not usually related to any musical parameters extracted from this representation, rather from the sequence of numbers presented by this model. Numerous topics on the subject of time domain audio effects covering examples such as filtering, fading, modulation, non-linear processing, and many others can be found in numerous textbooks on the subject [49, 50].

Many of the above mentioned audio effects can be applied to a kick and bass sound represented in the time domain after an audio producer has shaped the sound. Fading, modulation and compression are some of the most common tools used for sculpting synthesised kick and drum sounds. A good sound model should be capable of providing parameters which capture the details of how a sound evolves over time so that these parameters can be used in transforming and re-synthesizing the sound in intuitive ways. These types of models are known as Parametric Models.

Spectral Models represent a sound in the frequency domain and are based on a former simplified model of human hearing which is based on the ear operating as a "kind of Fourier analyzer. That is, sound is spread out along the inner ear according to frequency, much like a prism separates light into various colors. As a result, hearing in the brain is based on a kind of short term spectrum analysis of sound" [51]. Recent research into auditory perception has however shown that the process of the ears ability to perceive complex sounds goes beyond this, and includes mechanisms such as temporal coding, phase locking, and neural synchronization. Temporal processing, spatial cues, and cognitive processes also play significant roles in sound perception [52, 52–58]. However, treating hearing as a frequency analyzer is still a useful approach in many contexts, and although it is a simplification, and modern spectral models should be complemented by these other factors, the human auditory system remains sensitive to different frequencies and a sounds spectral content. Spectral models are therefore still able to provide a model where transformations and feature extraction are intuitive to understand as the model parameters acquired from spectral analysis are directly related to how hearing works through spectral decomposition. "Other well known signal models rooted in the spectral point of view include the phase vocoder, additive synthesis, and so-called spectral modeling synthesis" [51]. The model adopted for use in this thesis is based on the foundations of spectral modelling, sinusoidal models and additive synthesis, which are presented in more detail in Sections 2.5.2 and 2.5.1.

2.3 Signal Representations

2.3.1 Time-Domain Representation

Digital audio is a time-domain representation of sound in a digital format. When recording musical instruments, the sound is converted into an analog electrical signal through a microphone and then converted into a digital representation with an analog to digital converter which samples the signal at a specified sample rate, determines the resolution of the signal with regards to the number of bits used to represent it and finally assigns a binary value to the sampled audio at that specific time. Temporal models are examples of this digital time-domain representation of sound. The loudness of the sound determines the value assigned to it, in a floating point data system, these values are normalised to between -1.0 and 1.0. The louder the sound is the higher amplitude value assigned.

Figure 2.9 shows the process of a continuous signal and the discrete resolution of representing the amplitude of the signal over a number of sampling points.

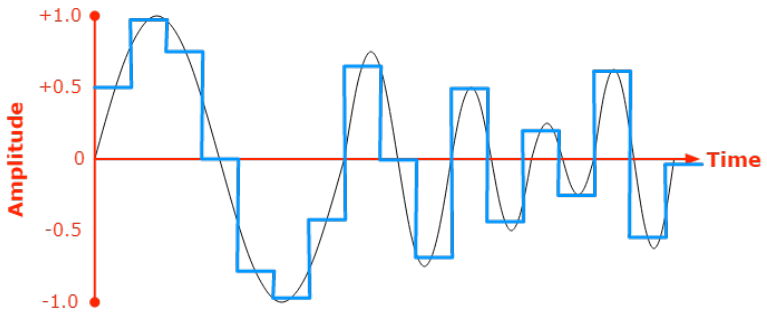


FIGURE 2.9: Digital Sampling

Figure 2.10 shows the aliased output of not sampling at a high enough rate to reconstruct a continuous signal. A higher rate of sampling is required to capture the 7 kHz sinusoid which when sampled at 8 kHz resembles a 1 kHz signal. A Sample rate of twice the frequency is required known as the Nyquist theorem. In this case a sampling rate of 14 kHz is required to represent the 7 kHz signal.

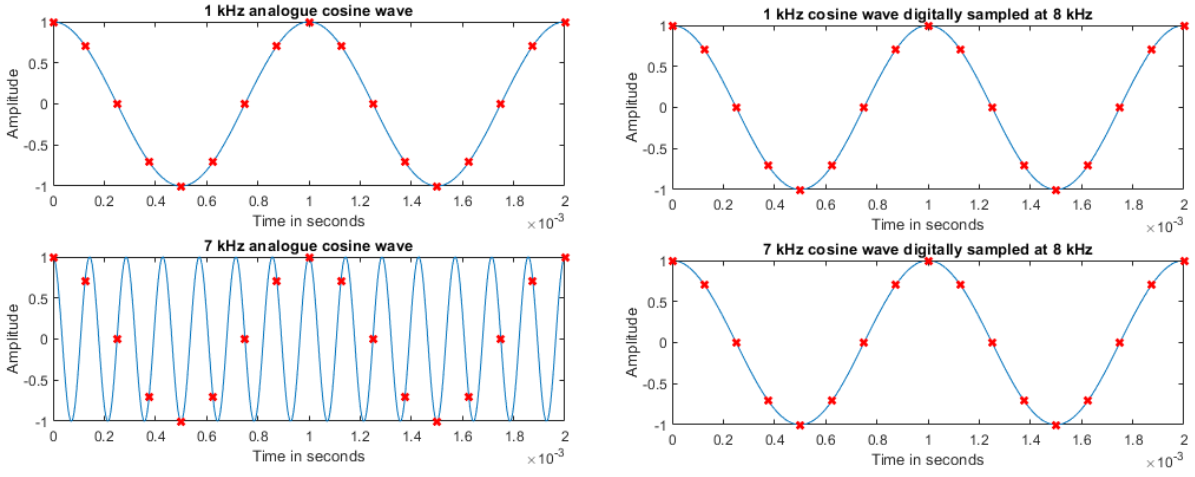
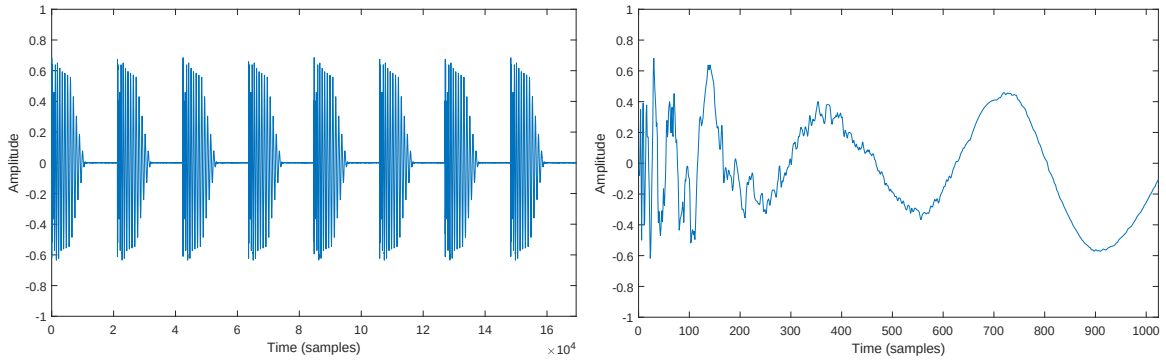


FIGURE 2.10: (A) Continuous signal and discrete sampling points. (B) Output of sampled signal showing an aliased representation

When processing digital audio in the time domain, the digital audio system processes the audio in groups of samples called a frame. An example of a recording of a kick drum is shown in Figure 2.11a. This has been segmented into audio frames 1024 samples long in Figure 2.11b which shows 0.5 seconds of audio. When processing sound in a digital audio system, a large frame of audio such as this would be split into slices of smaller frames, usually in powers of 2 for efficiency. Common examples are 32, 64, 128, to 1024 or higher.

In a real-time audio system, a delay is introduced at the output which is directly proportional to the size of the the frame size. A buffering system will sample and store a number of audio samples until a specified number of samples are available for processing. This introduces a delay between the input and the output known as latency. Low latency audio systems will try to reduce the frame size.



(A) Recording of a kick drum (B) Frame containing 1024 samples of a kick drum

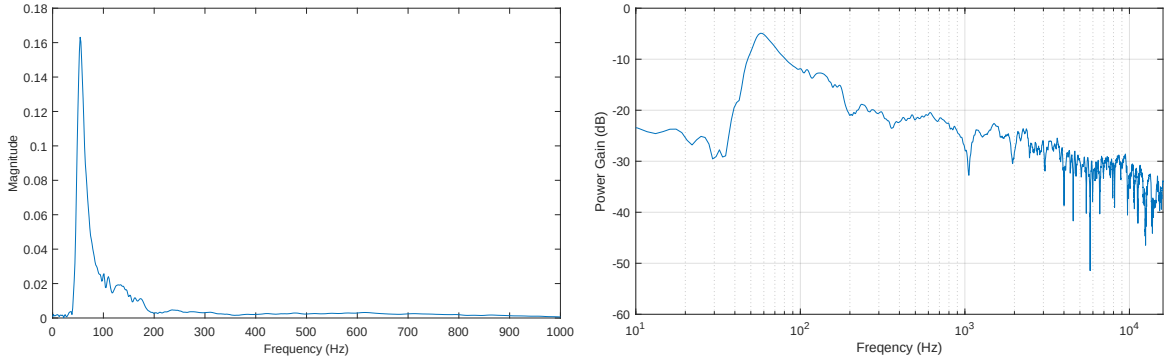
FIGURE 2.11: (A) recording of a kick drum (@48 kHz) and (B) an audio frame containing 1024 samples of the kick drum

2.3.2 Frequency-Domain Representation

Before Pythagoras concerned himself with the harmonic relationships between planetary orbits, the Babylonian astronomers used a form of Fourier analysis and harmonic series to predict and record tables of astronomical positions in a book knows as an ephemeris. Modern Fourier theory can be attributed to Daniel Bernoulli, Leonhard Euler and Joseph Louis Lagrange who's combined work introduced trigonometric functions. It was not until the insights of Jean Baptise Joseph Fourier that argued that any function could be described as trigonometric sums [59].

"Fourier took this type of representation one very large step further than any of his predecessors. Specifically, he obtained a representation for aperiodic signals - not as weighted sums of harmonically related sinusoids - but as weighted integrals of sinusoids that are not harmonicaly related." [60]

Figure 2.12 shows the frequency reasponse of the kick drum presented in Figure 2.11b, plotted along the linear frequency axis in (A) and the Log-Magnitude domain in (B).



(A) Linear FFT spectrum (B) Log Magnitude spectrum

FIGURE 2.12: (A) FFT spectrum of a Kick drum from 2.11 and (B) Log Magnitude spectrum

2.3.3 Time-Frequency Representation

Time Frequency analysis of digital audio data has been at the forefront of describing audio in an intuitive way by transforming the audio from the time domain to the frequency domain representation. Representing a one dimensional signal (time) in two dimensions (time and frequency) was initially proposed by Gabor in [61] where he applied Gaussian window functions to decompose a signal into time and frequency coordinates. Any complex sound can be broken down into a collection (possibly infinite) of individual sinusoidal (sine and cosine or complex exponential) functions with certain time varying amplitude, frequency and phase parameters. However there exists a slight problem with practical use of Fourier analysis and any Time-Frequency transform which is described by the Heisenberg Uncertainty Principle which states that the information between the Time and Frequency domain representations is indirectly-proportional to one another. The more information required about when an even occurs in the time-domain results in a loss of information about the event in the frequency-domain and vice versa. "The position and the velocity of an object cannot both be measured exactly, at the same time" [62]. This presents itself in the practical implication of the Discrete Fourier Transform (DFT) where a frame of audio containing a certain number of samples is converted into the frequency domain. The result being that the shorted the frame is in the time domain results in a loss in resolution in the frequency domain. The more information and number of samples contained in an audio frame in the time domain will result in a greater detail in the frequency domain, but at the cost of having less information about what happened in that duration of time.

The Fourier Transform and DFT use a basis of stationary sinusoidal functions (sine and cosine) that span the entire signal. The sinusoidal components returned from the DFT do not provide direct information about how a sound changes over time.

The practical solution to this is to slice the audio in the time domain into small slices which can be processed separately, known as the Short Time Fourier Transform (STFT). This gives a better understanding of what sinusoidal partials parameters are at that moment in time, but the information and parameter coefficients computed are still averages over the length of the frame, and there is a trade-off between the time and frequency resolution which is proportional to the frame length. Longer audio frames provide better frequency information but at a loss of time information.

Figure 2.13 presents the time-frequency resolution trade off between using a short window with a better time resolution, compared to using a longer window with better frequency resolution at the cost of the loss of temporal information.

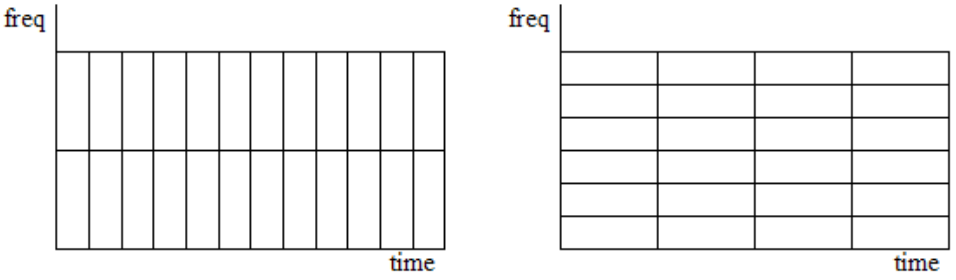


FIGURE 2.13: Comparison of Time-Frequency resolution of two different analysis frame sizes and the STFT

"Where signals under transformation contain non-stationary components a common approach is to divide them into shorter analysis frames within which those components can be considered to be quasi-stationary. This is known as the short-time Fourier transform. Longer frames increase frequency resolution but at a cost of temporal resolution and the optimum frame length is often determined to be the point at which the assumption of component stationary breaks down. A problem here is that useful and interesting audio signals tend to be multi-component with different localisation properties in both time and frequency. This leads to a compromise between time and frequency resolution in which the quasi-stationary assumption is violated for at least some of the components. Information about non-stationary is not lost in the Fourier domain (since the transform is perfectly invertible) but it is embedded in the relationships between the phase and magnitude of multiple transform bins, rather than being more directly accessible" [10]

Wavelets analysis, presented in Section 2.10, is another time-frequency transform which aims to overcome the restrictions of a fixed time-frequency resolution, and achieve a better resolution in both time and frequency by the use of a time-scale system.

2.4 Spectral Analysis

This section presents a brief introduction to the DFT and STFT. Properties of the STFT are presented with other common practises of Windowing, Zero Padding and Zero-Phase Windowing. The DFT can suffer from what is known as 'spectral leakage' when a non-integer multiple of the period given by the length of an analysis frame is analysed, resulting in a spreading of the frequency information into neighbouring bins. Windowing a signal $s(t)$ by a tapering window w minimises this problem but requires analysis frames to overlap by a specific amount to achieve perfect reconstruction. Zero-padding, is the process of appending zeros to the end of the audio frame. This extends the analysis frame size N therefore improving the frequency resolution of a spectral bin. This does not reveal more information about the spectrum, longer windows are required for that, but it does improve the resolution by interpolating between spectral bins.

2.4.1 Discrete Fourier Transform

The DFT is often expressed in the form of a complex exponential using Eulers notation of $e^{j\theta}$. Euliers identity is the quality of $e^{i\pi} + 1 = 0$ where e is the base of natural logarithms, i is the imaginary unit ($i^2 = -1$), π is the ratio of the circumference of a circle to its diameter, and θ is the angle in radians.

The DFT is defined in [26] as:

$$X(k) \triangleq \sum_{n=0}^{N-1} x(n) e^{-j2\pi nk/N}, \quad k = 0, 1, 2, \ldots, N-1 \tag{2.1}$$

where X is the frequency spectrum, $X(k)$ is the spectrum at the k th spectral sample (bin), and $x(n)$ is the input signal at time (sample) n, containing N samples. The samples of the DFT are complex and contain a real X_r and imaginary X_i part.

The magnitude $|X(k)|$ is give by:

$$|X(k)| = \sqrt{X_r(k)^2 + X_i(k)^2} \tag{2.2}$$

The phase $\phi(k)$ is computed by:

$$\phi(k) = \arctan \frac{X_i(k)}{X_r(k)} \tag{2.3}$$

2.4.2 Zero Padding

Zero-padding, which entails appending the input signal with zeros in the time domain, is equivalent to interpolating in the frequency domain. This does not result in a higher frequency resolution in the frequency domain, but provides more detail by interpolation between spectral samples.

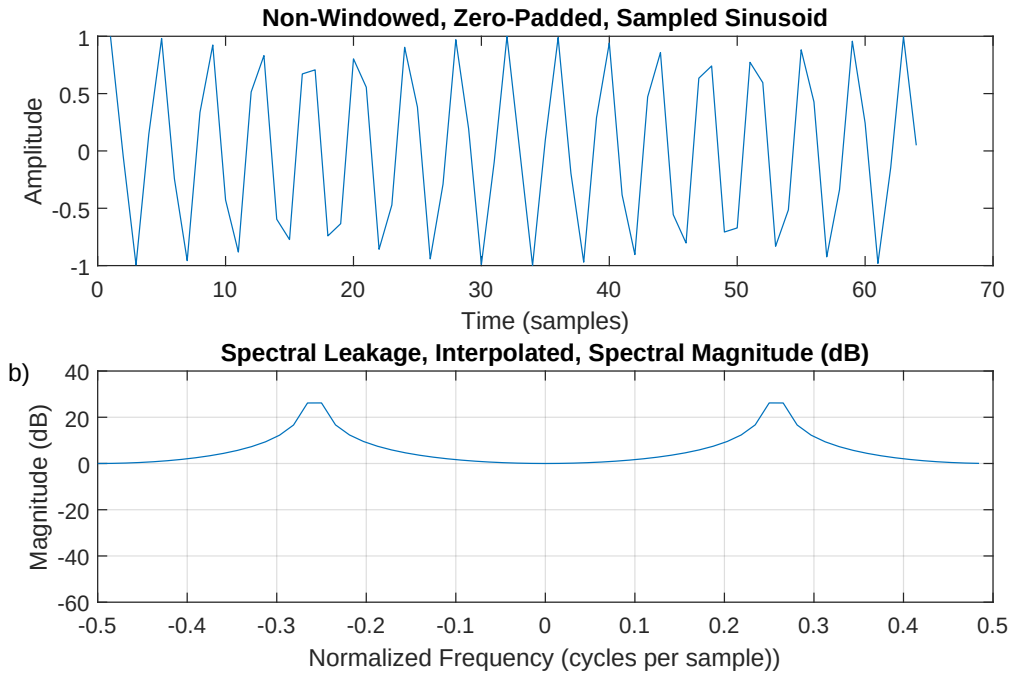


FIGURE 2.14: Non-padded, non-windowed sinusoid. a) Time-domain waveform. b) Magnitude spectrum (dB)

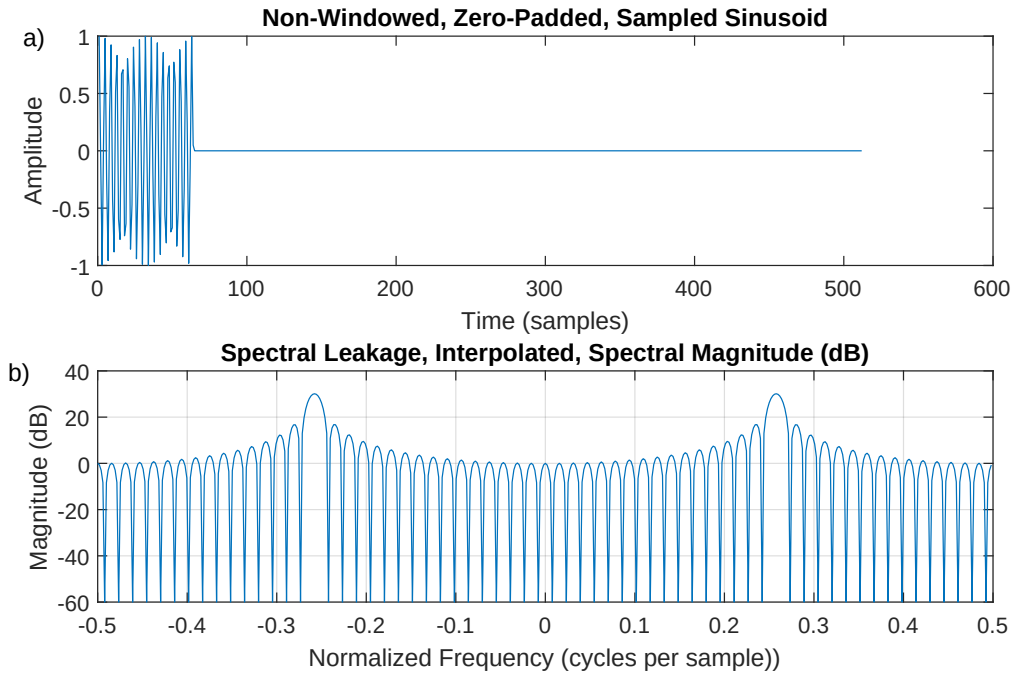


FIGURE 2.15: Sinusoid from 2.14 Zero-padded. a) Time-domain waveform. b) Magnitude spectrum (dB)

Figures 2.14 and 2.15 compare the output of the magnitude spectrum with and without Zero-Padding.

2.4.3 Windowing

Spectral leakage is primarily caused when a frequency does not perfectly align with the DFT basis vectors. This causes the energy of the sinusoid to spread to neighboring frequency bins which results is what is known as spectral leakage. The effect Windowing has on the magnitude spectrum is can be seen in he comparison of Figures 2.15 and 2.16. Windowing mitigates spectral leakage by applying tapering window to the input signal. The DFT equation in 2.4.1 is modified to:

$$X(k) \triangleq \sum_{n=0}^{N-1} x_w(n) e^{-j2\pi nk/N}, \quad k = 0, 1, 2, \ldots, N-1 \tag{2.4}$$

where the window function is represented by $w(n)$

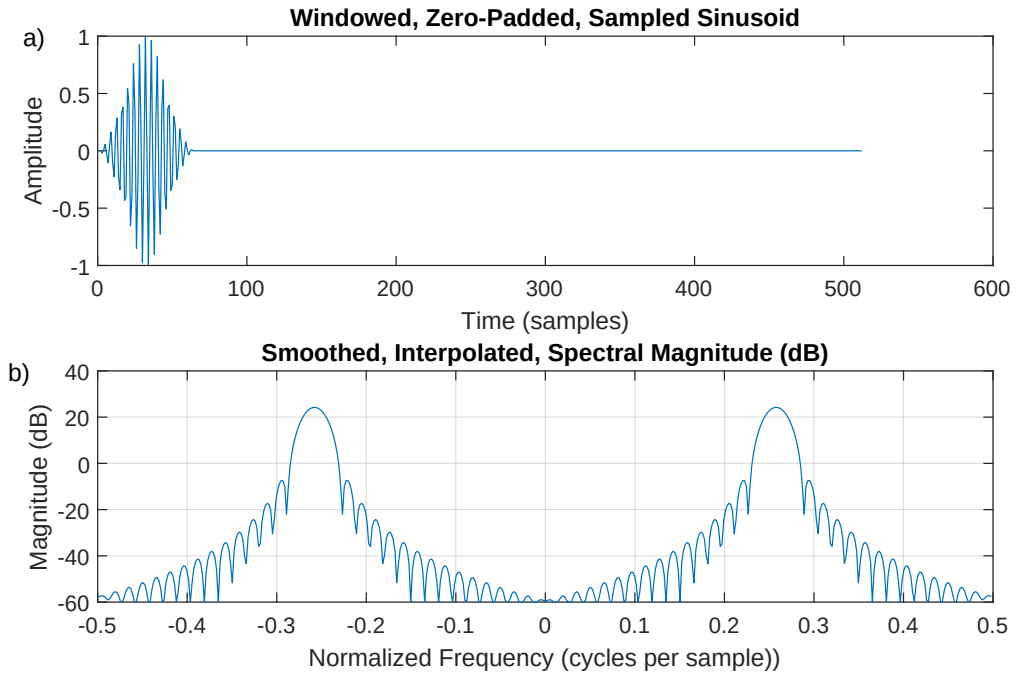


FIGURE 2.16: Sinusoid from 2.14 Windowed and Zero-padded. a) Time-domain waveform. b) Magnitude spectrum (dB)

2.4.4 Zero-Phase Padding

Zero Padding a signal in the time domain causes phase shifts in the resulting phase plots which are related to the amount of zero samples added to the input frame. Zero-Phase windowing and padding, uses a odd frame length with the signal centered in the middle of the window. The audio buffer is then rotated around the mid point of the frame such that the second half of the audio frame moves to the beginning of the frame and vice versa. Centering the signal to the middle of the audio frame at 0 results in a non-causal signal which spans from [-1//2, 1//2] and not [0, 1]. Zero-Phase padding places the padding at the center of the audio frame rather than at the end, which results in a flat phase response across the main lobe of a stationary sinusoidal peak. Figure 2.17 displays a 1 kHz stationary sinusoid, windowed (Hanning) and zero-padded with the resulting phase plot, in comparison to the same signal zero-phase padded in Figure 2.18.

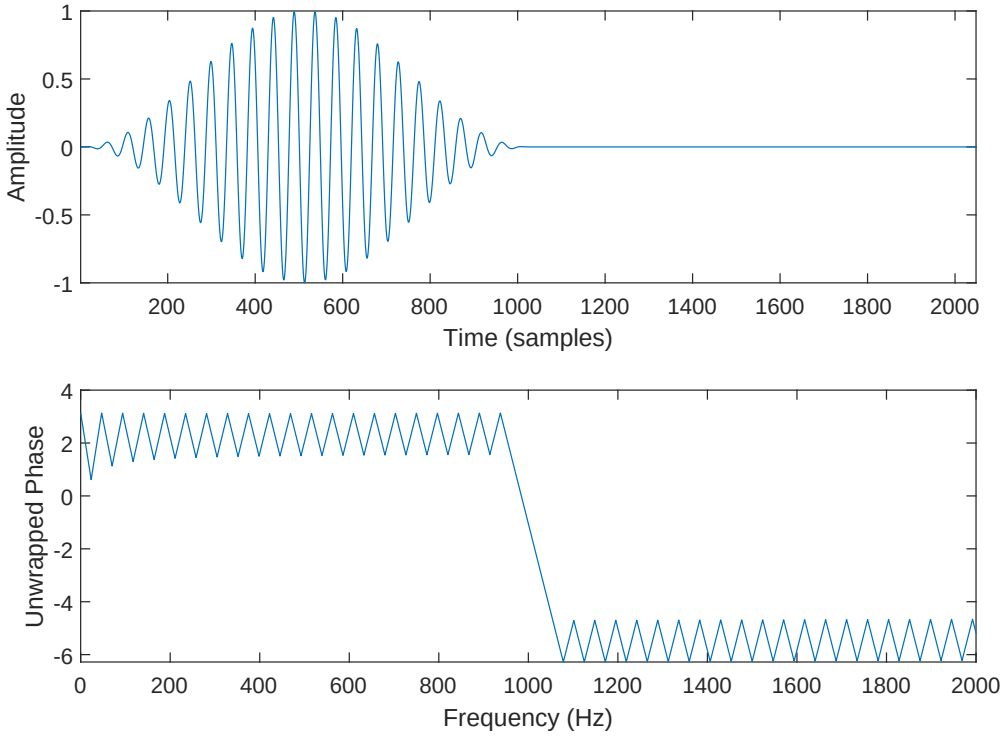


FIGURE 2.17: 1kHz sinusoid (@48 kHz) Windowed and Zero-Padded, and the unwrapped phase

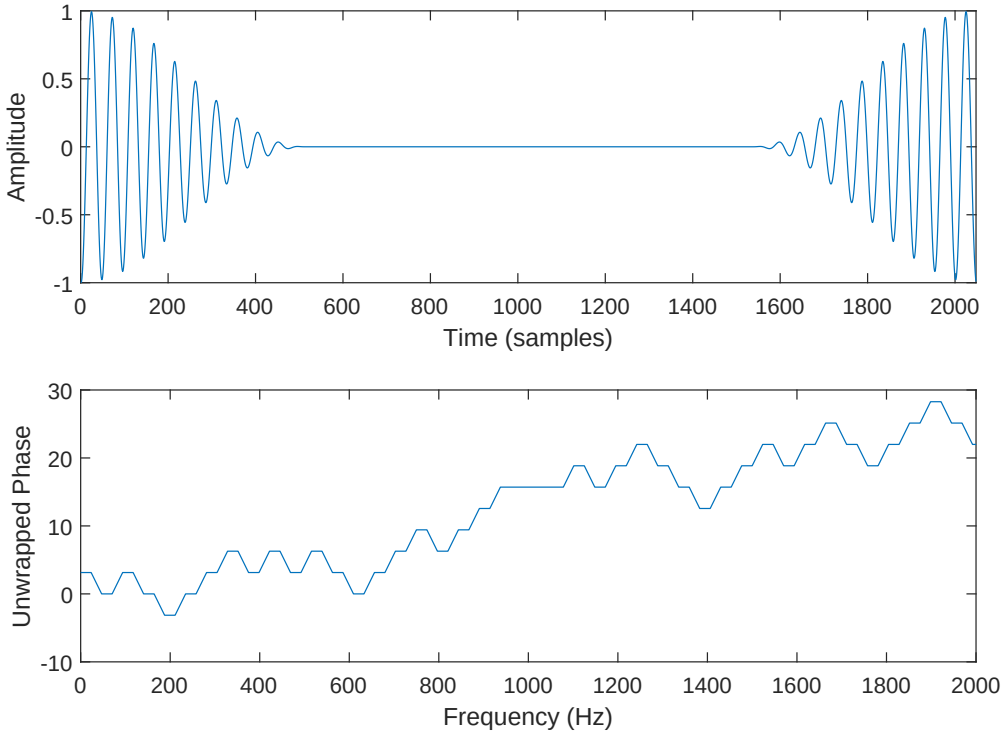


FIGURE 2.18: 1kHz sinusoid (@48 kHz) Windowed and Zero-Phase Padded, and the unwrapped phase

2.4.5 Overlap Add

In spectral modelling analysis and re-synthesis, Overlap-Add (OLA) is often applied and required when a non-rectangular window is applied to the analysis frame. The windowing reduces the abrupt changes at the frame boundaries, resulting in smoother transitions, and minimizes spectral leakage from possibly abrupt amplitude's at the start and ed of an analysis frame. The windowing process can however also introduce some spectral leakage due to the tapering and gradual decrease in amplitude at frame boundaries, which results in some energy spreading across neighbouring bins. Overlap-Add is used to to compensate for the spectral leakage and results in a more accurate reconstructed signal. Figure 2.19 displays the application OLA for performing the forward and inverse DFT on a segmented audio signal. OLA helps preserve the re-synthesised signals spectral information by reducing the effect of spectral leakage and restoring the energy caused by the widowing of the analysis frame. Although the overlap-adding of windowed frames endures a smooth transition between frames and improves the accuracy of he reconstructed signal, the overlapping of frames in the time domain can introduce some filtering and phase effects on the reconstructed signal.

The overlapping process combines the frequency content of adjacent frames which can result in changes to the frequency and phase response of the reconstructed signal. The filtering and phase effects caused depend on the window type, frame length and hop size between frames. The overlap-adding can cause some low pass filtering of the reconstructed signal, and although the phase relationships between different frequency components can be altered due to the overlapping process, these effects are minimal and generally considered acceptable for most applications. If time stretching or pitch shifting effects are applied, such as in the case of the the Phase Vocoder 2.5.3, to the reconstructed signal then phase coherence becomes important. As the frames are added together, the phase information of the overlapping regions can interfere with each other, leading to phase discrepancies. These phase discrepancies can result in artifacts such as phase cancellation or smearing of transient signals. Phase correction algorithms can be employed to mitigate these effects between overlapping frames with the aim of preserving the phase coherence and minimize phase distortions during the overlap-add process.

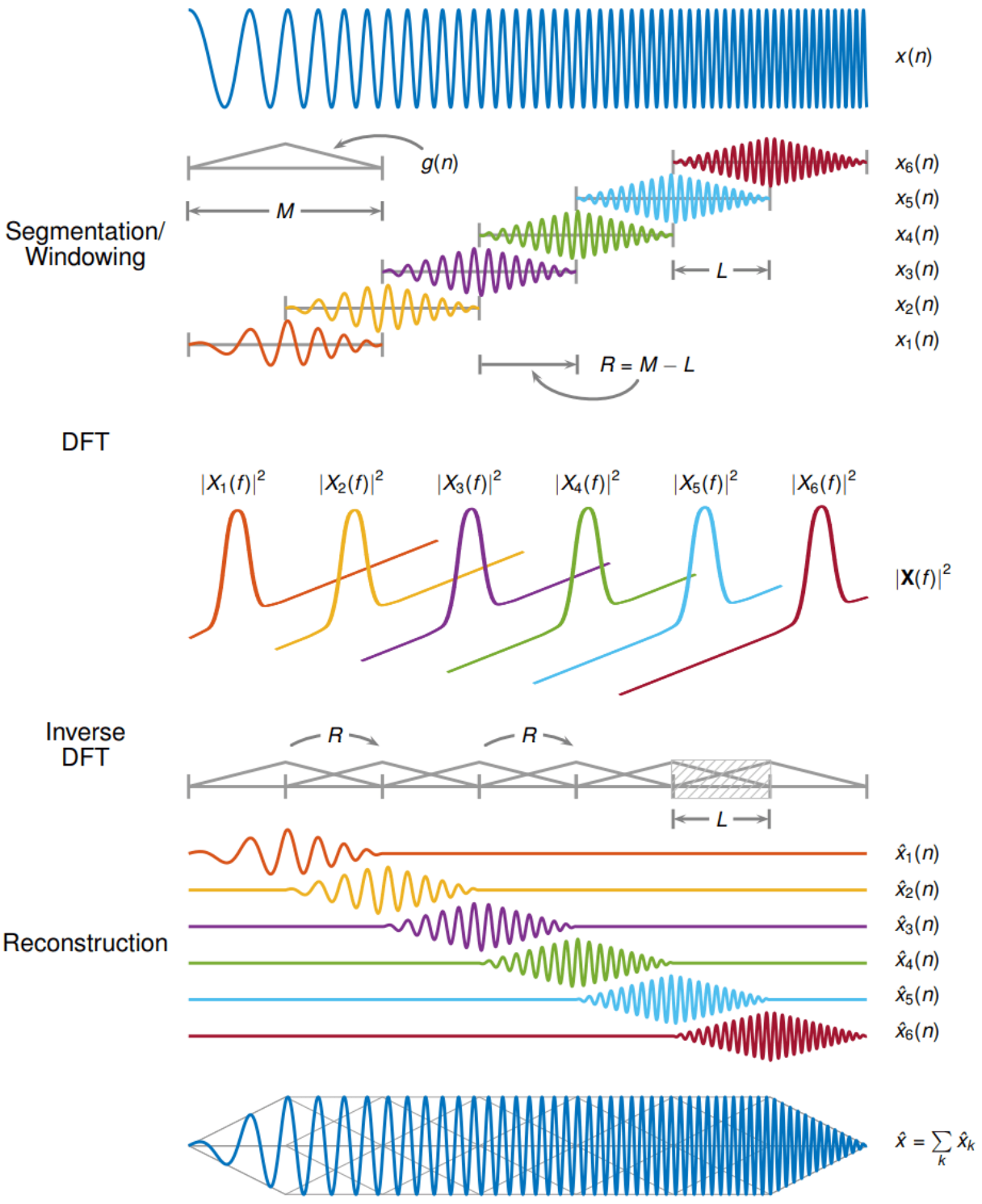


FIGURE 2.19: Overlapping of Windowed SFTF to achieve reconstruction [63]

Other overlap-add methods include the synchronised OLA algorithm (SOLA) [64], which aims to "avoid pitch period discontinuities or phase jumps at waveform-segmented joins proposes to realign each input segment to the already formed portion of the output signal before performing the OLA operation", Pitch Synchronous Overlap and Add (PSOLA) [65, 66], Waveform Similarity Overlap-Add (WSOLA) [67]. Another form of synchronization is obtained by applying a time-domain pitch-synchronized OLA technique (TD-PSOLA) [68].

2.5 Spectral Modelling Synthesis

Most musical instruments and many other sounds produce a perceivable pitch over time. The oscillations produced by these "periodic (quasi-periodic) signals exhibit a harmonic spectrum" [69] which reflect as sinusoidal peaks within the frequency spectrum that can be modelled as quasi-stationary components over short periods of time through the STFT. Additive synthesis which is based on Fourier theorem states that any periodic waveform can be modelled as a sum of sinusoids with time varying amplitudes, frequencies and phase. Additive synthesis is extensively described in [70], and is viewed as the original spectrum modelling technique [71]. A general definition of Spectral Modelling Synthesis is any "parametric creation of a short-time Fourier transform" [51], meaning any process which takes the short-time Fourier transform and processes the Fourier data in some way before converting it back to the time domain.

2.5.1 Additive Synthesis

Additive synthesis is a method well suited to producing pitched sounds through a bank of controllable sinusoidal oscillators. Figure 2.20 displays how a square wave is approximated through the addition of components with odd-integer harmonic frequencies to that of the fundamental. Additive synthesis is the method used in this thesis for generating the audio output from the models parameters.

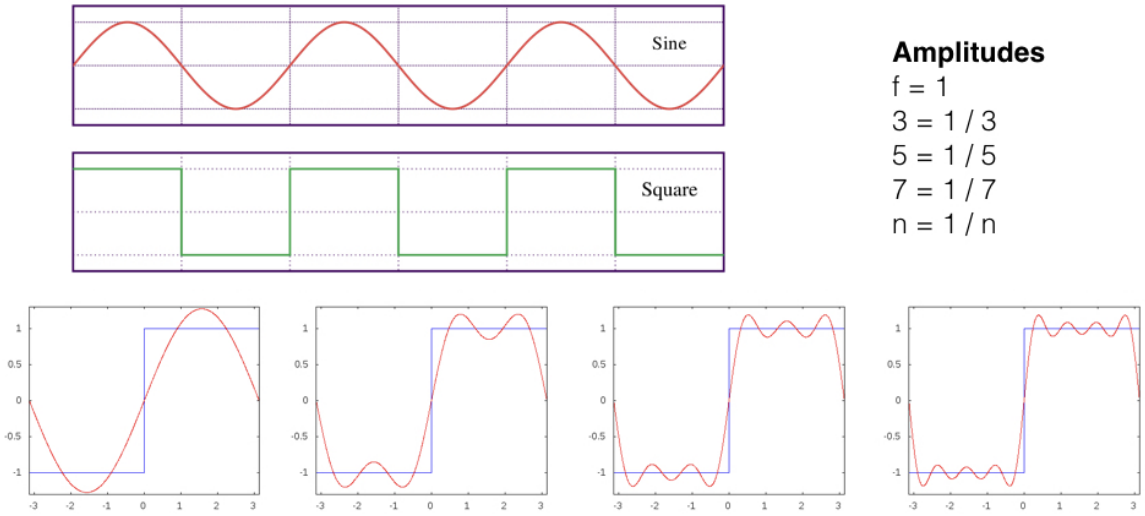


FIGURE 2.20: Approximation of a square wave with Additive Synthesis

2.5.2 The Sinusoidal Model

A sinusoid is defined as:

$$s(t) = A(t)\cos(\theta(t)) \tag{2.5}$$

where t is time in seconds, A is the instantaneous amplitude, θ is instantaneous phase.

When the phase is time varying this is defined as:

$$s(t) = A(t)\cos(\omega t + \phi(t)) \tag{2.6}$$

where ωt is the instantaneous radian frequency with a time-varying phase offset $\phi(t)$

The instantaneous frequency is given by taking the time derivative of the instantaneous phase:

$$\frac{d}{dt}[\omega t + \phi(t))] = \omega + \frac{d}{dt}\phi(t) \tag{2.7}$$

A sinusoid is fully described in terms of amplitude, frequency and phase and so a system modelling a single sinusoidal partial for resynthesis will take the following form:

$$s(t) = A(t)\cos(2\pi ft + \phi) \tag{2.8}$$

where a, f, and ϕ are the amplitude, frequency and phase parameter estimates.

A spectral modelling system taking into account a number of sinusoidal components known as partials is expressed as a sum of these, given by

$$s(t) = \sum_{p=1}^{P} A_p(t)\cos(2\pi f_p t + \phi_p) \tag{2.9}$$

where P is the total number of sinusoidal components and p represents a specific partials parameter estimates.

When expressed as a complex exponential a non-stationary sinusoidal model with time varying amplitudes and frequencies can be expressed as:

$$s(t) = \sum_{p=1}^{P} A_p(t)e^{j\phi_p(t)} \tag{2.10}$$

Sinusoidal models which adapt a linear amplitude curve will atopt the definition given in 2.10 while models adopting exponential amplitude envelopes can represent $A_p(t)$ as an exponential with a log-amplitude polynomial argument expressed as:

$$e^{log(A_p(t))} \tag{2.11}$$

Incorporating this into the model has the advantage of expressing the model as a single exponential with a complex argument given by:

$$s(t) = \sum_{p=1}^{P} e^{\log(A_p(t))+j\phi_p(t)} \tag{2.12}$$

2.5.3 Phase Vocoder

The phase vocoder, originally developed for encoding speech [72–74], decomposes a signal into a collection of frequency bins containing magnitude and phase information from the STFT. The spectrum is then transformed in some way and the output from the Inverse Fourier Transform is overlapped and added in the time domain to get the output signal. One of the problems with the phase vocoder is that it is restricted to analysing and altering "a fixed number of filter banks returned from the FFT, and the frequency of each sinusoid can not vary outside the bandwidth of its bank" [71], making in-harmonic and frequency modulated sounds difficult to analyze.

An in-harmonic phase vocoder was developed in [75] known as PARSHL which attempted to overcome some of these problems by supporting "inharmonic and pitch-changing sounds" by using a simple peak tracking algorithm. Instead of tracking the magnitude and phase of each bin, this implementation tracked the dominant peaks and the associated amplitude, frequency and phase trajectories from one DFT frame to the next. PARSHL perfroms data reduction on the STFT through a process of sinusoidal peak selection where only the most prominent sinusoidal peaks are tracked between overlapped frames and parameters for amplitude and frequency estimated. PARSHL had the option to use either Additive (using oscillator-control envelopes) or Overlap-Add (using inverse FFT) Synthesis. If additive synthesis is used, then amplitude and frequency trajectories where linearly interpolated across frames from previous and current instantaneous amplitude and frequency estimates given by

$$F_k(m) \triangleq \frac{\Theta_k(m) - \Theta_k(m-1)}{2\pi HT} \quad (\text{ Hz}) \tag{2.13}$$

where the instantaneous frequency $F_k(m)$, is given from differentiating the unwrapped phase between current and previous frames. H is the hop size and $T\,(= 1/f_s)$ the sample period (in seconds). $A_k(m)$ and $\Theta_k(m)$ are the magnitude and phase estimates for each bin (k) in each frame (m), estimated by converting between rectangular to polar form using 2.4.1 and 2.4.1. The instantaneous amplitude is improved using parabolic interpolation as described in 2.7. The phase $\Theta_k(m)$ is usually discarded and calculated from the instantaneous frequency if required by

$$\hat{\Theta}_k(n) \triangleq \hat{\Theta}_k(n-1) + 2\pi T \hat{F}_k(n) \tag{2.14}$$

Other improvements to the phase vocoder over the years mail aim at improving transient analysis and improving time and pitch scale modifications. Maintaining phase coherence between neighbouring frequency bins is an example of improving time and pitch scale modifications in the frequency domain with the phase vocoder [76, 76–82].The inflexibility and fixed resolution of traditional overlap-add implementations has lead to a number of improvements which aim to align overlapped frames in such a way that periodic structures in the frames waveforms are aligned in the overlapping regions" [67]. These methods aim at synchronising overlapping frames in a way which aligns repeated regions within the time domain signal with the position of the overlapping Windows such as synchronized OLA (SOLA) [64, 83], Pitch Synchronous OLAd (PSOLA), waveform similarity OLA (WSOLA) [67] and Time-Domain Pitch Synchronous OLA (TD-PSOLA) [65, 66, 68, 84]. Other improvements to the phase vocoder over the years range from improving the quality of the time or pitch scaled output by maintaining phase coherence and phase locking techniques, to transient modelling and processing [76, 85–96].

2.5.4 The Deterministic and Stochastic Model

Spectral Modelling Synthesis (SMS) [71, 97] is based on deterministic and stochastic decomposition. The assumption is that any sound can be decomposed into a deterministic and stochastic part, where the deterministic part is comprised of the dominant partials from the signal, and is a collection of sinusoids with piecewise linear amplitude and frequency functions. The stochastic component can be referred to as time-varying filtered noise, or as a collection of magnitude spectrum envelopes that function as a time varying filter that is excited by white noise. Essentially the shape of the magnitude spectrum is applied to white noise, this shaping of the noise spectrum filters it, and so the stochastic part is a relationship between its "amplitude probability density versus frequency, or its power spectral density" [71].

The basic principles of modelling the deterministic components within a signal using the STFT were introduced simultaneously by Smith and Serra for use in music, and by McAuleay and Quatieri for speech analysis [88, 98], with some differences highlighted in [71]. These techniques provided an improved model over the phase vocoder by tracking frequency components across neighbouring FFT bins. The drawback with a purely deterministic model is that it is computationally expensive to model noisy components such as transients / musical onsets due to the number of short-term broadband spectral components within these signals. SMS extended the sinusoidal model used in PARSHL, to

include a Stochastic part through the inclusion of a noisy residual signal in the model after sinusoidal extraction. One of the aims of this model was to synthesis noise like components more efficiently. The objective of Spectral Modelling Synthesis was to develop an "analysis/synthesis system that allows the largest possible number of transformations on the analysis data before resynthesis." [97]

The model introduced by Serra in [97] differs slightly from that proposed in [71], in that the initial model kept track of the phase information and calculated the residual signal by subtracting the synthesised components in the time domain. The model introduced by Serra and Smith did not preserve the phase of the original signal within the deterministic component, and therefore time domain subtraction could not be preformed. The residual signal was therefore calculated using spectral subtraction.

Both the phase vocoder and sms perform the analysis on overlapping audio frames. The length of the analysis window, the type of window function chosen and the amount of overlap dictated by the hop size, are important factors influencing the models performance. The frame size is important for dictating the time and frequency resolution. The frame size needs to be sufficiently long enough to capture closely spaced components. The type of tapering window used will have differing properties such as the width of the main lobe and the height of the largest side lobe. Ideally a window with the narrowest main lobe and lowest side lobe would be chosen [99]. The hop size is also important for improving the resolution in time dictated by the frame length by providing more analysis frames and a smoother result, but at a greater computational cost.

Peaks are defined as local maxima in the magnitude spectrum such that a frequency bin's magnitude is greater than both magnitude values of the frequency bins on each side. The detection of peaks can be improved by constraining the frequency range and only searching for values above a certain magnitude.

Periodicity or a fundamental frequency in the signal can improve the results from tracking a set of peak trajectories. Pitch synchronous analysis can then be applied by setting the size of the analysis window to that of the fundamental frequency.

The next stage of the analysis is to distinguish the stable sinusoids by comparing the peaks from current and previous frames. Peaks from each frame are organised into peak trajectories through a peak continuation analysis stage. The peak continuation algorithm can be adapted and changed to provide a better fit with different types of sounds. Serra and Smith's base their peak continuation algorithm on the idea of "frequency guides" that advance in time through the spectral peaks looking for best matches between peaks in consecutive frames and forming trajectories out of them.

The peak continuation algorithm can be modified to improve the results if the sound being analysed is harmonic in nature, as the frequency guides can actively seek harmonically related partials to track.

The attach portions of most natural sounds are very noisy and so it is very hard to search for partials in these very frequency rich, unstable and noisy sections of the sound. One approach for overcoming this is to start the analysis process from the end of the sound and work towards the beginning. Detecting musical note onsets and transient events is another approach discussed in the following section on transient modelling 2.6.

Once the deterministic part of the signal has been found, one can find the residual signal by either subtracting the deterministic signal from the original in the time domain, if we have kept track of the phase information (Serra 1989), or we can subtract the deterministic signals magnitude spectrum from the original signals magnitude spectrum resulting in a set of residual magnitude spectrums. If we have kept track of the phase and have derived the residual from time domain subtraction, then we need to perform an additional FFT on the residual time domain signal to arrive at the residual signals magnitude spectrum before we can perform the residual magnitude line segment approximation. Generating the deterministic signal and preserving the phase information is very computationally expensive and so the frequency-domain subtraction is preferred as one does not need to maintain any phase information, although the results from time domain subtraction are more accurate.

SMS assumes that the residual signal is stochastic, and that it is fully described by its amplitude and frequency characteristics. “It is unnecessary to keep either the instantaneous phase or he exact frequency information. Based on this, the stochastic residual can be completely characterised by the envelopes of the magnitude-spectrum residuals: i.e., these envelopes keep the amplitude and the general shape of the residual spectrum. The set of envelopes forms the stochastic representation.” [71]

Essentially the residual signal is a collection of magnitude envelopes that approximate the shape of the residual signal after the deterministic components magnitude spectrum has been subtracted from the original signals magnitude spectrum.

This has been extended further to include Transient modelling into the model, Multi-resolution approaches, and other higher level approaches such as “Feature based analysis / synthesis”.

2.6 Transient Modelling

Sinusoidal and other adaptive models, have been extended traditional sinusoidal modelling in the past to include the modelling of transients separately, expanding the components into a model with sinusoids, transients and noise [100–110]. Audio compression standards such as MPEG, AAC, Dolby-AC-3 include transform coding techniques to model and encode transients [111–113].

There are numerous methods and approaches to Transient / Steady-State (TSS) separation, partially due to an unclear definitions and clear distinguishing between 'transient' and 'steady-state' regions for musical signals [114]. The application of TSS is also applied in different applications such as audio encoding, music information retrieval and in audio effect algorithms, where it may be desirable to modify only the transient part, or to leave the transient section unchanged while altering the steady-state part. Transients from sharp percussive instruments differ greatly from slower rising attacks of other instruments. The selected method of modelling musical instrument note onsets has a direct effect on the components remaining in the residual signal and thus impacts how the residual signal is modelled. An evaluation of several methods including Linear Prediction (LP), MDCT, DWT and other spectral modelling methods are compared and evaluated in [114].

Transient Modelling Synthesis (TMS) was introduced by Verma et al in [104–108] as an extension to traditional sinusoidal models [71, 98] to include transients by exploring the time-frequency duality property of the Fourier transform, where a pure sinusoid appears as an impulse in the frequency domain and where an impulse in the time domain appears periodic in the frequency domain. The Discrete Cosine Transform (DCT) is chosen to provide the mapping between the time and frequency domains such that transients in the time domain become sinusoidal in the DCT domain and can then be modelled using spectral analysis from this new representation in the same manner as the sinusoidal part is. The DCT is essentially a middle step which provides a mechanism for encoding transients as sinusoidal components which provides a sparse representation for modelling transients that is especially interesting in audio compression. In [115] Levine et al describe a sines-transient-noise (STN) model for compression as well as time and pitch modification which uses the multi-resolution sinusoidal modelling from [116] and a simplified transform coder where the transient window is separated into short (256 sample) segments and encodes the transients using 24 overlapping windows for a total length of 66 ms. The latest versions of AAC uses a standard switched MDCT (Modified Discrete Cosine Transform) filterbank with an impulse response (for short blocks) of 5.3ms at 48 kHz compared with 18.6 ms for MPEG Layer-3 which reduces the amount of pre-echo artifacts [117, 118].

The above mentioned methods all result in block lengths under 1024 samples, which is the maximum constraint for a quasi real-time implementation. However, they rely on an additional analysis stage of onset and transient detection.

TMS and transient encoding used by MPEG standards are reliant on information about temporal events such as attack time, and require an additional step of onset detection. In [119] a large class of onset detection methods (spectral difference, phase deviation, wavelet regularity modulus, negative log-likelihood, and high-frequency content) are compared and revisited in [120] and [121]. Further notes on onset detection using energy, phase and pitch based methods are available from Zhou and Reiss in [122, 123]. An interactive approach of transient detection using the STFT is given by [124] which builds on from the works by Ono et al [125, 126] of separating harmonic and percussive components. A method for estimating the spectral envelope for pitch shifting with spectral envelope preservation is given in [127]. This work has been expanded on and used for evaluating temporal evolution for automatic segmentation in [128]. Time frequency reassignment has been used by Roebel as a measure of the center of gravity in an onset detection and classification scheme in [129, 130]. However, all of these methods are particularly suited to certain types of signals and are unable to perform equally well for all musical signals and different instruments.

The need for an explicit transient model is presented in [108], highlighting that although it is possible to model transients and noise by a sum of sinusoidal signals, this is an inefficient representation requiring many sinusoidal components. It is also argued that this is not meaningful, "because transients are short lived signals while the sinusoidal model uses sinusoids that live on a much longer time-scale".

In [115] Levine highlights that sinusoidal modelling systems struggle to model sharp attacks as they are not efficiently represented as a sum of sinusoids, but does elaborate that it is possible, "but such a system would need hundreds of sinusoidal parameters, consisting of amplitudes, frequencies, and phases".

2.7 Improving Parameter Estimation based on DFT

The accuracy of parameter estimates for a sinusoidal peak derived directly from the DFT are limited by the frequency resolution of the DFT. The frequency resolution of a frequency bin from the DFT is related to the sampling rate and the length of the analysis frame. This results in a fixed frequency resolution with the frequency estimate of a sinusoidal peak given at the center of the frequency bin.

The amplitude estimate is also given as the peak magnitude value at the center of that bin. Amplitude and frequency estimates of sinusoids with frequencies deviating from the exact center of a DFT bin will result in slightly incorrect results due to this.

The STFT uses stationary sinusoidal basis functions to decompose a signal into sinusoidal components which are assumed to be stationary for the length of the analysis frame. Many signals, including musical signals with vibrato or tremolo, however have continuously varying amplitudes and/or frequencies. Amplitude and frequency change has the effect of flattening the magnitude of a spectral peak in the frequency domain, leading to biased instantaneous amplitude estimates.

Improving frequency estimates from the phase information between analysis frames has been presented in 2.5.3. Quadratic (parabolic) interpolation [131–136] is another popular and relatively inexpensive method of improving amplitude and frequency estimates of a spectral peak by fitting a parabola between the points around a peak. "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 [136]."

The estimated frequency of a sinusoidal peak using parabolic interpolation is given by

$$f_n = B(n + \frac{1}{2}\frac{M_{n-1} - M_{n+1}}{M_{n-1} - 2M_n + M_{n+1}}) \tag{2.15}$$

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

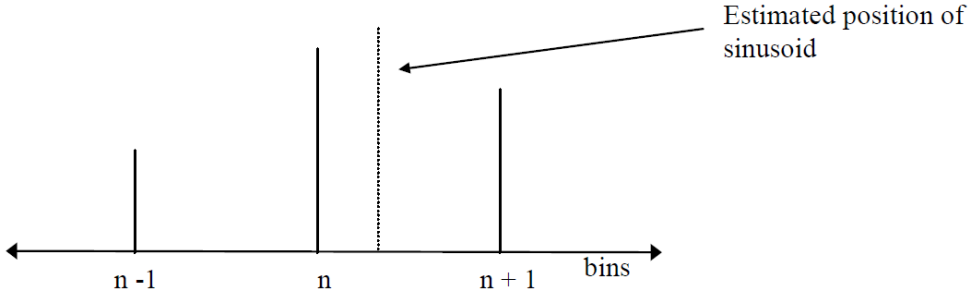


FIGURE 2.21: Parabolic interpolation of a spectral peak

The estimated amplitude of a sinusoidal peak a_n using parabolic interpolation is expressed in dB by

$$a_n = M_n - \frac{1}{8}\frac{(M_{n-1} - M_{n+1})^2}{(M_{n-1} - 2M_n + M_{n+1})} \tag{2.16}$$

2.8 Non-Stationary Signal Decomposition

The analysis of audio in signal processing is concerned with decomposing a signal into a set of elementary building blocks. The set of functions used, such as sinusoidal or wavelet functions, form a basis for the signal space. A basis is a linearly independent set of functions that 'spans' that signals space [137–140]. Linear independence means that you can not express one of these functions as a linear combination of the other functions in the space. Spanning the entire signal space means that any value within that space can be expressed as a unique linear combination of these functions. If both of these properties are true, then these functions form a basis for the signal space. A basis can be thought of as a set of building blocks with certain features which can be used to break down a complicated signal into a subset of simpler components. This is known as signal decomposition, or expansions after Denis Gabor suggested a signal could be expressed in both time and frequency and 'expanded' into a discrete set of guassian atoms [61].

Adaptive signal models can be expressed as:

$$x[n] = \sum_{i=1}^{I} \alpha_i g_i[n] \tag{2.17}$$

where a signal x can be decomposed into a weighted (α_i) linear combination of these expansion functions $g_i[n]$, where α_i are the expansion coefficients. In linear algebra terms α_i is referred to as a 'scalar', which is the value returned by performing the 'dot product' 4.2 of each atom $g_i[n]$ with x, resulting the a measurement of similarity between the signals. Adaptive signal models are designed to improve the accuracy of the model by incorporating more components in the expansion. The improvement to the model is usually measured by calculating the mean-square error. This iterative method of continuously improving the approximation of the signal model is known as a successive refinement framework [100, 141, 142].

Non-stationary sinusoidal models aim to improve the accuracy of a sinusoidal models parameter estimates by including estimates for amplitude and frequency change from within an analysis frame. Noise subspace methods are one approach for estimating the parameters of complex sinusoids [143, 144]. These techniques can be computationally expensive [145, 146], however a recent efficient implementation of the Multiple Signal Classification (MUSIC) algorithm has been presented in Fast Music [147].

The techniques used in this thesis for modelling non-stationary sinusoidal parameters are related to phase based methods such as Phase Distortion Analysis [9, 148], Time-Frequency Reassignment [149, 150] and the Derivatives method [131, 151–155].

2.8.1 Reassignment and Derivative Methods

The Derivatives method is another algorithm for improving frequency estimates from the DFT, but it requires an additional DFT of the signals derivative. "This is effectively a high pass filtering operation whose frequency dependent gain can be calculated. Therefore the difference in derivative (high pass filtered) and standard (non high pass filtered) DFT magnitudes can be used to produce an estimate of the frequency of the sinusoid" [136].

Reassignment is another method for estimating frequency deviation from the center of an analysis bin as well as using spectral data for estimating the time deviation from the center of an analysis frame. Reassignment was generalized by Auger and Flandrin in [149]. The STFT extracts information from a signal with a fixed frame and hop size, and returns a time-frequency distribution with estimates of energy at fixed time and frequency intervals. In [156, 157], reassignment is used to overcome the "localization and interference trade-off that is usually observed in classical Time-Frequency analysis".

Time Frequency Reassignment shifts the coefficients away from the center of an analysis frame, to the 'center of gravity of the windowed energy'. Reassignment is able to give estimates in time and frequency of where the energy is concentrated within a frame by making use of the phase spectrum [150], and the first order derivative of the analysis window. Reassignment does this by taking two additional DFTs, one with a time ramped window (analysis window multiplied by time), and the second with a frequency ramped window (the derivative of the analysis window) as shown in Figure 2.22

The estimate of the time deviation from the centre of a frame is given by:

$$-\frac{1}{F_s}\Re\left\{\frac{DFT_{\text{time ramped window}}}{DFT_{\text{standard window}}}\right\} \tag{2.18}$$

where Fs is the sampling rate of the signal.

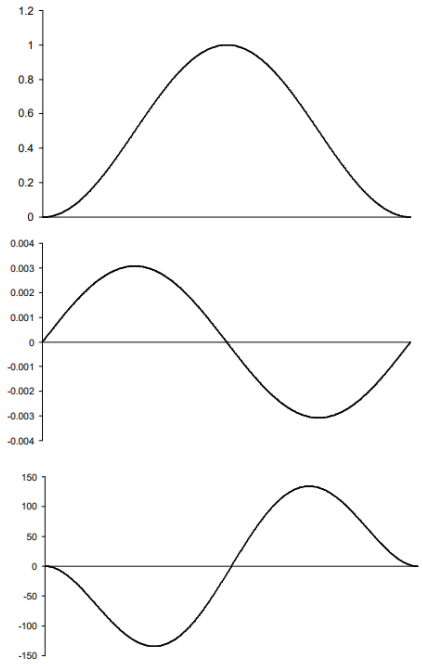


FIGURE 2.22: Hanning window (top), Frequency ramped window (middle), Time ramped window (bottom) [136]

Estimation of the deviation in frequency from the centre of an analysis bin is given by:

$$-B\Im\left\{\frac{DFT_{\text{frequency ramped window}}}{DFT_{\text{standard window}}}\right\} \tag{2.19}$$

where B is the bin width in Hz.

Reassignment [149, 150] and the Derivatives method [131, 152–155] have been shown to be theoretically equivalent [153, 154]. Such models assume exponential amplitude change within an analysis frame due to the model parameters occurring within the exponent. Having the amplitude expressed as an exponential with a polynomial argument is based on the fact that we perceive loudness on a logarithmic scale, and allows the signal to be expressed as a single exponential with a complex argument [158]. Expressing the signal as a complex exponential simplifies some of the equations and allows for the order of the polynomial to be increased, resulting in more complex non-monotonic amplitude modulations.

A signal with linear log-AM and linear FM is defined as:

$$s(t) = \exp(\underbrace{(\lambda_0 + \mu_0 t)}_{\lambda(t)=\log(a(t))} + j\underbrace{\left(\phi_0 + \omega_0 t + \frac{\psi_0}{2}t^2\right)}_{\phi(t)}) \tag{2.20}$$

where $\hat{\mu}$ is the amplitude modulation which is the derivative of λ (the log-amplitude), and ω_0 (the frequency), ψ_0 (the frequency modulation) are respectively, the first and second derivatives of ϕ (the phase). A comparison of these methods is carried out in [153] where where $s(t)$ is defined as the input signal, and $w(t)$ is defined as the window function over time t. S_w is a function of time and frequency expressed as:

$$S_w(t,\omega) = \exp(a(t,\omega) + j\phi(t,\omega)) \tag{2.21}$$

The derivative method is generalized to the non-stationary case in [154], which includes the generalised reassignment equations found in C, resulting in the following equations for non-stationary parameter estimation.

A generalised estimation of the local maximum (discrete) frequency $\hat{\omega}_0$ is given by:

$$\hat{\omega}_0 = \Im\left(\frac{S'_w}{S_w}(\omega_m)\right) \tag{2.22}$$

An estimate of amplitude modulation $\hat{\mu}_0$ given by:

$$\hat{\mu}_0 = \Re\left(\frac{S'_w}{S_w}(\hat{\omega}_0)\right) \tag{2.23}$$

And frequency modulation $\hat{\psi}_0$:

$$\hat{\psi}_0 = \Im\left(\frac{S''_w}{S_w}(\hat{\omega}_0)\right) - 2\hat{\mu}_0\hat{\omega}_0 \tag{2.24}$$

Having estimated values for $\hat{\omega}_0$, $\hat{\mu}_0$, and $\hat{\psi}_0$, the initial amplitude $\hat{a}_0$ and and initial phase $\hat{\phi}_0$ of the signal are then given by:

$$\hat{a}_0 = \left|\frac{S_w(\hat{\omega}_0)}{\Gamma_w\left(0,\hat{\mu}_0,\hat{\psi}_0\right)}\right| \tag{2.25}$$

$$\hat{\phi}_0 = \angle\left(\frac{S_w(\hat{\omega}_0)}{\Gamma_w\left(0,\hat{\mu}_0,\hat{\psi}_0\right)}\right) \tag{2.26}$$

2.9 Single-Frame Discrimination of Nonstationary Sinusoids

A single-frame non-stationary sinusoid is described in [9, 159–161] described by

$$s(t) = A(t) \sin\left(\int_{\tau=0}^{\tau=t} 2\pi f(\tau) d\tau + \phi\right) \tag{2.27}$$

where, for a single frame, $A(t)$ is a function describing the amplitude trajectory and $f(t)$ is a linear function describing the frequency trajectory over time t, and ϕ is the phase of the sinusoid at the start of the frame. In [161] the amplitude is modelled as a piecewise exponential function, the frequency is piecewise linear and the phase is piecewise quadratic. Systems for the single frame estimation of the parameters of non-stationary sinusoids include [89, 162–164] as well as the reassignment and derivatives methods discussed in the previous Section 2.8.1. The method adopted in [9, 159–161] for estimating non-stationary sinusoidal parameters from a single analysis frame, uses phase distortion (PD) analysis [164].

2.9.1 Phase Distortion (PD)

Phase distortion is a measurement of the 'phase shift' [165] with respect to the flat phase response across a sinusoidal peak of a stationary sinusoid, taken from a zero phase padded FFT. Figures 2.24 - 2.27 show the effect amplitude and frequency change has on the phase of a zero phase windowed sinusoid, compared to a stable sinusoid in Figure 2.23. Amplitude and frequency non-stationarities produces changes in the window shape in the Fourier domain, compared to that of a stationary sinusoid. In [9, 159–161], intra-frame linear frequency Δf_n, and exponential amplitude change ΔA_n are estimated from measuring the phase differences either side of a zero-padded spectral peak. "The relationship between these measures and the actual amplitude change (dB per frame) and frequency change (bins per frame) is dependent upon the window type and is empirically determined [159]." This is described by:

$$\Delta A_n = g\left(\phi_{n+1} - \phi_{n-1}\right) \tag{2.28}$$

$$\Delta f_n = h\left(\phi_{n+1} + \phi_{n-1}\right) \tag{2.29}$$

where n is the index of a magnitude spectrum peak, and ϕ is phase. The functions $g(x)$ and $h(x)$ relate the phase difference across a sinusoidal peak to the intra-frame parameter change.

The procedure for deriving the analytical equations for exponential amplitude change estimates from the phase difference is explained in [10] where intra-frame amplitude change can be interpreted as a window function with the amplitude change applied to it. The Fourier Transform of a modified window function was taken as a function of frequency, and the phase response and first derivative were examined for exponential amplitude change applied to a Hanning window as a function of time t, and is given by 3.9

Positive amplitude change results in a negative phase slope while negative amplitude change results in a positive phase slope. Conversely, positive frequency change results in a concaved phase curve across a sinusoidal peak, while negative frequency change results in a convexed phase curve.

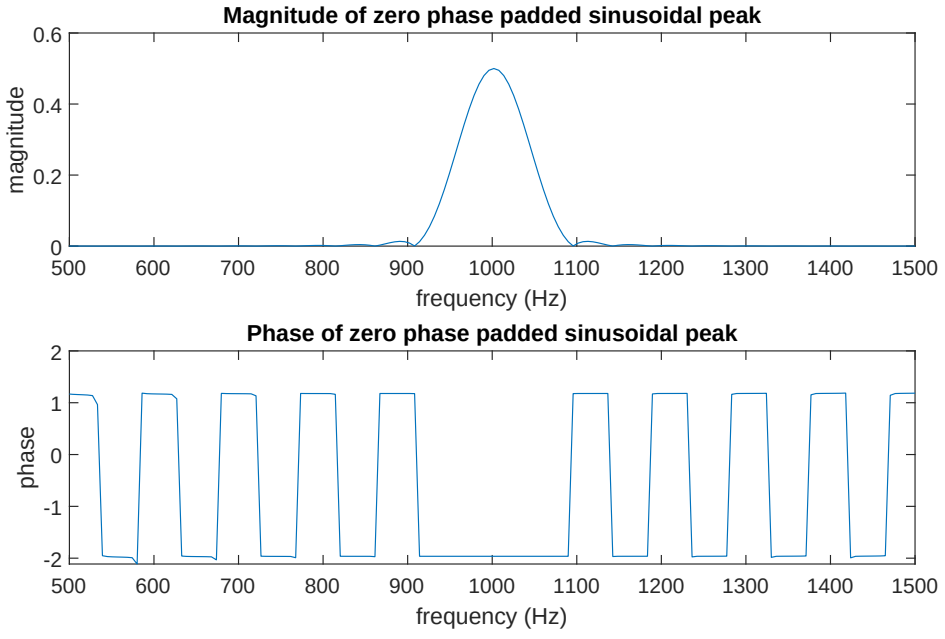


FIGURE 2.23: Phase Distortion from 1 kHz sinusoid (@48 kHz) with frequency change of -400 Hz

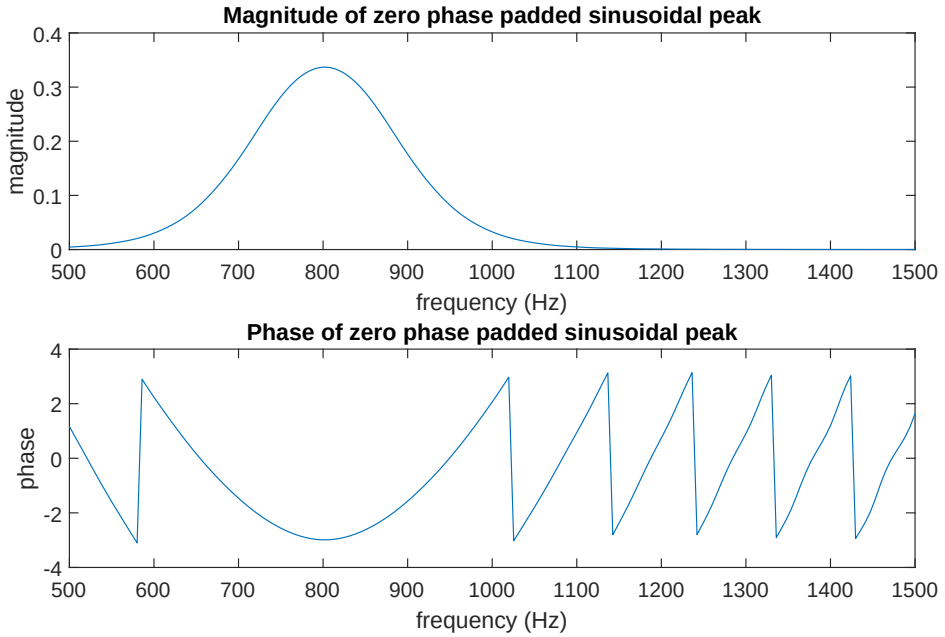


FIGURE 2.24: Phase Distortion from 1 kHz sinusoid (@48 kHz) with frequency change of -400 Hz

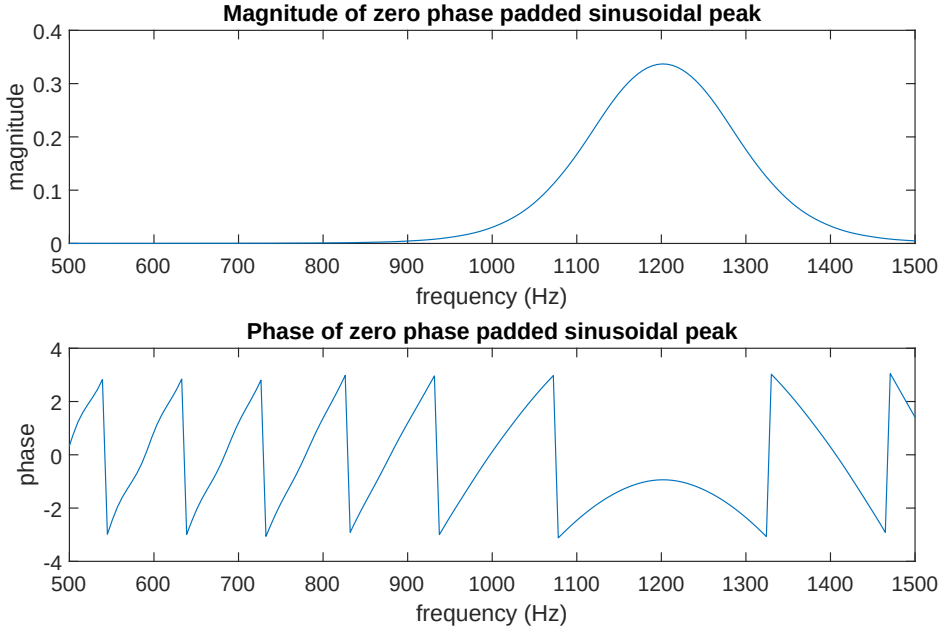


FIGURE 2.25: Phase Distortion from 1 kHz sinusoid (@48 kHz) with frequency change of 400 Hz

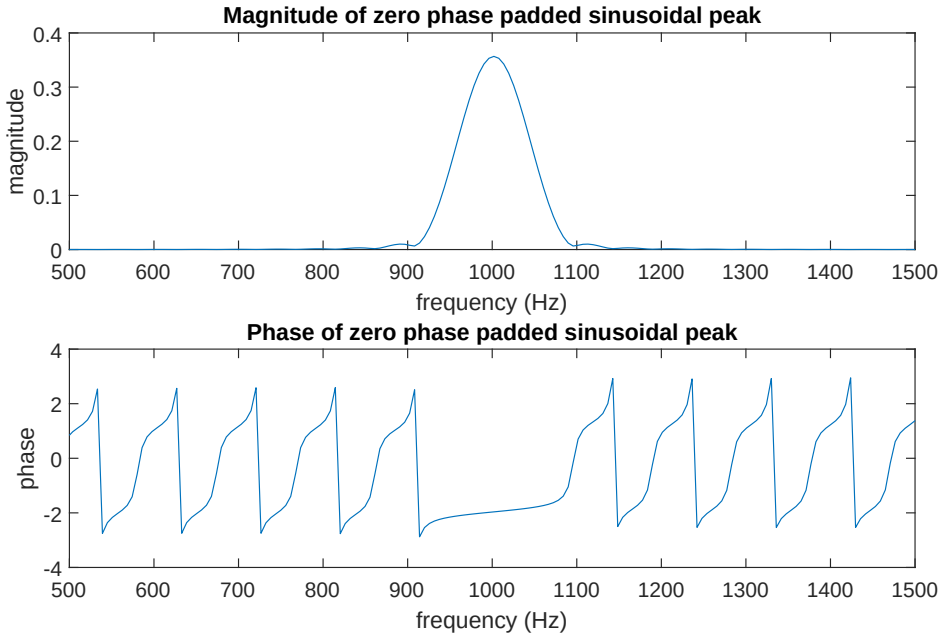


FIGURE 2.26: Phase Distortion from 1 kHz sinusoid (@48 kHz) with amplitude change of -6 dB

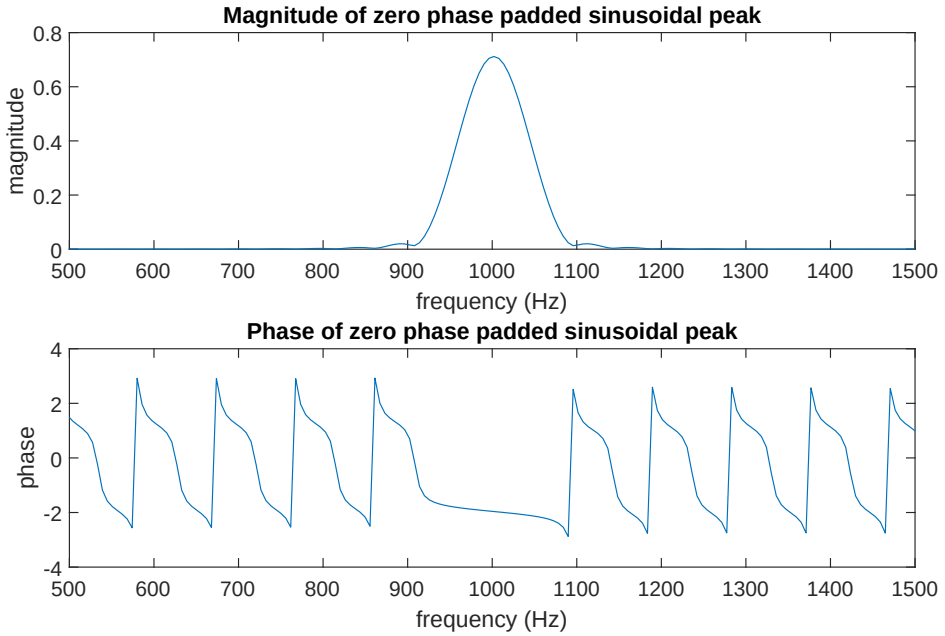


FIGURE 2.27: Phase Distortion from 1 kHz sinusoid (@48 kHz) with amplitude change of 6 dB

The derivation of the phase difference measure for non-causal exponential intra-frame amplitude change with a rectangular window applied to an amplitude-stationary sinusoid is given in C.

2.10 Wavelets and Filter Banks

The aim of spectral modelling and that of time-frequency analysis is to find out what frequencies occur at specific moments of time in the signal and how they evolve over time. The STFT has a better time-frequency representation to the Fourier transform which contains no temporal information. The problem with the STFT is that although you can get some temporal information from analysing small sections of the signal at a time, you still don't know how the frequency and magnitude of the sinusoidal components change over time within that window. The fixed window length also results in a trade-off between a good time resolution and a good frequency resolution. Longer windows increase the frequency resolution, but at the expense of decreasing the time resolution and vice versa.

With the STFT the time-frequency resolution is fixed and is set by the length of the analysis window. This determines the resolution in both time and frequency, resulting in a time-frequency grid which is evenly spaced. The wavelet transform uses a basis function of a 'small waves' with certain properties. These 'small waves' act as filters which when stretched and dilated change the bandwidth of the filter. The scaling in length of these filters allows for the signal to be separated into different time-frequency bands where each component can then be studied with a resolution matched to its scale. The notion of scale is the fundamental idea behind wavelets, and makes them ideal for approximating signals with sharp discontinuities.

The Continuous Wavelet Transform (CWT) is defined as:

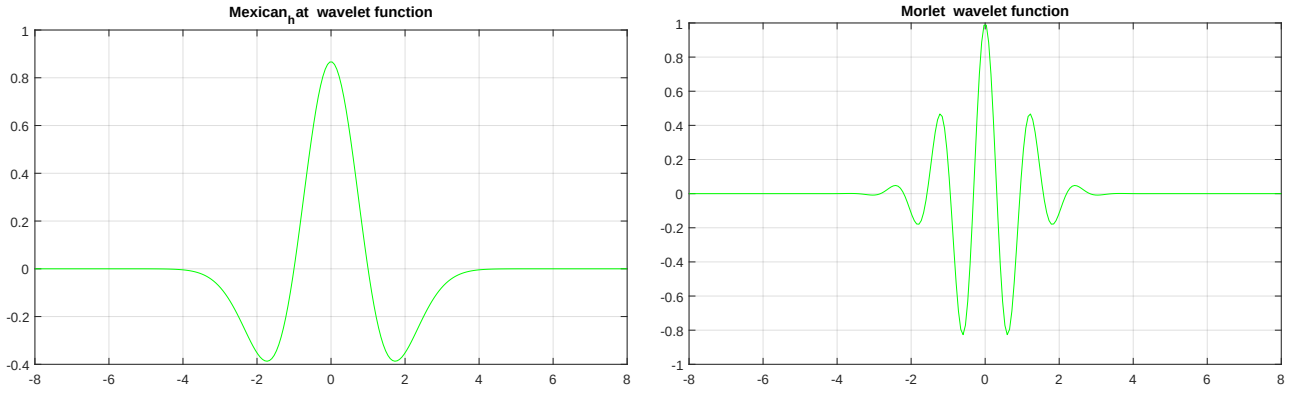
$$CWT_x(\tau, s) = \frac{1}{\sqrt{|a|}} \int x(t) h\left(\frac{t-\tau}{s}\right) dt \tag{2.30}$$

where the wavelet function is defined as [166]:

$$h\left(\frac{t-\tau}{s}\right) \tag{2.31}$$

with τ being the translation in time and S determining the scale, or size, of the wavelet.

In wavelet analysis the scale of the wavelet function is integral to the resolution in time and frequency of the analysis of the signal. Wavelets are short wave like functions with zero mean and a finite duration. For example the Mexican had wavelet in Figure 2.28a



(A) Mexican hat (B) Morlet

FIGURE 2.28: Comparison Mexican Hat and Morlet Wavelets

In order to be classified as a wavelet a function must satisfy certain mathematical criteria [167]

A Wavelet must have finite energy

$$E = \int_{-\infty}^{+\infty} |\psi(t)|^2 dt < \infty \tag{2.32}$$

The wavelet must also have zero mean, in other words no zero frequency components.

The wavelet transform is based on the translation and dilation of the small wave function. The scale of the wavelet changes its length and so by expanding or contracting the length of the wavelet, different signal characteristics can be extracted. A long window or wavelet function is good at analysing low frequency components while a short window or wavelet function is good at analysing high frequency components. This ability to scale the wavelet function allows for a multi-resolution analysis of the data whereby high frequency components can be measured with wavelets which have been compressed, and low frequency components can be measured when the wavelet function is expanded. Thus the wavelet transform can extract both high frequency details with a good time resolution and low frequency details with a lower time resolution, but since low frequency components evolve slowly there is less of a need for a fine time resolution compared to high frequency components which change quickly and discontinuities which have short duration's and so require a better resolution in time from the analysis process. Thus sharp discontinuities can be detected with a good time resolution by using a compacted scale of the wavelet function.

2.10.1 Discrete Wavelet Transform (DWT)

The discrete implementation is given by the Discrete Wavelet Transform (DWT):

$$\mathrm{DWT}(u,s) = \frac{1}{\sqrt{s}} \sum_{n} x[n] \cdot \psi \times \left[\frac{n-u}{s} \right] \tag{2.33}$$

Depending on the basis wavelet used, a practical implementation is via convolution/filtering and down-sampling[166]. An example of this can be found in dyadic wavelets, where at each stage the time domain signal is downsampled by a factor of 2 resulting in the Dyadic wavelet coefficient structure, as the filter bandwidth is effectively halves at each level as depicted below:

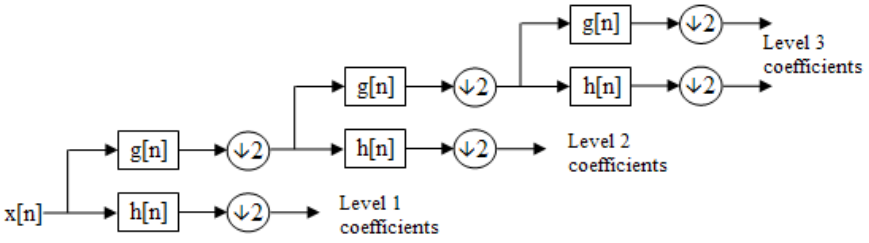


FIGURE 2.29: DWT Filter Bank Implementation

The signal is analyzed in a cascade of frequency bands each containing half the samples of previous bands. The signal is decomposed into approximation and detail coefficients by applying a high pass and low pass filter to the input signal x. The approximation is then successively down-sampled and approximated again using the same filtering steps. This process can be expressed by:

$$a_{j+1}[n] = \sum_{k=0}^{m-1} a_j[2n-k+m-1]g[k] \tag{2.34}$$

$$d_{j+1}[n] = \sum_{k=0}^{m-1} a_j[2n-k+m-1]h[k] \tag{2.35}$$

where a_j and d_j are the approximation and detail coefficients at decomposition level j, $a_1 = x$, h and g are the high-pass and low-pass wavelet filter coefficients with a length of m.

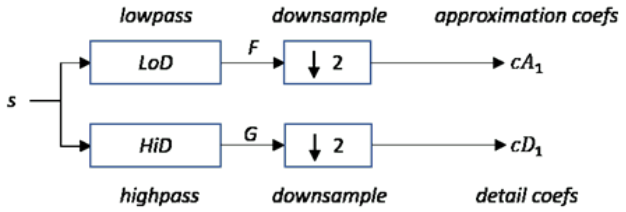


FIGURE 2.30: Forward DWT

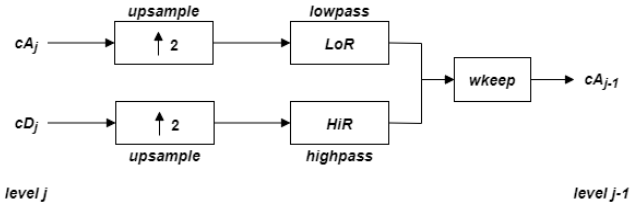


FIGURE 2.31: Inverse DWT

The inverse DWT is performed in a similar manner as the forward DWT using the convolution operation, however the up-sampling, convolution, and summation of the outputs into the above levels reconstructed approximation coefficients, are done simultaneously through a sub-sampling process described by:

$$\begin{aligned} a_j[n] = &\sum_{k=0}^{\left\lfloor \frac{m-1+(n \bmod 2)}{2} \right\rfloor} a_{j+1}\left[\left\lfloor \frac{n}{2} \right\rfloor - k\right] \tilde{g}[2k + (n \bmod 2)] \\ &+ \sum_{k=0}^{\left\lfloor \frac{m-1+(n \bmod 2)}{2} \right\rfloor} d_{j+1}\left[\left\lfloor \frac{n}{2} \right\rfloor - k\right] \tilde{h}[2k + (n \bmod 2)] \end{aligned} \tag{2.36}$$

where $\tilde{h}$ and $\tilde{g}$ are the reconstruction wavelet filters, and mod is the "modulo" binary operation.

Fourier analysis compares a set of evenly spaced harmonic components with the input signal, deriving the coefficients of magnitude and phase, wavelets work in a very similar way by convolving the input signal with wavelets of different sizes and at different time intervals. The process of applying shifted and scaled versions of the mother wavelet to the input signal results in a filtered time-domain signal which is separated into a dyadic multi-resolution time-frequency grid as shown in Figures 2.32 and 2.33.

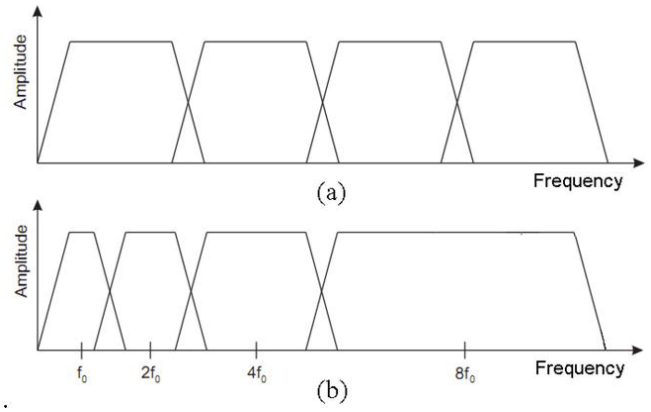


FIGURE 2.32: Comparison of (a) the STFT uniform frequency coverage to (b) the logarithmic coverage of the DWT.

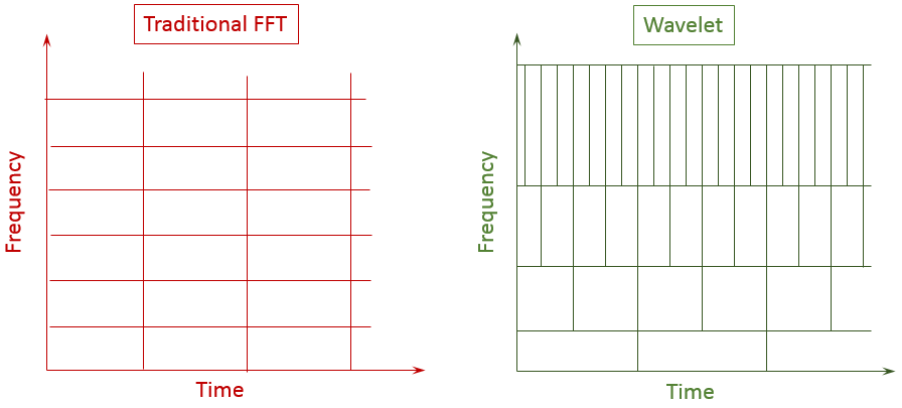


FIGURE 2.33: Wavelet dyadic multi-resolution time-frequency grid

2.11 Summary

In this chapter the musical signals of interest and how their amplitude envelopes are most commonly shaped has been discussed. Linear and exponential amplitude envelopes are commonly used for manipulating the amplitude of kick and bass. Chapter 3 investigates modelling both linear and exponential amplitude from single frame phase based analysis methods, as well as discriminating between the two curve types. How sound is represented in both time and frequency has been presented, leading on to achieving better time-frequency representation by using non-stationary estimation methods such as Time-Frequency Reassignment and Phase Distortion, or by using tools such as the DWT instead of traditional DFT analysis. Chapter 5 investigates modelling residual signals after sinusoidal modelling; using the undecimated implementation of the DWT in a segmented framework, following an atomic decomposition presented in 4.

Chapter 3

Non-Stationary Modelling of Amplitude Envelopes

> Writing a tune is like sculpting. You get four or five notes, you take one out and move one around, and you do a bit more and eventually, as the sculptor says, "In that rock there is a statue, we have to go find it."
>
> *John Williams*

In this chapter an analysis method for distinguishing between linear and exponential amplitude change within a single analysis frame is presented. The base signals of the model used in this thesis are sinusoidal components with stationary or non-stationary amplitude (monotonic linear or exponential) and frequency (monotonic linear). Other signal components such as noise are treated separately, while transients provide an interesting choice of being treated as a separate signal component all together, or as a sum of many short lasting sinusoidal components with time varying amplitude and frequency trajectories. The chapter sets out to explore the detection, modeling and discerning between sinusoidal components with monotonic exponential or linear amplitude change.

In Section 3.1 the main motivations behind the modelling of different types of amplitude envelope shapes is discussed. Section 3.2 presents the new method for estimating the amount of linear amplitude change present within a single frame (when a linear amplitude curve is being applied) before a method of distinguishing between linear and exponential amplitude change is discussed in Section 3.3.

Finally, Section 3.4 presents the evaluation of the discriminator, and the method used for the estimation of amplitude change within a single frame are then compared with existing phase-based methods, both in terms of their computational cost and effectiveness.

3.1 Motivation

Spectral modelling and time frequency analysis has been covered in detail in Chapter 2. Overlapping frames and single frame discrimination of sinusoidal components in an additive synthesis system have been introduced along with methods for approximating the quasi-stationary sinusoidal properties of instantaneous amplitude and frequency. The model adopted by this thesis and the other models used for comparison are based on non-stationary sinusoidal parameter estimation, the motivation for this is that stationary signals in music are not of great interest. Music is interesting and pleasant due to the mixing and modulation of frequencies. Non-stationary models and the methods of parameter estimation employed make assumptions about the underlying properties of the signal being analysed. The mean amplitude (quasi-instantaneous) of a signal component is taken as the average amount of energy over the duration of the analysis frame and is assigned to the mid point of the frame. The mean frequency (quasi-instantaneous) is measured as the first derivative of the phase and is also assigned to the mid point of the frame.

In general frequency modulation is modelled as a linear rate of change, calculated as the second derivative of the phase with regards to frequency. Amplitude modulations are either modelled exponentially as in the case of exponentially dampened sinusoidal models such as those using Reassignment [149, 150] or the Derivatives methods [151, 152]. Some Quasi-Harmonic models such as the adaptive Quasi-Harmonic Model (aQHM) [**?**] and the extended adaptive Quasi-Harmonic Model (eaQHM) [168] estimate amplitude modulations as a linear rate of change. Traditional Spectral Modelling Synthesis (SMS) approaches [71] also model amplitude change over time by piece-wise linear envelopes, tracking the amplitude (quasi-instantaneous) between frames.

Smith in [26] identifies the importance of exponentials and the following key properties regarding exponential growth and decay. Exponential decay occurs naturally when an amount is reducing at a rate proportional to its current value. Smith goes on to describe all momentary excited oscillations that are linear and time-invariant as examples of signals which decay exponentially. Musical examples of these include the vibrations of tuning forks and plucked string instruments. Another example of exponential decay can be found in how the reverberant sound of a room decreases over time.

Exponential growth on the other hand is when a signal is growing at a rate proportional to the current value which can cause a signal to become unstable without limiting the amount of growth. Exponential amplitude decay is often adopted by Exponentially Damped Sinusoidal (EDS) models [169–175] due to the observation of its predominance in many naturally decaying sinusoidal systems. Human perception of loudness works on a logarithmic scale, as such signals which decay exponentially sound more natural. Signals which have a linear decay can sound less natural as our ears perceive the middle part of the linear envelopes decay as being too loud. Analog synthesises use capacitors for controlling the rate at which the synthesised signal increases and decreases in volume. The attack shape from the capacitor storing charge takes a logarithmic shape, while the the discharge controlling the decay of the sound falls at an exponential rate which attributes to the pleasant sound associated with analog synthesizers.

However, with music production and art, there are no fixed rules and a music producer can sculpt the audio with a range of envelope shapes, selecting what sounds best to them. Electronic music or audio edited on a Digital Audio Workstation (DAW) can have the amplitude envelope shaped by fading, or by use of an Attack, Decay, Sustain, Release (ADSR) envelope which does not always conform to applying a single type of envelope curve. A music producer has many tools for sculpting, manipulating and fine tuning the audio used for creating kick and bass lines. One of the key tools are amplitude envelopes which have great control over the punch of transient sounds such as the kick, and also how the sound evolves over time and fades out. Linear or logarithmic attacks are generally preferred for attack portions of a sound while linear or exponential envelopes are preferred for controlling the shape of how a sound decays over time.

Spectral modelling systems employing overlap-add (OLA) have been presented in Section 2.4.5. Overlapping analysis and synthesis frames smooths out discontinuities between frame boundaries and can give a finer resolution when using short frame lengths, small hop sizes, or a combination of both. The model adopted in this thesis is based on previous work on single frame (non-overlapping) analysis/synthesis methods [9, 10, 159, 161]. High accuracy frame-by-frame non-stationary sinusoidal modelling requires accurate estimates of amplitude change. Incorrect estimates of amplitude change can result in audible discontinuities a frame boundaries. Phase based methods which presume exponential amplitude change result in biased and incorrect estimates in the case of linear or logarithmic amplitude change. These errors are more noticeable in a single-frame system which does not apply widowing and overlap frames. The estimation of linear amplitude change and the discrimination between exponential and linear amplitude change improves the accuracy of the single-frame models, reducing audible discontinuities at frame boundaries.

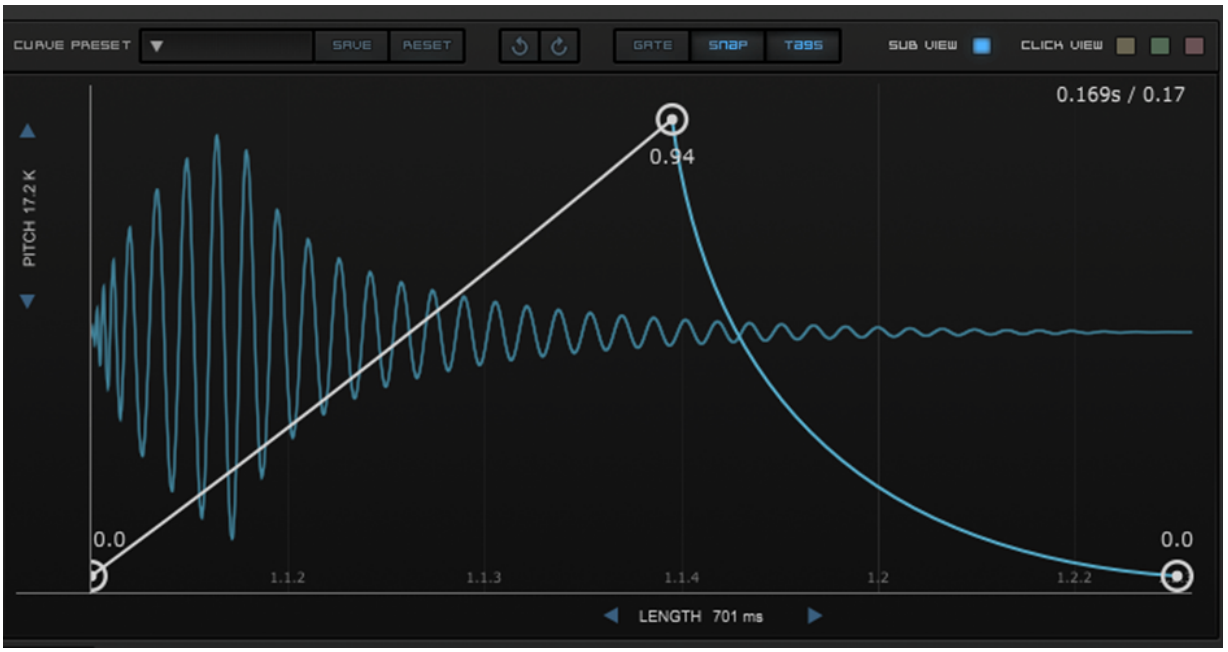


FIGURE 3.1: Linear and exponential amplitude curves employed in Kick2 VST highlighting the options a music producer has for shaping audio [8]

Figure 3.1 displays an example of a kick drum with a linear attack and an exponential release. The discrimination between linear and exponential amplitude change in such a case would result in more accurate estimates and an improved signal model.

3.2 Representation of linear and exponential amplitude change

Sinusoidal models provide a description of signals in terms of sinusoidal basis functions. Signals are modelled as a sum of sinusoids with various amplitudes and harmonically related frequencies. The discrete Fourier Transform (DFT) yields a compact representation of a stationary sinusoid when the frequency of the sinusoid is harmonically related to the analysis frame length, but that is not the case when describing sinusoids with time-varying amplitudes and non-harmonically related frequencies. The work described in this Chapter aims to enable the identification of intra-frame amplitude change of individual signal components and the detection of the amplitude envelope type, with the application in a quasi-real-time time spectral modelling framework. Sinusoids are analysed and re-synthesized from a single analysis frame which requires highly accurate estimates of amplitude and frequency, as well as how they change within a frame to minimise discontinuities at frame boundaries.

In a non-stationary sinusoidal model, it is often convenient to represent sinusoids as complex exponentials with amplitude, frequency and their modulations represented by polynomial arguments, as this allows the signal to be combined into a single exponential with a complex argument, and expressing a sinusoidal function in complex exponential form considerably simplifies many operations, particularly the solution of differential equations. The order of the polynomial dictates the signal model which can be distinguished [158]. As mentioned, the system intended to be used in this Chapter is a quasi-real-time system with the underlying assumption that a signal evolves slowly enough to be accurately modelled by monotonic amplitude change. Reassignment [149, 150] and the Derivatives method [151, 152] estimate amplitude change as the first derivative of the phase with respect to frequency, and the models have been shown to be theoretically equivalent [153], although the practical implementation constraints are different. Such models assume exponential amplitude change within an analysis frame due to the model parameters occurring within the exponent.

3.3 Defining equivalent amplitude curves

A sinusoid modelled by a first order amplitude and second order frequency modulations is described by:

$$s(t) = A(t)\sin\left(\phi + 2\pi(ft + \frac{\Delta f t^2}{2T})\right) \tag{3.1}$$

Where $A(t)$ describes either a linear or exponential time varying amplitude envelope, ϕ is the mid point phase, f is the instantaneous frequency and Δf is the linear intra-frame frequency change.

Given that $A(t)$ can be either an exponential or linear function over time, centered at time equals zero, there is a requirement to be able to express both functions relative to one another. Nepers (C.14) are used in the derivation of the equations in this Chapter because the use of the natural logarithm simplifies the solving of the differential equations used for solving the amplitude change within a frame due to the derivative of the natural log (at 1) being equal to one (C.15).

An exponential amplitude curve with unity gain as a function of time t (centered at time = 0) is given by:

$$\mathrm{A(t)_{exponential}} = \mathrm{e}^{\alpha \mathrm{t}}, |\mathrm{t}| \leq \frac{1}{2} \tag{3.2}$$

where α is the intra-frame amplitude change given in Nepers (Np)

$$\alpha = \frac{\Delta A \ln 10}{20} \tag{3.3}$$

ΔA is the amplitude change within a frame in decibels (dB) given by:

$$\Delta A = 20\log_{10} \frac{A(t_{\text{end}})}{A(t_{\text{start}})} \tag{3.4}$$

Where the intra-frame amplitude change is given by the ratio of the amplitude as the start and end of an analysis frame (A(t_{start}) and A(t_{end})). In the context of this thesis, amplitude change is derived from estimates of the group delay (time reassignment / center of gravity) given from the first derivative of the phase with respect to frequency approximated by its first-order difference ($\Phi_{n+1} - \Phi_{n-1}$) as described in Section 2.9.1. However, unlike PDA and other single frame phased based methods for estimating amplitude modulation on the assumption of exponential amplitude change, the methods used in this thesis extends the underlying amplitude function to be either exponential or linear. This chapter presents the equations for calculating the amount of amplitude change within a frame in relation to the phase difference measurements around a sinusoidal peak.

An example of an exponential curve $\left[e^{\alpha t}, \{t, -0.5, 0.5\}\right]$ centered around zero with an amplitude change of $\alpha = 5$ Nepers is shown in 3.2

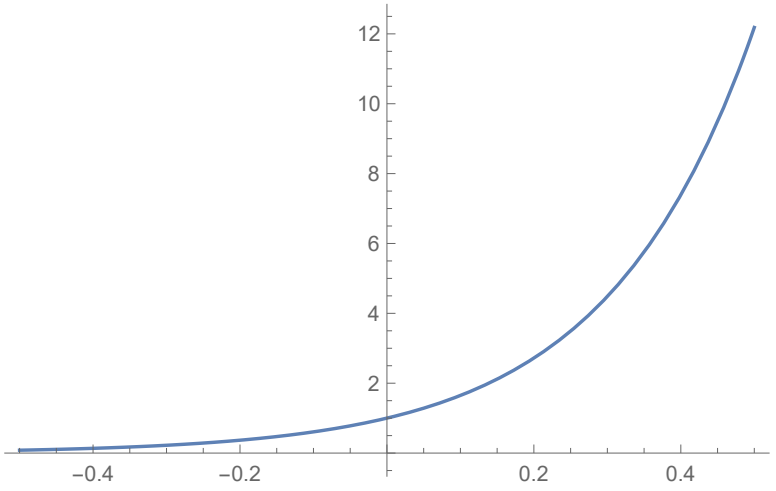


FIGURE 3.2: Exponential amplitude change $\alpha = 5$ in Nepers over {t,-0.5,0.5}

Nepers uses the base e which is more convenient for the integration formulas used in this chapter for deriving the equations used for estimating amplitude change within a frame. To compare a linear amplitude curve with an exponential curve, there is a requirement to express it in terms of $A(t)_{\text{exponential}}$ with regards to ΔA. This enables the derivation of the phase distortion equations used in this chapter to be calculated using equal amplitude values. Keeping in mind that there is a requirement to discern between linear and exponential amplitude curves, and the Fourier transform is perfectly inevitable, meaning that information about non-stationary is embedded within the phase and magnitude information, the section regarding envelope type discrimination will require linear and exponential amplitude curves with equal phase distortion measurements in order for the magnitude spectrum to be compared for differences.

Given the equation for a linear curve to produce the same start and end values as an exponential amplitude curve allows for the solving of these equations to give us the amount of linear amplitude change required to give the same reassignment offset / phase difference value as an exponential amplitude change.

Given the amount of amplitude change required for a linear ramp to give the same shift in energy as an exponential curve with its (different) amount of amplitude change, we can therefore examine the magnitude spectrum to see where the differences lie between an exponential and linear curve, because the phase difference is equal, the only other difference is encoded in the shape of the magnitude spectrum.

The linear equivalent over the time interval of $\{t, 0, 1\}$ with $\alpha = 5$ in Nepers is given by 3.5 and shown in Figure 3.3

$$\left(e^{\alpha/2} - e^{-\frac{\alpha}{2}}\right) t + e^{-\frac{\alpha}{2}}, \{t, 0, 1\} \tag{3.5}$$

To calculate the linear equivalent of 3.2 centered around zero, 3.5 needs to be recalculated by shifting it to the interval $\{t, -0.5, 0.5\}$.

$$\left(e^{\alpha/2} - e^{-\frac{\alpha}{2}}\right) t + \frac{1}{2}\left(e^{\alpha/2} - e^{-\frac{\alpha}{2}}\right) + e^{-\frac{\alpha}{2}}, \{t, -0.5, 0.5\} \tag{3.6}$$

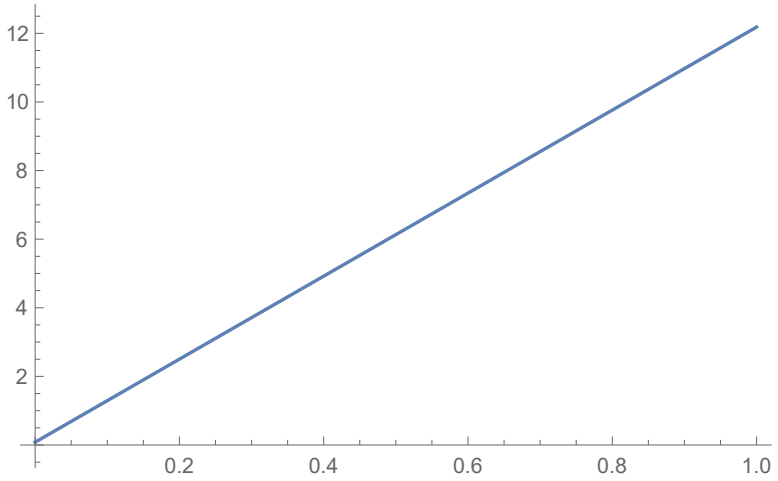


FIGURE 3.3: Linear amplitude change $\alpha = 5$ in Nepers over {t,0,1}

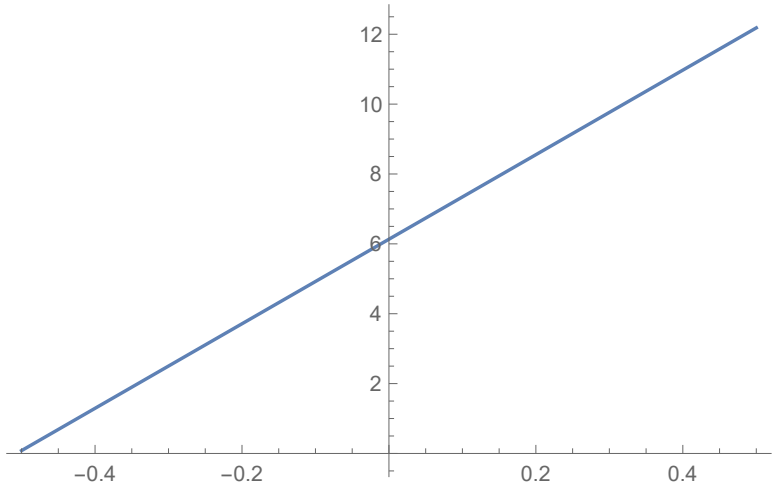


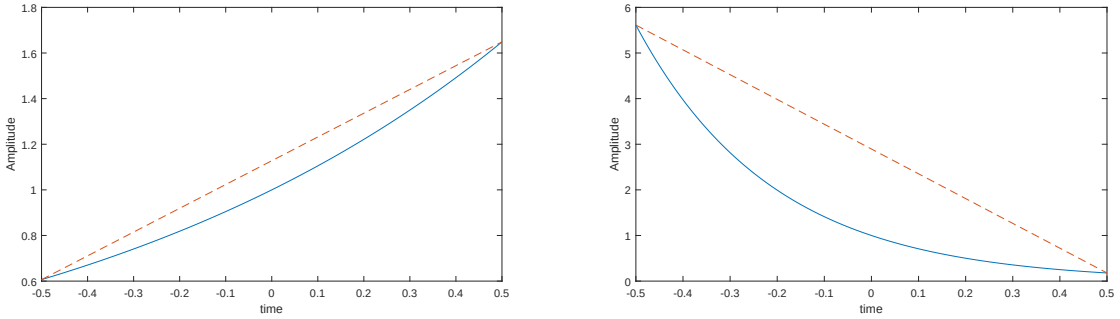
FIGURE 3.4: Linear amplitude change $\alpha = 5$ in Nepers over {t,-0.5,0.5}

Figure 3.4 clearly shows a linear curve centered around zero with the same amount of amplitude change as that shown in Figure 3.2.

The equivalent linear curve which produces the same intra-frame amplitude change with the same start and end values as $A(t)_{\text{exponential}}$ can be simplified from 3.6 to:

$$A(t)_{\text{linear}} = \frac{e^{\frac{\alpha}{2}} + e^{\frac{-\alpha}{2}}}{2} + t(e^{\frac{\alpha}{2}} - e^{\frac{-\alpha}{2}}), |t| \leq \frac{1}{2} \tag{3.7}$$

Figure 3.5 plots the linear and exponential amplitude curves given in (3.2) and (3.7) with different values of α.



(A) Amplitude Change = 9 dB (B) Amplitude Change = -30 dB

FIGURE 3.5: Linear and exponential amplitude curves with equivalent amplitude changes, $|t| \leq \frac{1}{2}$ (a) $\alpha = 1$, (b) $\alpha = -3.45$

3.4 Parameter estimation from reassignment and phase difference

Reassignment extracts information from the phase spectrum by making use of the first derivative of the analysis window, this enables it to reassign the time and frequency atoms (Linearly spaced Fourier bin information relating to the average magnitude and phase over the duration of the analysis window) from the geometrical centres of the analysis window and frequency bin to the centre of mass of where the atom's energy is concentrated. Reassignment gives a measure of the deviation of an atom's energy from the geometric centres by applying two additional DFT's, One for the time deviation and another for the frequency deviation. The time deviation is given by using a time ramped version of the analysis window used for taking the original DFT. The DFT provides the average amount of energy present at at the center of the time-domain frame, as well as the center of the frequency band, where the resolution of the frequency bin is determined by the frame size.

Where a peak in a zero-padded and zero-phase DFT analysis is due to a stationary sinusoid, the phase across the main lobe is constant. Where there is amplitude or frequency non-stationarity (or both) the phase is no longer flat and phase distortion analysis (PDA) independently estimates exponential amplitude and linear frequency change from this [176]. Reassignment distortion analysis (RDA) uses a similar method to estimate the same parameters [149, 159]. In fact, it can be shown that the time-reassignment measure (as described in [10]) is equivalent to the first derivative of the phase at a peak, but whereas time reassignment assumes displacement, from the centre of a frame in samples, PDA assumes an underlying exponential amplitude function and estimates the amplitude change across the frame. The time offset value from the centre of a single analysis frame for an impulsive component is estimated by the reassignment method as:

$$t_{k,\text{reassignment}} = \Re(\frac{X_{th,k}}{X_{h,k}}) \tag{3.8}$$

where $X_{h,k}$ and $X_{th,k}$ are the kth bin of the DFT of the input sequence weighted by the window function h and the time ramped window th. This gives one the offset of the center of gravity for the spectral bands and reassignes the time information in the audio frame to the point at which the energy is most evenly distributed. this reassignment of the time offset provide a position in time where the amplitude estimate is concentrated. The procedure for deriving the analytical equations for exponential phase difference is explained in [10] where intra-frame amplitude change can be interpreted as a change in the window (Hann) function applied to a signal with a stationary amplitude.

As explained in [10] intra-frame amplitude change can be interpreted as a window function with the amplitude change applied to it. Exponential amplitude change applied to a Hann window as a function of time t is given by 3.9

$$w(t)_{\text{exponential}} = e^{at}(\frac{1}{2} + cos(2\pi t)/2), |t| \leq \frac{1}{2} \tag{3.9}$$

More details of the analytical derivation of the exponential amplitude estimator can be found in [10] where the Fourier Transform of the modified window function is taken as a function of frequency, and the phase response and first derivative are examined. 3.10 shows the first derivative of the phase with respect to frequency for exponential amplitude change.

$$\frac{\angle X_{th,k+1} - \angle X_{th,k-1}}{2} \approx \frac{2}{\alpha} + \frac{4\alpha}{\alpha^2 4\pi^2} - \coth(\frac{a}{2})) \tag{3.10}$$

3.5 Deriving Linear Amplitude

Following similar steps as above linear intra-frame amplitude change can be interpreted as a linear change in the window function applied to a stationary signal. The full proof is available in Section D.1.2. The linear ramped Hann window is given by 3.11

$$W(t)_{\text{linear}} = (\frac{1}{2} + \frac{cos(2\pi t)}{2})(\frac{e^{\frac{\alpha}{2}} + e^{\frac{-\alpha}{2}}}{2}) + t(e^{\frac{\alpha}{2}} - e^{\frac{-\alpha}{2}})), |t| \leq \frac{1}{2} \tag{3.11}$$

The Fourier transform of the linear ramped Hann window from 3.11 applied to a stationary signal as a function of frequency; represented by the complex exponential $e^{-j2\pi f}$, is given by 3.12 which simplifies to 3.13.

$$W(f)_{\text{linear}} = \int_{-\frac{1}{2}}^{\frac{1}{2}} (\frac{1}{2} + \frac{cos(2\pi t)}{2})(\frac{e^{\frac{\alpha}{2}} + e^{\frac{-\alpha}{2}}}{2}) + t(e^{\frac{\alpha}{2}} - e^{\frac{-\alpha}{2}}))) e^{-j2\pi f} dt \tag{3.12}$$

$$= \frac{\begin{array}{r} e^{-\frac{\alpha}{2}-jf\pi}(-1+3f^2+e^{\alpha+2jf\pi}(-1+3f^2) \\ +e^{\alpha}(1-3f^2+2jf\pi-2jf^3\pi) \end{array} + e^{2jf\pi}(1-3f^2-2jf\pi+2jf^3\pi))}{8(-1+f)^2(1+f)^2\pi^2} \tag{3.13}$$

The first derivative of $f = 0$ at the limit is given by 3.14 which reduces to 3.15

$$\begin{aligned} \frac{d(\arg(W))}{df}|\lim_{f\to 0} = -\Big(\Big(\left(-1+e^{2a}\right)\pi\left(-1-3f^4+2f^2\left(-1+f^2\right)^2\pi^2+\left(1+3f^4\right)\cos[2f\pi]\right)\Big)/ \\ \Big(-2e^a\cos[2f\pi]\left(-1+f^2\left(6-9f^2+2\left(-1+f^2\right)^2\pi^2\right)+\left(1-3f^2\right)^2\cosh[a]\right)+ \\ 2e^a\left(-\left(1-3f^2\right)^2+\left(\left(1-3f^2\right)^2+2f^2\left(-1+f^2\right)^2\pi^2\right)\cosh[a]\right)- \\ 2\left(-1+e^a\right)^2\left(f-4f^3+3f^5\right)\pi\sin[2f\pi]\Big)\Big) \end{aligned} \tag{3.14}$$

$$= \frac{(-1+e^{\alpha})(-6+\pi^2)}{3(1+e^{\alpha})\pi} \tag{3.15}$$

Which gives us a the equation for calculating phase difference values in regards to linear amplitude change, this is used to create a lookup table which when searched through and the closest values interpolated gives the amplitude change estimate referred to from here as the phase difference estimator.

The equation for exponential amplitude change from [10] is given by:

$$\left.\frac{d(\arg(W))}{df}\right|_{f=0} = \pi\left(\frac{2}{\alpha} + \frac{4\alpha}{\alpha^2+4\pi^2} - \coth\left(\frac{\alpha}{2}\right)\right) \tag{3.16}$$

As is in the case of 3.10, the reassignment measure is a scaled version of the phase derivative with the scaling being equal to N/2π, where N is the DFT size. The greater the value of N in relation to the frame size, the better the estimate given of the phase derivative by the phase difference.

To demonstrate this and to show that the analytical derivation of the linear amplitude change estimator matches the phase difference, Figures 3.6 and 3.7 show the first-order phase difference plotted against the continuous phase derivative for a sinusoid whose frequency is exactly at the centre of an analysis bin. As can be seen by these figures, as N increases relative to the frame size(ie. increased zero-padding), the closer the difference measure becomes to the continuous derivative measure.

The phase derivative approximated by the phase difference is plotted in Figure 3.8. This shows the difference between the phase derivative for linear and exponential amplitude changes. As one would expect the phase difference for linear amplitude changes flattens out and plateaus before the exponential plot, as increasing linear amplitude change has diminishing effects on the displacement of the centre of mass in a frame compared to exponential changes which continue to shift the energy further away from the centre.

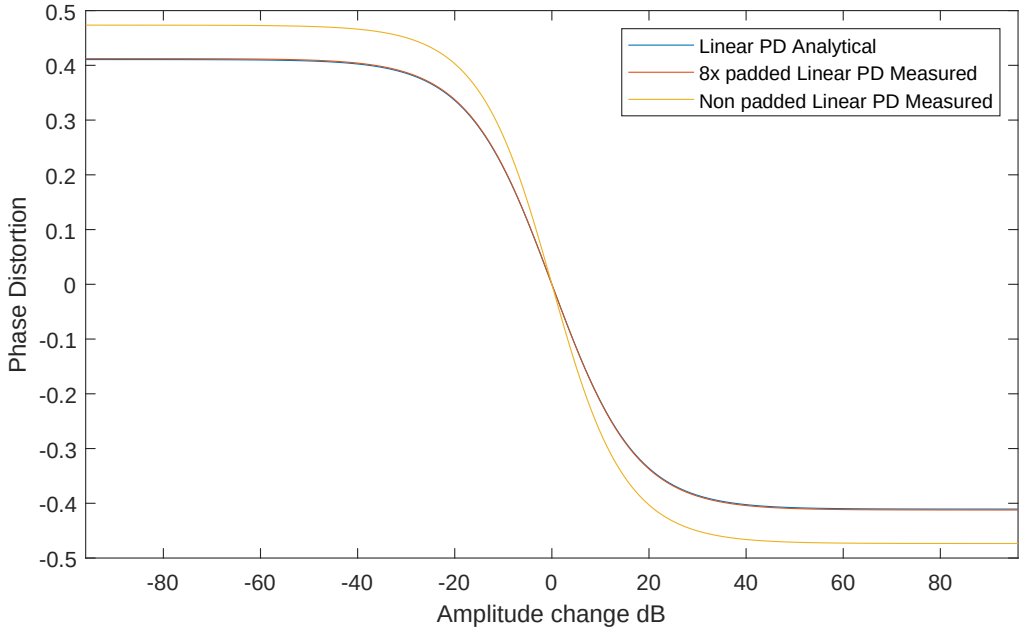


FIGURE 3.6: Linear Analytical and Measured phase difference compared

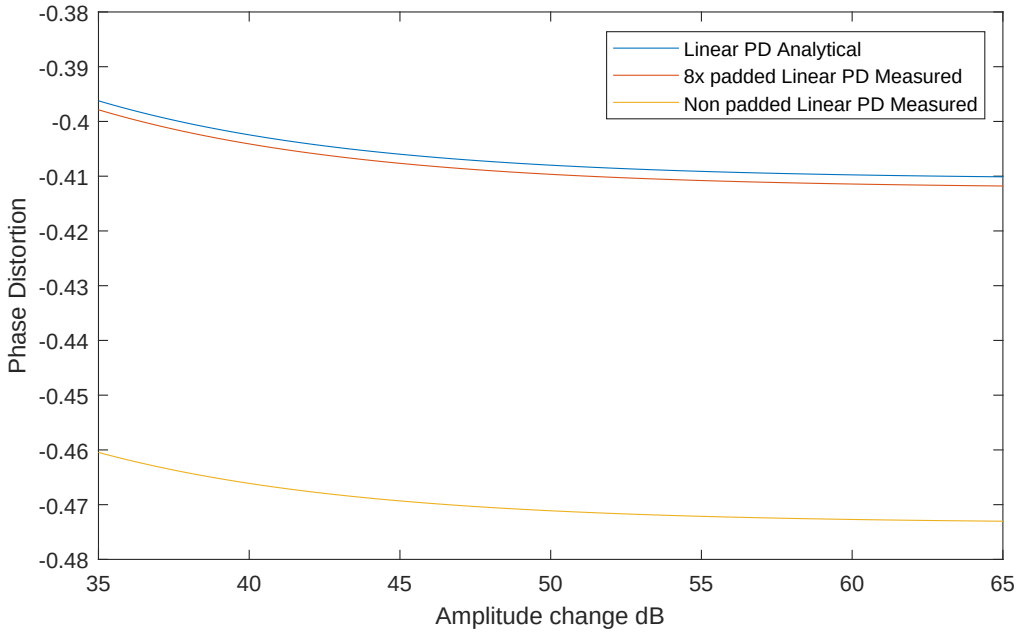


FIGURE 3.7: Linear Analytical and Measured phase difference compared

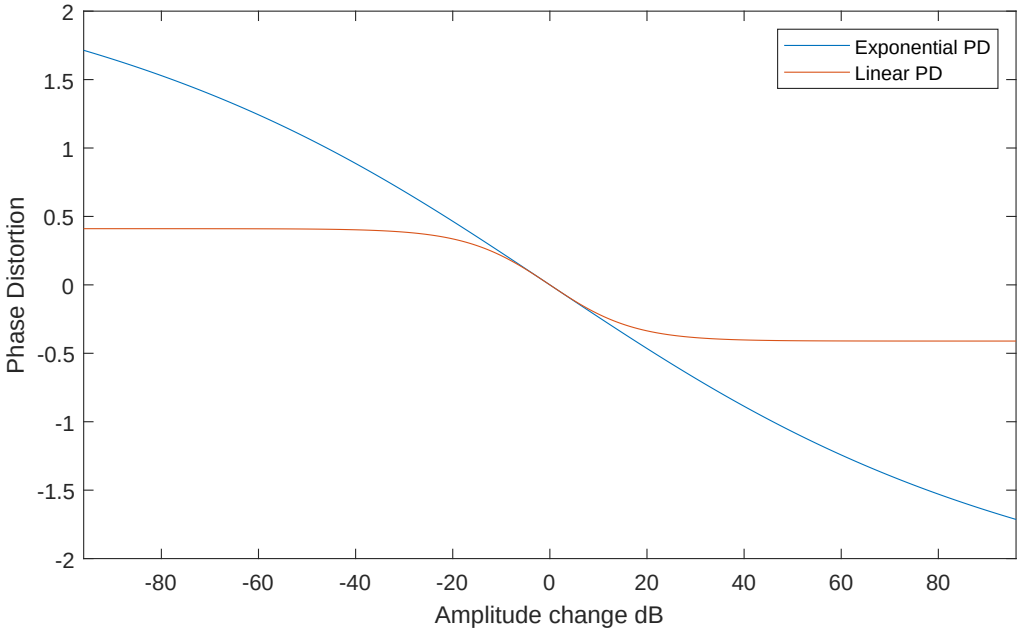


FIGURE 3.8: Linear vs exponential phase difference

Figure 3.8 shows the difference between exponential and linear phase difference measurements:

3.6 Envelope Type Discrimination

Although Fourier analysis provides a model where sinusoids are presumed to be stationary, information about non-stationarity is not lost as the transform is perfectly invertible. This information about non-stationarity is embedded within the relationship between the magnitude and the phase [176]. Given that we can have an estimate of the shift in the centre of gravity within a frame from the phase difference measure, how then do we distinguish between exponential and linear amplitude change?

The time offset value from the centre of an analysis frame for an impulsive component is estimated by the reassignment method as defined in 3.8 where $X_{h,k}$ and $X_{th,k}$ are the kth bin of the DFT of the input signal weighted by the window function h and its time ramped version th [149].

Given a change in amplitude and the resulting time reassignment offset from an exponential envelope, the corresponding amplitude change for a linear envelope to result in the exact same time reassignment offset can be calculated by examining 3.8 with respect to linear amplitude change. This is given by taking the ratio of a linearly ramped window function with its time ramped version. The equation for a linear amplitude curve used in 3.11 does not provide a compact expression which is neatly solvable given an exponential time reassignment value, and in this case, it is convenient to have the different curves being examined to start at the same value, in this case 1 (0 dB), so that the value at the end of the curve expresses the change in amplitude within an analysis frame in db.

$$W_{h,\text{linear}} = \int_{-\frac{1}{2}}^{\frac{1}{2}} (\frac{1}{2} + \frac{cos(2\pi t)}{2})((\alpha * t) + ((\alpha + 1) - (\frac{\alpha}{2}))) \qquad dt, |t| \leq \frac{1}{2} \tag{3.17}$$

$$W_{th,\text{linear}} = \int_{-\frac{1}{2}}^{\frac{1}{2}} (\frac{1}{2} + \frac{cos(2\pi t)}{2})((\alpha * t) + ((\alpha + 1) - (\frac{\alpha}{2}))) * t \quad dt, |t| \leq \frac{1}{2} \tag{3.18}$$

The resulting ratio for calculating the time reassignment offset for a linear amplitude change applied to a Hann window is given by 3.19.

$$\frac{W_{th,\text{linear}}}{W_{h,linear}} = (\frac{\nu * (-6 + \pi^2)}{24\pi^2})/(\frac{2 + \nu}{4}) \tag{3.19}$$

where ν is the value of the equation which needs solving given an exponential time reassignment offset.

The amplitude change required dA_{linear} is therefore found by solving.

$$dA_{\text{linear}} = \text{Solve}\left[\frac{\left(\frac{\nu*(-6+\pi^2)}{24\pi^2}\right)}{\left(\frac{2+\nu}{4}\right)} == \frac{-t_{0,\text{reassignment}}}{N}\right] \tag{3.20}$$

Where $t_{0,\text{reassignment}}$ is the DC time reassignment offset of an exponential curve applied to a window function h, scaled by N, in this case the frame size. The time reassignment measure is a scaled version of the phase derivative and as such results become more accurate with larger frame sizes.

Figure 3.9 shows two examples of linear and exponential amplitude change that produce the same time reassignment (and therefore phase difference) values by calculating the required amount of linear amplitude change required to match the exponential amplitude change from 3.20

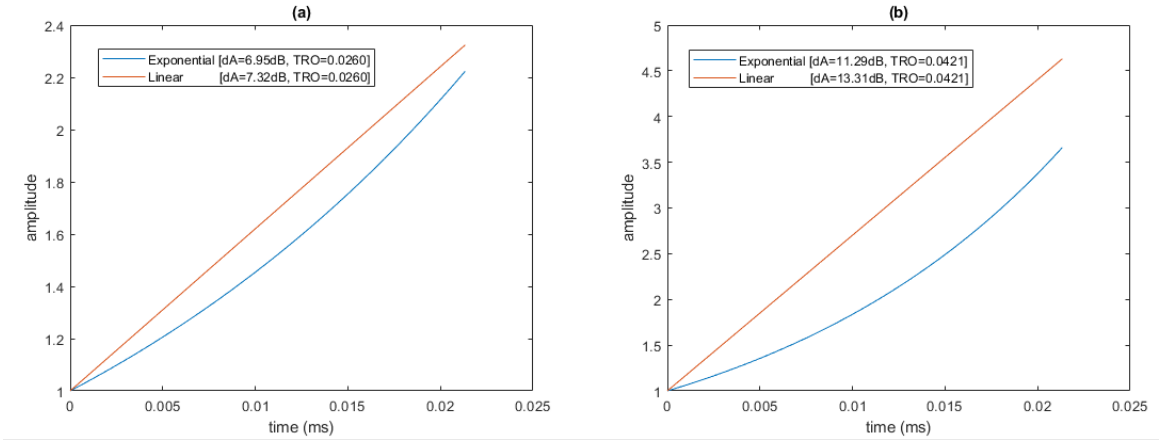


FIGURE 3.9: Linear vs exponential amplitude ramps with equivalent phase difference values

Amplitude change causes a flattening of the lobe around a peak [10, 177], which can be seen in Figure 3.10 where a linear amplitude change is compared to an exponential with the same time reassignment measure. Linear amplitude requires greater intra-frame change to produce the same phase differences as exponential amplitude. This manifests itself as a broader peak in the normalised DFT magnitude spectrum, as shown in Figure 3.11. This information can be used to then distinguish between the two possible amplitude change estimates given by equations 3.10 and 3.15.

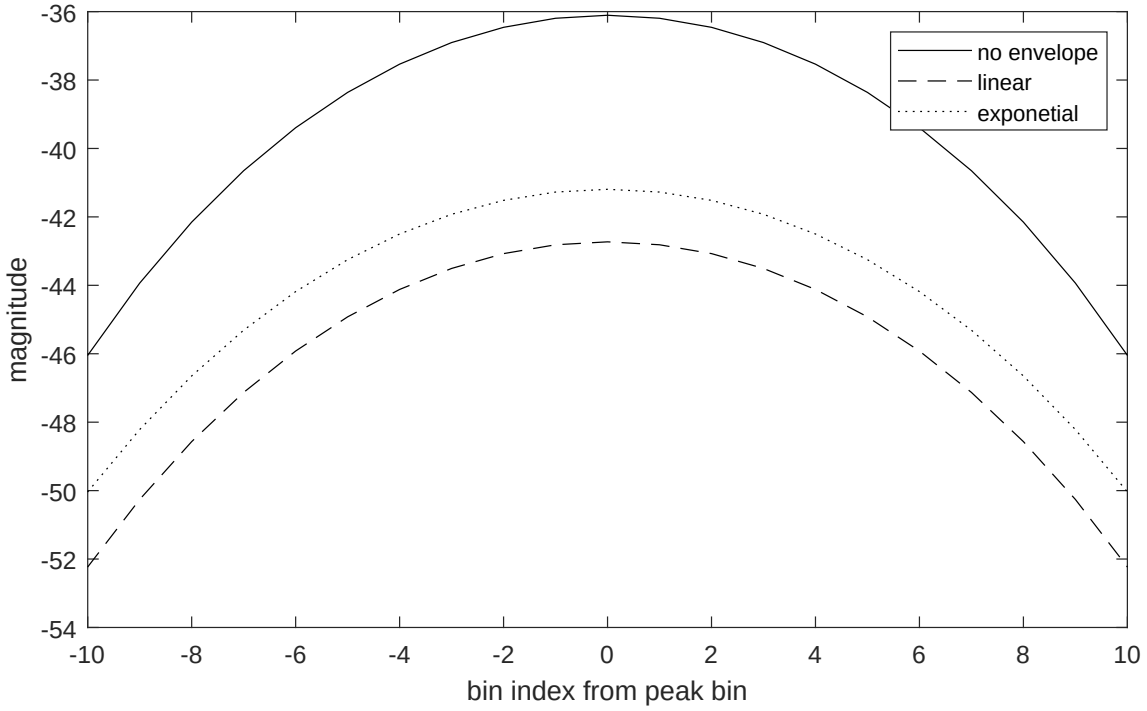


FIGURE 3.10: Flattening of a peak from amplitude change

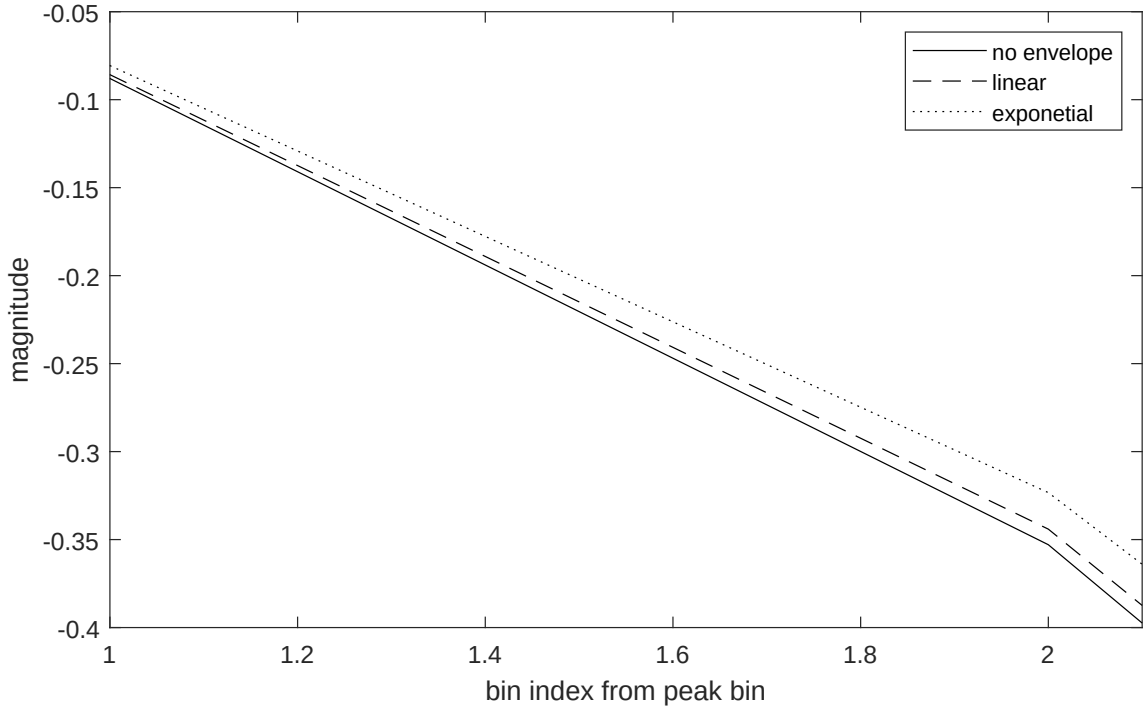


FIGURE 3.11: Widening of a peak with normalised magnitude

3.6.1 Magnitude Second Order Difference

The second derivative can be used as a measure of concavity of a peak. The second derivative of the magnitude across a sinusoidal peak can be approximated by its second-order difference. The normalised magnitude second order difference is given by 3.21

$$\mathrm{M2d_k} = \frac{(|X_{h,k+1}| - |X_{h,k}|) - (|X_{h,k}| - |X_{h,k-1}|)}{|X_{h,k}|} \tag{3.21}$$

Where $|X_h|$ is the magnitude of the DFT, of the input sequence weighted by the window function h, and k is the peak bin number. Normalization is required to compare magnitude curves with equal peak values, removing the bias introduced by spectral peaks with different amplitude levels.

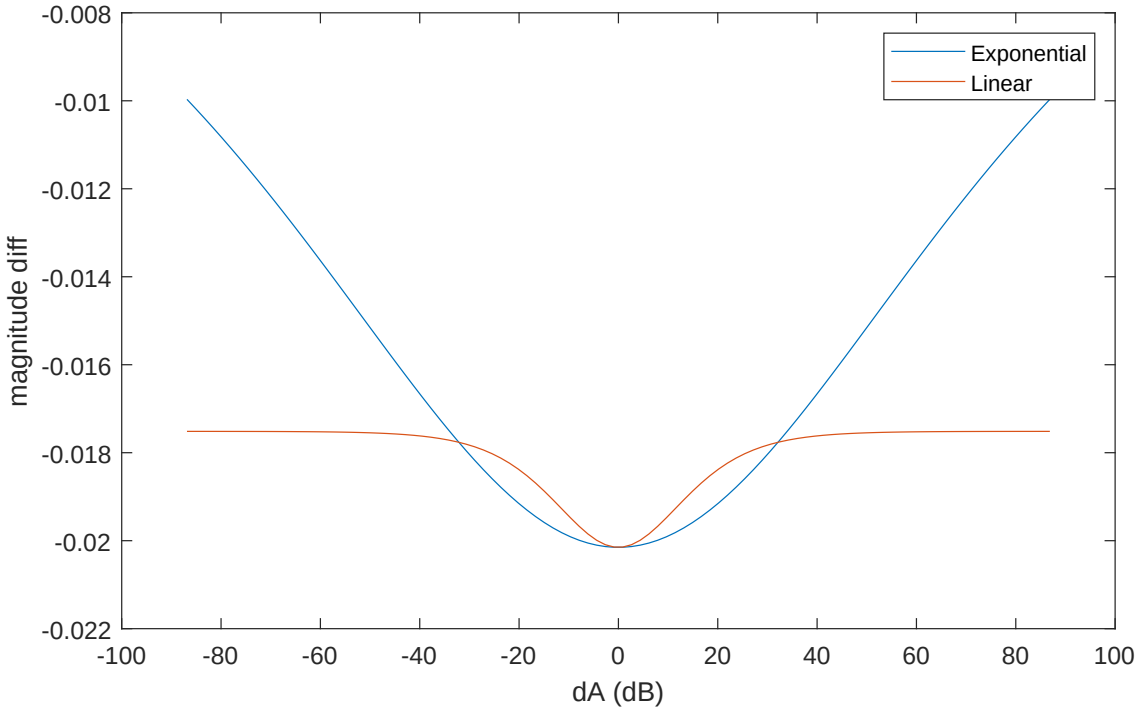


FIGURE 3.12: Magnitude second order difference measures for different amplitude changes [-100 to 100] dB, frequency is centred in the middle of an analysis bin, and $\Delta f = 0$.

Figure 3.12 shows the magnitude second order difference for linear and exponential amplitude change over a certain range. As with the phase difference measure, the magnitude second order difference measure is unable to distinguish between linear and exponential curve types alone. However, a curve type with a specific change in amplitude will have a unique combination of phase and magnitude difference measure. Given an expected magnitude second order difference value from a recomputed lookup table

in conjunction with the measured magnitude second order difference produces a discriminator where a high correlation returns a value of 1.

$$\text{Envelope Likelihood Measure}_{\text{k}} = \frac{\text{M2d}_{\text{k,expected}}}{\text{M2d}_{\text{k,measured}}} \tag{3.22}$$

The steps for selecting the envelope type are:

1. Given a measured phase difference value, two possible amplitude change estimates are looked up from the linear and exponential amplitude change lookup tables generated with equations 3.10 and 3.15

2. With an estimated values for change in amplitude (one for linear the other for exponential), expected values for the magnitude second order difference can be looked up from two lookup tables. These are generated by applying linear and exponential amplitude changes over a specified range and measuring the magnitude second order difference from3.21

3. Both expected magnitude second order difference values are then compared against the measured magnitude second order difference value.

4. The result with the highest correlation to the measured magnitude difference is selected as the estimated curve type, along with the corresponding amplitude estimate.

Figures 3.15 3.16 3.17 3.18 show the magnitude second order difference plotted for linear and exponential amplitude change in the presence of noise and frequency change. The liner plot and resulting lookup table, clearly shows different values to the exponential plots. They can therefore be used in conjunction with the phase difference and amplitude change estimate to discern between envelope curve types.

The magnitude second order difference is mostly robust to levels of added noise, but high levels of noise do distort the measurement, as can be seen in Figure 3.16. Noise at this level of 20dB or greater results in unreliable measurements and is no longer useful for distinguishing between linear or exponential amplitude changes.

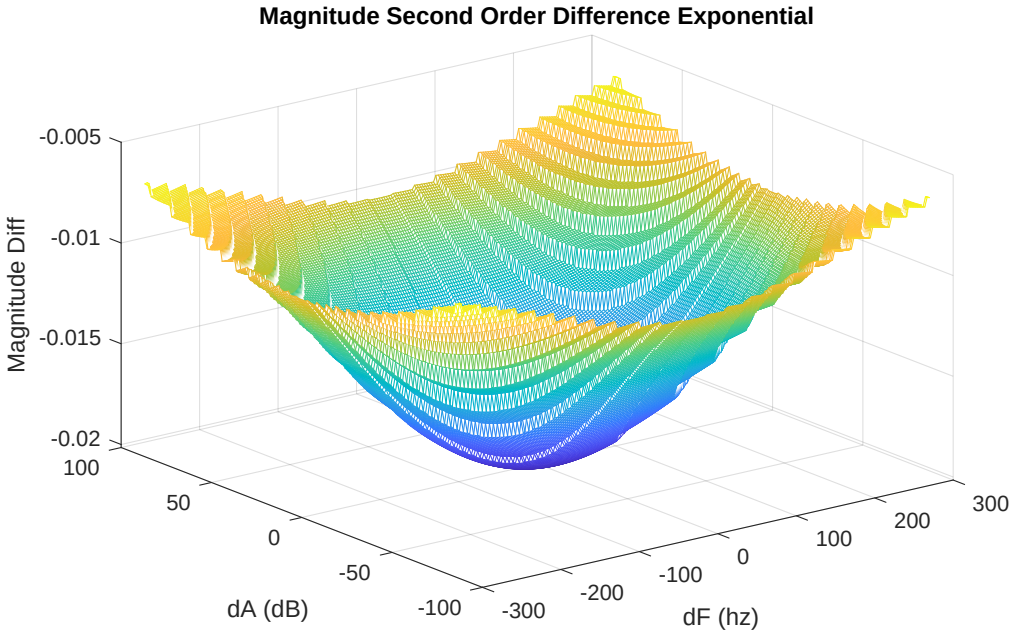


FIGURE 3.13: The magnitude second order difference measures for Exponential ΔA [-100 to 100] dB

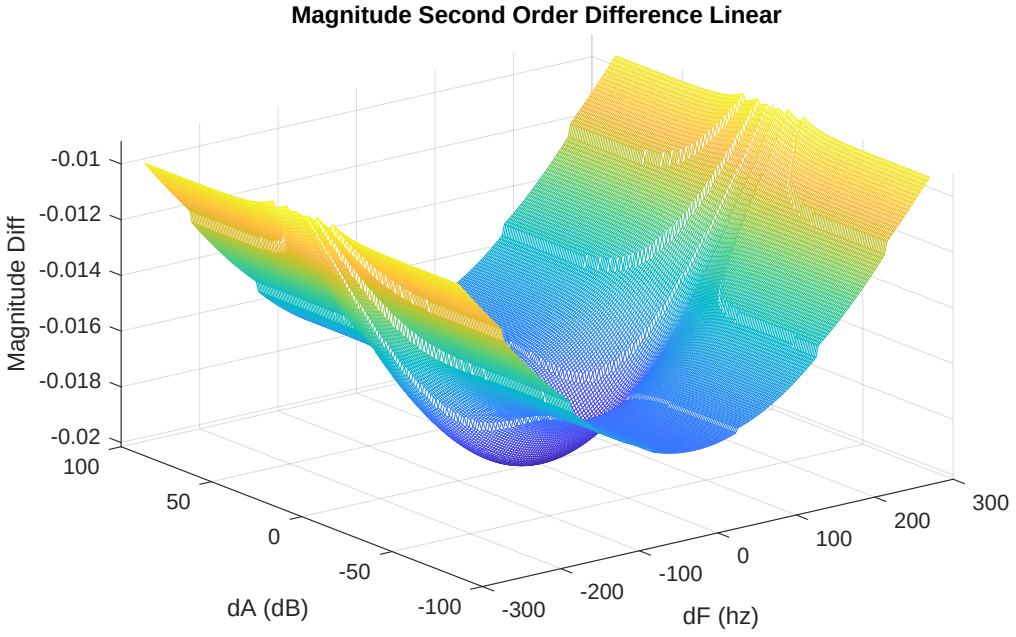


FIGURE 3.14: The magnitude second order difference measures for Exponential ΔA [-100 to 100] dB

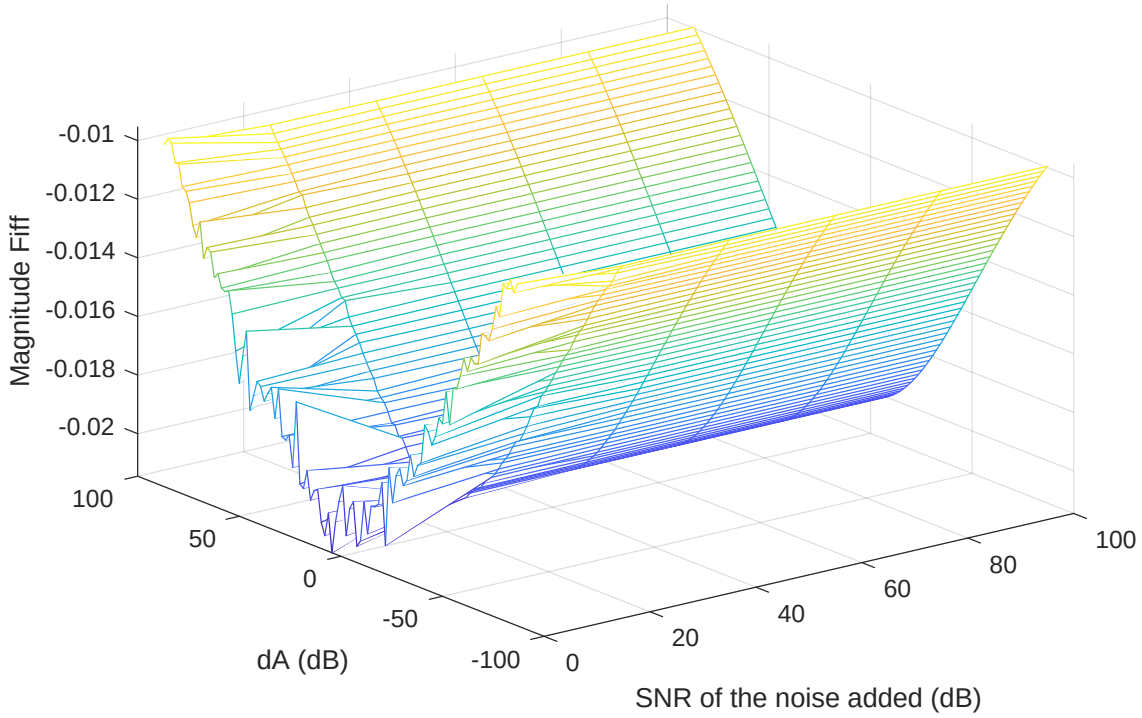


FIGURE 3.15: The magnitude second order difference measures for Exponential ΔA [-100 to 100] dB with SNR Noise [0 to 100] dB

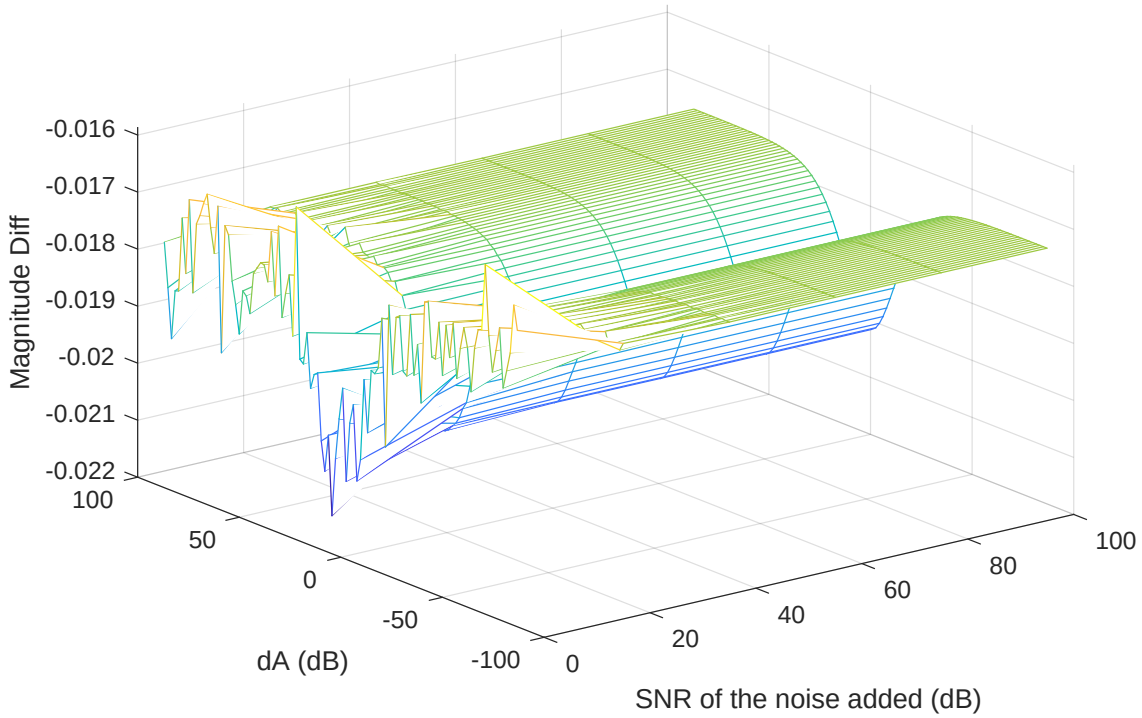


FIGURE 3.16: The magnitude second order difference measures for Linear ΔA [-100 to 100] dB with SNR Noise [0 to 100] dB

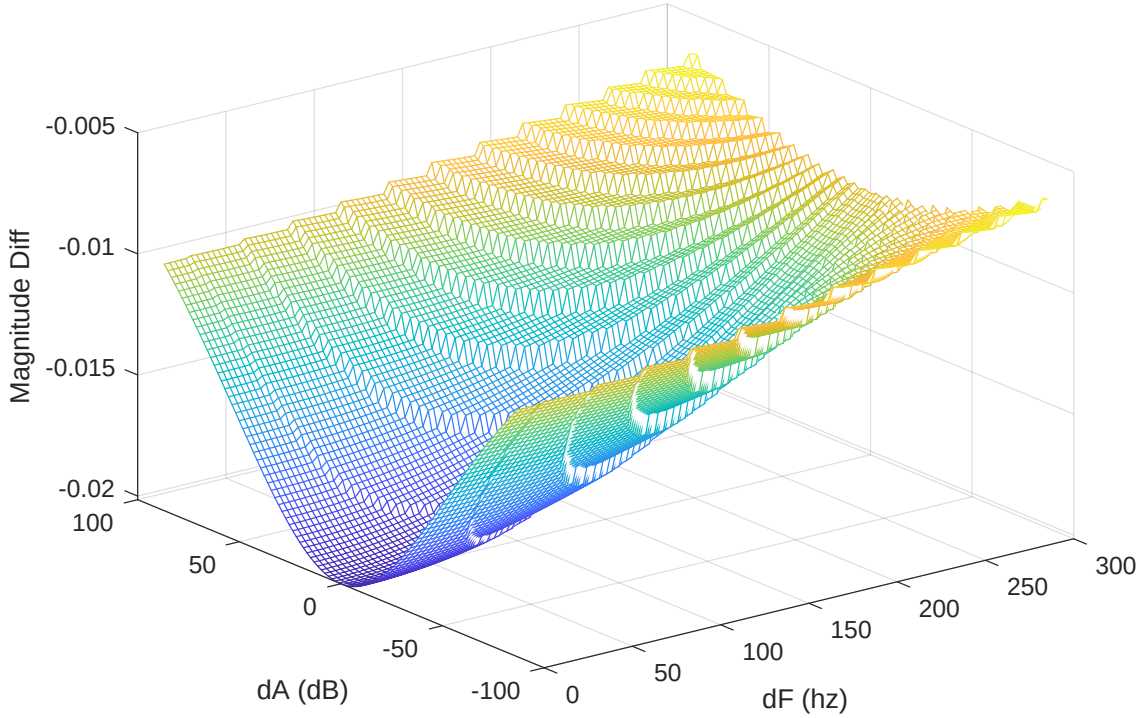


FIGURE 3.17: The magnitude second order difference measures for Exponential ΔA [-100 to 100] dB with dF [0 to 300] Hz

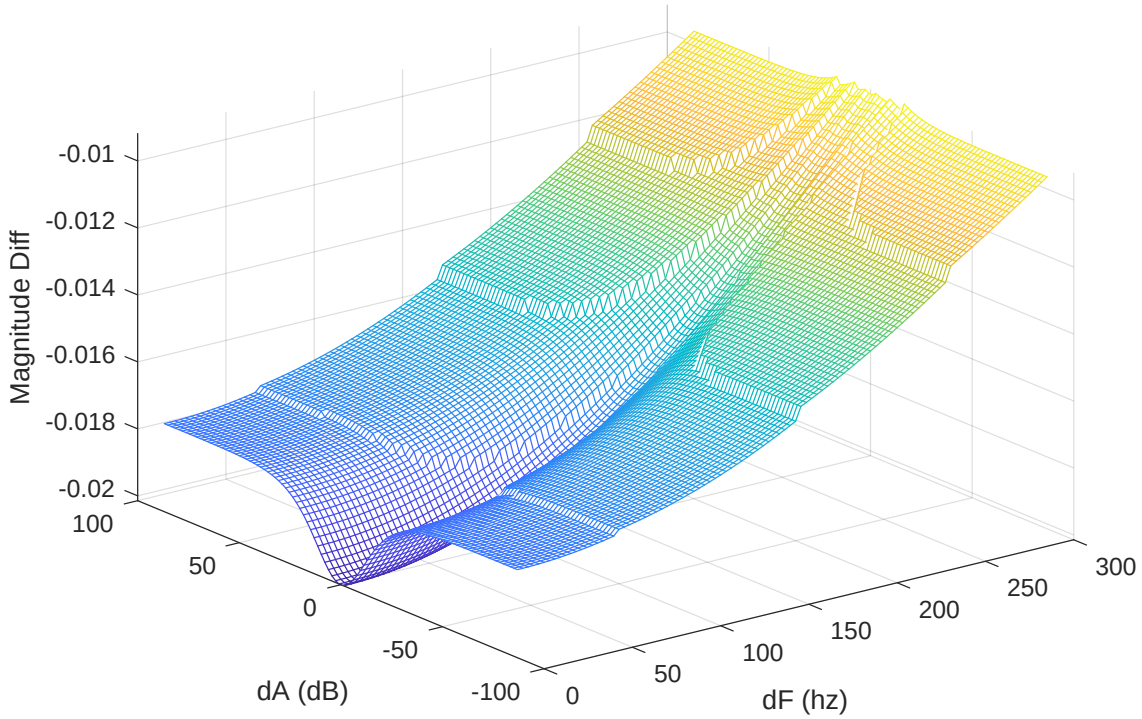


FIGURE 3.18: The magnitude second order difference measures for Linear ΔA [-100 to 100] dB with dF [0 to 300] Hz

3.7 Comparison of Linear and Exponential Amplitude changes

The reassignment and derivatives methods are highly effective for estimating exponential amplitude change. Figure 3.19 compares the SNR measurements of reassignment amplitude modulation estimates for both exponential and linear amplitude change using the DESAM Toolbox [178]. The change in amplitude is measured from -96 to 96 dB which is the maximum dynamic range (96dB) of a 16-bit digital audio system.

The estimates returned when linear amplitude modulation is applied deteriorate rapidly while the exponential amplitude estimates preform much better.

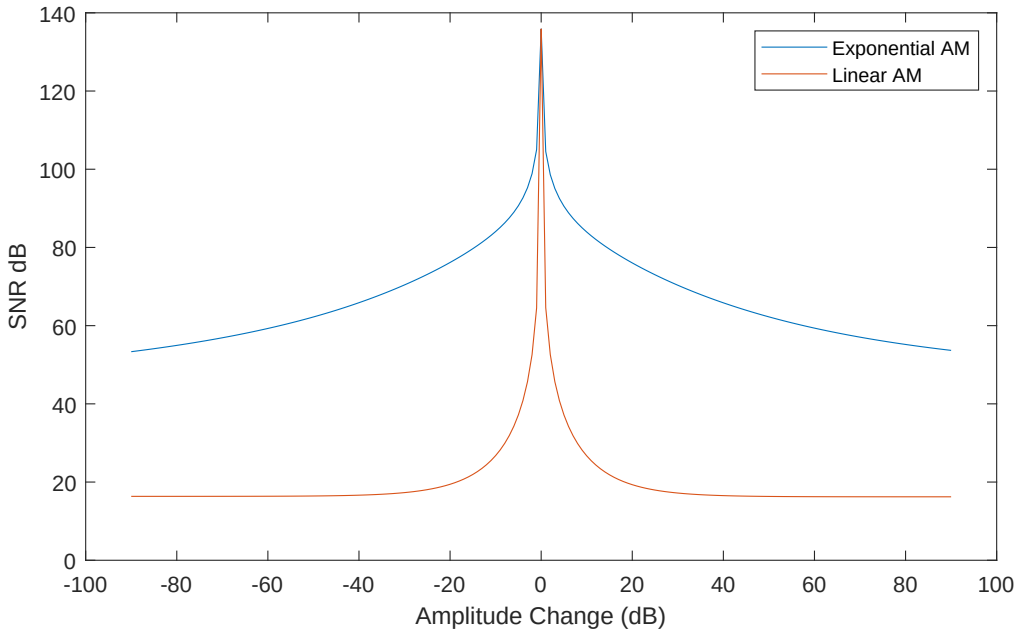
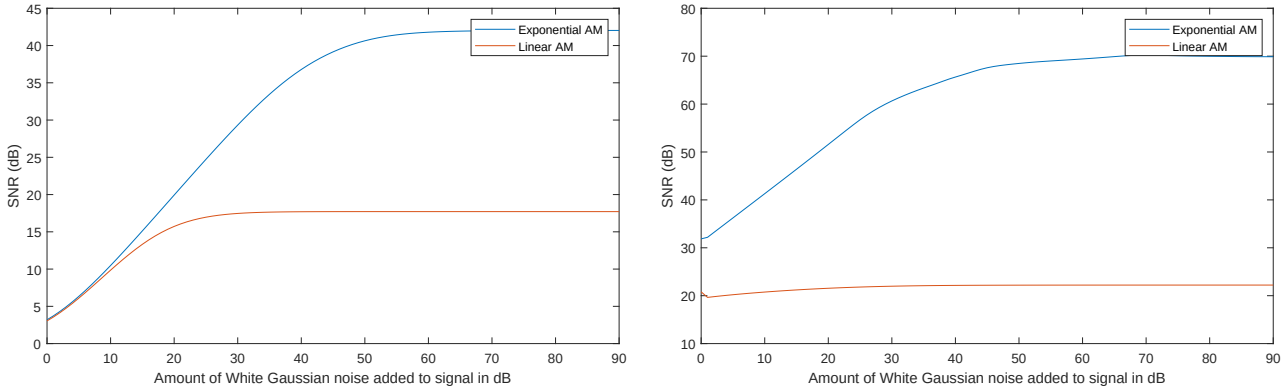


FIGURE 3.19: SNR results comparing a stationary sinusoid with linear and exponential amplitude changes applied to it with estimates from Reassignment.

Figure 3.20a shows the average SNR returned using 1000 sinusoidal components with random parameter values ranging from A=U[0.1, 1], f=U[600, 16000], pi=U[-2pi, 2pi], dA=[0, 60] dB, for each calculation of added Gaussian Noise ranging from 0 to 90 dB. The SNR is calculated by comparing the original signal without the added noise, to the synthesised signal from all of the estimated parameters returned from reassignment. In comparison, Figure 3.20b displays the SNR results of comparing the original signal, with a signal synthesised using the original signals parameters except for the estimates returned from reassignment for $\hat{\mu}$ and $\hat{a}$. The SNR results for comparing amplitude modulation with just $\hat{\mu}$ and $\hat{a}$ shows a significant improvement for exponential amplitude change when compared to a signal

synthesised with all of the reassignment parameter estimates, where slight deviations from original parameter values bias the result of isolating amplitude modulation in the test. The results in the presence of linear amplitude change remain quite poor with all results returning SNR values around 20 dB from the isolated test.



(A) Reassignment SNR of Linear vs Exponential using estimates of all parameters

(B) Reassignment SNR of Linear vs Exponential using only estimates of $\hat{\mu}$ and $\hat{a}$

FIGURE 3.20: Reassignment SNR measured with added Gaussian noise [0, 90] dB, each with 1000 random data points with A=U[0.1, 1], f=U[600, 16000], pi=U[-2pi, 2pi], dA=[0, 60] dB

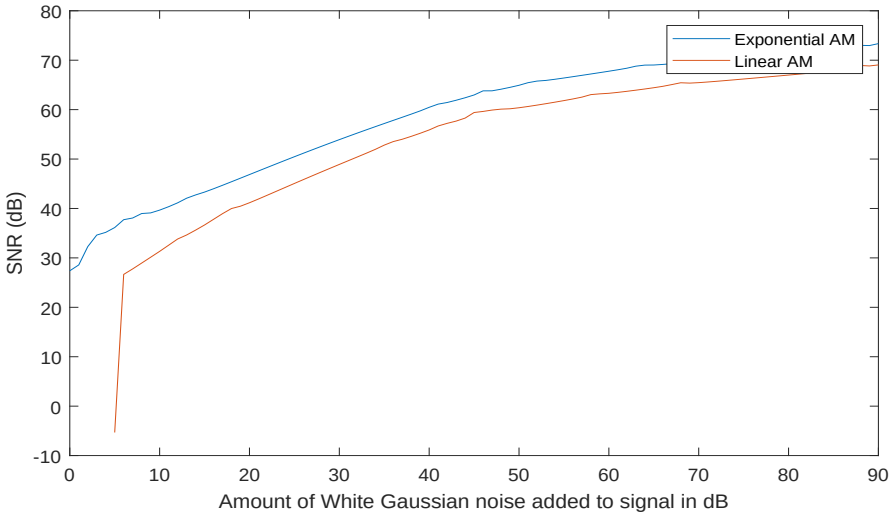


FIGURE 3.21: Phase Distortion SNR measured with added Gaussian noise [0, 90] dB, each with 1000 random data points with A=U[0.1, 1], f=U[600, 16000], pi=U[-2pi, 2pi], dA=[0, 60] dB

Figure 3.21 displays the results from using phase distortion measurements where estimates of amplitude modulation are isolated from other parameter estimates. Results compared between exponential amplitude modulation using reassignment and phase distortion show similar results. The use of the analytical equation used for estimating linear amplitude change clearly performs better than the results

returned from reassignment, however linear phase difference measurements have difficulties estimating amplitude change with AGWN levels below 10 dB.

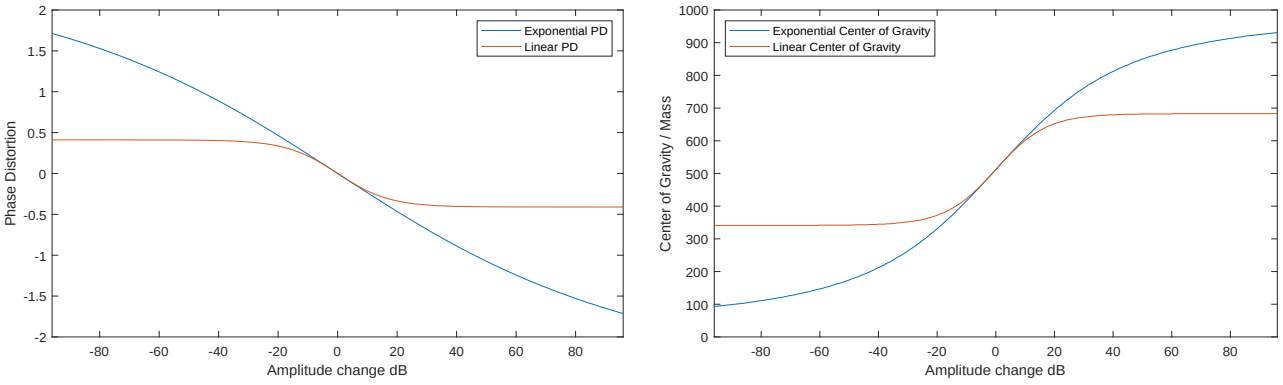
The difference in quality returned by linear amplitude change compared to exponential amplitude change, can be attributed to the effect these curves have on displacing the energy within a frame.

Exponential amplitude changes result in larger displacements of energy due to the curvature of this function. Comparing phase distortion measurements for equal changes in amplitude (dB) highlights this with the resulting curve for linear phase distortion flattening out as the effect on the phase starts to decrease after a certain point with greater changes in amplitude.

Figure 3.22a displays the phase difference curves calculated from the equations (3.16, 3.15) presented in Section 3.5, for amplitude changes from -96 to 96 dB. The linear curve in Figure 3.22a approaches a limit from around 30 dB, after which the differences in phase distortion measurements diminish with increasing amplitude change.

A similar result is seen when inspecting the effect which exponential and linear amplitude change has on the 'Center of Gravity' (COG), as shown in Figure 3.22b. The COG of the x-axis coordinate $\bar{x}$ for a signal with N samples is given by:

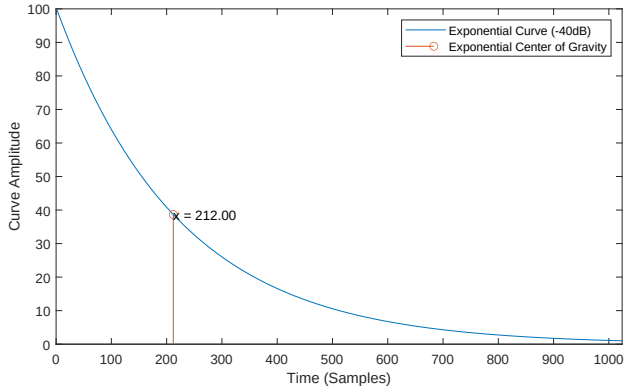
$$\bar{x} = \frac{\left(\sum_{i=1}^{N} x_i y_i\right)}{\left(\sum_{i=1}^{N} y_i\right)} \tag{3.23}$$



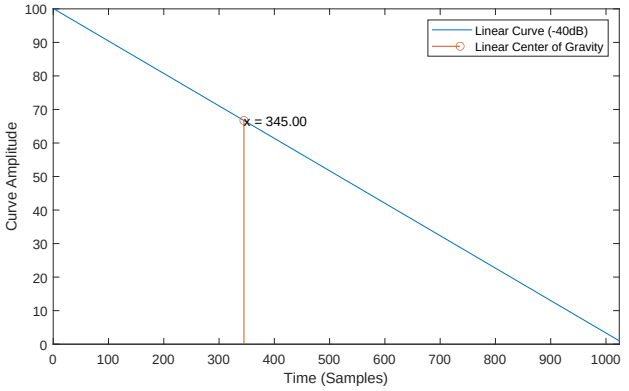
(A) Comparison of phase difference for exponential and linear amplitude changes from -96 to 96 dB

(B) Comparison of Center of gravity measurements for exponential and linear amplitude changes from -96 to 96 dB

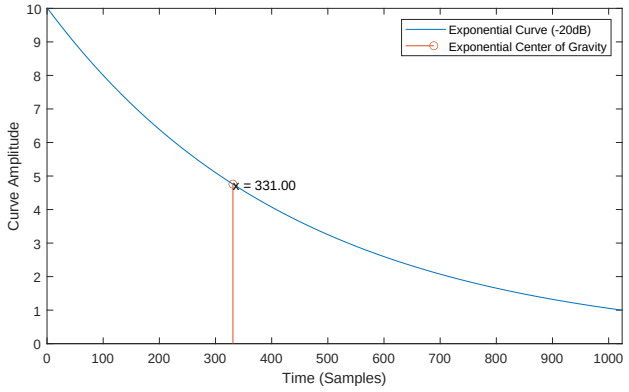
FIGURE 3.22: Comparison of phase difference against the center of gravity for exponential and linear amplitude change



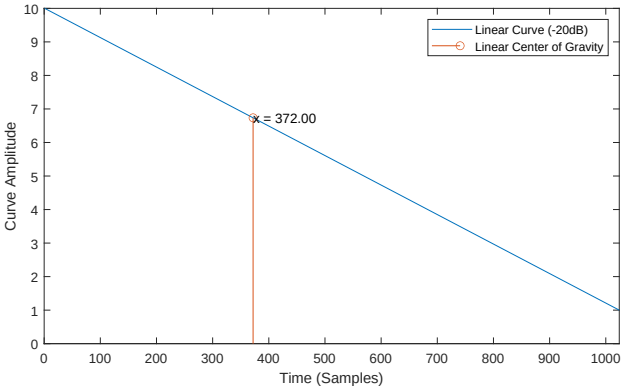
(A) Exponential Curve Center of Gravity (-40 dB)



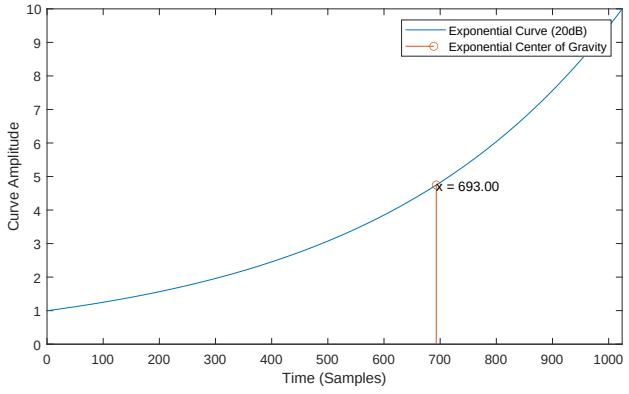
(B) Linear Curve Center of Gravity (-40 dB)



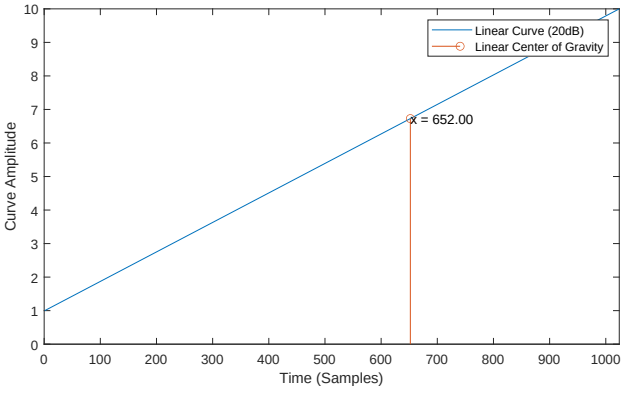
(C) Exponential Curve Center of Gravity (-20 dB)



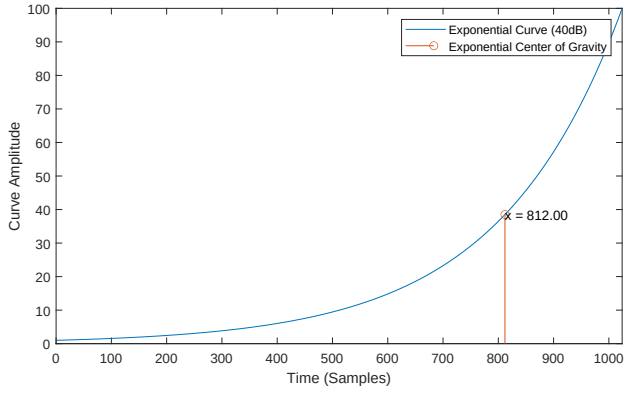
(D) Linear Curve Center of Gravity (-20 dB)



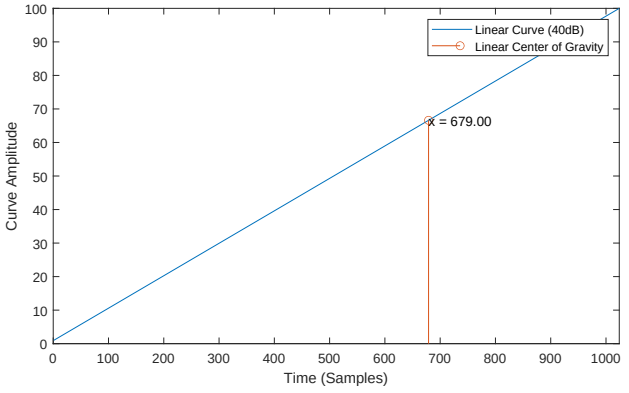
(E) Exponential Curve Center of Gravity (20 dB)



(F) Linear Curve Center of Gravity (20 dB)



(G) Exponential Curve Center of Gravity (40 dB)



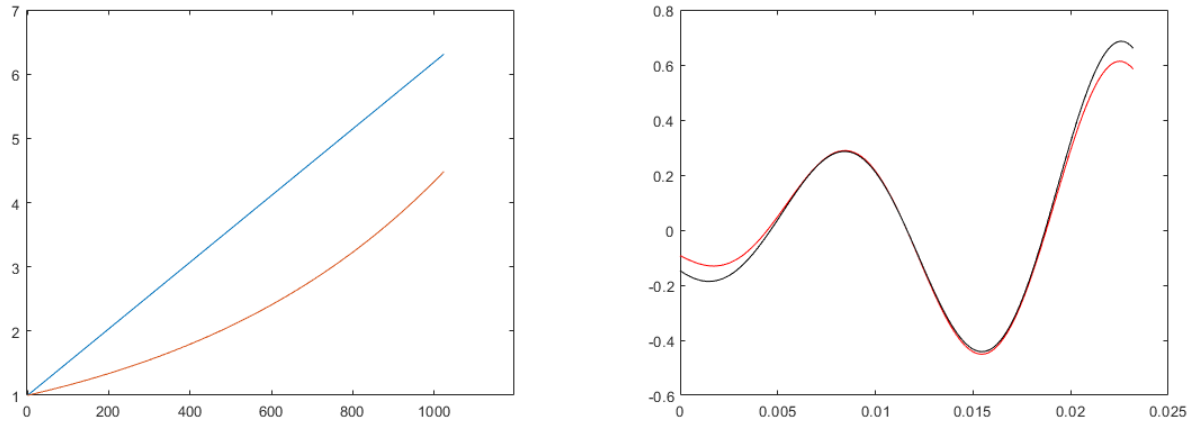
(H) Linear Curve Center of Gravity (40 dB)

FIGURE 3.23: Comparison of the center of gravity measurements for exponential and linear amplitude curves with varying changes in amplitude.

Figure 3.23 shows how the center of gravity is affected by different amplitude changes. Exponential and linear measurements are displayed next to one another with the same amounts of amplitude change applied for comparison. This clearly shows how the exponential amplitude change has a greater effect on shifting the COG due to the radius of the curve.

3.7.1 Performance of Linear amplitude change using Exponential AM models

Below is an example of a Linear amplitude change of 16 dB. Reassignment estimates this as a 13.0313 dB exponential amplitude change. Figure 3.24a shows the difference between the reference linear amplitude curve and the estimated exponential amplitude curve, as well as the resulting sinusoid with the incorrectly estimated amplitude change.

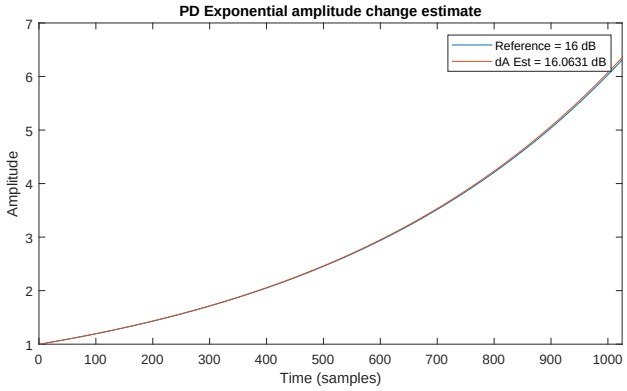


(A) [Linear vs Estimated Exponential curve

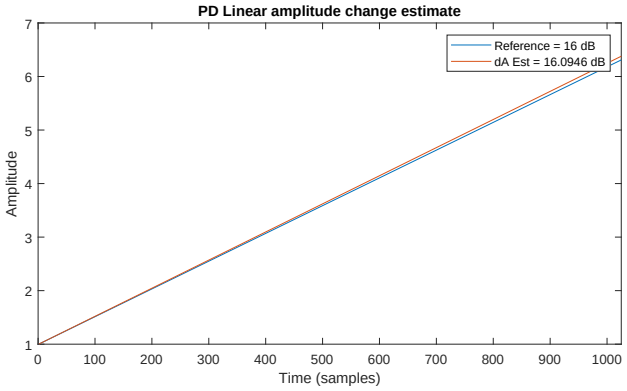
(B) Linear vs Estimated Exponential Sinusoid

FIGURE 3.24: 13 dB estimated exponential amplitude change compared to reference of 16 dB linear amplitude change

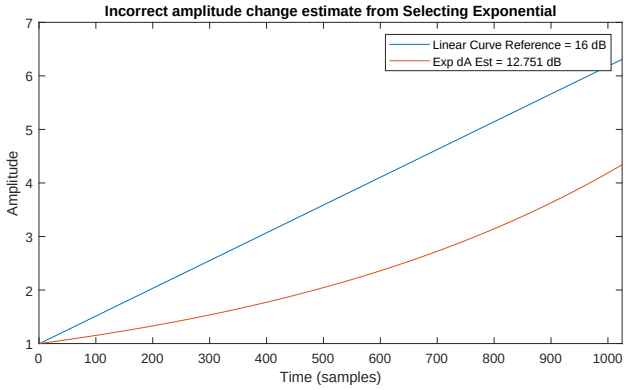
The estimation methods described in this chapter can equally provide incorrect estimates of amplitude change if the incorrect curve type is selected. Figure 3.25 below displays the results from phase difference estimates from linear and exponential amplitude curves of 16 dB. The incorrect amplitude estimates resulting from selecting wrong curve type are shown in 3.25c and 3.25d. The incorrect selection of Linear amplitude change in the case of exponential amplitude change results in a larger discrepancy than the opposite case where an incorrect value of 27 dB (difference of 11 dB) is estimated compared with only 12.751 dB (difference of 3.25 dB) in the case of presuming or selecting exponential amplitude change int he presence of linear.



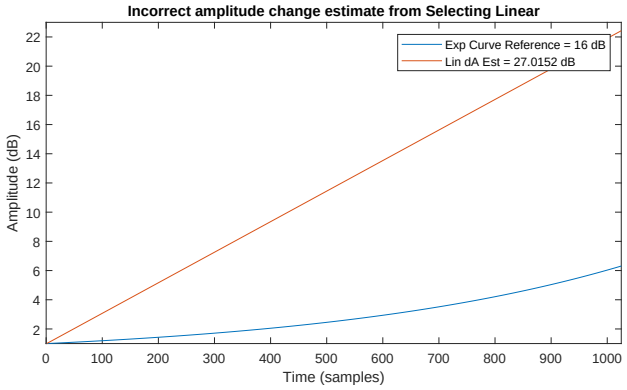
(A) Exponential Curve (16 dB)



(B) Linear Curve (16 dB)



(C) Incorrect Exponential Curve (16 dB)



(D) Incorrect Linear Curve (16 dB)

FIGURE 3.25: Comparison of the phase difference estimates of amplitude change (16 dB) and the incorrect estimates which would result from incorrect curve selection.

The incorrect estimation of amplitude change in a single frame non-overlapping (OLA) system can cause discontinuities at frame boundaries. In such a case the discrimination of amplitude curve types and a more accurate amplitude change estimate in the case of linear amplitude change can improve the output of the model and mitigate discontinuities at frame boundaries.

3.8 Performance of Discriminators

In this section the novel approach introduced by the research in this chapter regarding the use of the phase difference and magnitude second order difference measures are combined to distinguish between exponential and linear amplitude change, and the results of the estimator are presented.

As described in the previous section, the magnitude second order difference is robust to noise above 20 dB but is affected by changes in frequency within an analysis frame. In practice changes in frequency and the magnitude second order difference are not independent of each other and a 2D lookup table is required for estimating the curve types correctly in the presence of frequency change. Frequency change is discussed in more detail in the following Chapter 4. However, change in frequency affects both the phase and magnitude spectrum. Change in frequency causes the width of the main lobe of a sinusoidal peak to spread out across neighbouring frequency bins as shown in Figure 4.2. Change in frequency also affects the slope of the phase causing the phase across a sinusoidal peak to concave as described in [159, 176]. The effect frequency change has on the phase and resulting phase difference measurements are shown in Figures 4.14 and 4.15.

This is discussed in more detail in Section 4.3.1 regarding the base atom used in the spectral modelling system used in this thesis. Figure 3.26 displays the SRER metric from C.18 to evaluate the performance of the envelope discriminator and estimate of amplitude change in the presence of added white Guassian noise. Change in amplitude ranges from -60 dB to 60 dB for each result obtained from the added noise. The Signal-to-Noise-Ratio (SNR) of added noise ranges from 20 to 90 dB in steps of 10 dB. The method of discriminating between linear and exponential amplitude change is given by 3.22. Amplitude estimates are looked up for both linear and exponential amplitude change from first order phase difference measurements. The estimated amount of amplitude change is used to lookup the expected magnitude second order difference for each of these curve types. The measured magnitude second order difference is then compared with the expected value with the closest value used to indicate which envelope type selected. The test shows 952 correct results for estimating the envelope type, and only 16 incorrect result. In general, the incorrect results are when amplitude change is close to 0 dB and the resulting phase difference values and amount of amplitude change between exponential and linear curves is negligible. Figure 3.27 shows some statistics taken from these results for each amount of added noise. This clearly shows that results from SNR noise amounts below 30 dB start to deteriorate. Figure 3.28 displays a closer view where it is clearly shown that only one incorrect result is recorded

for added noise between 40 and 80. The mean of this error for all results is 0 dB which in effect is a stationary sinusoid, resulting in no actual difference between linear or exponential amplitude change.

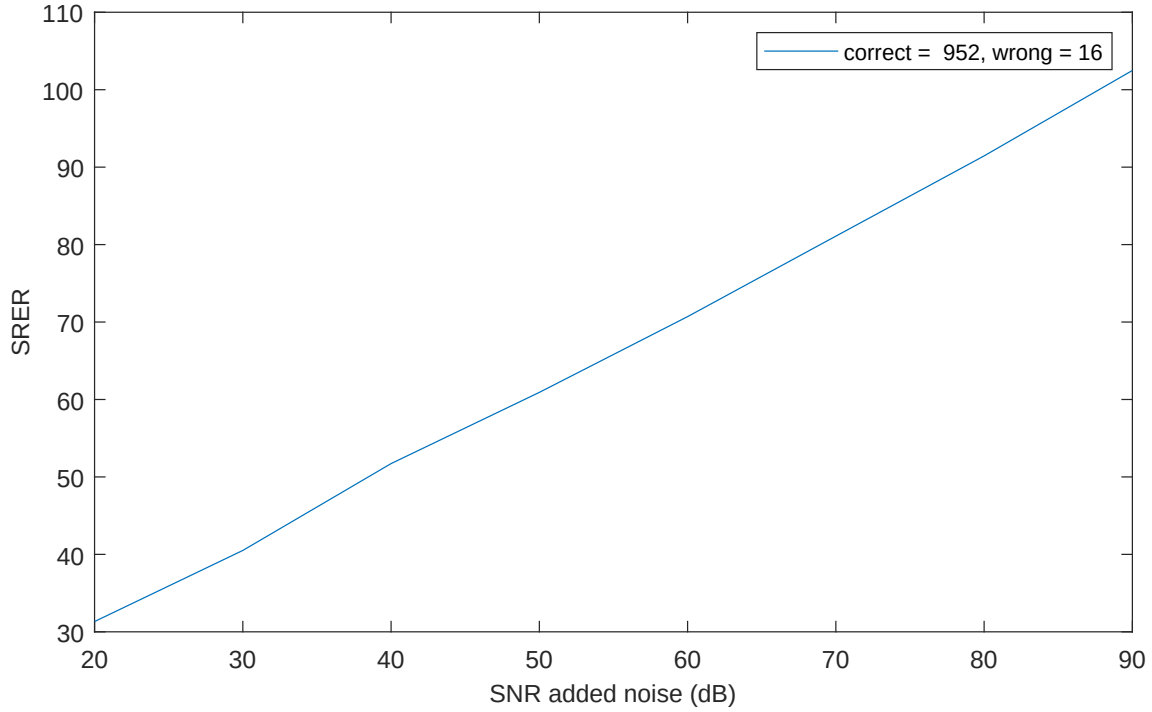


FIGURE 3.26: SRER Envelope Type Discrimination and estimation of Amplitude Change

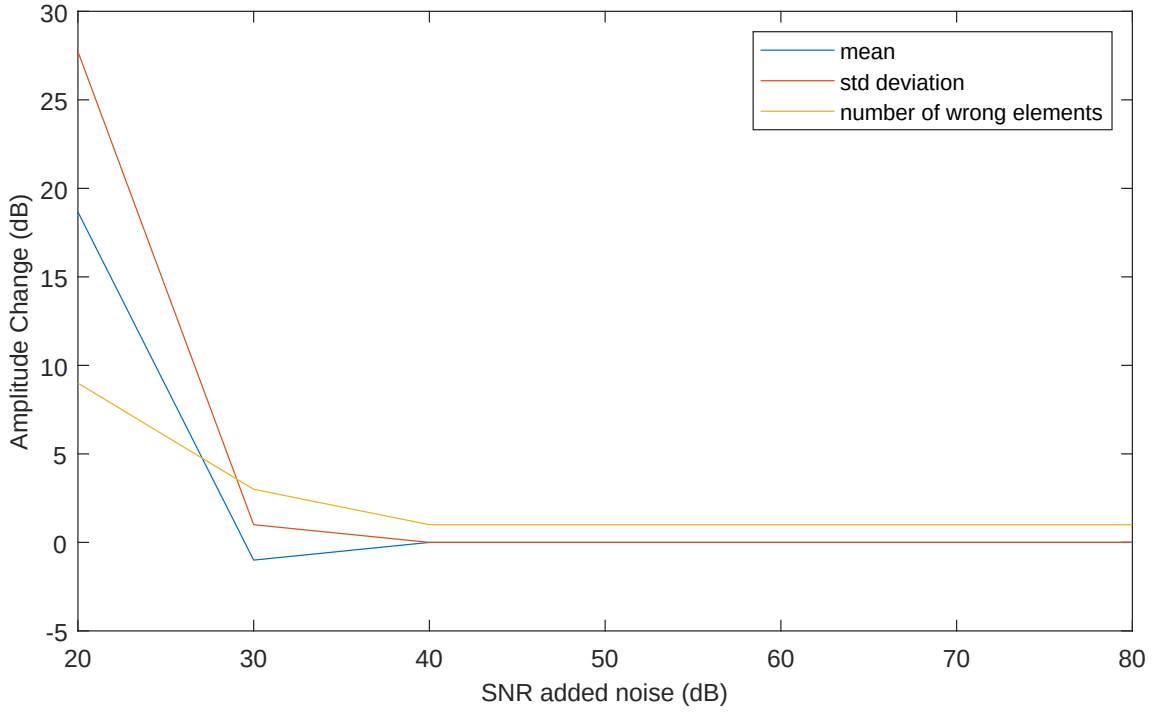


FIGURE 3.27: SRER Envelope Type Discrimination Statistics (20:80)

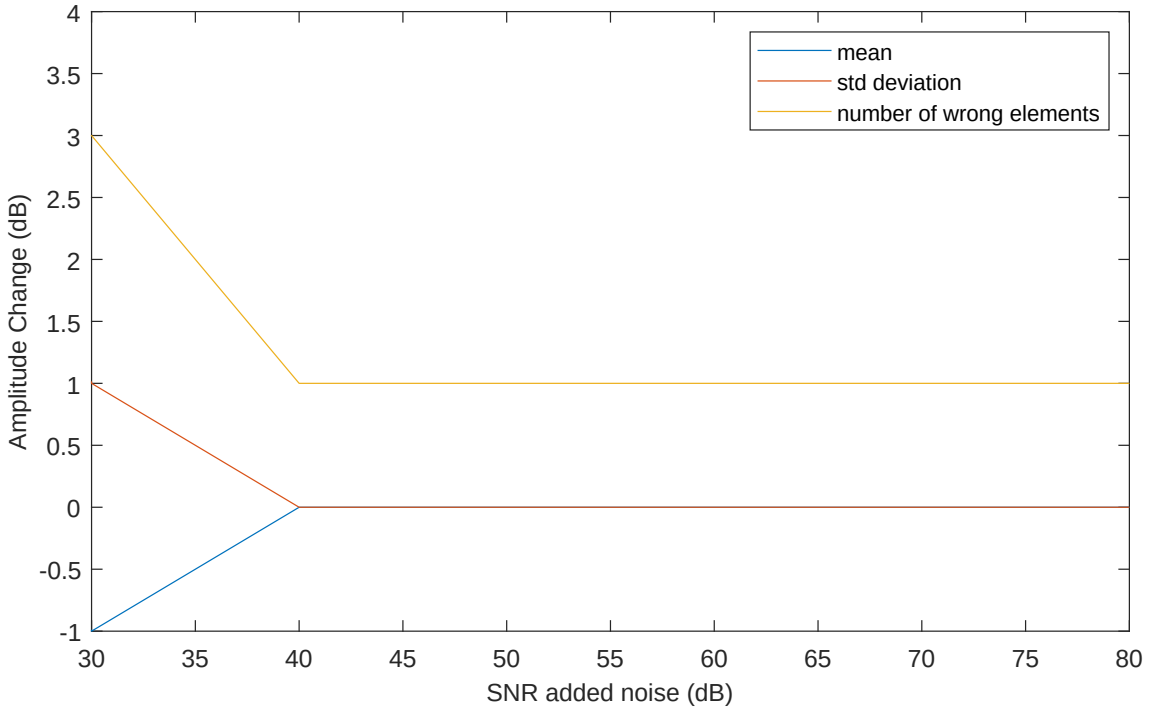


FIGURE 3.28: SRER Envelope Type Discrimination Statistics (30:80)

3.8.1 Performance of Envelope Type Discrimination

Receiver Operating Characteristics (ROC) graphs are used here to compare the performance of the discriminator for selecting a linear or exponential amplitude curve for a given sinusoid with a range of ΔA and Δf values combined with random envelope types, random frequency, and a specified value of added noise. These graphs offer a way of evaluating binary classifiers and have been used to measure the performance of sinusoidal discriminators [161]. They are used here measure the performance of the envelope type discriminator described previously and to compare the selection of linear and exponential amplitude changes against random curve types.

The ROC graph is produced by measuring the true positive rate (TPR) against the false positive rate (FPR). The top of the graph along the y-axis where TRP = 1 indicates a discriminator that is performing perfectly whereas points closer to the bottom of the graph where TRP = 0 indicate that the performance is poor. A perfect classifier will produce a line which moves from (0.0, 0.0) to (0.0, 1.0) and from there to (1.0,1.0). The closer a graphs area under a curve is to covering the entire space (AUC = 1), the better is its performance under the conditions for which it is being tested.

For each of the ROC figures presented here 1000 instances of a 1025 sample sinusoid have been generated rom 3.1. For each instance, f is randomly chosen with a uniform distribution within the interval 20 Hz – 20 kHz, A (the amplitude of the sinusoid) is at full scale = 1, the envelope type is randomly chosen (linear or exponential), ΔA and Δf are a range of uniformly distributed random values specified for each individual plot. The resultant signal has been combined with white Gaussian noise, whose energy relative to that of the sinusoid is specified for each plot. The ROC measure is used as a binary classifier to measure if an envelope type has been selected correctly, hence linear curve selection and exponential curve selection are individually calculated against the set of randomly selected curve type.

For each of measures presented:

1. The maximum peak is selected from the FFT of an $8x$ zero-phase padded frame taken of the Hann-windowed signal.

2. The phase difference around a peak is measured used to estimate changes in amplitude for linear and exponential amplitude change as described in 3.10 and 3.15.

3. The linear and exponential amplitude change estimates are used to lookup the expected magnitude second order difference values related to those amplitude changes. A 2D lookup table can be used if an approximation for the change in frequency has also been estimated. The tests conducted here combined a lookup table uniformly distributed and measured every 30 Hz for frequency changes between 0 and 300 Hz.

4. The envelope type discriminator is calculated for both linear and exponential amplitude curve types as described in 3.22 and the result closest to 1 selected as the estimated curve type.

5. The estimated curve type is then compared to the actual curve type for calculating the accuracy of the discriminator. Given a specific curve type, a measure of how accurate the discriminator is at correctly selecting that curve type is measured and the resulting ROC curves relating to the FPR and TPR rates are plotted.

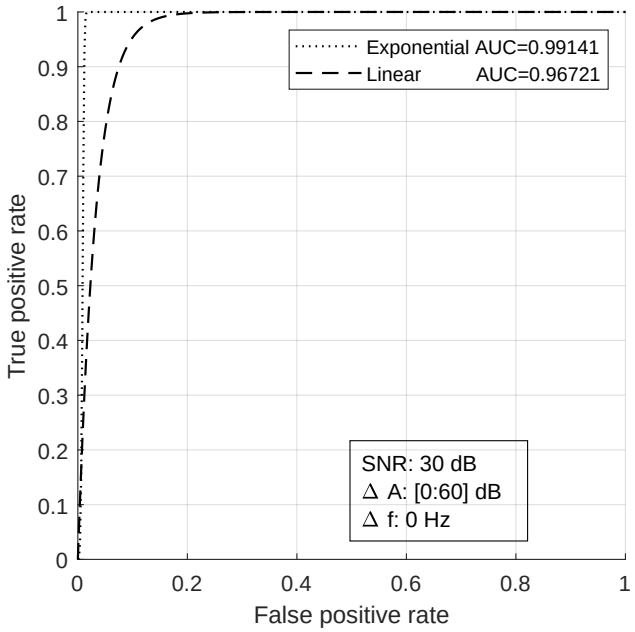


FIGURE 3.29: ROC plot for 2D estimate of envelope type. SNR: 30 dB, dA: [0:60 dB]

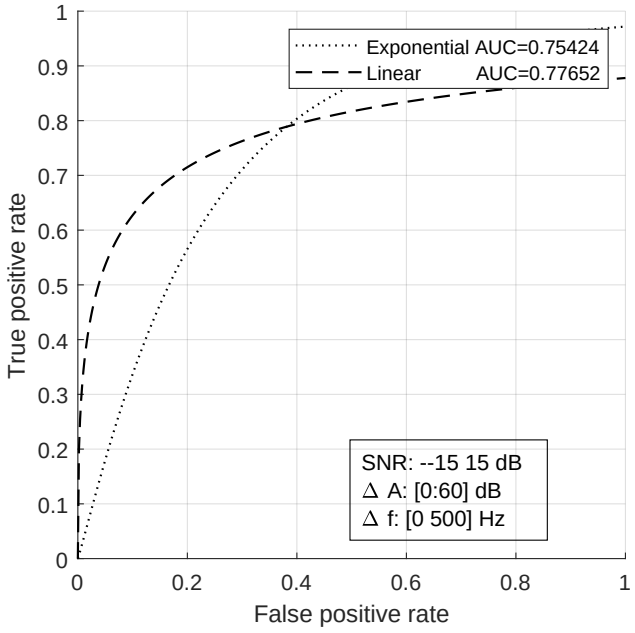


FIGURE 3.30: ROC plot for 2D estimate of envelope type. SNR: [-15:15 dB], dA: [0:60 dB], dF: [0:500 Hz]

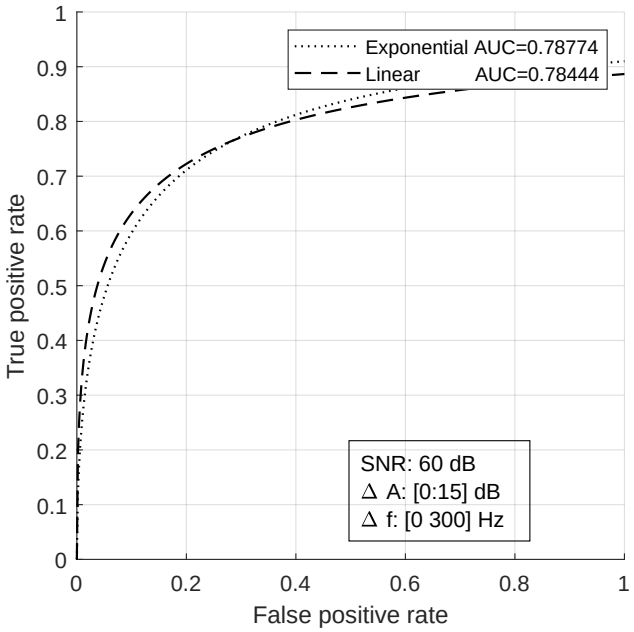


FIGURE 3.31: ROC plot for 2D estimate of envelope type. SNR: 60dB, dA: [0:15 dB], dF: [0:300 Hz]

Figures 3.29-3.31 show ROC curves for selecting the correct curve type between exponential and linear amplitude changes using a 2D lookup table, at differing levels of noise and other non-stationary values for change in amplitude and frequency. The tests are robust to added noise and to changes in frequency within a frame, up until a certain limit. Relative noise levels at -20 dB continue to provide fair results for exponential amplitude curve estimates but linear curve estimates fall below the random classifier range of 50% with added noise levels below 20 dB. This is unsurprising as high levels of noise added to a signal provide spurious magnitude second order difference estimates, especially for linear amplitude changes, as seen in Figure 3.16. Large changes in frequency can have an effect on the accuracy of the discriminator but even with changes of 300 Hz within a frame, the results seen in Figure 3.31 are well above the random classifier range with Exponential AUC = 0.787 and Linear AUC = 0.7848.

3.8.2 Cramer Rao Bound (CRB)

The Cramer Rao Bounds (CRB) are the theoretical lower bounds on the variance of estimation error for an unbiased estimator. They are used as a reference in experiments involving noise where high levels of noise are added down to no noise and the parameters estimated are evaluated in regards to the variance of the error.

The performance of the phase difference estimator, implemented as a lookup table, is evaluated in the presence of noise and in terms of the variance of the estimation error with respect to the CRB which is defined as the limit to the best performance achievable by an unbiased estimator. The DESAM toolbox [178] is used to compare the phase difference estimator against reassignment and derivatives estimators which assume exponential amplitude change. The DESAM toolbox is a set of Matlab functions written by Sylvain Marchand and Philippe Depalle implementing non-stationary parameter estimation models such as reassignment and the derivatives method.

The phase difference methods derived for estimation of linear and exponential amplitude change are affected in terms of quality by the amount of zero padding employed. This can be seen in Figure 3.32 which shows the CRB for exponential amplitude change estimation using the phase difference method, with six times zero padding against 8x zero padding, for a sinusoid whose frequency is exactly at the centre of an analysis bin.

The phase difference (PD) method is compared to the reassignment method (R), estimated derivative method (ED) and to the theoretic derivative method (TD). The PD method is implemented using a lookup table of 200 entries spanning a range of -100 to 100 dB. In the experiments shown, the sample rate is set to 48000 Hz, N = 1025, and the signal-to-noise ratio (SNR) rang-es from 20 to 100 dB in steps of 5 dB. Frequency and starting phase are set up and distributed as in [151]. Amplitude change is tested separately for linear and exponential envelope types, with 11 entries uniformly distributed between -75 and 75 dB. Smaller ranges of amplitude change ([-20, 20] dB) were also tested and showed similar results but are not shown here for compactness.

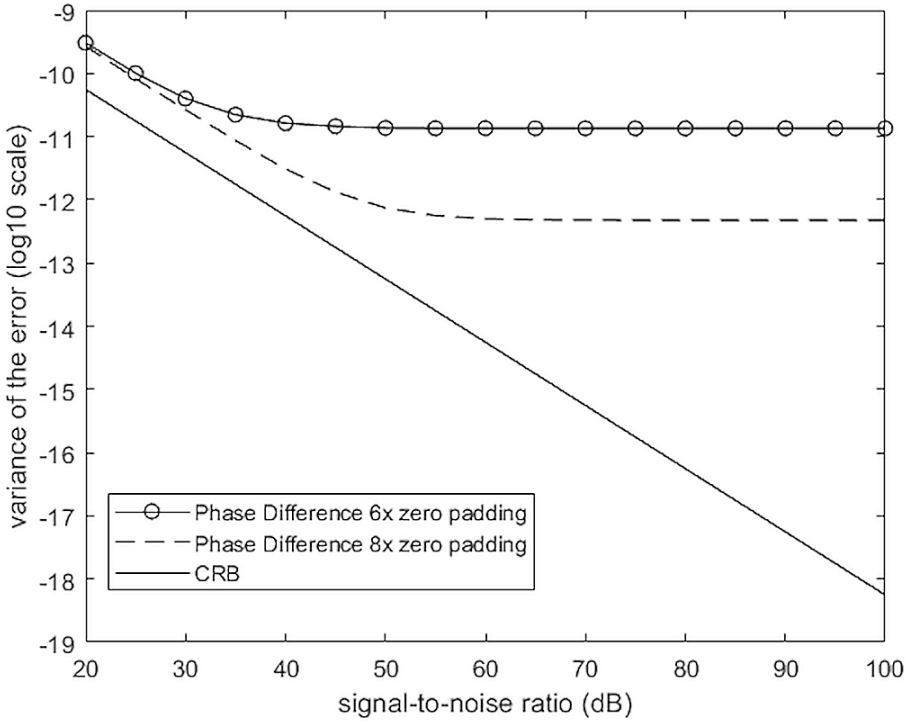


FIGURE 3.32: Increased zero-padding improves ΔA results.

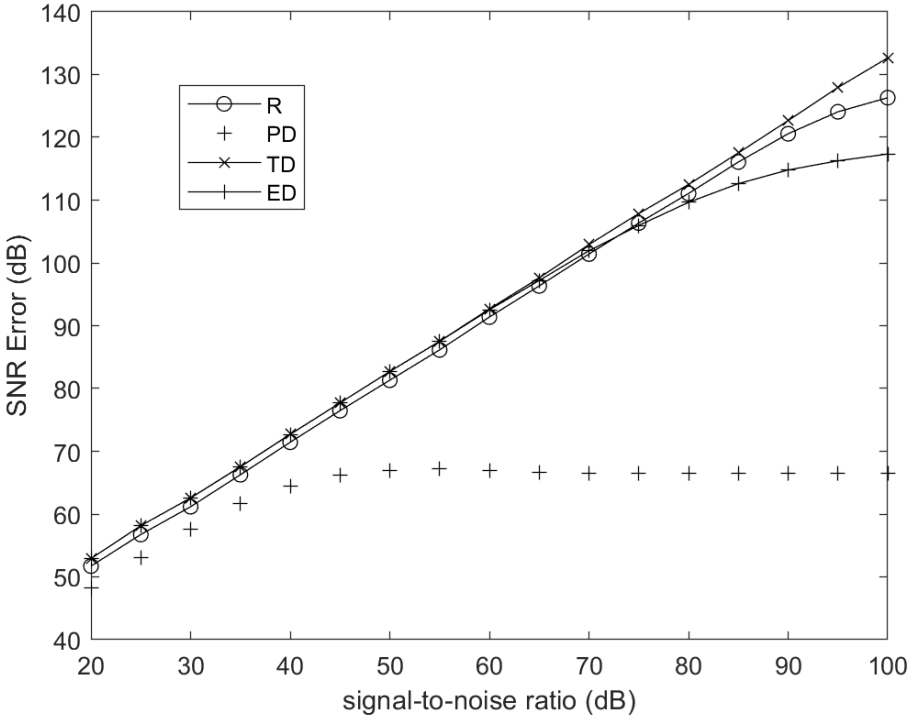


FIGURE 3.33: SNR plot of exponential amplitude change for reassignment, phase difference and the derivatives methods.

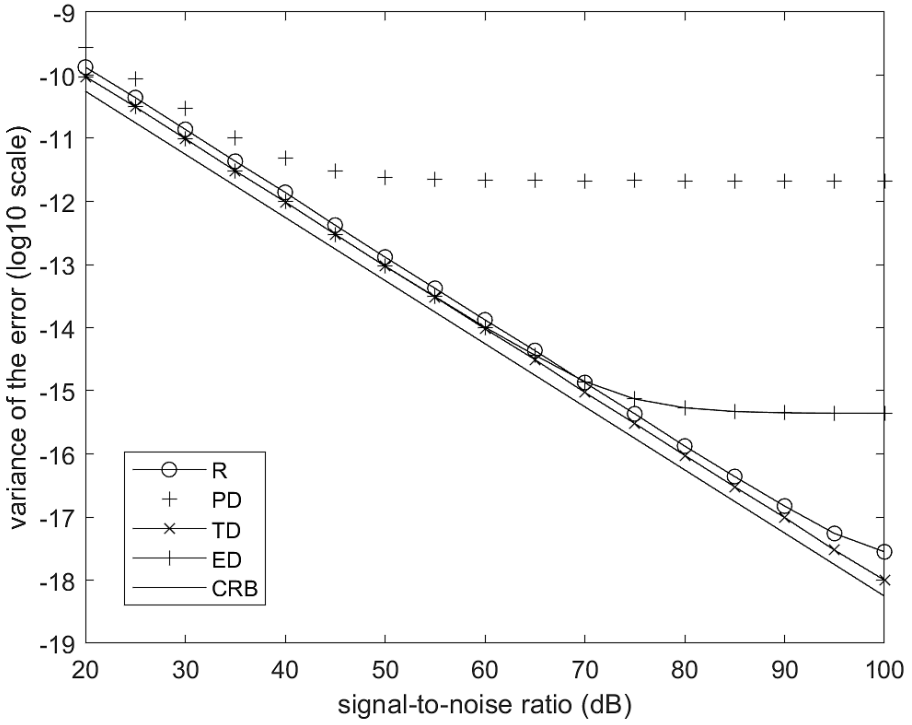


FIGURE 3.34: CRB plot of exponential amplitude change for reassignment, phase difference and the derivatives methods.

For each SNR, and amplitude envelope type, the above set of input signals is approximated with the mentioned methods with regards to estimation of the change in amplitude. The residual error of the approximated amplitude envelope and the variance of the error are compared to the CRB for each of the methods. The PD measure performs worse than other methods, showing a bias once the presence of noise falls below a certain level as can be seen in Figures 3.33 and 3.34. The residual of the error is also shown to reach a limit constrained by the lookup table size and amount of zero padding.

Figures 3.33 and 3.34 show that the lookup table method performs worse than reassignment and the derivatives methods and is constrained to a residual error of around 65 dB compared to other methods where the performance is better. In theory, the phase difference estimator is equivalent to time reassignment and should be capable of achieving similar results with larger lookup tables and more zero padding.

For linear amplitude changes, the performance of the phase difference method with use of a lookup table has similar results for the CRB and SNR measurements as with exponential amplitude change as can be seen in Figures 3.35 and 3.36. The results are better at estimating linear amplitude changes than the other methods which assume exponential amplitude change.

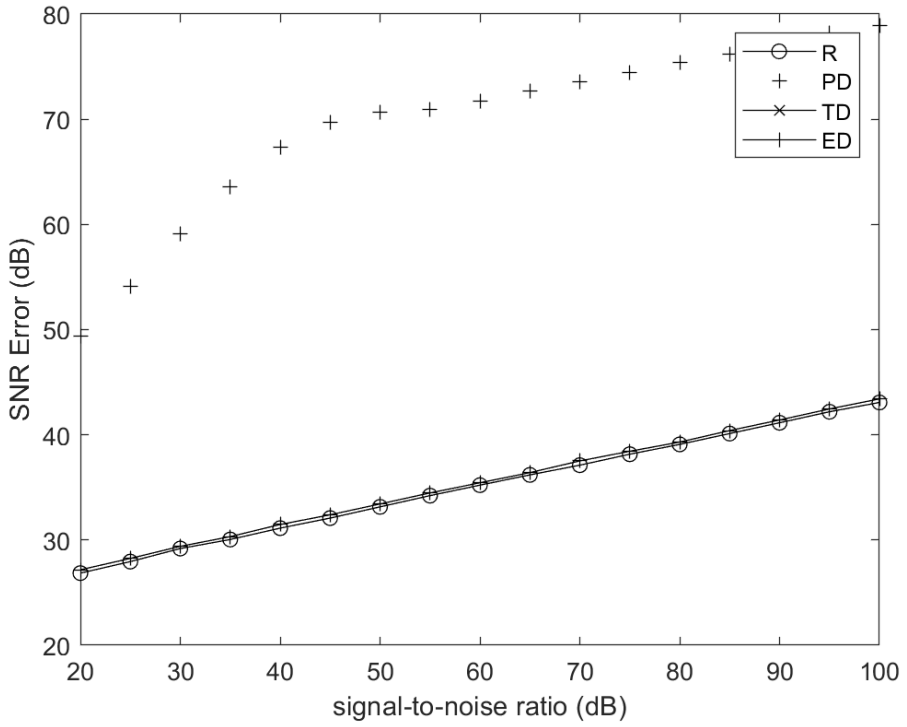


FIGURE 3.35: SNR plot of linear amplitude change for reassignment, phase difference and the derivatives methods.

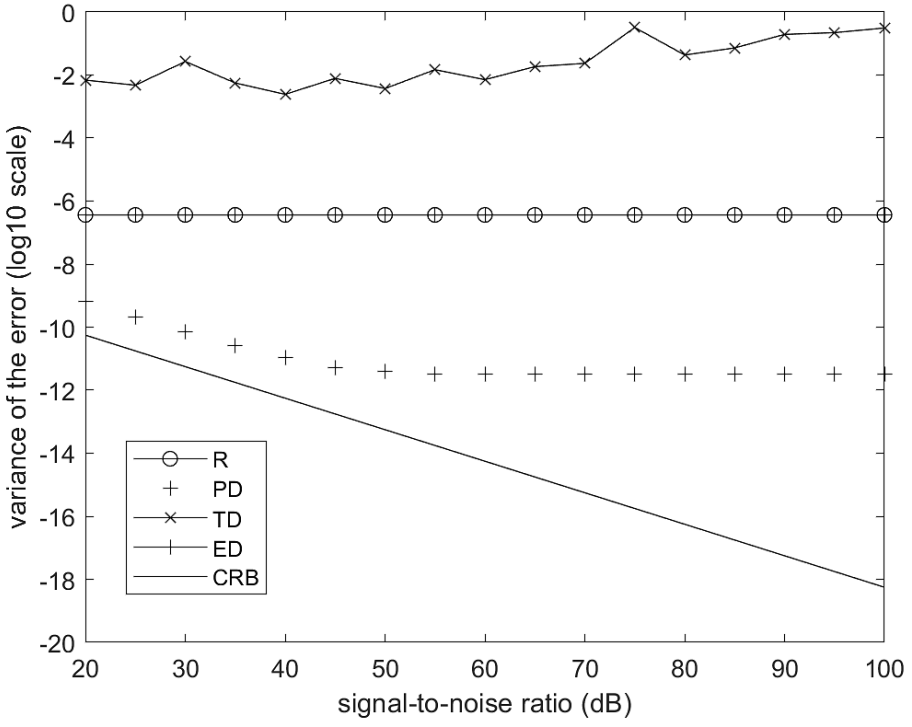


FIGURE 3.36: CRB plot of linear amplitude change for reassignment, phase difference and the derivatives methods.

Although the CRB results of the phase difference method show some bias and a limit in the accuracy and variance of the error relative to the amount of zero padding and table size, the method is the fastest out of all the methods tested as is shown in Table 3.1. The times recorded in Table 3.1 show the average time taken for each of the above-mentioned methods to estimate the change in amplitude in the test case presented. The phase difference measure is the fastest estimator tested, reassignment is the second fastest estimator but is still three times slower, the derivatives methods show the best results for exponential amplitude change estimation but is the slowest of the estimators.

Method	Time (seconds)
Phase Difference	0.000123
Reassignment	0.000379
Derivative TD	0.000596
Derivative ED	0.001512

TABLE 3.1: Measured computational time for each method tested.

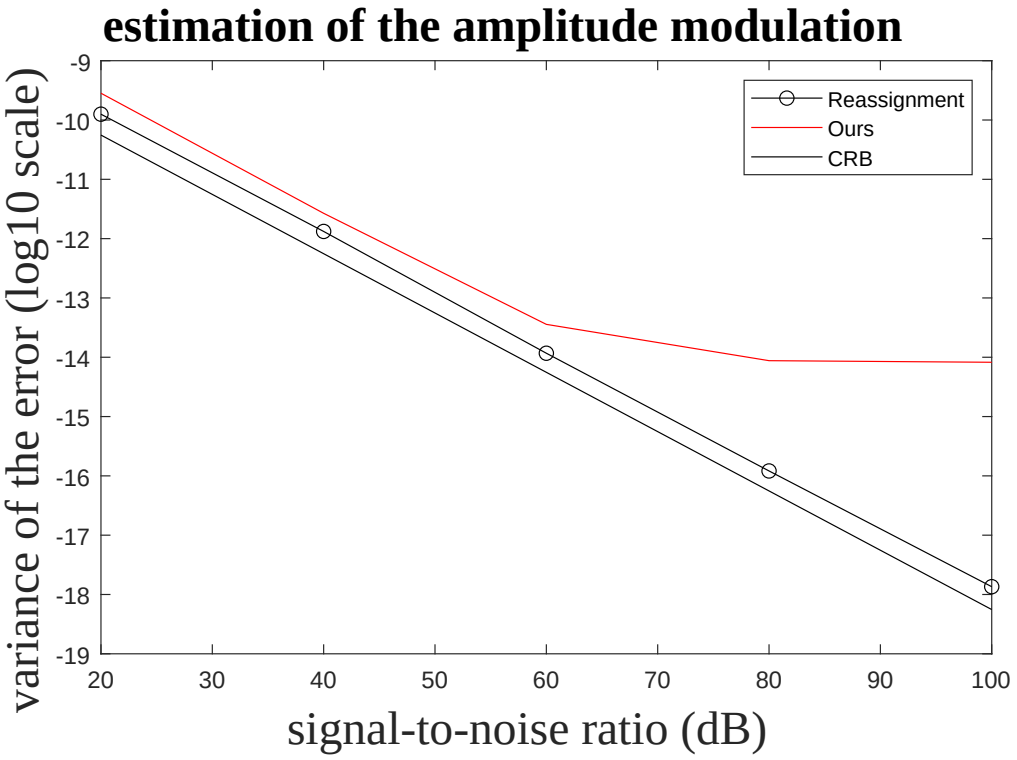


FIGURE 3.37: Decreased zero-padding worsens ΔA results.

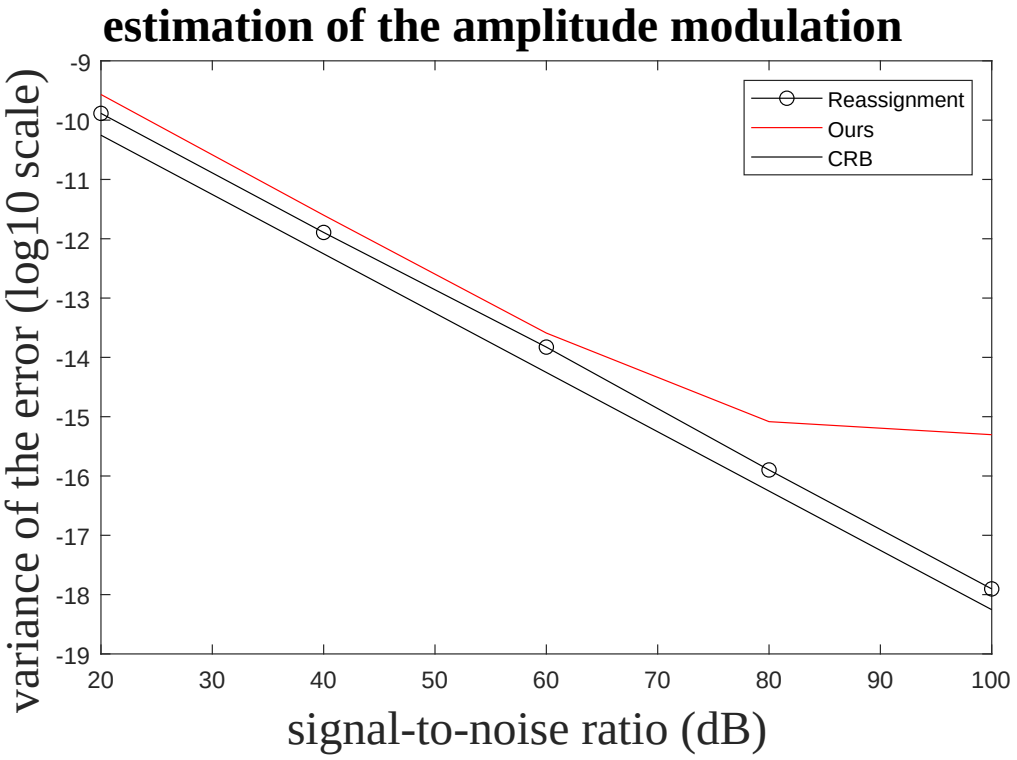


FIGURE 3.38: Increased zero-padding improves ΔA results.

3.9 Conclusions

Having a measure for linear and exponential amplitude change is not enough to distinguish between curve types. A second measure is required for the discrimination of linear and exponential amplitude envelopes. A method for generating linear amplitude change with the same shift in center of gravity and therefore same phase difference measure as an exponential envelope has been presented. This allows for the examination of the magnitude spectrum around a peak which showed a change in the concavity of a peak between the two amplitude curves. The second order derivative of the magnitude, can be used as a measure of concavity around a peak. In practice an expected value of the magnitude difference is compared to the measured value to select the curve with the highest correlation. Having these two different measures therefore allows one to distinguish between linear and exponential envelope curve types. The method used to approximate amplitude change is compared in terms of performance against the Cramer Rao Bound with different methods. Ultimately, we can see that these different methods perform better when their assumption of exponential amplitude change is correct, but they perform worse when their assumption of amplitude change is in-correct.

Chapter 4

Non-Stationary Modeling using Modelled Pursuit

> Reflecting a general pragmatic disposition towards "everything and the kitchen sink", modellers began to experiment with a variety of ad hoc tweaks before, during, and after training to improve the performance of deep networks.
>
> *The Routledge Handbook of the Computational Mind [179]*

4.1 Introduction

Chapter 3 examined a method for distinguishing between linear and exponential amplitude change from the decomposition of a signal into sinusoidal components using a Fourier basis. Chapter 5 examines the use of time-frequency analysis with Wavelet basis for the decomposition of the signal which has not been well represented by Fourier basis functions, as these generally provide a poor representation of signals well localised in time. This Chapter examines the use of signal adaptive expansion of the Fourier basis functions for providing a more accurate signal representation. The use of an adaptive decomposition using a Fourier basis of sinusoidal atoms which include estimates for change in amplitude are used to model non-monotonic amplitude change and transient components which are well localised in time. The Derivative (GDM) and Distribution Derivative (DDM) Methods have been shown to model complex amplitude changes by extending the order of the polynomial used in the complex amplitude and log-AM/FM model described in Section C.1.1.

However, the possibilities and limitations of using an overcomplete representation from non-overlapping single frame estimation methods has not been examined in detail.

Section 4.2 introduces 'Matching Pursuit' which forms the foundation of the algorithm adapted by 'Modelled Pursuit' described in Section 4.3, detailing the adaption of this approach in a segmented non-overlapping framework for modelling non-stationary sinusoids. Future research directives of 'Guided Modelled Pursuit', which show positive results from initial investigations are presented in Section 4.4. Section 4.5.11 investigates the modelling and transformation of transient signals using MoP. Section 4.5 provides an in-depth analysis of the use of an over-complete representation using MoP in the case of monotonic and non-monotonic amplitude change, and the effect that time stretching and pitch shifting causes on the output of an MoP model. Finally Section 4.5.1 demonstrates the results of the model on a number of different non-stationary signals compared with other established methods.

4.2 Matching Pursuit

Matching Pursuit (MP) was initially introduced by Mallat and Zhang [11]. MP is an iterative adaptive approximation algorithm that decomposes any signal into a linear expansion of waveforms. The algorithm is described as 'greedy' as at each iteration the algorithm "makes the best local improvement to the current approximations in hope of obtaining a good overall solution" [180]. A dictionary of unit-norm vectors known as atoms are constructed for comparison with the input signal. A 'complete' dictionary spans the entire signal space, the dictionary is known as 'redundant' if the atoms are linearly-independent. An 'overcomplete' dictionary refers to a set of atoms that spans the signal space but includes more functions than is necessary to do so, making it linearly dependent, meaning an atom within the set can be expressed as a linear combination of the other atoms within that set.

MP chooses the atom that best represents the signal from a redundant dictionary of functions at each step, selecting the atom which has the highest 'inner product' product with the signal. These waveforms are chosen in order to best match the signal structures. As described by Mallat and Zhang, "Matching pursuits are general procedures to compute adaptive signal representations" [11, 181]. MP is also know as a 'Sparse Representation' (SR) of a signal by aiming to represent the data with as few atoms as possible from an 'overcomplete' dictionary [182]. Mallat likens the dictionaries used in MP to dictionaries used in natural languages. "A richer dictionary helps to build shorter and more precise sentences. Similarly, dictionaries of vectors that are larger than bases are needed to build sparse representations of complex signals" [166]. Meaning that a more accurate and more compact

signal representation can be found, which captures a greater range of time-frequency patterns, by using larger dictionaries composed of a wider range of richer functions.

MP decomposes a signal into elementary building blocks such that a signal s is represented as a linear combination of expansion functions g_M from a dictionary D and $D\alpha$ represents the weights (α_m) applied to each of the dictionary elements g_m.

$$s = D\alpha, \quad D = \left[\begin{array}{cc} g_1 & g_2 \cdots g_m \cdots g_M \end{array} \right] \tag{4.1}$$

An iterative approach of decomposing a signal into a linear sum of a set of vectors, obtained from a redundant dictionary $\mathbf{g}$, is achieved by calculating the 'inner product' of the signal with all of the elements contained within the dictionary. The 'inner product' is a generalised way of multiplying two vectors together which returns a single scalar value representing how alike the two vectors are.

In a vector or Euclidean space $\mathbf{R}^n$, the inner product is given by the dot product $\mathbf{X} \cdot \mathbf{Y}$

$$\begin{aligned} &< (\mathrm{x_1}, \mathrm{x_2}, \cdots, \mathrm{x_n}), (\mathrm{y_1}, \mathrm{y_2}, \cdots, \mathrm{y_n}) > \\ &\qquad = \sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{x_i y_i} \\ &\qquad = \mathrm{x_i y_i} + \mathrm{x_i y_i} + \cdots + \mathrm{x_n y_n} \end{aligned} \tag{4.2}$$

The first element with the highest correlation is selected as the initial atom within the decomposition, and then subtracted from the original signal giving a residual signal, similar to the sinusodial plus residual approach of Spectral Modelling Synthesis [71].

The initial stage of the MP decomposition can be expressed as:

$$\mathbf{s} = \langle \mathbf{s}, \mathbf{g}_{\gamma_0} \rangle \, \mathbf{g}_{\mathbf{y}_0} + \mathbf{R}^0 \mathbf{s} \tag{4.3}$$

Here $\mathbf{s}$ is the signal at all time instants, $\mathbf{g}_\gamma$ is a dictionary element (of index γ, with $\boldsymbol{\gamma_0}$ the index of the zeroth chosen atom) and $\mathbf{R}^0\mathbf{s}$ is the residual after approximating $\mathbf{s}$ with the zeroth index atom. The process is repeated on the residual giving a decomposition with a second term:

$$s = \langle s, g_{\gamma_0} \rangle \, g_{\gamma_0} + \langle s, g_{\gamma_1} \rangle \, g_{\gamma_1} + R^1 s \tag{4.4}$$

This process continues until a stopping point is reached (such as the energy of $\mathbf{R^n s}$ reducing to a specified threshold, or after a certain number of iterations). $\mathbf{R^n s}$ approaches 0 with increasing n.

For a M term expansion MP decomposes the signal s

$$s = \sum_{n=1}^{M} \alpha_n g_{\gamma_n} \tag{4.5}$$

where α_n is the expansion coefficient (scalar related to atoms similarity) applied to the dictionary function g_{γ_n} at iteration n. MP decomposes a signal into a sequence of inner products $\alpha_n = \left\langle \mathbf{R}_{\mathbf{x}}^{n-1}, \mathbf{g}_{\gamma(n)} \right\rangle$ and indexes γ_n. The task at each stage n is to find the atom g_{γ_n} that minimizes the Euclidean (L2) norm of the residual signal in a similar way to a least-squares optimal solution which minimizes the sum of the squares of the residuals, thus choosing the atom which minimises the energy remaining in the residual.

$$g_{\gamma_n} = \arg\min \left\| R^{n+1} \right\|^2 = \arg\min \left\| R^n - \alpha_n g_{\gamma_n} \right\|^2 \tag{4.6}$$

This can be rewritten as selecting the atom with the highest inner product of the signal as:

$$g_{\gamma_n} = \arg\max \left| \langle g_{\gamma_n}, R^n \rangle \right| \tag{4.7}$$

$$\alpha_n = \langle g_{\gamma_n}, R^n \rangle \tag{4.8}$$

with α_n being the scalar result of the inner products result.

Atoms which remove the most energy are therefore the first to be selected and will automatically have the highest weight (scalar result from the inner product calculation) α_n, with successive atoms receiving lower values at each iteration.

The algorithm will continue for M iterations giving

$$s \approx \sum_{n}^{M} \alpha_n g_{\gamma_n} = \sum_{n=1}^{M} \langle g_{\gamma_n} \mid R^n \rangle \, g_{\gamma_n} \tag{4.9}$$

The number of iterations carried out is typically determined by predefined stopping criteria dependent on underlying signal characteristics. Typical criteria include a fixed number of iterations or continuing until after the M_{th} residual's energy falls below a certain threshold.

MP has been used extensively in the past in Spectral Modelling as well as Time-Frequency analysis. High resolution of sound using MP has been proposed in [183, 184]. Atomic decomposition and overcomplete signal representations have been presented by Goodwin et al [86, 100, 141, 169, 185]. Exponentially dampened sinusoids have been used with MP in [169, 186], and been shown to be more effective at modelling transient signals than Gabor atoms in [100, 141, 185]. MP has also been used in parametric audio coding [86, 187–190] and in Transient modelling [105, 191], where it has received further attention with regards to modelling transients for parametric audio coding in [192, 193].

4.3 Modelled Pursuit

Modelled Pursuit (MoP) is introduced by Wells in [14] where it is applied to decomposing Impulse Response (IR) recordings into a sinusoidal model. The modal decomposition of IRs into a sum of exponentially decaying sinusoids presents a novel approach for manipulating the characteristics of a captured reverberation model through the resynthesis (on a sample by sample basis) of the IR using additive synthesis after the manipulation of sinusoidal atom parameters.

Increasing or decreasing reverberation times are achieved by changing the decay rate of each sinusoid (taking into account that this modification is frequency dependant [194]), while the density of the reverberation model can be reduced by removing atoms or reducing their amplitudes. Reverberation density can also be be increased by adding additional atoms to the resynthesised sound with their frequencies scaled by $\sqrt{0.5}$ providing an irrational number which is the mean between two octaves. In general echo density (the number of reflected repetitions per second) increases with t^2 while mode density (modes per unit frequency) increases with f^2. Modes should be spread out uniformly, but not too regularly spaced, as this results in audible periodicity in the time-domain [45].

MoP is an iterative approach for the decomposition of a signal, similar to matching pursuit (MP), but instead of having a dictionary with a fixed set of atoms, MoP creates an atom, or multiple atoms, from "using parameter estimates derived directly from single frame estimation methods using the DFT." [14]

Estimates of the amplitude of the sinusoidal atom can be taken directly from the DFT which in effect results in a MP expansion with all expansion coefficients being set to to 1 and the amplitude of the sinusoid therefore replacing α. However another option is also presented and both methods compared in [14] where the estimated amplitude of the atom can be found by adopting the method used in MP by calculating the inner product of the residual and a 'energy-normalised' atom.

In either case the decomposition then becomes:

$$s = \sum_{n=1}^{M} g_{\gamma_n} \tag{4.10}$$

where α_n is assigned to the amplitude A_n of the n_{th} non-stationary sinusoidal atom g_{γ_n}:

$$g_{\gamma_n} = A_n(t) \sin\left(2\pi f_n t + \phi_n\right) \tag{4.11}$$

The causal implementation used in [14] sets the amplitude estimate A_n at the beginning of the frame at time $t = 0$, using information of the window shape in the Fourier domain from C.1.3. This non-stationary sinusoidal atom has time varying exponential amplitude $A_n(t)$ with the corrected start amplitude set at time $t = 0$ and a stationary frequency estimate f_n estimated using parabolic interpolation of the log magnitude spectrum.

The adapted non-causal implementation of MoP used in this thesis incorporates both linear and exponential amplitude change estimates as presented in Chapter 3. These methods uses zero-phase windowing and padding which results in the amplitude estimate being set to the center of an analysis frame at time $t = 0$, $|t| \leq \frac{1}{2}$ with the function centered around zero. If change in frequency estimates are incorporated into the model the n_{th} non-stationary sinusoidal atom g_{γ_n} is adopted from 3.1:

$$g_{\gamma_n} = A_n(t) \sin\left(\phi_n + 2\pi(f_n t + \frac{\Delta f_n t^2}{2T})\right), |t| \leq \frac{1}{2} \tag{4.12}$$

where $A_n(t)$ describes either a linear or exponential time varying amplitude envelope with a mid-point amplitude estimate at $t = 0$. ϕ_n is the mid-point phase estimate, f_n is the mid-point instantaneous frequency and Δf_n is the linear intra-frame frequency change.

The iterative nature of MoP decomposition and the selection of the spectral peak with the highest magnitude at each iteration results in components with the most energy being the first to be selected. The order of the atoms within the decomposition therefore reflects each atoms energy contribution to the signal as a whole, with the atoms ordered from strongest to weakest. This is the same as in MP where at each stage the atom which minimises the energy remaining in the residual is selected. In its simplest form, the dictionary used in MoP at each iteration could potentially contain only a single atom created from the single-frame estimates obtained from the DFT. An expansion of this is to create a dictionary compiled from a number (hundreds or thousands) of atoms, with a normal distribution of slightly random parameter values from this initial atom. Such an approach is more consistent with that of MP where each slightly varying atom is compared to the residual signal by taking the inner product and selecting the atom with the highest correlation. Such an extension of MoP is highly parallelizable and lends itself well to an implementation on a Graphics Processing Units (GPU). A further refinement to the algorithm based on Guided Matching Pursuit [12] can use the scalar results with the highest values to derive 'Guided Maps' for further refining the process using non-uniform distributions of random values for each parameter guided in the correct direction. The atomic decomposition of the signal iterates over sinusoidal peaks in the frequency spectrum, subtracting the best matching atom from the signal, resulting in a new residual signal with a reduction in energy at each iteration. This process is repeated until one of the following criteria is met:

1. There is an increase in energy in $\mathbf{R^n s}$ relative to the input signal s. "This criterion should not be met when the inner product method is used to estimate component amplitude, since residual energy decreases monotonically with MP. However, this situation can occur with direct estimation of amplitude particularly if a side lobe in the Fourier domain is mistaken for a main lobe, since this can cause extreme values of amplitude and/or frequency change to be estimated" [14].

2. The energy in $\mathbf{R^n s}$ is reduced to or falls below a specified threshold such as -60 dB (SNR reduction by one thousand times the initial value ($\mathbf{R^n s} < 0.001 \cdot \mathbf{R^1 s}$)), which is commonly used as a cut-off limit for audio [195].

3. The maximum number of atoms specified to model the input signal is reached such as 128 or 512 sinusoidal atoms, or some other maximum relative to a real-time constraint. In practise 256 atoms was found to strike a good balance between performance and quality.

Th atomic decomposition of the signal is based on spectral peak selection in a similar way to Spectral Modelling Synthesis, where the peak with the highest magnitude is selected. A dictionary of atoms modelled on the parameter estimates derived directly from the information around the selected peak in the DFT are created and each atom then compared to the residual signal. The atom which maximises the inner product with the residual signal is selected, and subtracted from the residual to create a new residual signal. This process is repeated using new DFT information from recalculating the STFT on the new residual signal. The resolution and amount of information returned from the STFT will have an affect on the quality and accuracy of the modelled atoms compared to the components embedded within the original signal. The size of the analysis window, windowing and the effect this has on spectral leakage have been discussed in Sections 2.3.3, 2.4, and 2.5.3. Stationary signals in the Fourier Domain will experience some form of spectral leakage when the frequency does not coincide with the trigonometric basis vectors used in the DFT. Windowing and the choice of window reduces the spectral leakage but requires overlapping frames (OLA) 2.4.5 which results in the averaging out of the results from the magnitude spectrum which "reduces the variance of the measurements" [196]. The choice of the analysis frame size which affects the frequency resolution of the DFT, the amount of overlap and windowing all contribute to the quality of any sinusoidal models decomposition. Quasi-stationary signal components which have monotonic amplitude and frequency modulations and are well isolated in frequency will have a minimal bias on parameter estimates. Signals with large or non-monotonic changes in frequency and or amplitude will result in the information being spread out in the Fourier Domain. Transient signals with fast attack and/or decaying elements will also be widely spread out in the Fourier Domain with more energetic components possibly masking and therefore reducing the accuracy of the estimates of these weaker components. The amount of energy spread is determined by the amount of change, with larger changes causing a greater spread across bins in the DFT. This spread of energy over multiple bins in the DFT is shown in Figures 4.1 and 4.2, where changes in amplitude and frequency are shown to affect the magnitude response from the broadening of the main lobe in relation to these changes. The position of the spectral peak does not shift in the presence of either monotonic amplitude or frequency change. The effect of only one of these non-stationarities only changes the width of the main lobe and the instantaneous amplitude. Figures 4.3 and 4.4 show similar results to figures 4.1 and 4.2 but without magnitude normalisation.

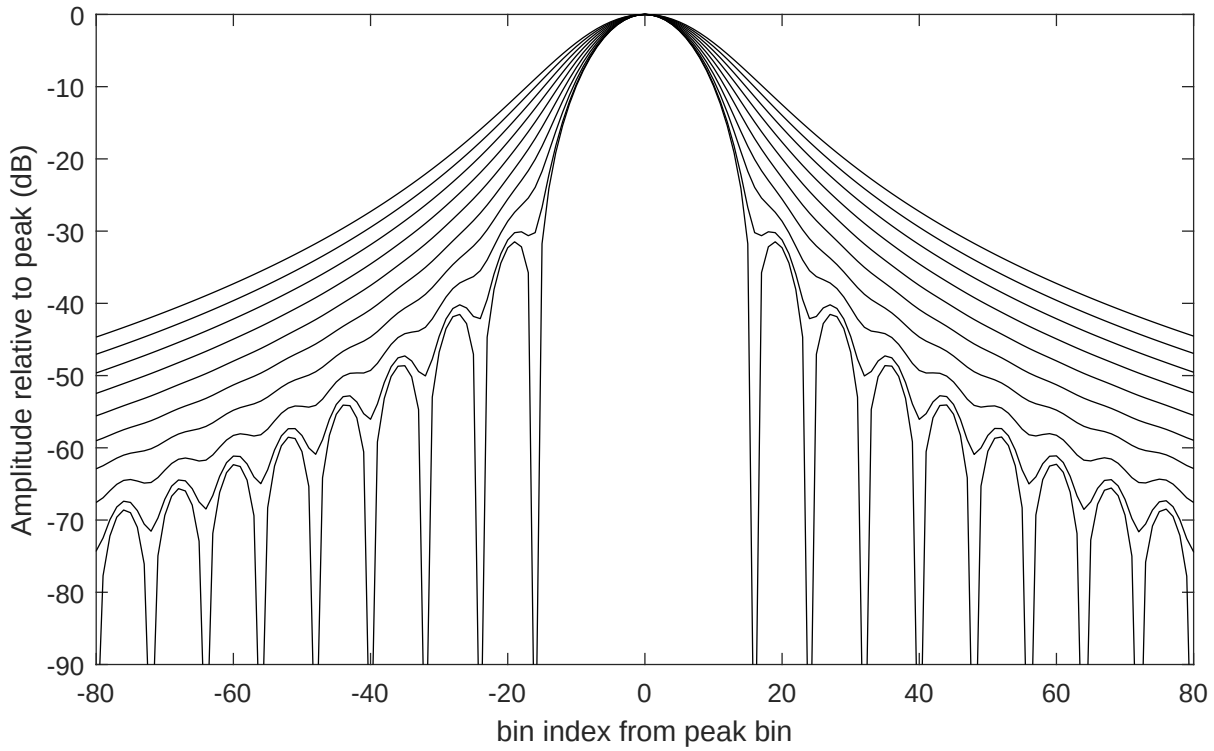


FIGURE 4.1: Normalised magnitude response of the Hann window multiplied by an exponentially changing amplitude function. The amplitude change is in 10 dB increments from 0 to 90 dB.

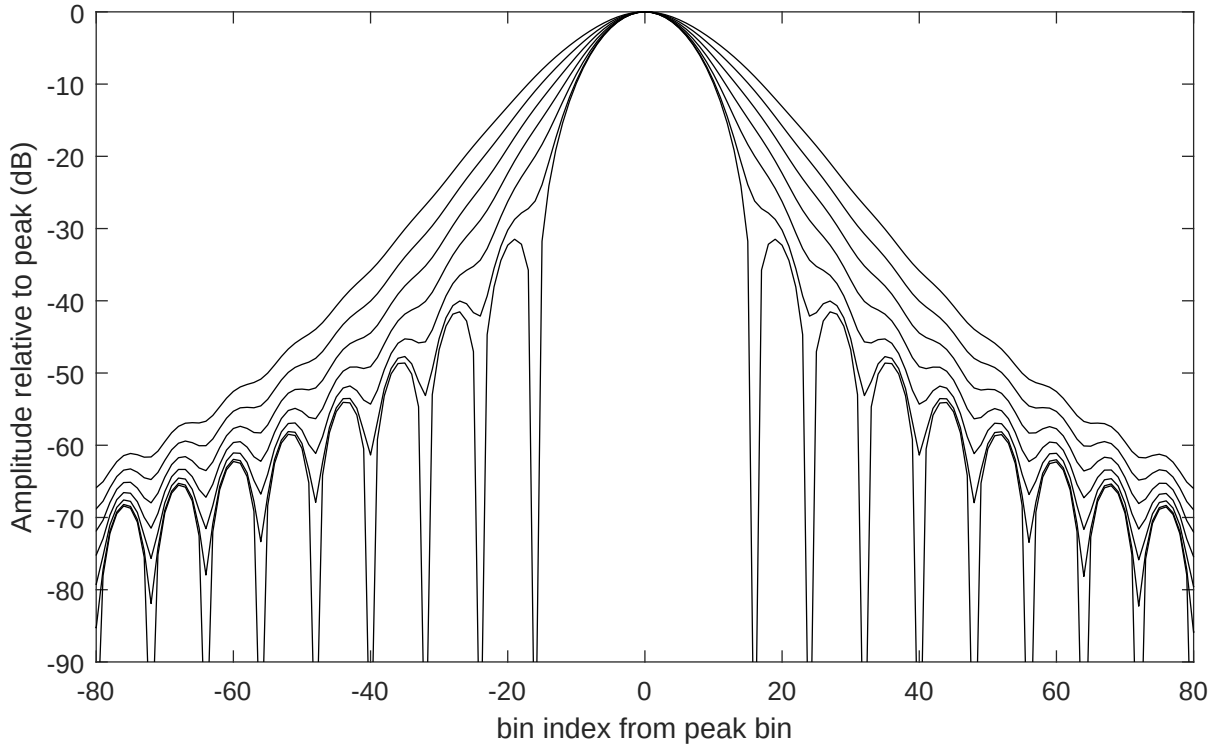


FIGURE 4.2: Normalised magnitude response of the Hann window linear chirp. The chirp rate increments in steps of 46.875 Hz (bin width × FFT zero padding factor) from 0 to 281.25 Hz

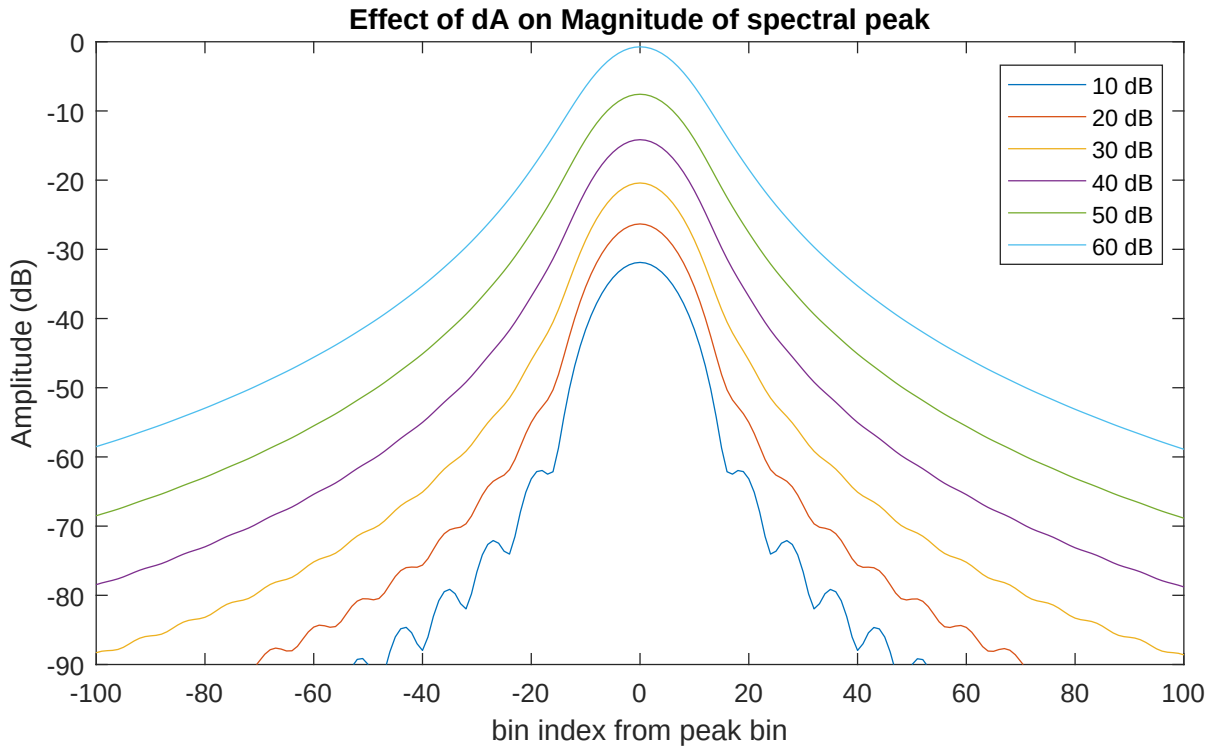


FIGURE 4.3: dA effect on mag peak

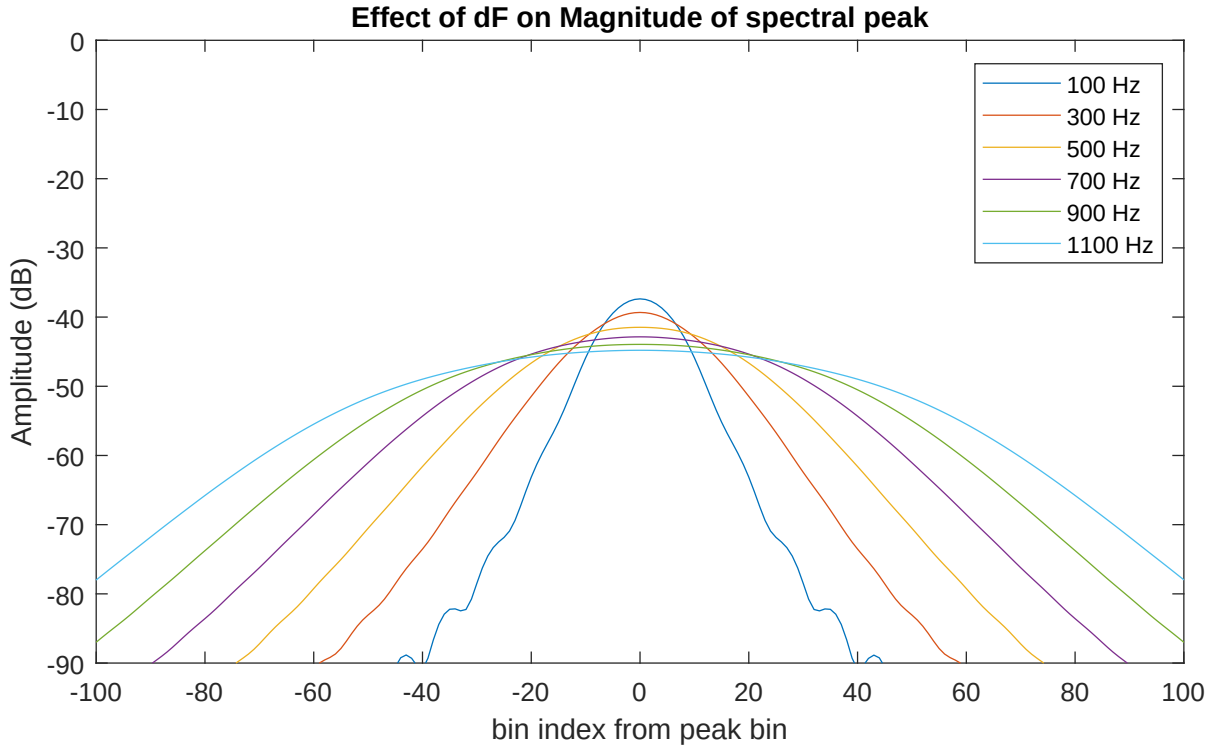


FIGURE 4.4: dF effect on mag peak

Figures 4.5 and 4.6 display similar results but plotted along the frequency axis. This displays that the spectral bin's position is not affected when only a change in amplitude, or a change in frequency is applied.

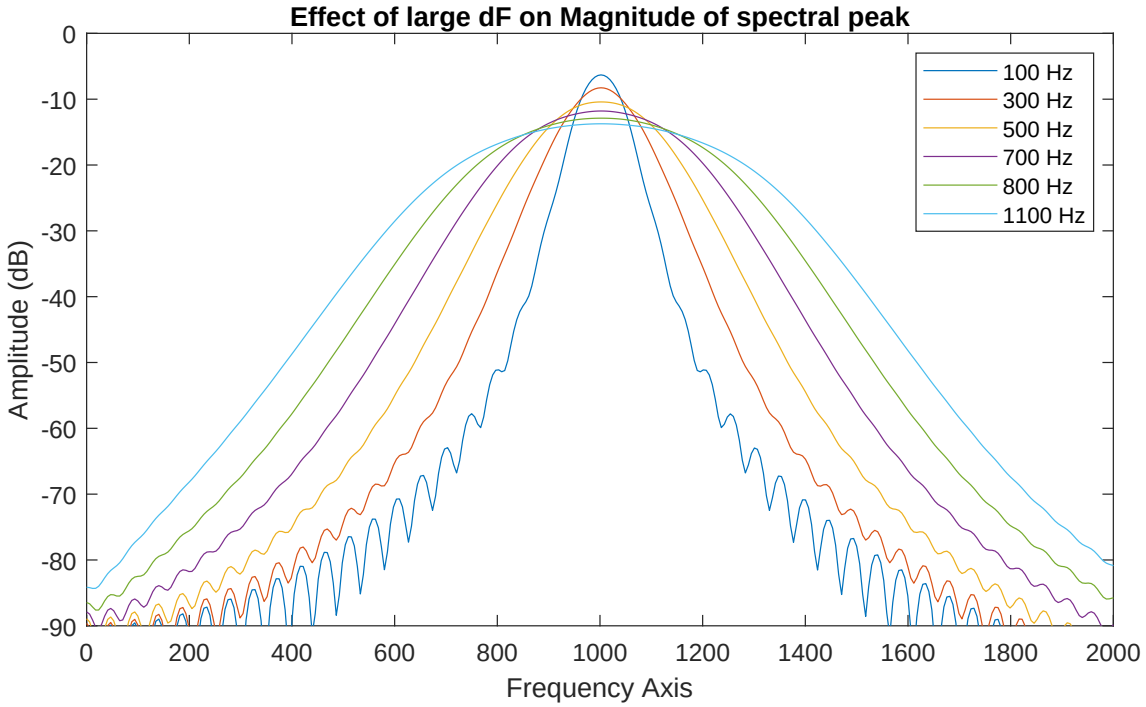


FIGURE 4.5: Large dF effect on mag peak

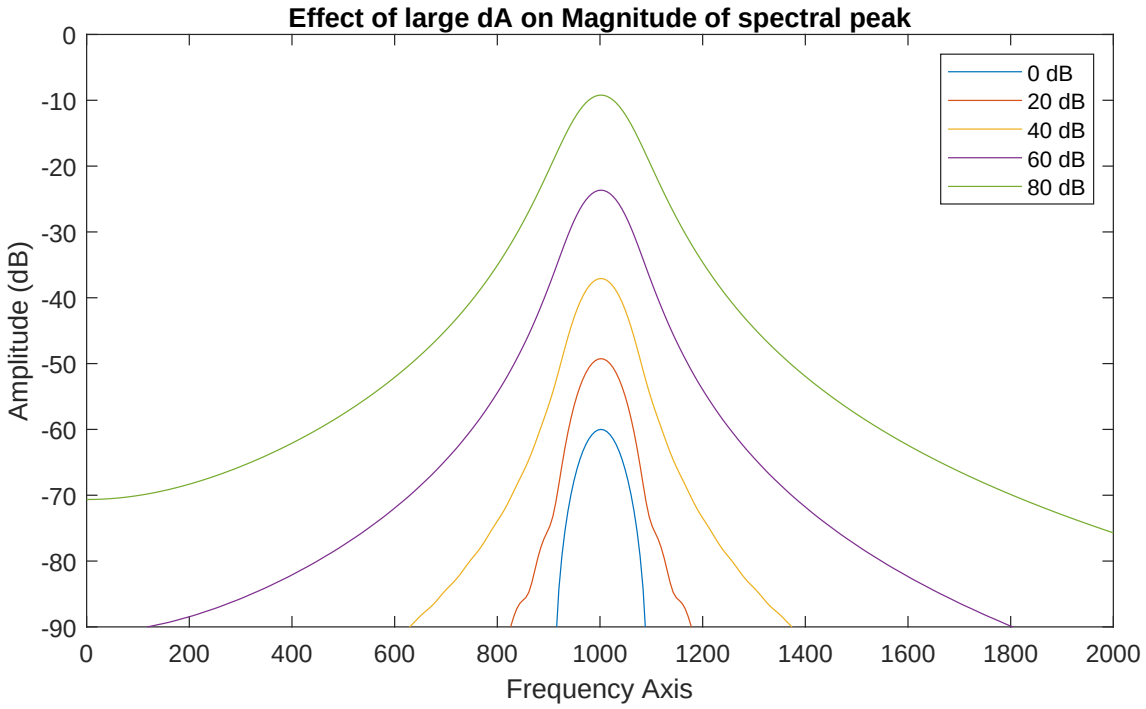


FIGURE 4.6: Large dA effect on mag peak

Figures 4.7 displays the effect of applying both amplitude and frequency change has on the magnitude spectrum around a sinusoidal peak. The peak frequency bin's index k is aligned and positioned together at bin 0 on the x-axis, with bin indexes before and after shown relative to the peak centered at 0.

While Figures 4.8, 4.9, and 4.10 are plotted along the frequency axis which clearly shows how the combination of both amplitude and frequency change shifts the peak of the sinusoid along the frequency axis leading to biased estimated of instantaneous frequency. The test signals have been generated using initialising parameters at the middle of the frame, and as such the frequency at the middle of the frame remains centered around the frequency bin. Change in amplitude also doesn't effect the position the sinusoidal peak. However, the combination of amplitude change and frequency change and the resulting energy and center of mass, and its distribution across the frame, shifts the sinusoidal peak along the frequency plane.

Figures 4.9 and 4.10 compare the amount of shifting in comparison to large changes in amplitude against large changes in frequency. This displays that a change in frequency has a greater effect on the shift of the spectral peak compared to change in amplitude, as one would expect.

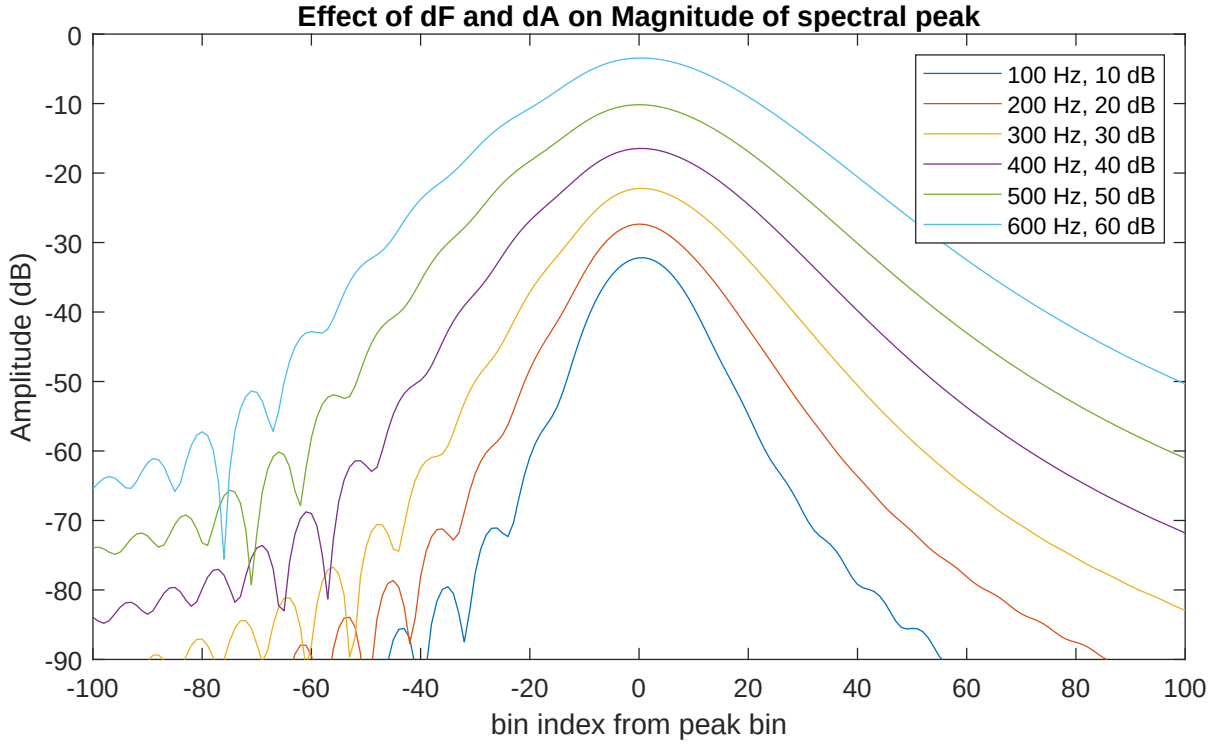


FIGURE 4.7: dF and dA effect on mag peak uncentered

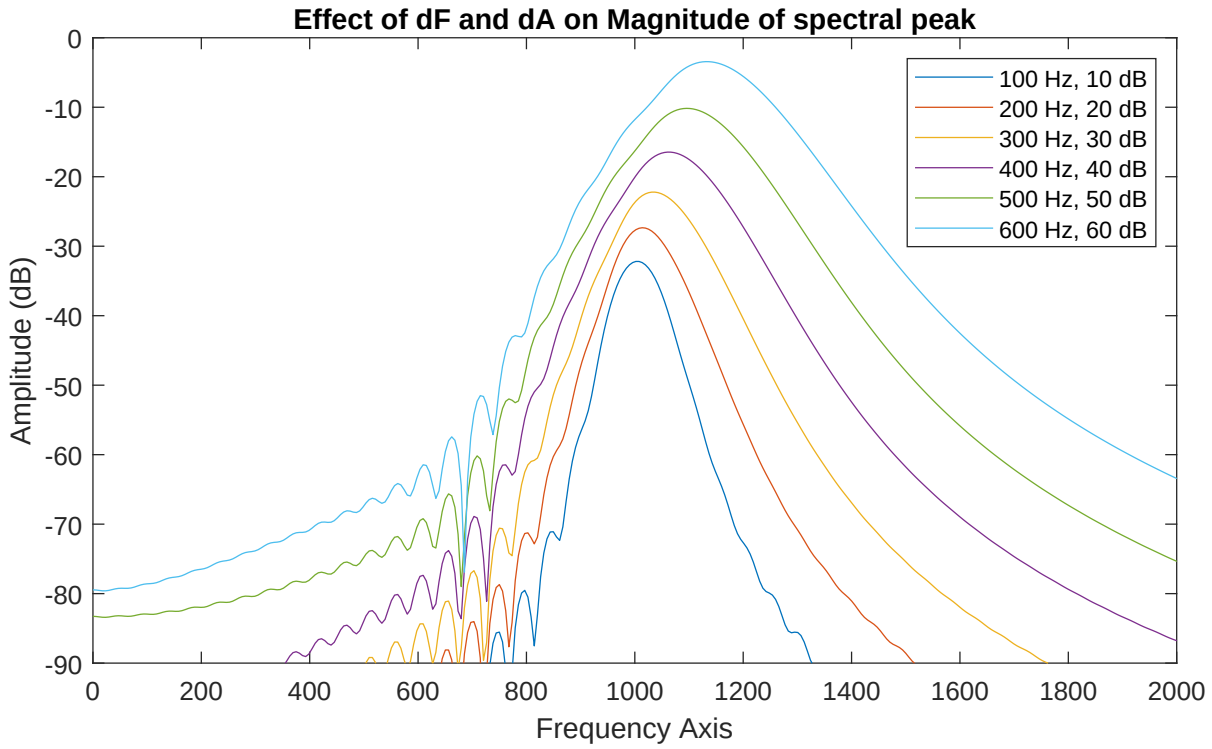


FIGURE 4.8: dF and dA effect on mag peak uncentered

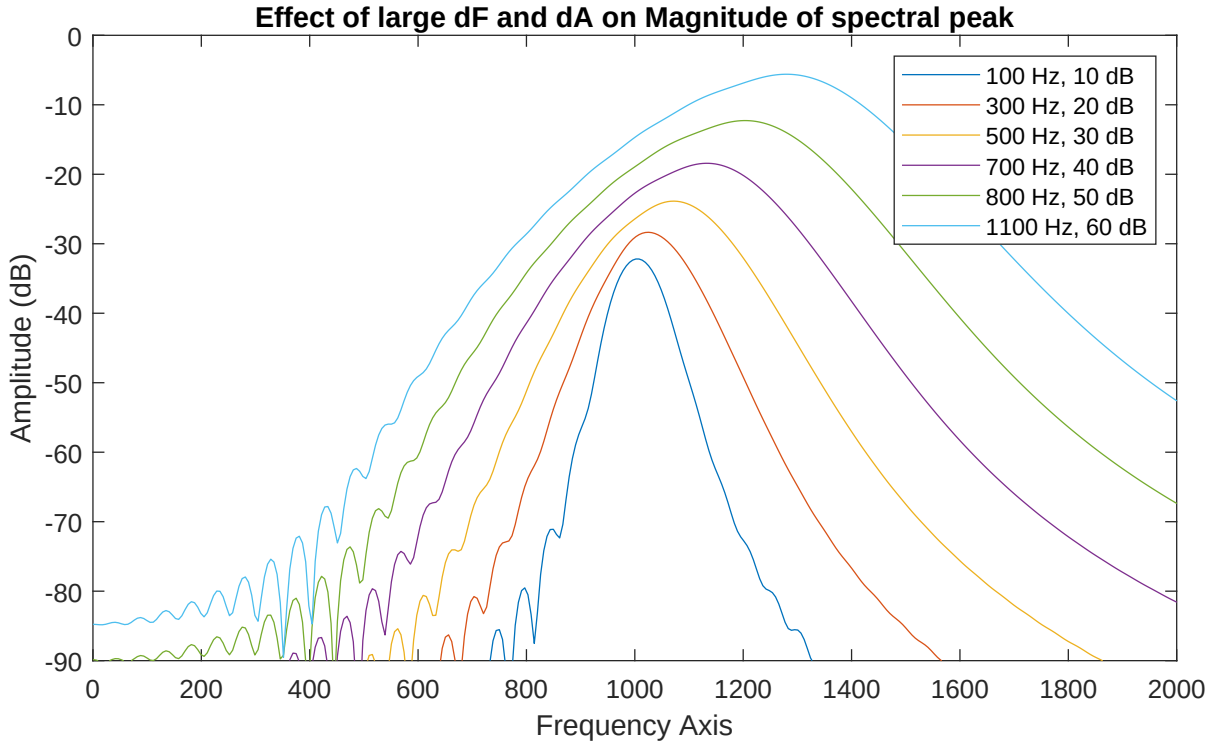


FIGURE 4.9: large dF and dA effect on mag peak

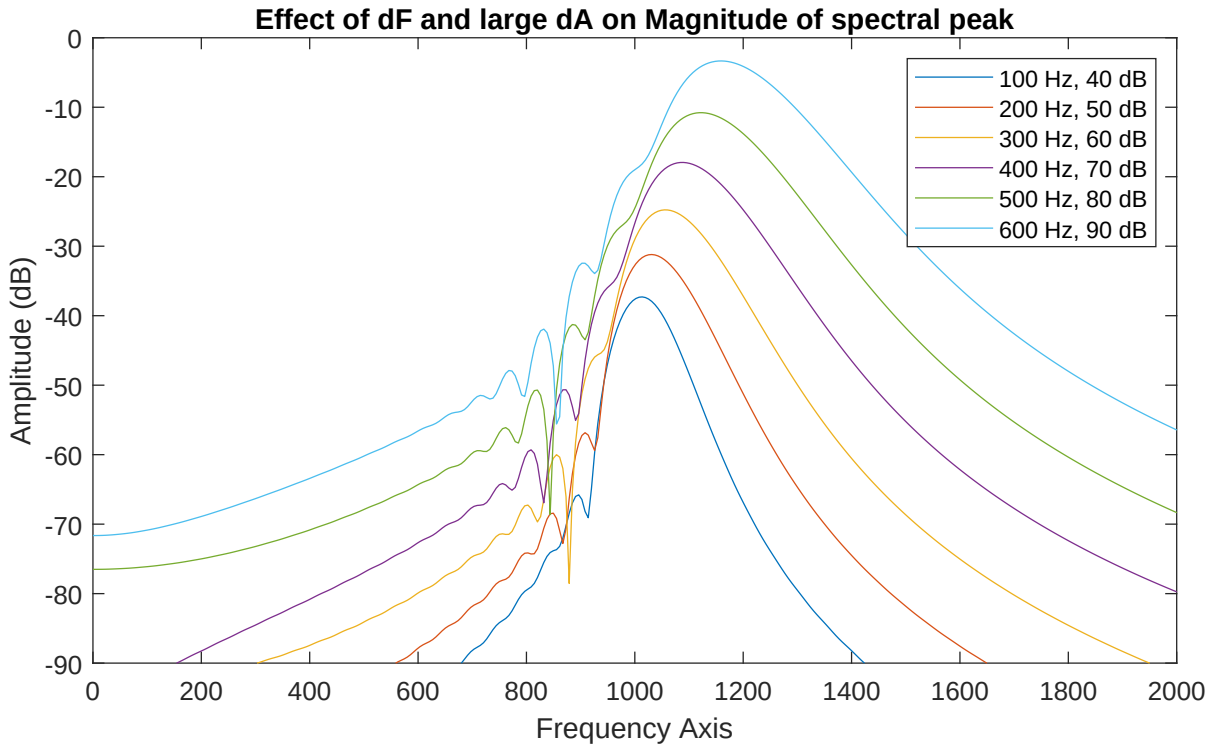


FIGURE 4.10: dF and large dA effect on mag peak

The first and second order differences of the phase across a sinusoidal peak were explored for estimating frequency change. The second order difference of the unwrapped phase provided the most accurate results. But wrapping the phase to between 0 and 2pi, and negatively wrapping the phase were also compared. Normalising the phase was also included in the tests where the peak phase value was initially selected as the normalization factor as in the magnitude second order difference measure. In [197] change of frequency in linear chirp signals is also estimated by approximating the second order derivative with a second order difference.

$$\hat{\alpha} \approx \frac{-jY^{\text{Hann}}(0)}{2}\left(\left(\frac{K}{2\pi}\right)^2 \frac{\Delta^2 Y^{\text{Hann}}(k)}{\Delta k^2}\bigg|_{k=0}\right) \tag{4.13}$$

Where $\hat{\alpha}$ is one-half the chirp rate in radians per sample, and K is the optionally zero-padded FFT length. The normalization applied in [197] uses the second order difference at bin k normalized by twice multiplying by $\frac{K}{2\pi}$.

The second and first order differences of the phase modulate after a certain frequency change, dependant on the analysis frame size. This limits the range of unique frequency change points within the curve and lookup-table. The second order PD measure for FM from -400 to 400 Hz is shown in Figure 4.11,

for a 1025 sample analysis frame. Frequency change estimates for the range of -300 to 300 Hz provide a unique measure of dF. Higher ranges of frequency change result in more than one possible result, which could potentially be estimated from a second lookup table which stores second order differences greater than the unique range.

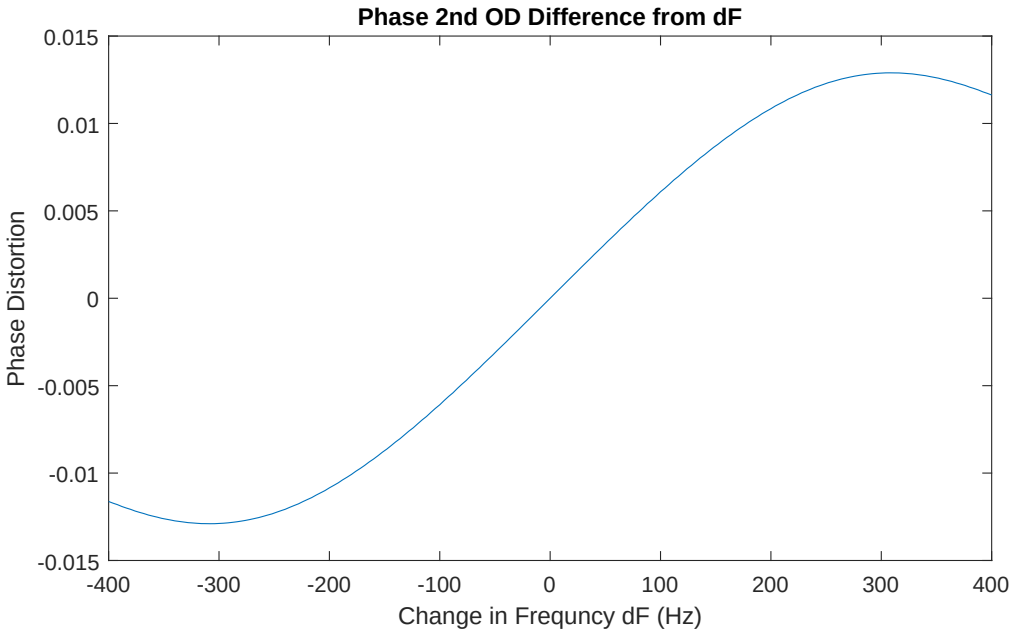


FIGURE 4.11: Unwrapped Phase Second Order Difference for dF

Additionally, as mentioned in [10], the phase value derived directly from Fourier Analysis when using zero-phase padding is not correct in the presence of amplitude and frequency non-stationarities. The correction of the estimate of the phase therefore requires two additional two-dimensional (2D) lookup tables (one for exponential and another for linear amplitude change) for correcting the phase value, when using zero-phase windowing and padding.

Typical impulsive and percussive components containing transients (such as kick drums) contain a large number of components with fast changes in a short time which could cause a significant bias on parameter estimates and the model as a whole when an analysis frame is not aligned in time and length of the percussive sound. One of the objectives of the the thesis is to examine if transients can be modelled by MoP with non-stationary sinusoidal components.

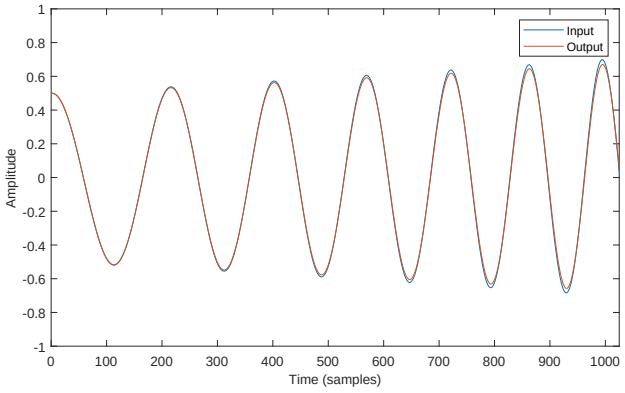
The windowing of signals reduces spectral leakage in the DFT, but at the cost of blurring spectral components of complex signals such as transient components. Modelling of transients is discussed in detail in the following Chapter 5 in Section 2.6.

The bias introduced from amplitude and frequency modulations is generally reduced by the iterative process of MoP as long as the parameter estimates are accurate, and "components removed at each stage will not bias subsequent analysis stages" [14]. The iterative nature of MoP is computationally expensive but the process of comparing the input signal with the compiled dictionary of slightly randomised atoms is highly parallelizable.

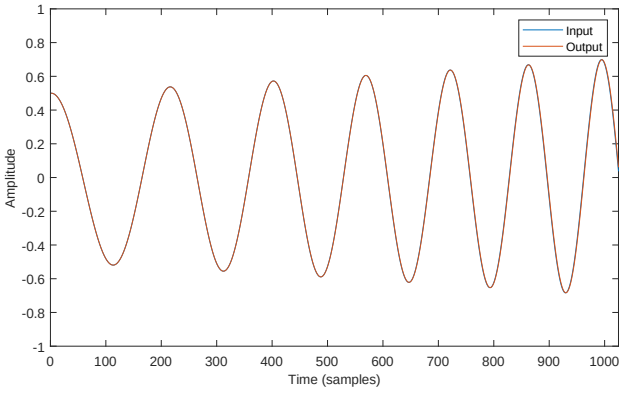
The following sections go over these approaches in more detail. Iterative estimation of parameters from single frame DFT methods are covered in Sections 4.3.1 and 4.3.2. Section 4.5.11 investigates the modelling of transient signals using an adapted segmented non-overlapping frame implementation of MoP, and Guided Modelled Pursuit is introduced in section 4.4. Section 4.5 examines the atomic decomposition of MoP applied to signals containing monotonic or non-monotonic amplitude change. Finally Section 4.5.1 presents the results of numerous tests using MoP to model a wide range of signals. MoP is compared against eaQHM and the Synthesis non-stationary signals used in [142], and against full EDM tracks in section 4.5.10. The limitations of an over-complete representation for performing time and pitch scale modifications are presented in 4.5.5 and 4.5.8. Creative transformation from performing modification MoP atoms is presented in 4.5.7.

4.3.1 Base Atom

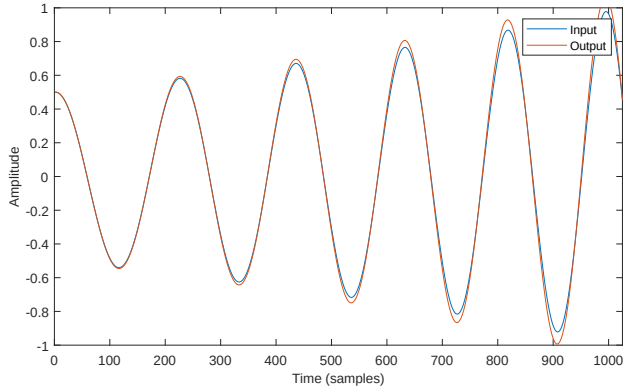
The implementation of MoP used in this thesis is based on Sinusoidal Modelling, as such the base atom for the compiled dictionary is that of a sinusoid; given in 4.12; with estimates for amplitude, frequency, phase and time varying parameters for monotonic amplitude change (which can be either exponential of linear). Estimates of amplitude and frequency are taken from parabolic interpolation due to its relatively low computational cost, but other phase based methods; which are generally more accurate; such as Reassignment, or the Derivatives method could be incorporated into the model at an additional cost. However, these methods assume exponential amplitude change. Monotonic amplitude change is estimated from the phase distortion measure across a peak with respect to the flat phase response of zero phase padded stationary sinusoid. The normalised magnitude second order difference as given in 3.21, is used to select either linear or exponential amplitude change. Change in frequency does effect the slope of the phase. In [159, 176] it is shown that phase is concave at a peak in the Fourier spectrum due to linear frequency change, and that an equal negative frequency change causes the phase response to be inverted, without affecting the magnitude response. Linear frequency change can be estimated from the second order difference of the phase as in [159], alternatively linear estimates of frequency change could be estimated using Reassignment or the Derivatives method at an additional cost.



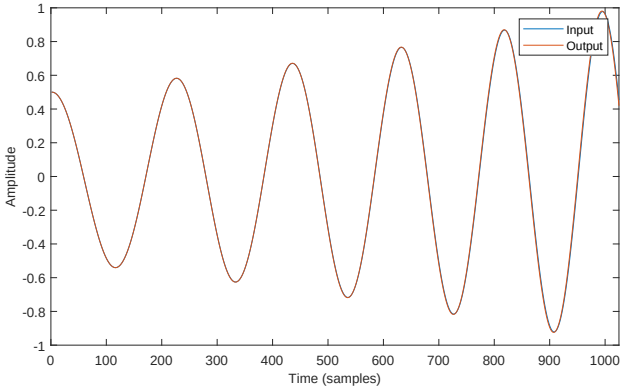
(A) Output of 200 Hz sinusoid with 175 Hz linear frequency change and 3 dB amplitude change without correcting phase measurement for estimating amplitude change



(B) Output of 200 Hz sinusoid with 175 Hz linear frequency change and 3 dB amplitude change with correcting phase measurement for estimating amplitude change



(C) Output of 200 Hz sinusoid with 75 Hz linear frequency change and 6 dB amplitude change without correcting phase measurement for estimating amplitude change



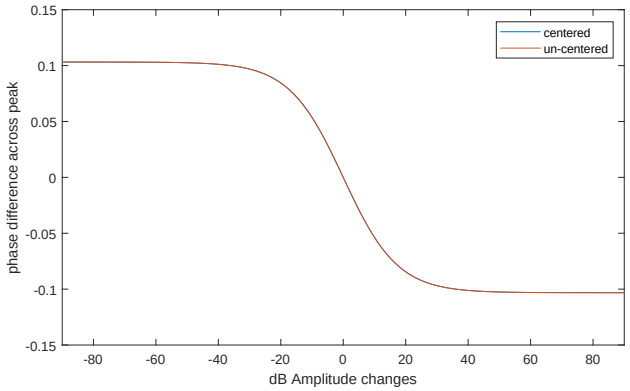
(D) Output of 200 Hz sinusoid with 75 Hz linear frequency change and 6 dB amplitude change without correcting phase measurement for estimating amplitude change

FIGURE 4.12: Examples of estimating change in frequency and change in amplitude with and without correcting phase difference measurement for correct amplitude change estimation of sinusoid @48 kHz

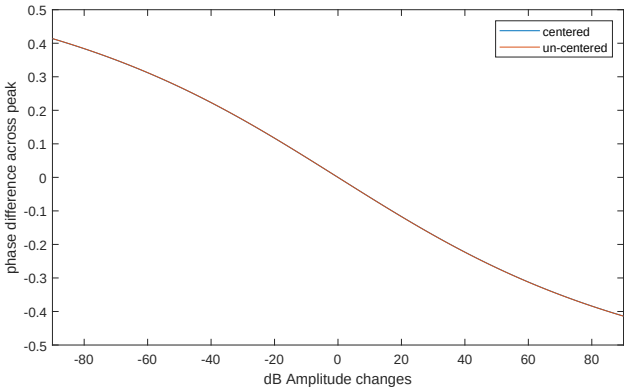
Frequency change affects the first order phase difference measurements and as such can lead to slight miscalculations in amplitude change. Initial investigations on using a 2D lookup table of second order phase difference measurements, with a range of frequency changes, proved promising for estimating frequency change. Figure 4.12a shows the results of this approach for estimating a 200 Hz sinusoid with 175 Hz change in frequency and 3 dB change in amplitude (exponential). In comparison Figure 4.12c displays a 200 Hz sinusoid with 75 Hz change in frequency and 6 dB change in amplitude (exponential). Neither of these two examples have taken the effect of frequency change on the phase into account, resulting in erroneous estimates for change in amplitude. An approach for correcting the amplitude estimate was examined by offsetting the phase difference measurement, by calculating and storing the the amount of phase difference introduced to a stationary sinusoids phase with the equivalent amount of frequency change. The results of the two examples with the first order phase difference measurement offset by the this value is displayed in Figures 4.12b and 4.12d.

This is a relatively simple method for estimating frequency change from second order phase difference measurements and correcting the estimate for amplitude change, by offsetting the first order phase difference from a pre-calculated measurement. However, two additional lookup tables are required, and this only works for frequencies at the center of an analysis bin. Deviation in frequency from the center of an analysis bin results in a slight miscalculation of the amount of frequency change from the 2D lookup table using second order phase difference measurements. This has a cascading effect on the rest of the calculations. Correcting estimates of amplitude change from first order phase difference in the presence of frequency modulations requires further investigation and is left as a future research directive.

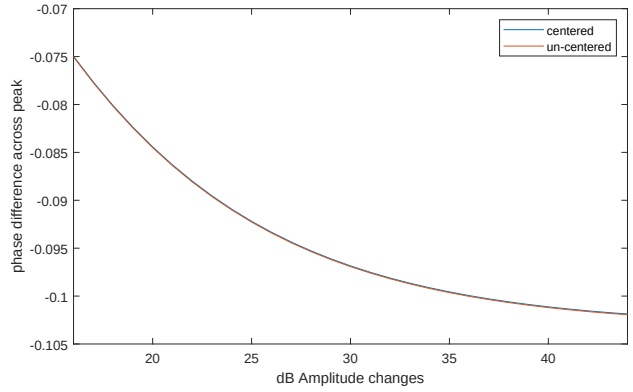
The estimation methods derived for linear and exponential amplitude change in Chapter 3 purposely left frequency change out with the intention of investigating and solving this problem as a future objective.



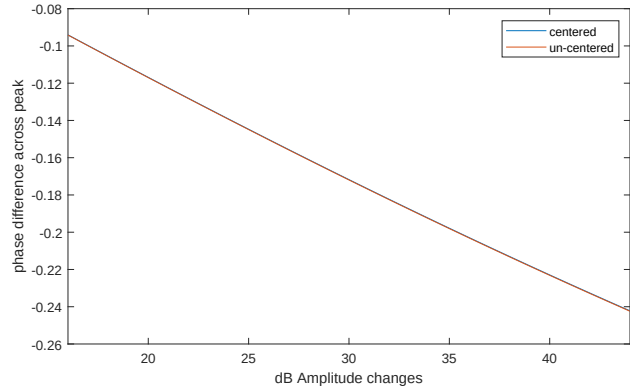
(A) Linear Phase Difference measurements between frequency centered at analysis bin and close to maximum deviation from the center of an analysis bin



(B) Exponential Phase Difference measurements between frequency centered at analysis bin and close to maximum deviation from the center of an analysis bin



(C) Zoomed view of Linear Phase Difference measurements between frequency centered at analysis bin and close to maximum deviation from the center of an analysis bin

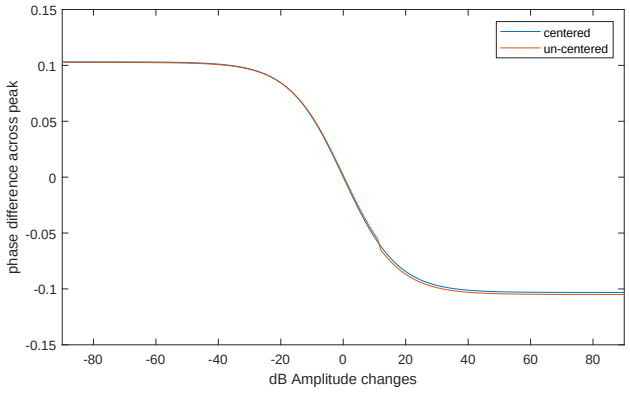


(D) Zoomed view of Exponential Phase Difference measurements between frequency centered at analysis bin and close to maximum deviation from the center of an analysis bin

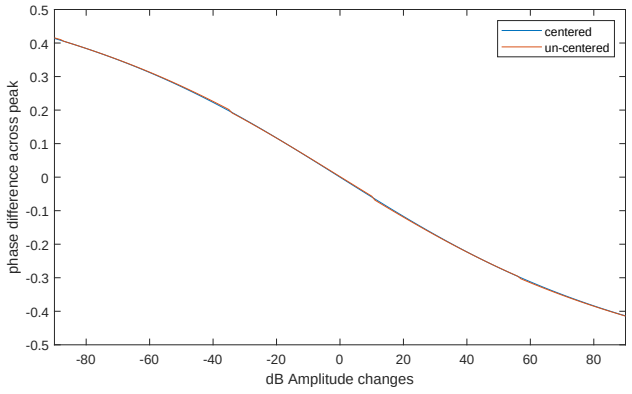
FIGURE 4.13: Examination of effect deviation in frequency from center of an analysis bin has on Phase Difference Measurements

The effect of deviation in frequency from the center frequency of an analysis bin on a sinusoid has been investigated. This showed negligible differences on the first order phase measurements and resulting calculations of amplitude change. Figure 4.13 shows the negligible difference a deviation in frequency from the center of an analysis bin has on first order phase difference measurements.

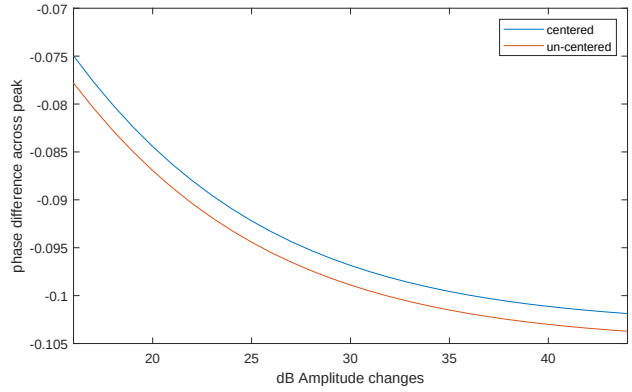
Frequency change however, has a far greater effect on first order phase difference measurements. Figures 4.14 and 4.15 display how a change in frequency of 50 Hz and 200 Hz respectively, effect the phase difference measurements. Higher changes in frequency are expected to have an effect on the phase and magnitude spectrum as mentioned previously in this chapter where Figure 4.2 showed the effect change in frequency has on the magnitude spectrum and spreading information across bin boundaries. The same is true for change in frequency on phase measurements which are clearly shown by the phase difference measurements across a sinusoidal peak.



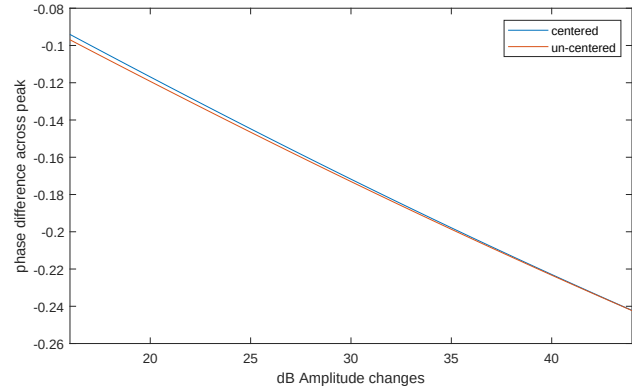
(A) Effect of 50 Hz Frequency change on Linear Phase Difference Measurements



(B) Effect of 50 Hz Frequency change on Exponential Phase Difference Measurements

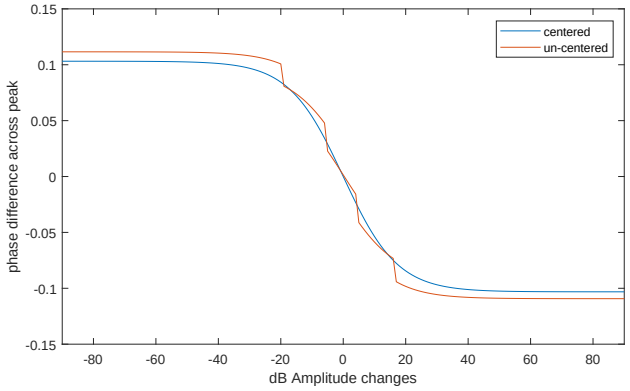


(C) Zoomed view of effect of 50 Hz Frequency change on Linear Phase Difference Measurements

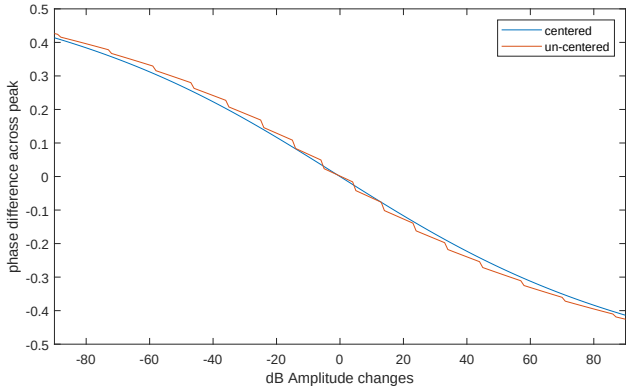


(D) Zoomed view of effect of 50 Hz Frequency change on Exponential Phase Difference Measurements

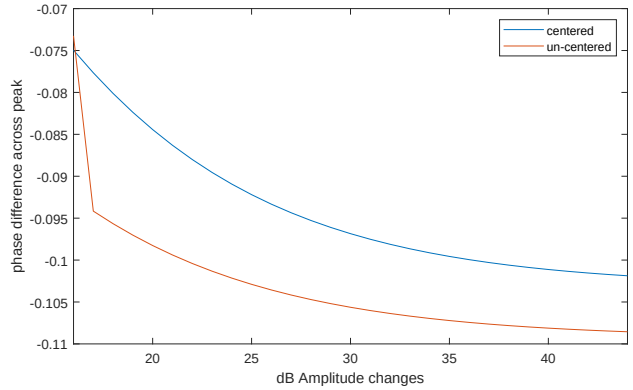
FIGURE 4.14: Effect of 50 Hz Frequency change on Phase Difference Measurements



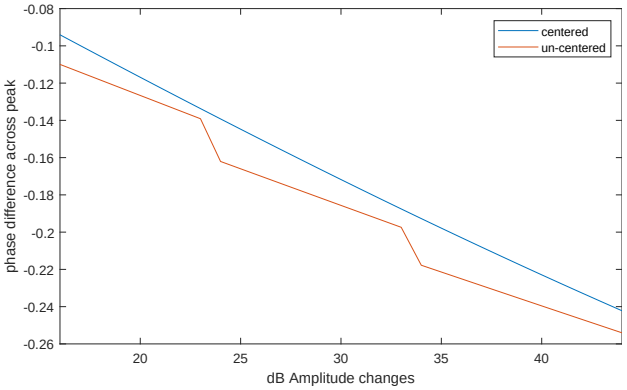
(A) Effect of 200 Hz Frequency change on Linear Phase Difference Measurements



(B) Effect of 200 Hz Frequency change on Exponential Phase Difference Measurements



(C) Zoomed view of effect of 200 Hz Frequency change on Linear Phase Difference Measurements



(D) Zoomed view of effect of 200 Hz Frequency change on Exponential Phase Difference Measurements

FIGURE 4.15: Effect of 200 Hz Frequency change on Phase Difference Measurements

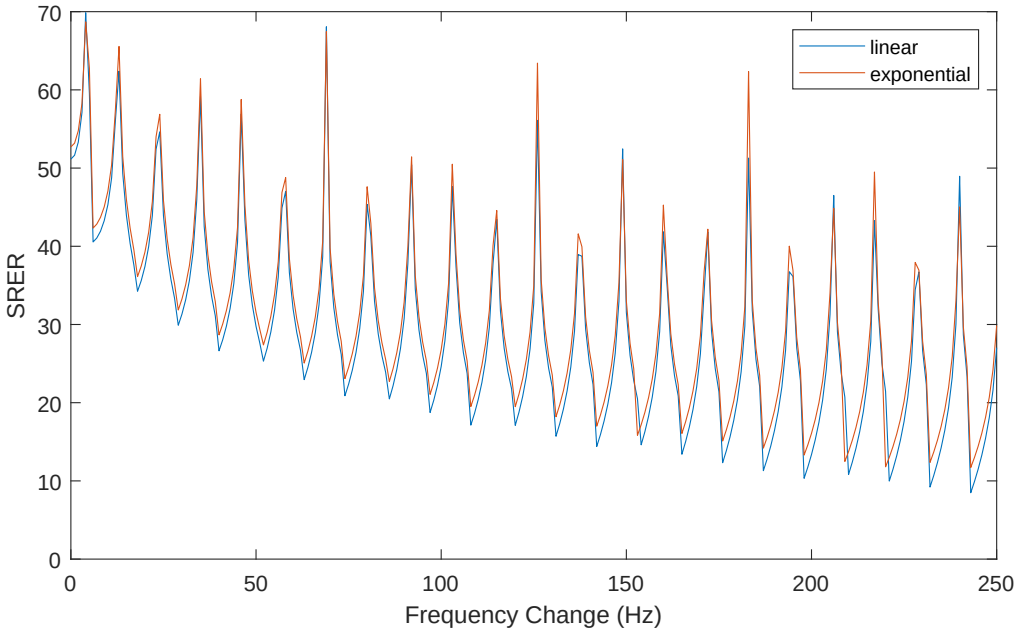
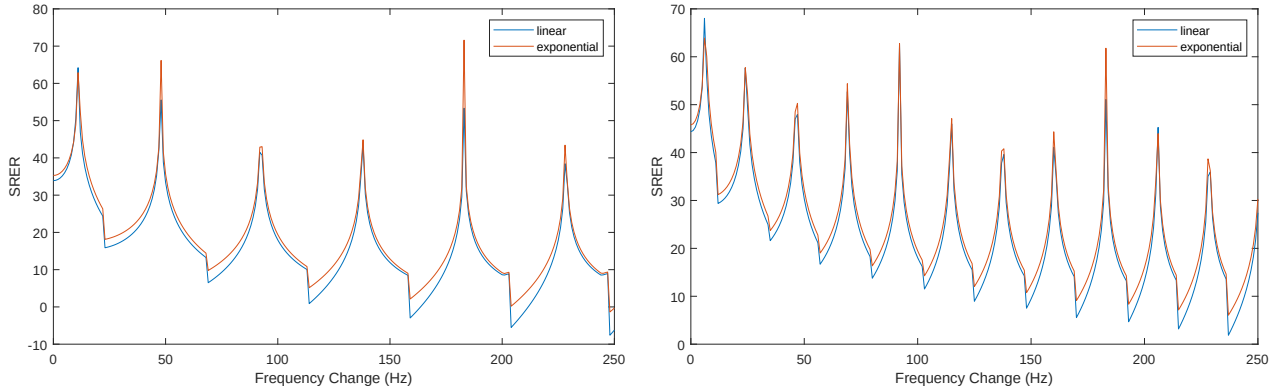


FIGURE 4.16: SRER of Amplitude Estimate for Linear and Exponential amplitude change of 6 dB, and change in frequency ranging from 0 to 250 Hz



(A) 2 time zero padding (B) 4 time zero padding

FIGURE 4.17: Effect zero padding and phase difference measurements SRER results from changes in frequency ranging from 0 to 250 Hz

Figure 4.16 shows the SRER results and the effect frequency change has on amplitude estimates. Figure 4.17 shows the effect zero padding has to these results. The distance between the maximum SRER peaks increases as zero padding factor decreases which indicates there are less multiples of frequency change where the phase difference measurements align with the correct values. Increasing zero padding decreases this with the fluctuations in SRER over frequency change increasing the rate of these fluctuations. The mean SRER result decreases at change in frequency increases.

In signal compression and sparse representations it is desirable to represent the signal with as few signal components as possible. However, in the presence of large amplitude and frequency changes, the combined effect on phase information can lead to a bias in parameter estimates. This is especially true for large changes in frequency within a single analysis frame and the effect this has on estimating amplitude change from the first derivative of the phase.

It is desirable to have accurate estimates of change in amplitude as well as change in frequency to have a model which is as accurate and sparse as possible, however, in the presence of biased estimates from large amplitude and frequency modulations, frequency change can be omitted from the model parameters and base atom, while still being able to model the signal in great detail, even in the presence of large amplitude and frequency changes within a frame. The model in such a case uses an an overcomplete representation of multiple sinusoidal components to model these changes.

The following Section 4.3.2 describes the iterative decomposition and option of using a dictionary of slightly varying atoms in more detail.

4.3.2 Iterative estimation from repeated DFTs

The segmented implementation of MoP uses a default frame size of 1024 samples, and a zero padding factor of 8 which results in an FFT size of 8192, and a frequency bin resolution of 5.8594 Hz when using a default sampling rate of 48 kHz. The threshold for ending the iterative process of decomposing the residual signal is set to -60 dB by default.

Two different segmented implementations have been investigated. Both implementations are implemented in a framework where analysis frames do not overlap. The one implementation is based on the analysis methods described in detail in Chapter 3 where non-causal first order phase difference measurements are used to estimate amplitude change. This implementation uses a Hanning window and zero-phase padding. Windowing the analysis frame and not overlapping the output frames is an uncommon approach but it has been shown that the analysis methods described in Chapter 3 are able to re-synthesise quasi-stationary sinusoidal components with monotonic amplitude change from a zero-phase padded and windowed input. The windowing has a known effect on blurring information at frame boundaries, as a result this implementation only models quasi-stationary sinusoidal components with monotonic amplitude change, ignoring frequency change for simplicity and leaving any remaining components within the residual. In this case a peak selection option traditionally used in SMS is more appropriate than an iterative decomposition, where the same sinusoidal peak can be modelled by multiple atoms, as is the case with the second implementation used for modelling transients. Due to the windowing process used in the non-causal system, the spectral leakage is reduced, this results in smoother peaks in the Fourier domain, minimising the need for multiple atomic decomposition's of a sinusoidal peak, which is the case with the causal implementation used for modelling transients in an overcomplete representation which uses a rectangular window.

The analytical equations for estimating the first order phase difference for exponential and linear amplitude change are given by 3.15 and 3.16 respectively. These are used to create two lookup tables for estimating amplitude change from measurements of the first order difference of the phase taken from the DFT. Another two pre-calculated tables; containing the magnitude second order difference measurements; with regards to changes in linear and exponential amplitude, are also stored for discriminating between the two amplitude curve types.

Two additional lookup tables are also required for correcting the phase parameter as the phase value derived directly from Fourier Analysis when using zero-phase padding is not correct in the presence of amplitude and frequency non-stationarities [10].

If estimation of frequency change is to be taken into account, further lookup tables would be required and so for simplicity, estimates of change in frequency are omitted from both implementations, with the result of frequency change being left to remain within the residual signal, or modelled by multiple atoms within the decomposition, depending on the implementation used.

The non-causal implementation with envelope type discrimination iterates over results returned from a peak analysis process. In this case a single estimation from the DFT is extracted for each peak. If a dictionary of slightly random atoms is compiled, then the traditional approach if MP is applied where each atom is compared with the residual signal. Once the atom with the highest correlation to the signal represented by that peak is selected, that peak is discarded from the list and the MoP decomposition using a number of slightly varying atoms is applied to the next peak. This is in direct odds with the second implementation which is used for modelling transients, using a rectangular window. This results in a signal which is not blurred at frame boundaries due to the windowing process, but at the cost of spectral leakage across multiple bins. This implementation does not reply on peak picking as is the case with the previous implementation.

The segmented MoP implementation used for transient modelling, and eventually used in further chapters for modelling complete compositions incorporating multiple polyphonic components, is implemented in such a manner that a single sinusoidal peak in the Fourier domain can be modelled by multiple components in a truly iterative decomposition from repeated DFT's. This approach uses causal measurements and amplitude change estimates from [14] in a sinusoidal model where the signal is better expressed as 4.11. This approach minimises the amount of energy contained within the final residual signal. Transient components and non-monotonic amplitude and frequency modulations are captured in an overcomplete decomposition. A single sinusoidal peak in the Frequency domain containing nonstationarities; where the main lobe is quite possibly spread across multiple bins, is therefore modelled by multiple sinusoidal components.

Which approach is more accurate for modelling kick and bass in the presence of time and pitch scale modifications remains a future research directive. The non-causal implementation is currently able to distinguish between linear and exponential amplitude change. However, the causal implementation could be expanded in the same manner with the analytical expression for modelling the first order difference of the phase derived using a rectangular window and linear amplitude change in a similar approach as in C.1.3.

The non-causal approach results in less components, as only a single best fitting atom is selected per sinusoidal peak in a similar approach to traditional SMS. This in conjunction with the windowing process results in a residual signal containing transients, non-monotonic amplitude change and non-stationary frequency changes. The qusi-stationary sinusoidal components with monotonic-amplitude change modelled by the decomposition are simple to apply pitch and time scale modification to without any difficulties. However, this approach results in a rich final residual signal with a large amount of energy, and how pitch and time scale modification are applied with these components remaining in the residual requires further investigation.

The second approach of using a rectangular window and an overcomplete representation is discussed in detail in Chapter 6. This approach is capable of modelling highly complex signals from single frame analysis. However, capturing highly non-stationary components as a number of sinusoidal atoms presents challenges regarding time and pitch scale modifications. This approach performs the best for modelling any type of signal from a non-overlapping single-frame analysis system. Having a highly accurate overcomplete model however, is not of interest to a music producer with the intent of changing a kick and bass lines tempo or key if such a representation is incapable of achieving this while maintaining the high quality low end separation of the original input.

Details of the two approaches are presented below. The audio input x is processed in frames of 1024 samples. The output of of the MoP decomposition contains an output signal y, and arrays for each sinusoidal parameter containing the estimates used by each atom for amplitude, frequency, amplitude change, and phase. Within the MoP decomposition the Fourier Transform is applied to the input signal x, and the sinusoidal parameters are extracted from the component with the highest magnitude in the DFT. These parameters of Amplitude (a), Frequency (f), Amplitude change (dA), Frequency change (dF) [optional], Amplitude Curve Type (Linear or Exponential) [optional - can presume exponential as default], and phase (starting or mid depending on analysis method) are then used to create a base atom which is aimed to have a high correlation to that selected sinusoidal component. This atom can be used directly or a dictionary of slightly random atoms generated and compared by use of the inner product of each atom with the residual signal $\mathbf{R^n s}$ at each iteration. The atom with the highest correlation g_{γ_n} is selected and subtracted from the residual resulting in a new residual signal $\mathbf{R^n s}$. As long as the resulting residual signals energy is less than the input signal, meaning that there is at least some correlation and energy has not been added (resulting from incorrect parameter estimates), then the residual signal becomes the new input signal of the next decomposition stage $(n + 1)$. A new DFT

is applied, and the process repeated N times or until some criteria is met such as the residual signals energy is reduced below some threshold such as -60 dB.

4.3.3 Non-Causal Implementation with zero-phase padding using Hanning window

1. Set $R^n s$ = input x
2. Calculate Input Energy inEn
3. Set Residual Energy rEn to inEn
4. Zero-Phase Pad Rn
5. Calculate DFT
6. Find Peaks in Magnitude Spectrum
7. Select Max Peak
8. Calculate first order phase differnece (PD)
9. Lookup Change in Amplitude (dA) from PD from Linear and Exponential tables (in Nepers)
10. Convert dA estimates to dB.
11. Lookup expected magnitude second order difference m2od from tables for exponential and linear amplitude change given estimates of linear and exponential amplitude change dA.
12. Compare measured m2od with expected results for linear and exponential amplitude change.
13. Select amplitude curve with closest value of expected m2od with measured m2od
14. estimate amplitude and frequency from parabolic interpolation
15. Correct phase estimate from lookup table.
16. Synthesise base atom from estimates of amplitude (A), frequency (f), change in amplitude dA in dB, curve type (Exponential or Linear), with corrected phase offset. (Change in frequency is ignored for simplicity and left as a future improvement).
17. Create dictionary of slightly varying atoms from Base atoms parameter estimates.
18. Iterate over each atom within the dictionary and compare the inner product.
19. Set the atom with the highest correlation as the atom for this sinusoidal peak.
20. Check that no energy is added as this method uses the direct amplitude estimating method rather than the inner product method.
21. If the energy of the residual signal $R^n s$ has been reduced, store the atoms parameters, otherwise discard the peak and associated atom and move on to next peak / component with the highest magnitude from the DFT.
22. Update parameter estimates with values from atom with highest correlation.
23. Remove peak from list and move on to new peak
24. Repeat for all peaks.

Depending on the input signal, the remaining residual signal might still contain a lot of energy consisting of transient components, non-monotonic amplitude change, sinusoids with frequency change and noise. Any transients or other non-tationary elements remaining in the residual will un-affected by the windowing process, leaving them intact for the next stage of Residual modelling.

4.3.4 Causal Implementation with Rectangular window presuming exponential amplitude change

1. Set ending criteria for reduction in energy (endThresh) to eg. -60 dB
2. Set $R^n s$ = input x
3. Calculate Input Energy inEn
4. Set Residual Energy rEn to inEn
5. Calculate DFT with zero padding
6. While Energy Reduction enReduction > endThresh (-60 dB)
7. Select Max Peak
8. Calculate first order phase differnece (PD)
9. Lookup Change in Amplitude (dA) from PD Exponential table (in Nepers)
10. Convert dA estimates to dB.
11. Estimate Amplitude (A) and frequency (f) from parabolic interpolation
12. Interpolate phase estimate
13. Correct amplitude estimate (Aest) at start of frame from knowledge of window shape.
14. Synthesise base atom from estimates of amplitude (A), frequency (f), change in amplitude dA in dB (exponential), with interpolated phase. (Change in frequency is ignored for simplicity and left as a future improvement).
15. Create dictionary of slightly varying atoms from Base atoms parameter estimates.
16. Iterate over each atom within the dictionary and compare the inner product.
17. Update parameter estimates with values from atom with highest correlation.
18. Set the atom with the highest correlation as the atom for this iteration.
19. Subtract atom from Residual signal
20. Check energy of $R^n s$ has been reduced otherwise exit and use inner product method rather than amplitude estimate directly.
21. store new signal energy
22. Set Residual to new signal
23. Repeat until energy reduction falls below ending threshold

4.4 Guided Modelled Pursuit

Guided MoP is a possible future adaptation of MoP. The creation of a dictionary of atoms with parameters varying using a normal random distribution, can be seen as a pre-processing stage, similarity to Guided Matching Pursuit (GMP) [12]. GMP adds a "pre-processing step into the main sequence of the MP algorithm. The purpose of this step was to perform an analysis of the signal and extract important features, termed guide maps, that are used to create dynamic mini-dictionaries comprising atoms which are expected to correlate well with the underlying signal structures thus leading to focused and more efficient searches around particular supports of the signal."

The scalar results with the highest correlation to the sinusoid being modelled, returned from the the initial decomposition could be used to derive 'Guided Maps' for further refining the process using non-uniform distributions of random values for each parameter guided by the results of the pre-processing decomposition stage.

This is similar in a way to that of MultiResolution Molecular Matching Pursuit (MRMMP) proposed in [198], where a dictionary of Gabor atoms are used for tracking sinusoidal partials. MRMMP selects a seed atom from initial estimates; the candidate atoms for the next iteration are restricted to a subset of the entire dictionary where frequency is constrained to that of the base atom, and the scale of the atoms within the min-dictionary are scaled both forward and backwards in time.

Initial tests in creating additional dictionaries with slightly random parameter estimates with a skewed distribution targeted in the direction of achieving an improvement in the accuracy of parameter estimates shows some promise, although at a much higher computational cost than using the base atom directly, or the above mentioned iterative decomposition using a normalised distribution. Further work is required for fully exploring the potential of this method. A metric for measuring the amount of accuracy gained in comparison to the computational cost of the extra processing incurred is left as a future research direction.

Dictionaries composed of kick and base atoms with known parameters has also shown promising results from initial tests regarding proof of the concept. Such dictionaries could potentially be combined with other mini dictionaries guided by the results of the initial atomic decomposition.

4.5 MoP System Investigation

The following section provides an analytical evaluation of the presented method of MoP. The decomposition of a simple sinusoid with second order amplitude change is decomposed and presented in detail displaying how the combination of sinusoidal atoms from the modelled pursuit decomposition, which only take monotonic amplitude change into account, combine to shape the non-monotonic amplitude of the signal. The effect of Pitch and Time scale modifications are then presented, highlighting the need for further and future research in this area to achieve modified kick and bass sounds that maintain the quality of the low-end separating.

Synthetic Non-Stationary sinusoidal signals adapted from [142] are then tested using the non overlapping frame implementation of MoP with a fixed frame size of 512 samples and a maximumn number of 64 partials. These results are then compared against reassignment, eaQHM and the exponentially damped sinusoidal model (EDSM) in terms of accuracy, where the signal-to-reconstruction-error ratio (SRER) metric is use, and for speed (MIPS) for a number of different frame and hop sizes combinations.

MoP applied to released EDM tracks is performed in a non overlapping single frame analysis system. It is shown how limiting the maximum number of sinusoidal partials allowed to be used by the model affects the quality of the output from the model and the effect this has on the residual signal. The tests performed limited the maximum number of partials to 128, 256, and 512. The quality of the resulting synthesised signal increases with the number of partials allowed, but a full featured dance track can still be modelled accurately with as little as 128 partials. The results of these tests are presented in Appendices B.

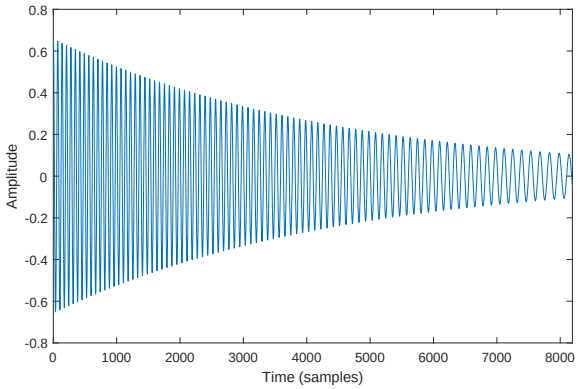
Finally, modelling of transients using modelled pursuit is presented along with a non-segmented implementation of MoP for the analysis and re-synthsis of percussive sounds.

4.5.1 Testing and Results

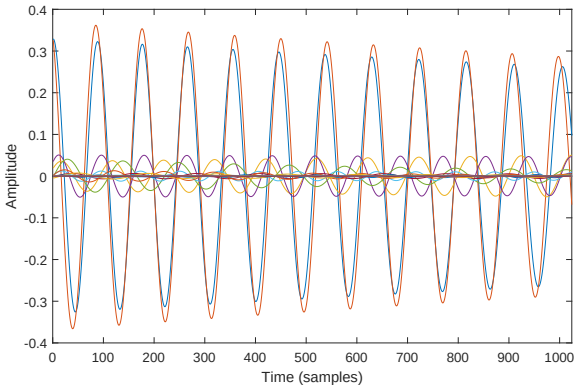
The test results in the following sections provide the atomic decomposition of signals containing monotonic and non–monotonic amplitude change, as well as both amplitude and frequency change. MoP is shown to encapsulate frequency change within the atomic decomposition as multiple sinusodal components with stationary frequency estimates. Only amplitude change is incorporated into the non-stationary sinusoidal parameter estimates.

4.5.1.1 Single Sinusoid: [dF=400 Hz, dA=-16 dB]

Figures 4.18, 4.19 and 4.20 show how a single sinusoid with a linear change in frequency of −400 Hz and an explonential change in amplitude of −16 dB; −50 Hz and −2 dB per frame, can be modelled using MoP. Here the change in frequency is captured using an overcomplete decomposition from atoms which do not include frequency change within their parameter estimates. Figure 4.18a displays the sinusoid sampled at 48 kHz with a starting frequency of 700 Hz, and ending at 300 Hz, generated over 8192 samples. Figure 4.18b shows the individual sinusoidal components from the $4th$ frames atomic decomposition using single frame phase difference measurements for estimating amplitude change.

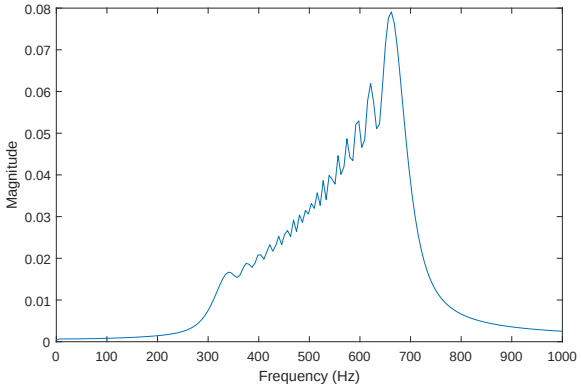


(A) 700 Hz sinusoid with -16 dB exponential amplitude change and -400 Hz frequency change

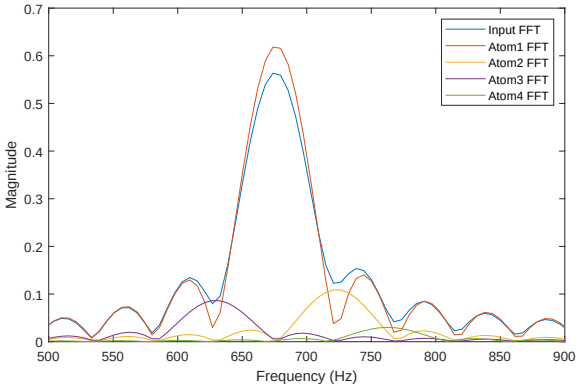


(B) MoP decomposition using a frame size of 1024 sample at Frame=4, samples=[4097:5120]

FIGURE 4.18: (A) Input signal (@48 kHz) and (B) MoP decomposition of $4th$ analysis frame of 48 kHz sampled sinusoid kHz with a starting frequency of 700 Hz, with −2 dB amplitude change and −50 Hz frequency change over each 1024 sample frame



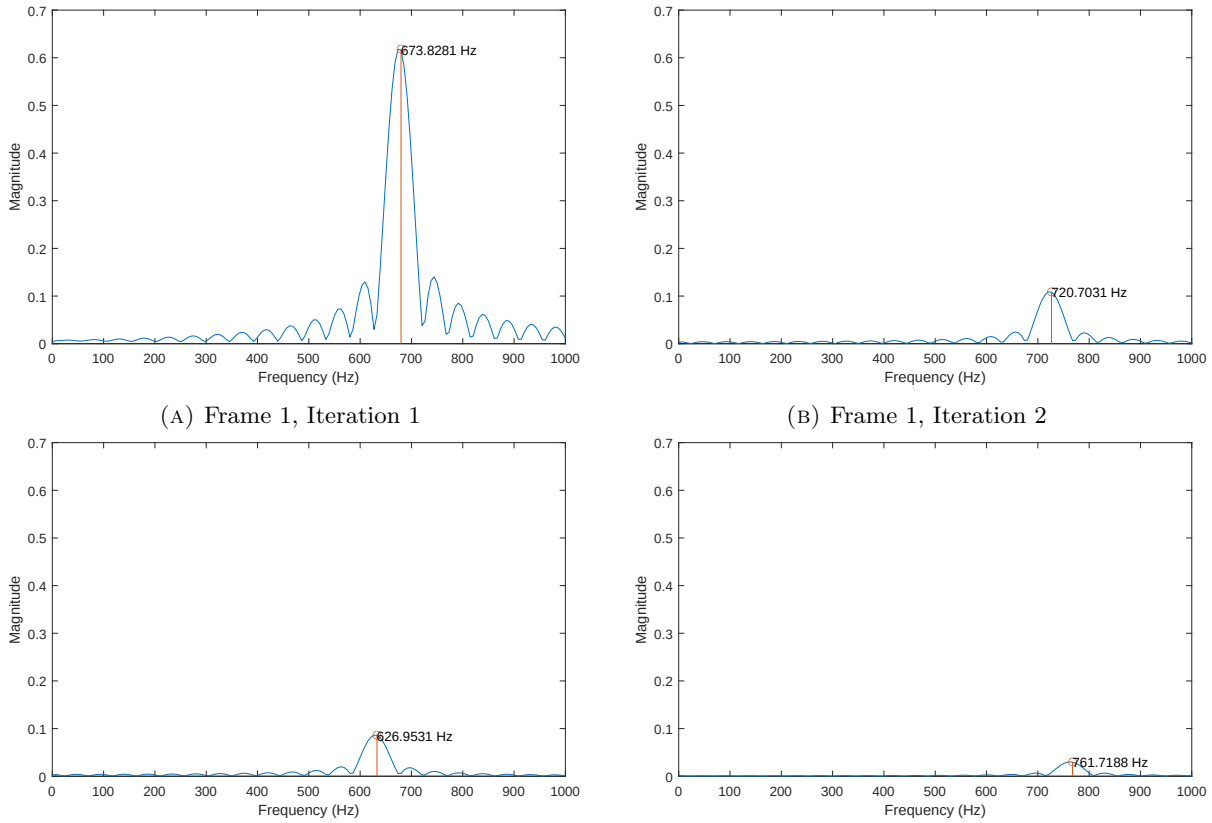
(A) FFT of Sinusoid with -16 dA and -400 Hz dF



(B) FFT of Frame 1 input and atoms

FIGURE 4.19: FFT of input signal (A) and FFT of first frames decomposed atoms

Figure 4.19a shows the frequency spectrum of the entire signal, while Figure 4.19b displays the frequency spectrum of the residual signal from the $1st$ frame at each step of the decomposition. Figures 4.20a to 4.20d show the frequency spectrum of the residual signal at each stage of the decomposition. The reduction in the sinusoidal peak in the Frequency domain clearly shows that the estimates of dA provide a high degree of correlation with the input signal.



(A) Frame 1, Iteration 1

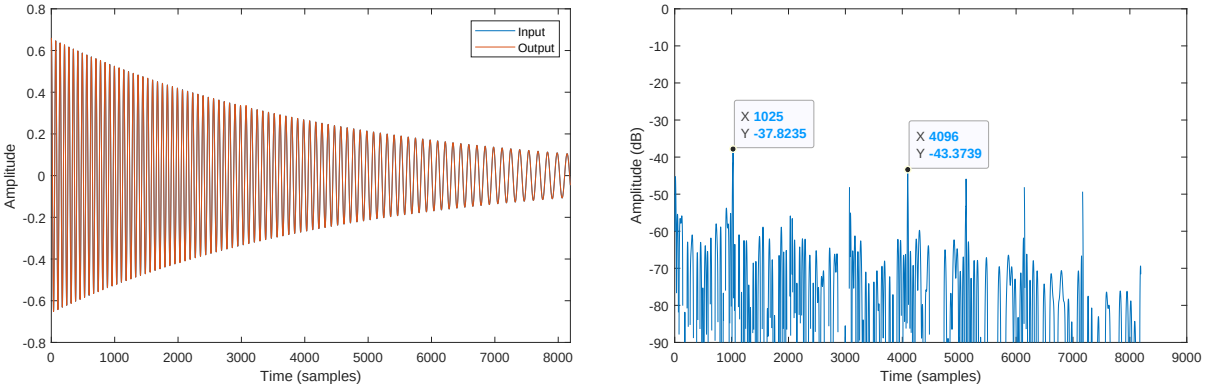
(B) Frame 1, Iteration 2

(C) Frame 1, Iteration 3

(D) Frame 1, Iteration 4

FIGURE 4.20: Iterative MoP decompostion of sinusoid with dA=−2 dB and dF=−50 Hz each frame

Figure 4.21a shows a sinusoid with a large change in frequency of -4 kHz (-625 Hz per frame), ad the resulting synthesised output from he MoP decomposition where frequency change is presumed stationary. Figure 4.21b shows the final residual signal in dB with a mean error of -60 dB. The peak error values of around -40 dB result from not taking phase coherence between frames into account which can cause some discontinuities at frame boundaries.



(A) Sinusoid with -16 dA and -400 Hz dF (B) Residual of sinusoid with mean error of -60 dB

FIGURE 4.21: (A) Input/Output sinusoid (@48 kHz), (B) residual signal (dB)

4.5.1.2 Single Sinusoid: [dF=-4427 Hz, dA=-16 dB]

The previous test case demonstrated how a sinusoid with a change in amplitude and a change in frequency of 50Hz per frame can be accurately modelled using MoP and an overcomplete decomposition, without taking change of frequency into account. The second test case examines a very large change in frequency of 625 Hz per frame. Figure 4.22 shows the frequency spectrum of the entire signal, exhibiting a large spread in spectral energy across the frequency range of 2500 to 7000 Hz. Figure 4.23 shows the synthesised output signal in comparison to the input.

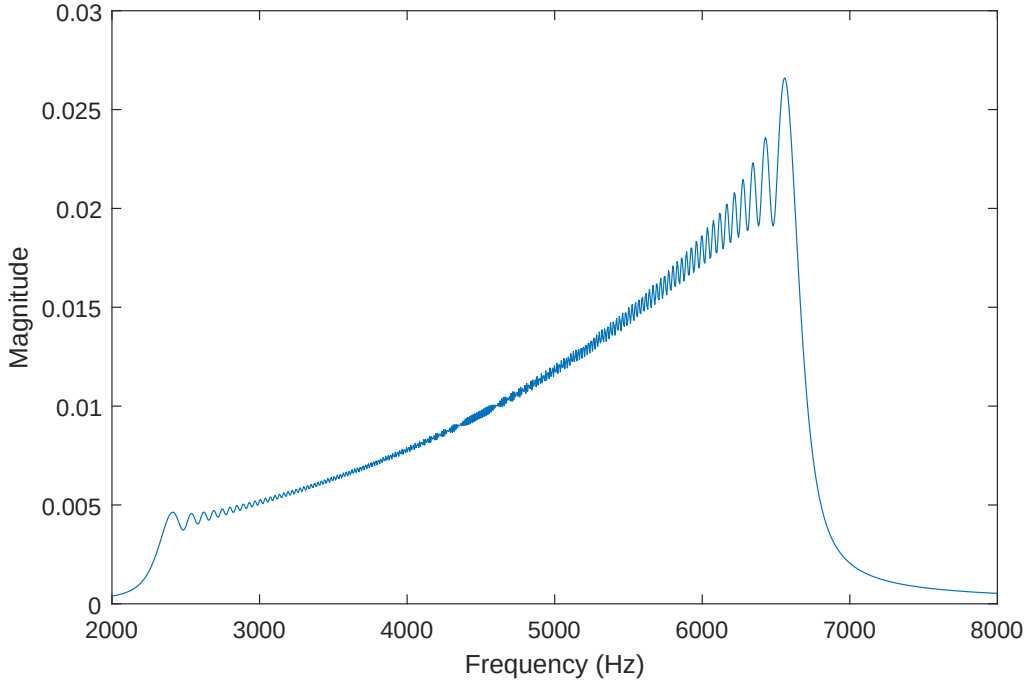


FIGURE 4.22: FFT of sinusoid (@48 kHz), dF=-5000 Hz, dA=-16 dB

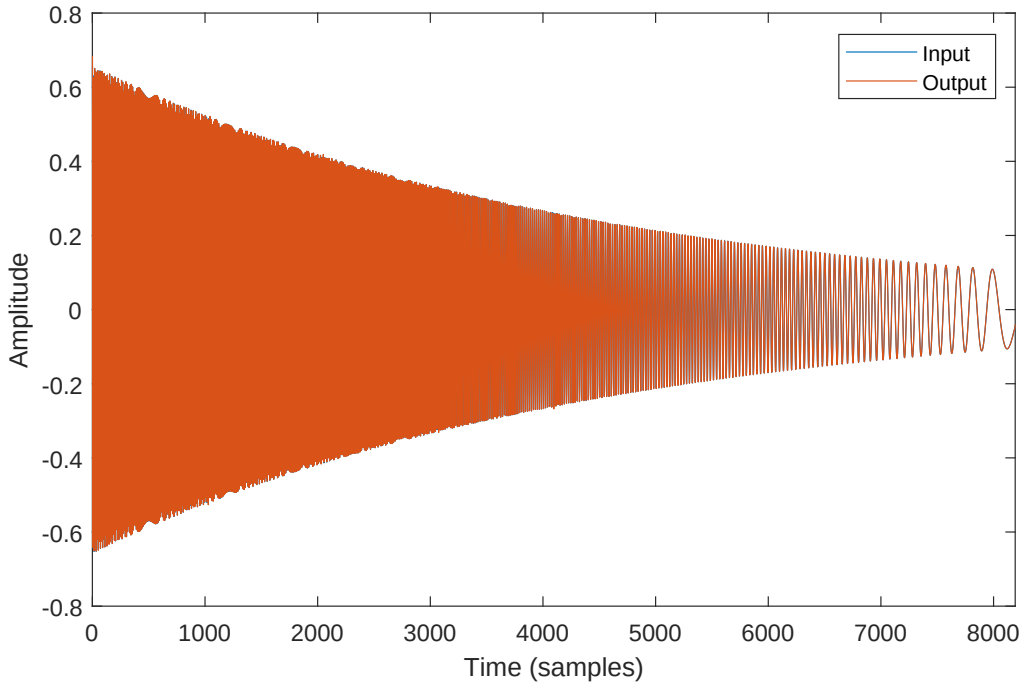


FIGURE 4.23: Input and Output from Multiple Frame MoP Decomposition of sinusoid (@48 kHz)

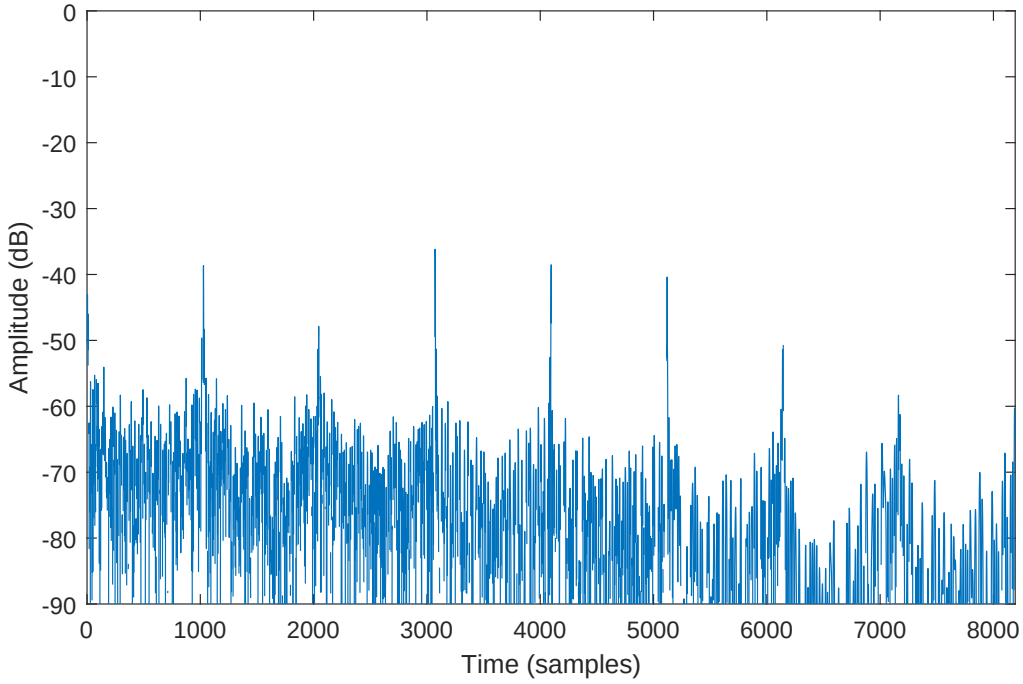
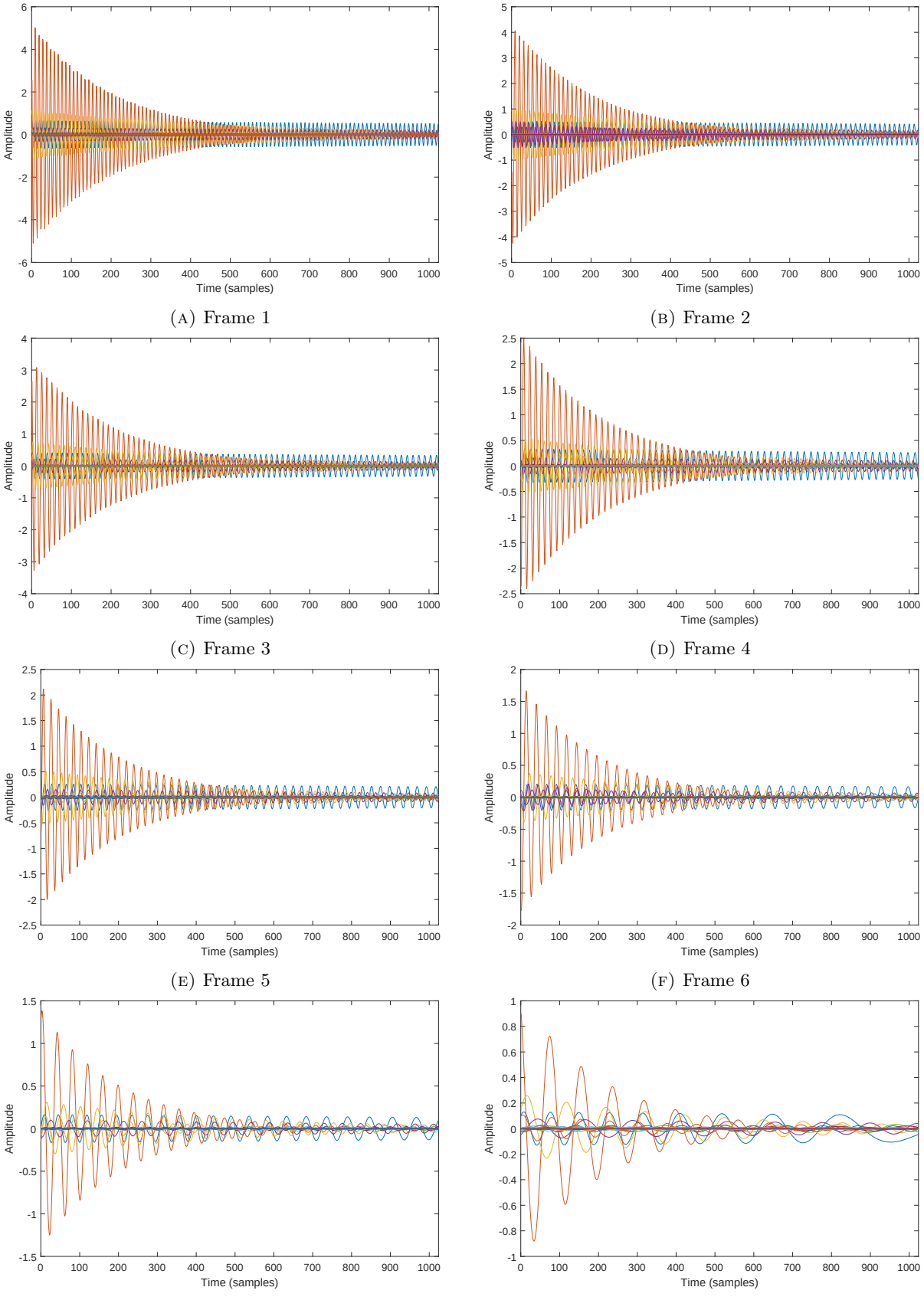


FIGURE 4.24: Input and Output from Multiple Frame MoP Decomposition

Figure 4.23 shows the residual signal with the peaks at frame boundaries around -40 dB, but with an overall mean of -80 dB. Figure 4.25 shows a number of the sinusoidal components decomposed using MoP for each frame.



(A) Frame 1

(B) Frame 2

(C) Frame 3

(D) Frame 4

(E) Frame 5

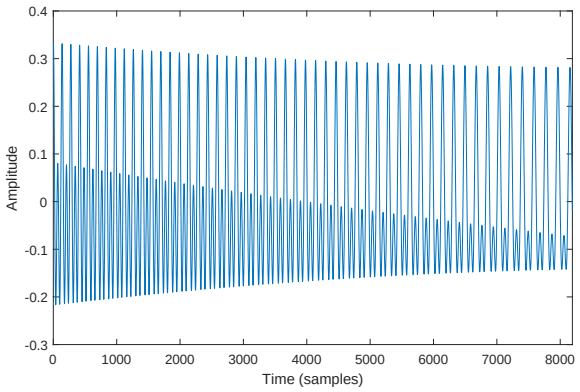
(F) Frame 6

(G) Frame 7

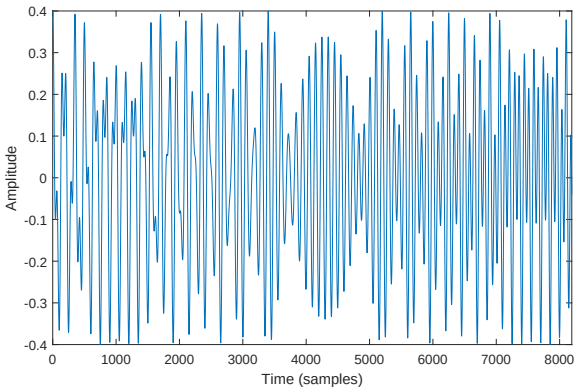
(H) Frame 8

FIGURE 4.25: Frame-by-frame decomposition of sinusoid from 4.23 with large frequency change into multiple semi-stationary components

4.5.2 Multiple Components with MoP

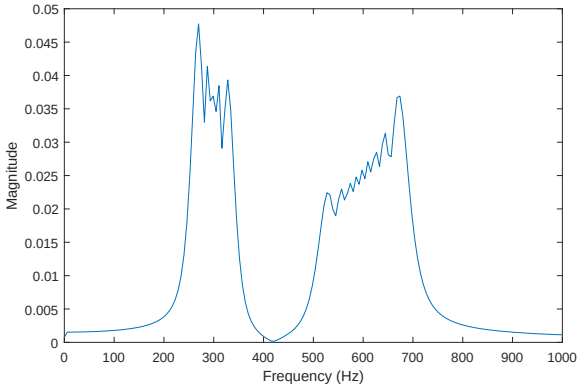


(A) 2 sinusods (@48 kHz), (a) dA=-6 dB, dF=-200 Hz; (b) dA=3 dB, dF=-100 Hz

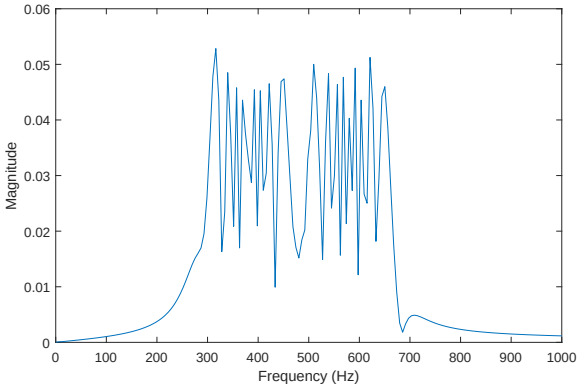


(B) 2 sinusods (@48 kHz), (a) dF=-400 Hz; (b) dF=400 Hz

FIGURE 4.26: Test signals composed of 2 sinusoidal components. (A) 2 sinusoids decreasing in amplitude and frequency. (B) 2 sinusoids with no amplitude change, increasing and decreasing in frequency, with overlapping frequencies overlapping around the middle of the 8192 sample audio frame.



(A) FFT sine(a): dA=-6 dB, dF=-200 Hz; sine(b): dA=3 dB, dF=-100 Hz

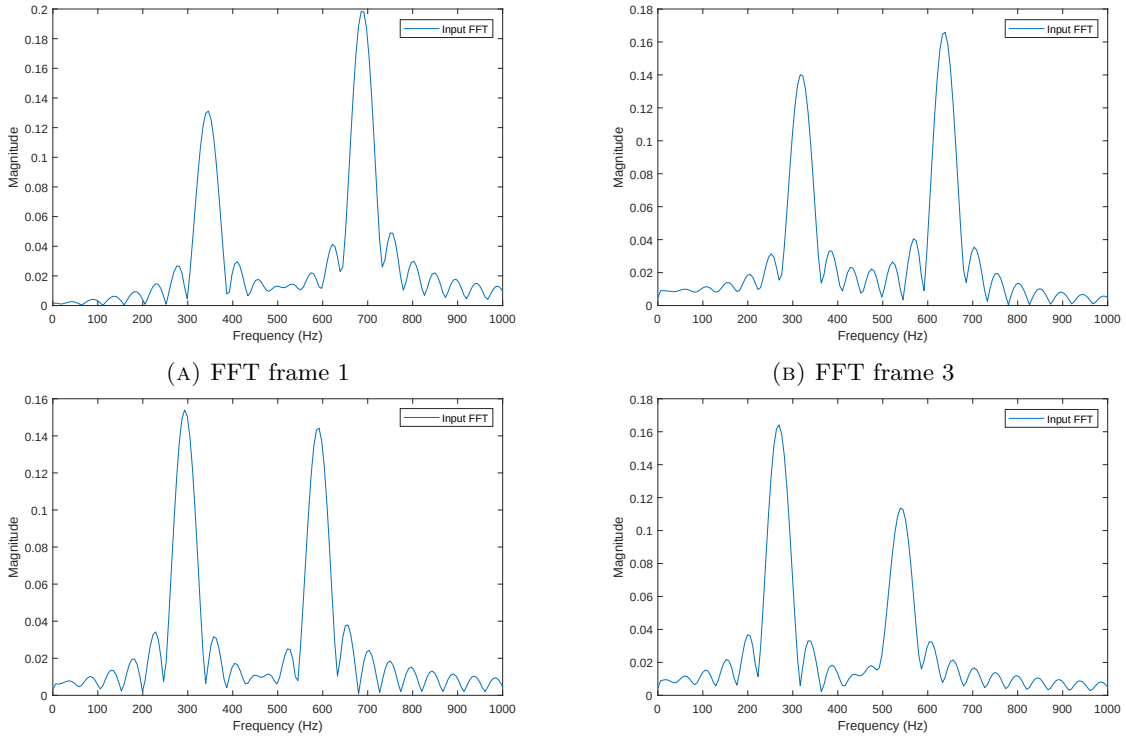


(B) FFT sine (a) dF=-400 Hz; sine(b): dF=400 Hz

FIGURE 4.27: FFT of test signals composed of 2 sinusoidal components. (A) 2 sinusoids decreasing in amplitude and frequency. (B) 2 sinusoids with no amplitude change, increasing and decreasing in frequency, with overlapping frequencies overlapping around the middle of the 8192 sample audio frame.

Figure 4.26 shows two test signals both containing two sinusoidal components. The first signal contains a sinusoid with starting at 350 Hz, and ending at 250 hz (dF = -100 Hz, mid F = 300 hZ, dA = 3 dB). The second component starts at 700 Hz and ends at 500 Hz (dF = -200 Hz, mid F = 600 hZ, dA = -6 dB). The second singal contains two sinusodal compoennts which overlap in the frequency domain. The first component has a start frequency of 690 Hz and an end frequency of 290 Hz (dF = -400 Hz), while the second component has a start frequency of 270 Hz and an end frequency of 670 Hz (dF = 400 Hz). Figure 4.27 shows the magnitude spectrum of both of these signals.

Figure 4.30 shows the DFT magnitude spectrum from the MoP decomposition, displaying multiple atoms from each of the two test signals which are composed of 2 non-stationary sinusoidal components. (A) 2 sinusoids decreasing in amplitude and frequency. (B) 2 sinusoids with no amplitude change, increasing and decreasing in frequency, with overlapping frequencies.



(A) FFT frame 1

(B) FFT frame 3

(C) FFT frame 5

(D) FFT frame 7

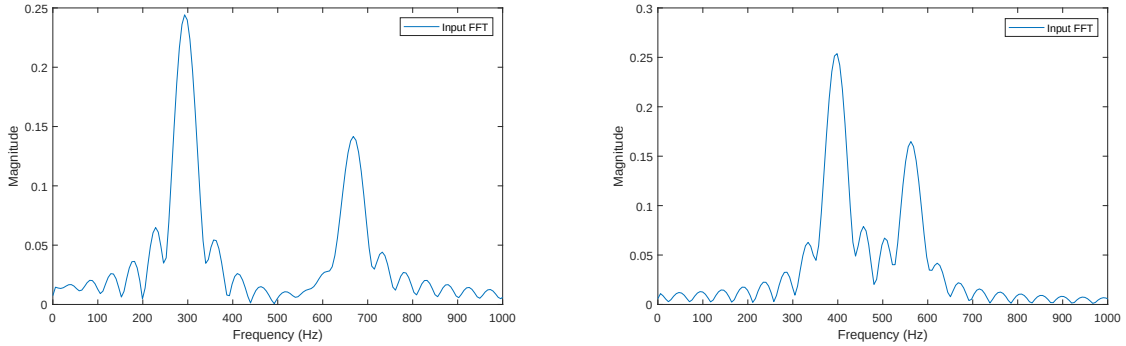
FIGURE 4.28: DFT frames (1024 samples) of test signals composed of 2 sinusoidal components from 4.26a

Figures 4.28 and 4.29 show the DFT magnitude spec tum and how this changes over a number of analysis frames for each of the two test signals from Figures 4.26a and 4.26b respectively.

Figure 4.30 show the reconstructed output signal from the MoP decomposition of the first signal (A). The residual of the signal is shown in Figure 4.31.

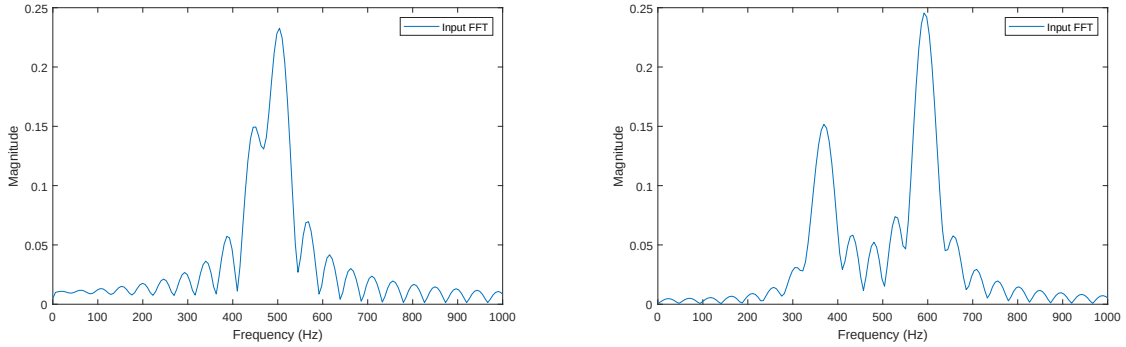
Figure 4.31 show the reconstructed output signal from the MoP decomposition of the second signal (B). The residual of the signal is shown in Figure 4.32.

The results clearly show that MoP is able to accurately decompose and accurately re-synthesise a signal with multiple components, including components which overlap in frequency, in an over-complete atomic representation.



(A) FFT frame 1

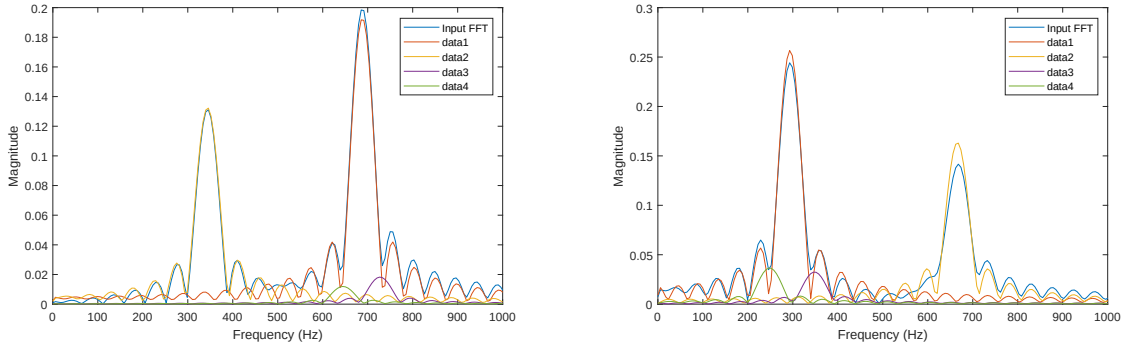
(B) FFT frame 3



(C) FFT frame 5

(D) FFT frame 7

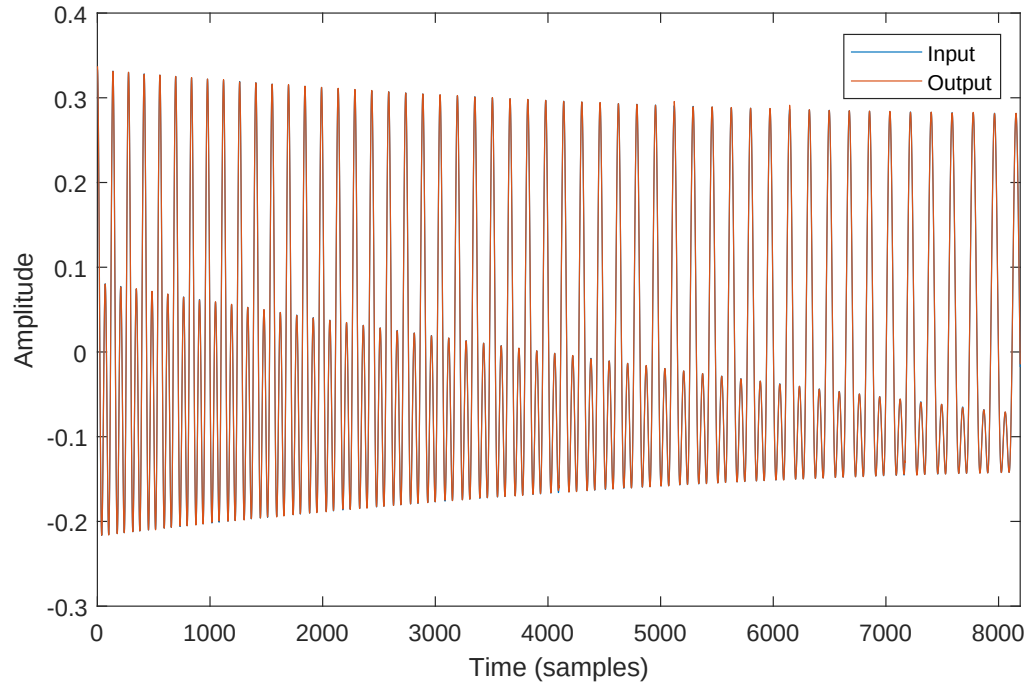
FIGURE 4.29: DFT frames (1024 samples) of test signals composed of 2 sinusoidal components from 4.26b



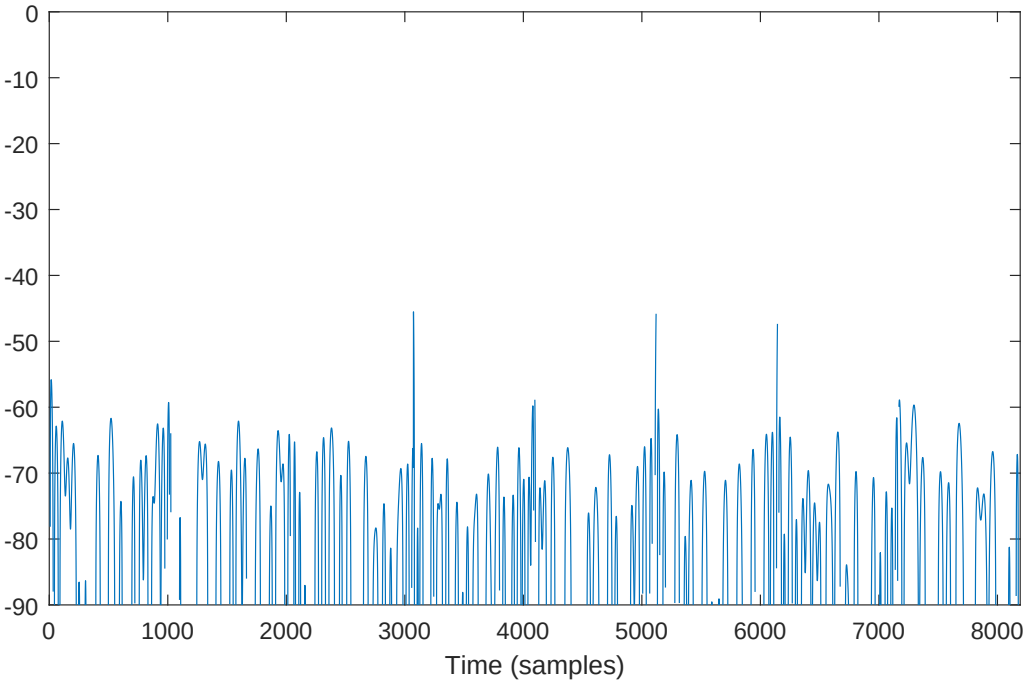
(A) FFT frame 1

(B) FFT frame 1

FIGURE 4.30: FFTs from MoP decomposition displaying multiple atoms of a test signals composed of 2 sinusoidal components. (A) 2 sinusoids decreasing in amplitude and frequency. (B) 2 sinusoids with no amplitude change, increasing and decreasing in frequency, with overlapping frequencies overlapping around the middle of the 8192 sample audio frame.

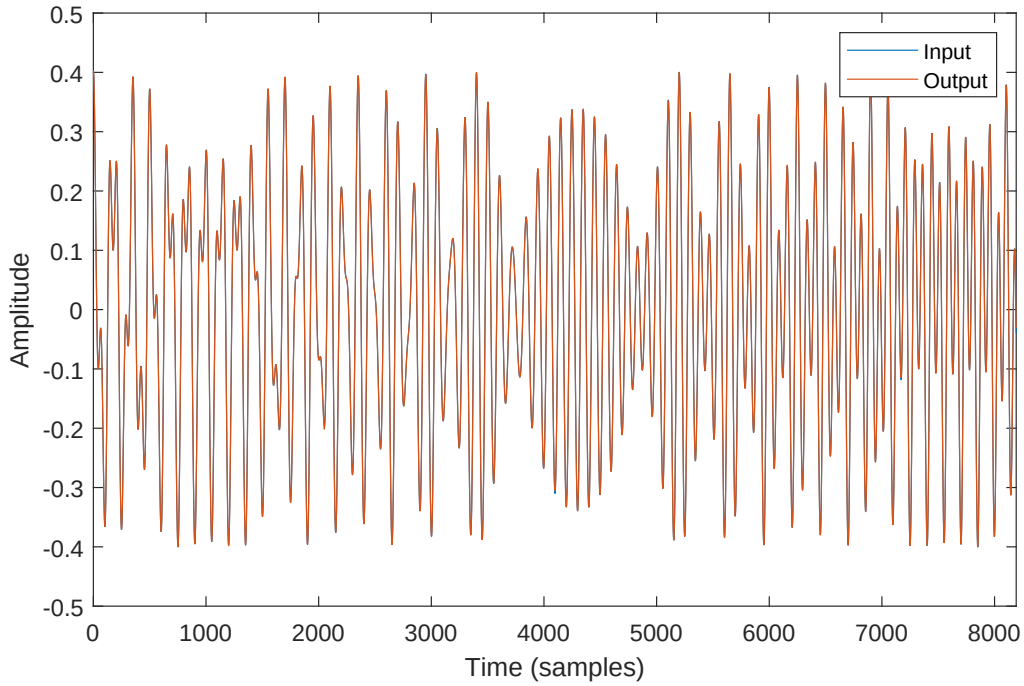


(A) 2 sinusods (@48 kHz), (a) dA=-6 dB, dF=-200 Hz; (b) dA=3 dB, dF=-100 Hz

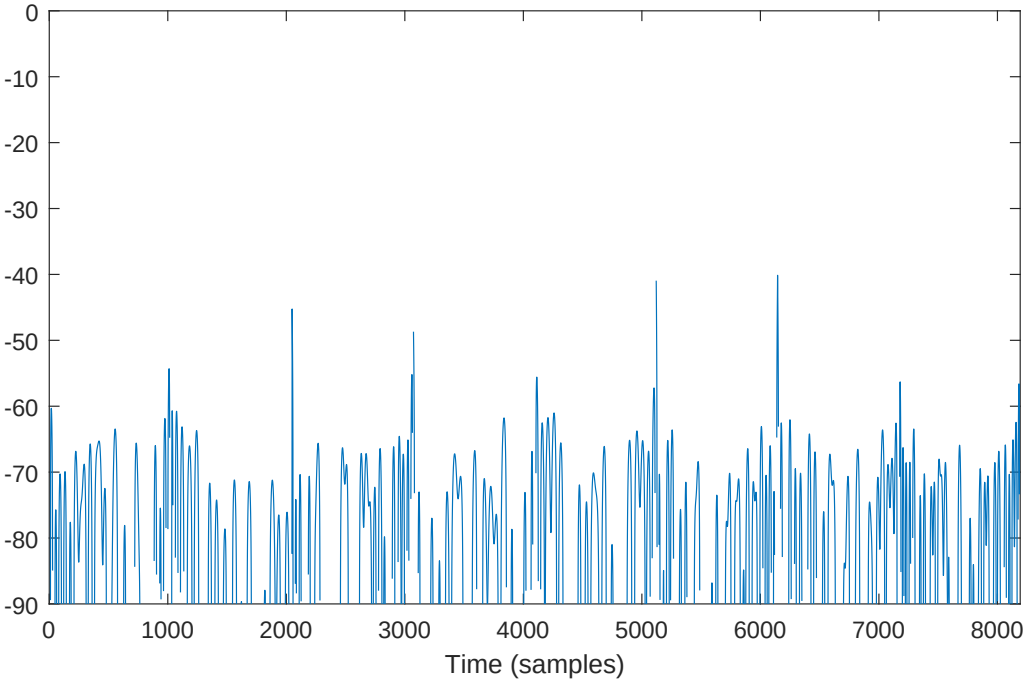


(B) sine (a) dF=-400 Hz; sine(b): dF=400 Hz

FIGURE 4.31: Test signals composed of 2 sinusoidal components. (A) 2 sinusoids decreasing in amplitude and frequency. (B) 2 sinusoids with no amplitude change, increasing and decreasing in frequency, with overlapping frequencies overlapping around the middle of the 8192 sample audio frame.



(A) 2 sinusods (@48 kHz), (a) dF=-400 Hz; (b) dF=400 Hz



(B) sine (a) dF=-400 Hz; sine(b): dF=400 Hz

FIGURE 4.32: Test signals composed of 2 sinusoidal components. (A) 2 sinusoids decreasing in amplitude and frequency. (B) 2 sinusoids with no amplitude change, increasing and decreasing in frequency, with overlapping frequencies overlapping around the middle of the 8192 sample audio frame.

4.5.3 Modelling non-monotonic amplitude change using MoP

The causal analytical equation for calculating the exponential phase difference measurements from a given range of amplitude change a, in Nepers, using a rectangular window is given by:

$$\left.\frac{d(\arg(W))}{df}\right|_{f=0} = 2\pi\left(\frac{1}{1-e^{a}} + \frac{1}{a} - 1\right) \tag{4.14}$$

The derivation of 4.14 can be found in Section C.1.3. The results of this equation for a range of amplitude in dB is shown in Figure 4.33.

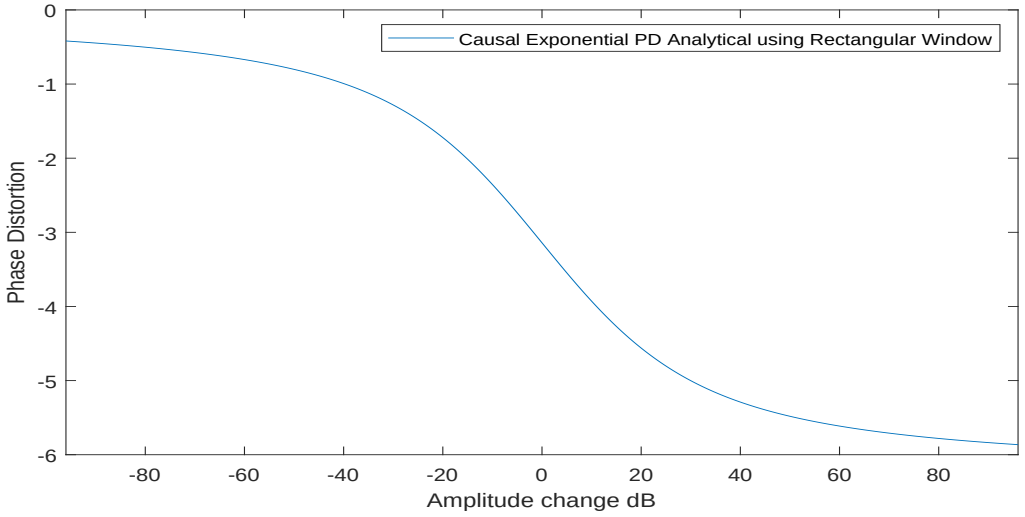
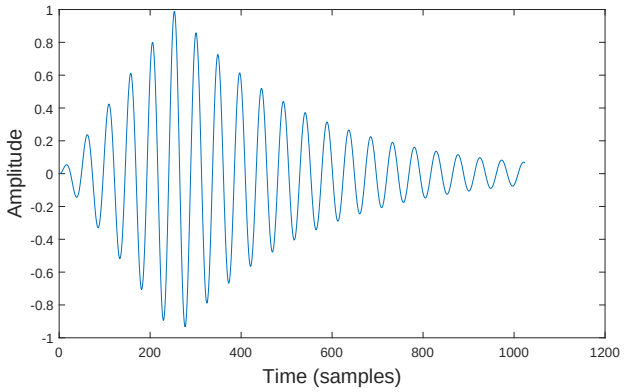


FIGURE 4.33: Results from 4.14 calculating the first order phase difference from causal measurements using a rectangular window

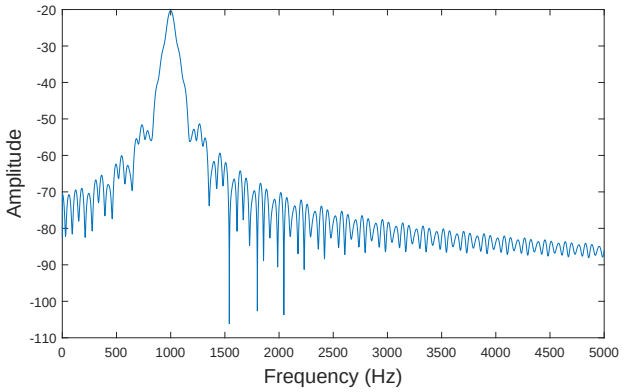
Causal in this case means that FFT measurements are taken from the beginning of the frame, rather than the middle, as zero-phase padding is not employed. A frame size of 1024 is used with a rectangular window. A zero-padding factor of 8 is also used, resulting in an FFT size of 8192, with a bin resolution of 5.8594 Hz when using a sample rate of 48 kHz.

A rectangular window has been selected for investigating the modelling of transients. As mentioned previously, windowing can smear information in the time domain, such as transients, and an overlap add mechanism is required for re-synthesis. The objective of this method is to use single frame analysis with non-overlapping frames to investigate modelling of transients and complex audio signals using MoP.

Figures 4.34b and 4.35b show the spectral leakage incurred from the results of the DFT due to the use of the rectangular window on the input signal, shown in Figures 4.34a and 4.35a.

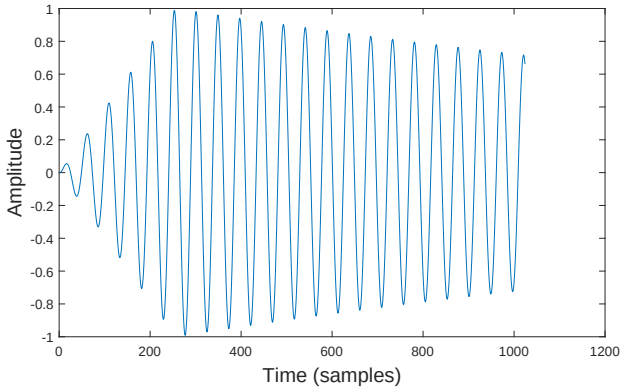


(A) Rectangular windowed 1 kHz sinusoid (@48 kHz) with non-monotonic AM (fast decay)

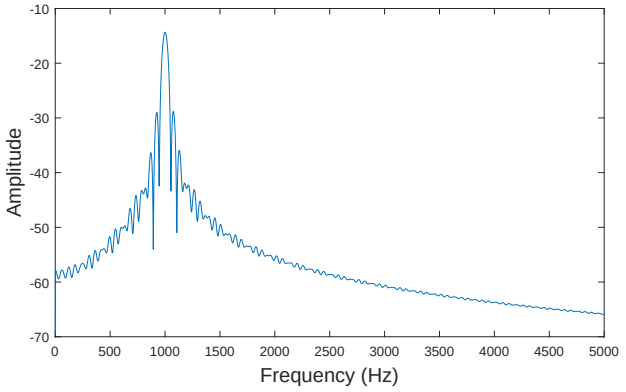


(B) Magnitude spectrum of non windowed FFT showing spectral leakage

FIGURE 4.34: Sinusoid with an attack and release modelled with MoP using a rectangular window



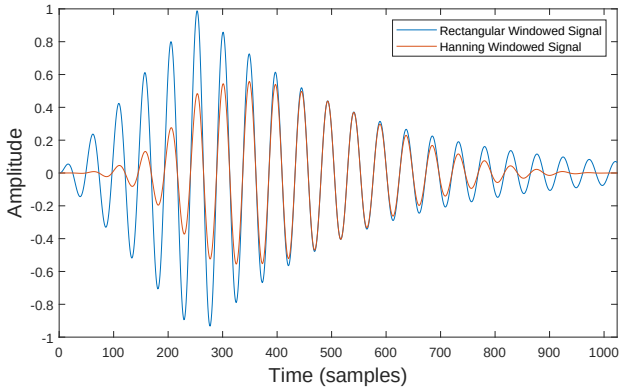
(A) windowed 1 kHz sinusoid (@48 kHz) with non-monotonic AM (slow decay)



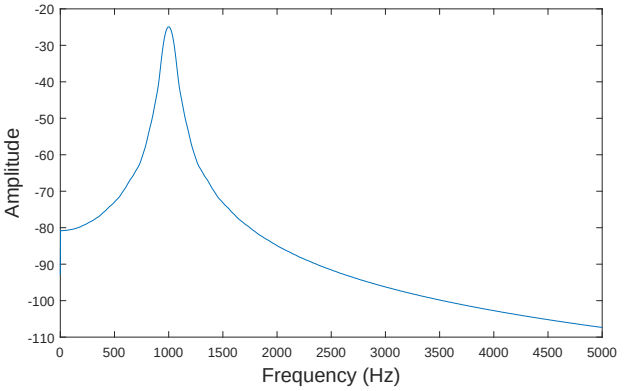
(B) Magnitude spectrum of non windowed FFT showing spectral leakage

FIGURE 4.35: Sinusoid with an attack and release modelled with MoP using a rectangular window

Figures 4.36 and 4.37 show the input signal compared against the Hanning windowed signal and the effect this has on the attack and release regions in time. The attack is clearly delayed in time from the windowing operation. The resulting magnitude spectrum's are also displayed showing the removal of the spectral leakage effects due to the windowing. Windowing therefore requires an overlapping procedure to accommodate these affects, averaging the results using an overlap add procedure. A non-overlapping frame framework using rectangular winding was therefore selected as the mechanism for modelling transients and complex signals in an overcomplete representation using MoP. Figure 4.38 shows the results of using this framework to model a sinusoid with a second order amplitude modulation using 16 and 8 sinusoidal components.

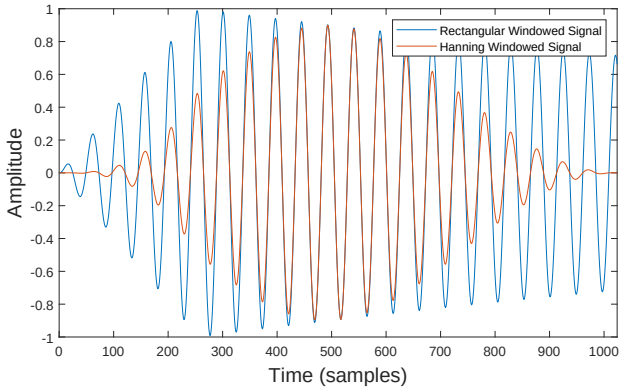


(A) Rectangular compared to Hanning windowed Non-Monotonic Amplitude Sinusoid with fast decay

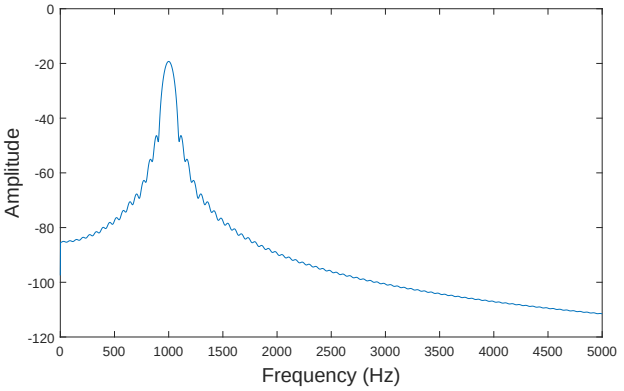


(B) Magnitude spectrum of Hanning windowed FFT showing reduction in spectral leakage

FIGURE 4.36: Comparison of input (@48 kHz), windowed input and the effect the windowing has on the Magnitude spectrum as well as the temporal information.

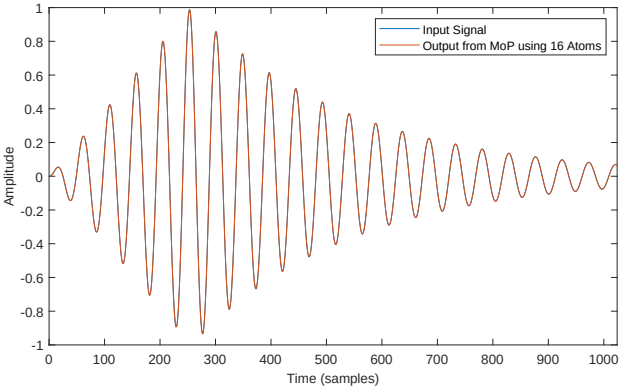


(A) Rectangular compared to Hanning windowed Non-Monotonic Amplitude Sinusoid with slow decay

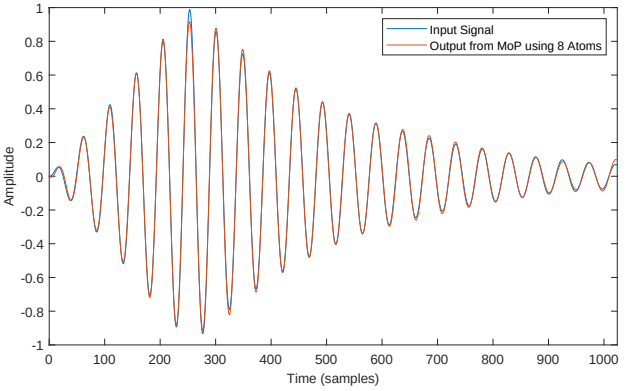


(B) Magnitude spectrum of Hann windowed FFT showing reduction in spectral leakage

FIGURE 4.37: Comparison of input signal (@48 kHz), windowed input and the effect the windowing has on the Magnitude spectrum as well as the temporal information.



(A) Input signal and resynthesised signal using 16 atoms



(B) Input signal and resynthesised signal using 8 atoms

FIGURE 4.38: Comparison of modelling using 16 or 8 atoms (@48 kHz)

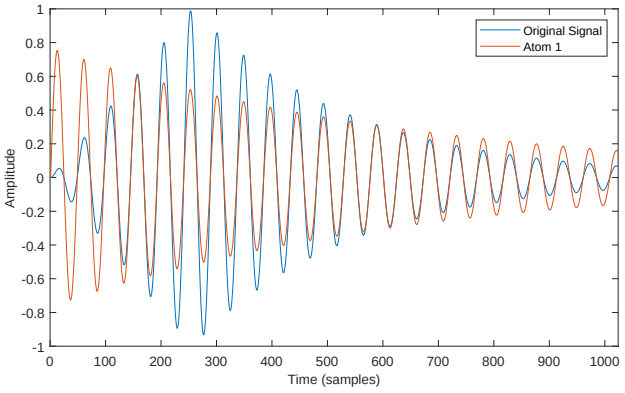
4.5.4 Examining atomic decomposition of non-monotonic AM

Figure 4.38a shows an improvement of using 16 sinusoidal partials compared to only using the first 8 atoms which are returned from the decomposition, as shown in Figure 4.38b. The following section displays the atomic decomposition of the signal shown in more detail.

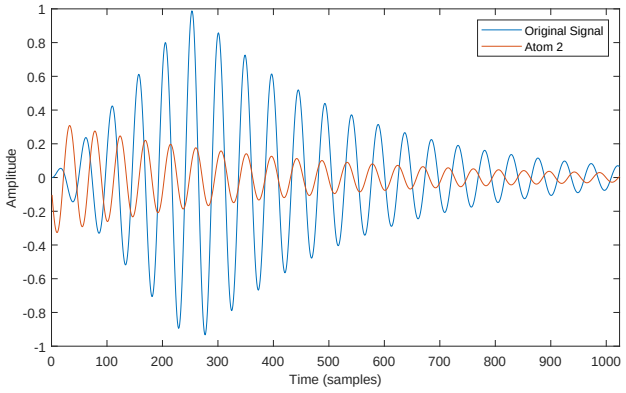
The following list of Figures 4.39, 4.40 and 4.41 breaks down the atomic decomposition of this signal into 8 sinusoidal components. Improvements in modelling the attack and release envelopes are achieved by the atomic decomposition including additional sinusoids, which are either in or out of phase with one another, and combine in such a way that the sinusoidal peaks and troughs either amplify or reduce parts of signals amplitude through constructive and destructive interference. An almost accurate approximation of the signal is achieved with as little as 8 atoms. The final 4 atoms are very low in amplitude, but when combined with the other atoms in the model result in an amplitude modulated sinusoid which is very close to the original input signal.

Some additional figures of the atomic decomposition along with overlayed sinusoidal components are displayed which does pose the question of how meaningful such a model of transients is, and how would this combination of in-phase and out of phase relationships need to be handled and possibly maintained when performing transformations such as pitch shifting, or time stretching with transient preservation, and for maintaining the low-end separation of kick and bass.

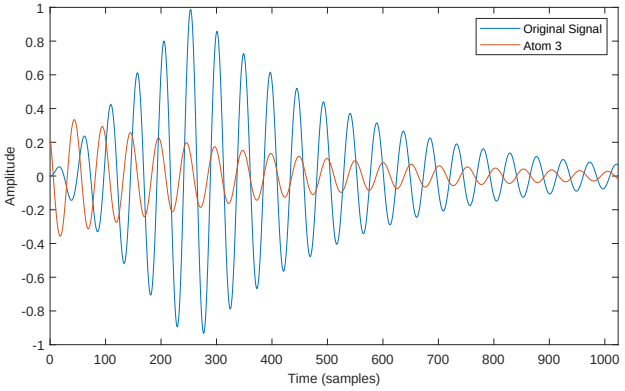
Figure 4.39a shows the first monotonic sinusoidal atom extracted from the MoP decomposition, displayed in red. The sinusoid is in phase with the original signal and it displays a correct estimate for the frequency and phase offset of the signal. The amplitude of the release section of the ADSR at the end of the frame is slightly out, while the signal at the beginning of the frame doesn't take the attack section into account since the method models monotonic amplitude modulations. Figures 4.39b and 4.39c displays the second and third atoms respectively, synthesised from the decomposition of the resulting residual after subtracting the initial atom from the original signal. This displays a signal which is out of phase with the original signal. The destructive interference of the phase combines in such a way as to shape the amplitude of the signal being modelled, resulting in an improvement of the amplitude change and resulting signal with each iteration. Figure 4.39d shows these first 3 atoms overlaid with one another. Figure 4.39e displays the resulting waveform of the combination of the first 2 atoms against the original signal, while Figure 4.39f presents how the addition of the 3rd atoms combines to further improve the signal model.



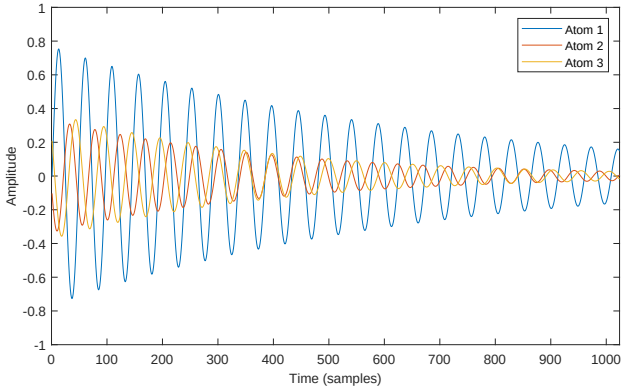
(A) 1st atom extracted against input signal



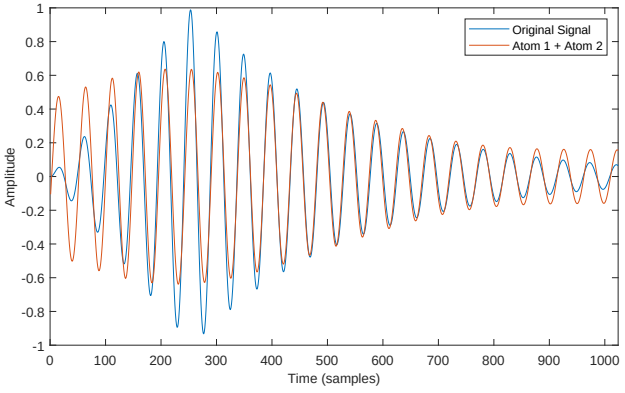
(B) 2nd atom extracted against input signal



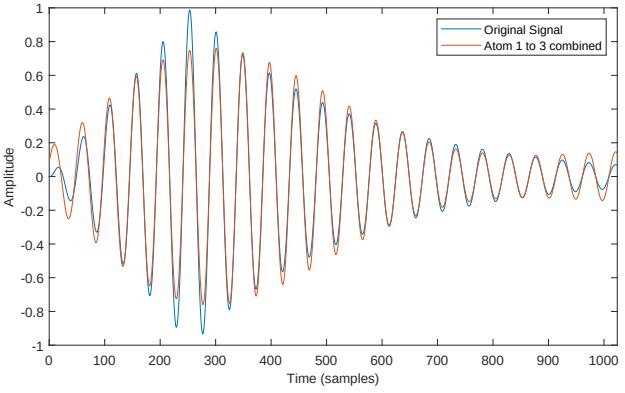
(C) 3rd atom extracted against input signal



(D) First 3 Atoms Extracted displaying phase relationships



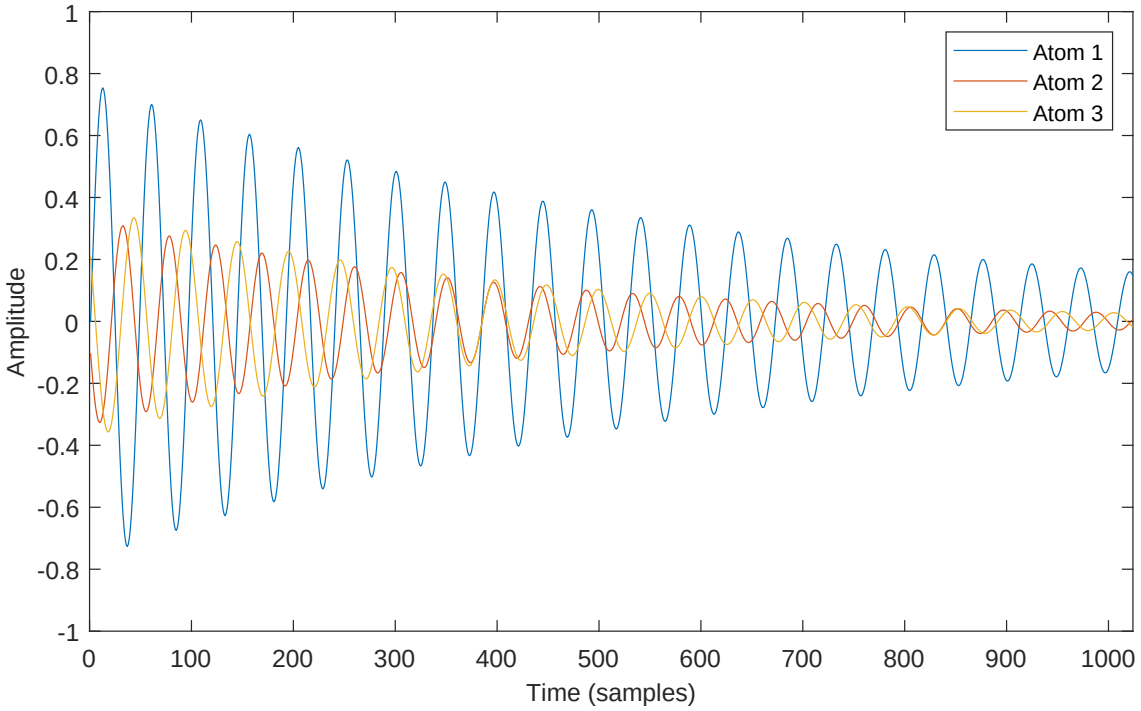
(E) First 2 Atoms combined against input signal



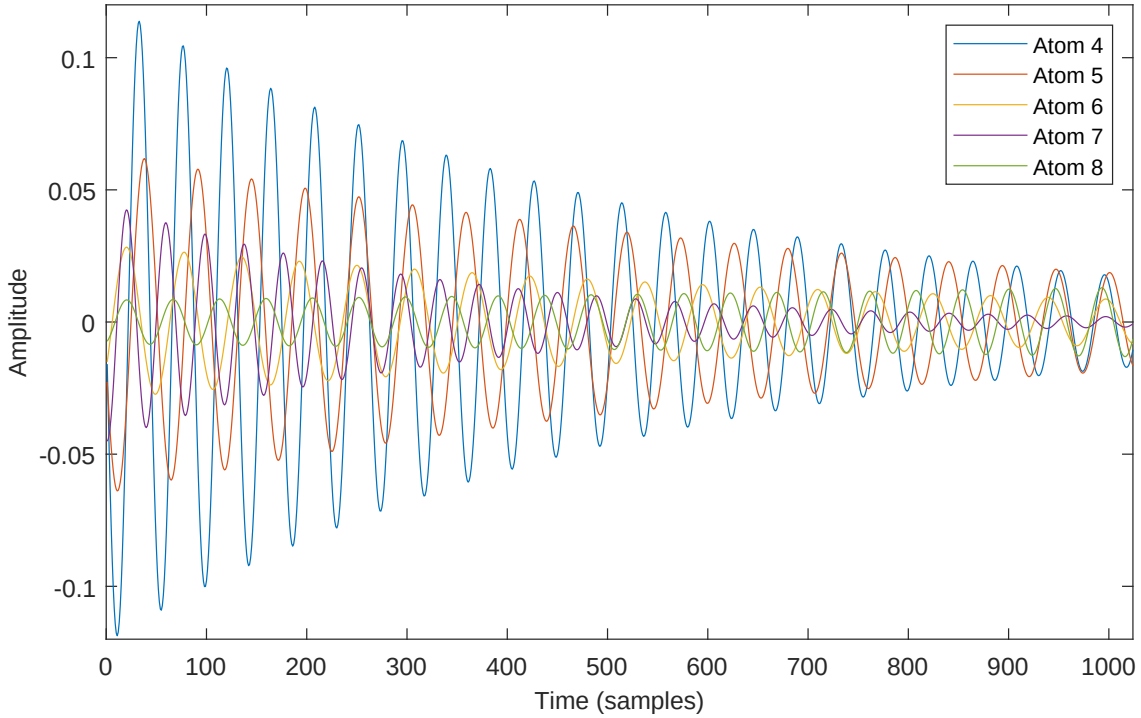
(F) First 3 Atoms combined against input signal

FIGURE 4.39: Comparison first 3 atoms extracted from input signal (@48 kHz)

Figure 4.40a shows a larger plot containing the first 3 atoms which have the greatest effect on the modelled signal, compared to the remaining 5 atoms of the decomposition, which is displayed in Figure 4.40b. These final 5 atoms are low in amplitude and each successive atom has a lesser effect on the results of the model. the effect these atoms have on the resulting signal can clearly be seen by examining the difference in the models output between Figures 4.39f and 4.41d.



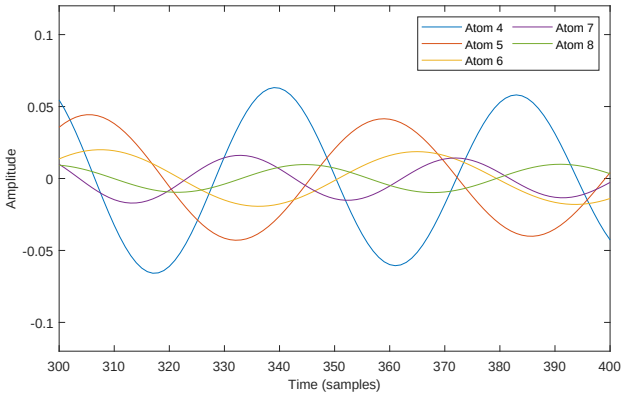
(A) First 3 Atoms extracted



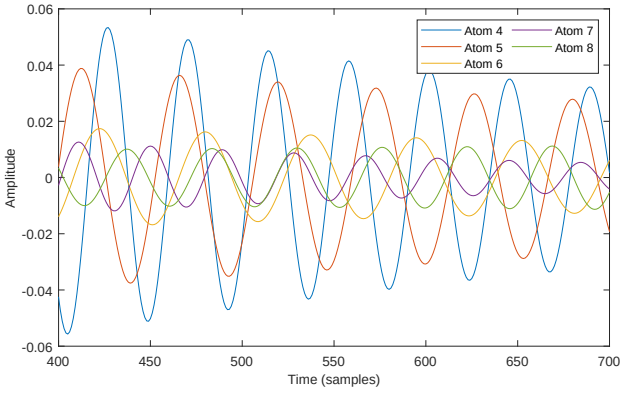
(B) Atoms 4 to 8

FIGURE 4.40: Comparison of first 3 atoms (a) which have high amplitude, compared to remaining 5 atoms (b) with much lower amplitudes

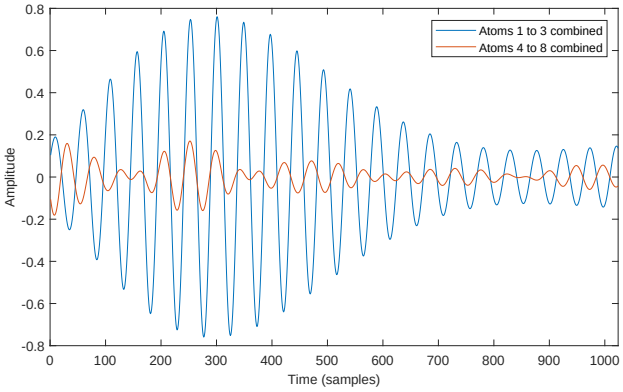
Figures 4.41a and 4.41b provide a closer inspection of the final 5 atoms from the decomposition, as this is less clear when visualising the entire signal as in Figure 4.40b.



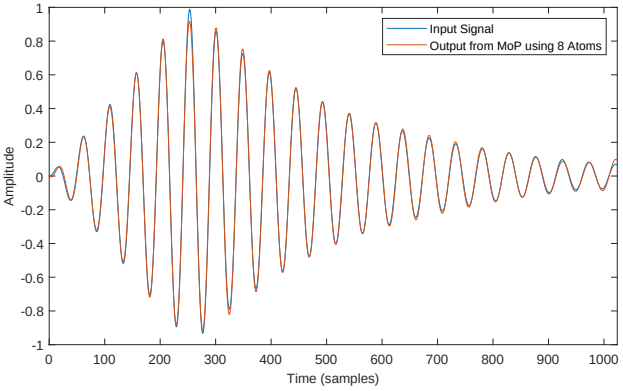
(A) Atoms 4 to 8 zoomed in [300:400] samples



(B) Atoms 4 to 8 zoomed in [400:700] samples



(C) Signal from atoms 1 to 3 (Blue) compared to signal from atoms 4 to 8 (Red)



(D) Input signal correctly modelled after combining remaining atoms with first 3

FIGURE 4.41: Examination of atoms 5 to 8 as well as their combined effect on the resulting signal when combined with the first 3 atoms

Figure 4.41c displays the output signal from the combination of the first 3 atoms in blue. The output signal from the combination of the last 5 atoms is displayed in red for comparison. The combination of these two signals, which equates to the combination of all 8 atoms is shown in Figure 4.41d

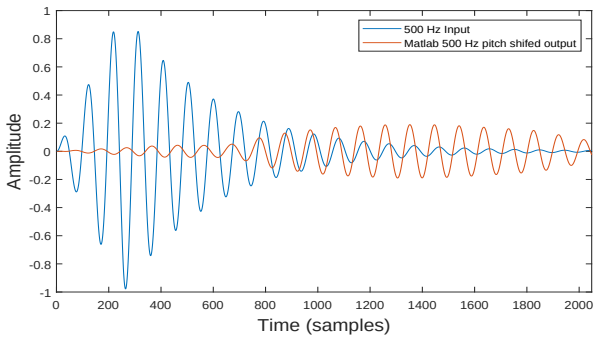
4.5.5 Examining the effect of Pitch Shifting

Pitch shifting atoms with monotonic amplitude and frequency change is relatively simple as these can be represented by a single sinusoid. Section 4.5.4 presented a detailed examination of the atomic decomposition of a non-stationary sinusoid with non-monotonic amplitude change. Transients and higher order amplitude modulations are modelled by MoP as multiple sinusoidal components where the constructive and destructive interference of the atoms combine in such a way as to restore the temporal envelope of the original signal.

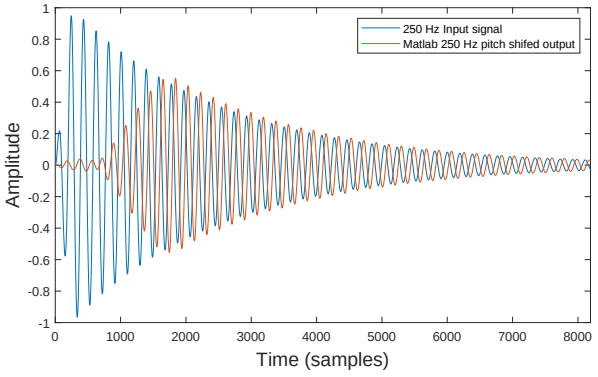
Given that the sinusoidal model used in this thesis is able to re-synthesise each sinusoidal component on a sample by sample basis, pitch shifting of monotonic AM/FM components is straightforward, requiring the values of the frequency estimates to be modified before re-synthesis; importantly, the phase needs to be coherent across frame boundaries. When performing a time or pitch based modification, the estimated phase from the next frame will no longer be valid, as it is no longer in sync with the modified signals phase. This requires storing the phase value at the end of a synthesis frame for use at the beginning of the next frame. Relying on phase estimates returned by the MoP decomposition does not maintain phase coherence across frame boundaries and introduces a discontinuity in amplitude caused by a phase shift, which results in a signal no longer in phase with the expected output.

In general pitch shifting does not retain the temporal envelope of a signal, and can be considered a time based effect which can shorten or lengthen the duration of the audio signal. "Psychephysical experiments show that the pitch of a short sine wave tone depends upon the amplitude envelope of the tone." [199]

The effect of pitch shifting by an octave is presented below using Matlab's implementation, which is based on modifying the time-scale of the signal using a phase vocoder and then resamples the modified signal based on the methods presented in [94, 200]. Figures 4.42a and 4.42b clearly show that the resulting pitch shifted signal's amplitude envelope has been affected and the resulting signal is no longer aligned with the reference signals generated at the targeted frequencies for comparison.



(A) Effect on amplitude envelope when pitch shifting a 1kHz signal an octave down to 500 Hz in Matlab

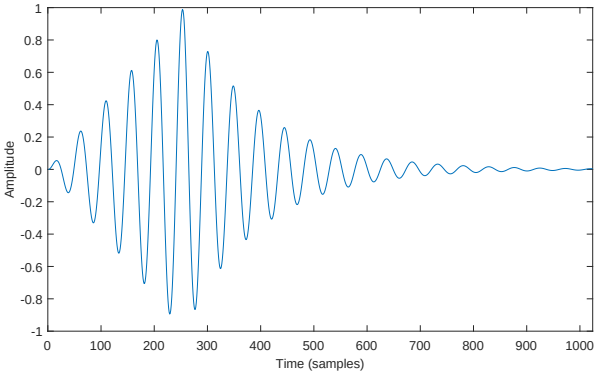


(B) Effect on amplitude envelope when pitch shifting a 500 Hz signal an octave down to 250 Hz in Matlab

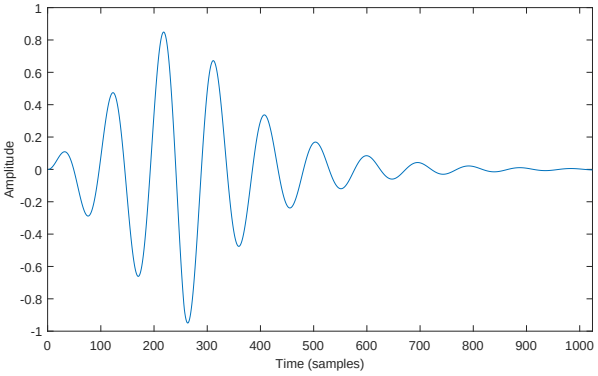
FIGURE 4.42: Effect on amplitude envelope from pitch shifting in signal (@48 kHz) Matlab

This approach clearly introduces a delay and smearing of the attack portion of the envelope in time, and a signal which is no longer in phase with the original signal.

In order to examine the effect of pitch shifting on a MoP model, three reference signals were generated at 1000 Hz, 500 Hz and 250 Hz. The same amplitude envelope with the same attack and release curves were then applied to all three signals. The resulting 1000 Hz and 500 Hz reference signals are presented in Figures 4.43a and 4.43b.



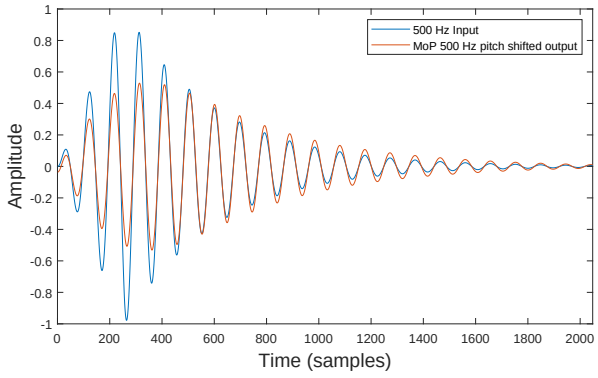
(A) 1 kHz signal with attack and release envelope



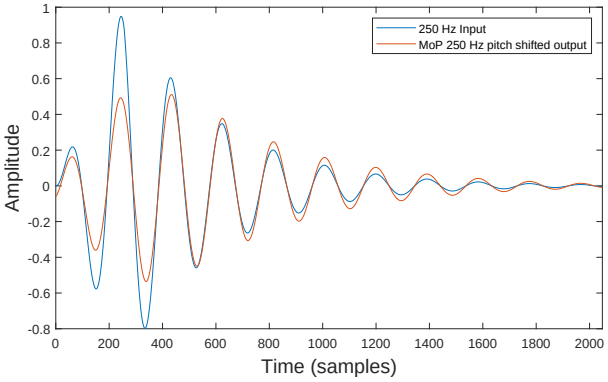
(B) 500 Hz signal with attack and release envelope

FIGURE 4.43: (A) 1 kHz sinusoid with non-monotonic AM and (B) a 500 Hz sinusoid with the same attack and release envelope applied for reference (@48 kHz)

Figures 4.44a and 4.44b display the results of re-synthesising a pitch shifted output signal from MoP estimates of model parameters. The first 8 atoms returned from the decomposition are used to re-synthesise a signal with a frequency one octave below the estimated frequency values, and summed to produce the pitch shifted output.



(A) 1000 Hz MoP decomposition re-synthesised with fEst an octave down compared to 500 Hz signal with same amplitude envelope applied for comparison.



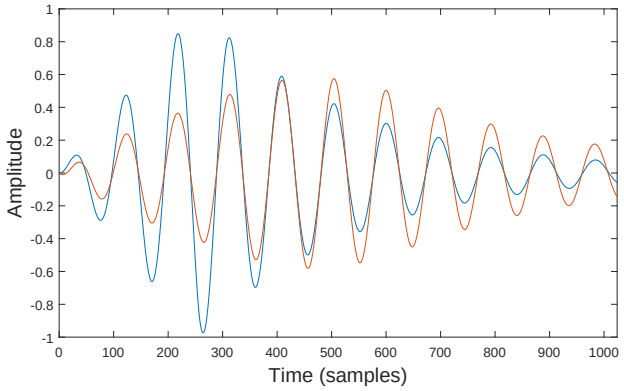
(B) 500 Hz MoP decomposition re-synthesised with fEst an octave down compared to 250 Hz signal with same amplitude envelope applied for comparison.

FIGURE 4.44: Comparison of the effect pitch shifting using MoP parameter estimates has on the resulting amplitude envelope of the pitch shifted signal compared to the reference signal

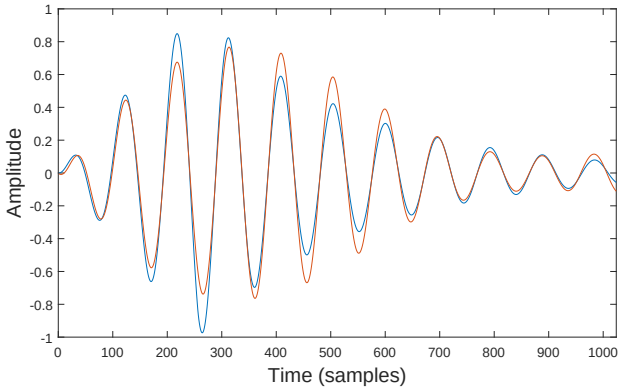
The resulting pitch shifted signal is quite accurate in terms of frequency and phase. The re-synthesised signal does not suffer from any of the pitch shifting issues observed with the Matlab examples. Unfortunately, the amplitude envelope of the re-synthesised signal is not preserved in comparison to the reference signal. Re-synthesis of the signal from modified frequency estimates resulted in a signal which does not match the reference signal with regards to amplitude shape preservation. In order to examine the cause of the effect pitch shifting has on the amplitude envelope, a signal with the same attack and decay curves was synthesised at 500 Hz (one octave down from the original 1000 Hz signal) and model estimates derived from MoP analysis for comparison.

The differences between model estimates from the pitch shifted 500 Hz waveform are compared to the estimates extracted from analysing the 500 Hz reference waveform and presented in Tables A.1, A.2, A.3 and A.4 for comparison. What is clearly shown is that simply changing the frequency parameter of the model estimates, does not result in a pitch shifted signal with the same amplitude envelope. The signal is correctly changed in pitch, but the resulting output's amplitude envelope does not match the original attack and release curves. This is due to the complexities of the lower order atoms and the effect of the constructive and destructive interference of their frequencies.

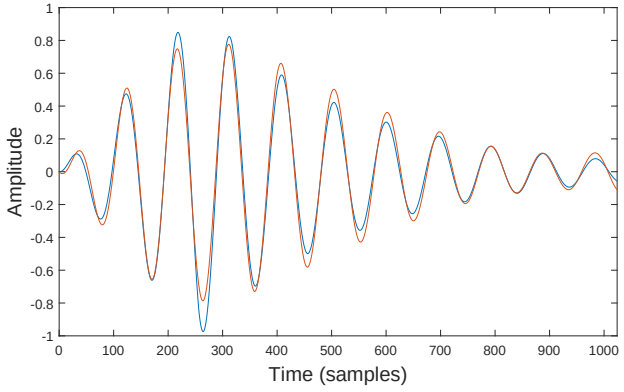
The differences in amplitude estimates are negligible, as are the differences in phase estimates for most atoms. The estimates of amplitude change (daEst) range from -1.77 dB to 2.19 dB with the exception of the 6th atom. The 6th atom has some of the worst differences for all parameter estimates, with a 6.08 dB difference in amplitude change, the largest difference in phase of -0.32 and the second largest difference in the frequency estimate of 87 Hz.



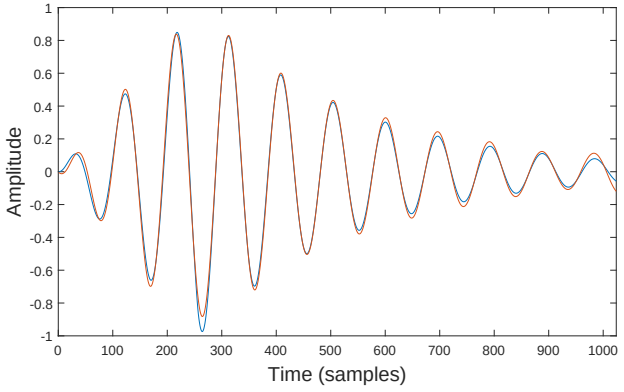
(A) Synthesised signal using modified frequency estimates from original signal



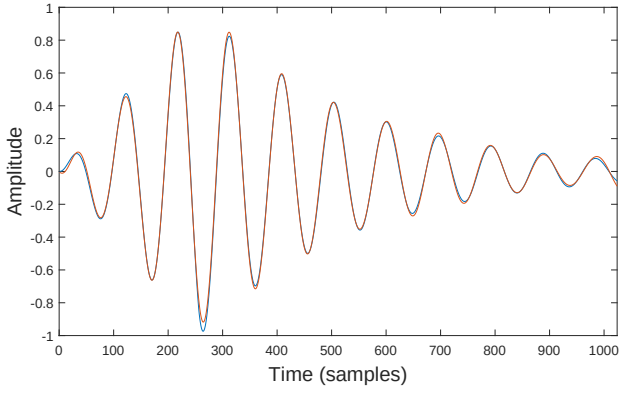
(B) Comparison of using measured frequency estimates of Atoms 2 and 3 from reference signal



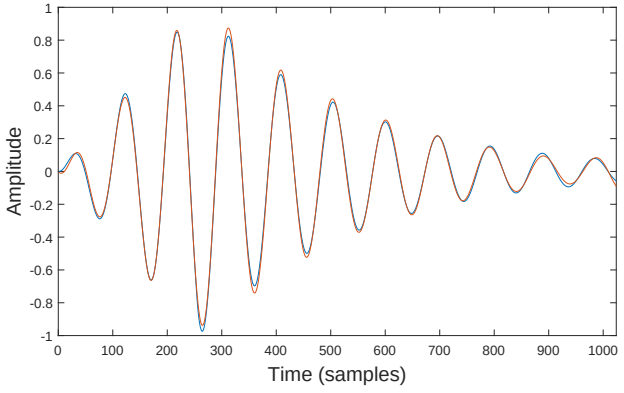
(C) Comparison of using measured frequency estimates of Atoms 2 to 4 from reference signal



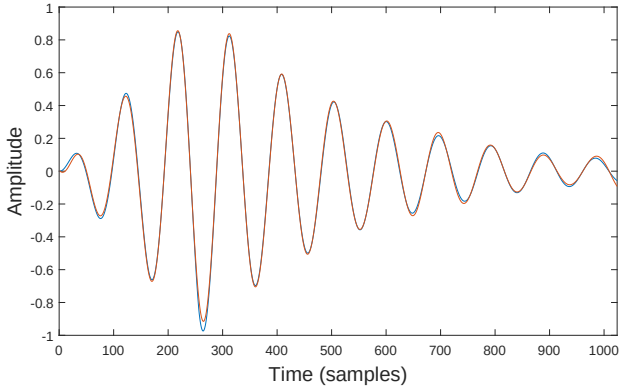
(D) Comparison of using measured frequency estimates of Atoms 2 to 6 from reference signal



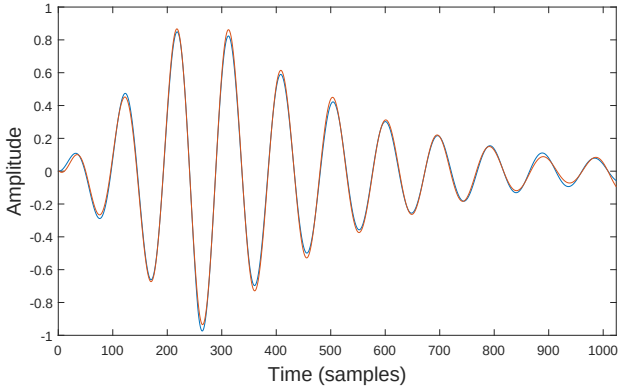
(E) Comparison of using measured frequency estimates of Atoms 2 to 8 from reference signal



(F) Effect of using measured amplitude change (dA) estimates to the resulting signal are minimal



(G) Effect of using measured phase estimates to the resulting signal are minimal



(H) Effect of using measured amplitude, amplitude change (dA) and phase estimates to the resulting signal are minimal

FIGURE 4.45: Comparison of replacing different measured estimates from the reference signal when re-synthesising the pitch shifted signal.

The difference between estimates of the first atom or all parameters is relatively minor and in some cases return the same result. The are no differences in the frequency estimates for the first atom between the pitch shifted and reference signals, but there are differences for the remaining atoms ranging from −116 Hz to 87 Hz. Figure 4.45b shows how even small differences in frequency of −28 Hz and 26 Hz for the second and third atoms make a big difference on the resulting signal due to these atoms having the second and third largest amplitude estimates.

The differences in amplitude change and phase are negligible to the restoration of the amplitude envelope. This is clearly presented in Figure 4.45 and demonstrated by replacing the calculated frequency shifted values with those measured from the reference signal, showing how this affects the output signal in relation to the amplitude envelope of the signal.

Changing parameter estimates of Aest, daEst and Phase from the measured estimates of the pitch shifted signal, to values measured from the reference signal, and re-synthesising the output with these changes make negligible differences to the outputs. Even the 6 dB difference in change in amplitude from the 6th atom has hardly any affect on the resulting output signal.

The only values which directly affect the output of the synthesised waveform against those measured by decomposing the original waveform with the estimated frequency shifted down by an octave (1 kHz to 500 Hz) are the frequency estimates of the atoms. The result of the shifted frequency values do not result in the same values of the referenced signal. This represents a problem when wanting to utilise this model for pitch shifting applications such as changing a modelled kick or bass sounds frequency or the 'key' in which a song is written in, while maintaining the amplitude envelope.

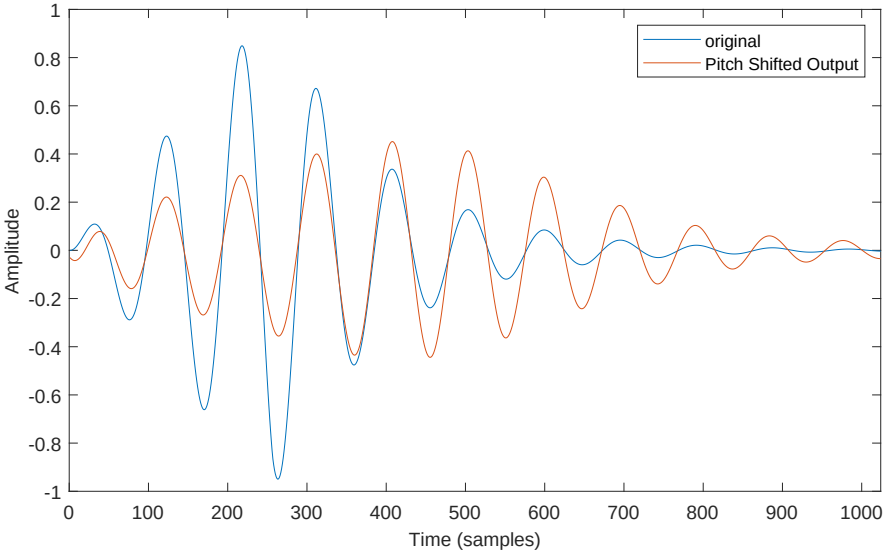
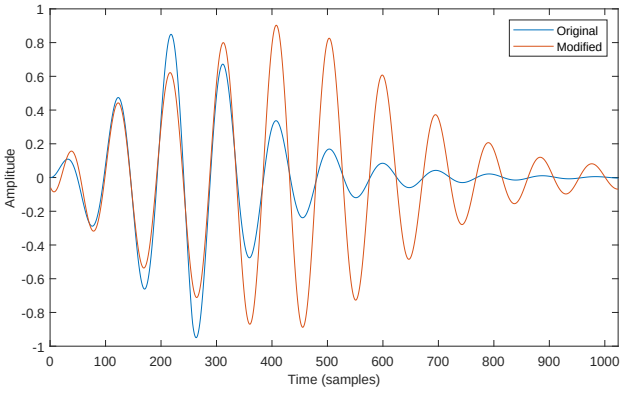
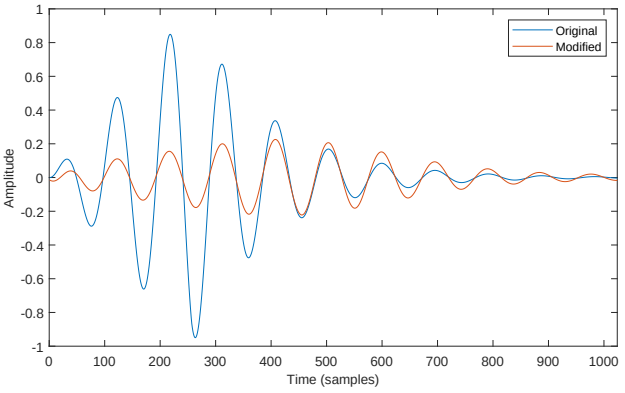


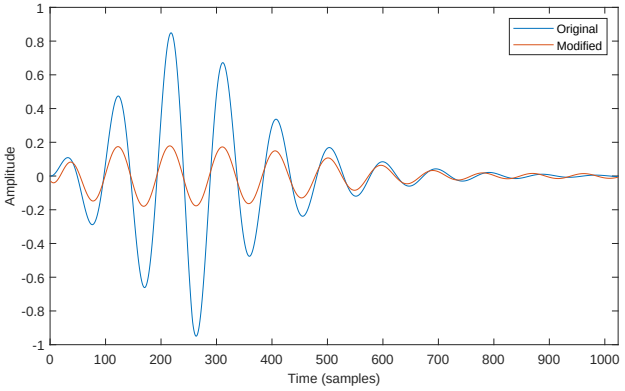
FIGURE 4.46: Comparison of 500 Hz reference signal with pitch shifted signal



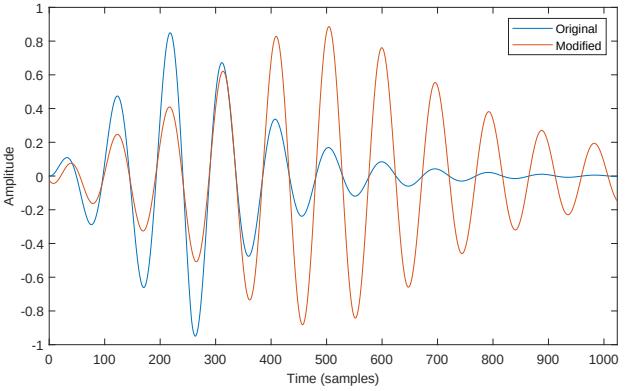
(A) octave down 2xAest



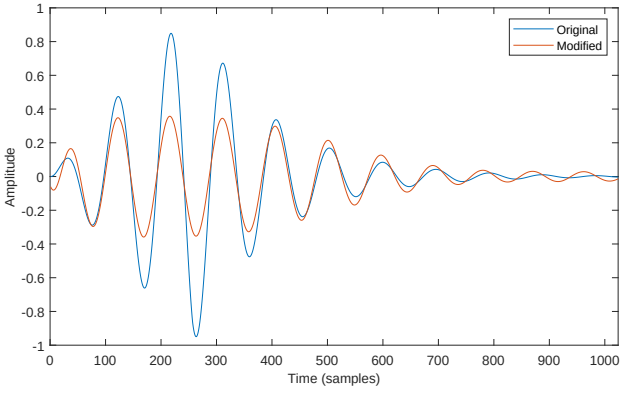
(B) octave down div2Aest



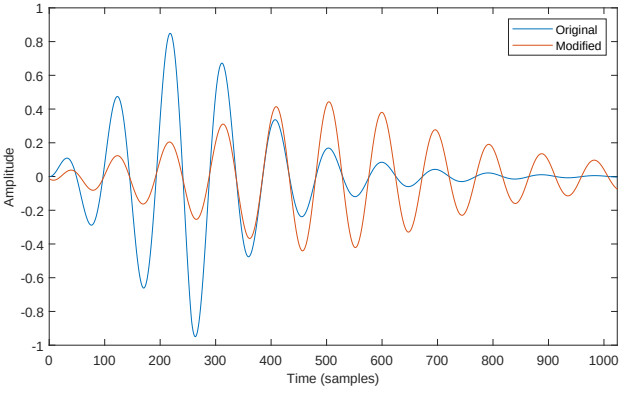
(C) octave down 2xdaEst



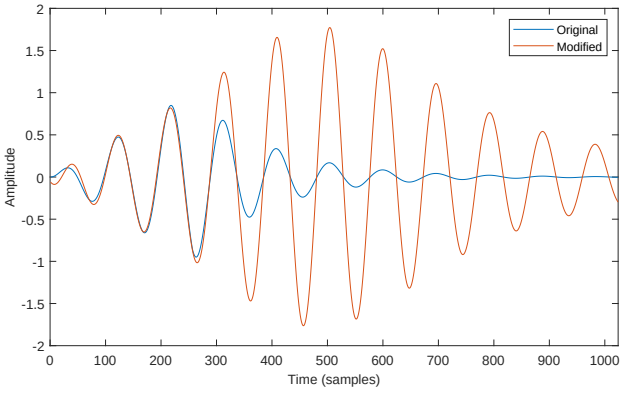
(D) octave down div2daEst



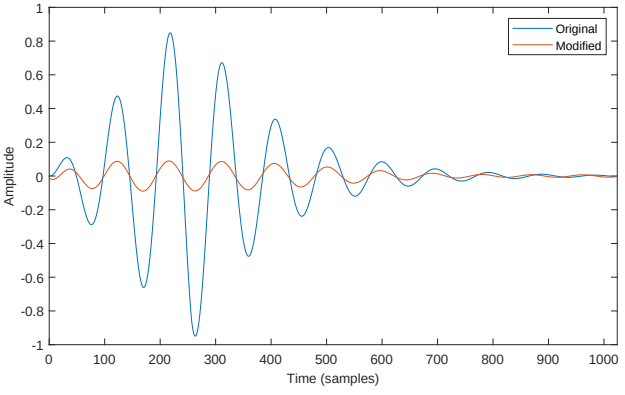
(E) octave down 2xAest 2xdaEst



(F) octave down div2Aest div2daEst



(G) octave down 2xAest div2daEst



(H) octave down div2Aest 2xdaEst

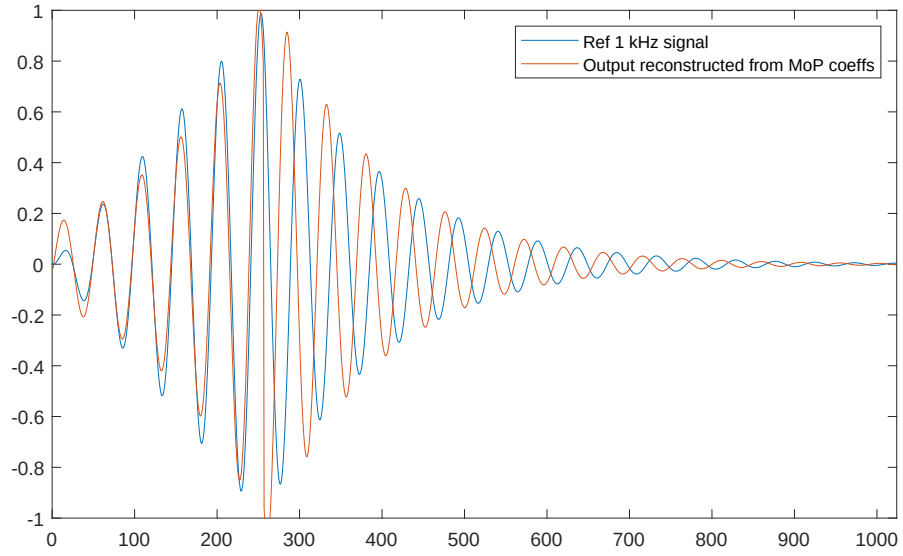
FIGURE 4.47: Comparison of signals synthesised but using modified parameter estimates from original analysis

Figure 4.47 displays the outputs of re-synthesised signals with multiple changes to a combination of estimated model parameters, compared against the 500 Hz reference signal with the same amplitude envelope applied to the 1 kHz signal. All output signals have been re-synthesised using the frequency estimates derived from the MoP decomposition of the 1 kHz sinusoid and pitch shifted an octave down to 500 Hz, which as discussed and displayed in Figure 4.46 results in a well aligned pitch shifted signal, but the amplitude envelope is not preserved. Adjusting estimated parameters of amplitude and change in amplitude results in some interesting changes to the output signals, but although these changes result in some interesting transformations, they do not help retain the original amplitude envelope.

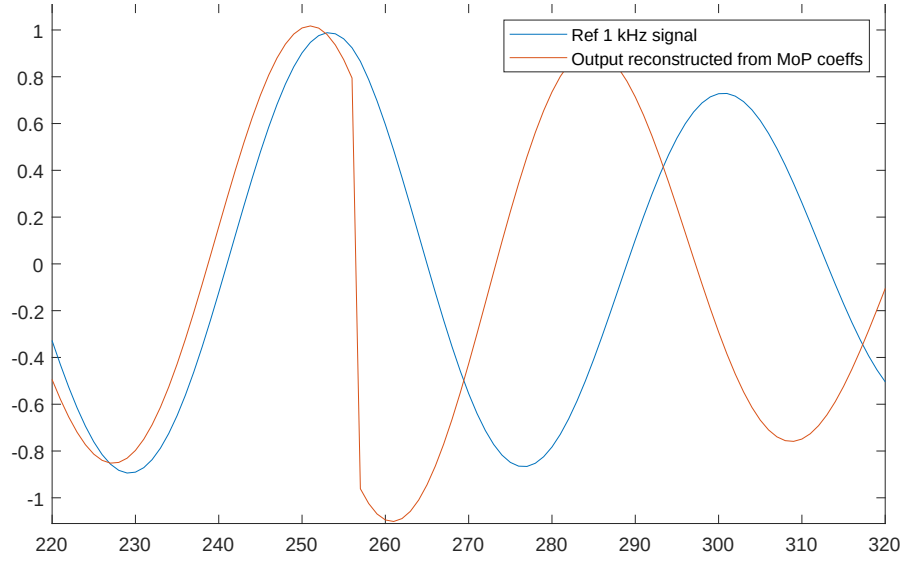
It has been presented that a non-overlapping single frame analysis framework is able to model complex sounds and complete music tracks using modelled pursuit. This method works well for monotonic amplitude changes where changing the pitch is accomplished by simply changing the frequency estimates. However, in the presence of non-monotonic amplitude change, even though the model is able to represent the signal accurately, performing pitch modifications on non-monotonic sinusoidal components modelled with a number of atoms does not maintain the amplitude envelope of the output. Further investigation is required to evaluate the effect of this on changing kick and bass frequencies while maintaining the low-end quality.

The above examples were re-synthesised using a single analysis frame, performing pitch shifting in a real system with multiple frames requires the phase of the sinusoidal atoms to maintain coherence across frame boundaries. Pitch shifting is a time based effect, the effect of changing a sinusoids pitch has an affect on the periodicity and therefore the phase of the signal. Phase estimates from a new analysis frame after pitch shifting has been applied to the previous frame will no longer be valid. The value of the phase for the re-synthesised sinusoids needs to be stored and used with an additional phase increment related to the pitch of that sinusoid at the start of the new re-synthesis frame to maintain phase coherence.

This is clearly demonstrated in Figure 4.48 where phase coherence is not applied. The model used in this thesis performs re-synthesis on a sample by sample bases, therefore the phase value at the end of the frame as well as the value of the phase increment is readily available.



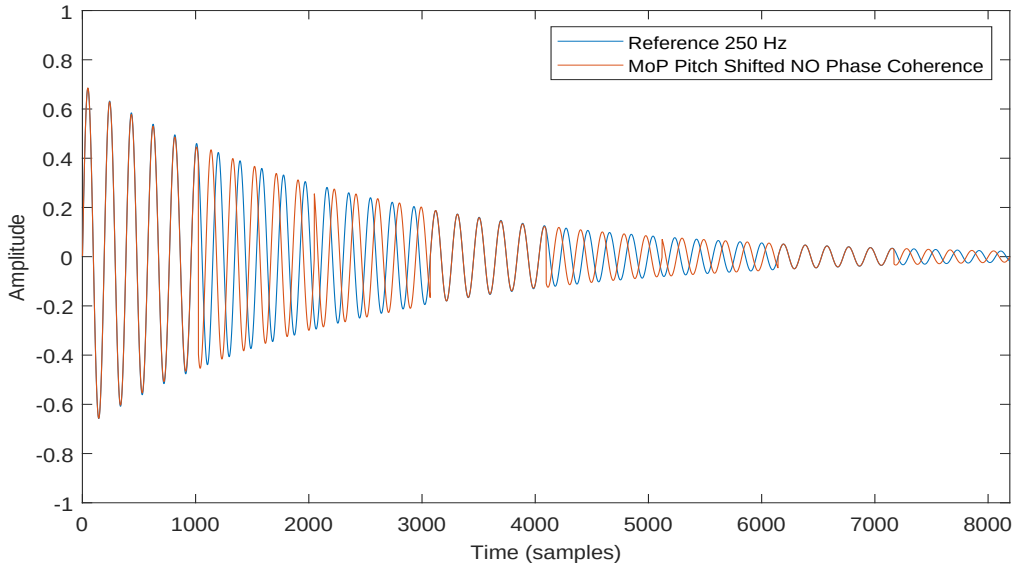
(A) Pitch shifted sinusoid using phase estimates from decomposition does not match the reference signal



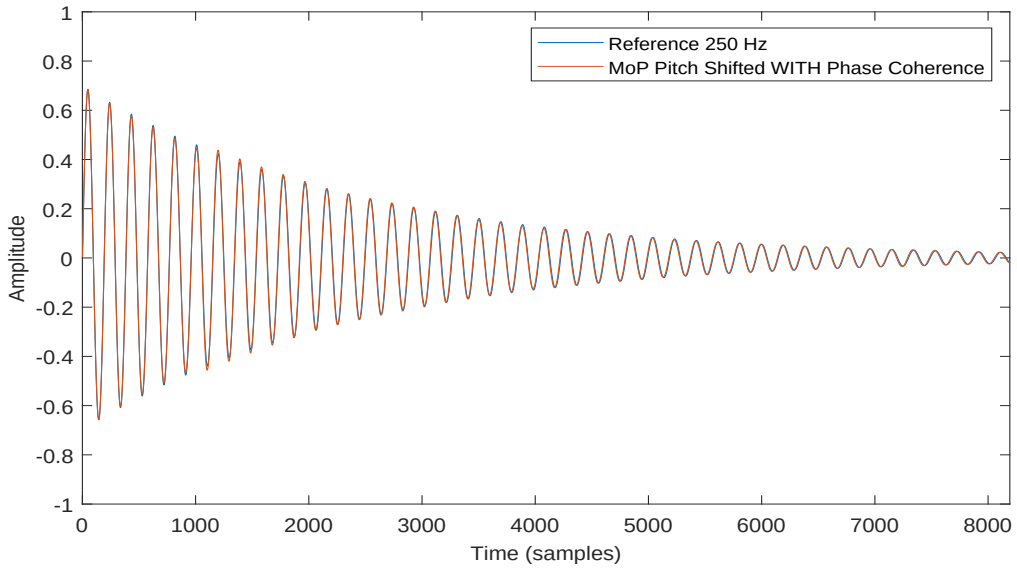
(B) Closer inspection shows the phase discontinuity resulting in the incorrect output

FIGURE 4.48: Pitch shifting has an affect on phase, therefore using estimates of phase from successive frames causes phase discontinuities

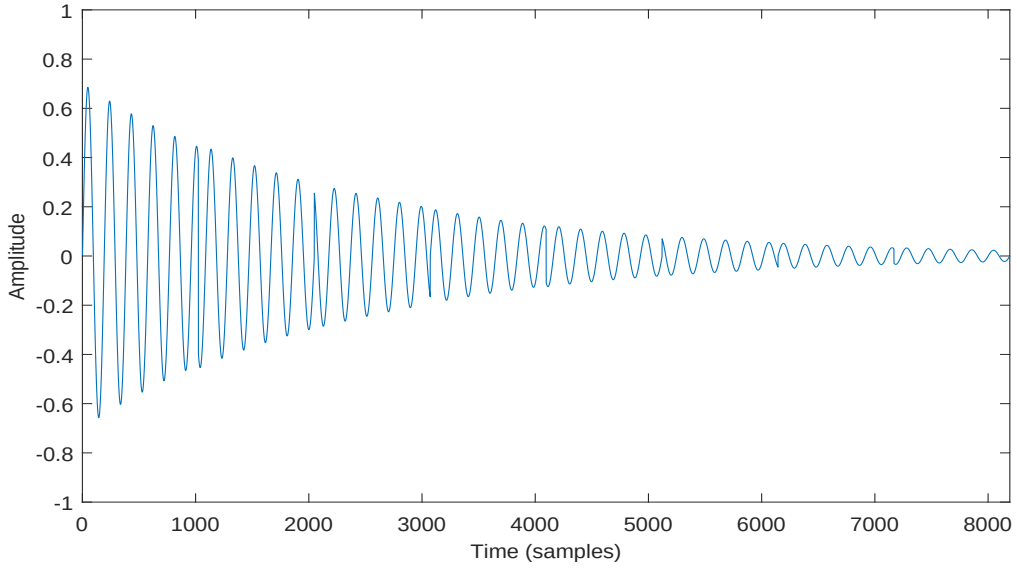
Another example of pitch shifting with and without phase coherence is presented in Figure 4.49 where the output signal with and without maintaining phase coherence is compared with the referenced output signal. Figure 4.49a displays the errors between the two signals when not maintaining phase coherence between frames, while Figure 4.49b displays the correct output when phase coherence is maintained. Figure 4.49c displays the incorrect output on its own for a more detailed view of the discontinuities.



(A) Comparison of reference signal and pitch shifted output using original phase estimates results in discontinuities at frame boundaries



(B) Comparison of reference signal and pitch shifted output while maintaining Phase Coherence



(C) Output without maintaining Phase Coherence

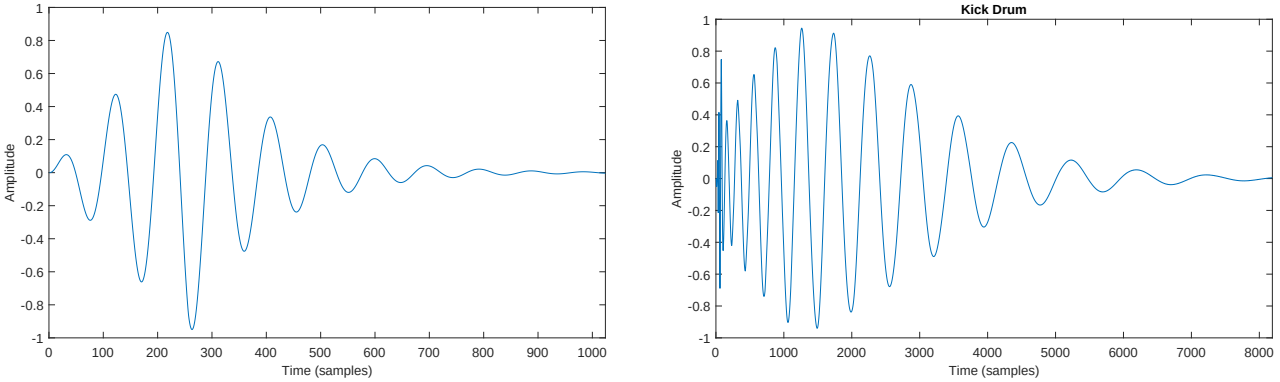
FIGURE 4.49: Comparison of Pitch Shifting with and without maintaining Phase Coherence

4.5.6 Maintaining Amplitude Envelope

In the previous section, pitch shifting using estimates of parameters derived from MoP were discussed in detail. The current implementation of MoP is based on estimating monotonic amplitude change. Future developments could extend this to include dictionaries with higher order amplitude modulations, but the current implementation, although able to perform pitch shifting accurately, does not preserve the amplitude envelope after pitch shifting when non-monotonic amplitude changes are present within the analysis frame.

One possible solution to maintaining amplitude preservation when pitch shifting using a system which is designed to model monotonic amplitude change, could potentially split an analysis frame containing non-monotonic amplitude change into two separate frames, one frame containing the attack part of the sound, and the second frame containing the rest of the signal, including the release section of the signals amplitude envelope.

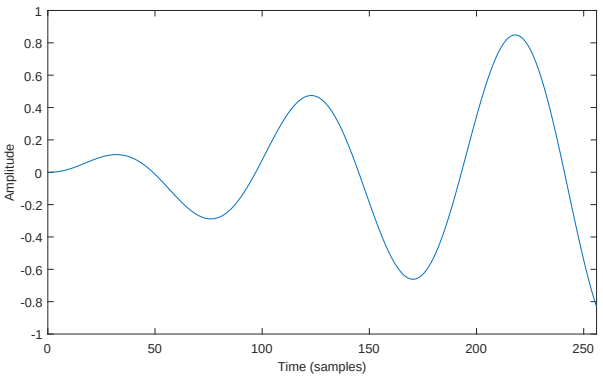
Figure 4.50 displays a reference signal of a 500 Hz sine wave with an attack and release. Although this is a simple signal which is used in testing, kick drums can be quite similar as shown in Figure 4.50b.



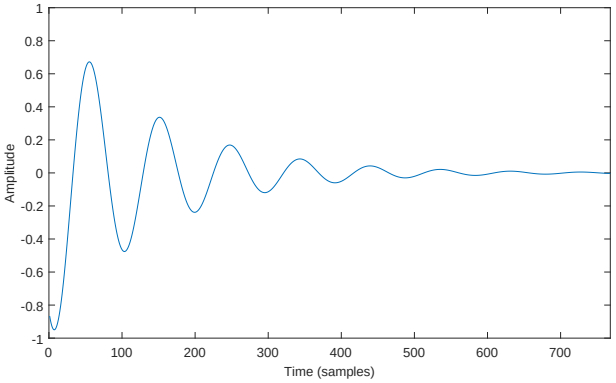
(A) 500 Hz reference signal with non-monotonic AM (B) Kick Drum

FIGURE 4.50: 500 Hz reference signals and Kick Drum (@48 kHz)

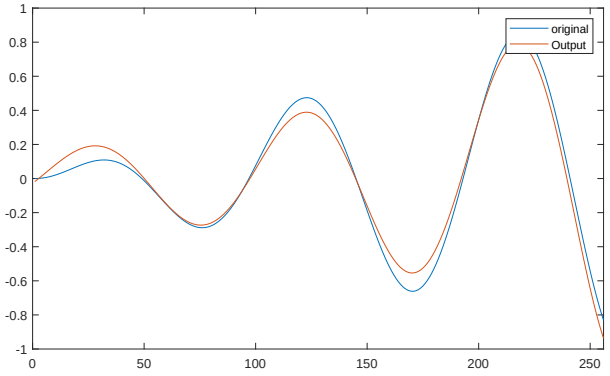
Figure 4.51 displays the 500 Hz test signal split into two frames. Figure 4.51a displaying the attack portion of the sound and Figure 4.51b displaying the release part. The signal has now been segmented into two parts containing only monotonic amplitude change. Figures 4.51c and 4.51d show the output of the MoP analysis. The results of the shortened attack frame are not as accurate as the larger release frame due to the resulting resolution in frequency but this can be improved. Figure 4.52 displays the result of the 500 Hz signal pitch shifted up an octave to 1 kHz and compared to a reference signal with the same amplitude envelope.



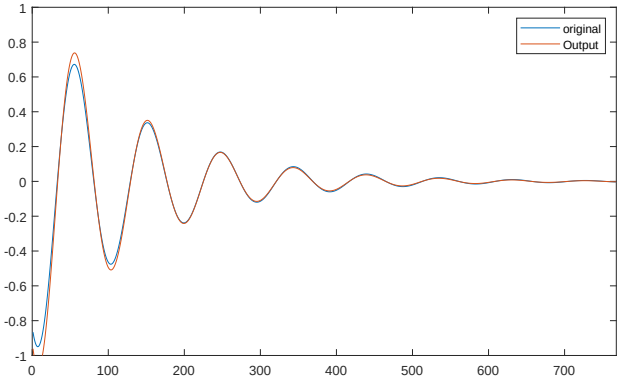
(A) 500 Hz reference signal Attack



(B) 500 Hz reference signal Release



(C) MoP output of Attack frame



(D) MoP output of Release frame

FIGURE 4.51: MoP output of split signal into the attack and release parts

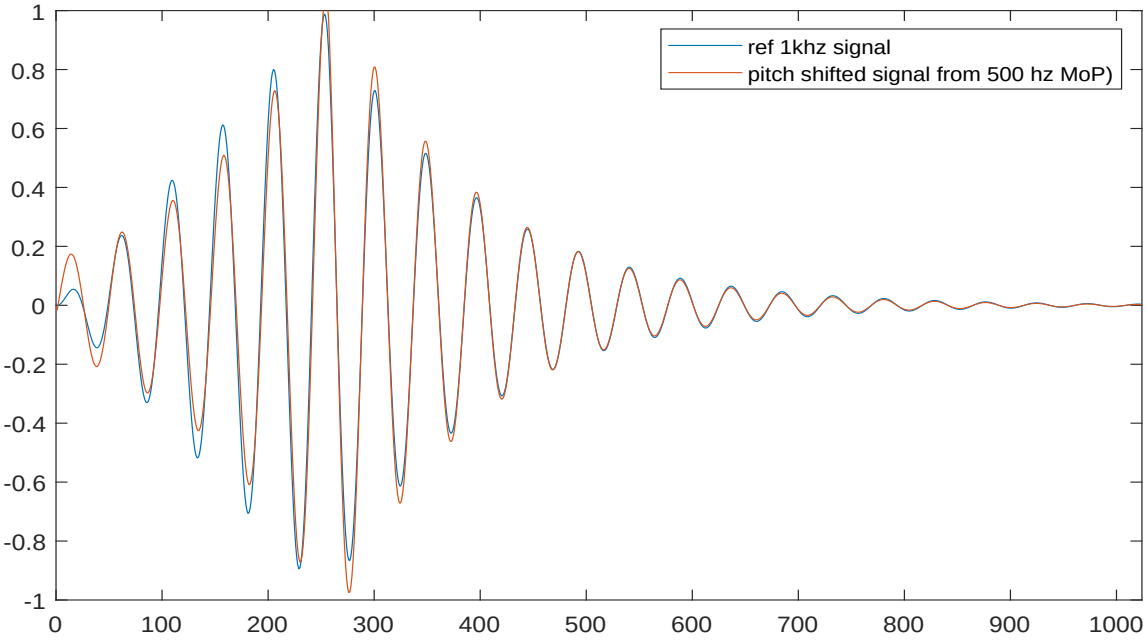
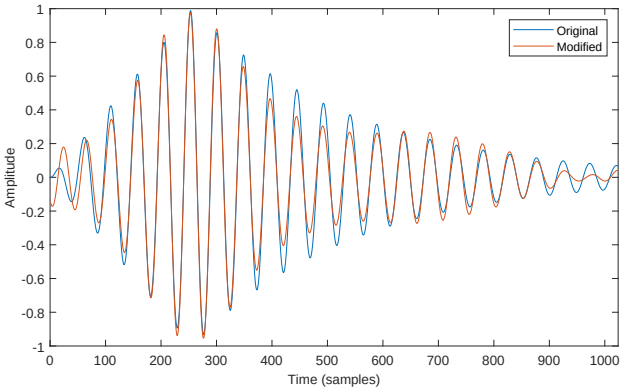


FIGURE 4.52: Pitch shifted output using split frames and phase coherence maintains the amplitude envelope of the pitch shifted signal

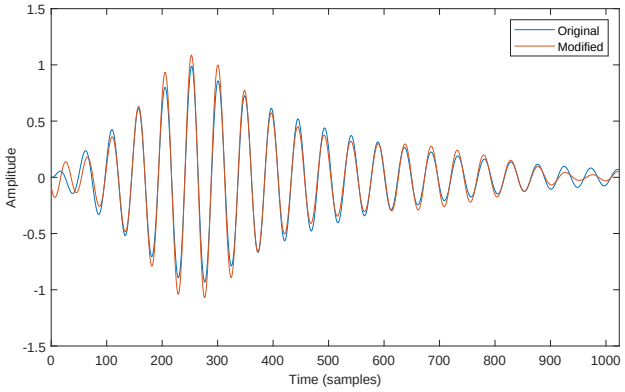
The results the MoP decomposition are not exact, and therefore the resulting pitch shifted signal is not a perfect match to the reference signal. However, the amplitude envelope has been preserved which provides a possible solution for pitch shifting using MoP with amplitude envelope preservation.

4.5.7 Performing modifications on atoms

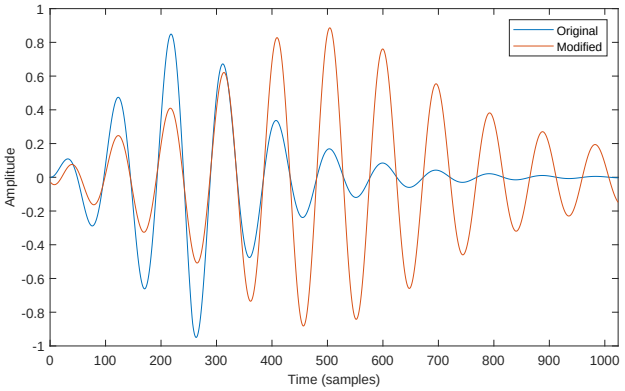
An interesting observation resulting from the above inspection of re-synthesising pitch shifted signals with varying changes to amplitude and estimates of amplitude change, as shown in Figure 4.47, is the possibility to apply interesting modifications to the re-synthesised signal by changing specific parameters of different atoms before re-synthesis. Figure 4.53 demonstrates how the amplitude envelope of a models output can be manipulated by adjusting a couple of atoms amplitudes which directly affects the constructive and destructive interference and therefore the amplitude envelope of the resulting synthesised sound.



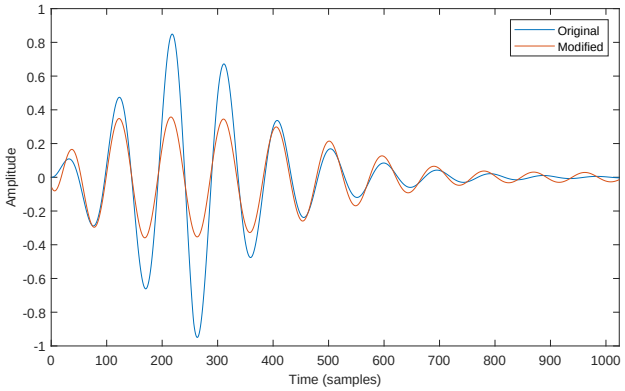
(A) Input signal and output signal with modified amplitude envelope from adjusting amplitudes of specific atoms from the decomposition before resynthesis



(B) Input signal and output signal with modified amplitude envelope from adjusting amplitudes of specific atoms from the decomposition before resynthesis



(C) Input signal and output signal with estimates for amplitude change decreased by a factor of 2



(D) Input signal and output signal with estimates for amplitude and amplitude change doubled

FIGURE 4.53: Comparison of the effect changes in parameter estimates have on output

4.5.8 Examining effect of Time Stretching

Stretching a signal in time without altering the pitch is a simple process for signals containing monotonic amplitude change as presented in 4.54.

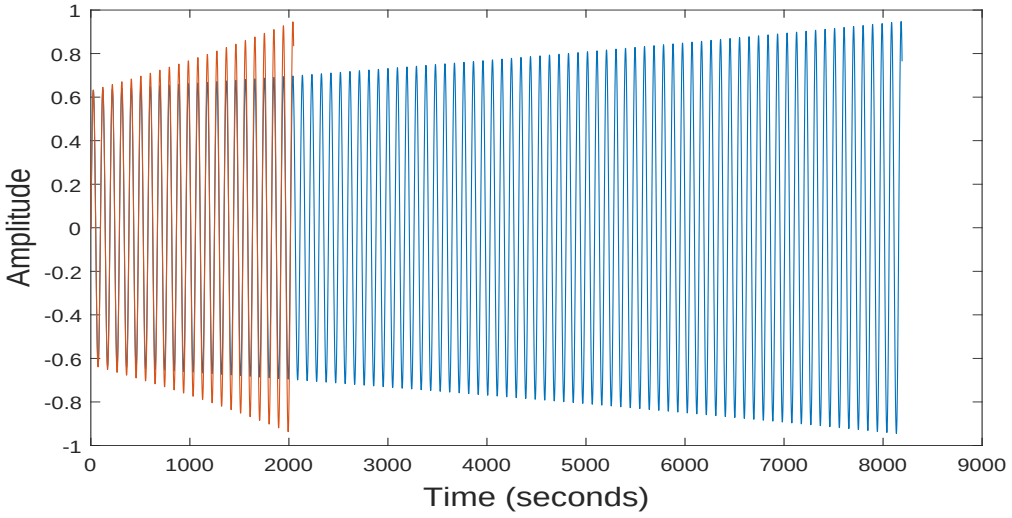


FIGURE 4.54: A sinusoid (@48 kHz) with monotonic amplitude change is time stretched as expected

Signals containing non-monotonc amplitude change, modelled with MoP, encounter similar issues when time stretching as those discussed in Section 4.5.5 regarding pitch shifting. Time stretching a signal down to a shorter length produces reasonable results with the exception that the amplitude envelope is not preserved as shown in 4.55.

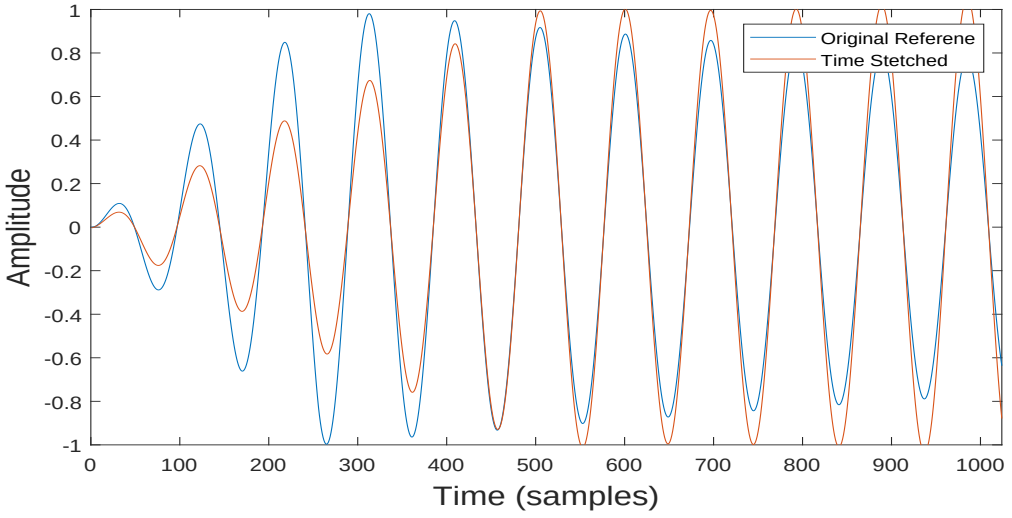
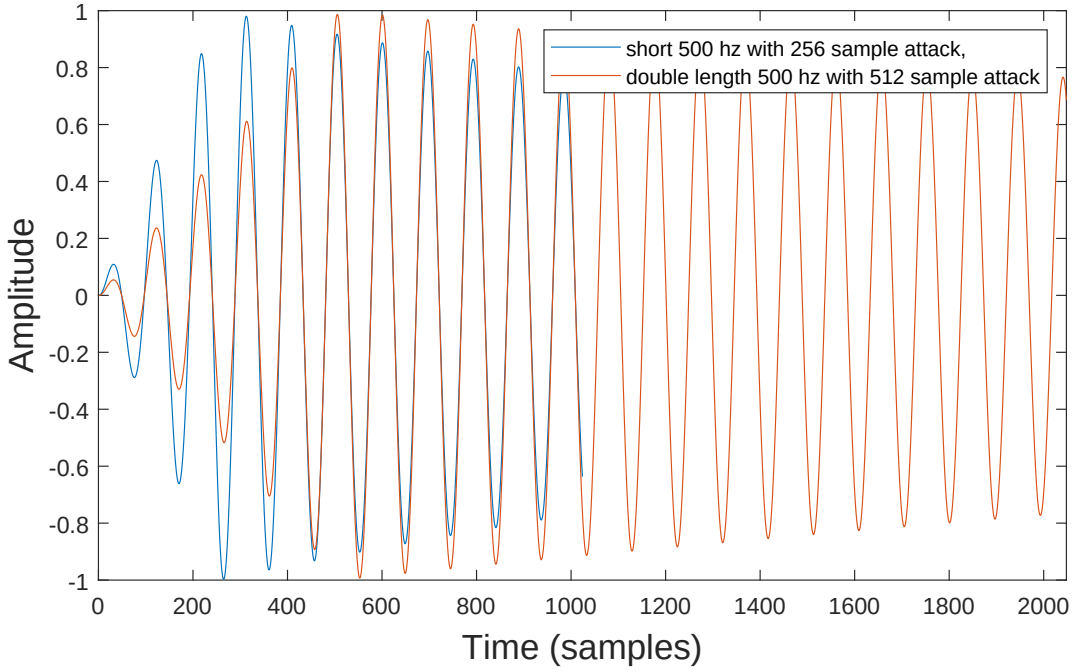
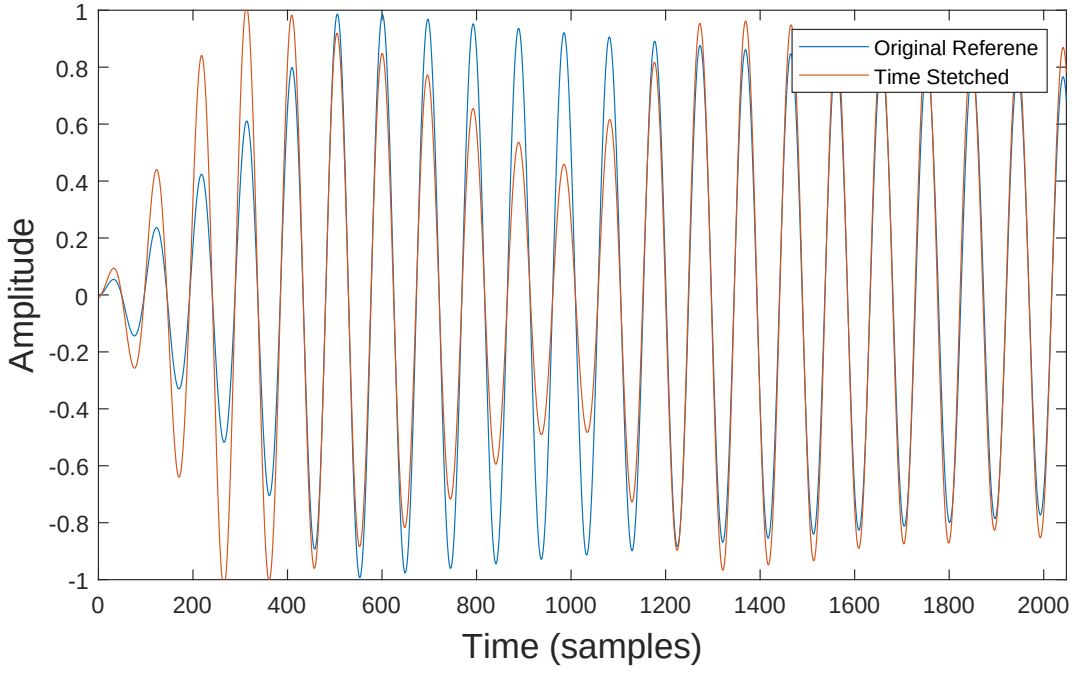


FIGURE 4.55: Non-Monotonic signals stretched down work reasonable well with the exception of the amplitude envelope not being preserved

Two reference signals are displayed in 4.56a for comparing the output of increasing a signal in time from atoms derived from MoP. A 500 Hz signal with an attack time of 256 samples is displayed against a signal representing the expected output when doubling the duration of this signal in time.



(A) Reference Signals for examining Time Stretching



(B) Non-Monotonic signal with length doubled in time

FIGURE 4.56: 500 Hz input sinusoid (@48 kHz) with a length of 1024 samples displayed against (A) a reference signal twice as long with the attack and release times updated appropriately, and (B) where this reference signal is compared to the time stretched output from applying the time stretching to all MoP atoms.

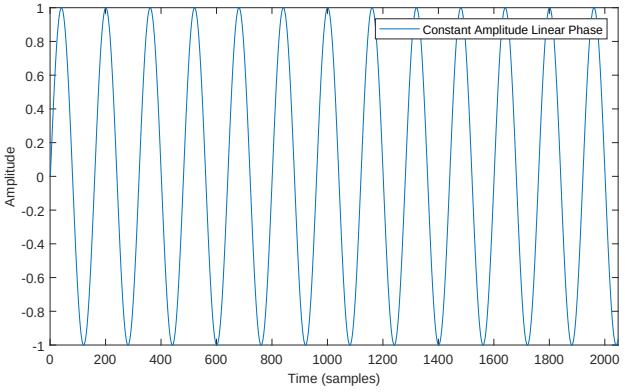
Unfortunately, extending the duration of a signal in time from parameter estimates derived from a MoP decomposition, in the presence of non-monotonic amplitude change, results in a signal with some unexpected amplitude modulation in the middle of the extended signal, as shown in Figure 4.56b. The presence of the amplitude modulating in the middle of the time stretched output signal is caused from the constructive and destructive interference of the underlying sinusoidal components derived by the MoP decomposition. Time stretching these sinusoidal atoms results in a similar problem to that which occurred when examining pitch shifting, where the amplitude envelope of the resulting output signal did not match that of the reference signal.

The modification of non-monotonic sinusoidal components within the non-transient stage of the signal being modelled, does not result in the expected time-stretched or pitch shifted outputs. A possible solution to modelling non-monotonic amplitude change when an audio frame contains both an attack and release section of the ADSR amplitude envelope was presented in 4.5.6 where the attack and release sections are separated into two frames containing only the positive or negative amplitude change. A MoP decomposition of these two separated frames is then able to model the signal with the monotonic amplitude change estimates. Transformations can be performed on the separate frames without the errors introduced in the presence of non-monotonic amplitude change, and then combined.

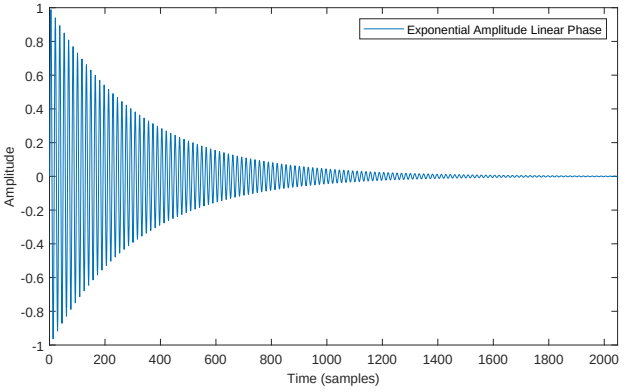
The estimation methods for estimating amplitude change presented in Chapter 3 are derived from monotonic amplitude change which imposes this limitation on the system. Adaptive decomposition's such as the Quasi-Harmonic Model (QHM) [201] and the extension of QHM, the extended adaptive Quasi-Harmonic Model (eaQHM) [202] [168] are able to incoporate non-monotonic amplitude and frequency modulations within the model through iterative refinement. The Distributed Derivatives Method [203] [204] incorporate non-monotonic amplitude and frequency modulations in the model using higher order polynomial parameter estimation. Extending the current base atoms used by MoP to include higher order polynomials within the decomposition remains a further research directive.

Although the manipulation of multiple atoms from an over-complete decomposed using MoP can provide some interesting transformations, the resulting time and pitch-scale modifications are not accurate. Kick sounds in general contain a basic fundamental frequency which changes rapidly at the start, followed by a much more gradual monotonic amplitude and frequency change within the lager semi-steady state section of the sound. The System Implementation presented in Chapter 6 describes in detail how MoP is therefore implemented to find a single best fitting monotonic atom for each spectral peak. Any remaining amplitude and frequency modulations from higher order modulations for those spectral peaks, remain within the residual signal for further analysis.

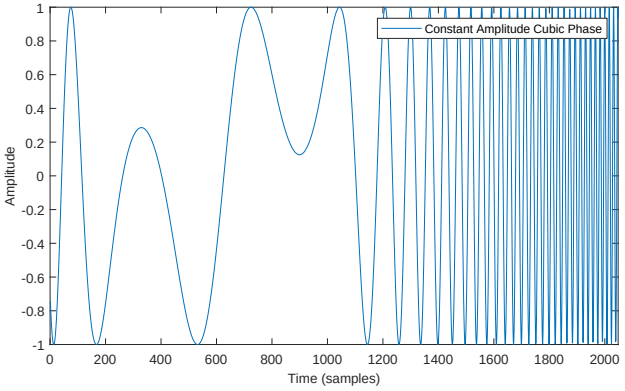
4.5.9 Synthetic Non-Stationary Testing



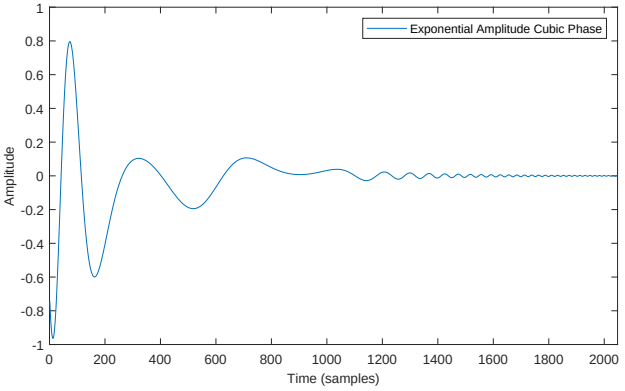
(A) Constant Amplitude Linear Phase



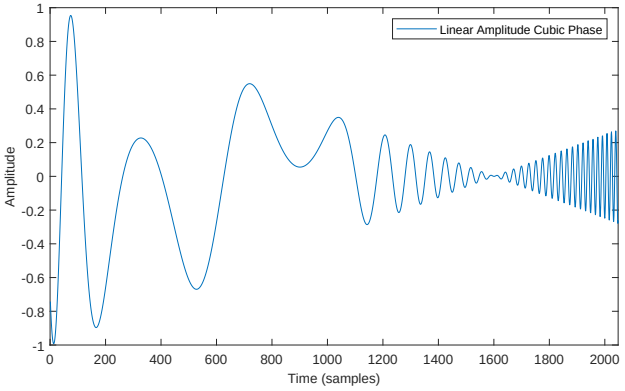
(B) Exponential Amplitude Linear Phase



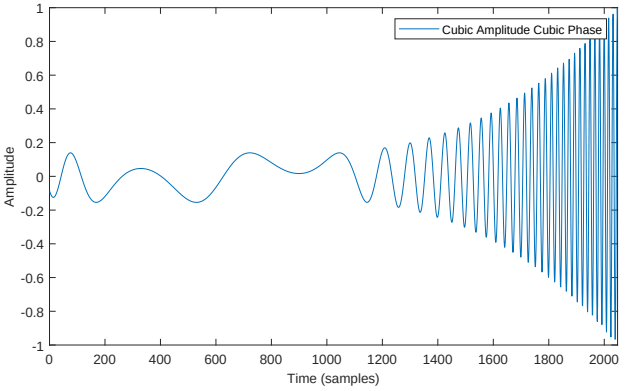
(C) Constant Amplitude Cubic Phase



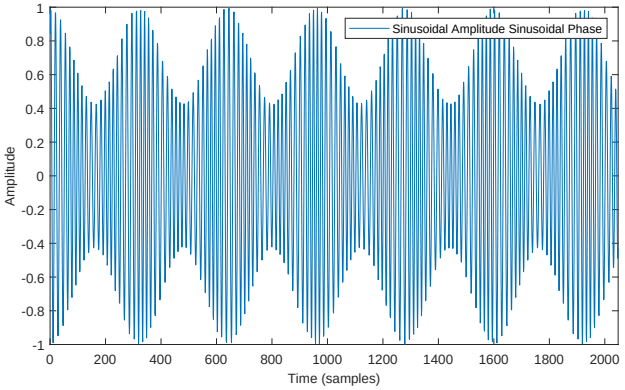
(D) Exponential Amplitude Cubic Phase



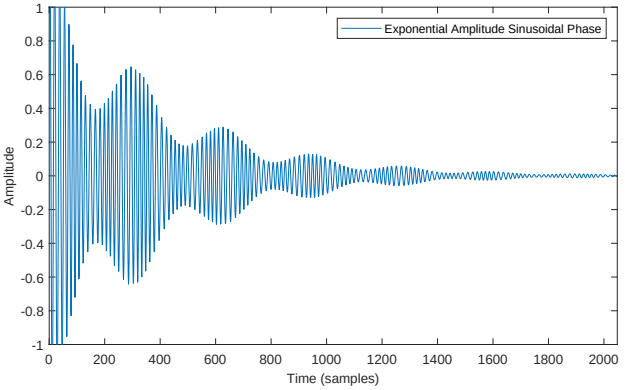
(E) Linear Amplitude Cubic Phase



(F) Cubic Amplitude Cubic Phase

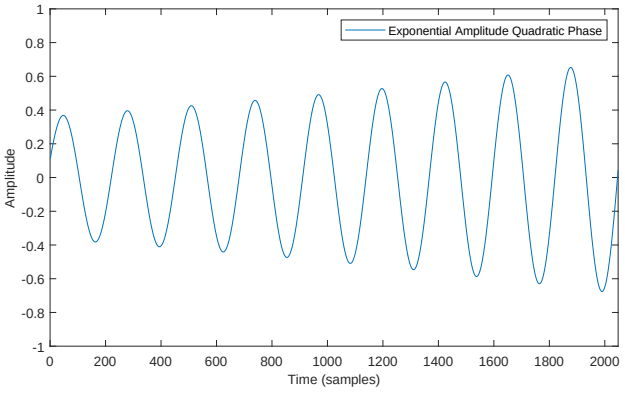


(G) Sinusoidal Amplitude Sinusoidal Phase

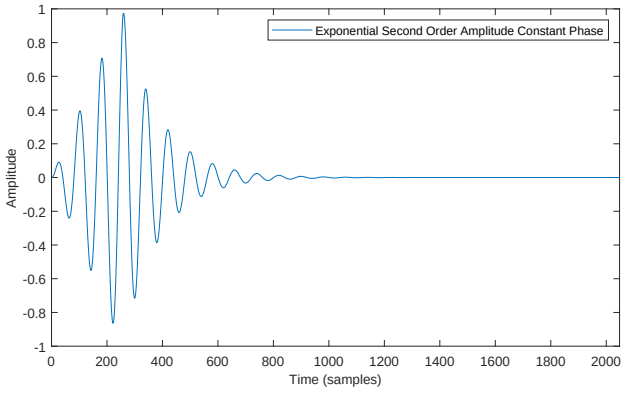


(H) Exponential Amplitude Sinusoidal Phase

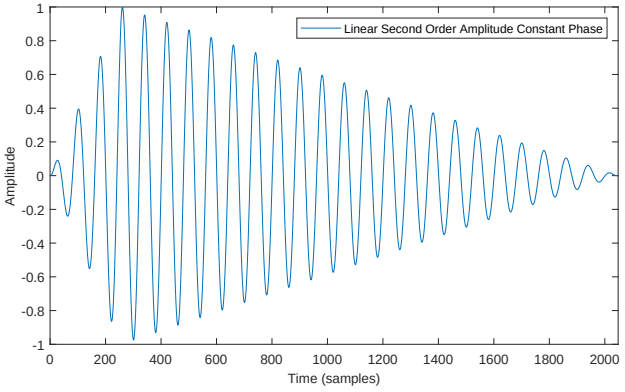
FIGURE 4.57: Illustration of 8 of the waveform used for testing



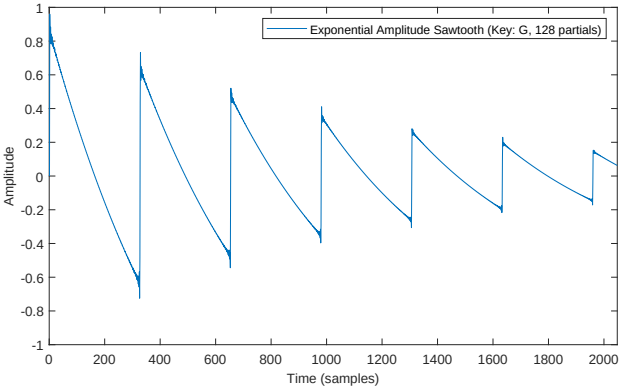
(A) Exponential Amplitude Quadratic Phase



(B) Exponential Second Order Amplitude Constant Phase



(C) Linear Second Order Amplitude Constant Phase



(D) Exponential Amplitude Sawtooth (Key: G) synthesised with 128 partials)

FIGURE 4.58: Illustration of remaining 4 waveforms used for testing

In [142] different analysis techniques are compared against a number of synthetic non-stationary sinusoids with different amplitude and frequency modulations. The performance of eaQHM, exponentially damped sinusoids estimated with ESPRIT, and log-linear-amplitude quadratic-phase sinusoids estimated with the Reassignment method are compared. These tests have been repeated with the addition of MoP using a non-overlapping framework, and the inclusion of three additional signals, two with second order amplitude modulations and a third sawtooth signal with exponential amplitude change. The 12 test signals are displayed in Figures 4.57 and 4.58, A table containing the waveform names and abbreviated labels, which are sometimes referred to is given in Section A.2. More details on the first 9 signals can be found in [142] including the equations used for generating the sinusoidal signals.

The default sample rate for all test signals was set to 16 kHz. The settings for MoP for all of the tests used a frame size of 512 samples, and a maximum of 64 sinusoidal partials allowed. MoP performed reasonably well. The average SRER over all the test signals was 44.5 dB, but the end threshold

specified for ending the decomposition was also set to −60 dB which would have limited the results to this level of accuracy. The range of SRER values returned for different signals ranged from a maximum of 64 to 40 with the exception of the Exponential Amplitude Cubic Phase (EA-C3P) which was not modelled accurately with these settings, but improved to 50 dB when increasing the frame size to 1024 and maximum number of partials to 128. Increasing the ending threshold in this case would have resulted in even better results.

The standard deviation of the SRER measurements for MoP ranged from 13 to 9 dB. In comparison the standard deviation for eaQHM's SRER results ranged from 82 to 76 dB. Reassignments standard deviation ranged from 20 to 15 dB with EDS having similar results, ranging from 20 to 18 dB. eaQHM produced the best overall mean result from all test signals of 94.96 dB using a hop size of 8 samples, and a frame size of 16 samples, however the standard deviation with this result was still 78 dB. The best result returned from Reassignment was only 30.75 dB using a hop size of 15 samples and a frame size of 32 samples. EDS performed slightly better with the best result of 48 dB resulting from using a hop size of 8 samples and a frame size of 32 samples. In general eaQHM, Reassignment and EDS all returned the best results when using the smallest hop and frame sizes used in testing of 8 and 32 samples respectively, the only exception being the results from Reassignment where using the smallest setting resulted in a negligible difference of 30.68 dB compared to 30.75 dB using a hop size of 15 samples, but using a hop size of 15 samples reduced the MIPS by half, from 0.33 to 0.16 seconds.

It should be noted that by default the Matlab script from [142] does not use the same default window length for each method and differs between test signals. eaQHM always uses half of the default window length during its analysis; Reassignment uses a frame size less 8 samples when evaluating Exponential Amplitude Linear Phase (EA-LP) and a frame size four times larger than the default when evaluating Exponential Amplitude Quadratic Phase (EA-QP), but otherwise uses the default parameters. EDS always uses the default hop and frame size parameters, while MoP uses a fixed frame size of 512 samples with no overlap for all test cases.

The default settings from the Matlab script from [142] uses a hop of 1 ms, and the frame size set in relation to the sample rate and frequency of the analysis signal expressed as:

$$\text{frame size } = \left\lfloor 3 \times \frac{fs}{f_0} \right\rfloor \tag{4.15}$$

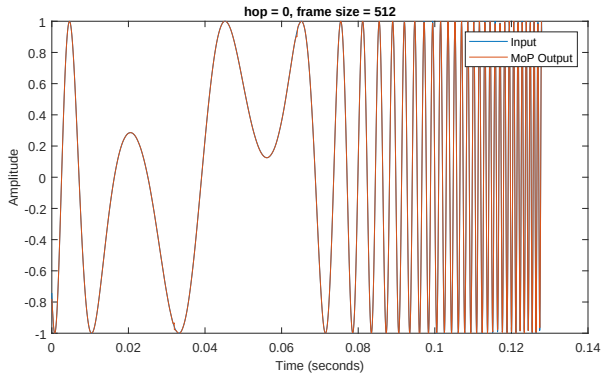
The standard deviation for MoP remains relatively constant for MIPS with measurements resulting in deviations of between 0.25 and 0.24. The standard deviation of eaQHM is more variable than the other methods with the standard deviation ranging between 0.17 and 0.03. Reassignment's measurements of the standard deviation in MIPS ranged between 0.04 and 0.03. EDS's standard deviation ranged between 0.29 and 0.13, with a worst case outlier from default frame and hop size setting measured at 0.778.

The test conducted with MoP were done with a default frame size of 512 samples, and a maximum of 64 sinusoidal partials. Increasing the maximum number of partials in the case of these relatively simple signals had a minimal effect on the MIPS. The average overall MIPS increased from 0.3613 seconds to 0.44771 seconds, with an increase the maximum number of partials to 128. Increasing the frame size to 1024 samples with the maximum number of partials increased to 128 decreased the MIPS to 0.4079 seconds. Increasing the maximum number of partials improved the SRER results slightly, with the maximum number of partials set to 128, but the ending threshold limit of -60 dB would again be a limiting factor. The average SRER of 44 dB increased to 46 dB when increasing the frame size to 1024 samples. The effect of changing the frame size and maximum number of partials of the MoP decomposition had a minimal effect on the quality of most of the signals being tested. The exceptions being the Exponential Amplitude Cubic Phase (EA-C3P) test signal. With the default frame size of 512 samples and a maximum number of 64 partials, the SRER returned was only 6 dB. Increasing the frame size to 1024 samples and the max number of partials to 128 improved the SRER of this signal to 50 dB. This change unfortunately had a detrimental effect on the Constant Amplitude Cubic Phase (CA-C3P) signal, where the SRER dropped from 41 dB to 23 dB. The decrease in quality to the Constant Amplitude Cubic Phase (CA-C3P) signal resulting from increasing the frame size and maximum number of partials is presented in Figures 4.59a and 4.59b. Alternatively, the increase in quality of this change to a Exponential Amplitude Cubic Phase (EA-C3P) is presented in Figures 4.59c and 4.59d.

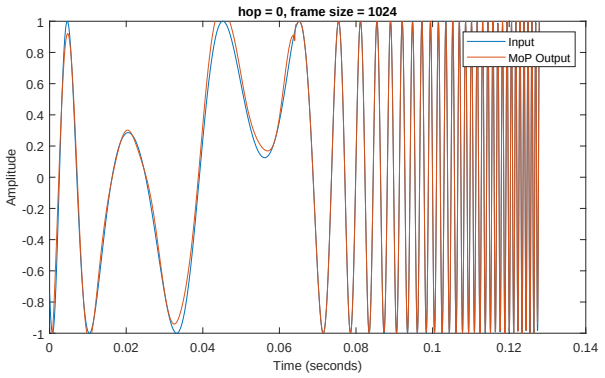
The tests conducted for eaQHM, Reassignment and EDS varied the frame size and hop amount between tests with smaller hop and frame sizes generally performing the best.

"Nonstationary sinusoids with time-varying frequency are more challenging to model with longer windows. The modeling performance of eaQHM and EDSM decreased when L increased for all the synthetic nonstationary sinusoids" [142]. A hop size of 5ms was originally presented in [205], and as little as 1 sample in [206].

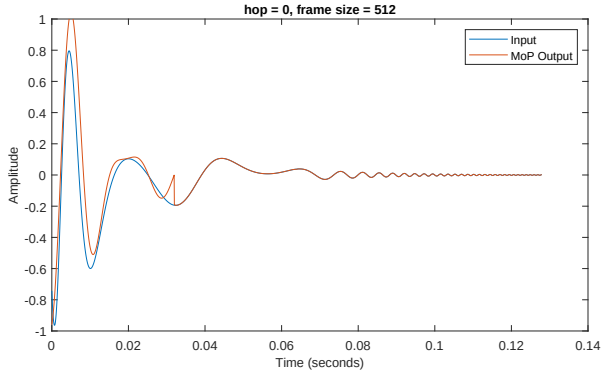
While MoP is an overcomplete representation, using far larger frame sizes and no overlap between frames; the Reassignment and EDS method are in this case restricted to only one sinusoidal partial. The smaller frame and hop size is therefore crucial for these methods to return accurate results with this restriction. eaQHM is restricted to the number of fundamental frequencies provided as an input to this implementation. Increasing the number of partials allowed in eaQHM (without providing the frequency estimates), Reassignment or the EDS implementations resulted in errors.



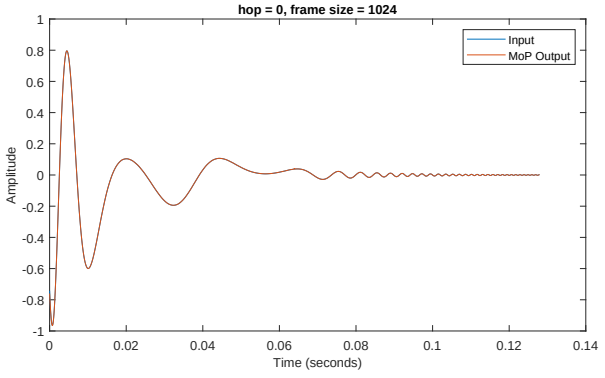
(A) Constant Amplitude Cubic Phase (CA-C3P) MoP frame=512, max partials = 64



(B) Constant Amplitude Cubic Phase (CA-C3P) MoP frame=1024, max partials = 128



(C) Exponential Amplitude Cubic Phase (EA-C3P) MoP frame=512, max partials = 64



(D) Exponential Amplitude Cubic Phase (EA-C3P) MoP frame=1024, max partials = 128

FIGURE 4.59: Plots demonstrating the effects of changes to frame size and maximum number of partials from (513 and 64) to (1024 and 128) on two of the tested signals

EDSM uses a square window for analysis and a Hamming window for overlap-add (OLA) re-synthesis. Reassignment uses Hamming windows for both analysis and OLA re-synthesis, eaQHM uses a Hamming window for analysis and re-synthesis directly from C.21. Sections A.2.2 and A.2.15 contain the results from a range of tests with varying frame and hop sizes in which the resulting input and output signals are displayed along with tables containing the SRER and MIPS results. The following observations are apparent when examining the results of the SRER table A.2.2.

eaQHM performs incredibly well with the Constant Amplitude Linear Phase (CA-LP) signal with most SRER results around 290 dB. The decrease in accuracy resulting from MoP appears to be an incorrect estimate resulting in the initial value of the sinusoid starting at 0.003 instead of at 0. Using zero-phase padding as presented in Section 6.2.1 would be more appropriate for this simple signal where no change in amplitude would display a flat phase response across the sinusoidal peak, indicating no amplitude change. The resulting signal's performance would then only be biased by the methods used in evaluating estimates of amplitude, frequency and phase which vary between different methods such as QIFFT and other interpolation techniques.

The signals returned from RSM and EDSM both start with sample values aligned with the results from eaQHM. However, there are resulting errors at the end of the frame which skew the results. Measuring the SRER for the signal truncated by a single frame length greatly improves these measurements for some of the test signals; this is especially the case for the CA-LP metrics.

In [142] the poor SRER measurements for RSM are possibly attributed to the implementation used by the DESAM toolbox rather than the method itself. It is unclear if this is true as the results for the entire signal excluding the final frame are very accurate. The poor results measured appear to result from the incorrect re-synthesis of the final frame, suggesting that the framework used to run these tests should take this into account, possibly padding the signal by a frame length with zeros, and then truncating the results before measuring the SRER. Recalculating the SRER metrics using the the original and re-synthesised signal with the last frame truncated, results in much better metrics for RSM and EDSM, as shown in Table 4.1. Not all signals are affected by this. For RSM there are significant improvements to the SRER metric for CA-LP, C3A-C3P and EA-QP, and slight improvements regarding CA-C3P, LA-C3P and SA-SP. For EDSM there were significant improvements for most signals: CA-LP, EA-LP, CA-C3P, LA-C3P, C3A-C3P and EA-QP, while EA-C3P and SA-SP showed only small improvements over the original results.

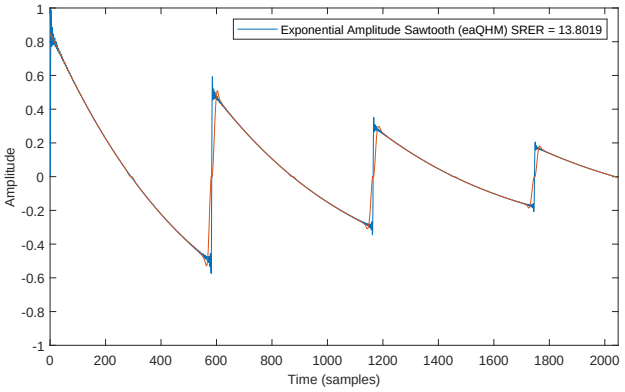
It was previously stated that eaQHM produced the best overall result with a mean overall test signals of 94.96 dB, With the newly measured results which aim to reduce the bias of the error in the resulting final frame of the re-synthesised signal, EDSM has the best performance, with a mean of 127.6 dB over all test signals.

EDSM is the only other method besides MoP which is able to accurately model the EA-NM and LA-NM signals which are composed of an attack and a release where E denotes Exponential and L denotes Linear amplitude change. It is unclear why eaQM and RSM are unable to model these signals

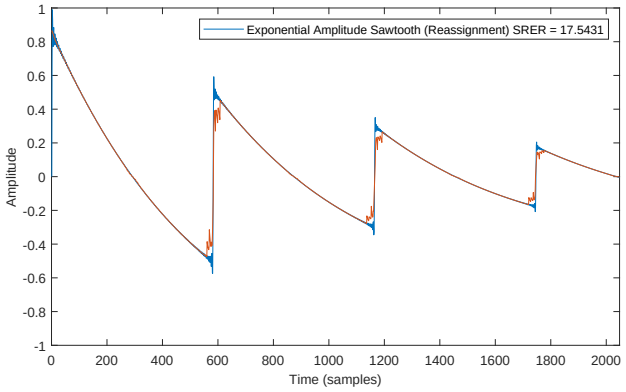
using such small hop and frame sizes of 8 and 32 samples respectively, sometimes performing better with larger hop and frame sizes as can be seen in the plots in Section A.2.13. In general, smaller hop and frame sizes improve the accuracy of eaQHM, RSM and EDSM with the exception of a couple of signals.

Signal	eaQHM original	RSM original	RSM truncated	EDS original	EDS truncated
CA-LP	290.58	31.83	64.43	37.21	290.65
EA-LP	97.13	61.06	61.07	81.75	286.03
CA-C3P	118.01	32.17	44.06	37.09	100.83
EA-C3P	96.87	41.42	41.42	81.61	103.72
LA-C3P	43.72	36.06	41.95	42.31	106.19
C3A-C3P	101.97	23.91	47.10	28.19	95.40
SA-SP	77.05	34.80	38.97	41.21	57.21
ESA-SP	73.31	35.81	35.81	56.26	56.30
EA-QP	138.35	27.77	83.67	34.72	219.36
EA-NM	5.61	6.44	6.44	39.09	39.09
LA-NM	1.95	6.185	6.18	48.83	48.83

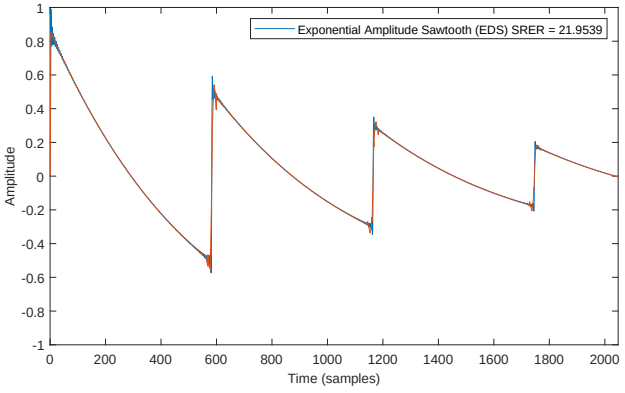
TABLE 4.1: Comparison of SRER metric recalculated with truncated output (truncated by one frame) for RSM and EDSM using best performing setting with Hop of 8 samples and Frame sizes of 32 samples



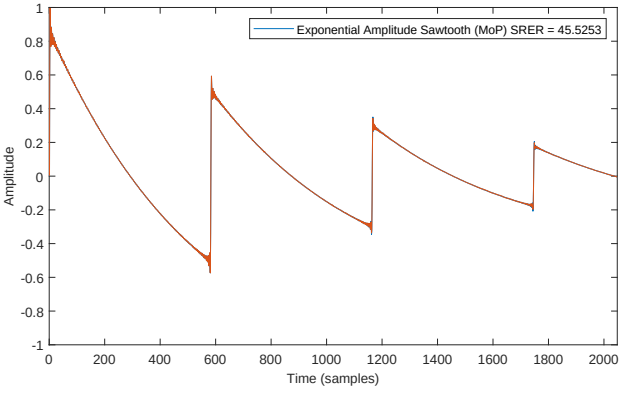
(A) Sawtooth modelled using eaQHM



(B) Sawtooth modelled using Reassignment



(C) Sawtooth modelled using EDS



(D) Sawtooth modelled using MoP with 128 partials

FIGURE 4.60: Outputs of different models for Sawtooth Input synthesised using 128 partials

4.5.10 Tests on released EDM Tracks

eaQHM performs well with percussive instruments and and other synthesised sounds [202, 207]. This is an iterative procedure which reiterates over the entire signal multiple times updating sinusoidal partials non-stationary amplitude and frequency estimates to achieve an optimal solution.

A Python script provided by George Kafentzis and Panagiotis Antivasis [208] has been adapted for testing eaQHM on fully produced EDM songs. The default values of the Python script use a frame size of 32 samples and a hop size of 15 samples. The script is targeted at speech signals with a small number of partials, but has been adapted to include up to 256 partials, and the frequency range extended, ranging from 50 Hz to 15 kHz. A number of EDM tracks have been analysed using the eaQHM python scrip which returned reasonable resulting re-synthesised waveforms. However, these re-synthesised waveforms do contain audible artifacts, and the time taken to run the scrips on excepts from the full songs, which are mostly under 30 seconds long, took a very long time. The average time taken to analyse and re-synthesise each of the 5 short 30 second snippets was 13 hours and 40 minutes on an Intel i7-5930K (CPU @ 3.50GHz, 64 Gig RAM) desktop PC.

MoP is able to re-synthesise the audio samples with a very high degree of accuracy, absent of glitches at frame boundaries even though phase and amplitude coherence across frames is not applied. However, this representation uses multiple sinusoidal atoms to model non-monotonic amplitude and frequency changes occurring over a spectral peak. An over-complete sinusoidal representation is able to capture overlapping components within the frequency domain. The potential of grouping these atoms into their separate individual components requires further investigation..

The following Sections 4.5.10.1 and 4.5.10.2 present the modelling of these short audio samples for eaQHM and MoP. More detailed plots and details of other tests are presented in Appendices B B.2.

4.5.10.1 eaQHM Results

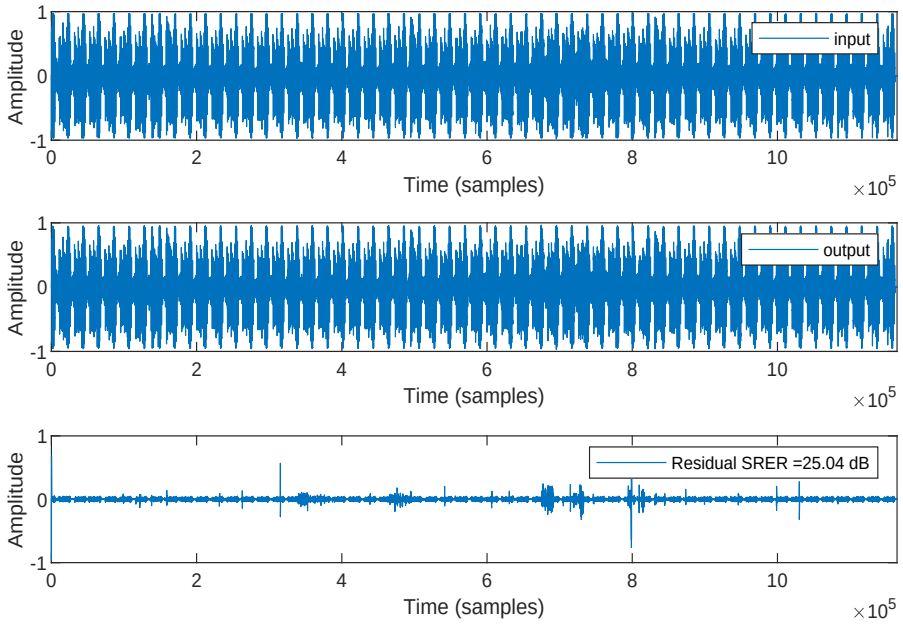


FIGURE 4.61: eaQHM Results of Lumen - Gruntled (@48 kHz)

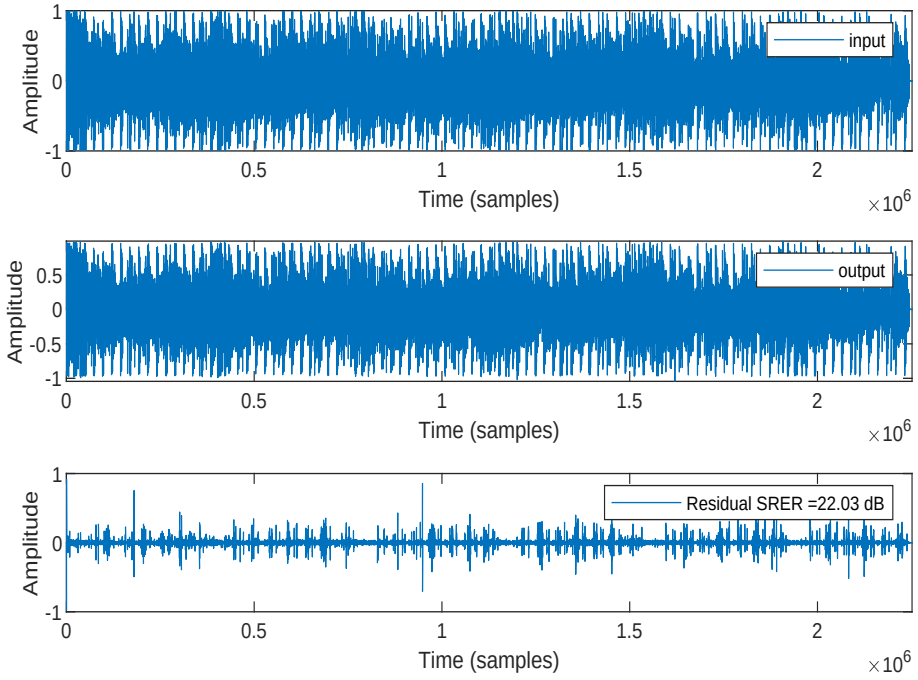


FIGURE 4.62: eaQHM Results of Pippi Ciez - Baohum (@48 kHz)

Figures 4.61 and Figures 4.62 display the Input, Output and Residual signals from a eqQHM analysis/synthesis using the adapted python script from [208].

The SRER ranges from a worst case of 22 to 25 dB. The re-synthesised waveforms unfortunately contain audible artifacts, and the time taken to run the scrips on the 30 second long audio snippets from fully produced EDM tracks, which contains many polyphonic complex components, took a very long time.

4.5.10.2 MoP Results

Figures 4.63 and 4.64 display the Input, Output and Residual signals from an MoP decomposition. The residual signals contain very low level of energy, as most of the signal components; including complex polyphonic sounds, have been modelled accurately by the overcomplete representation. The average SRER was 40 and 44 dB with 256 components specified as an exit criteria. The over-complete MoP decomposition allowing multiple atoms for sinusoidal peaks, which does not reply on phase or amplitude continuation between partials, shows an improvement compared to the results presented above using eaQHM.

The average time to execute the MoP decomposition in Matlab [16] without any parallelization was between 180 and 410 seconds, which is still between 10 and 20 seconds of analysis/synthesis time per 1 second of audio data. Matlab is a scripting language, and used here for easy of use and visualisatrion of data when protopying the algorithms used in this thesis. Matlab is highly efficient at matrix multiplications but performance decreases when using dynamic memory allocation [209]. Porting the scripts to a compile language like C++ would improve this performance. Additionally, MoP is a highly parallelizable process which can be implemented on multi-threaded CPU cores for accelerated computing. Implementation on a GPU could improve this further with recent improvements in GPU pipelining of instruction [210].

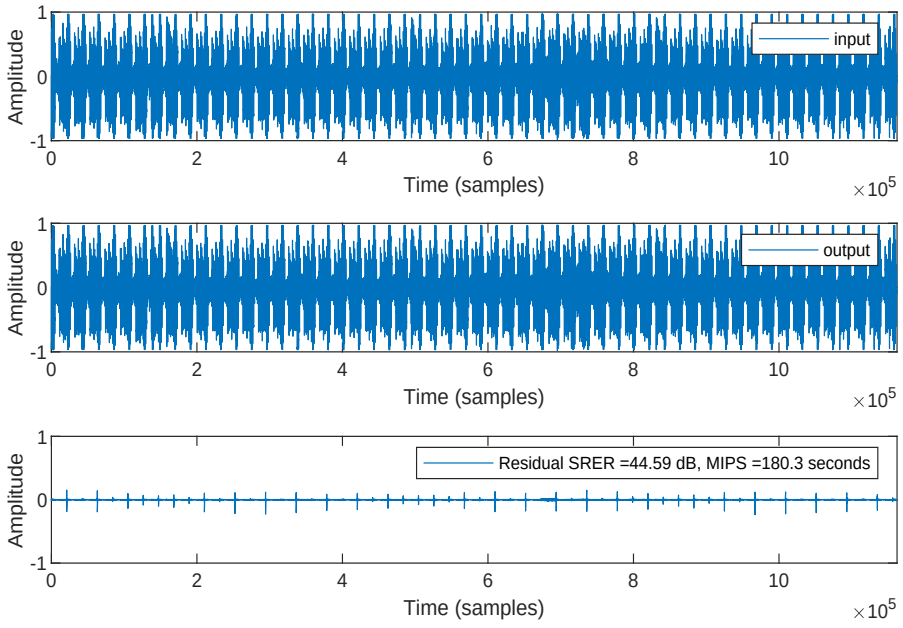


FIGURE 4.63: MoP Results of Lumen - Gruntled (@48 kHz)

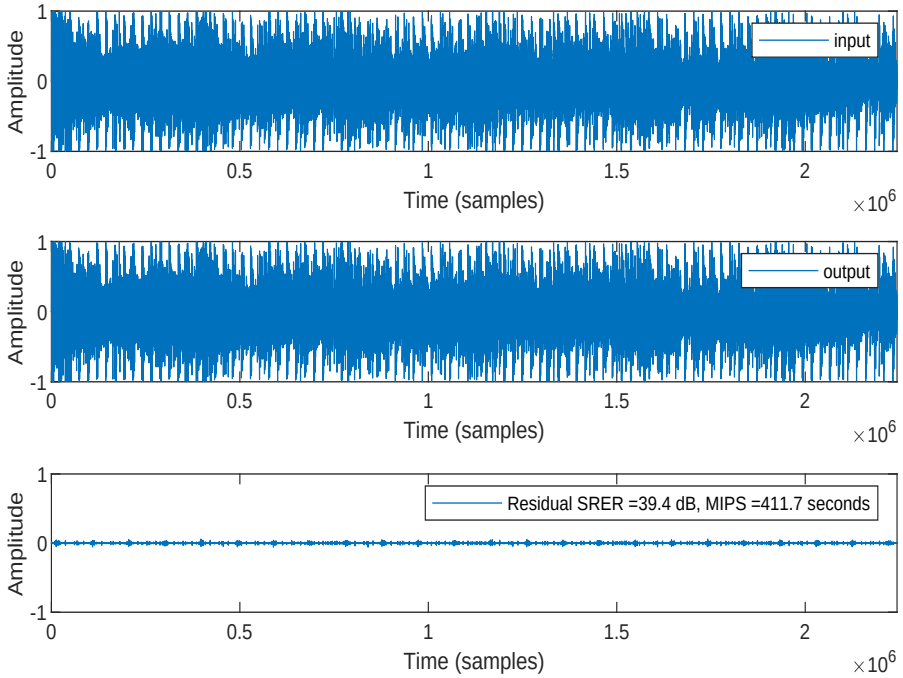


FIGURE 4.64: MoP Results of Pippi Ciez - Baohum (@48 kHz)

4.5.11 Modelling transient components using Modelled Pursuit

Section 2.6 in Chapter 2 provides an overview of transients and transient modelling. Any sudden change resulting in a significant increase in energy within a signal can be regarded as a transient component. In the context of kick and bass, this can be defined as a percussive element's sudden onset where there is a short burst and rise in energy followed by some decay (generally) over a longer period of time.

Spectral models have been extended in the past to include Transient modelling and synthesis such as in [101, 107, 211] where a flexible analysis/synthesis model for transient signals was proposed that effectively extended the Spectral Modeling Synthesis (SMS) parameterization of signals from sinusoids and noise to sinusoids, transients and noise. The explicit handling of transients provided a more realistic and robust signal model. These models reply on a transient detection stage, responsible for identifying note onsets [119–122, 124, 129, 130, 212–214]. Transients are usually detected and removed from the input signal prior to the sinusoidal and residual modelling stages. However, extraction of sinusoidal components leaving transients to be modelled and extracted from the residual signal is not an uncommon approach.

In a fixed-size single-frame non-overlapping analysis system the length of the analysis window could potentially be too long for an accurate decomposition of a transient well localised in time. Transient components are not modelled by the sinusoidal analysis stage using this approach, leaving them in the residual signal. The following Chapter 5 presents a method for modelling the residual signal using the undecimated discrete wavelet transform. Transient components are able to be separated from the noise within the residual signal using popular denoising techniques commonly used by this representation. However, investigation into different wavelet types and different threshold levels related to the amount of energy remaining in the residual signal for finding an optimal solution with this method remains as a future research item. Once separated, the localisation of transients in time can be detected, and an appropriate sub-frame containing the transient extracted and decomposed by MoP.

In the following section a number of short percussive sounds (snares, hi-hats) are decomposed and evaluated using MoP using the causal implementation with a rectangular window to model transient components using multiple sinusoidal components in an over-complete decomposition.

Due to the shorter window, which reduces the frequency resolution and the number of broadband components, the decomposition of short percussive sounds using MoP is found to be effective and does not suffer from the issues encountered with non-monotonic signals using an over-complete representation, presented in section 2.8.

Figure 4.65 shows a software synthesizer Kick2 [8] used for generating kick sounds. The note onset quickly decays from 17.197 kHz to 448 Hz in a matter of 48 samples which would cause a large bias and poor modelling from traditional non-stationary sinusoidal methods. However, using an iterative MoP approach with a dictionary of semi-stationary sinusoidal atoms (only change in amplitude is modelled not frequency change) is able to provide an accurate decomposition from a linear combination of these atoms.

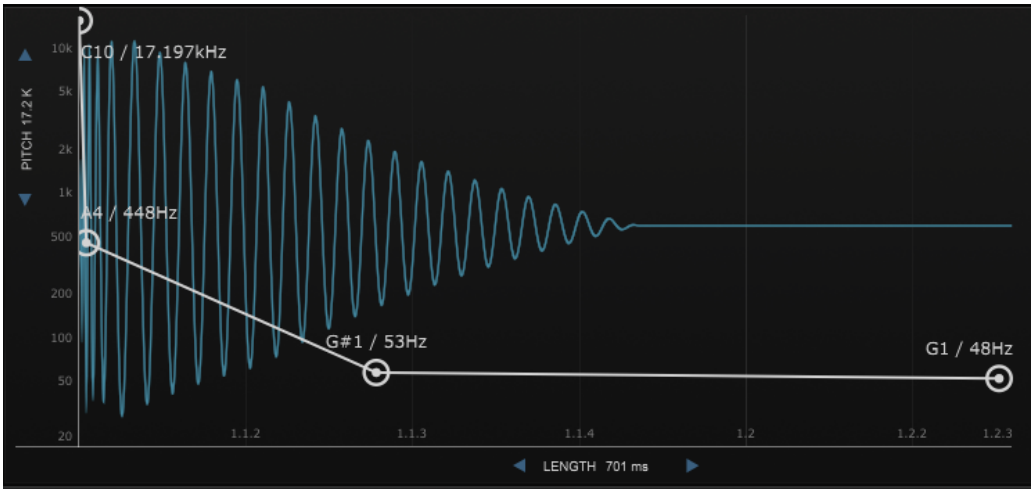


FIGURE 4.65: Kick2 Synthesiser with large change in pitch at onset

The implementation of MoP used in this section for modelling percussive sounds uses a segmented non-overlapping frame based analysis framework, with a rectangular window and parameters estimated using causal measurements. The implementation first performs the analysis using the default setting for MoP which use the less costly direct amplitude estimation method. However, in the case that the iterative MoP method described above using the direct amplitude estimation method cause the energy in the residual signal to increase, then the estimation method is replaced with the more expensive inner product amplitude estimation method where this issue is avoided since residual energy decreases monotonically with MP. A flexible framework whereby the MoP implementation is able to switch between the direct amplitude estimation method and the more costly inner product method when an increase in energy in $\mathbf{R^n s}$ occurs, provides a flexible compromise between computational complexity and accuracy.

4.5.11.1 Examining atomic decomposition of transients components using MoP

One of the research objectives outlined within this thesis was to examine modelling short transient signals with non-stationary sinusoidal bases. Transients are described in detail in Chapters 2 and 5 but are essentially short-term broad-band components. Transient / Steady-State (TSS) separation is commonly applied in a signal model allowing these components to be modelled separately.

An over-complete decomposition of percussive sounds using MoP is able to model and re-synthesise these signals accuracy with an average SRER of over 40 dB . Snare drums, Hi-Hats and other percussive sounds are in general short signals containing broad band components. The presence of a large number of frequency components, and often elements of noise enables MoP to decompose these short signals into an atomic decomposition which able to perform time and pitch scale modifications in a meaningful and expected way.

Kick Drum Example:

Figure 4.66 displays a simple kick drum. The first 512 samples are displayed in Figure 4.67. The transient region is roughly around 200 samples long before there is a transition into a more sinusoidal; deterministic / steady sate; stage of the kick drum. A frame of the first 256 samples was chosen as this is the closest number of samples to that transition point from which the Fast Fourier Transform can be computed. Figure 4.68 compares the input signal with the output of a 64 atom decomposition using causal MoP. The output of the MoP decomposition is very close to the input signal. Figure 4.69 displays the 16 most prominent atoms from the decomposition. This shows diverse range of components contributing to the transient signal which is expected to consist of short-term broadband components.

Figure 4.70 displays the pitch shifted transient, while Figure 4.71 displays the transient component, time stretched by a factor of 0.5 and 2, and re-synthesised with the remaining unmodified signal from MoP parameter estimates.

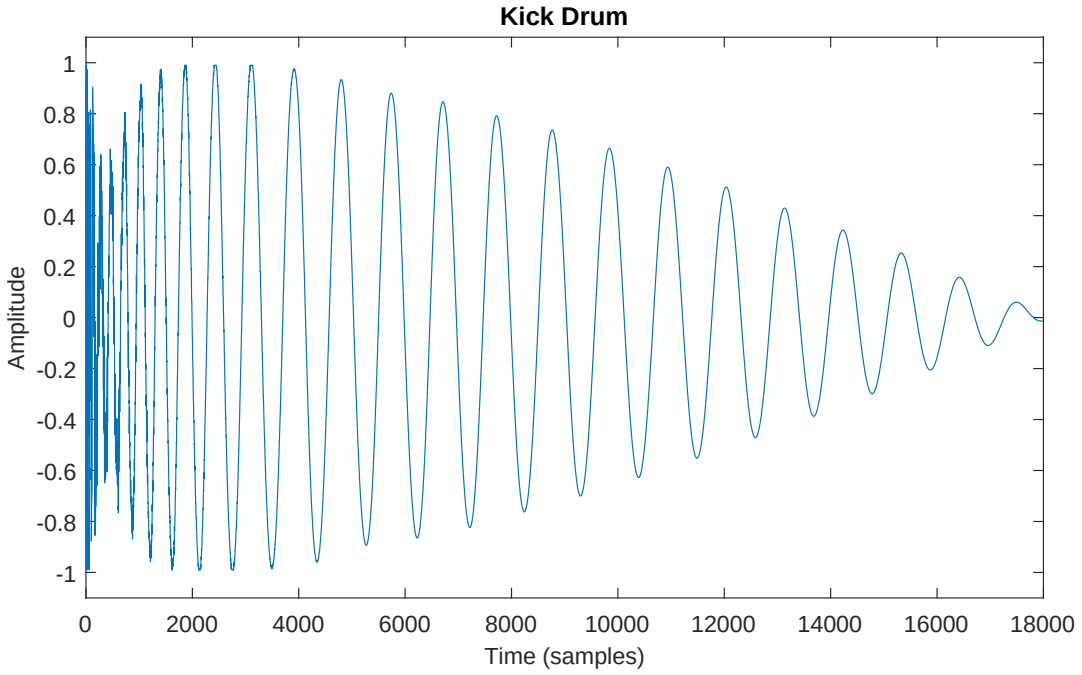


FIGURE 4.66: Kick Drum (48 kH) Transient

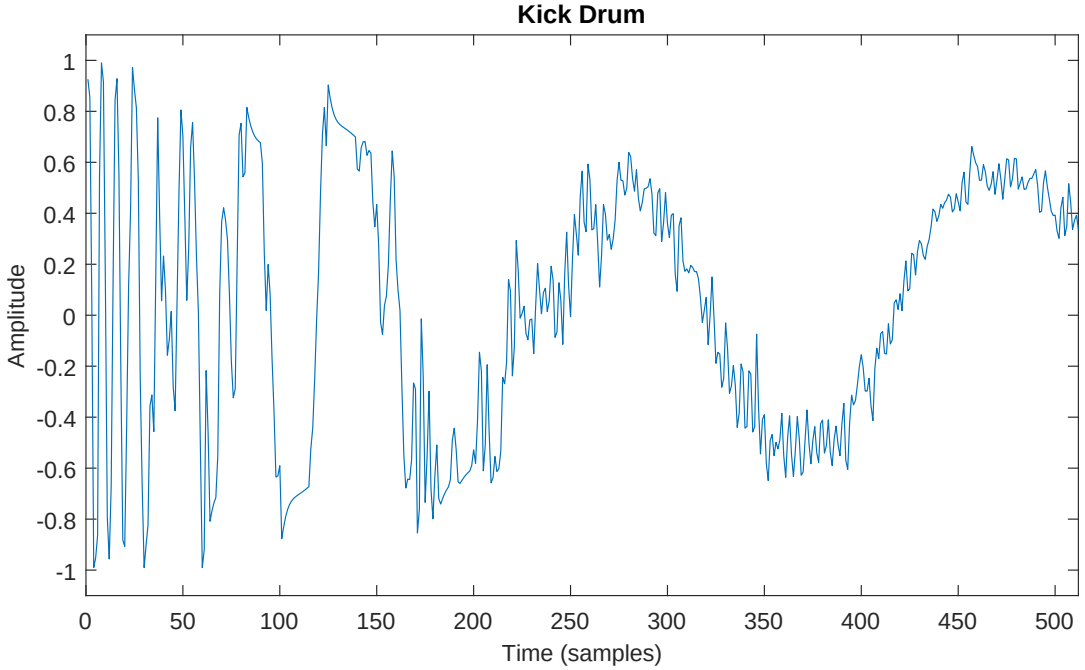


FIGURE 4.67: Kick Drum Transient (512 samples)

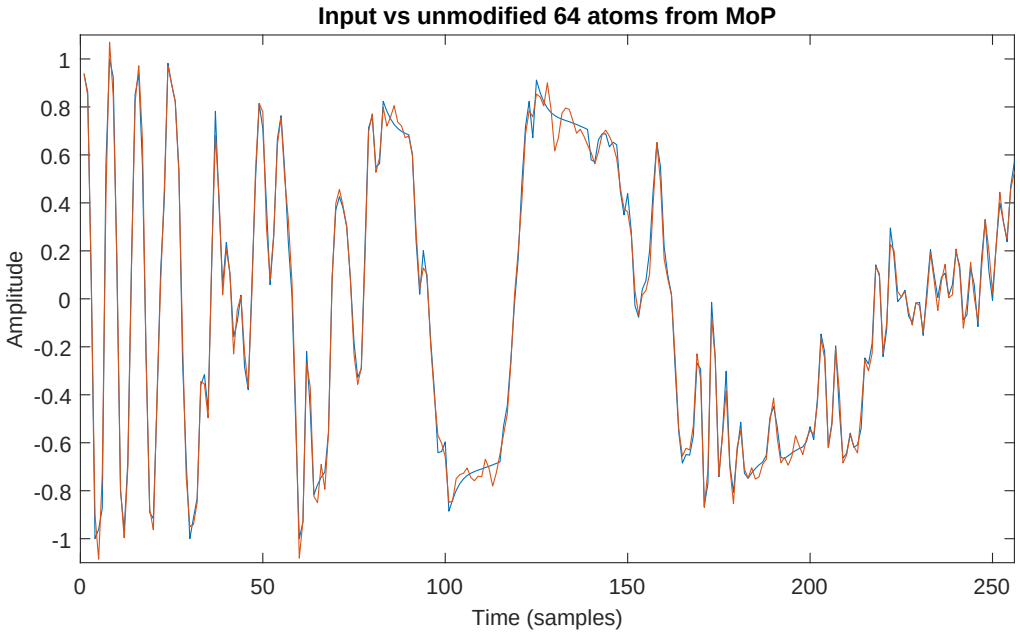


FIGURE 4.68: Transient from 4.67 compared to output of 64 MoP Atoms

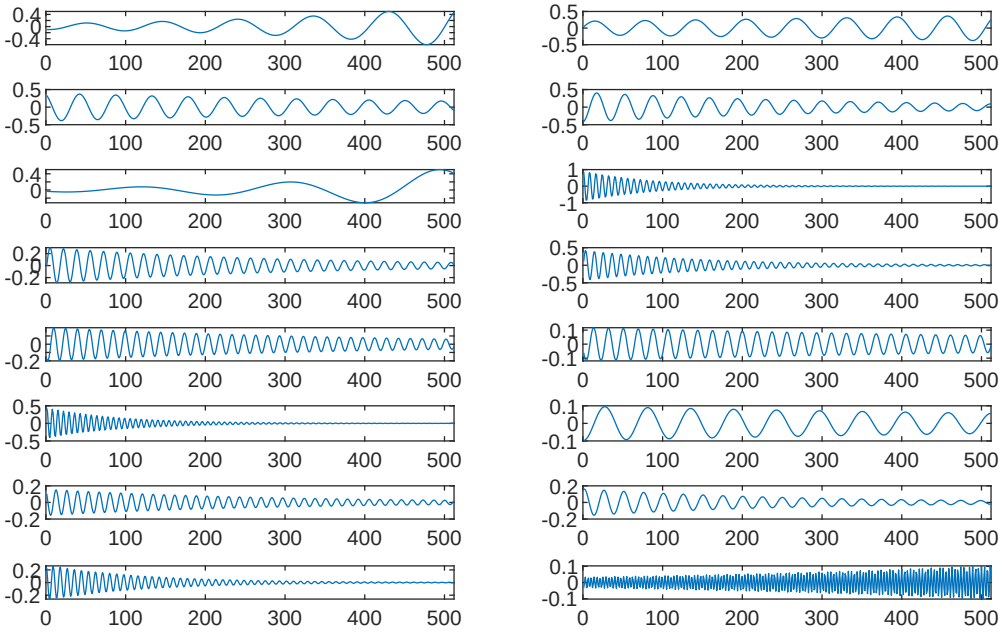


FIGURE 4.69: 16 of the atoms from MoP which have the most influence

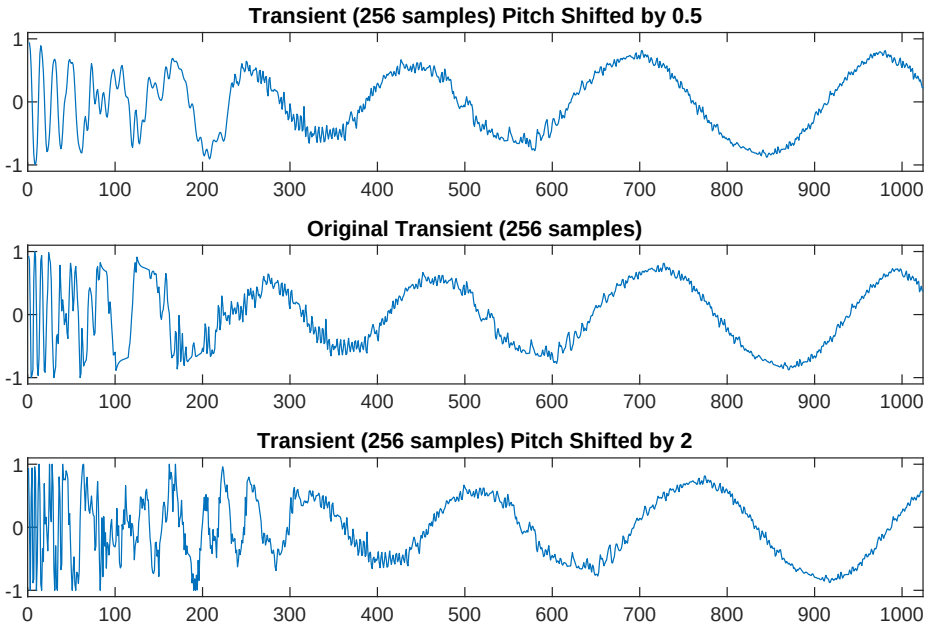


FIGURE 4.70: Kick Drum Transient Component of 256 samples from 4.67 pitch shifted by 0.5 and 2

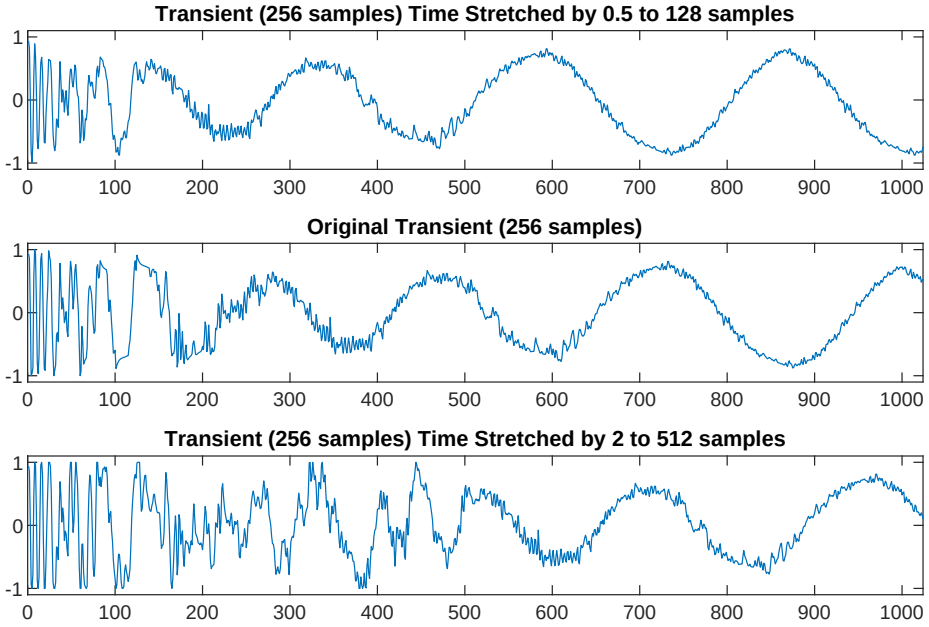


FIGURE 4.71: Kick Drum Transient Component of 256 samples from 4.67 time stretched by a .5 and 2 xlim(0 1024)

Open Hi Hat Example Pitch Shifted:

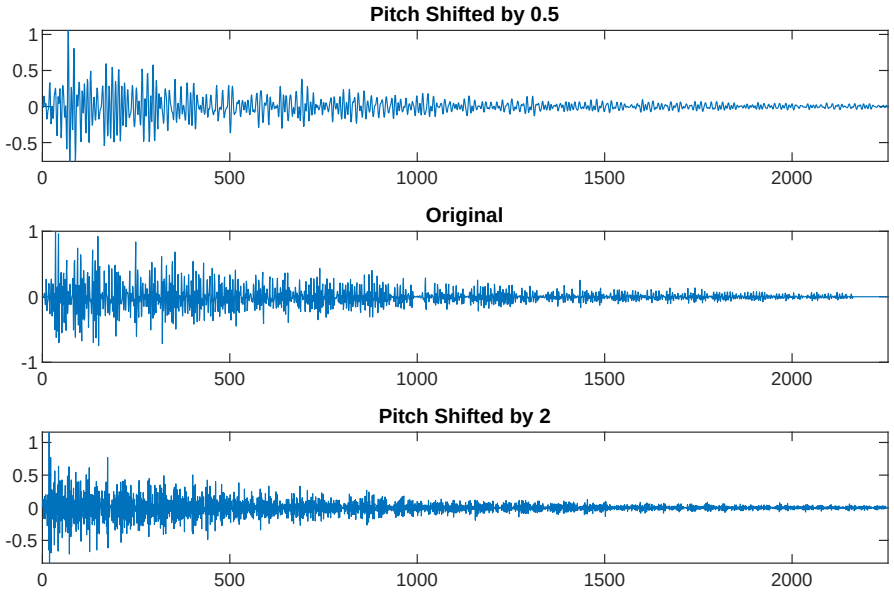


FIGURE 4.72: Hi Hat Pitch Shifted (@48 kHz)

Snare Examples Pitch Shifted:

A)

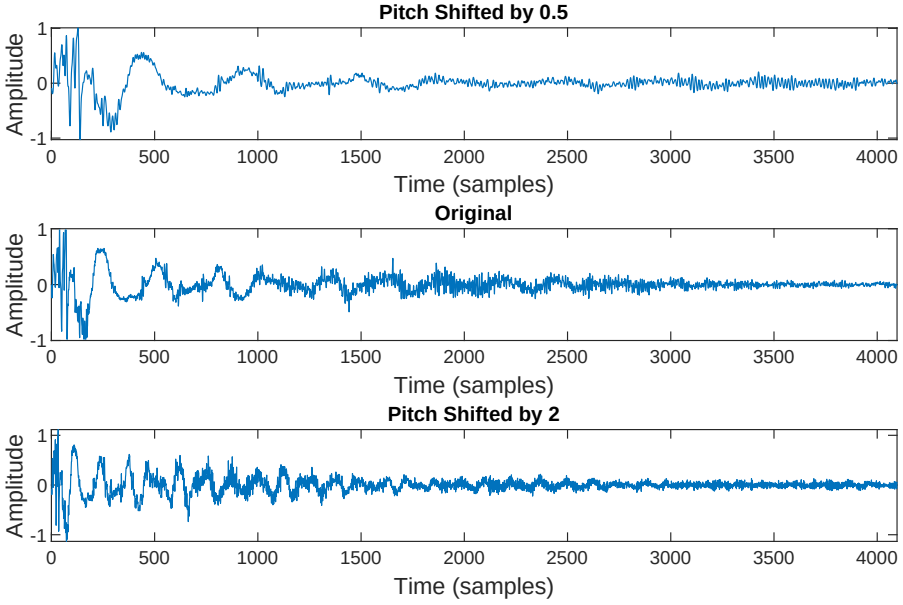


FIGURE 4.73: Snare (@48 kHz) Example Pitch Shifted

B)

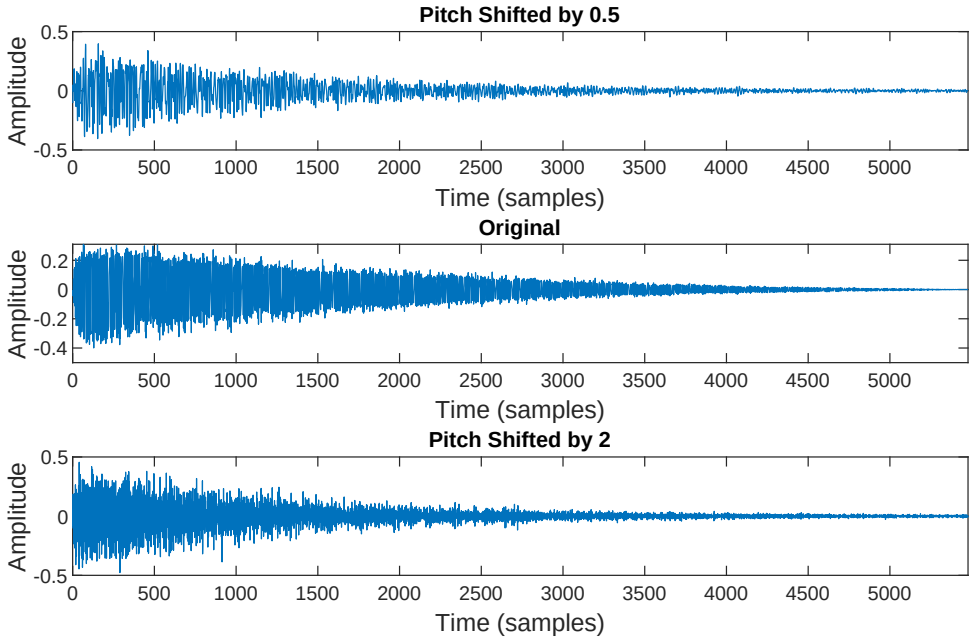


FIGURE 4.74: Snare (@48 kHz) Example Pitch Shifted

C)

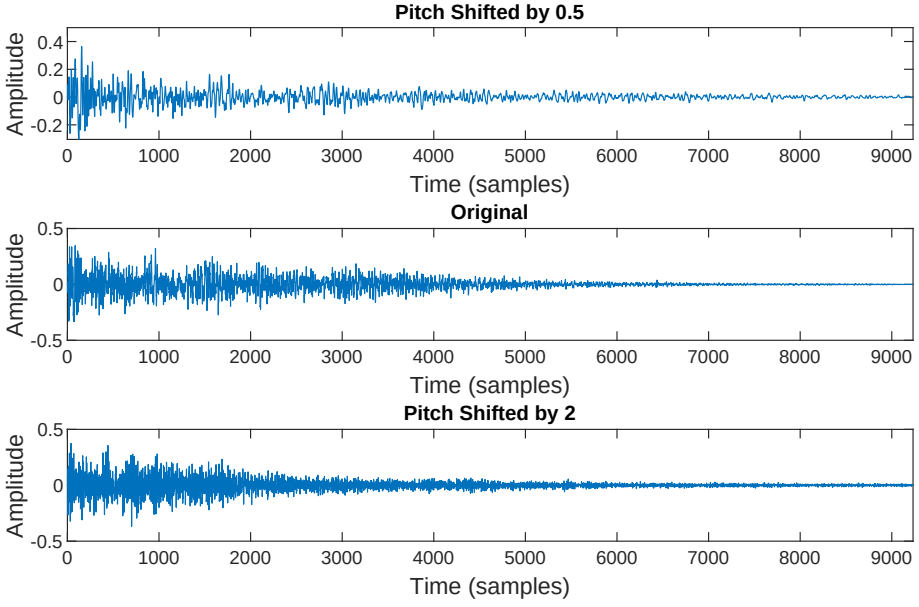


FIGURE 4.75: Snare (@48 kHz) Example Pitch Shifted

Open Hi Hat Example Time Stretched:

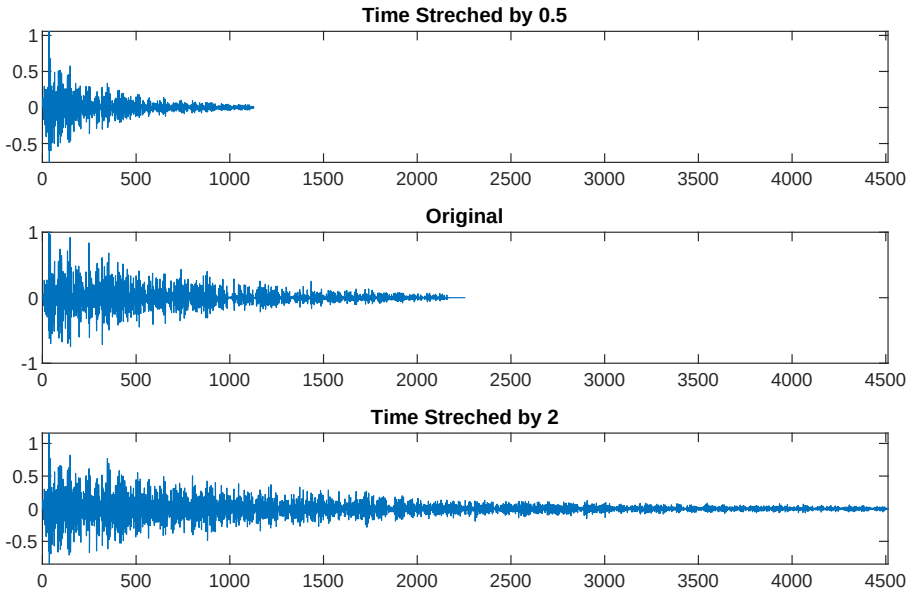


FIGURE 4.76: Open Hi Hat (@48 kHz) Example Time Stretched

Snare Examples Time Stretched:

A)

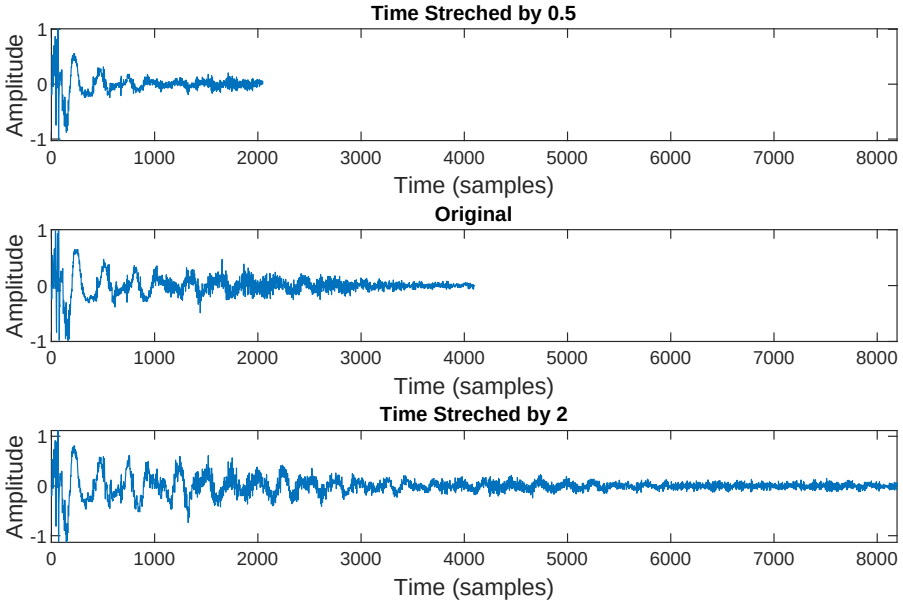


FIGURE 4.77: Snare (@48 kHz) Example Time Stretched

B)

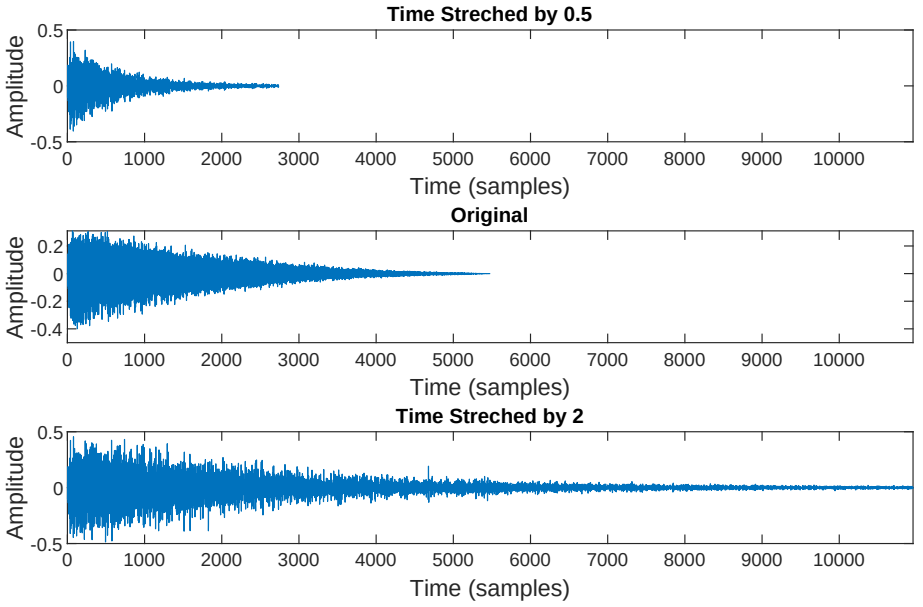


FIGURE 4.78: Snare (@48 kHz) Example Time Stretched

C)

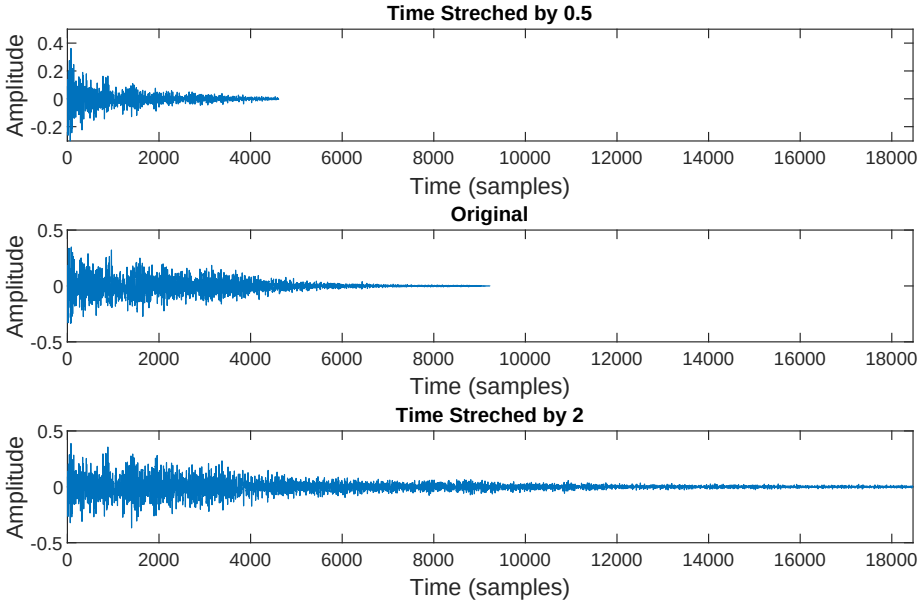


FIGURE 4.79: Snare (@48 kHz) Example Time Stretched

The audio files for the above examples are available in the list of Accompanying Material in E. The modelling of percussive using matching pursuit performs highly effectively. It ha bee shown that the original signal can be reconstructed to a very high degree of accuracy. Pitch shifting and Time Stretching have been shown to work well with short broadband components using smaller windows localised around the percussive sound such as a hi-hat or snare. Interesting sound have been demonstrated from the selection or reconstruction of a limited set of atoms, creating new timbers / sounds.

The following section investigates modelling both transient and simi-stationary sinusoidal components from single frame estimates. MoP is able to accurately model and reconstruct the sound even though the basis functions with extend the length of the analysis window may be longer than the percussive or transient sections contained within the frame. The limitation a fixed basis length leads to non-monotonic components being decomposed into multiple atoms with similar frequencies but wit varying amplitudes, phase and monotonic estimates of amplitude an frequency change. This leads to errors in the resynthesised signal when performing time and pitch scale modifications. Extending the dictionary of atoms of various lengths and localizing them at the correct time within a frame would result in more accurate decomposition where spectral elements do not span the entire analysis space. The following section describes this in detail.

4.6 Conclusion

The atomic decomposition method of MoP has been introduced and adapted for use in a non-overlapping frame by frame analysis system. Two different implementations using causal and non-causal measurements have been presented. Change in frequency and the effect this has on first and second order phase difference measurements have been discussed with possible solutions for incorporating frequency change estimates from the second order derivative of the phase, and methods for correcting the amplitude estimates from the effect this has on first order phase difference measurements.

MoP has been shown to capture frequency change as multiple semi-stationary (only amplitude change, no frequency change) components. moP is able to decompose and re-synthesise complicated signals containing complex amplitude and frequency changes, including components which overlap in the frequency domain.

EDM dance music has also been captured in an over-complete single frame representation which does not rely on data from previous frames. The discontinuities at frame boundaries are below -40 dB.

Non-monotonic amplitude change is shown to be problematic when time and pitch shifting transformations applied, due to the change in constructive and destructive interference the atoms have on the re-synthesised sound. Percussive sounds with short broadband components which have shorter attach times show promising results for time and pitch scale modifications from a causal MoP implementation using a rectangular window for a single analysis frame encompassing the entire sound.

The following Chapter 5 presents a novel method for handling the residual signal after an MoP atomic decomposition.

Chapter 5

Residual Modelling

"In the Wave Lies the Secret of Creation"

Walter Russell (1931) [215]

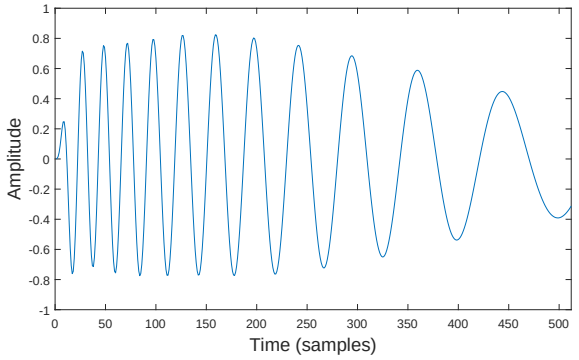
5.1 Introduction

Signal models assume an underlying structure of the signal being modelled. Sinusoidal models capture harmonic and periodic functions and so aim to model the oscillatory waveform's embedded within an audio signal. Source filter models represent speech as a combination of periodic and noise sources for the generation of the vocal chord vibration, air turbulence and frication (noise generated from the glottis) respectively, while resonant all-pole filters model the vocal tract [216].

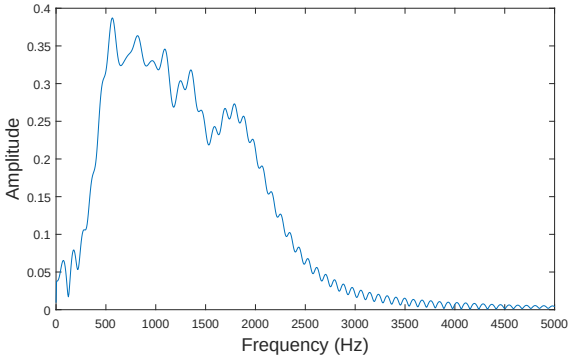
The sinusoidal model used in this thesis to analyse and re-synthesise an audio signal decomposes the original signal into its non-stationary sinusoidal components. The single frame modelled pursuit approach employed performs this iteratively, where the sinusoidal peaks with the highest magnitudes are selected, synthesised using the parameters estimated from the methods discussed in Chapter 3, and subtracted from the original signal. This leaves an altered signal to be analysed again which has been reduced in energy by the removal of the modeled components. This process continues until certain criteria are met as discussed in Section 4.3. What is left at the end of this process is the difference between the original signal and the signal synthesised from the parameters of all the modelled sinusoidal components. This remaining signal is termed the 'residual' in spectral modelling systems,

also referred to as the stochastic component [71]. The quasi-stationary parts of the audio signal are considered as long-term narrow-band components because they are modelled from sine and cosine basis functions which are non-finite, and in general are modelled with small frequency changes which do not span a large range within the frequency spectrum. However, the residual part of the signal is "comprised of both long- and short-term broad-band components." [217].

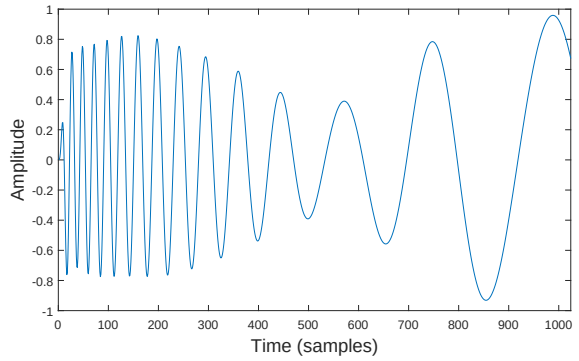
Long-term broadband elements encompass other longer lasting noisy components not well modelled using sinusoidal basis functions and can include "such musical phenomena as flute breath noise or violin bow noise. Synthesis without such "noise" tends to sound artificial; it is desirable to improve the synthesis realism by modeling the residual in such a way that it can be re-injected in the synthesized signal." [218]



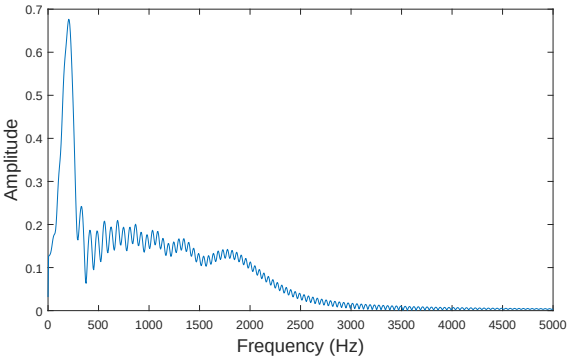
(A) First 512 samples of an example kick drum



(B) Magnitude Response of kick drum in Figure 5.1a



(C) First 1024 samples of an example kick drum



(D) Magnitude Response of kick drum in Figure 5.1c

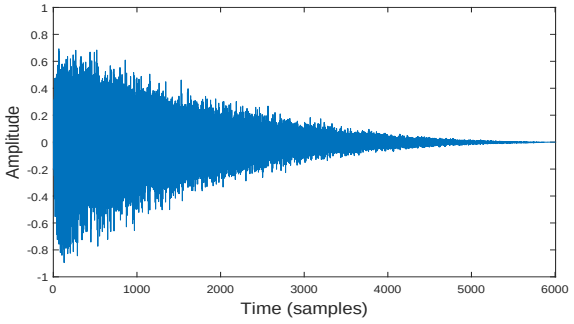
FIGURE 5.1: Example of a kick drum containing a fast attack and part of the more gradual release, and the magnitude response.

Short-term broad band components are classified as transients which are associated with note onsets where a sudden attack causes a short burst of energy manifested as broadband noise in the frequency domain. An example of a kick drum audio sample and the magnitude spectrum from the first 512 and 1024 samples are shown in Figure 5.1.

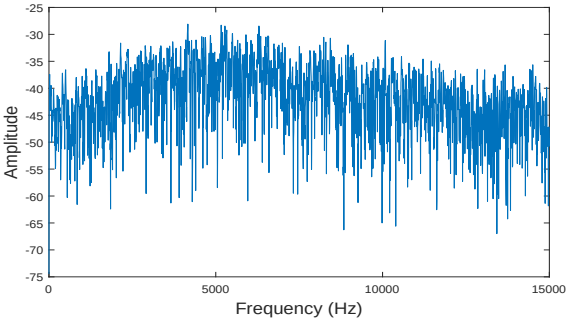
Different music instruments exhibit different transient behaviours. A kick drum is a relatively simple signal, and can be synthesised using a sinusoid with a defined ADSR envelope for its amplitude as well as a frequency ramp. The frequency of a kick drums usually starts at a high frequency with a very fast decay to a lower frequency and then a slower decay to an even lower frequency as is shown in Figure 5.2 where the frequency of the sinusoid starts at C10 (17.9 kHz), suddenly drops o A4 (448 kHz), and then gradually decays and settles to G 1 (53 Hz) at the end of the release part of the sound.

FIGURE 5.2: Kick Drums pitch envelope

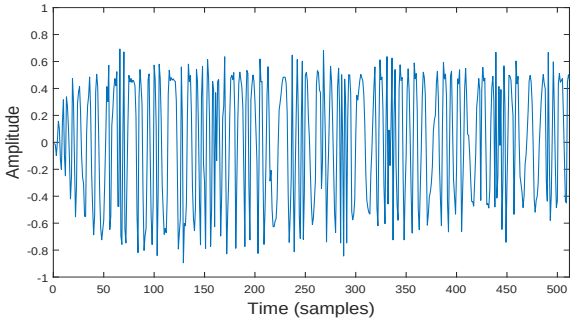
A snare drum for instance is a percussive instrument which in comparison contains far greater broadband noise as shown in Figure 5.3.



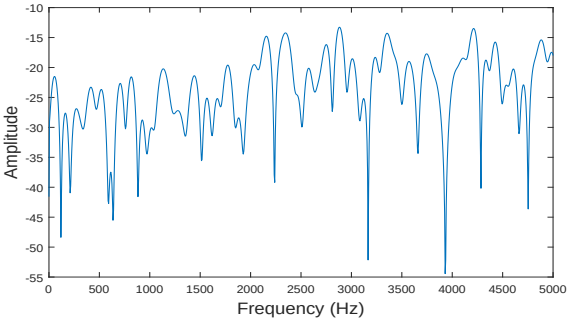
(A) Example of a snare drum



(B) Magnitude Response of snare from Figure 5.3b



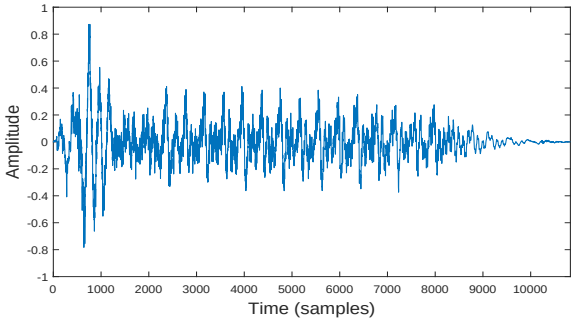
(C) First 1024 samples of an example snare drum



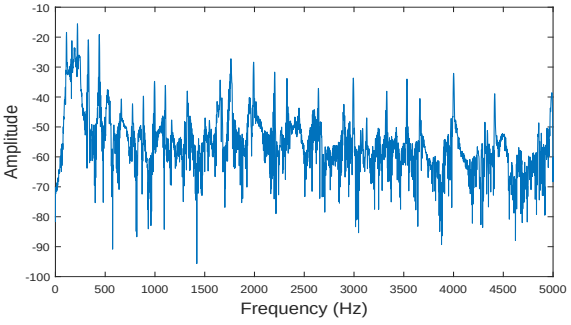
(D) Magnitude Response of snare from Figure 5.3c

FIGURE 5.3: Example of a snare drum (@48 kHz) and the magnitude response.

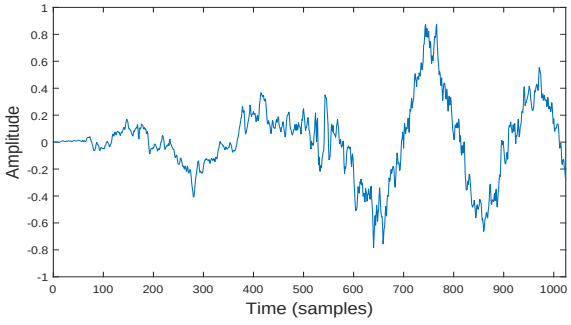
A guitar will also have different characteristics depending on the type of string; the shape and material of the body, and how the string is plucked. Examples of which are shown in Figures 5.4 and 5.5.



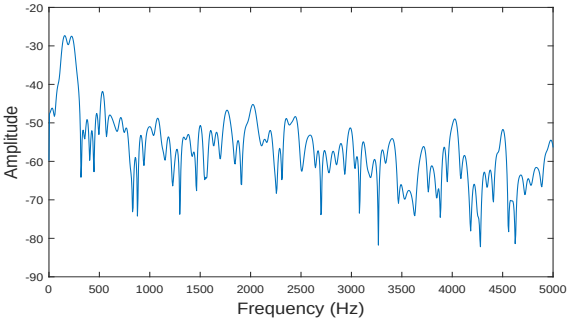
(A) Example of a guitar string plucked



(B) Magnitude Response of guitar from Figure 5.4a

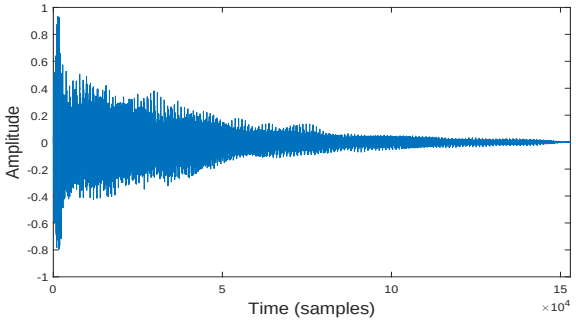


(C) First 1024 samples a plucked string from a guitar

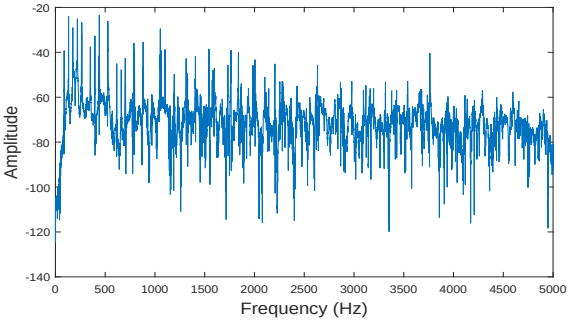


(D) Magnitude Response of guitar from Figure 5.4c

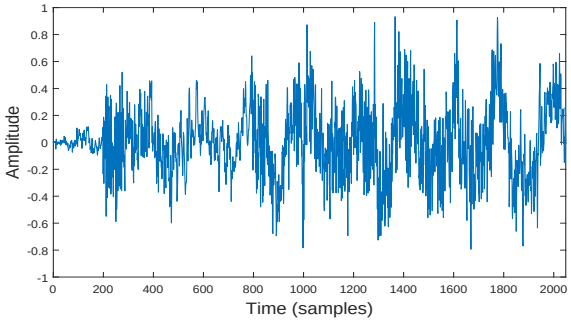
FIGURE 5.4: An example of a plucked string from a guitar (@48 kHz) and the magnitude response



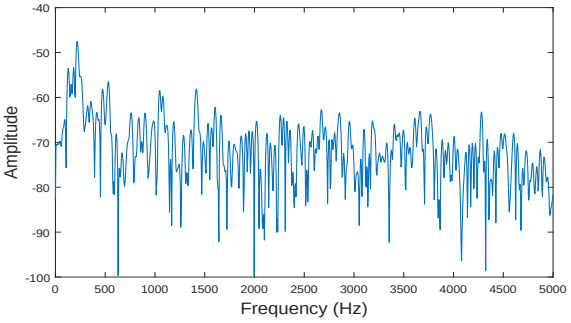
(A) Another example of a guitar string plucked



(B) Magnitude Response of guitar from Figure 5.5a

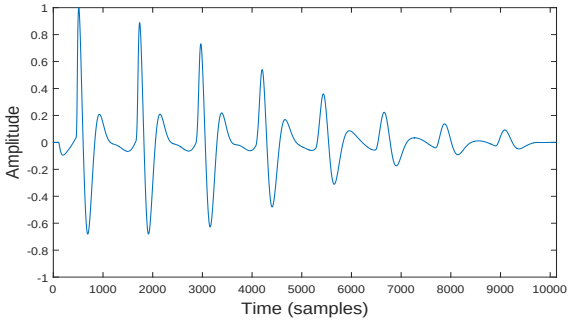


(C) First 2048 samples a plucked string from a guitar

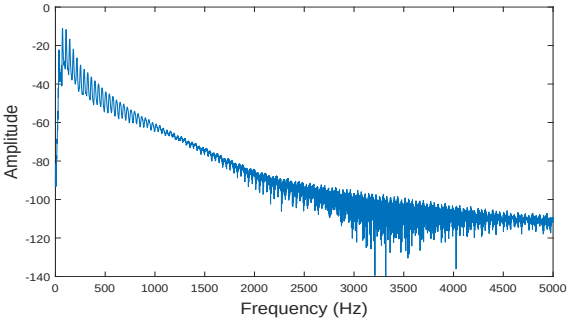


(D) Magnitude Response of guitar from Figure 5.5c

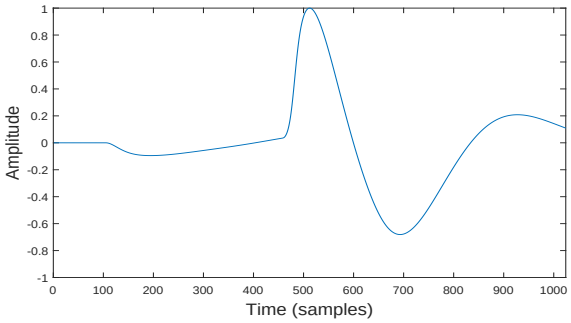
FIGURE 5.5: Second example of a plucked string from a guitar (@48 kHz) and the magnitude response



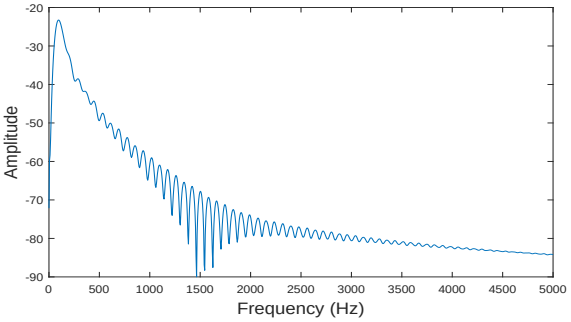
(A) Single Bass note



(B) Magnitude Response of Single Bass note from Figure 5.6a

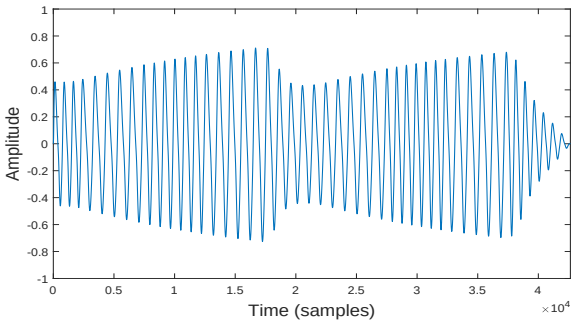


(C) Single Bass note (1024 samples)

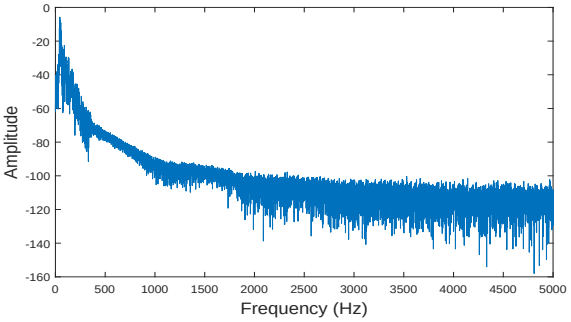


(D) Magnitude Response of a Single Bass note from Figure 5.6c

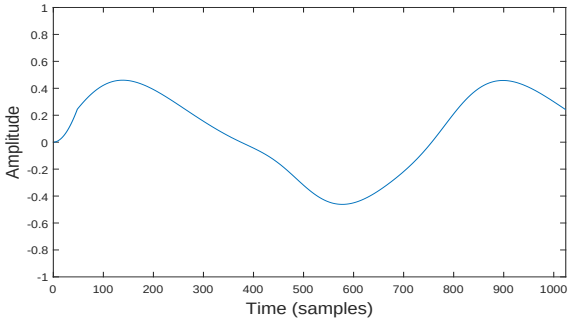
FIGURE 5.6: Single Bass note (@48 kHz) with magnitude spectrum



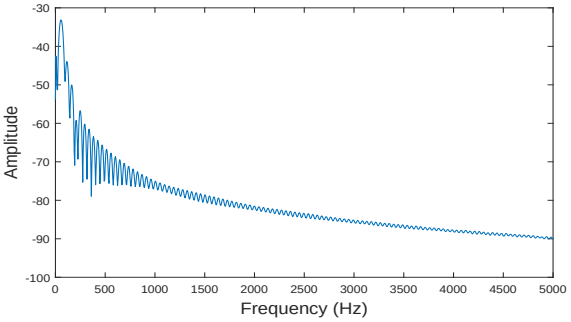
(A) Wobbly Bass Example



(B) Magnitude Response of Wobbly Bass from Figure 5.7a

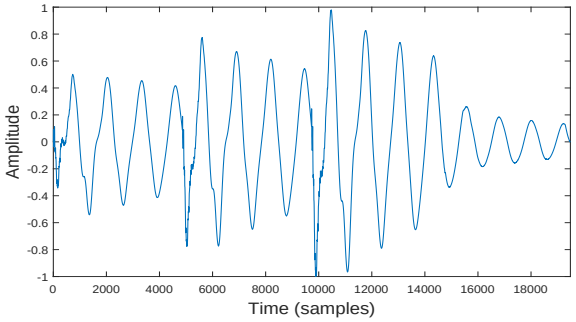


(C) Wobbly Bass (1024 samples)

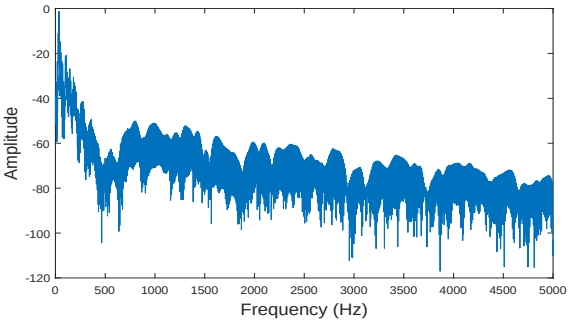


(D) Magnitude Response of Wobbly Bass from Figure 5.7c

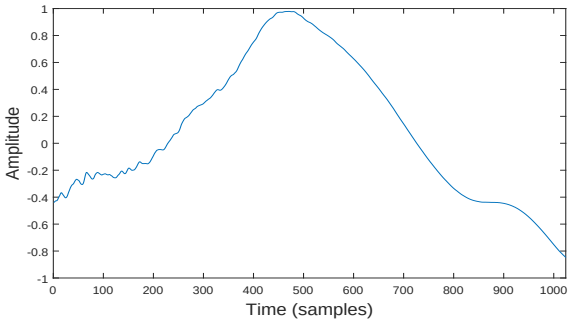
FIGURE 5.7: Wobbly Bass (@48 kHz) example with magnitude spectrum



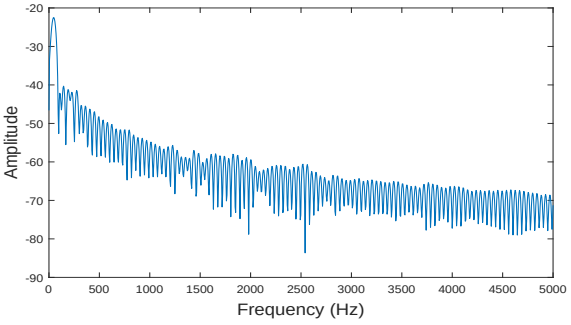
(A) Tight Bass Example



(B) Magnitude Response of Tight Bass Example from Figure 5.8a

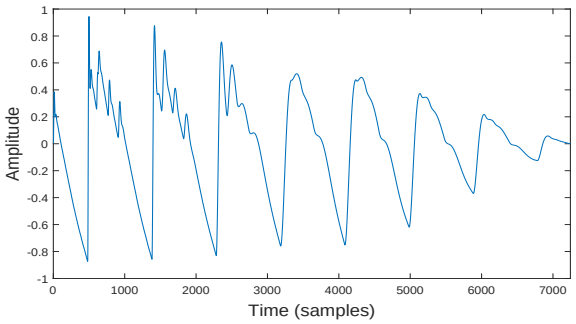


(C) Snippet of Tight Bass Example (1024 samples)

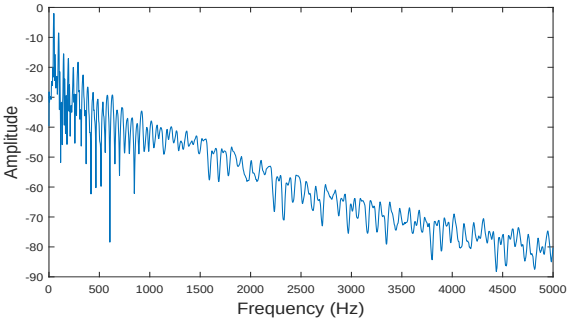


(D) Magnitude Response of Tight Bass from Figure 5.8c

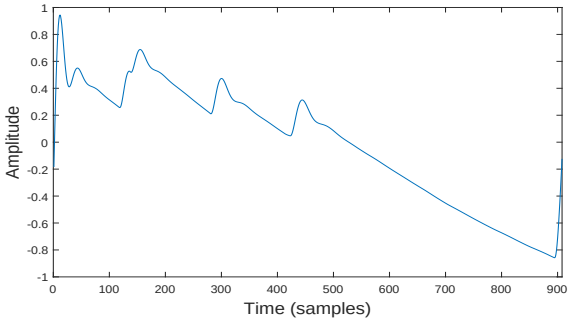
FIGURE 5.8: Tight Bass Example (@48 kHz) with magnitude spectrum



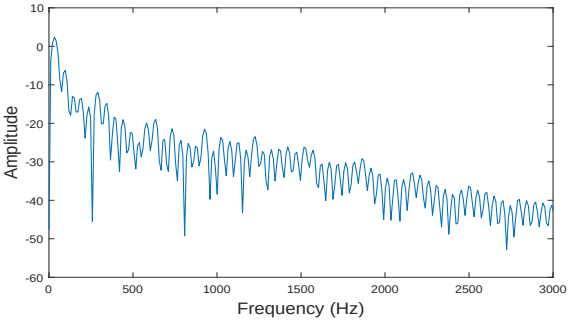
(A) Sawtooth Bass Example



(B) Magnitude Response of Sawtooth Bass from Figure 5.9a



(C) Snippet of Sawtooth Bass Example



(D) Magnitude Response of Sawtooth Bass from Figure 5.9c

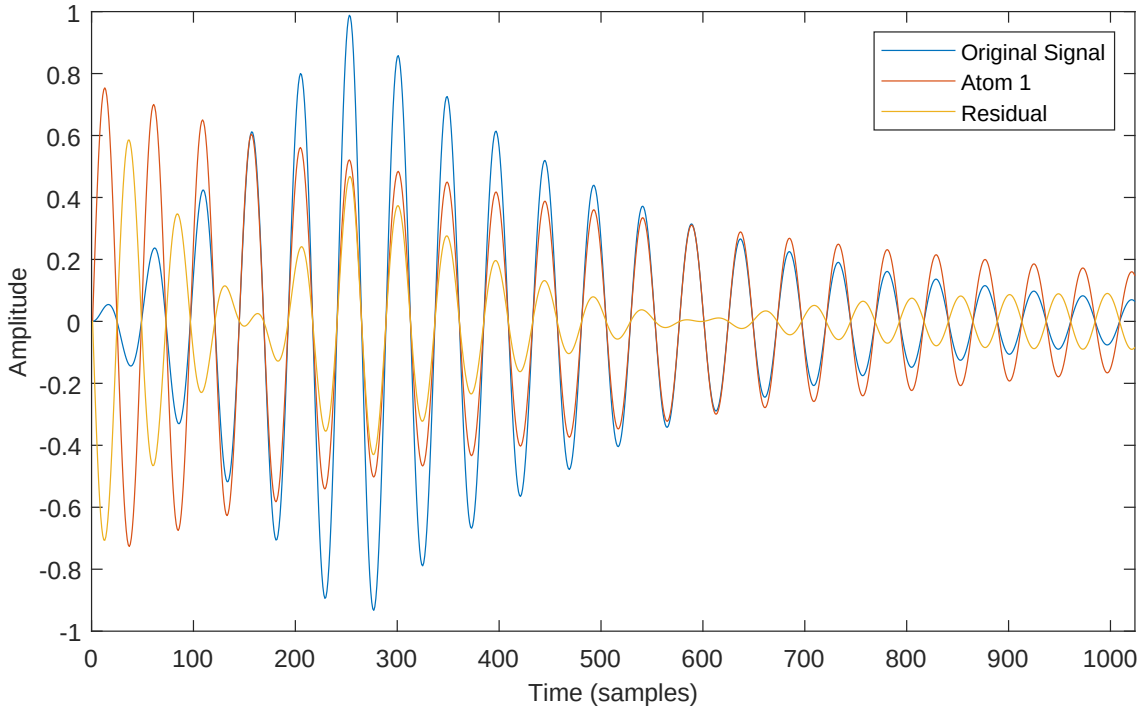
FIGURE 5.9: Sawtooth Bass (@48 kHz) example with magnitude spectrum

Electronic Bass sounds are a lot more diverse than kicks drums. You can have a very tight bass line, where the notes start and stop quickly allowing for a clear sounding bass line due to the notes not overlapping, which has quite some punch and transient to it. Alternatively, bass lines can be modulated to create a wobbling effect and depending on he genre can be much longer in length and overlap with the kick drums to create a slower but deeper sound. Longer bass lines can however cause a muddy bass line, where the low frequencies of the kick and bass blend together and blur the details of the low end, while also influencing and blurring the transients. An example of a single bass note is shown in Figure 5.6, Figure 5.8 shows a tight 3 note bass line for use with a kick drum in dance track with a 4/4 time signature, which is the default for most electronic dance tracks. A 4/4 time signature means that there are four beats in a bar with each quarter note getting one beat. In electronic dance music the 'kick and bass' is composed wit the kick drum as the first note, followed by 3 bass notes on each quarter note within the bar. Figure 5.7 shows what can only be described as a wobbly bass with not much of an attack to it. Examining Figure 5.7b shows a lot of broadband components across the entire spectrum for this bass sound with a relatively smooth modulated attack and release over the length of a bar. However, the magnitude spectrum presented in Figure 5.7d displays a much smoother magnitude spectrum for a frame of 1024 samples compared to the other examples.

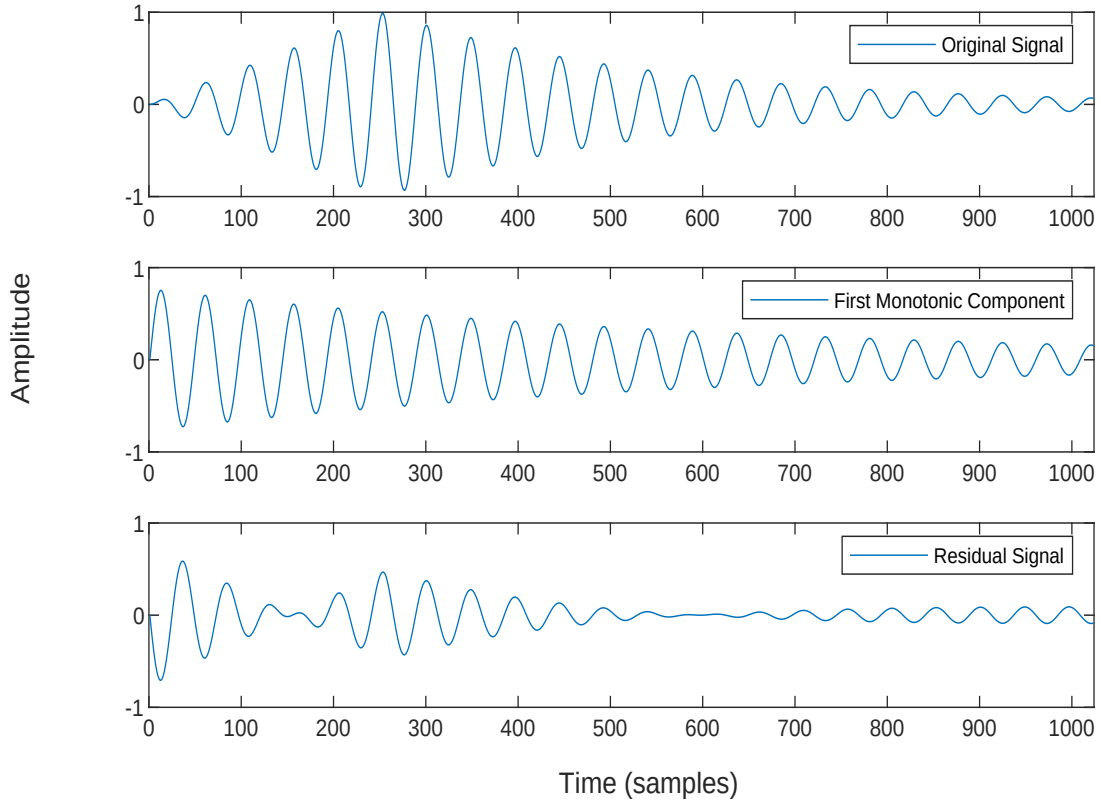
Figure 5.9 shows and example of a richer sounding bass generated using a sawtooth waveform containing both even and odd harmonics of the fundamental frequency.

The above examples all use a rectangular window, and so suffer from some spectral leakage, but this window is selected here and in the causal implementation of the non-overlapping frame based implementation of MoP as it does not smear any transient components in time. The examples are presented here to display the differences in broadband components between different signals which are relevant to the following Section on Transient Modelling 2.6.

The real-time aim of the framework presented in this thesis works best with frame lengths of 512 or at a maximum 1024 samples to reduce the latency required for real-time performance, which translates to about 23 or 46 milliseconds. The method employed to model non-statonary sinusoidal components in this thesis is based on the sparse over complete method of modelled pursuit discussed in Chapter 4. PDA methods of amplitude change provide accurate estimates of monotonic amplitude change. However, transient signals which consist of an attack and decay stage are non-monotonic. Two approaches have been taken in this thesis for modelling transients. The first method only models monotonic amplitude change, leaving non-monotonic amplitude change within the residual signal. In this thesis, the residual signal is modelled using the Undecimated SWT which is discussed in the following sections. Figures 5.10a and 5.10b presents a sinusoid with non-monotonic amplitude change consisting of an attack and decay stage. The initial monotonic component and the resulting residual signals are presented. The initial atom captures the dominant sinusoidal partial from which the sound has been synthesised from along with the more prominent decay stage. However, the resulting residual signal does not represent the transient component or actual signal very clearly. Although this is a simple signal, kick drums in particular can be synthesizes in a similar manner. Leaving this residual signal to be modelled in the wavelet domain, while applying time and pitch modifications to the sinusoidal components presents an interesting problem when combining the two to produce the modified output signal after modification.



(A) Sinusoid with non-monotonic attack and decay stage, initial monotonic component and the residual



(B) Sinusoid with non-monotonic attack and decay stage, initial monotonic component and the residual

FIGURE 5.10: Sinusoid with non-monotonic attack and decay stage, initial monotonic component and the residual

This observation, along with advances in GPU technology and research in GPU audio processing, and a conversation regarding modelling transients as multiple sinusoidal components resulted in the following hypothesis:

"Given advances in parallel processing, how accurate can we model non-stationary sinusoidal components while aiming to be as close to real-time processing as possible. The idea being, given enough resources, let us look at throwing 'everything and the kitchen sink' as the problem." [219].

Traditional sinusoidal models which do not attempt to model transients separately often suffer from what is referred to as pre-echo due to the attack becoming smeared in time [108]. Linear ramped trajectories of sinusoidal peaks between overlapping analysis frames often causes these pre-echo artifacts. Multiband sinusoidal modelling such as that used in [220, 221] uses different window sizes to allow shorter sinusoids at high frequencies which gives a better time resolution and thus diminishes the pre-echo effect. All-inclusive systems for modelling transients and noise with a sinusoidal model have been presented in [222, 223] but it is 'unclear if modifications will be natural' [108]. These systems however reply on an overlap-add method for analysis/synthesis which as mentioned is often responsible for the pre-echo due to linear ramped trajectories of sinusoidal peaks between overlapping frames.

The second approach taken in this thesis as a result of the above hypothesis is to model sinusoidal components including transients and non-monotonic amplitude change as a number of sinusoidal atoms in an atomic decomposition using non-overlapping frames and a rectangular window with causal single frame PDA analysis methods as described in Chapter 4. This technique is is able to model non-stationary complex signals to a high degree of accuracy and as result the amount of transients audible in the residual signal after sinusoidal modelling using this implementation of MoP is very low, but not totally eliminated, and pre-echo artifacts are minimised.

The examples presented below are used for displaying the accuracy of MoP for modelling of transients which has just been discussed, and secondly, to present examples of the residual signal. The number of sinusoids used has a direct impact on the quality of the models output. The use of fewer sinusoidal partials can lead to clicks and discontinuities at frame boundaries in a on-overlapping framework. This can be corrected by maintaining amplitude and phase coherence across frame boundaries with the additional required of tracking and matching up sinusoidal components between frames. The residual signal is also directly affected by the choice of modelling transients as sinusoidal components in an overcomplete representation, or only modelling quasi-stationary sinusoidal components as in the case of the the non-causal implementation where transients are smeared by the use of a Hanning window.

Leaving transients and other broadband components to remain within the residual signal results in a more complex and louder residual signal. However, the techniques for modelling the residual in either case and extracting transient components from noise remains the same. Modelling transients as sinusoidal component in an overcomplete decomposition, or excluding them from the sinusoidal part of the model requires further analysis and qualitative testing, leaving it as a future research directive.

5.1.1 Kick and Bass example

In general kick and bass sounds have been presented separately. An example of a kick, combined with a bass, then reduced by 6 dB before mastering is presented in Section B.1. This section presents the process of combining a kick and a bass line, along with examples of the effect mastering tools have on the resulting signal. The decomposition of the signal and the re-synthesised output from the MoP model, along with the resulting residual signal are presented. The signals consist of a combination of quasi-stationary sinusoidal components, shot-time broadband components, and noise.

Figures 5.11, 5.12 and 5.13 present a kick and bass line with the resulting output, and the residual signal. The residual signal consists of some clicks at note onsets and some broadband noise in between.

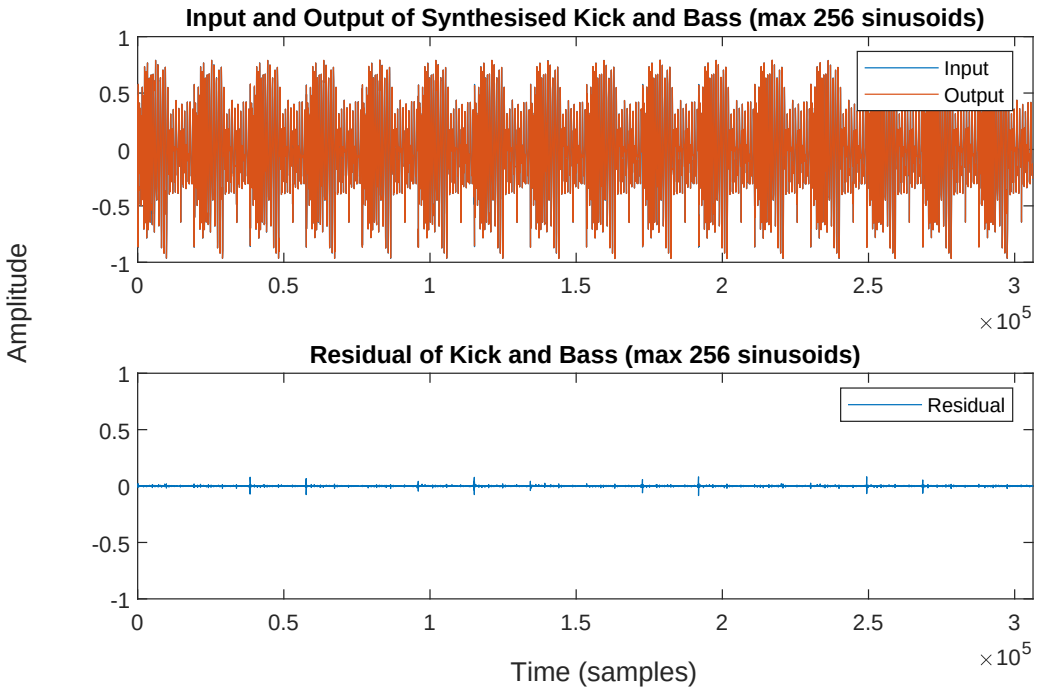


FIGURE 5.11: A kick and bass line (@48 kHz) modelled using MoP and the resulting residual signal. The output from the MoP decomposition is highly accurate.

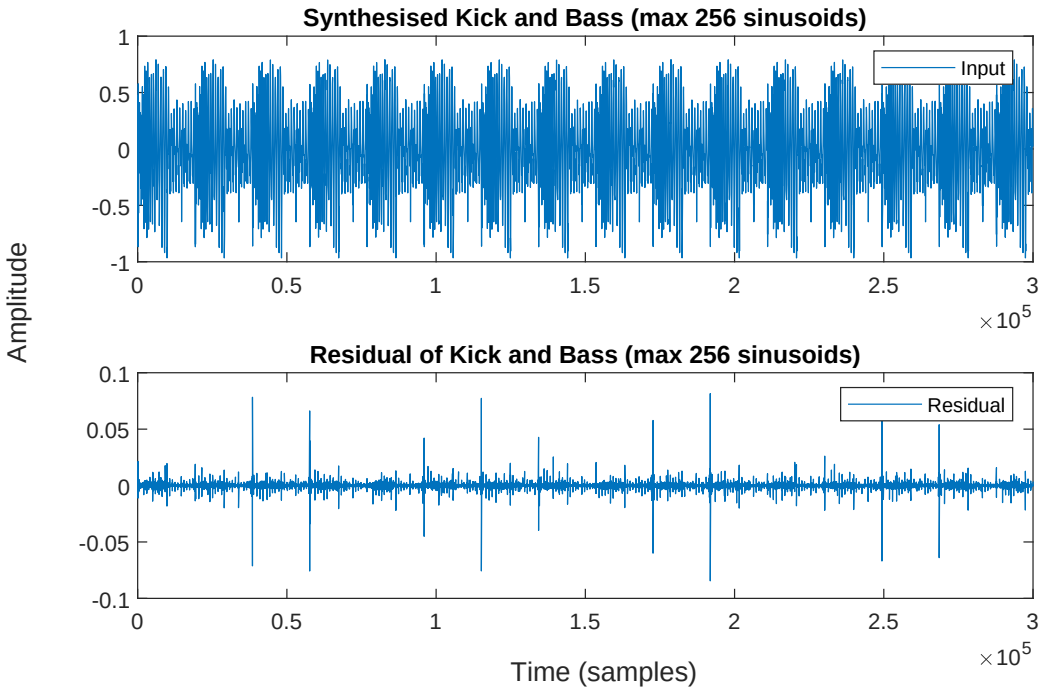


FIGURE 5.12: A kick and bass line (@48 kHz) modelled using MoP and the resulting residual signal zoomed in Amplitude for clearer inspection of clicks at note onsets. [-0.1 0.1]

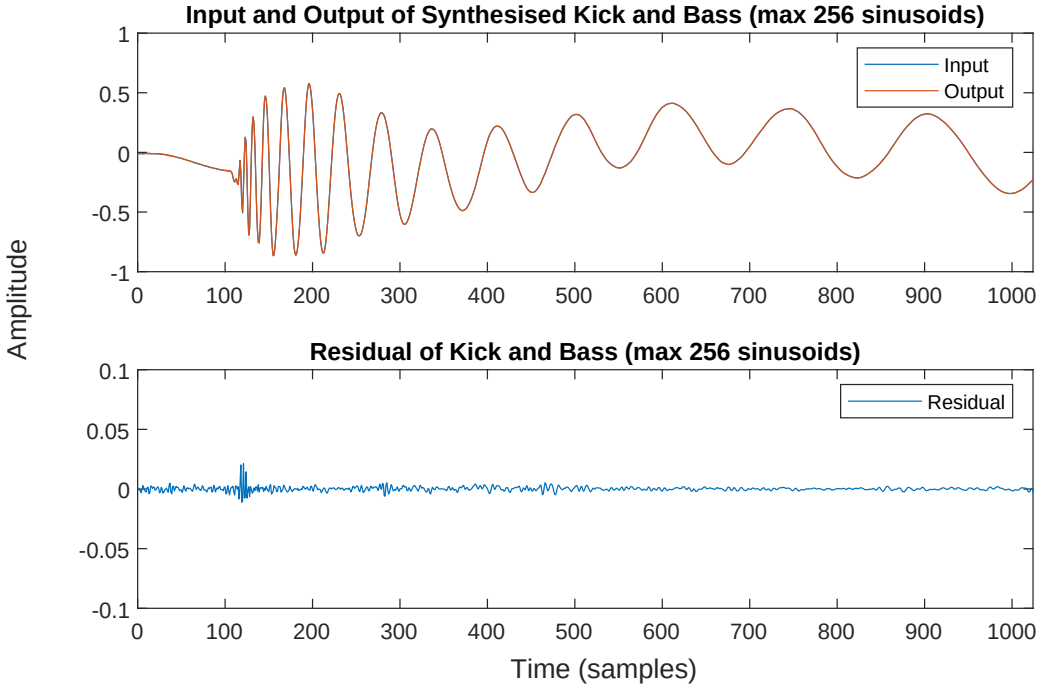
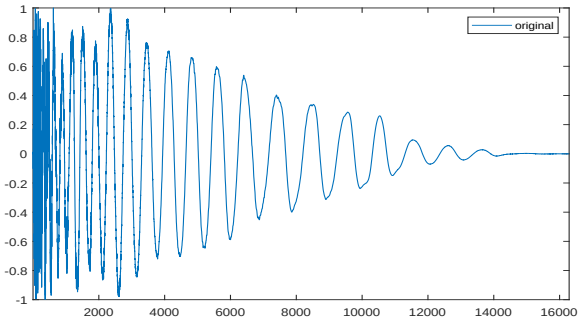


FIGURE 5.13: The first frame (1024 samples) of a kick and bass line 5.11 modelled using MoP and the resulting residual signal for a closer inspection of Input, Output and Residual

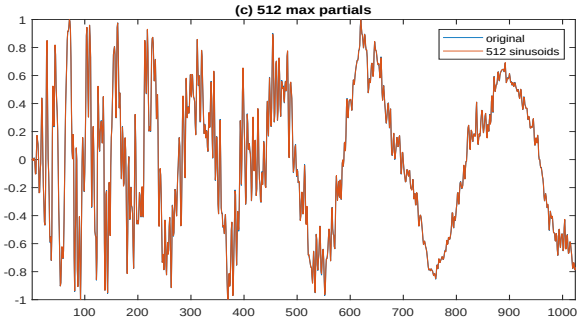
5.1.2 Kick Drum example

Another example of modelling only a kick drum with a number of sinusoidal components is shown in Figure 5.14 which shows the first frame of a kick drum modelled using MoP with non overlapping frames of 1024 samples. The input and modelled output of the first analysis frame comprising of the note onset and and part of the decay are presented. The number of sinusoids used to model the transient part of the kick drum has a direct effect on the quality of the model. Figures 5.14d, 5.14c, 5.14b show how the number of sinusoids used improves the models accuracy with 512 samples providing an almost prefect reconstruction.

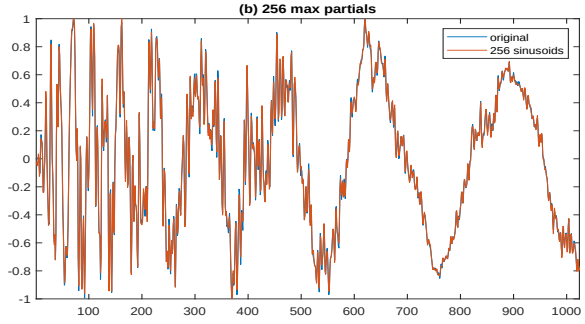
Figures 5.15d, 5.15c, 5.15b show a zoomed in look at the details of modelling the transient part of the kick drum using 128, 256, and 512 sinusoids respectively. While Figures 5.16a, 5.16b, 5.16c show the residual signal of the model. Using 512 sinusids for modelling the transient part of this particular kick drum clearly reproduces highly accurate reconstructed signal.



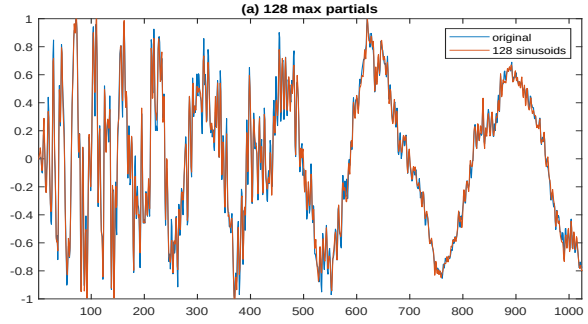
(A) Kick Drum Example



(B) Kick Drum 512 partials (1024 samples)

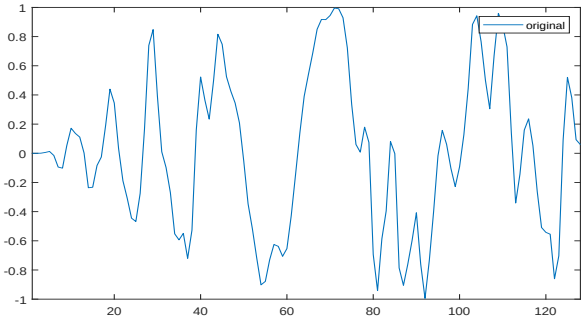


(C) Kick Drum 256 partials (1024 samples)

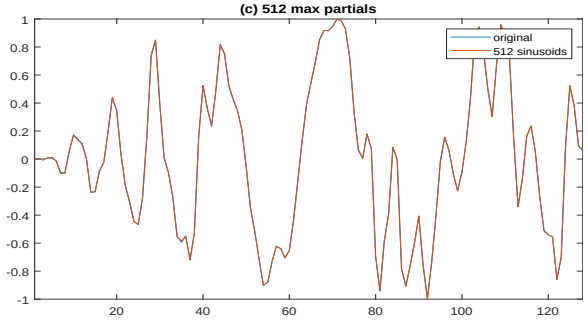


(D) Kick Drum 128 partials (1024 samples)

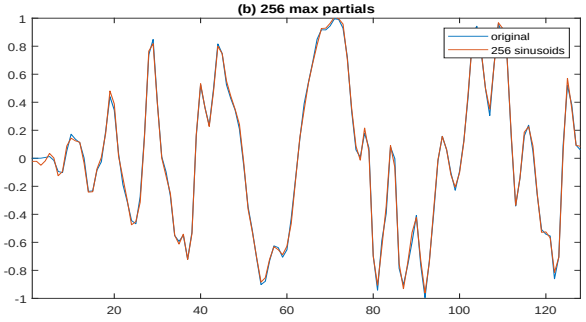
FIGURE 5.14: Kick Drum Example (@48 kHz) modeled with different number of maximum partials



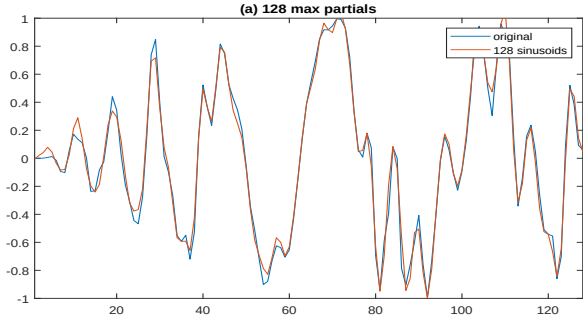
(A) Kick Drum Example



(B) Kick Drum 512 samples(128 samples)

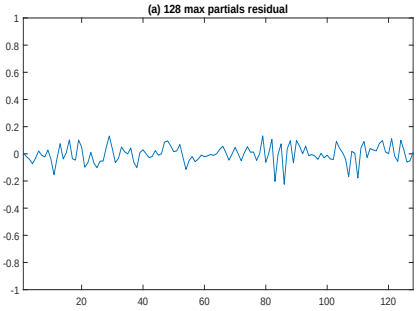


(C) Kick Drum 256 partials (128 samples)

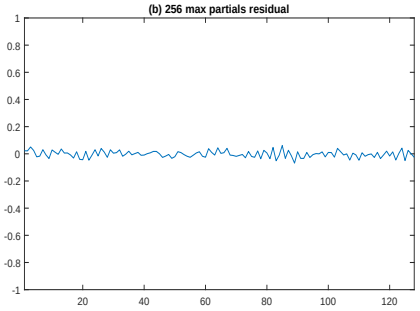


(D) Kick Drum 128 partials (128 samples)

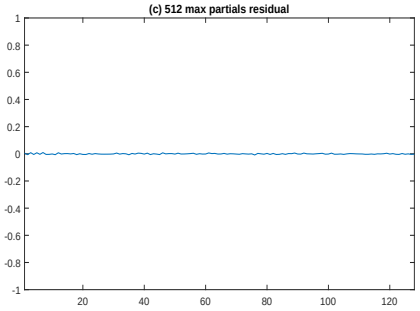
FIGURE 5.15: First 128 samples of a Kick Drum 5.14 modeled with different number of maximum partials



(A) Kick and Bass audio signal to be modelled



(B) Residual signal after spectral modelling



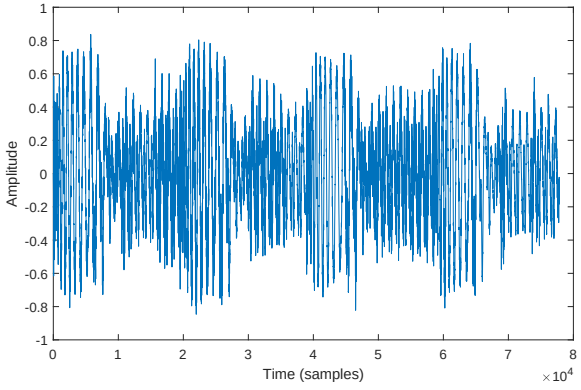
(C) Residual signal after spectral modelling

FIGURE 5.16: Comparison of input signal and residuals

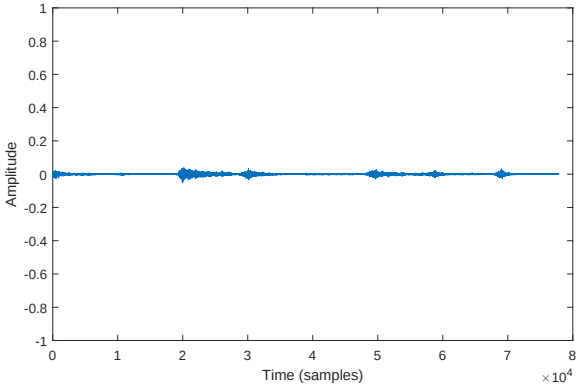
The implementation of MoP is given a maximum number of atoms to use if the modelled pursuit criteria of the residual signal falling below a certain threshold are not yet met. This can be the case for the modelling of transient signals. In this example the number of sinusoids needed to model the sustain and release parts of the kick drum reduces down to only 14 sinusoidal atoms at the most stable part of the kick drum, compared to the 512 atoms required to accurately model the first frame.

5.1.3 Multi-Instrument Dance Track example

Figure 5.17 shows the residual signal in comparison to an input signal comprised of a complete electronic dance track which includes not only kick and bass sounds but percussion and lead synth sounds as well. The residual signal has an energy mean of -50 dB, but does contain a peak amplitude at 0.038 or -28.4126 dB which is low but still audible. The concentration of the energy within the residual signal appears at transient events where percussive elements such at high hats and snare sounds overlap with the kick ad bass components. The maximum number of sinusoidal stoms used for the modelling of this complex audio signal was limited to 256 sinusoids, with an analysis frame of 1024 samples. The modeling of the transients can be improved by extending the number of sinusoids used in the model but computational demands or more complex signals will have an impact on the models accuracy and so the approach of modelling both short and long term components which remain in the residual signal requires further investigation.



(A) Kick and Bass audio signal (@48 kHz) to be modelled



(B) Residual signal after spectral modelling

FIGURE 5.17: Comparison of input signal and residuals

5.1.4 Discussion

Having presented some examples of transient modelling using MoP, and the resulting residual signals, this section concludes with a brief overview of some other approaches to modelling transients. This is followed by Section 5.2 on using wavelets for modelling the residual signal followed by Section 5.3 in which details of the generalised approach for implementing the shift-invariant undecimated version of the discrete wavelet transform in a segmented framework are presented. The chapter concludes with Section 5.4 in which an approach to separating transient components from noise within the residual signal are presented.

Glover et al present an approach of musical note segmentation by identifying the boundaries of note onsets, attack, sustain and release regions in real-time using linear-prediction and other Onset Detecting Functions (ODF) [101] and [214, 224, 225]

Numerous other models have been proposed for the detection and modelling of transients in audio signals. A hybrid sinusodial and source-filter model has been presented in [213]. Exponentially Dampened Sinusoidal models have been extended to Damped and Delayed Sinusoidal (DDS) model as well as modal decomposition of musical instrument sounds via optimization-based non-linear filtering [226] have been explored for modelling transients [174]. Adaptive models such as eaQHM have also been used successfully to model transients, percussive musical instrument sounds, speech and synthetic non-stationary sinusoids [142, 202, 227].

Wavelets have also been investigated for their use in transient modelling due to their ability to zoom in on finer details and detect discontinuities within signals. Wavelets have been suggested for transient modelling in [100] and with application to parametric audio coding in [175, 228]. The Continuous Wavelet Transform (CWT) has been proposed for use in a additive synthesis based Sinusoidal plus Transient Model in [229]. Transient modelling by matching pursuit with a wavelet dictionary for parametric encoding has been presented in [228]. Separation of transient information from musical signals using multiresolution analysis and wavelets has also been presented in [230] using a high frequency content (HFC) onset detection function proposed in [148]. Transient detection and encoding using wavelet coefficient trees has been used to some degree of success [231] but the tree estimation methods failed to resolve transient in all cases requiring further training. Transient modelling using matching pursuit with the application of parametric audio coding was been explored in [175] where wavelet-packet trees using Daubechies filters (asymmetric) provided best results, but suffered when the signal contained a high degree of sinusoidal; content. An anlysis-synthesis model for transient sounds using the Stationary Wavelet Transform (SWT) and Singular Value Decomposition (SVD) was used in [232] but details of using this in a real-time segmented audio framework are not presented. Wells et all describe an approach for modelling residual signals including transients in [9, 217].

In the following section details for using an overcomplete redundant shift-invariant representation using the SWT for modelling the residual signal in a segmented real-time audio system are given.

5.2 Decomposition using Wavelets

"They can't see the forest for the trees" is an expression for when someone is concentrating on the finer details of something too much, and can no longer see the bigger picture. This analogy has been used to explain wavelets in the past describing the wavelet transform as a multiresolution process able to seperate the time frequency resolutions into a non-uniform grid which is able to capture a better resolution of temporal data at high frequencies and capturing low frequency information which evolves over more time on a larger time scale [233–236].

This is an apt analogy for describing the Heisenberg uncertainty principle where the position and the velocity of an object cannot both be measured exactly at the same time. This applies directly to the STFT where the length of the analysis window (in time) has a direct effect on the frequency resolution. A longer analysis frame results in a more detailed frequency resolution, but the information in the frequency domain concerning 'when' something occurs is blurred as the frequency information is given as an average over that span of time. A shorter analysis frame improves the time resolution, while reducing the frequency resolution. Improving the accuracy of both time and frequency information is a strong driving force behind much of the research on signal analysis.

Sines and cosines which form the bases of Fourier analysis span the length of the analysis frame in the STFT. They therefore generally do a poor job at approximating sharp discontinuities or spikes in the data.

As explained in Section 2.10.1 wavelets and the DWT offer a multi-resolution approach to signal analysis where half band filters divide the spectrum into a high frequency band (approximation coefficients g) and a low frequency band (detail coefficients h). This processes is repeated on the low frequency approximation coefficients for a number of levels j as shown in Section 2.29. The wavelet transform provides low time resolution at low frequencies and a high time resolution at high frequencies. A multi-resolution approach offered by wavelets therefore provides a useful tool for modelling transient signals which are left after sinusoidal modelling as well as modelling other long-term noisy components left in the residual signal. The detail coefficients (high passed frequencies) provide good time localisation properties which are appropriate for detecting discontinuities and modelling transients, while the approximation coefficients (low passed frequencies) provide low frequency information regarding the more stationary components [217]. Wells et al propose using wavelet analysis as a multi-resolution approach for the modelling of the residual signal [9, 217]. The assumption that the residual is fully

described by its amplitude and its frequency characteristics implies that it is unnecessary to keep either the instantaneous phase or the exact spectral shape information [50]. However this assumption for short-term transient components within the residual signal is not correct [106]. For long term stationary noise this information is not required, but for impulsive components the phase and magnitude information is in fact critical for capturing the temporal detail of these transient components [9].

The method employed by Wells et al to derive the residual signal from the original and synthesised model uses spectral subtraction which retains the phase and magnitude information in the residual signal, thus retaining the timing information of these components. This is then passed on to complex wavelet analysis for deriving the outputs of this analysis, which are time-varying parameters (gain, centre frequency and bandwidth) used to filter synthesised random noise through a bank of parametric equaliser filters. The aim of this approach was to combine both transient and long term noisy components in a single model [9].

However the method of relating the wavelet filters to the bandwidth of the equalisers was not very satisfactory and the authors highlighted that the need for another approach may be required, while highlighting that the shift invariant undecimated wavelet transform may offer more details of the signal through a redundant representation of the data. The shift invariant of the undecimated wavelet transform makes it an attractive analysis tool [9]. This is related to the disadvantage of the DWT approach as it is critically sampled, meaning there is a decimation stage which causes it to be shift variant. This is an undesirable property as important information is lost in the time domain, therefore applying changes to the wavelet coefficients can result in aliasing in the frequency domain.

The result of discarding every second sample at each decomposition level is that a shift in the input signal can cause large variations in the distribution of energy between wavelet coefficients at different levels or scales, which can be interpreted as aliasing. One method to overcome this property is to have a fully sampled transform which omits the sub-sampling. This is commonly referred to the algorithm à trous [237].

Because there is no down-sampling, the algorithm is said to be overcomplete or redundant (non-orthogonal), but it is shift-invariant. This property comes at the cost of additional computation and memory, and the filters must be dilated (by inserting zeros) at each level of the transform. This dilation of the mother wavelet causes the filters to double in size at each level rather than the number of wavelet coefficients halving in size, and as such poses a slightly different problem for implementation in a real-time segmented system.

Some statistical applications of the SWT are highlighted in [238] where it is clearly shown how the redundant overcomplete multi-resolution approach to the wavelet transform provides more detail and is an important tool in transient analysis and de-noising. The undecimated wavelet transform is also shown to be an important tool for transient detection and de-noising in [239] and [240] respectively.

Wavelets provide another set of functions which are well localised in time and frequency and so provide insights into signal details such discontinuities and short bursts or changes in the signal, which sinusoids can not provide. The ability of the undecimated wavelet transform for detecting and modelling transients as well as its use in de-noising makes it an attractive and interesting tool for modelling the residual signal in a spectral modelling framework.

However, one of the potential issues of real-time processing in the wavelet domain, is the problem of block-end artefacts at frame boundaries due to convolution. Applying the DWT to a segmented real-time signal while eliminating any block-end artefacts has been a topic of interest in recent years. There have been several papers on the wavelet transform and applying the DWT, wavelet packet transform (WPT), and M-band wavelet techniques on finite audio frames for specific applications [241, 242] including a number of papers on a more general and transferable approach to applying the wavelet transform in real-time to finite length audio frames [243–246].

Work for implementing a framework for applying effects in real-time in the wavelet domain was presented in [247, 248]. The SegWT algorithm, provides a general approach to applying the forward and inverse DWT while eliminating block-end effects.

Implementing an undecimated or partially decimated wavelet transform in a multi-resolution approach to residual modelling and transient detection was proposed in [217] but its implementation did not consider the effects of windowing on the process in much detail.

The redundant undecimated wavelet transform, algorithm à trous, and Stationary wavelet transform are all related and provide a redundant shift-invariant wavelet decomposition by avoiding the downsampling at each decomposition level and instead upsampling the filters by inserting zeros between the coefficients. This has become a popular tool for both transient detection and de-noising but as far as this author is aware, details of using this in a real-time segmented audio framework with details of a generic solution to dealing with convolution block-end artifacts at frame boundaries has not been presented.

The method of deriving filter coefficients for a parametric bank of equalisers for re-synthsising the residual in [217] was unsatisfactory but a segmented inverse undecimatated transform on modified coefficients could provide a more accurate model for re-synthesising the residual, offering the flexibility of de-noising and filtering certain frequency bands before reconstruction.

The following section aims to address the practical issues of implementing such a system in a real-time single frame analysis synthesis system. The problem of dealing with block-end effects and the different delay times introduced by the filtering process at each level is presented for both the forward and inverse transforms. The use of this for transient detection and de-noising is then presented in the context of modelling the residual signal in a spectral modelling framework. The ability to split the residual signal into different frequency bands while retaining shift invariant properties and inverting the transform in a frame based system leading on to future thoughts about filtering certain frequency bands, de-noising the residual signal to improve transient detection and opens up the possibility of using the tool as a pre-processing step for transient detection and modelling before the spectral modelling stage where transients are currently modelled as an overcomplete representation of short lasting sinusoidal components.

5.3 Residual Modelling in a single frame Framework

Segmenting an audio signal into frames is a common way of efficiently processing an audio stream and is necessary for any real-time application. Each of these frames is then processed individually, possibly altered in some way and then presented at the output as a frame with the same number of samples as in the input frame. The processing and streaming audio in real-time is implemented in this way, as are audio encoders/decoders which depending on the application, may require storing samples from the previous frame when processing the current frame. An example of this is the Phase Vocoder [72] where frequency estimates from Fourier data are improved by using the derivative of the phase from the previous and current frames.

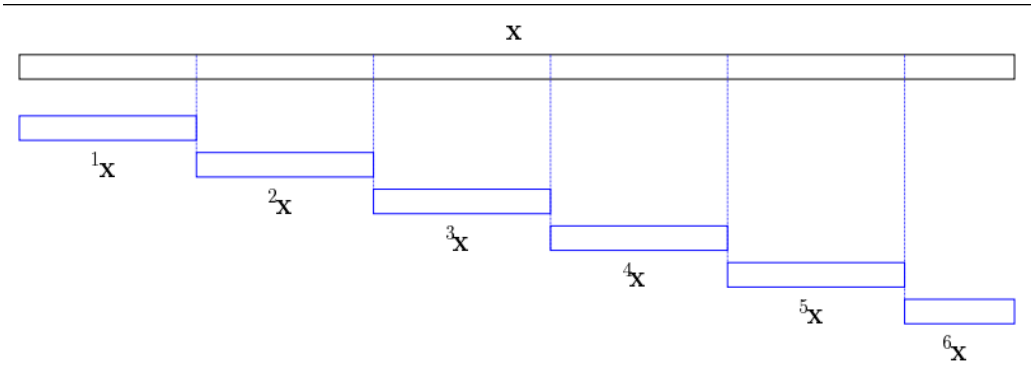


FIGURE 5.18: Example of signal segmentation

Smaller audio frames reduce the latency of the data sent to the output buffer after being processed. Many low latency audio devices will work with frame sizes as low as 32 samples in length. Spectral models requiring a trade-off between time and frequency resolution will in general require larger frames to improve the frequency resolution. The frequency resolution of the DFT is given by the sampling frequency divided by the size of the FFT. A frame size of 1024 samples at 48 kHz sampling rate with no zero padding will give you a frequency resolution of 46.875 Hz and a latency of 10.7 ms. For real-time performance any latency greater than 10 ms will be noticeable by the performer [9]. When sampling at 48 kHz, it is therefore desirable to keep the frame size at or below 1024 samples. The frequency resolution can be improved by zero padding as explained in 2.4.2, in this case zero padding by a factor of 8 will improve the frequency resolution to 5.8594 Hz.

Many spectral models employ windowing for reducing spectral leakage. This requires overlapping frames and using Overlap-Add (OLA) for adding overlapping output frames together. Care needs to be taken with the hop size to achieve the Constant OverLap-Add (COLA) property as presented in Section 2.4.5.

Other segmented audio frameworks which implement convolution for reverberation or other convolution based effects require an Overlap-Add or Overlap-Save mechanism for dealing with block-end artifacts due to the convolution operation resulting in more samples in the output frame than the amount of samples from the input frame.

Overlap-add resolves the issue of applying convolution in a segmented framework by zero padding the current segment, applying convolution to the frame and saving the extra samples from the convolution operation beyond the frame length, and then add them to the overlapping output samples from the beginning of the next frame.

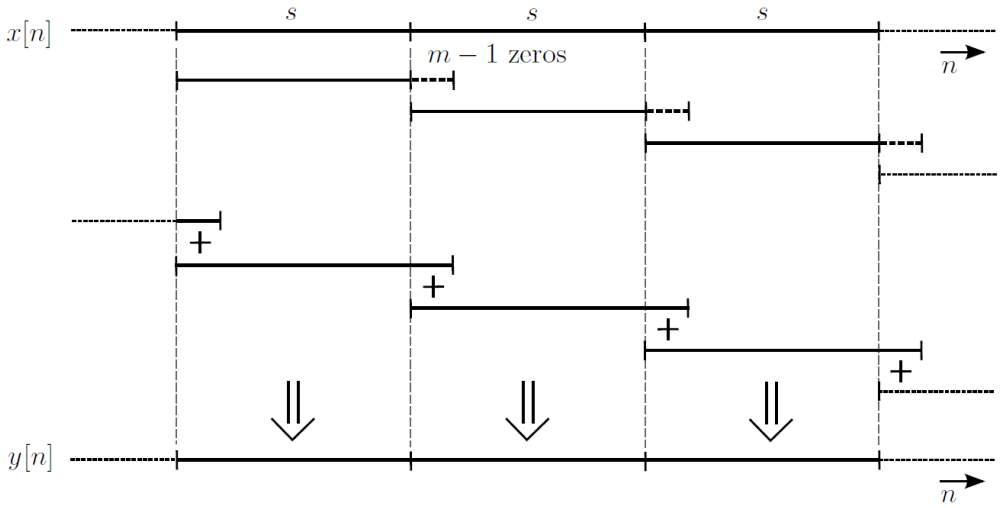


FIGURE 5.19: Overlap Add: $m-1$ zero samples are appended to the end of each segment. After convolution, these overlapping samples are added to the beginning of the next segment.

Conversely, with an Overlap-Save implementation, the signal is divided into equal length frames. Some samples from the previous frame are prepended to the current input frame. The start of the current frame therefore overlaps the end of the previous fame. After the convolution operation, the output samples which correspond to the current audio frame are saved and presented in the output frame.

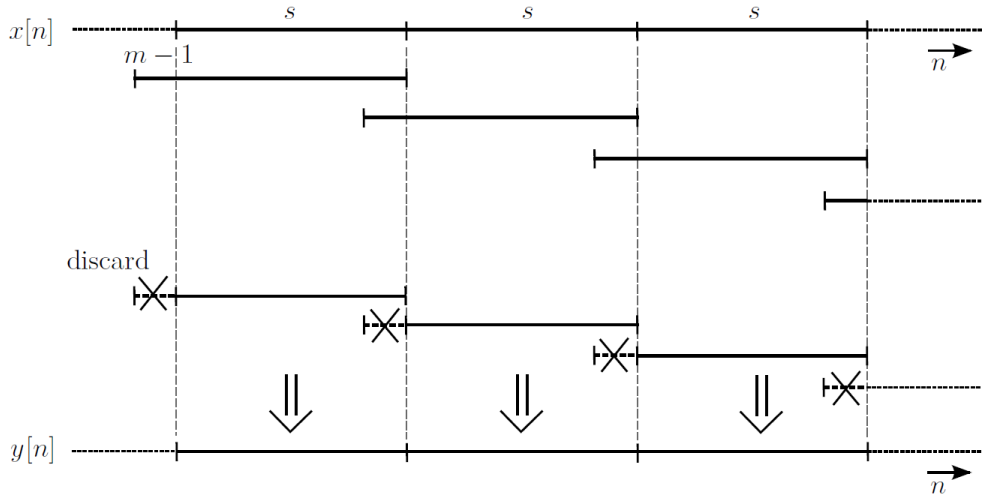


FIGURE 5.20: Overlap Save: The last $m-1$ samples of each segment are saved and prepended to the following segment.

Analysis and re-synthesis of the residual signal in a single frame segmented framework using the forward and inverse wavelet transform also requires the implementation of an overlap extension scheme for the elimination of block-end artifacts due to convolution. The details of which are made more complex in the undecimated implementation, due to the length of the wavelet filters doubling at each level of the decomposition.

The following section discusses the segmented Wavelet Transform (SegWT), as well as the derivation of a general solution for implementing the undecimated wavelet transform in a segmented framework.

5.3.1 Segmented DWT

Wavelets can be thought of as filter banks [249], where the filter coefficients are applied by convolution. A large portion of the documentation on wavelets approaches the problem from the view that the entire audio signal is known, and the entire audio signal is processed and treated as a single segment. Due to the nature of convolution, the number of output samples is greater than the number of input samples, which poses a problem when performing the operation in a segmented, block based framework. Methods for overcoming block-end effects include OLA and OLS, as well as real-time partitioned convolution algorithms [250, 251]. In the following section, the following parameters are important with regards to calculating frame boundary extension lengths and overcoming these block end effects.

m wavelet filter length, $m > 0$,

j transform depth, $j > 0$,

s length of segment, $s > 0$.

Methods for dealing with block-end effects in the wavelet domain have been proposed in the past. The most notable contributions on processing the DWT in real-time by [243–248]. Block-end effects at frame boundaries are avoided in [241] by using samples and transform coefficients from the previous block to carry out the filtering operations of the wavelet packets forward and inverse transform. They found that for any segment $s(n)$, $m-1$ samples from the previous frame must be prepended to the current frame in the analysis stage, and one transform coefficient from the previous segment $s(n-1)$ was required in the synthesis process. The initial block of transform coefficients would have to be prepended with zero-value samples and so a total delay of

$$\left(2^j - 1\right)(m - 1) \tag{5.1}$$

samples are introduced in the overall analysis and synthesis process at each decomposition level j. This technique was adopted and modified in [252] where it was successfully applied a wavelet table approach to apply audio effects in the wavelet domain at the cost of increased bandwidth and complexity. They implemented a lapped wavelet packet transform where the wavelet transform block was doubled in size, while keeping the reconstructed block the same length.

In [243–248] a new method of the segmented wavelet transform was presented, which makes it possible to process the Discrete Wavelet Transform (DWT) in real-time. Their algorithm is a general approach for performing the DWT on any segment length as if the entire signal was known in advance and is not limited to the forward transform only. This is accomplished by extending the segments on both the left and right by specific amounts and performing the overlap and add (OLA) efficient convolution technique on the overlapping segments. The work for finding a general approach to applying the DWT to segmented audio data was motivated by the errors introduced in the reconstruction part of the past algorithms. In [243, 244] the SegWT algorithm was initially introduced, which demonstrated performing the analysis stage of the DWT in real-time by calculating the optimal border extensions of signal segments. The computational efficiency of the algorithm was improved generalised further to include biorthogonal wavelets in [245]. The inverse transform of the SegWT was proposed in [246] and presented as a VST plug-in in [248]. The SegWT finds a maximum number of left extension samples and a minimum number of right extension samples, which satisfies certain criteria, resulting in an expression for the number of shared samples between two consecutive segments. The SegWT is devised as a generic approach to resolving block end artifacts at frame boundaries using a segmentation extension technique whereby the optimum number of samples are used for extending individual segments at each level j in such a way that the coefficients are the same as if the wavelet transform had been applied to the entire signal as a whole.

An example of the DWT applied to segmented audio frames is shown in Figure 5.21. A signal of 1024 samples is segmented into 4 blocks of 256 samples. The resulting approximation and detail coefficients of the wavelet transform are shown at each level of the decomposition and reconstruction. The length of each block is halved at each successive decomposition level j due to the down-sampling operating applied to the wavelet coefficients. The blocks double in length at each level j during reconstruction

(iDWT) due to the upsampling operation applied to the approximation and detail coefficients at each level.

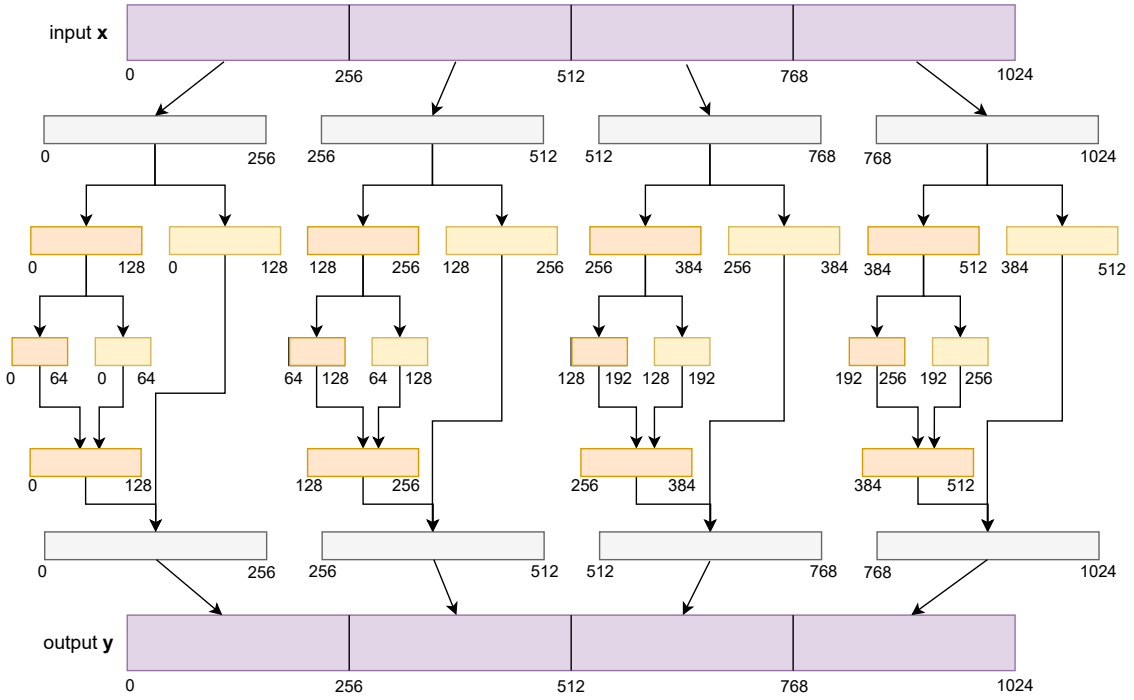


FIGURE 5.21: Example of signal segmentation and resulting frame sizes in level j=2 wavelet decomposition (DWT) and reconstruction (iDWT) with a frame size of 256 samples. Notice the approximation and Detail coefficients half in length at each successive level due to the down-sampling operation.

An example of the SegWT and the use of border extensions applied to an input signal at the initial decomposition stage is shown in Figure 5.22. The example shows different extension lengths from segment to segment.

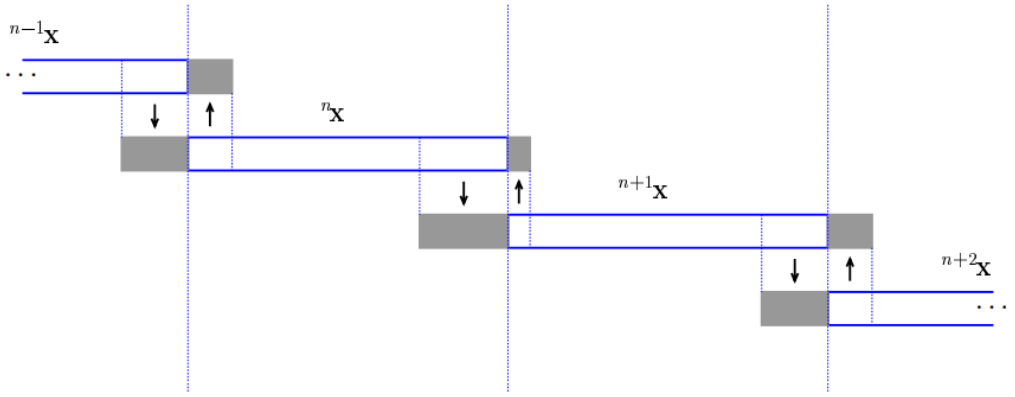


FIGURE 5.22: Segmentation of the signal x and the principle of border extensions. Note that the lengths of the extensions differ from segment to segment. [248]

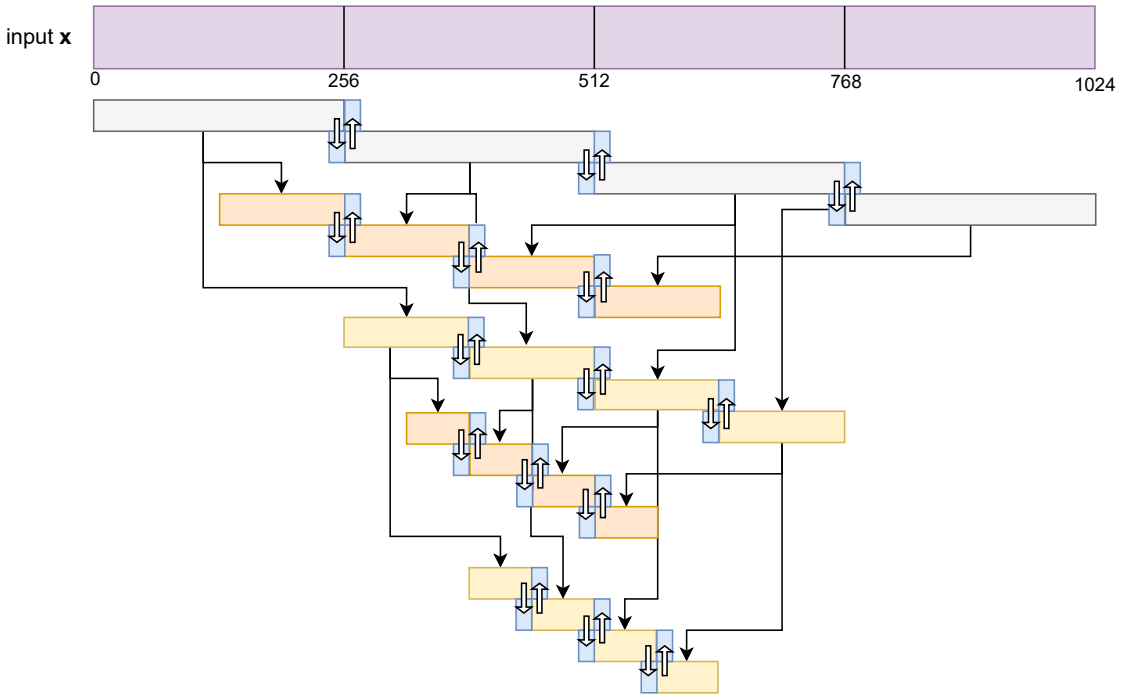


FIGURE 5.23: Example of signal segmentation and border extensions at different levels of the segmented DWT. The example shows the approximation and detail coefficients from Figure 5.21 down to level j=2 wavelet decomposition with an initial frame length of 256 samples.

Figure 5.23 shows the SegWT and overlap extensions applied to the wavelet coefficients at multiple decomposition levels. The extension lengths differ at each decomposition level, and can also vary between segments. A table with an example of overlap extension lengths for different segments and decomposition levels is shown in Figure 5.25.

The size of each left and right extensions is not constant at each decomposition level. The length of the left and right extensions can change between blocks, but there is a constant number of common samples at each decomposition level j, given by

$$r(j) = \left(2^j - 1\right)(m - 1) \tag{5.2}$$

where m is the wavelet filter length. Two consecutive segments will have $r(j)$ common input samples after they have been extended. At a decomposition level j, it is necessary to have $r(j)$ common samples in the two consecutive segments. "This extension has to be divided into the right extension of the first segment (of length R) and the left extension of the following segment (of length L) so that r(J) = R + L, however R, L $\geq$ 0 cannot be chosen arbitrarily" [246].

The maximum possible left extension is given by

$$L_{\max} = l - 2^j \text{ ceil } \left(\frac{l - r(j)}{2^j} \right) \tag{5.3}$$

where l is the length of a segment (including its current left extension).

The minimum possible right extension is given by

$$R_{\min} = r(j) - L_{\max} \tag{5.4}$$

Letting $L_{\max}(n)$ and $R_{\min}(n)$ equal to the left and right extension lengths respectively; of the nth segment, and denoting $l(n)$ as the length of the nth segment and the left extension allows Equation 5.4 to be written as

$$R_{\min}(n) = r(j) - L_{\max}(n+1) \tag{5.5}$$

The length of the right extension of the nth segment must comply with

$$R_{\min}(n) = 2^j \text{ ceil } \left(\frac{ns}{2^j} \right) - ns \tag{5.6}$$

where s is the length of the segment, and ns is the index of the left-most sample within the nth segment (in the global point of view, prior to the extension). The left extension of the $(n+1)$th segment is given by

$$L_{\max}(\mathrm{n}+1) = \mathrm{r(j)} - R_{\min}(n) \tag{5.7}$$

The left extension ensures that there are enough samples from a previous segment to correctly calculate the wavelet coefficients at the top level of the decomposition for the current segment. The right extension aligns the end of the segments to a power of and ensures the correct alignment of the wavelet coefficients. In this way, the SegWT can successfully apply the DWT in a block based way. The optimal number of extension and overlapping samples is ensured and the algorithm is generic enough to deal with blocks of different sizes.

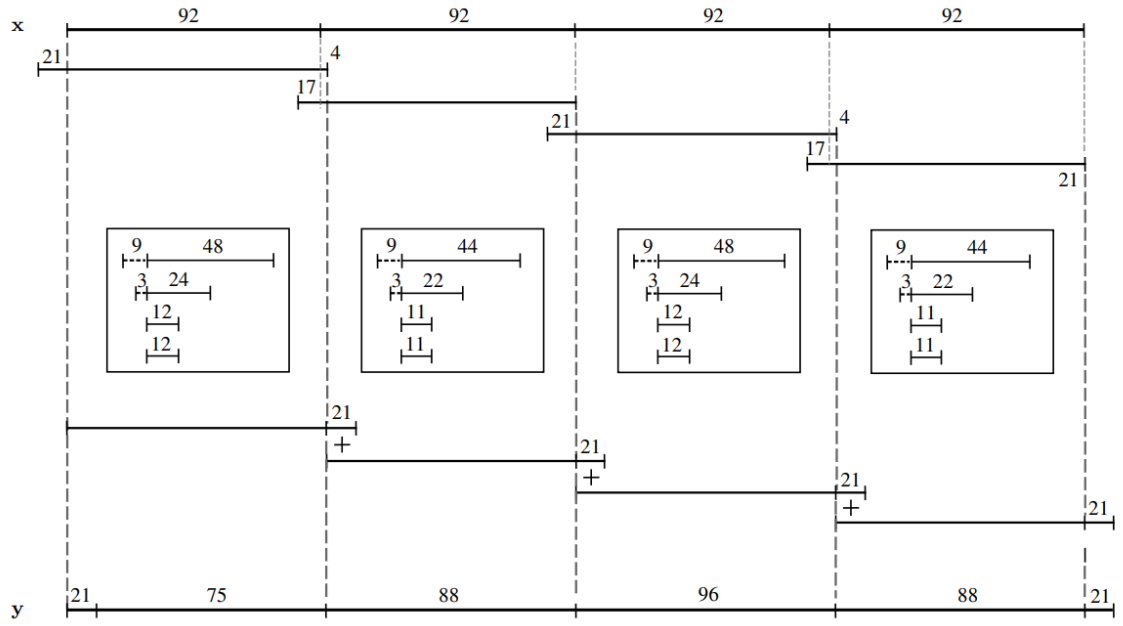


FIGURE 5.24: SegWT algorithm example: Input signal x is processed by segments of length $s = 92$. The length of the wavelet filters is $m = 4$ and the depth of decomposition is $J = 3$. This setup leads to $r(J) = 21$, which is divided between $L(Sn)$ and $R(Sn)$. Note that the reconstructed signal y is delayed by these $r(J)$ samples; the first $r(J)$ samples of the reconstructed signal can be viewed as the "reconstruction warmup" and should be set to zero. The values in the boxes represent wavelet coefficient vectors (from top to bottom, the detail coefficients vectors for $j = 1, 2, 3$ and one approximation coefficients vector for $j = 3$) belonging to the respective segments. The highlighted coefficients in levels $j = 1, 2$ are discarded, and in the inversion they are appended back as zeros. [246]

Figure 5.24 shows an example of different left and right extensions lengths applied to audio segments of 92 samples, a wavelet filters length of $m = 4$ and a decomposition level $J = 3$. This shows that the size of the extensions used by the SegWT can oscillate between the min and max L and R extension values between successive frames. This is dependant on the frame size, length of the wavelet filter and the specific decomposition level. Figure 5.25 shows a table with the resulting L_{max} and R_{min} calculations for different segment lengths.

The forward SegWT is implemented using "overlap-save". Previous samples are reused and unnecessary wavelet coefficients are discarded after the convolution operation. The inverse SegWT uses "overlap-add". The length of a reconstructed segment s_{rec} depends on the lengths of right extensions and it can be calculated by

$$s_{\text{rec}}\ (S_n) = s + R\,(S_{n+1}) - R\,(S_n) + r(J). \tag{5.8}$$

s	n	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	...
512	$L_{\max}(n)$	105	105	105	105	105	105	105	105	105	105	105	105	...
	$R_{\min}(n)$	0	0	0	0	0	0	0	0	0	0	0	0	...
	$\sum(n)$	617	617	617	617	617	617	617	617	617	617	617	617	...
513	$L_{\max}(n)$	105	98	99	100	101	102	103	104	105	98	99	100	...
	$R_{\min}(n)$	7	6	5	4	3	2	1	0	7	6	5	4	...
	$\sum(n)$	625	617	617	617	617	617	617	617	625	617	617	617	...
514	$L_{\max}(n)$	105	99	101	103	105	99	101	103	105	99	101	103	...
	$R_{\min}(n)$	6	4	2	0	6	4	2	0	6	4	2	0	...
	$\sum(n)$	625	617	617	617	625	617	617	617	625	617	617	617	...
515	$L_{\max}(n)$	105	100	103	98	101	104	99	102	105	100	103	98	...
	$R_{\min}(n)$	5	2	7	4	1	6	3	0	5	2	7	4	...
	$\sum(n)$	625	617	625	617	617	625	617	617	625	617	625	617	...
516	$L_{\max}(n)$	105	101	105	101	105	101	105	101	105	101	105	101	...
	$R_{\min}(n)$	4	0	4	0	4	0	4	0	4	0	4	0	...
	$\sum(n)$	625	617	625	617	625	617	625	617	625	617	625	617	...
517	$L_{\max}(n)$	105	102	99	104	101	98	103	100	105	102	99	104	...
	$R_{\min}(n)$	3	6	1	4	7	2	5	0	3	6	1	4	...
	$\sum(n)$	625	625	617	625	625	617	625	617	625	625	617	625	...
518	$L_{\max}(n)$	105	103	101	99	105	103	101	99	105	103	101	99	...
	$R_{\min}(n)$	2	4	6	0	2	4	6	0	2	4	6	0	...
	$\sum(n)$	625	625	625	617	625	625	625	617	625	625	625	617	...
519	$L_{\max}(n)$	105	104	103	102	101	100	99	98	105	104	103	102	...
	$R_{\min}(n)$	1	2	3	4	5	6	7	0	1	2	3	4	...
	$\sum(n)$	625	625	625	625	625	625	625	617	625	625	625	625	...
520	$L_{\max}(n)$	105	105	105	105	105	105	105	105	105	105	105	105	...
	$R_{\min}(n)$	0	0	0	0	0	0	0	0	0	0	0	0	...
	$\sum(n)$	625	625	625	625	625	625	625	625	625	625	625	625	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱

FIGURE 5.25: Example: Lengths of extensions for different lengths of segments s. The depth of decomposition is $j = 3$ and the filter length is $m = 16$. [246]

5.3.2 Segmented Undecimated Wavelet Transform

The undecimated wavelet transform otherwise known as the à trous algorithm or Stationary Wavelet Transform (SWT) and other names [253], is the undecimated version of discrete wavelet transform. It has the desirable property of shift-invariance which gives better performance in denoising. Also, as a discretized version of continuous wavelet transform, so it is useful in signal analysis and prediction.

5.3.2.1 Forward Transform

Unlike the DWT, the à trous algorithm omits the decimation stage. As such the number of wavelet coefficients remains constant as shown in Figure 5.26. It is proposed that the left extension of the SegWT could be removed, keeping only the right extension with the number of overlapping samples being related to the filter length at a specific decomposition level as shown in Figure 5.27.

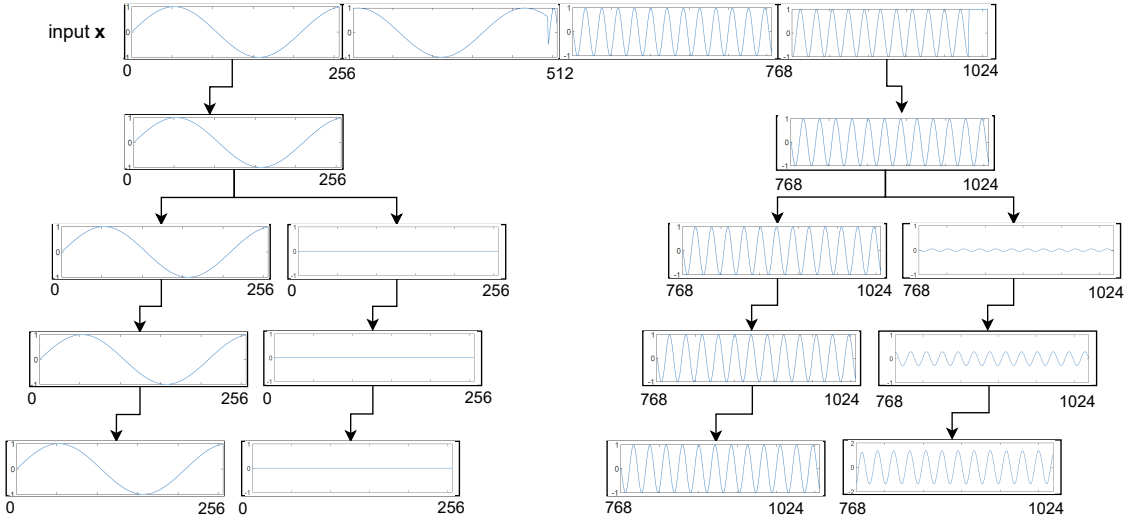


FIGURE 5.26: Example of signal segmentation for a level j=3 undecimated wavelet decomposition (UWT). Decomposition of the $1st$ and $4th$ frames are shown. The frame lengths remain the same at each decomposition level due to the absence of the down-sampling operation present in the DWT which is performed on the output coefficients at each level.

"The OverLap-Save (OLS) method, unlike OLA, uses no zero padding to prevent time aliasing. Instead, it discards output samples corrupted by time aliasing each frame, and overlaps the input frames by the same amount. In general, if the input frame size is $M = N$ and the FIR filter length is $L < N$, then after convolution, $L - 1$ samples of the output are invalid due to time aliasing For causal filters of length L : The invalid samples are the first $L - 1$ samples of each length N inverse FFT, because these samples are computed using time-aliased samples from the end of the length N FFT input frame Therefore, the input signal should have at least $L-1$ leading zeros The hop size is set to $R = N-L+1$ so that the last $L - 1$ samples of frame 1 become the first segment of frame 2. These samples have already been output from frame 1 and can now be overwritten by time aliasing by the processing of frame 2. The length N blocks overlap by $L - 1$ samples. $L - 1$ samples from the previous block are "saved" rather than reread from disk–hence the name "OverLap Save (OLS)". For anticausal filters: The invalid samples are at the end of the frame The input signal needs no leading zeros The hop size is again $R = N - L + 1$ Samples 0 through $R - 1$ are written out, ignoring the last $L - 1$ samples corrupted by time aliasing". [254]

As discussed previously with traditional convolution, the overlap length is given by $m - 1$, where m is the filter length. In the à trous algorithm the filter doubles in length at each decomposition level, and so the length of the filter is given by.

$$\left(2^{(j-1)}\right) m \tag{5.9}$$

Therefore, the overlap length at each level is taken by calculating the length of the filter at that level and then subtracting one.

$$\left(\left(2^{(j-1)}\right) m\right) - 1 \tag{5.10}$$

When you inspect the upsampled filters, the effect of inserting zeros between the coefficients results in several zero terms at the end of the filter. These zero valued coefficients have no impact on the result of the inner product and so a slightly more efficient implementation is to remove these zero valued coefficients which yields the calculation for the number of overlapping samples

$$\left(2^{(j-1)}\right) (m - 1) \tag{5.11}$$

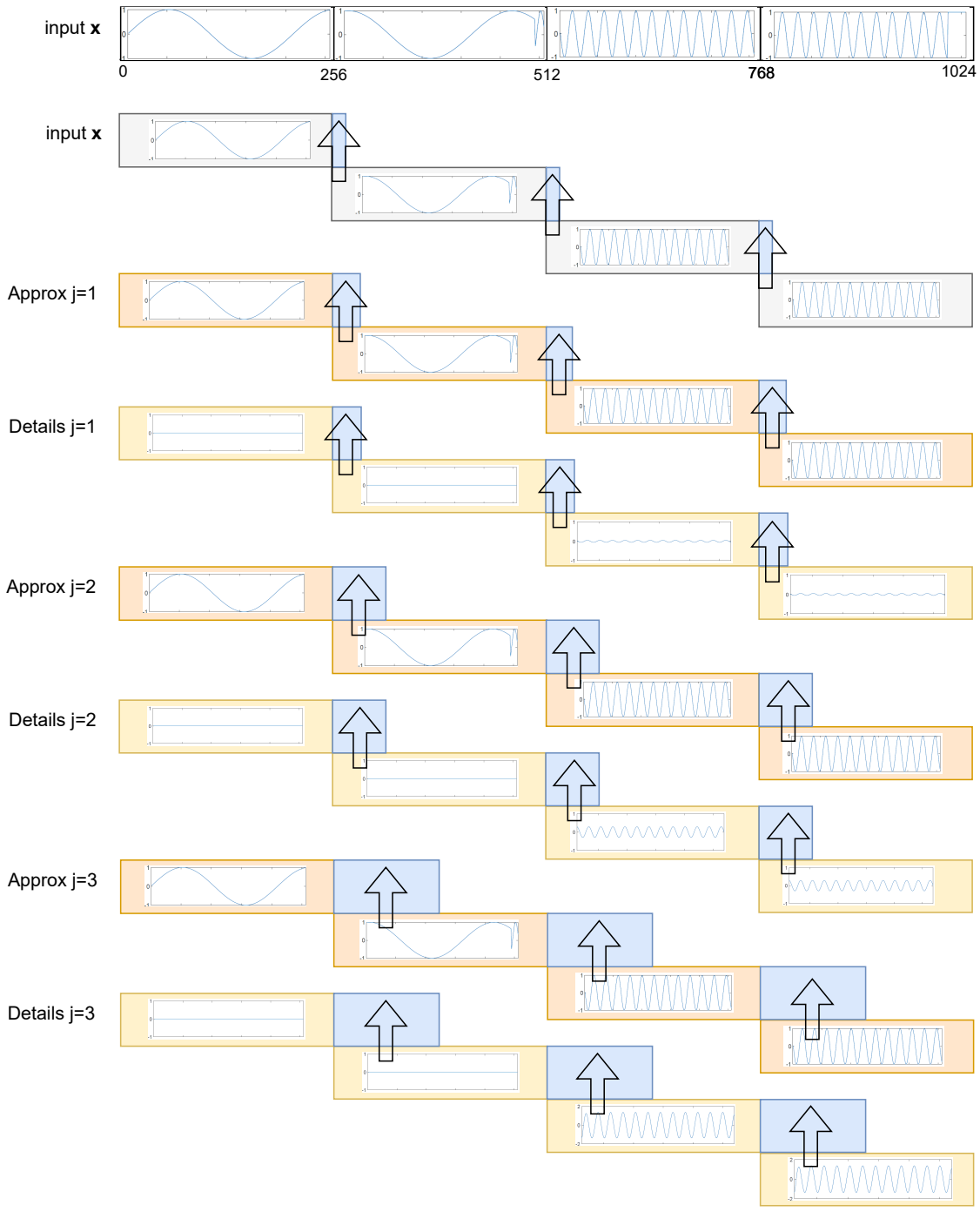


FIGURE 5.27: Example of signal segmentation and border extensions at different levels of the segmented UWT. The example shows the approximation and detail coefficients for a level j=3 wavelet decomposition with an initial frame length of 256 samples. The length of the overlap doubles at each level due to the upsampling of the wavelet filter coefficients at each level. The inverse transform is the same as depicted but starting from the approximation and detail coefficients at level j=3 and working up. The overlap length for the iUWT therefore halves at each level of the reconstruction compared to doubling in the forward transform.

Without the decimation stage of the DWT, the problem becomes one which is well known and can be solved by implementing the transform as a collection of FIR filters keeping track of the number of past filter states given by 5.11. This approach solves the condition where the filter length becomes greater than the frame size by holding a greater number of filter states in memory than the frame size, but the delay introduced by the large number of filter coefficients means that the final output samples can only be presented possibly after more than a single frame of audio has been processed.

5.3.2.2 Compensating Delays

As mentioned in the previous section, when you omit the decimation stage of the DWT, the number of wavelet coefficients remains the same at each level, and the algorithm can be implemented using standard convolution techniques where the number of overlapping samples increases at each level due to the doubling of the filter lengths.

The second practical problem to account for are the delays introduced at each level. Linear phase (symmetric and antisymmetric) filters have a group delay α of

$$\alpha = \frac{m-1}{2} \tag{5.12}$$

where m is the length of the filter [255].

The group delay for minimum or maximum phase FIR filers is not constant for all frequencies but for wavelet filters without linear phase it was found that $ceil(\alpha) = \lceil \frac{m-1}{2} \rceil$ samples were required for delay compensation. In practise the number of wavelet coefficients is padded with leading or trailing zeros and m is always an even number which results in $\alpha = (m_j)/2$ where m_j is the length of the filter at the j-th decomposition level.

Unless you take the delay introduced at each level into consideration when truncating your initial frames results, this delay propagates through the wavelet coefficients at each level. The delay introduced at each level is compounded with the delay introduced at the previous level, resulting in

$$\alpha_j = \frac{m_j}{2} + \alpha_{j-1} \tag{5.13}$$

where α_j is the delay compensation of the filter at the j-th decomposition level.

For the algorithm à trous, the filter length doubles at each level resulting in a larger delay compensations at each level. The amount of decomposition therefore affects the total / maximum amount of delay compensation required. The delay introduced at a specific level of decomposition is given by

$$\alpha_j = \frac{(m_j - 1)}{2 + \alpha_{j-1}} \tag{5.14}$$

This can be rewritten as:

$$\alpha_j = \frac{\left(\left(2^{(j-1)}\right)(m-1)\right)}{2} \tag{5.15}$$

One of the potential problems with the implementation of the segmented undecimated wavelet transform, is that the filter lengths can become longer than the input frame. It was initially proposed that this might be a limitation of the algorithm but this can be solved by implementing the convolution as an FIR filter and allowing the filter states to be $\left(2^{(j-1)}\right)(m-1)$ samples long. In the case that a decomposition filter is longer than an input frame, there will be a delay introduced which is larger than the length of the input frame. This results in an overall group delay and the output latency to be greater than a single frame. However, the filter states would keep track of the summation of the input samples and filter coefficients, and the final output of the decomposition is available once enough input samples have been processed by all the wavelet coefficients. The segmented undecimated wavelet transform presented here, handles the case where the wavelet coefficients become larger in length than the input frame. The level at which you can decompose the signal to is therefore not limited by this factor, but rather governed by the delay introduced at the largest decomposition depth.

5.3.3 Inverse Transform

The process of the undecimated wavelet transform is reversible. Reconstruction filters are applied in the same manner as they are for the inverse DWT, with a small modification of grouping the undecimated wavelet coefficients into subsets and performing the inverse transform on each of these and averaging the results. The deconstructin (D) and reconstruction (R) filter coefficients (Highpass (Hi) and Lowpass (Lo)) for the 'Daubechies 2' wavelet are presented in Table 5.1.

$D_Lo =$	-0.1294	0.2241	0.8365	0.4830
$R_Lo =$	0.4830	0.8365	0.2241	-0.1294
$D_Hi =$	-0.4830	0.8365	-0.2241	-0.1294
$R_Hi =$	-0.1294	-0.2241	0.8365	-0.4830

TABLE 5.1: Comparison of Daubechies 2 Deconstruction and Reconstruction Wavelet Filter Coefficients

The inverse of the time-invariant undecimated wavelet transform, does not insert 'holes' or zeros in-between the filter coefficients at each stage as is done in the forward transform. This process doubles the size of the filter coefficients at each decomposition level, rather than downsampling the results as is done with the DWT. The process is reversed by applying the reconstruction coefficients to subsets of the resulting approximation and detail coefficients and averaging the results.

In [256] transient impact sounds are analysed and synthesised by use of the inverse Stationary Wavelet Transform (iSWT) by Nelson et al [20]. An equation for the iSWT is given in [256] as 5.16:

$$\bar{s}_k = \text{ ISWT} \begin{cases} CD_k^1 & = \sum_{j=1}^{r} u_{(j,k)}^{CD^1} \bar{\Phi}_j^{CD^1} \\ \vdots & = \vdots \\ CD_k^L & = \sum_{j=1}^{r} u_{(j,k)}^{CD^L} \bar{\Phi}_j^{CD^L} \\ CA_k^L & = \sum_{j=1}^{r} u_{(j,k)}^{CA^L} \bar{\Phi}_j^{CA^L} \end{cases} \tag{5.16}$$

The application of the reconstruction filters is the same to that of the inverse DWT shown in Figure 2.31, where the subsets of samples are still upsampled by one at every even sample before the reconstruction filters are applied, and the reconstructed lowpass and highpass outputs summed together, and stored as the approximation coefficients as the input at the next level j_{n-1}. The difference in application of the inverse stationary wavelet transform, is to apply the reconstruction filters to subsets of the input coefficients and averaging the results. The number of subsets changes for each reconstruction level by $2^{(j-1)}$. This is done recursively, starting from level j back up to to level 1. At the final level of the inverse transform the step size $step_n = 1$, resulting in no subsets and the reconstruction filters being applied to all the coefficients at this stage.

The number of subsets and step size doubles at each level j_n and is given by:

$$step_n = 2^{(j-1)} \tag{5.17}$$

The main loops of the iSWT are outlined below in 5.3.3.

Algorithm 1 iSWT

```
1: procedure iSWT(swa, swd, R_Lo, R_Hi)
2:     a = a(size(a, 1),:)
3:     [n,nSamples] = size(swd)
4:     for j = n : −1 : 1 ... do
5:         step = 2^(j−1)
6:         last = step
7:         for first = 1, 2, ..., last do
8:             inds = first:step:nSamples
9:             lon = length(inds)
10:            subinds = inds(1 : 2 : lon)
11:            x1 = iDWT(swa(subinds),swd(j, subinds),R_Lo,R_Hi,lon,0)
12:            subinds = inds(2 : 2 : lon)
13:            x2 = iDWT(swa(subinds),swd(j, subinds),R_Lo,R_Hi,lon,−1)
14:            swa(inds) = 0.5 ∗ (x1 + x2)
15:        end for
16:    end for
17: end procedure
```

The input variables into the procedure iSWT() are swa, swd, R_{Lo} and R_{Hi}. The inverse transform iterates over the approximation (swa) and detail (swd) coefficients from the forward transform, applying the reconstruction filters; R_{Lo} to the approximation coefficients, and R_{Hi} to the detail coefficients, starting at the last level $j = n$ back up to level 1 for reconstruction. The variables n and $nSamples$ are the dimensions of the forward transform coefficient matrices, where n is the number of levels in decomposed from the forward transform, and $nSamples$ is the length of the audio frame.

The iDWT() procedure here, is the same as the typical inverse DWT given by 2.10.1, with the exception that the $x2$ calculations are circular shifted by one sample to align them in the correct position for the averaging of the results with $x1$. This is denoted by setting the last parameter in the iDWT method to -1. The reconstruction filter coefficients R_{Lo} and R_{Hi} are applied to the approximation ($swa(subinds)$) and detail ($swd(j, subinds)$) coefficients in the same way as the DWT, but this process is applied to multiple subsets of samples from the forward transforms output coefficients. This is a recursive operation, where the output of the iDWT is stored in the approximation coefficients (swa), becoming the input at the next reconstruction level $j - 1$.

At the beginning of the iSWT procedure, the approximation coefficients are set to the coefficients from the last level of the forward transforms decomposition; at line 2 of Algorithm 5.3.3: $a = a(\text{size}(a, 1),:)$. The rest of the approximation coefficients from preceding levels are discarded as presented in Section 2.10.1.

5.3.3.1 Segmented Undecimated Wavelet Transform

The segmented undecimated wavelet transform (SegUWT) is implemented in the same manner as the SWT and iSWT transforms. However, due to the segmentation of the input signal and block end border effects resulting from the convolution operations, consecutive input segments require an overlap of samples at frame boundaries to eliminate these artifacts. This is similar to the SegWT, however the number of coefficients stays the same at each decomposition level and the number of filter coefficients doubles. This results in each decomposition level requiring a greater number of overlapping samples between segments.

The number of sample which need to be overlapped at each level j is given by:

$$\text{overlap}_{\text{extensions}} = 2^{\text{j}-1} * \text{nFilter} \tag{5.18}$$

The number of iterations of the recursive sub-sampling with the two arrays of sub-indexes for the $x1$ and $x2$ calculations is equal to the number of iterations multiplied by 4 to account for $x1$ and $x2$ approximation and detail calculations. This is given by:

$$\text{numIterations} = (2^{\text{j}} - 1) * 4 \tag{5.19}$$

The matrix of convolution states required to continue calculations at frame boundaries using the FIR implementation is therefore given by:

$$\text{states} = [\text{numIterations}][\text{nFilter}] \tag{5.20}$$

For each calculation of $x1$ and $x2$, the number of OLS samples required for continuing the convolution sum on the next frame is given by:

$$\text{nStates}_{\text{x1}} = \text{nFilter} - 2 \tag{5.21}$$

$$\text{nStates}_{\text{x2}} = \text{nFilter} - 1 \tag{5.22}$$

The length of the arrays of samples required to be stored at the end of each calculation of $x1$ and $x2$, for the OLS implementation is given by:

$$\text{overlapSave}_{\text{x1}} = [\text{ij}][\text{nStates}_{\text{x1}}] \tag{5.23}$$

$$\text{overlapSave}_{\text{x2}} = [\text{ij}][\text{nStates}_{\text{x2}}] \tag{5.24}$$

where ij is the number of sub-sampling calculations over j levels and is given by:

$$\text{ij} = (2^{\text{j}}) - 1 \tag{5.25}$$

The above implementation of the SegUWT has been implemented in Matlab [16] and is compared with the non-segmented version provided by Matlab in the following Section 5.3.4.

5.3.4 Testing and Results

The generic segmented implementation of the iSegUWT has been tested using multiple frame sizes of 32, 64, 128, 256, 512, and 1024 samples. The decomposition depths have been tested for levels $j = 1$, 2, 3 and 4. The generic solution is able to handle all of these scenarios using different wavelets having different filter lengths. The results from the iSegUWT in the following tests, show a slight difference at the beginning of the first frame with the implementation of the iSWT in Matlab [16]. This is due to our implementation not padding the initial frame with zeros as is done with the iSWT in Matlab.

The rest of the signals output when implementing the above OLS scheme shows accurate results compared to applying the iSWT in a segmented framework and not handling convolution based errors at frame boundaries. Padding the initial frame with zeros, as is done in the SegWT implementation would resolve these differences at the cost of some minor additional latency.

The ability of using the undecimated shift-invariant implementation of the DWT provides an interesting opportunity to explore different audio effects which can be applied to the approximation and details coefficients returned by the decomposition. Thresholding of coefficients can be applied to explore dynamic equalisation and filtering. The SegUWT is also an interesting option for transient detection, and the popular use of it in de-noising techniques provides an method for separating transient components from noise.

Figures 5.28 and 5.29 show the SWT deconstruction using Debauchies 4 Wavelet up to Level 4, of a simple sinusoidal signal with a discontinuity in frequency, and a kick and bass line.

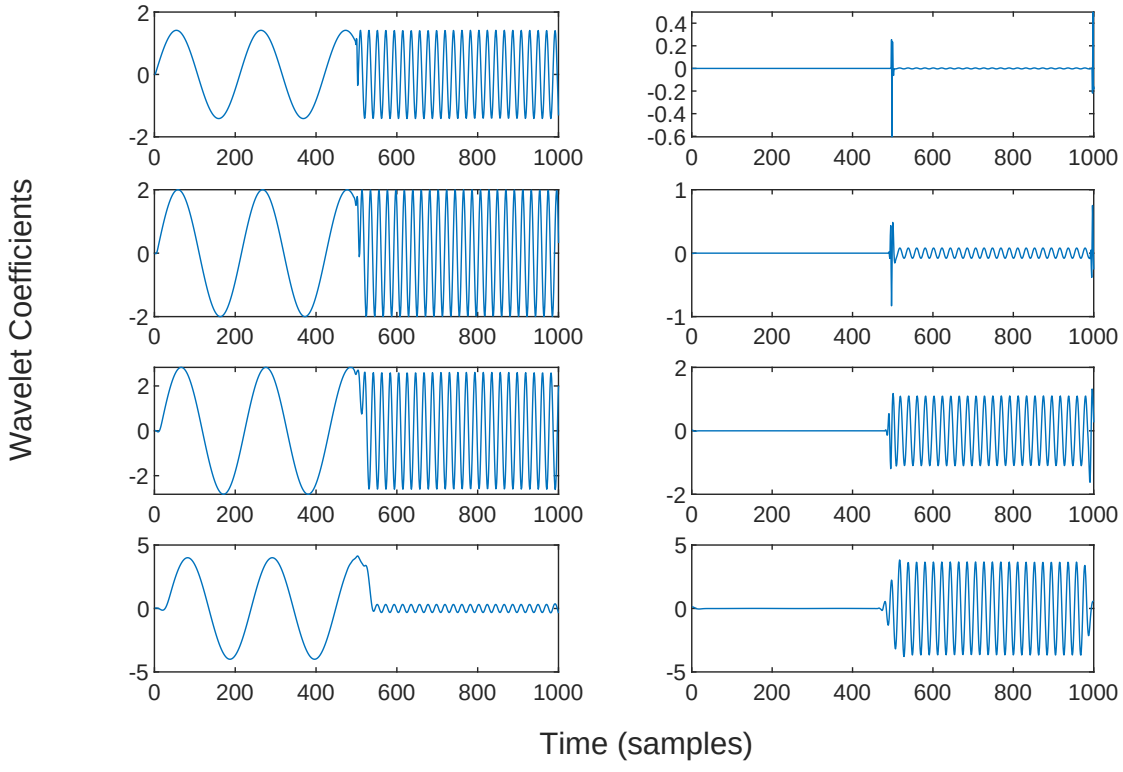


FIGURE 5.28: SWT of Sinusoid (@48 kHz) with frequency discontinuity, up to level 4 using dB4 wavelet

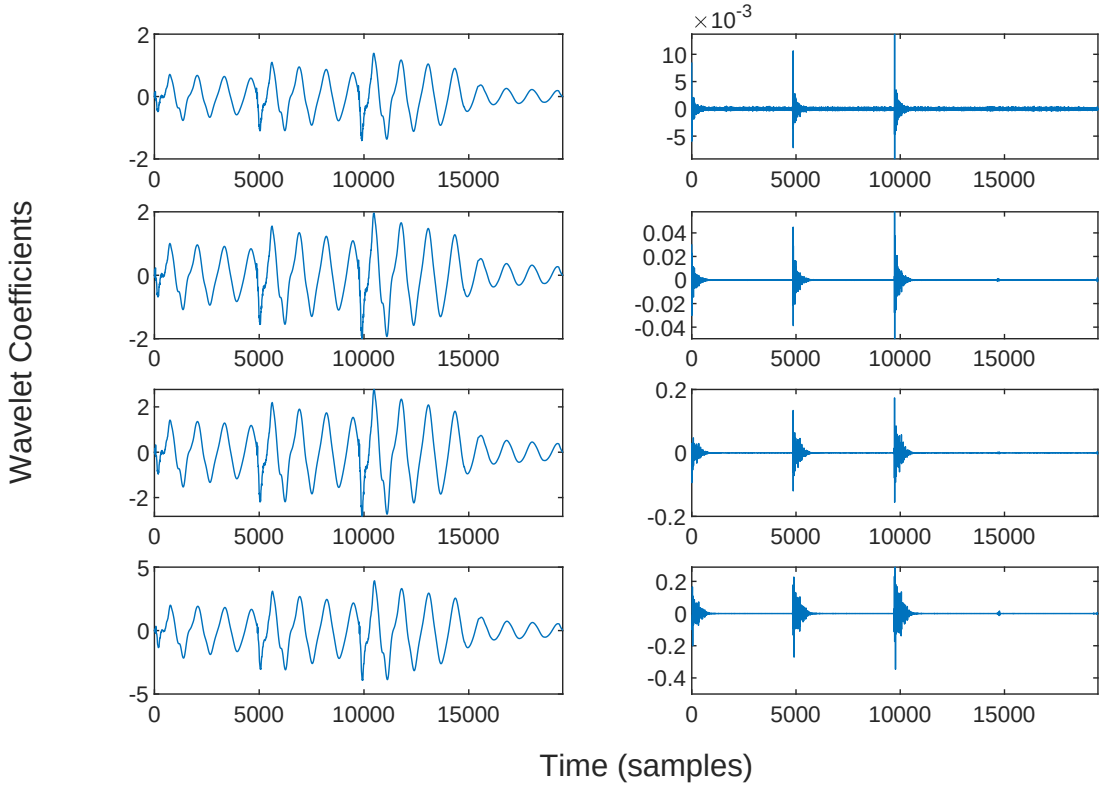


FIGURE 5.29: SWT of Kick and Bass (@48 kHz) up to level 4 using dB4 wavelet

Figures 5.30 shows the output of the iSWT applied to the entire signal in comparison to the output of the iSWT applied in a segmented framework with frames of 128 samples. Figure 5.31 shows the resulting residual signal of the iSWT (segmented) without handling convolution errors at frame boundaries.

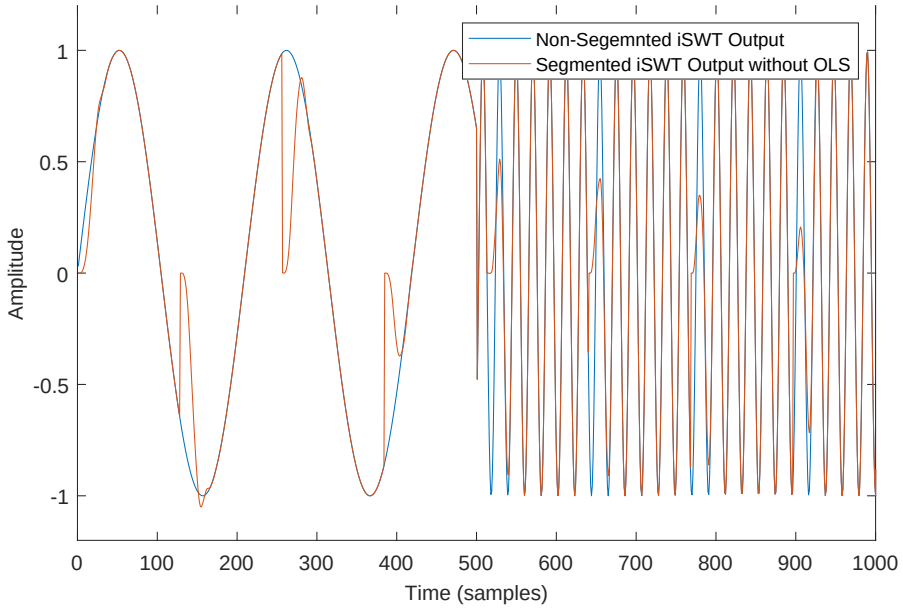


FIGURE 5.30: Segmented iSWT of sinusoid (@48 kHz) with frequency discontinuity, up to level 4 using the dB4 wavelet without OLS

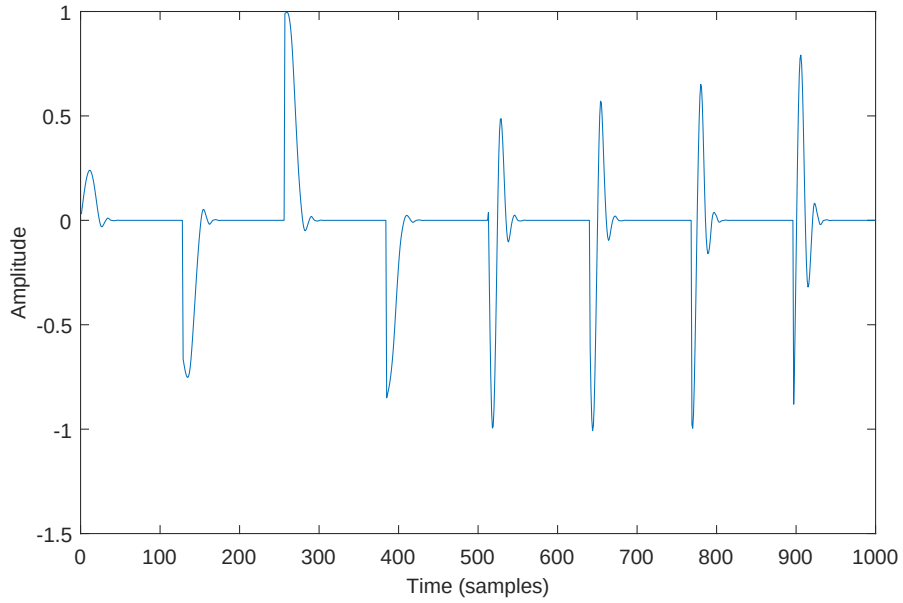


FIGURE 5.31: Residual signal from segmented iSWT of sinusoid, up to level 4 using the dB4 wavelet without OLS

Figure 5.32 compares the output of the iSegUWT with the results of the non-segmented iSWT. This shows a small difference at the start of the frame due to the different implementations, but remaining differences have been resolved. Figure 5.33 shows the resulting residual signal.

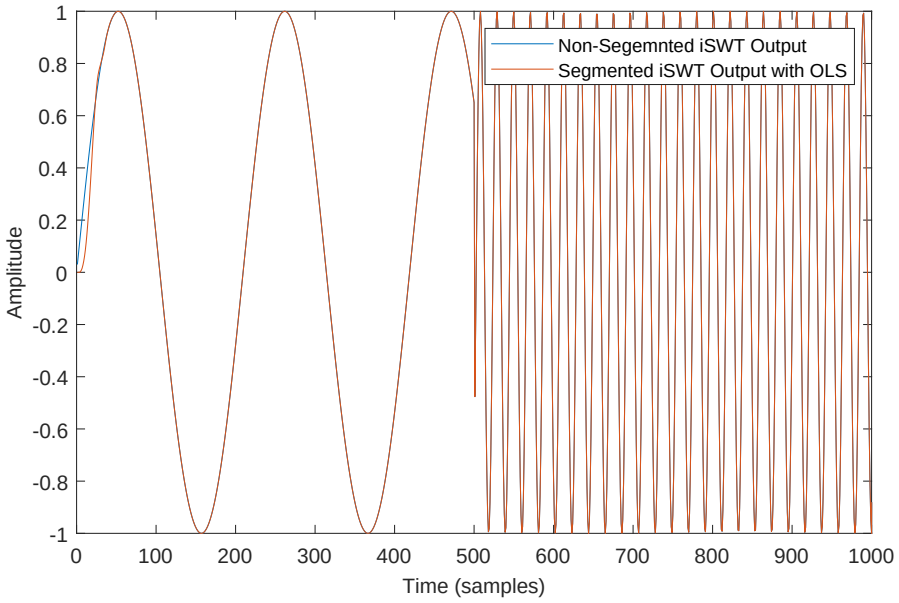


FIGURE 5.32: First couple of frames of iSegUWT of sinusoid with frequency discontinuity, up to level 4 using dB4 wavelet with OLS

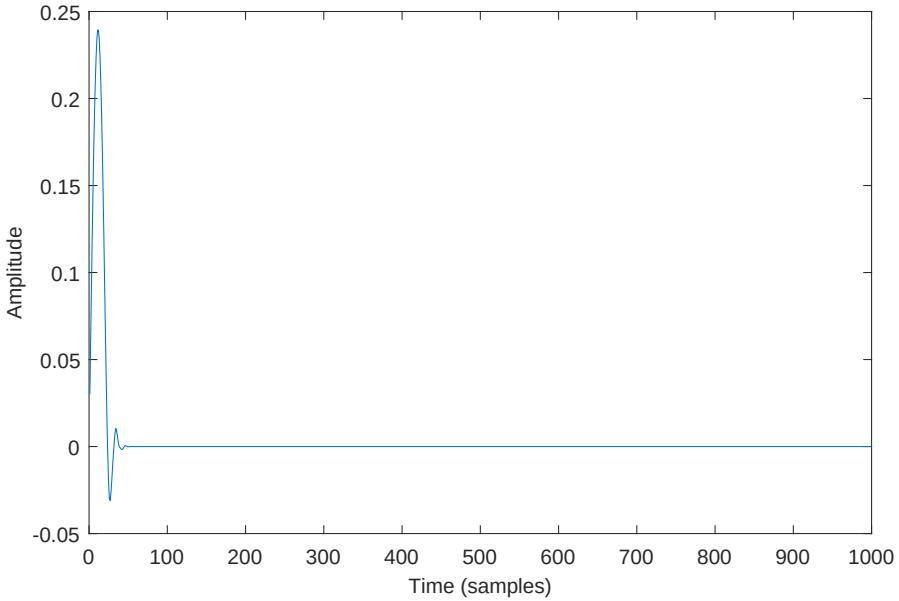


FIGURE 5.33: Residual signal from the output of the iSegUWT displays an error at the beginning of the first frame due to the absence of prepending the input signal with zeros.

Figures 5.34 and 5.35 show output of the iSWT (segmented) from the first 2048 samples of a kick and bass line segmented into 16 frames of 128 samples, and resulting residual signal without handling convolution errors at frame boundaries.

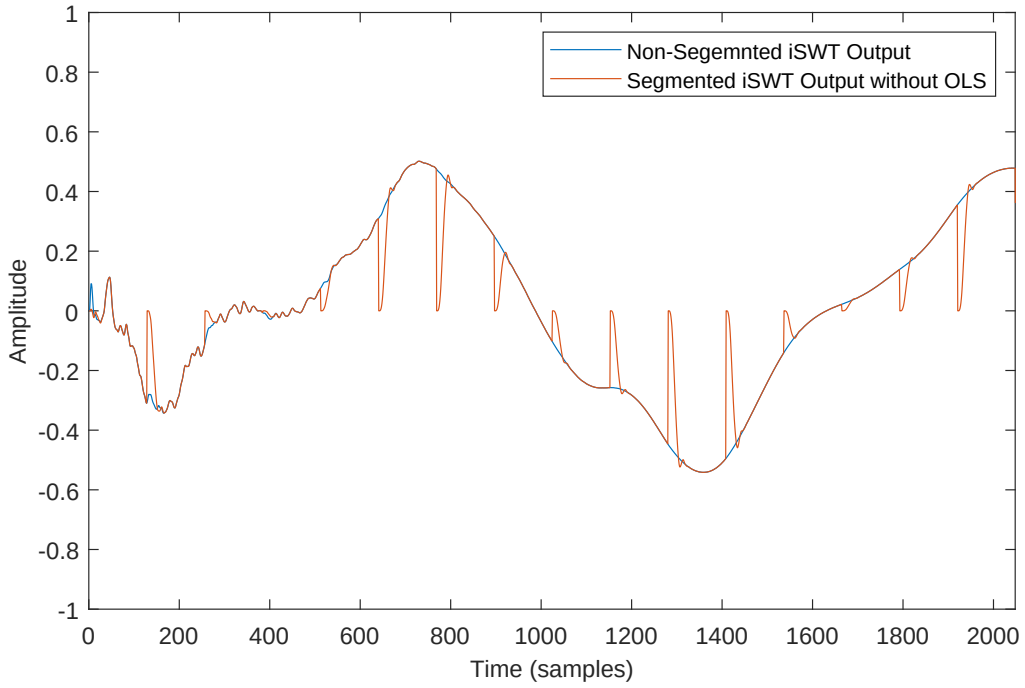


FIGURE 5.34: First couple of frames resulting from the segmented iSWT of a Kick and Bass clip, up to level 4 using dB4 wavelet without dealing with border artifacts

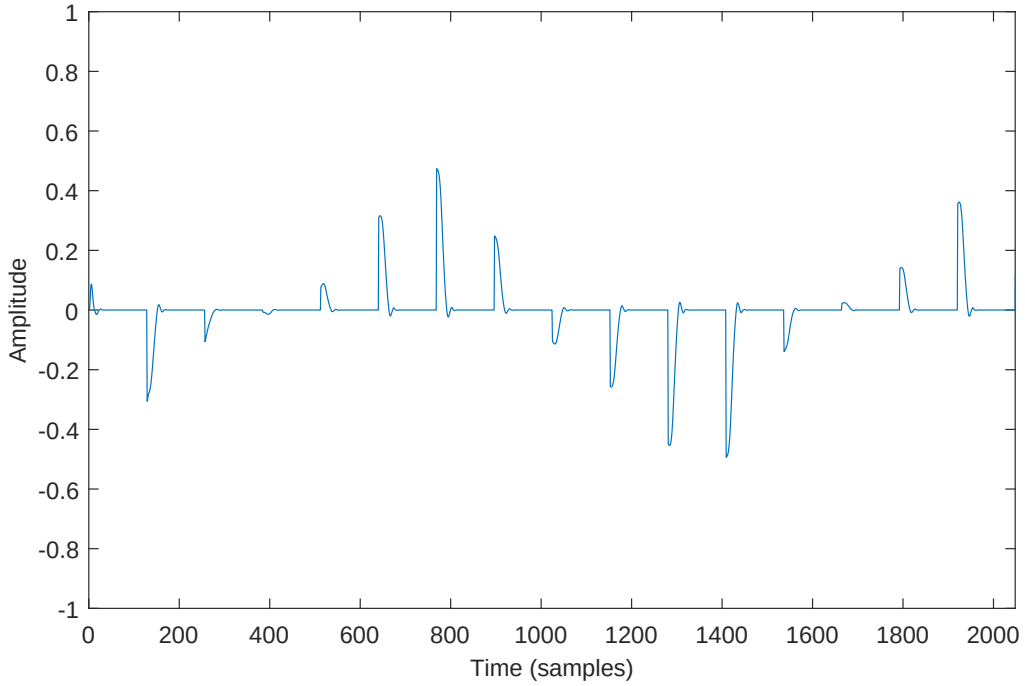


FIGURE 5.35: First couple of frames of residual signal resulting from the segmented iSWT of Kick and Bass clip, up to level 4 using dB4 wavelet without OLS

Figures 5.36 and 5.37 show the output of the SegUWT and the resulting residual signal, using OLS to overcome the convolution block end artifacts. There is a small difference at the start of the frame due to the different implementations, but the remaining differences have been resolved.

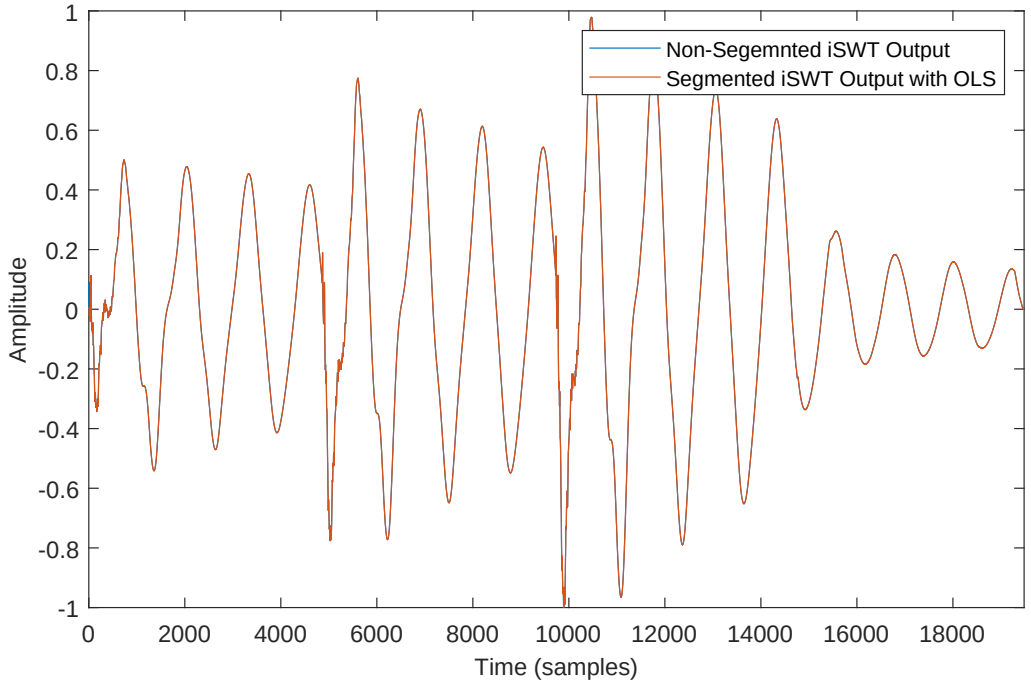


FIGURE 5.36: First couple of frames of iSegUWT of Kick and Bass (@48 kHz) up to level 4 using dB4 wavelet

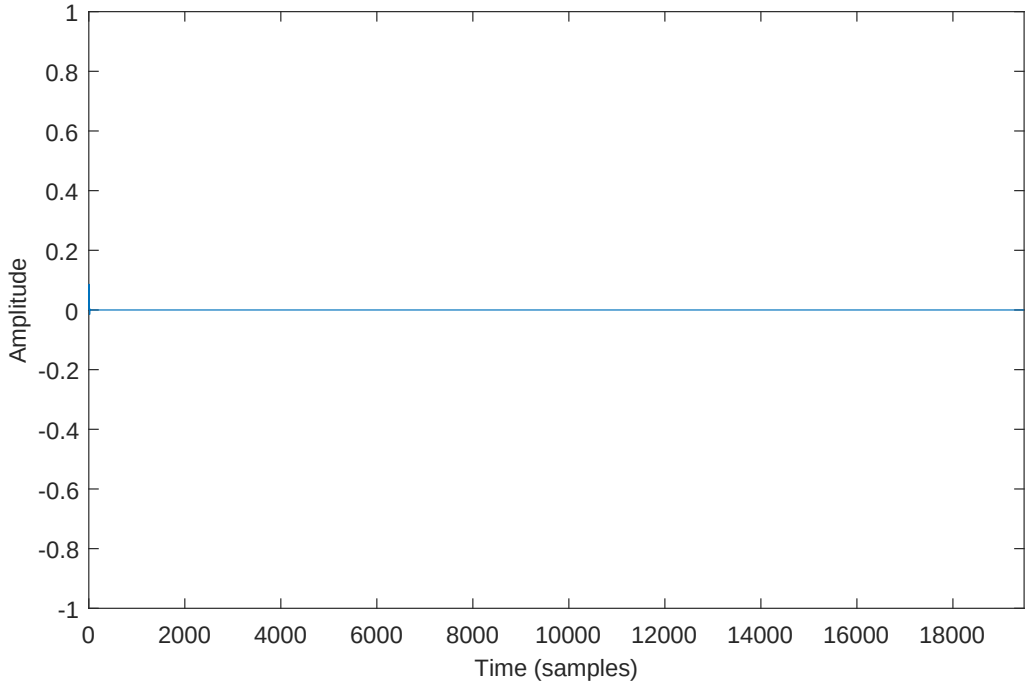


FIGURE 5.37: Residual signal from iSegUWT of Kick and Bass up to level 4 using dB4 wavelet

5.4 Transient separation from Residual

This section presents a brief investigation of using known de-noising techniques utilising the Undecimated Wavelet Transform. The examples use the Matlab [16] Wavthresh utility, to experiment with a residual signal, obtained from applying MoP to a EDM track. Details of the signal and residual can be found in Section B.2.10.

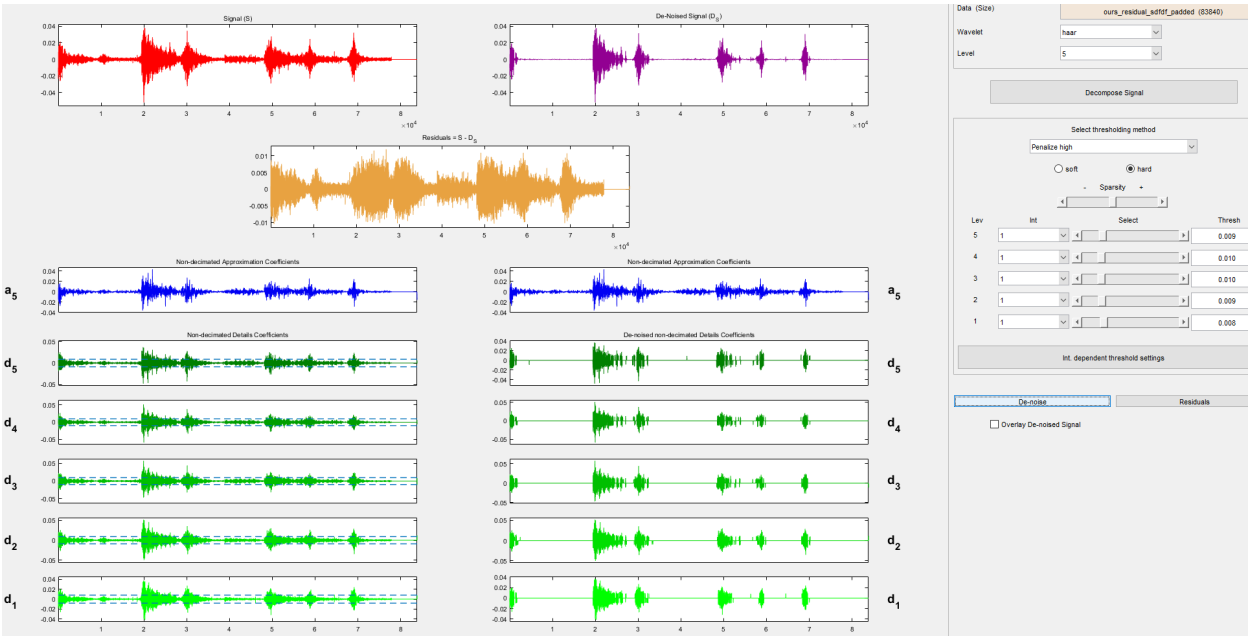


FIGURE 5.38: Wavthresh Setting for De-noising Residual and extracting Transients using Haar Wavelet

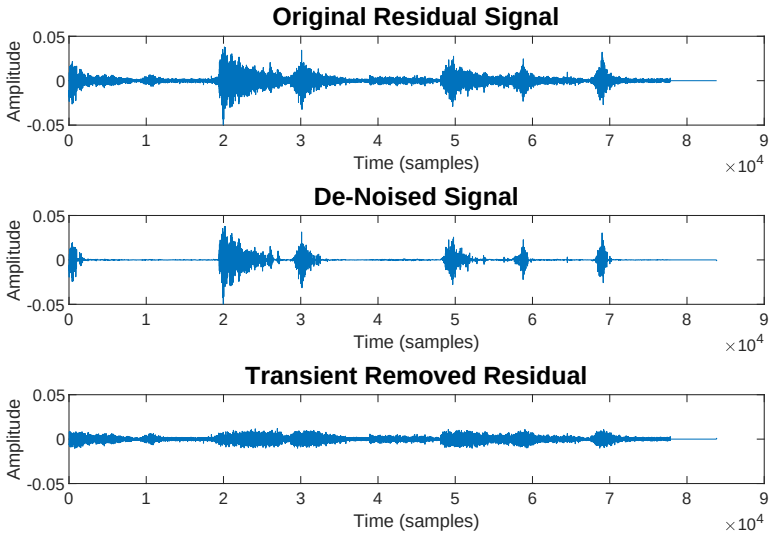


FIGURE 5.39: Extraction of Transient via De-Noising using the Segmented SWT

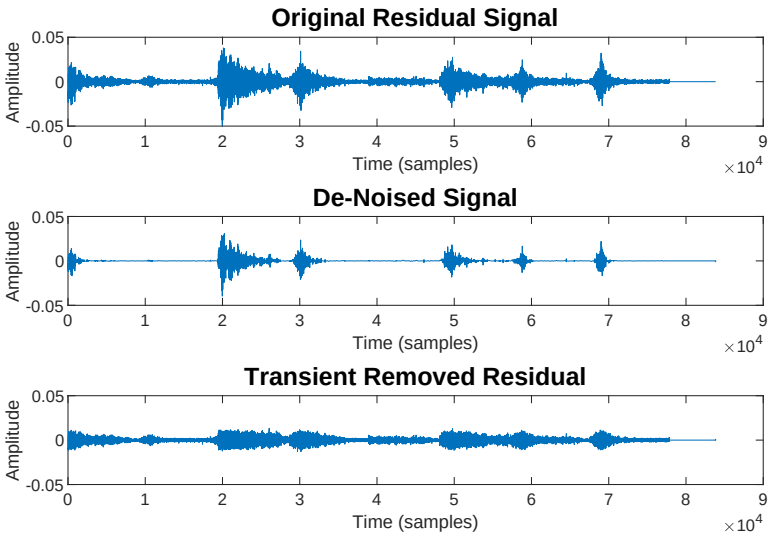


FIGURE 5.40: Extraction of Transient via De-Noising using the Segmented SWT Bior

Figures 5.39, 5.40 and 5.41 clearly show transient components which have successfully been separated from noise during reconstruction.

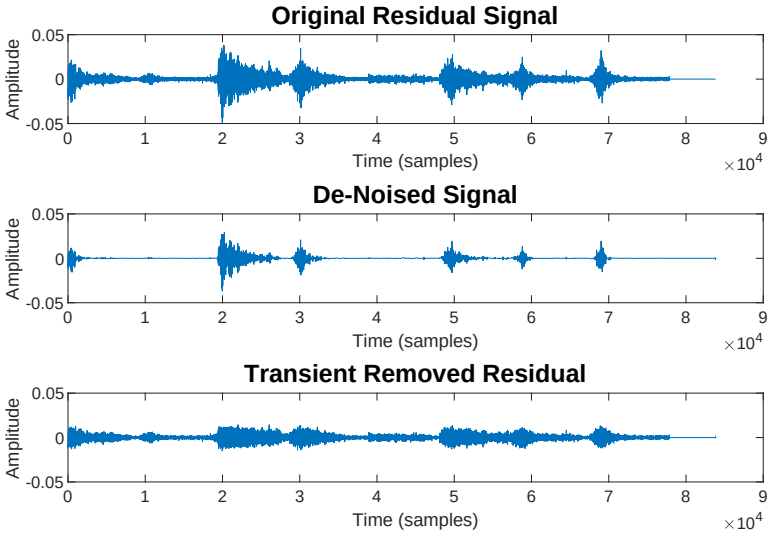


FIGURE 5.41: Extraction of Transient via De-Noising using the Segmented SWT Debuchies 7

Figures 5.39 shows a clearer separation from Figures 5.40 and 5.41. The Haar wavelet in this example perfroms better than the Bior and Debuchies 7 Wavelets, but thresholding coefficinets also contribute the quality of the seperation. De-noising techniques [19, 239, 240, 253], which are well established with the undecimated SWT, require further investigation for achieving optimal separation.

5.5 Conclusion

An approach of modelling the residual signal using the SegUWT has been discussed. A generic solution for implementing the shift-invariant Undecimated Wavelet Transform in a segmented frame by frame system, for both the forward and inverse operations has been presented. It has been shown to resolve block based errors occurring at frame boundaries due to the frame based implementation required for real-time processing. Calculations for the delays introduced depend on the decomposition depth j and length of the Wavelet filter coefficients.

Residual modelling using the SegUWT has been presente, transient components are shown to be separable from noise via known de-noising techniques, but further research is required for achieving optimal separation.

Transient modelling and specifically, how transients are handled within a MoP decomposition has been discussed. Transients might possibly need to be detected and removed before the spectral modelling stage due to the issues introduced by non-monotonic amplitude change in the sinusoidal model, but this is left as a future research topic. The ability of a causal implementation of MoP to accurately model non-monotonic amplitude change using a rectangular window has been presented. This technique is able to capture frequency changes and non-monotonic amplitude change, using an overcomplete sinusoidal representation.

The following Chapter 6 presents an overview of the system, incorporating details presented from Chapters 3, 4 and 5. Linear and Exponential amplitude change estimation methods are examined, and methods for modelling monotonic and non-monotonic amplitude change discussed. The atomic decomposition of non-monotonic amplitude change is examined in detail. MoP is compared with other known single frame non-stationary sinusoidal estimation methods, and applied to fully produced EDM tracks. Finally pitch and time shifting effects are examined using this overcomplete model.

Chapter 6

System Overview

Perfection is not attainable, but
if we chase perfection we can catch excellence.

Vince Lombardi (1959) [257]

6.1 Introduction

The previous three chapters of this thesis have described new techniques for using Fourier and wavelet analysis to decompose audio signals; with an emphasis on kick and bass sounds; into non-stationary sinusoidal, residual and transient models. This chapter puts these techniques into a practical context, describing a single frame system which uses the methods described to decompose the sound into these three models. These techniques have the potential of being combined in numerous ways for modelling and manipulating kick and bass sounds, with the intent of maintaining the quality of the original sound. The non-stationary sinusoidal model uses the parameter estimation and decomposition methods described in Chapters 3 and 4, to approximate monotonic sinusoidal components. Transients and noise are modelled using the undecimated Wavelet Transform (UWT) as described in Chapter 5.

Two separate single frame systems are introduced in Sections 6.3.1 and 6.3.2. A segmented system is initially presented and the challenges en-counted regarding a real-time implementation compared to an offline rendering system discussed. A single frame system performing the analysis and re-synthesis on short audio samples, rather than a steam of audio is then presented.

The non-stationary sinusoidal model can use either the causal or non-causal estimation methods discussed in Chapters 3 and 4, however the current method derived for amplitude curve discrimination uses a non-causal implementation which requires an odd frame size, while the inverse UWT requires an even frame size which is divisible by 2^{level}. A mismatch in audio frame sizes in the context of a real-time segmented audio system and the selection of causal or non-causal methods are discussed in Section 6.3.1.1.

The sinusoidal model can also be adapted to include an over-complete atomic decomposition, containing non-monotonic components, and components with high amplitude and frequency changes which overlap in the frequency domain, within the decomposition. However, Chapter 4 presented an issue with performing time and pitch scale modifications on an over-complete decomposition of non-monotonic sounds. Another current limitation of the MoP methods described is the use of a dictionary composed of atoms with a fixed basis size. This is not suitable for modelling components which do not encompass the entire signal space.

The non-stationary sinusoidal model used in Section 6.3.1 approximates each sinusoidal component with a single best fitting atom derived from MoP, for each peak in the frequency domain, leaving any remaining non-stationary sinusoidal components overlapping an extracted peak in the frequency domain within the residual signal. In contract, the non-segmented system implementation presented in Section 6.3.2 uses an over-complete atomic decomposition. This method is not suited to modelling signals with non-monotonic amplitude change, and is targeted at modelling short percussive signals with very short attack times followed by a longer release, as discussed in Sections 4.5.6 and 4.5.11.

The residual signal for both methods is decomposed using the undecimated wavelet transform described in Chapter 5, from which transient components can be extracted using a de-noising process on the wavelet coefficients before performing the inverse transform. Chapter 4 presented modelling transient signals with multiple short-term sinusoidal atoms when the audio frame is adapted to size and positioned around a detected transient. Time and pitch scale modifications using an over-complete MoP decomposition on the transient signals tested work well. However, this requires a transient detection stage and a multi-scale FFT decomposition which is more suited to a non real-time implementation.

The following section presents a discussion regarding these topics, the challenges encountered and the research decisions made over the course of the thesis. A detailed analytical evaluation of the systems and the limitations are then presented.

6.2 Discussion

The modelling of transients as a sum of sinusoids is deemed inefficient in spectral modelling systems due to the large number (hundreds) of sinusoidal components required to accurately capture these short-lasting broadband components [51, 104, 107, 108, 111, 115, 212]. Modelling of transients using a large number of sinusoids is also viewed as an inappropriate representation which "does not offer possibilities for meaningful transformations" [225]. Traditional SMS methods which model the transient in the residual signal suffer from a loss of sharpness and pre-echo effects. Rodet et al [258] attribute the loss of sharpness to "the use of a finite length window in the spectral estimation", and the pre-echo resulting from the windowing of the signal which can reduce the volume of the transient and delay the temporal information due to the roll-off caused by the windowing effect. Time Frequency Reassignment can greatly improve the sharpness of transients and reduce the pre-echo effect [259], but multi-resolution methods involving synchronisation of analysis windows to transient events can improve this further by ensuring the length of the analysis window and timing is positioned around a transient [96, 148].

The modeling of transients has been presented in Section 2.6, and although modelling transients as a sum of sinusoidal components is not compact, the single frame estimation methods presented using MoP in Chapter 4 are able to model transient components accurately. Recent work on full-band quasi-harmonic analysis and synthesis of musical instrument sounds, as well as adaptive sinusoidal modelling of percussive instrument sounds [202, 207] has also been shown to accurately model sharp onsets and highly non-stationary attack transients using sinusoidal components.

Simply modelling and re-synthesising the original signal is not attractive to a musician or composer with an interest in manipulating and applying meaningful transformations to the resulting output of the re-synthesised sound. Adaptive sinusoidal modelling using eaQHM explores time and pitch scale modifications on speech in [227, 260, 261]. However, time and pitch scale modifications with regards to musical and percussive instruments is left as a future perspective where further research is required to include the application of sound transformations such as "timbral variations, perceptually coherent time stretching and pitch shifting" to these types of signals [202, 207].

Sections 4.5.8 and 4.5.5 highlight the issues of preforming time and pitch scale modifications on an over-complete representation of a single sinusoidal partial with non-monotonic amplitude and frequency changes. The constructive and destructive interference that the amplitudes, frequencies and phases of these modified atoms have on each other, no longer combine in the same manner, resulting in

an incorrect output signal. This is a limiting factor with regards to certain kick and bass signals containing non-monotonic amplitude change, but has less of an influence on modelling and performing pitch and time scale modifications on kick and bass sounds with very short attack times, and other broadband percussive signals with sharp transients, such as snares and hi-hats. MoP was shown to accurately model percussive sounds as an over complete dictionary of monotonic non-stationary sinusoidal components. The tests presented show that time stretching and pitch shifting can be applied to this model. Timbral variations of the sound are easily achieved by selectively excluding a number of atoms and/or applying pitch and time scale modifications to the selected atoms before additive re-synthesis. Modelling these sounds with a non-stationary sinusoidal MoP model requires the entire audio sample to be analysed with a single FFT as is done with the IRs in the original implementation of MoP [14].

The non-overlapping segmented audio framework used in Section 6.3.1 uses a fixed frame size specified at run-time. Frame sizes ranging from 512 to 2049 have been evaluated for finding a balance between latency and frequency resolution. A frame size of 512 samples introduces a delay of 10.7 ms while a frame size of 2049 introduces a delay of 42.7 ms. The frequency resolution of a 512 sample FFT with a sample rate of 48 kHz is only 93.75 Hz. Kick and bass sounds often contain frequencies below this requiring frame sizes of 1024 or 2048 samples to acquire frequency resolutions of 46.875 Hz and 23.4375 Hz respectively. In practice, a single bins resolution is not enough to estimate the non-stationary sinusoidal parameters from phase difference measures. A couple of frequency bins are required to detect a sinusoidal peak and measure the first or second order difference of the magnitude, or phase with respect to frequency, across the peak.

The application of a fixed audio frame size is attributed to a loss of sharpness in the resulting re-synthesised signal. This occurs when a partial does not encompass an entire audio frame, such as a sound starting or ending in the middle of an analysis frame. The basis functions of the DFT span the entire length of the audio frame which results in erroneous parameter estimates and a re-synthesised signal which is distorted in time. Figure 6.1 shows a simplified demonstration of this where a simple sine wave starting at the middle of a 512 analysis frame is represented by estimates taken from the DFT. The output signal spans the entire frame and the amount estimated for amplitude and amplitude change is incorrect.

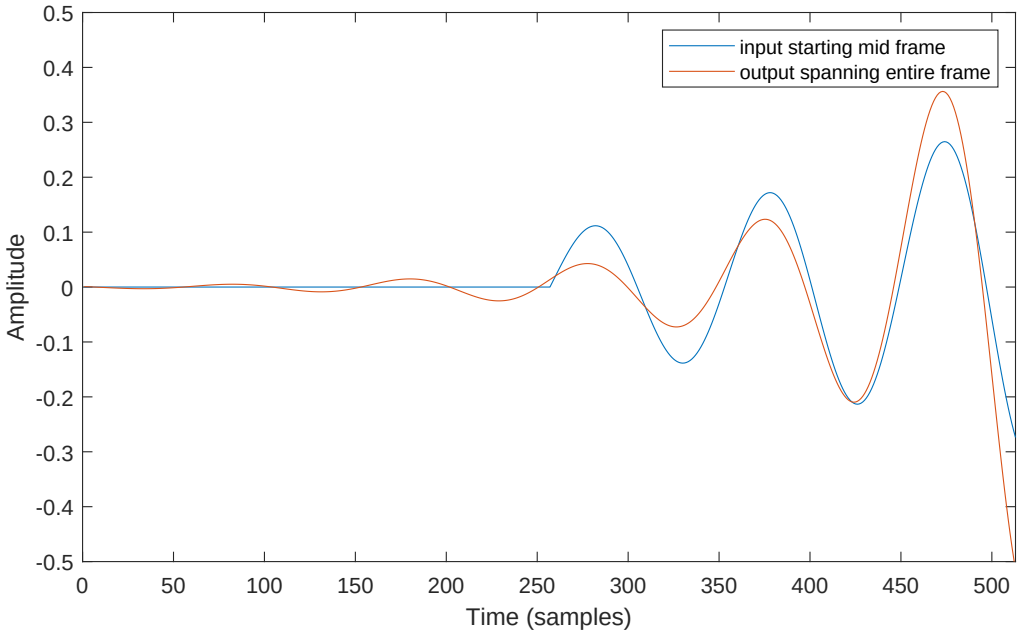


FIGURE 6.1: Comparison of 1 kHz sine wave (@48 kHz) starting mid frame, and resulting output starting at the beginning and spanning the entire frame.

A segmented real-time spectral modelling system which is able to adapt the frame size and alignment of analysis frames with onset and offset events can greatly improve the accuracy of the model. In the case of real-time kick and bass modelling, this could be achieved by the alignment of analysis frames with the audio onsets from temporal information such as the number of beats per minute (BPM) combined with onset/offset detection.

Masri et al [148, 164] provide a detailed investigation into detecting, modelling and improving the synthesis of transient attacks, which "play a vital role in our perception of timbre". The solution presented which aimed at incorporating attack transients directly into the model, synchronises the analysis and synthesis processes to the percussive note onsets. Transient detection was performed in the frequency domain due to its improved ability to detect energy and phase changes, compared to time-domain envelope based methods. The energy distribution, attack envelope and spectral dissimilarity methods for transient event detection are presented in [148], where a pre-analysis stage generates a list of attack onsets to be used in the analysis and synthesis stages later. A region list is created which then details the start and end samples of an attack region. The analysis and synthesis frames are then aligned to each region independently. The alignment of analysis and synthesis frames to onset and transient attack regions was shown to improve the output of the model, especially under time transformations, "where the timbral properties of the attacks are preserved to give the impression

of a more natural slowing of the sound". The authors conclude that ultimately, improvements are dependent on the ability to see more in the time-frequency domain.

Onset detection and the alignment of transient events with multi-resolution analysis and synthesis regions has been omitted from the current system due to the desired real-time aims. The uncertainty of the position of transient onsets and offsets within an analysis frame, where sinusoidal basis functions span the entire length of the audio frame, is one of the main limiting factors of the current system due to the biased parameter estimates produced when analysis frames are not aligned with transient events. The addition of a pre-analysis stage and alignment of analysis and synthesis regions is shown in [164] to incur a minimal impact to the computational expense of an existing non-aligned system, and is therefore left as a desirable future directive.

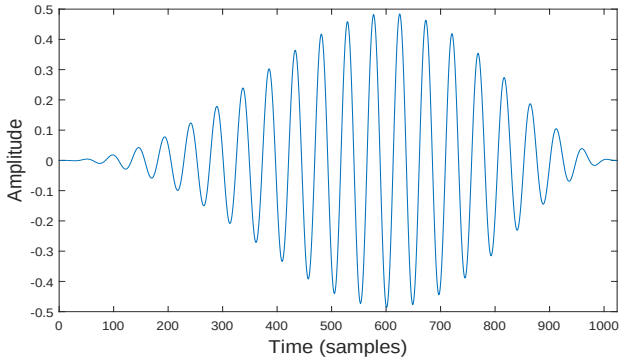
6.2.1 Modelling monotonic amplitude change

Chapter 3 presented a method for calculating linear amplitude change estimates from the first order difference of the phase across a spectral peak. Having presented a method for the discrimination between exponential and linear amplitude change, and providing a method for estimating the amount of monotonic linear amplitude change, the effectiveness of the discriminator and accuracy of the parameter estimates were assessed. The need for incorporating linear amplitude change into the model was assessed by presenting estimates of linear amplitude change from models which presume exponential amplitude change in Section 3.7. A comparison of modelling a sinusoid with a range of added Gaussian noise, amplitude, frequency, phase, and amplitude modulation values, for both linear and exponential amplitude change was also evaluated. Figure 3.20a displays the resulting Signal-to-Noise ratio (SNR) plot, highlighting that the results deteriorated significantly for signals containing amplitude change above a small amount.

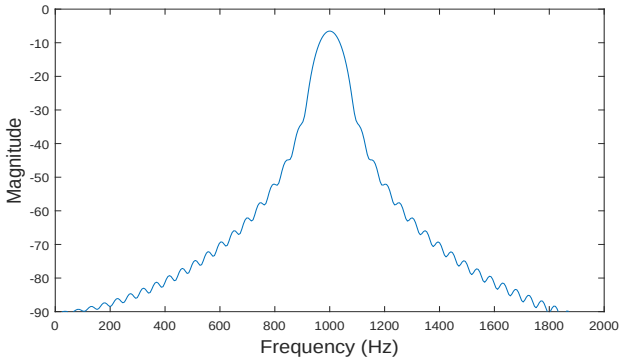
The analytical equations are derived from non-causal calculations, meaning the windowed audio frame has an odd length and is centered in time at zero. Zero-Phase windowing and padding is employed which results in a flat phase response across a stationary sinusoidal peak and parameter estimates calculated from the DFT being positioned at the center of the analysis frame in time. The non-causal analytical equation for producing the phase distortion measurements in the presence of linear amplitude change a in Nepers is given by 3.15. The non-causal analytical equation for producing the exponential phase distortion measurements from a given range of amplitude change values a in Nepers is given in [10] by 3.16. This is an analytical derivation of the amplitude modulation estimator from [176], which

is not exactly equivalent to the actual derivative as shown in Appendices C where a more detailed view of the difference and improvement of the accuracy of the results by zero-padding can be seen in Figures C.7 and C.5. Estimations of the change in amplitude are calculated from measuring the phase difference (first order derivative of the phase) across a speak in the DFT, and using that measurement to lookup the amount of amplitude change related to the measured amount of phase distortion, from a lookup table. A lookup table for estimating monotonic amplitude change is a simple, accurate and fast/efficient method of estimating monotonic amplitude change within a single analysis frame.

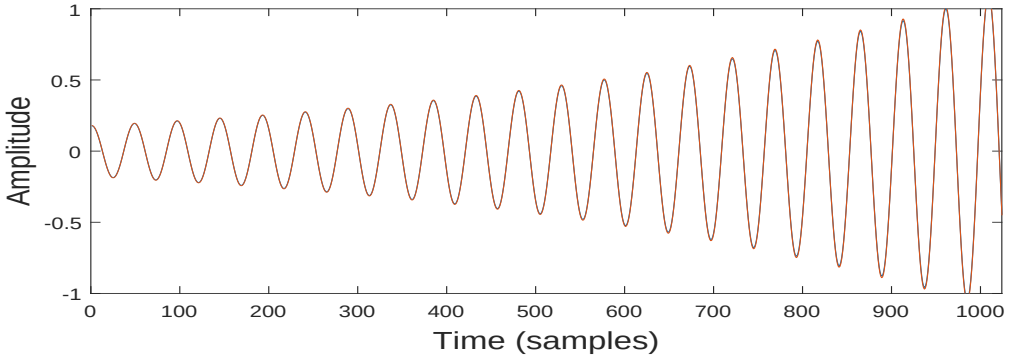
Quadratic interpolation (QIFFT) [134, 135] around spectral peaks is used for improving the accuracy of the amplitude and frequency parameter estimates as this information is readily available from the data returned from the single DFT. Many other methods such as quadratic interpolation using an exponential magnitude spectrum weighting [132], or triangular interpolation, could also be used. A comparison of a number of modified QIFFT methods are presented in [133]. A survey on some of these methods including parabolic and triangular interpolation, as well as reassignment and derivative algorithm are presented in [162].



(A) Windowed Sinusoid



(B) Magnitude spectrum of windowed sinusoid



(C) Original and output of a sinusoid synthesised from model parameter estimates

FIGURE 6.2: A 1 kHz sinusoid (@48 kHz) with 6 dB amplitude change modelled from information returned from a Hanning windowed zero-phase padded DFT

Figure 6.2a shows a windowed 1 kHz sinusoid with 6 dB amplitude and the resulting magnitude spectrum 6.2b. Figure 6.2c displays the output of the synthesised sinusoid using the parameter estimates described above.

6.2.2 Modelling Linear Frequency change

Linear FM is often adopted in spectral models because it simplifies the mathematical operations used by the model for estimating the rate of frequency change. If the model is phase based, the change in phase between frames or the mathematical operations involved in calculating the derivative of the phase are simplified if linear frequency change is assumed. Exponential frequency modulation can be modeled from the examination of the phase in the frequency domain. However larger frequency changes result in the relationship between phase and frequency to become nonlinear [262], and more advanced and computationally expensive techniques are required.

6.2.2.1 Estimating Linear Frequency Change from Phase Distortion

Chapter 3 presented methods for estimating both linear and exponential amplitude change from the slope of the phase across a sinusoidal peak in the DFT spectrum. Differentiating between linear and exponential amplitude change from examination of the magnitude second order difference in relation to the first order difference of the phase with respect to frequency, has also been presented. For simplicity, these methods assumed the absence of frequency change within the model. Figures 2.26 and 2.27 show the effect of amplitude change on the phase spectrum. A positive change in amplitude results in a negative phase slope, while a negative amplitude change results in a positive phase slope [164]. Figures 2.24 and 2.25 display the effect of frequency change to the phase across a sinusoidal peak, which becomes concaved or convexed across the peak. Positive frequency change has an upward convex curve while negative frequency change has a downward concave curve.

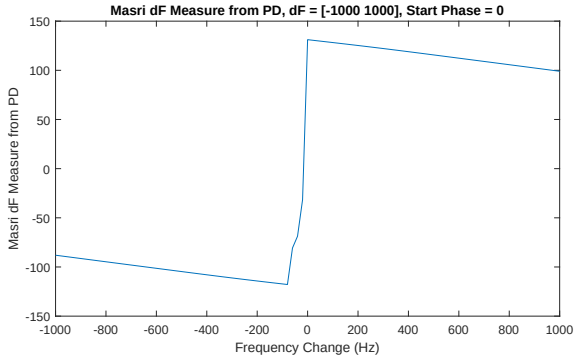
The independent estimation of amplitude change and frequency change from phase distortion is shown to be separable in [164], based on the fact that there is a relationship between frequency change and the difference in phase across the peak combined with a relationship between amplitude change and the combined differences between the phase either side of a peak. Reassignment information and phase distortion analysis is combined in [9], where PDA is extended from a first order polynomial approximation, to a second order polynomial, to estimate amplitude and frequency changes from time

reassignment information instead of phase. PDA estimates of amplitude and frequency change are not robust for large changes in either amplitude or frequency, this is compounded when both are present. RDA aims at improving the estimates of amplitude and frequency change by the use of an iterative (2D) array lookup method.

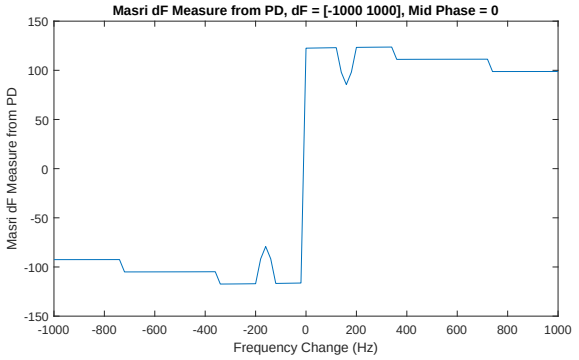
Frequency change was omitted from the linear and exponential amplitude curve estimation methods described in Chapter 3 for simplicity, and this omission is a necessity for solving the amplitude change estimation methods presented. Real-world signals however do not only contain amplitude non-stationarities, but frequency change as well. A non-stationary sinusoidal model therefore needs to incorporate both amplitude and frequency change estimates within the model, and should be robust to biases introduced in the presence of both.

In this thesis, the examination of the use of phase information for the estimation of frequency change, using both the first and second order differences of the phase with respect to frequency, were investigated. Raw phase data from the DFT, unwrapped phase data and negatively wrapped phase data; which uniquely unwraps the phase in a way that ensures that it is always monotonically decreasing with increasing frequency (prevents positive phase differences giving rise to exponentially increasing components and/or large errors due to extrapolation outside of the PD lookup table); have been investigated for estimating frequency change from phase information. Is was found that unlike estimating amplitude change with the first order difference of the phase with respect to frequency, this measurement was not as robust to estimating frequency change from the phase information.

The first order phase difference measure from 2.9.1 was evaluated for estimating linear frequency change. The phase measurement results using 2.9.1 are susceptible to starting phase conditions, and are not equal when compared to measures taken with the phase set to 0 at the start of a frame, and at the middle of the frame. Figure 6.3 shows the PD measure in 2.9.1 for estimating linear frequency change, with the phase of the input signal set to zero at the start of the frame and compared to an input signal with the phase set to zero at the middle of the frame. The measure in Figure 6.3a does not display any modulations within this frequency range, however Figure 6.3b does show some differences and the start of some modulations within the measure, with the phase set to zero at the middle of the frame. These measures are taken using an analysis frame size of 1025 samples, however, bass signals may require an analysis frame size of 2049 samples for detecting low frequencies as mentioned above. Figure 6.4 shows the PD measures with an analysis frame size of 2049 samples. The measure becomes more susceptible to modulations with larger framer sizes as shown in Figure 6.4b.

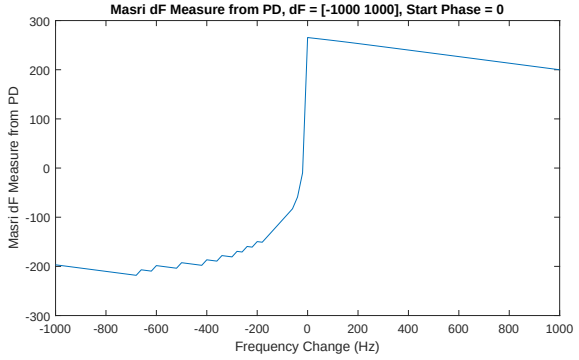


(A) Masri Phase First Order Difference [-1000 to 1000 Hz], Start Phase = 0

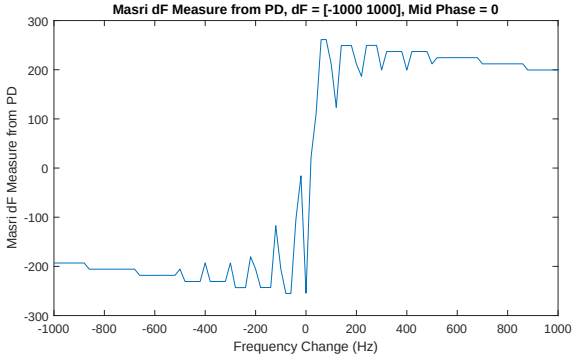


(B) Masri Phase First Order Difference [-1000 to 1000 Hz], Mid Phase = 0

FIGURE 6.3: Masri Phase First Order Difference, A) Start Phase, B) Mid Phase = 0



(A) Masri Phase First Order Difference [-1000 to 1000 Hz], Start Phase = 0

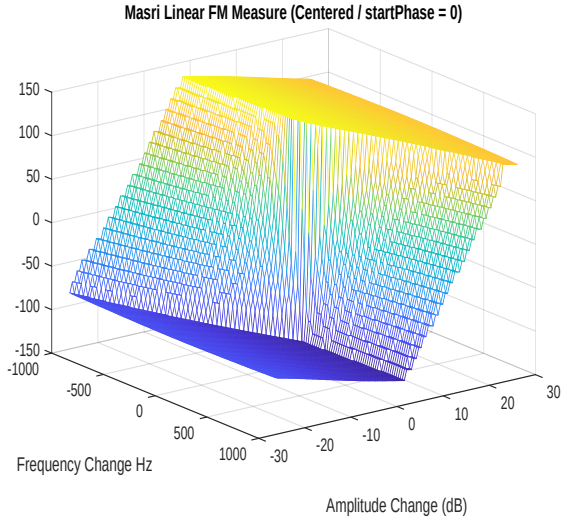


(B) Masri Phase First Order Difference [-1000 to 1000 Hz], Mid Phase = 0

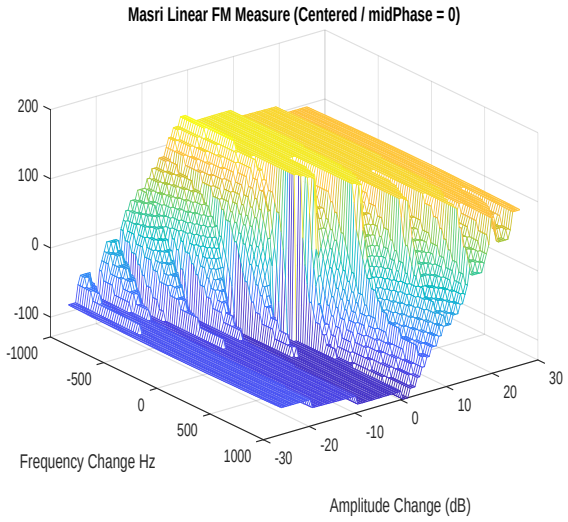
FIGURE 6.4: Masri Phase First Order Difference, A) Start Phase, B) Mid Phase = 0

The modulations within the measure used in 2.9.1 for linear frequency change (ΔF), become more apparent in the presence of both frequency and amplitude (ΔA) modulations. Figure 6.5 compares the PD measure with different initial phase conditions, a range of ΔF and ΔA values, and with a frame size of $N = 1025$.

Figure 6.5 compares the PD measure with the same conditions as above, but with a frame size of $N = 2049$ samples. The initial plots in Figures 6.5a and 6.6a appear quite smooth with the phase set to zero at the start of the frame. However, the result of the measure becomes noticeably worse in Figures 6.5b and 6.6b where the phase is set to zero at the middle of the frame, and the analysis frame increases from 1025 to 2049 samples .

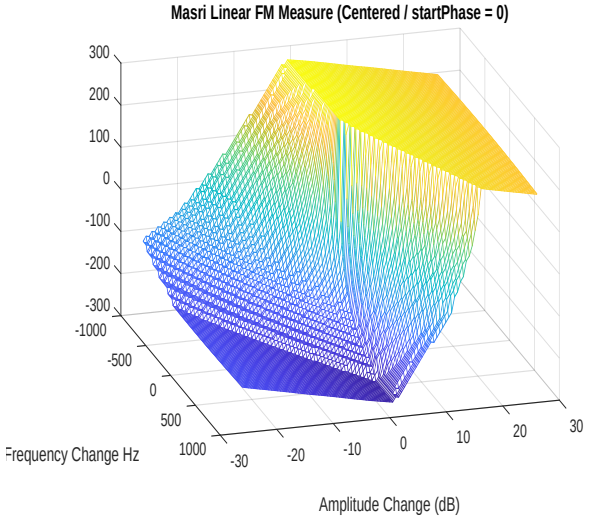


(A) Masri Phase Difference Measure of dF [-1000 to 1000 Hz], [-60 60 dB]

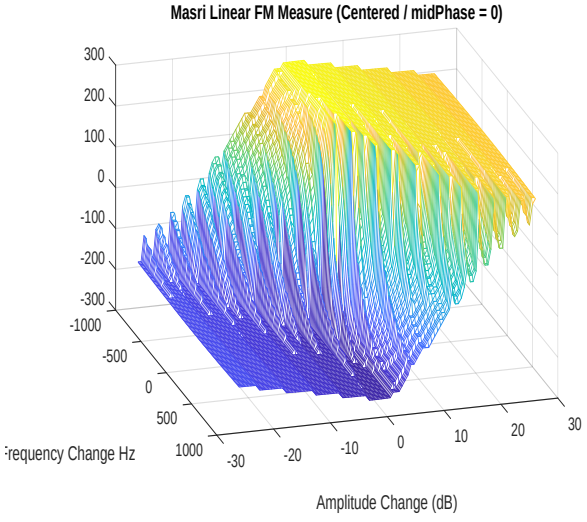


(B) Masri Phase Difference Measure of dF [-1000 to 1000 Hz], [-60 60 dB]

FIGURE 6.5: Comparison of Masri Phase difference measure from 2.9.1 with phase set to 0 at (A) the start of the frame, and (B) the middle of the frame.



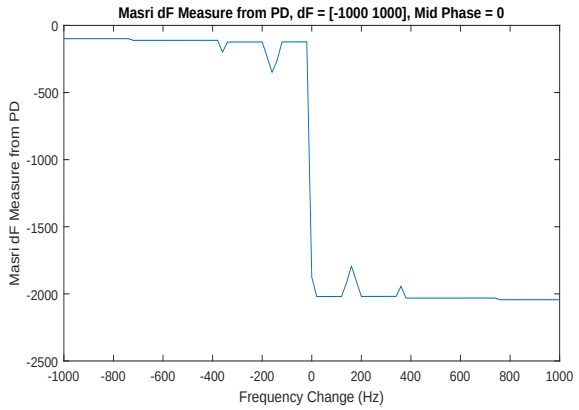
(A) Masri Phase Difference Measure of dF [-1000 to 1000 Hz], [-60 60 dB]



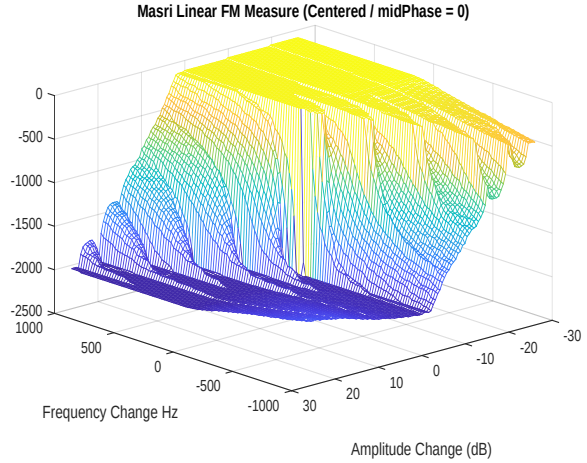
(B) Masri Phase Difference Measure of dF [-1000 to 1000 Hz], [-60 60 dB]

FIGURE 6.6: Comparison of Masri Phase difference measure from 2.9.1 with phase set to 0 at (A) the start of the frame, and (B) the middle of the frame.

The phase difference measure was also evaluated using phase information which is negatively wrapped. The results from the negatively wrapped phase are shown in Figure 6.7, which display similar modulations to the results presented above, which use the unwrapped phase returned from the DFT.



(A) negWrapped Masri Phase Difference Measure of dF



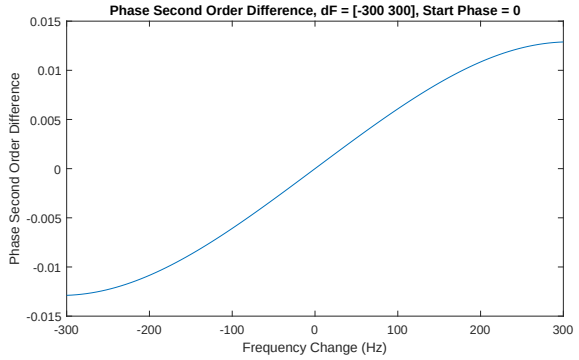
(B) negWrapped Masri Phase Difference Measure of dF [-1000 to 1000 Hz], [-60 60 dB]

FIGURE 6.7: A) negWrapped Phase Second Order Difference for dF, B) zoomed view to display slope

Due the measure used in 2.9.1 showing some modulations within the results, an alternative phase difference measure was examined for estimating linear frequency change from within a single analysis frame. The phase distortion across a peak is curved in the presence of frequency change, therefore, due to the above measurement results being influenced by the phase, investigation of the second order difference of the phase across a sinusoidal peak was examined, and shown to be useful for accurately estimating frequency change within a single analysis frame in Figure 6.8. The quality of the estimation methods using the second order difference of the phase across a sinusoidal peak was evaluated. The measure from the unwrapped phase provides a function from which unique frequency change values can be determined up to a certain amount, dependant on the size of the analysis window. For a 1025 window the unique points on the curve range from -300 to 300 Hz. Frequency change values greater than this are no longer unique as two possible frequency change results are represented by the curve of this function, as seen in Figure 6.9.

The second order difference of the phase with respect to frequency across a sinusoidal peak is show to be less susceptible to initial phase conditions. Figure 6.10 compares this measure with different initial phase conditions, a range of ΔF and ΔA values, and with frame sizes of $N = 1025$ and $N = 2049$. The result presented display a smoother function compared to the results presented from the measure used in 2.9.1.

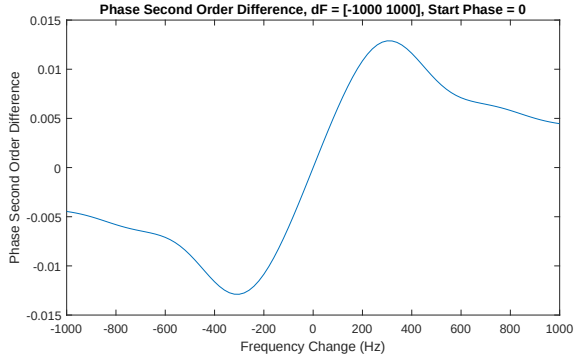
The system implemented in this chapter uses the second order difference of the phase with respect to frequency across a sinusoidal peak for estimating frequency change between -300 and 300 Hz. Extending the model to use higher frequency change estimates is left as a future directive.



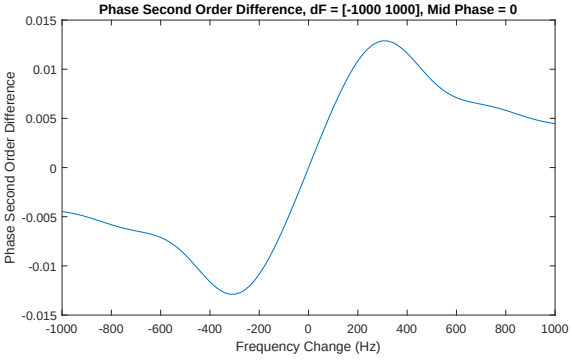
(A) Phase Second Order Difference [-300 to 300 Hz], Start Phase = 0

(B) Phase Second Order Difference [-300 to 300 Hz], Mid Phase = 0

FIGURE 6.8: Phase Second Order Difference [-300 to 300 Hz], A) Start Phase = 0, B) Mid Phase = 0

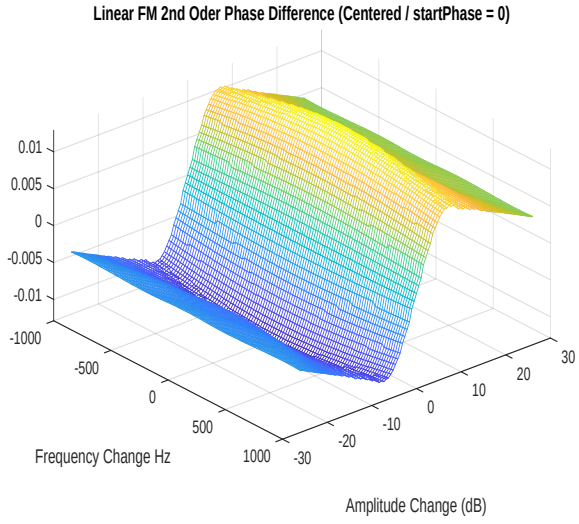


(A) Phase Second Order Difference [-1000 to 1000 Hz], Start Phase = 0

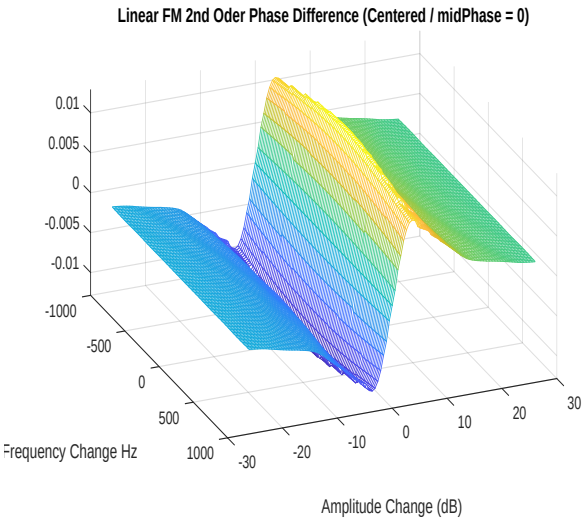


(B) Phase Second Order Difference [-1000 to 1000 Hz], Mid Phase = 0

FIGURE 6.9: Phase Second Order Difference A) Start Phase = 0, B) Mid Phase = 0



(A) Phase Second Order Measure of dF [-1000 to 1000 Hz], [-60 60 dB], N = 1025 samples



(B) Phase Second Order Measure of dF [-1000 to 1000 Hz], [-60 60 dB], N = 2049 samples

FIGURE 6.10: Comparison of Phase Second Order measure with phase set to 0 at (A) the start of the frame, and (B) the middle of the frame.

6.2.3 Correcting Estimate Biases

The effect of amplitude and frequency change on the magnitude spectrum has been presented in Chapter 4. The spread of energy in the magnitude spectrum across neighbouring frequency bins results in biased estimates for mean amplitude and frequency. Correcting these estimates is crucial for an accurate model. Methods for correcting amplitude and frequency estimates in the presence of these modulations is presented in [263]. However, the method described for correcting the mean amplitude estimate was not accurate enough for single frame estimates in the presence of large modulations. The model adopted in this thesis therefore implements a simple lookup table for correcting amplitude estimate biases. The correction table is generated by comparing measured amplitude estimates with known values from a reference input signal, and storing the differences in a two dimensional table for a range of amplitude and frequency changes. Figure 6.11 show the biased uncorrected amplitude estimate compared to the corrected amplitude estimate. The mean frequency is corrected from [9], using the estimates of ΔF and ΔA from the DFT when a Hanning window is applied.

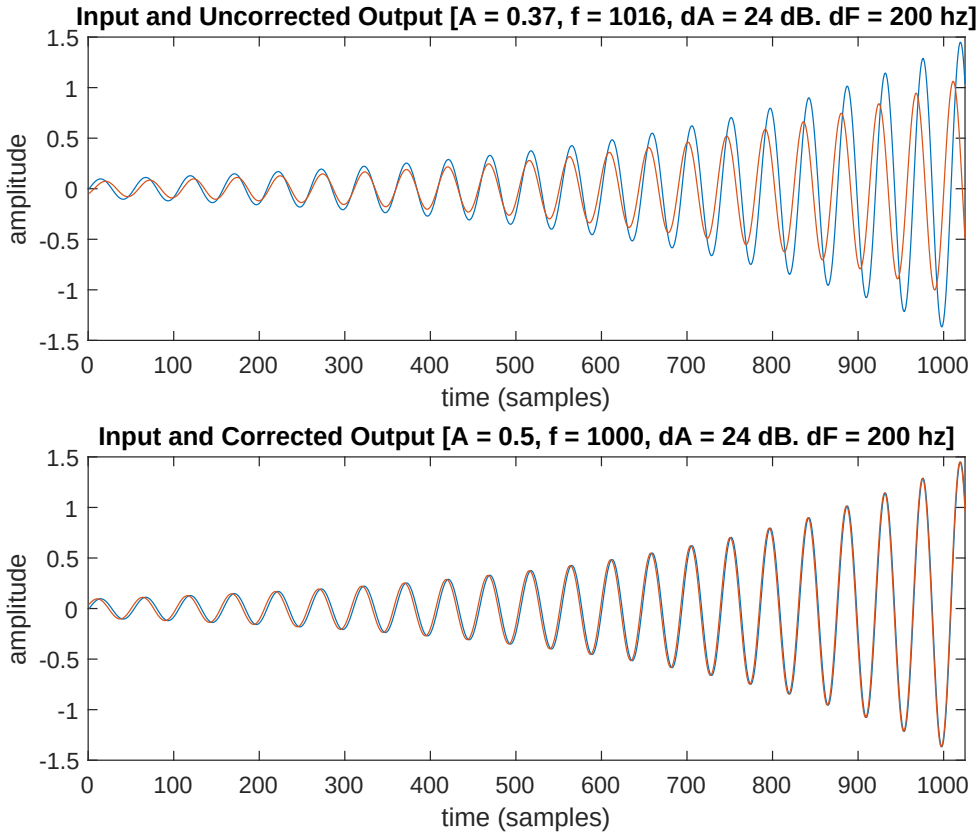


FIGURE 6.11: Output (@48 kHz) with dF, dA, and A corrected using 2D Lookup Tables

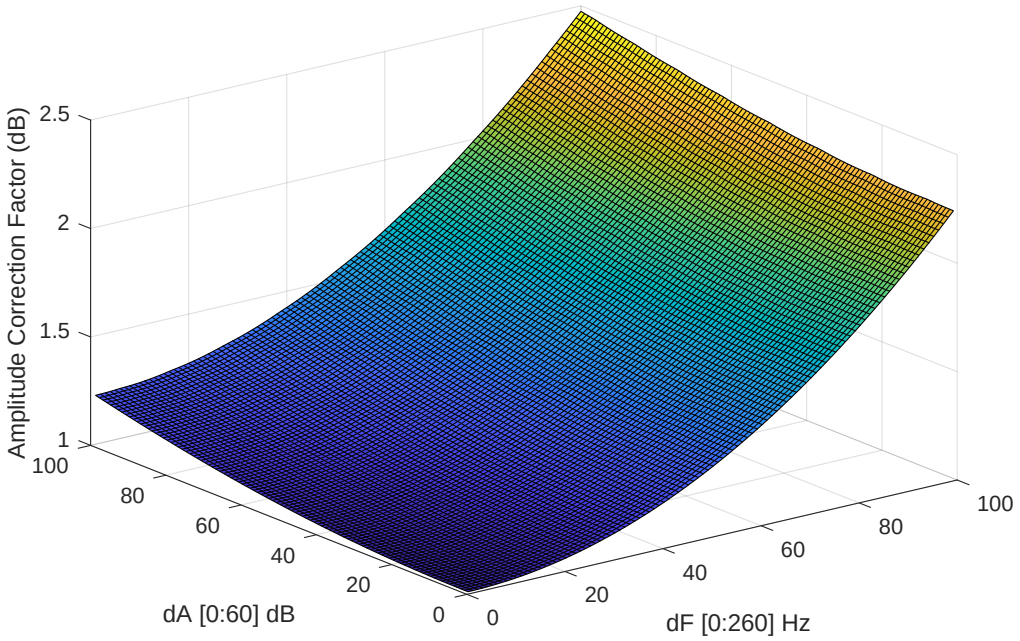


FIGURE 6.12: Amplitude Correction Factor Lookup Table

Figure 6.12 shows the amplitude correction table used in Figure 6.11 for correcting the estimate.

6.2.4 Frame Linking

Spectral Modelling Systems have been presented in Section 2.5. These models analyse an audio signal by segmenting the audio into frames and analysing each frame using the DFT. In [75] Smith and Serra adopted a peak-tracking technology from the Navy to search for peaks in the FFT and track the largest peaks between frames, in a similar method as the phase vocoder [73]. McAulay and Quatieri developed a similar peak continuation algorithm in which amplitude an frequency envelopes between frames are connected through breakpoints with "birth" and "death" criteria between frames [73, 88]. In [148, 164] the harmonic structure of a sound is used to aid linking of harmonic partials between frames. Phase coherence between frames has been discussed in Sections 2.4.5, 2.5.3, 4.5.5 and 4.5.6. Maintaining phase coherence between frames is important when performing time and pitch scale modifications. However, the current system implementation described in the following section achieves reasonable continuation of amplitude and phase alignment between frames from parameter estimates using MoP. The current system would benefit from a peak continuation algorithm which interpolates the amplitude and phase of tracked partials between frames, and so is left as a future improvement to the current implementation.

6.3 System Implementation

The system implementation section of this chapter is split into two separate frameworks; Segmented 6.3.1 and Non-Segmented 6.3.2. One of the research objectives outlined in the beginning of the thesis was to investigate the performance of an over-complete single-frame sinusoidal modelling system which was aimed at running in real-time. A non-overlapping single frame analysis system results in fewer analysis frames compared to an overlap-add system, but this potentially comes at a higher computational cost in order to achieve a similar quality.

Another research objective outlined during the undertaking of this thesis was the examination of modelling short broadband signals with non-stationary sinusoidal bases. This is described in Section 4.5.11 where percussive instruments with short attack times were shown to be well modelled using an over-complete non stationary sinusoidal atomic decomposition. This is explored further in Section 6.3.2 where a system combining MoP and the undecimated DWT is applied to modelling sounds using a single analysis frame encompassing the entire audio sample.

6.3.1 Segmented Framework

The segmentation of audio into frames has been presented in Section 5.3. This is a necessary procedure for processing large files or a stream of audio, is required for a real-time framework and for reducing latency.

Figure 6.13 displays a simplified flow diagram of the segmented system implementation. An audio (file/stream) is segmented into frames of equal length, and processed on a frame by frame basis. The analysis / synthesis process is described in two stages. Firstly, an MoP decomposition using non-stationary sinusoidal bases as described in Chapter 4 is applied for separating the input signal into a sinusoidal and residual signal. The undecimated DWT described in Chapter 5 is then applied for separating transient and noise components from the residual. These individual components are then potentially altered in some way and recombined before being presented at the output frame.

Figure 6.14 highlights the main steps involved within the sinusoidal analysis stage for estimating individual sinusoidal components within the analysis frame, and their relevant parameters.

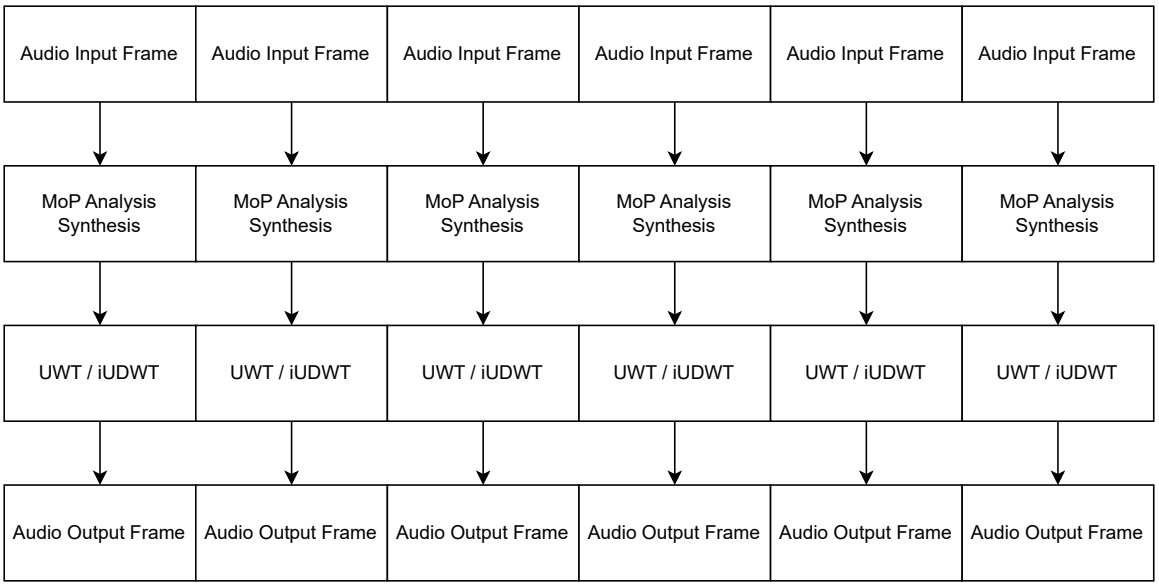


FIGURE 6.13: System Flow Diagram (Non-OLA Single Frame Analysis/Synthesis)

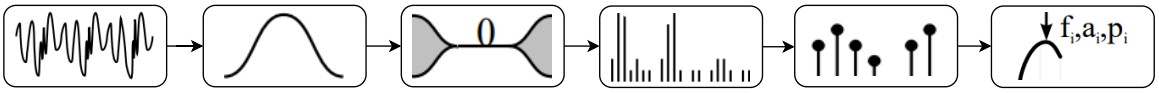


FIGURE 6.14: System Flow Diagram of a frame of audio, windowed, zero-phase padded, and parameters extracted from selected DFT peak

A Hanning window is applied to the analysis frame, which is then zero-phase padded before performing the DFT. The magnitude and phase spectrum's are calculated from the values returned from the the DFT, followed by a peak selection process extracted from the magnitude spectrum.

A sinusoidal peak in the current implementation is simply defined as a spectral bin within the magnitude spectrum which is greater than each of the two neighbouring bins on each side. The calculation of the magnitude second order difference measure requires two neighbouring bins on each side of a spectral peak. This limits the lowest frequency bin which is able to be modelled by the previously described methods to around 47 Hz for a DFT containing 2048 samples (@48 kHz), and around 94 Hz for a DFT containing 2048 samples (@48 kHz). The peaks returned from the selection process are ordered from those with the largest magnitude to those with the lowest magnitude. Further refinement to the system with more complex peak selection algorithms which avoid selection of side-lobes as spectral peaks is left as a future research objective. Parameters of mean amplitude, mean frequency, phase, amplitude change and frequency change are then estimated and corrected using the methods described in the previous section.

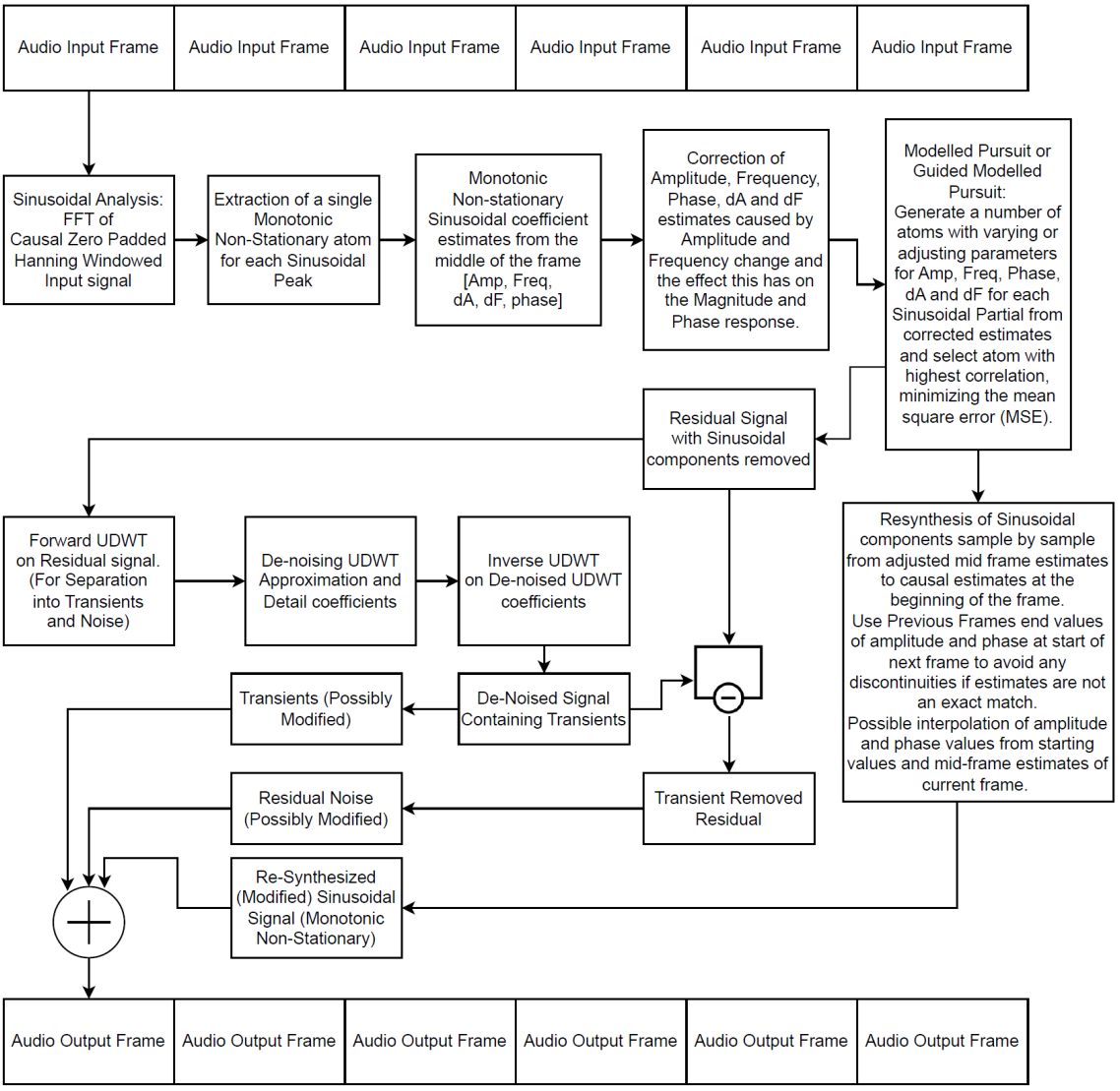


FIGURE 6.15: System Flow Diagram (Non-OLA Single Frame Analysis/Synthesis)

Figure 6.15 displays a more detailed flow diagram of the segmented single-frame analysis framework and how the sinusoidal, transient and noise components are separated and re-combined within the framework. Figure 6.15 highlights that the MoP implementation used within this framework uses a peak selection stage where MoP is only performed once per peak, selecting the best fitting atom for each peak and then moving on to the next peak, rather than allowing a single peak to be modelled by multiple atoms as is the case with an over-complete decomposition. The details of the flow diagram are listed below.

1. Segmentation: Split audio into frames of equal length
2. Zero-Phase Analysis:
   - Zero-Phase Pad Input Frame
   - Calculate DFT
   - Perform Peak Selection
   - Estimate Sinusoidal Peaks parameters using parabolic interpolation
   - Estimate change in amplitude using phase difference lookup table
   - Estimate frequency change using second order difference lookup table
   - Correct Amplitude Estimate using correction table
   - Correct Frequency Estimate using corrected amplitude estimate and estimates of ΔA and ΔF
3. Perform MoP on each sinusoidal component, selecting the atom which is the best fit
4. Separate signal into sinusoidal and residual parts
5. Sinusodial components then re-synthesised after optional modification (eg. time stretching / pitch shifting)
6. Residual signal modelled using undecimated DWT / inverse
   - Perform forward transform, separating residual into approximation and detail coefficients
   - Extract Transients from wavelet coefficients via de-noising techniques and inverse transform
7. Combine possibly modified sinusoidal, transient and noise components into output frame

The system implementation described above is dependant on the analysis and synthesis frame sizes to be of equal lengths. This is not a problem when using a causal implementation and frame sizes of an even length, as long as the frame size meets the undecimated wavelet transforms criteria of a frame length divisible by 2^{level}. In practise, a segmented audio system's frame size is a power of two and so this is not a problem unless the wavelet decomposition level results in a frame size requirement larger than the analysis frame size. A frame size of 1024 samples restricts the wavelet decomposition level to $j = 10$ ($2^{10} = 1024$).

A non-causal implementation with an odd frame size such as 1025 is only a single sample more but this mismatch in frame sizes can potentially cause the sinusoidal and residual output frames to become out of sync, or introduce extra latency into the system, such as when a workaround where two or more input frames are "buffered" before processing can begin, to ensure there are always 2^j samples available for processing.

The mismatch of audio frame sizes between a non-causal sinusoidal analysis implementation, and that required by the inverse undecimated DWT is described in more detail in the next section.

6.3.1.1 Causal Compared to Non-Causal Framework

The main difference between the causal and no-causal implementations is the zero-phase padding required for the non-causal implementation, which as explained in 2.4.4, requires an odd size frame length to keep the signal as symmetric as possible when applying zero-padding to remove unwanted shifting distortions. A causal implementation of MoP combined with the segmented UWT is shown in Figure 6.13. With both block sizes having the same even length, there are no problem with alignment and size of the output frames. However, when combining a non-causal MoP implementation which uses zero-phase padding and an odd frame length, in conjunction with the UWT, there are discrepancies between the required frame sizes.

There is a requirement when performing the forward SWT followed by the inverse SWT, that the frame size is be divisible by 2^{Level}. The forward SegUWT was adapted and able to handle odd frame lengths such as 513 and 1025. This is currently implemented as an FIR filter with filter states and so the processing of the froward SegUWT is unaffected by the frame length. The inverse SegUWT is however, affected by the frame length. The algorithm for the iSWT which has been adapted with suitable overlap extension lengths by the inverse SegUWT is given in 5.16 and 5.3.3.

The inverse operation of the SWT relies on a sub-sampling operation of 2 where two arrays ($x1$ and $x2$) are calculated using the inverse DWT on a subset of approximation and detail coefficients, and then summed together. The sub-sampling by a factor of two naturally requires an even input vector, which is the current limitation for the inverse SegUWT.

Initial investigations into the problem highlighted that the values within the sub-sampled vectors ($x1$ and $x2$) are very close to one another. A number of tests were conducted on different signals with different wavelet filters, filter orders and decomposition depths and the mean difference between the two vectors was found to be within the ranges between -45 and -90 dB.

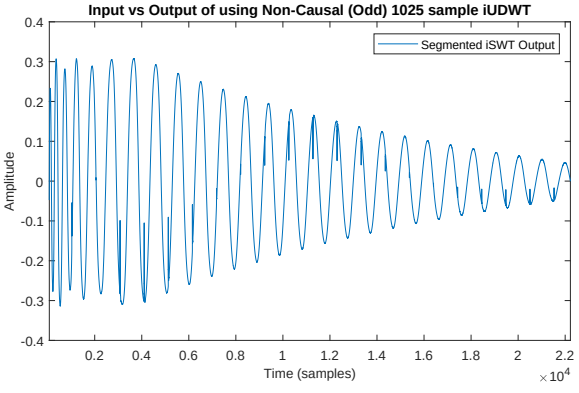
The current workaround used in the non-causal implementation of the inverse SegUWT copies the required number of missing samples from the end of the $x1$ vector to the end of $x2$.

An example of a kick drum decomposed and re-synthesised using the SegUWT and its inverse, using frame lengths of $N = 1024$ and $N = 1025$ are presented below, and compared against the outputs of the non-segmented SWT and its inverse.

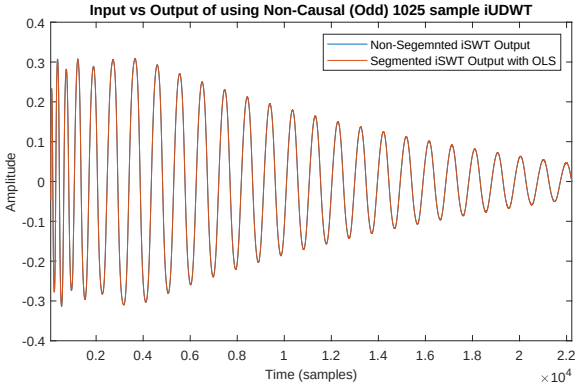
Figure 6.16a displays the output of the inverse SegUWT if the $x2$ vector is zero-padded by the missing amount of samples. This clearly shows the frame boundary border errors. In comparison, Figure 6.16b displays the input signal against the output of the SegUWT. This clearly shows the workaround above is able to improve the errors at frame boundaries.

Figures 6.17a, 6.17b, 6.18a, and 6.18b display a closer inspection of a frame boundary error. The errors in Figures 6.17b and 6.18b are not completely resolved with the workaround, but it is shown to be noticeably less than the error in Figures 6.17a and 6.18a.

Figure 6.19 displays the resulting residual error in dB between zero-padding and the workaround of copying neighbouring samples. The frame boundary errors are reduced from around -20 dB to below -60 dB. One of the exit criteria for MoP in Chapter 4 was for the residual signal to fall below a threshold of -60 dB, and so this workaround, although not optimal and able to provide perfect reconstitution, works well enough to reduce frame boundary errors to an acceptable amount.

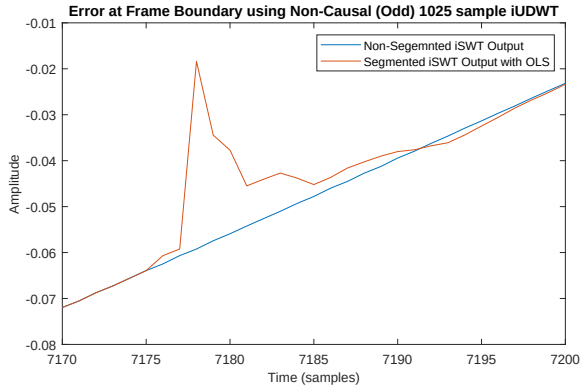


(A) iUDWT (1025 frame) Output inserting zeros

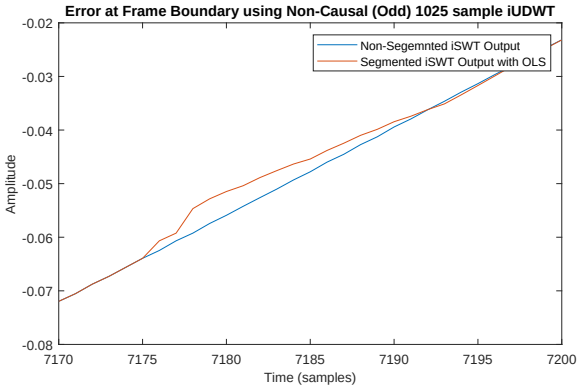


(B) iSWT (Full signal) vs iUDWT (1025 frame) with Workaround

FIGURE 6.16: iUDWT Output of Kick (@48 kHz) with (B) and without (A) Workaround

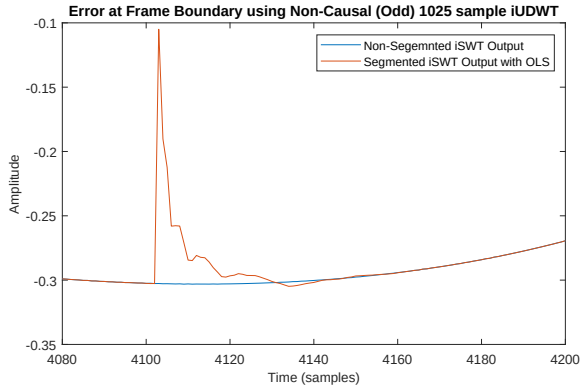


(A) iSWT (Full signal) vs iUDWT (1025 frame) border error

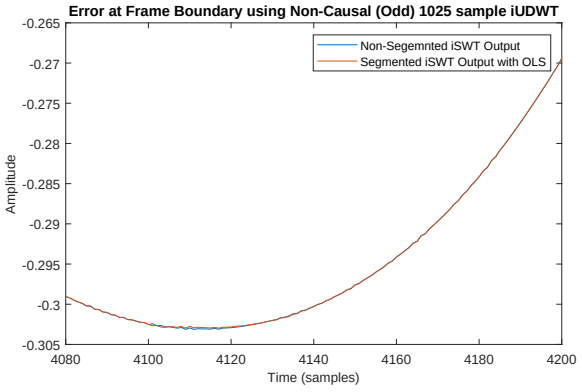


(B) iSWT (Full signal) vs iUDWT (1025 frame) border error

FIGURE 6.17: iSWT (Full signal) vs iUDWT (1025 frame) border error

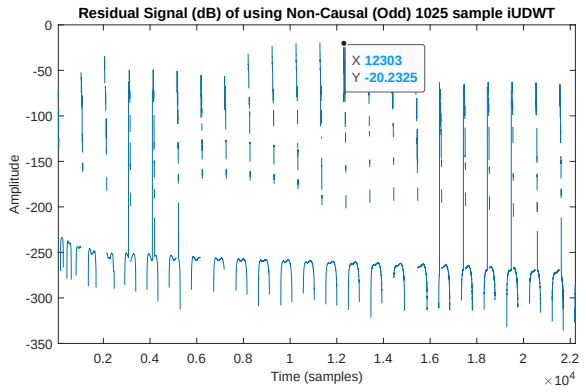


(A) iSWT (Full signal) vs iUDWT (1025 frame) border error

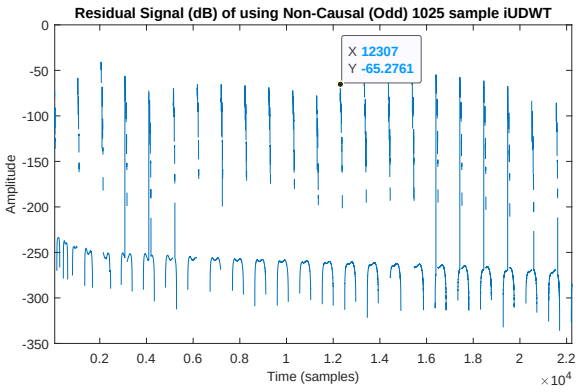


(B) iSWT (Full signal) vs iUDWT (1025 frame) border error

FIGURE 6.18: iSWT (Full signal) vs iUDWT (1025 frame) border error

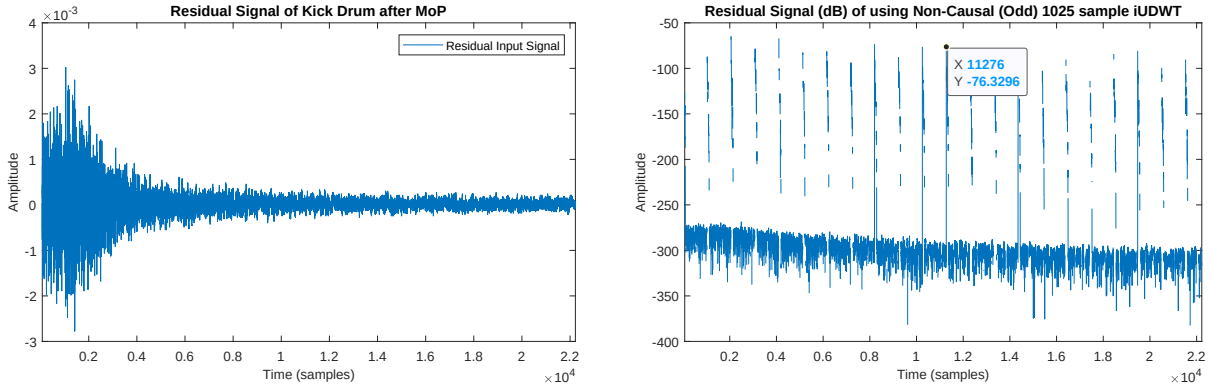


(A) iSWT (Full signal) vs iUDWT (1025 frame) Residual (dB)



(B) iSWT (Full signal) vs iUDWT (1025 frame) Residual (dB)

FIGURE 6.19: iSWT (Full signal) vs iUDWT (1025 frame) Residual (dB)



(A) Kick Example Input after MoP (B) iUDWT (1025 frame) Residual (dB)

FIGURE 6.20: Kick (@48 kHz) Example Input after MoP and iUDWT (1025 frame) Residual (dB)

The above example uses the input signal of a kick drum. In the proposed system, this would be modelled by the SMS model first, leaving a residual signal containing mostly transient components and noise. This residual signal would be lower in energy, and contains more broadband noise. Figure 6.20a displays the residual signal of the kick drum in the above example after a MoP decomposition. The peak amplitude of the signal is -48 dB so is already a very quiet signal, and so in this example the workaround would be an acceptable solution.

Processing an audio buffer in a framework which combines a non-causal zero-phase padded framework with odd frame lengths, with the UWT which ideally has an even frame size divisible by divisible by n^{Level} is a nontrivial problem. A causal MoP implementation with the same frame size as the UWT is well suited for a real-time implementation. The non-causal implementation with the discrepancies between the MoP (Odd) frame size and the UWT (Even and divisible by 2^{level}) frame size, is possibly better suited to an offline process where the entire audio file can be processed in a fixed frame size for MoP or the UDWT, and the results of that output can then be re-analysed from the beginning with a different frame size. The implementation challenges of a non-causal (odd frame size) in conjunction with the UWT (either before or after) limit this, or at the very least pose a challenging implementation task, of which the possible resolutions, may result in an additional delay between the input and output. The UWT already introduces a delay to the system, any additional delays hinder the implementation of the system to perform in real-time.

The workaround investigated above is deemed acceptable within this framework, however this is not an optimal solution which is able to provide perfect reconstruction. The extension of the inverse UWT in a non-causal framework is left as a current limitation and area for future improvement.

6.3.1.2 Multiple Frame Signal Tests

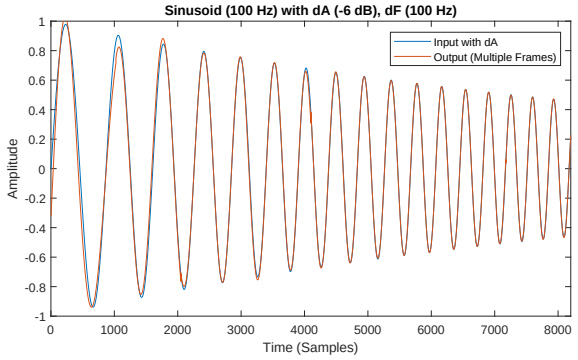
A segmented MoP and UWT framework has been presented in the previous section, and although the non-causal implementation presents a challenge regarding the mismatch between the different requirements of even and odd frame lengths for perfect reconstruction, both causal and non-causal MoP implementations are capable of extracting monotonic sinusoidal components from the segmented input stream before transient and noise separation from the residual using the SegUWT. The main functional difference between the two implementations is the envelope type discrimination, which has currently only been derived from non-causal phase distortion measures. Extending this functionality to the causal implementation remains a future research task.

The current implementation does not maintain phase and amplitude coherence between frames.

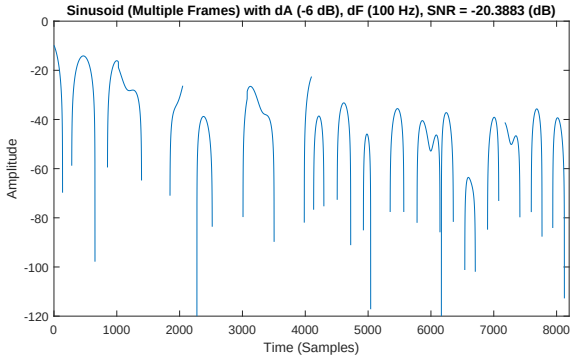
The results of single partial signals decomposed using MoP with correction tables, and re-synthesised using the segmented non-causal framework described in the Section 6.3.1 is presented below. Some simple sinusoids with varying amplitude and frequency changes are modelled using the non-causal implementation of MoP with a frame size of $N = 1025$. The test signal is 8200 samples resulting in 8 input/output frames. Figure 6.21 displays a simple 100 Hz sine wave with -6 dB ΔA and 100 Hz ΔF. Although the output is closely approximated by the output signal, the first frame is not an exact match which results in a disappointing SNR measure of -20 dB. The output of a 300 Hz sine wave with -6 dB ΔA and -200 Hz ΔF shows a slight improvement in Figure 6.22.

A third sine wave with a large amplitude change of -20 dB performs slightly better with an SNR of -34 dB. The improved measure in Figure 6.23 is attributed to a larger lookup table of 501 samples rather than a table of only 201 samples used in the other two tests. The different errors between analysis frames is clearly shown in Figure 6.23b which is also attributed to the larger lookup table used.

Even though amplitude and phase coherence are not employed, the single frame MoP analysis methods perform well and discontinuities at frame boundaries are negligible.

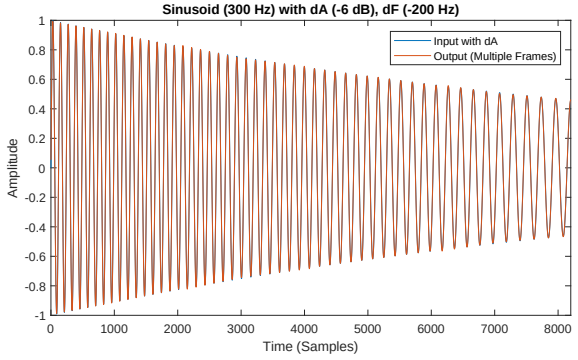


(A) Sinusoid modelled using MoP

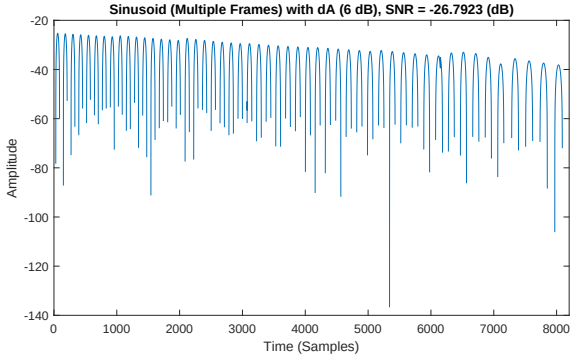


(B) SNR of sinusoid modelled using MoP

FIGURE 6.21: Output and SNR of sinusoid modelled with MoP over multiple frames (1025 samples)

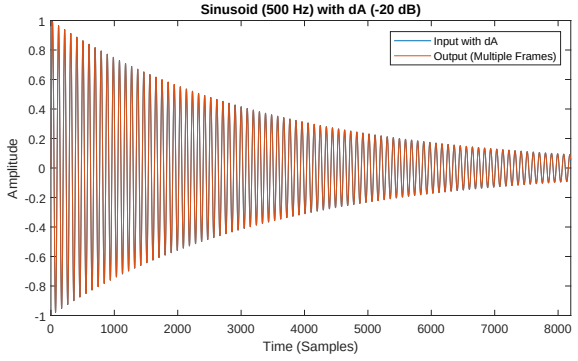


(A) Sinusoid modelled using MoP

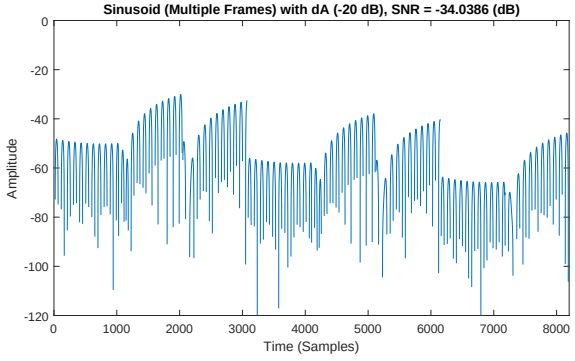


(B) SNR of sinusoid modelled using MoP

FIGURE 6.22: Output and SNR of sinusoid modelled with MoP over multiple frames (1025 samples)



(A) Sinusoid modelled using MoP



(B) SNR of sinusoid modelled using MoP

FIGURE 6.23: Output and SNR of sinusoid modelled with MoP over multiple frames (1025 samples)

Figure 6.24 displays an example of a Bass Guitar and the resulting output using a segmented MoP decomposition. This clearly shows the non-monotonic and transient components at the beginning of the signal which MoP has not modelled. The residual signal after subtracting the output from the MoP model and the original sound is shown below where the Transient is clearly visible. Figure 6.24 displays the first couple of frames of the output from MoP. There are clearly some discontinuities at the frame boundaries, and the amplitude between successive frames fluctuates, due to the interference's of many other non-monotonic broadband components within the transient section of the sample. The image below displays the bass guitar in the sustain region of the amplitude envelope curve. There are no discontinuities or fluctuations in the amplitude between output frames in the section of the sample, even though phase and amplitude coherence are not utilised here.

Figure 6.26 displays the magnitude spectrum of the input sample at these two sections for comparison. The spectrum in (A) over the transient region clearly has more spectral components compared to the sustain section presented in (B).

Figure 6.27 displays the SegUWT approximation and detail signals after thresholding and re-synthesis. The most appropriate wavelet filter, filter order and decomposition depth, and the best d-noising techniques for extracting transient components from noise are left as a future research task.

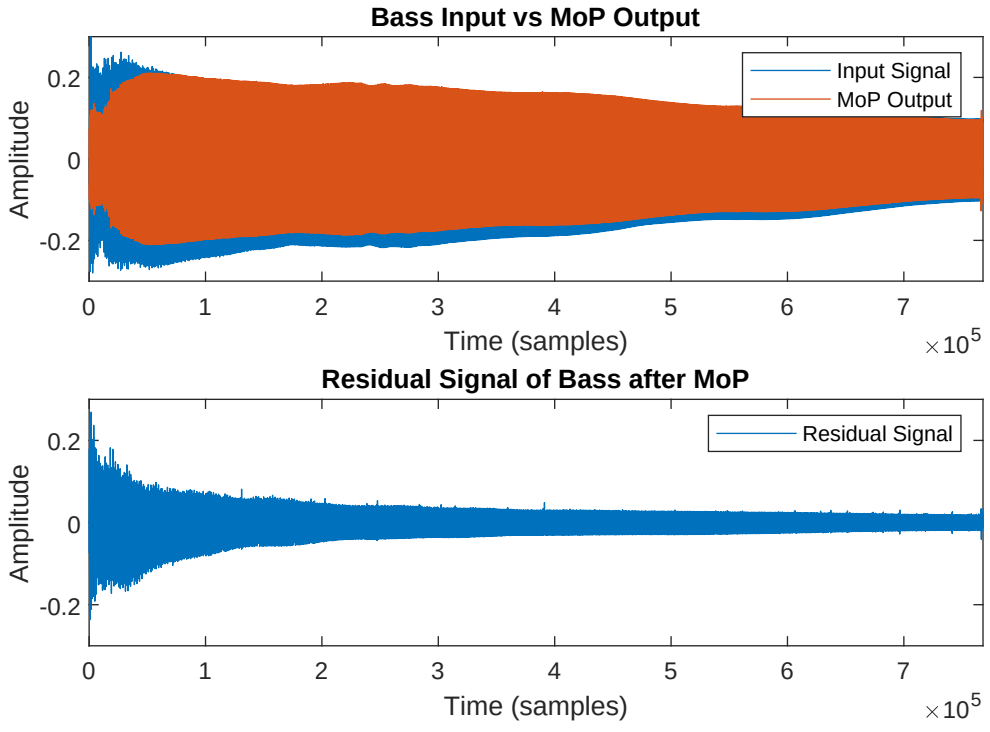


FIGURE 6.24: Nolly Guitar MoP Output and Residual

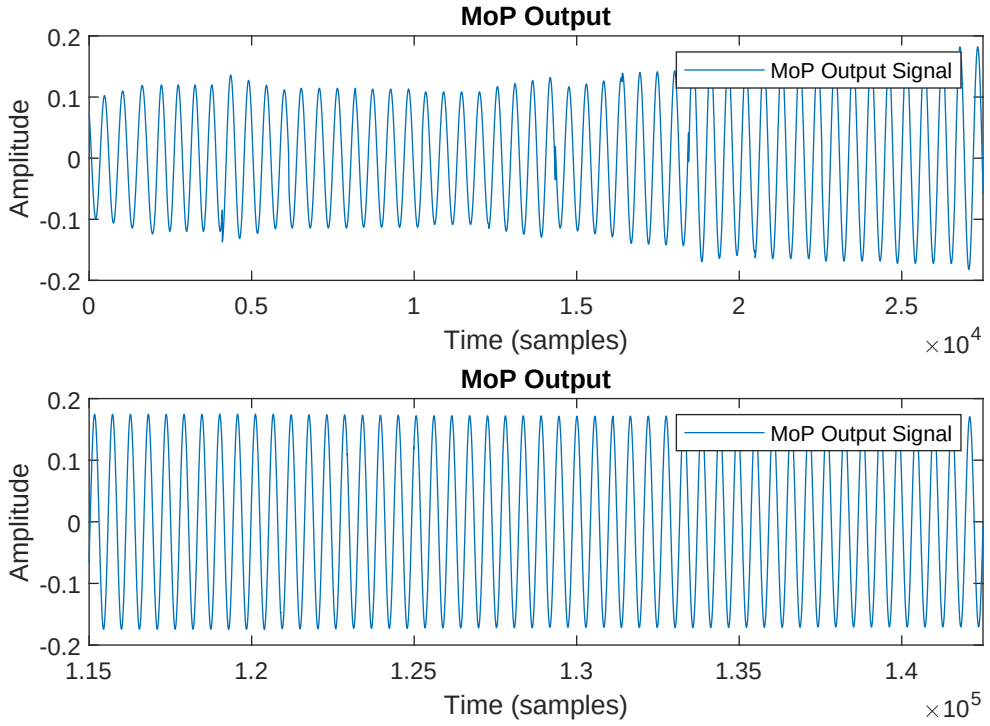
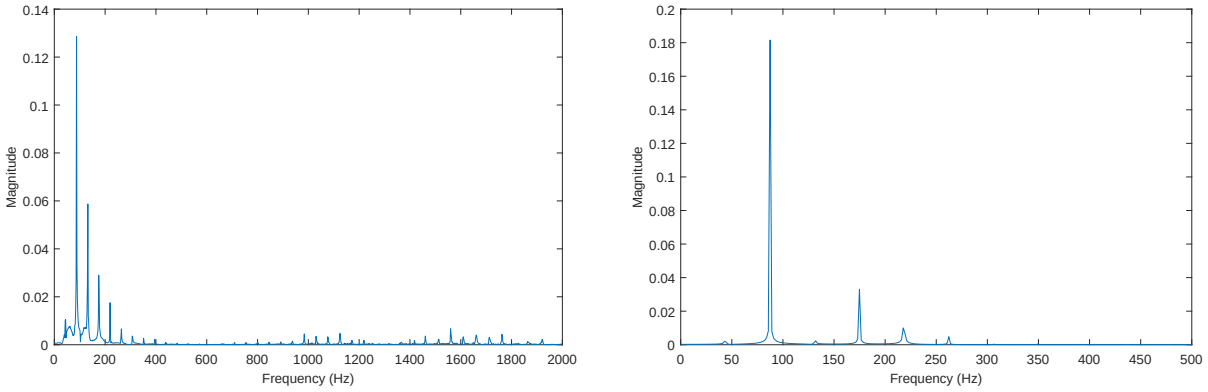


FIGURE 6.25: Nolly Guitar MoP Output and Residual



(A) Nolly Guitar Magnitude Spectrum (B) Nolly Guitar Magnitude Spectrum

FIGURE 6.26: Nolly Guitar (@44.1) Magnitude Spectrum

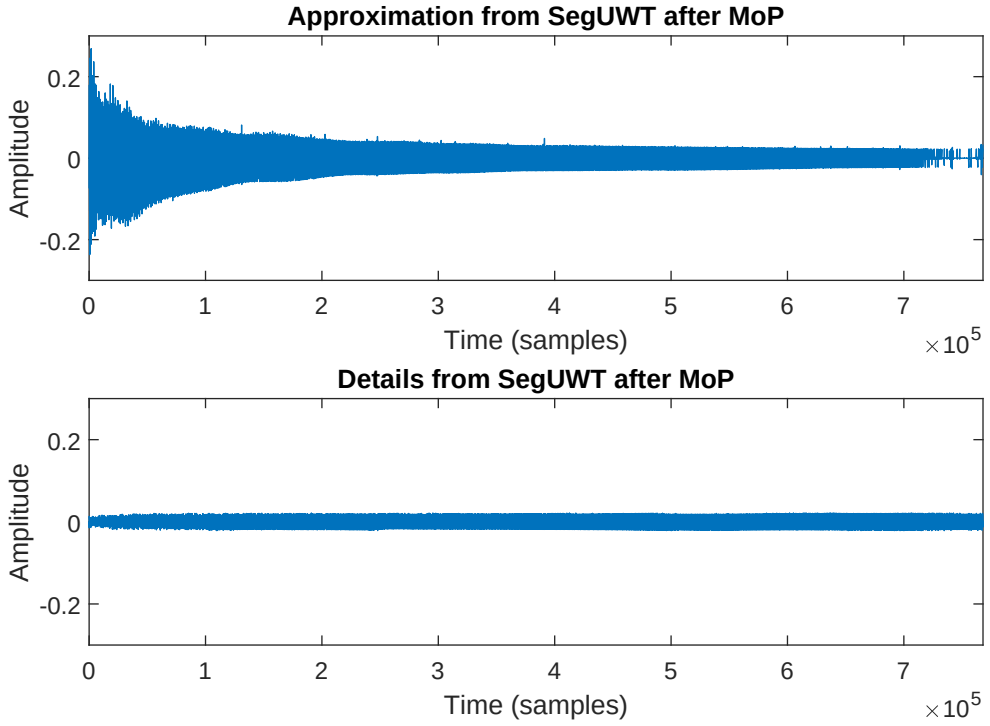


FIGURE 6.27: Nolly Guitar (@44.1) Approximation and Details from SegUWT

Figure 6.28 displays a synthesised bass sample and the output of a MoP decomposition. In this case, the maximum number of atoms was set to one, which results in the extraction of the strongest harmonic frequency from the sample. The residual signal containing the transient is displayed below it.

Figure 6.29 displays a section of the bass example during the sustain region of the sounds. This show a very good approximation, however there seems to be a DC component within the signal causing the mismatch between the input and output.

A small discontinuity between frames is visible just after 1×10^4 samples, otherwise MoP performs well, minimising discontinuities at frame boundaries caused by mismatches in amplitude and phase estimates.

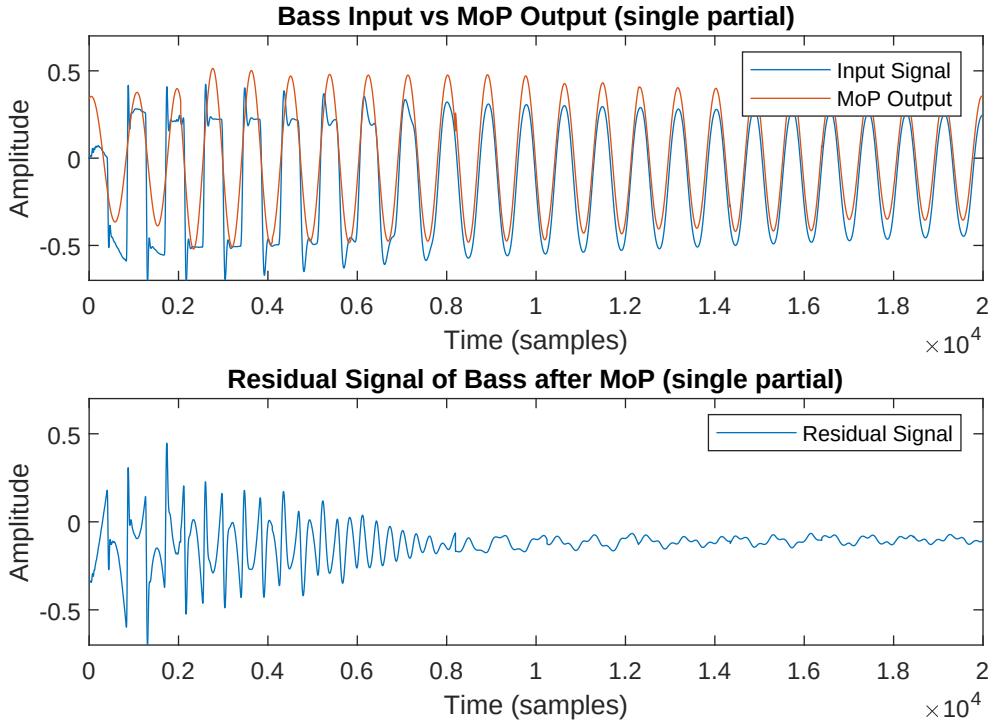


FIGURE 6.28: Bass (@44.1) kHz MoP Output (single partial) and Residual

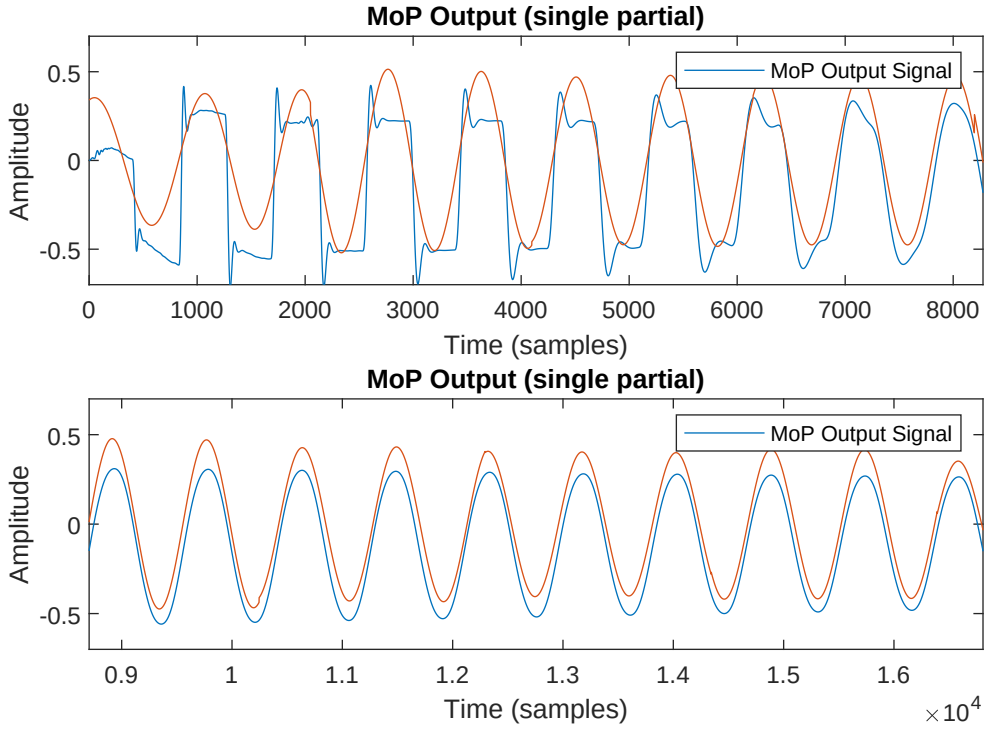


FIGURE 6.29: Bass (@44.1) kHz MoP Output (single partial)

6.3.2 Non-Segmented

The non-segmented implementation of MoP has been shown to work well for the analysis of percussive sounds with short attack times. It has been shown that non-monotonic amplitude modulation, meaning an attack time which falls at a midpoint within the analysis frame, followed by a portion of different section of the amplitude envelope, such as the sustain or release parts. This increase and hen sustain or decrease in energy over the analysis frame is captures with multiple components who's energy combines in such a way as to capture this non-monotonic behaviour. Such a representation is susceptible to errors when modifying the signal captures by the model, due to the atoms no longer combining in the exact same manner as to retain the original amplitude envelope.

The original implementation of MoP was applied to impulse responses where the attack is presumed instantaneous, followed by a n exponential amplitude release curve. Percussive components with short attack times have a similar amplitude envelope, where the attack section is short enough that it does not interfere with the resynthesis of the decaying part of the sound.

These sounds are well modelled using an MoP decomposition, where a single analysis frame is applied over the entire sound. The undecimated wavelet transform is able to separate high frequency content from low frequency content though the filterbanks and thresholding techniques presented in Chapter 5. These separate components containing high and low frequency components are able to be modelled separately using MoP. The high frequency components with short term broadband noise are modelled well using an MoP decomposition and perform well under time and pitch scale modifications.

The timbre of these sounds can be modified by the selection of partials within frequency bands and individually altered with time/pitch-scale or other modifications on a sample by sample basis.

The main criteria for such sounds to perform well and avoid the issues previously presented regarding non-monotonic amplitude change using MoP, is a very short attack time. In practise, percussive sounds with fast attacks less than 10 ms work well, and do not suffer from amplitude modulations in the re-synthesised sound after pitch and timescale modifications.

6.3.2.1 Single MoP decomposition on entire Percussive Sounds

The process for decomposing percussive sounds using a single analysis frame and a rectangular window are detailed below. The current implementation performs a single DFT on the entire (zero-padded) audio file, and uses causal estimation of exponential amplitude change. The percussive sounds tested have a very short attack followed by a longer decay. Exponential amplitude change is assumed here for simplicity, and because momentary excited oscillations that are linear and time-invariant can be assumed to be examples of signals which decay exponentially [26].

1. Open sound file (snare/hihat) and read entire sample into memory (analysis frame) x.
2. Optional: Separate Approximation and Details or Transient using UWT and inverse.
3. Perform MoP (As per 4.3.4) on sound file, or separate extracted parts.
4. Re-synthesise entire audio sample, or individual components from extracted parameters using bank of sinusoidal oscillators with control on a sample by sample bases for possibly applying transformations such as pitch shifting, time stretching.

Figure 6.30 displays an example of a snare which has been separated into the approximating and detail parts using the UWT and its inverse. The approximation captures the low-frequency 'shape' of the sound, while the details captures the high frequency broadband 'noise' component. Figures 6.31, 6.32, and 6.33 show the spectrograms of the approximation, detail and full-band reconstructed signal from the inverse UDWT, using the Debauchies 2 Wavelet with a decomposition level = $j = 3$.

The selection of the wavelet filter and the decomposition level have a significant effect on the filtering of the signal and resulting approximation and details signals. The impact and performance of several orthogonal and bi-orthogonal wavelet families, the filter order (wavelet filter length), and the decomposition depth, in the context of denoising are presented in [264]. The study concludes that a meticulous choice of wavelet parameters significantly alters the performance of the frequency bands and demonising output. An example of the effect of different wavelet families, the filter order and the decomposition level are presented in Figures 6.34 and 6.35.

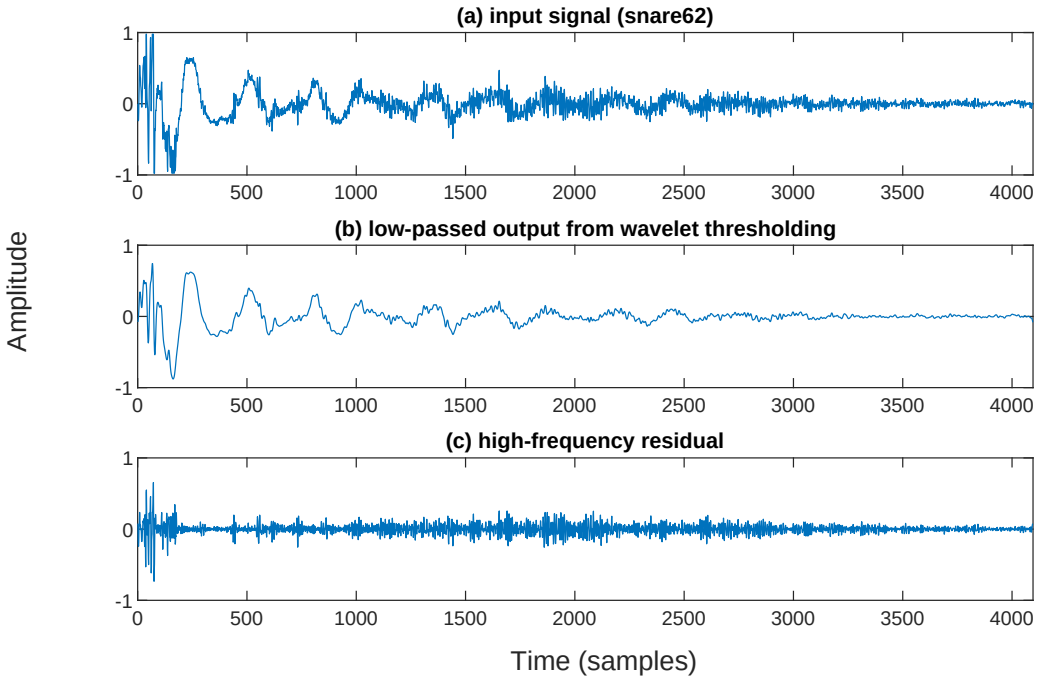


FIGURE 6.30: An example Snare separated into Approximation (low-frequency) and Detail (high-frequency) components using the UDWT

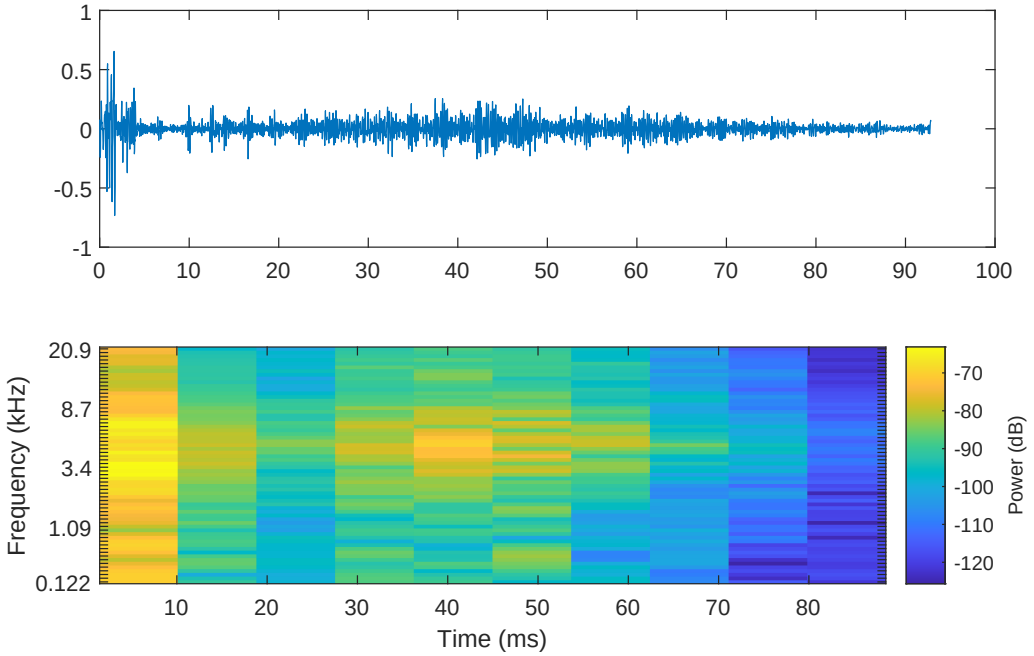


FIGURE 6.31: SNARE62 original low high

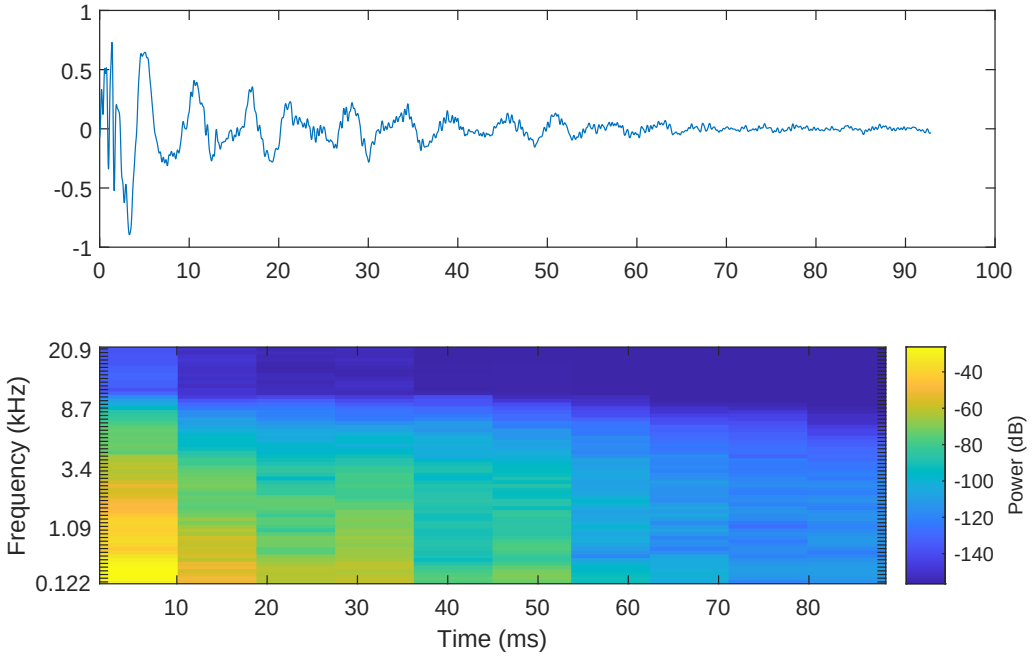


FIGURE 6.32: SNARE62 original low high

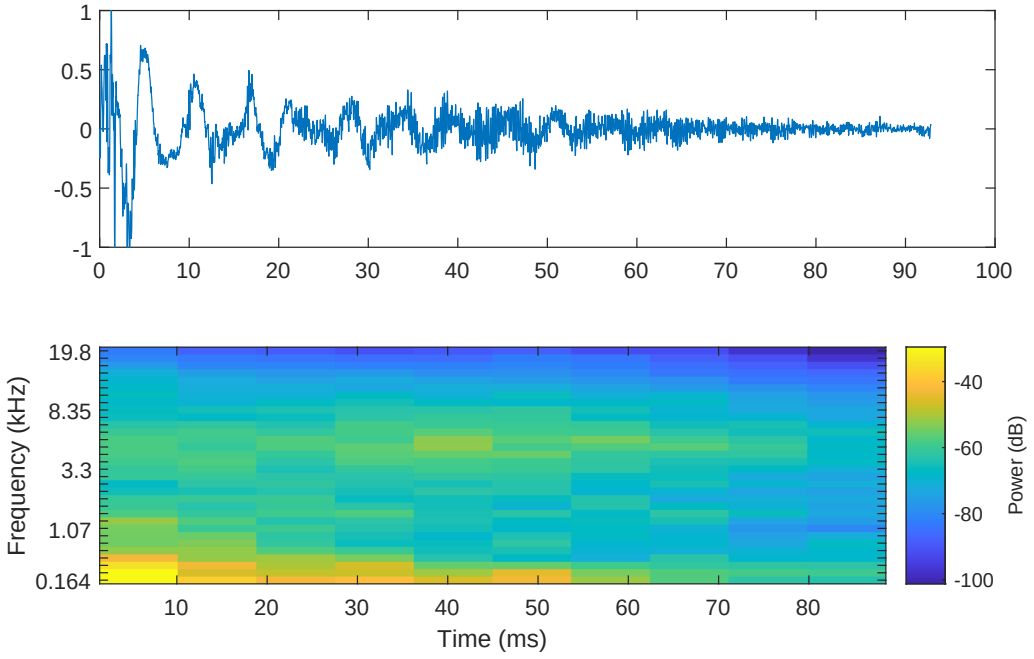
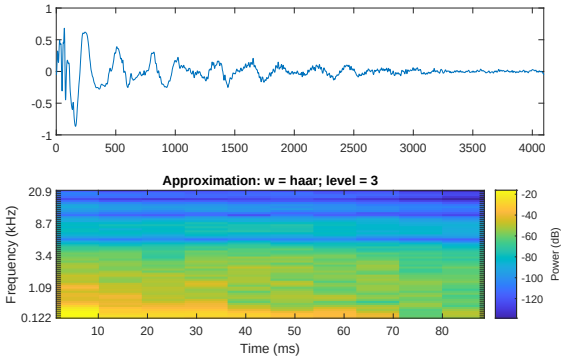
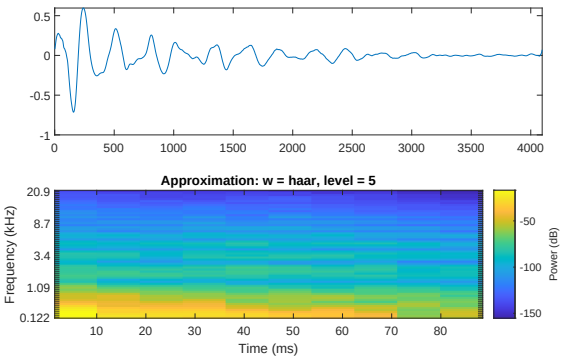


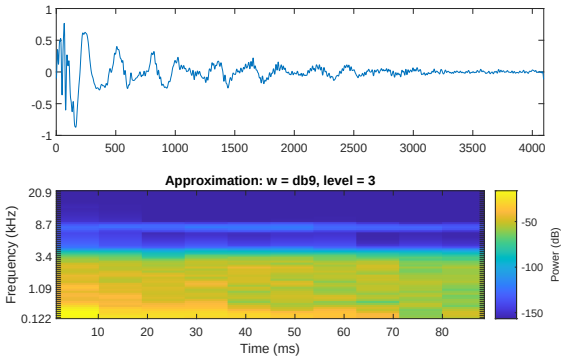
FIGURE 6.33: SNARE62 original low high



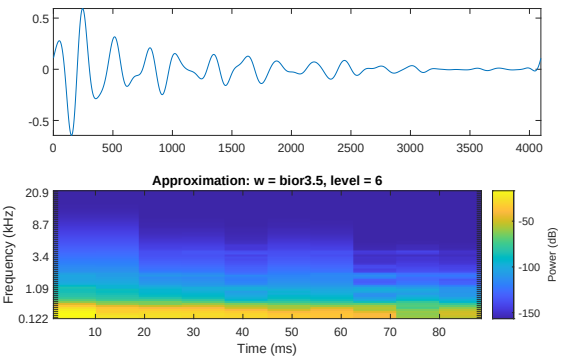
(A) approx haar level3



(B) approx haar level5

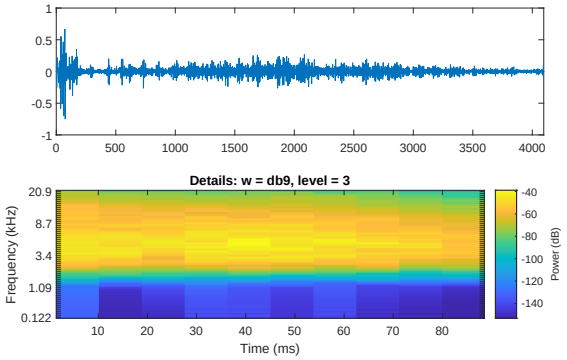


(C) approx db9 level3

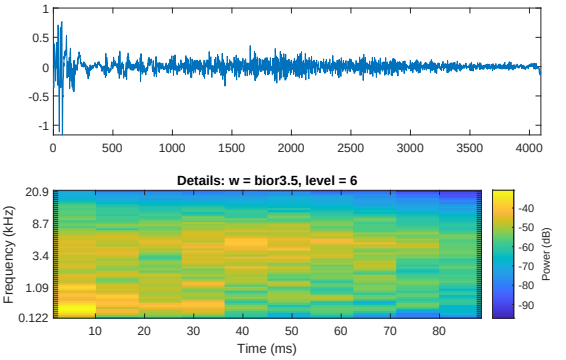


(D) approx bior3dot5 level6

FIGURE 6.34: Comparison of approximation signal (@48 kHz) from UDWT, (A) Haar wavelet at level 3 compared with Haar wavelet at level 5. (C) db9 wavelet at level 3 compared with (D) bior 3.5 wavelet at level 6



(A) details db9 level3



(B) details bior3.5 level6

FIGURE 6.35: Comparison of details signal (@48 kHz) from UDWT, (A) debauchies3 wavelet at level 3 compared with bior 3.5 wavelet at level 6

The wavelet filter and decomposition level effect the output of the wavelet decomposition and reconstruction.

Figure 6.36 displays the approximation signal extracted in Figure 6.30 pitch shifted by half and doubled. While Figure 6.37 displays the approximation signal time stretched by a half and doubled in length.

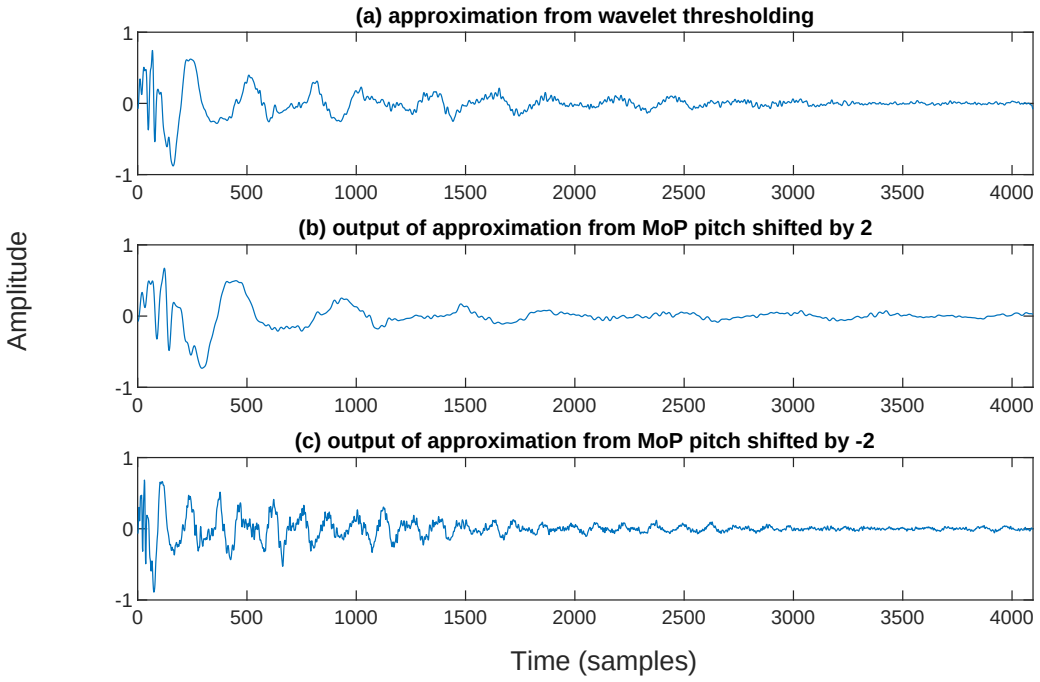


FIGURE 6.36: SNARE62 approximation Pitch Shifted

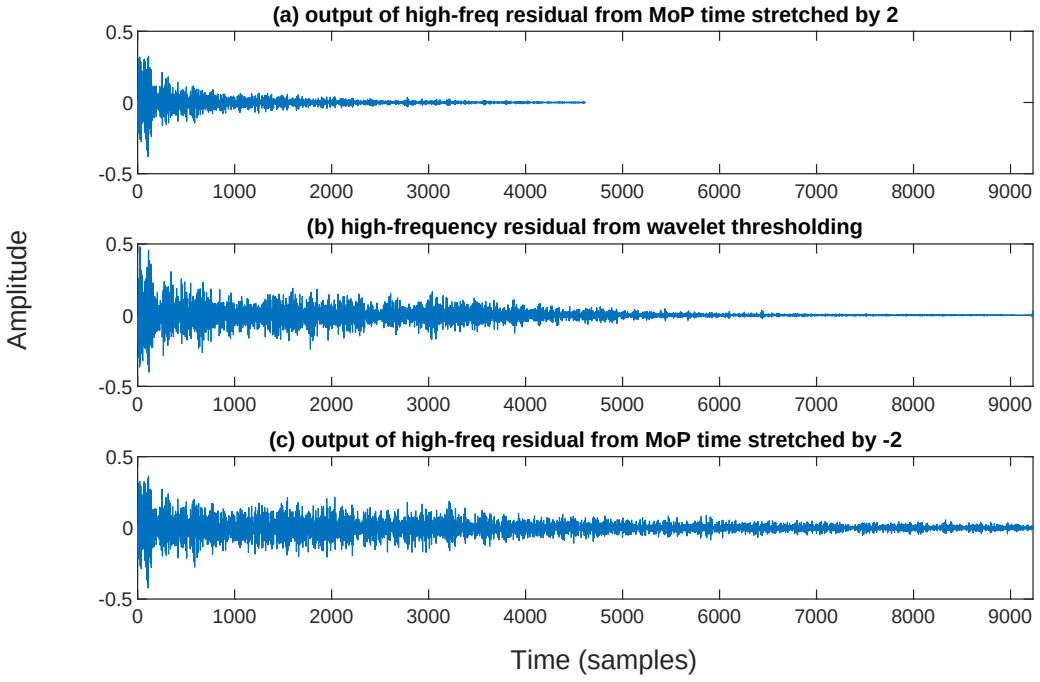


FIGURE 6.37: Snare 2 Time Stretched

Figures 6.38 and 6.30 displays a Raw Sawtooth Waveform in the key of C pitch shifted and time stretched respectively.

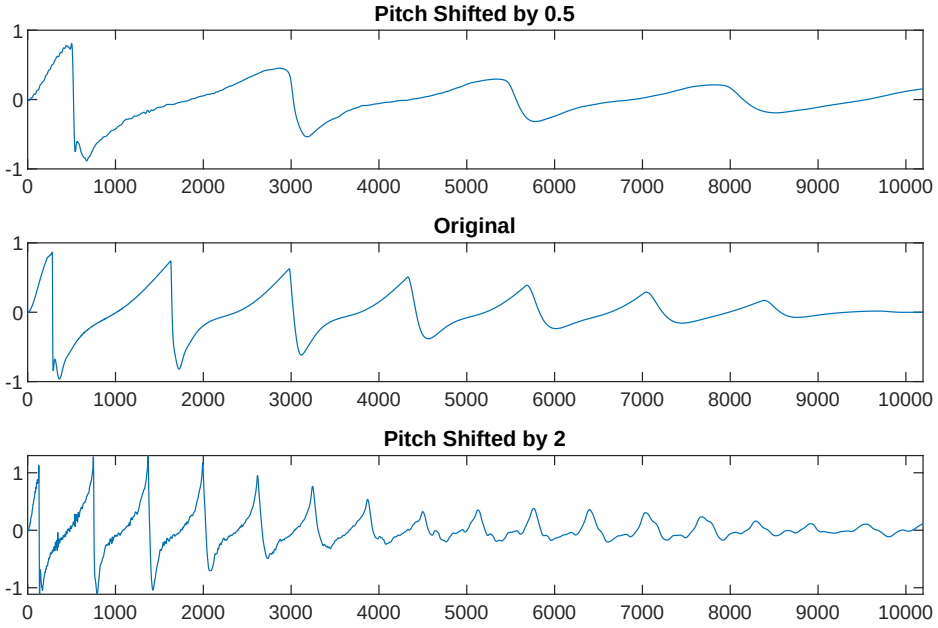


FIGURE 6.38: Raw Sawtooth Bass (@44.1 kHz) Pitch Shifted (@44.1 kHz)

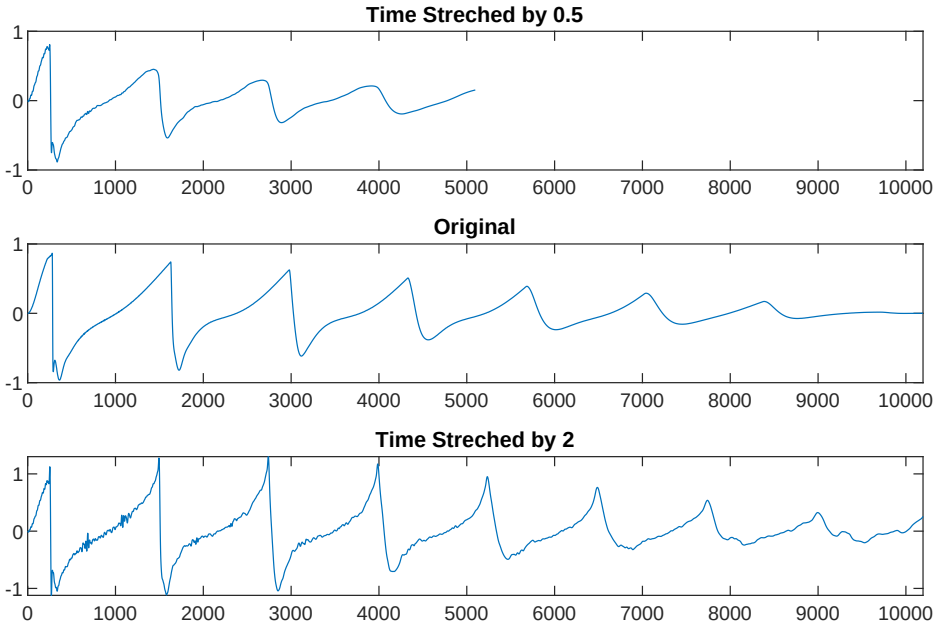


FIGURE 6.39: Raw Sawtooth Bass (@44.1 kHz) Time Stretched

6.4 Conclusion

This chapter has has presented an overview of two potential system implementations, and demonstrated a system for spectral modelling of audio that implements analysis methods developed in the previous three chapters. An overview of the systems performance in terms of quality has been assessed through the use of simple and complex audio examples. Details from Chapters 3, 4 and 5 have been brought together and discussed in detail.

A segmented system incorporating MoP for sinusoidal analysis and modelling of monotonic non-stationary components is combined with a segmented implementation of the undecimated Wavelet transform for modelling the residual signal as transient and noise components. The MoP implementation can use causal or non-causal parameter estimation methods, although the current implementation utilises the non-causal methods derived in Chapters 3 and 4. This does pose a issue when combining a non-causal system with an odd frame size together with the SegUWT which requires an even frame size for the inverse transform. An acceptable workaround for the current framework has been demonstrated and further improvements left as a future research objective.

The current implementation does not implement peak tracking and coherence of amplitude and phase trajectories between frames. As such there are examples of discontinuities between frames when the MoP decomposition is not accurate. It has however been demonstrated that the methods employed for correcting amplitude and frequency biases and for estimating monotonic amplitude and frequency change work very well during the sustain region of a sound. Improvements to model estimates and maintaining coherence of sinusoidal partials between frames during attack regions of a sound require further improvement.

A non-segmented system has also been proposed, aimed at modelling percussive sounds with very short attacks. Sounds with very short attacks have been shown to be modelled well with this approach, and the issues presented in Chapter 4 regarding modelling of non-monotonic amplitude change mitigated by very short attack times.

Chapter 7

Conclusions and Future Work

"One shouldn't take life so seriously.
No one gets out alive anyway."

Jim Morrison [265]

7.1 Conclusion

One of the aims of this thesis was to investigate modelling kick and bass sounds synthesized for electronic dance music in real-time from single frame, non-stationary sinusoidal estimation methods. There sounds are characterised by the way the frequencies and amplitude evolve over time and so it is of interest to model these non-stationary characteristics. The amplitude envelopes of these sounds can be shaped in numerous ways. Linear, piecewise linear, and exponential being common options, but logarithmic, s-curve and other variations of these where you can adjust the slope of the curve are also available as shown in Figure 2.8. Exponential decaying sinusoids and linear interpolation of amplitude between successive frames are the two most common methods of modelling amplitude change. Incorporating both of these two types of curves into a sinusoidal model, and distinguishing between them allows for a more flexible and accurate model.

A formula for the identification and description of linear intra-frame amplitude change from first order phase difference measurements has been presented in Chapter 3.

A method for distinguishing between linear and exponential amplitude change within a single frame is also presented from the examination of the effect that equal levels of energy change, applied by these two curves, has on the magnitude spectrum. The extension of existing phase distortion based methods for providing estimates of exponential intra-frame amplitude change to include estimates of linear intra-frame amplitude change, provides an adaptive model which incorporates both of these curve types. Modelling of other curve types is open to further investigation, as is the estimation of exponential curves with adjusted slopes. The inclusion of linear amplitude change was shown to improve parameter estimates and limit biases when presuming only exponential amplitude change, in the presence of linear change. Modelling of both linear and exponential monotonic amplitude change within a single frame has been presented and shown to provide a framework for applying audio transformations on a sample by sample basis from non-overlapping single frame analysis.

Chapter 4 introduced a modified approach to the popular Matching Pursuit algorithm for the atomic decomposition of non-stationary sinusoial components with monotonic (linear or exponential) amplitude change. This approach was then adapted from its original use in the modal decomposition of impulse responses to model transient components in an overcomplete representation. Chapter 5 presented a general approach for implementing the SWT in a segmented framework and dealing with delays, the number of filter states and number of samples required for overlapping segments in an overlap save approach for dealing with block end artifacts due to convolution and the up-sampling of filter coefficients at each decomposition level. Popular de-noising techniques were then explored for separating transient components from noise in the residual signal. Further research into wavelet filter types, the filter order and the decomposition depth, is required for achieving optimal separation.

In this work, we have presented a single frame approach in a spectral modelling framework, with application of modelling kick and bass sounds. Although the focus of the thesis has been on modelling kick and bass, it has been shown that an overcomplete sinusoidal model can accurately model a wide range of signals, including professionally produced and released dance music with multiple components and polyphonic elements. Initially single frame estimation methods based on phase distortion were extended for estimating and distinguishing between monotonic linear and exponential amplitude changes with the intention of modelling transient components from the residual signal. The undecimated SWT was then investigated and implemented in a segmented framework, with a general solution to dealing with convolution based block end artifacts presented. This shift invariant transform has been shown to be successful at separating low level transients remaining in the residual after utilising an overcomplete sinusoidal model derived from MoP.

Modelling a sinusoids with non-monotonic amplitude changes is possible using and over-complete decomposition usign modelled pursuit to recreate the non-monotonic amplitude change from combining atoms where constructive and destructive interference of the sinusoidal signals results in the restoration of the correct amplitude envelope. MoP has been shown to model transient and other complex sounds. It has also been demonstrated that although this over-compete model is highly accurate in modelling the original sound, there are some fundamental areas requiring further investigation for accurately performing pitch and time scale modifications.

Residual modelling using the segmented undecided SWT has been presented. This introduces extra latency due to the convolution and upsampling of the filter coefficients at each decomposition level but is implemented in a segmented frame by frame architecture. Transient components are shown to be separable from noise via well established de-noising techniques developed due to the desirable property of shift invariance that this transform offers. This method of modelling transients within the residual also presents interesting challenges with regards to time and pitch scale modifications as well as performing transient manipulations.

Chapter 6 presented a system overview. Details on parameter bias correction were presented, including a novel method for estimating linear frequency from the second order phase difference measure across a spectral peak. The second order phase difference measure across a peak is based on the curvature of the zero-phase padded phase spectrum across a spectral peak. This method provided an accurate estimation method which was robust to initial phase conditions.

A mismatch between a non-causal sinusoidal framework requiring an odd frame size and the inverse SWT requiring a frame size divisible by 2^{level} was investigated and a possible workaround presented. Further improvements to resolving this are left as a future improvement to the system.

The following Section highlights some of a many possible further directions and options available for working towards a sinusoidal model which is capable of modelling and manipulating kick and bass sounds while retaining the quality.

A highly important feature of the future scope of work relates to how well the quality of the 'low-end' separation stated in Chapter 1 is maintained after time, pitch and transient manipulations, and how best to maintain this quality.

7.2 Future Research Directions

One of the initial aims outlines at the beginning of the thesis was to work towards a real-time framework. Parallelization of the system as a whole is important for achieving real-time performance. Completing the implementation of MoP on a GPU is a future research item aimed at achieving real-time performance.

The use of de-noising for the separation of transient components remaining in residual signal from noise was explored. This also requires further investigation on how best to achieve the optimal level of separation through the combination of wavelet filters, the filter order, decomposition depth, and is also left as a future research direction.

'Guided Modelled Pursuit', is a possible future adaptation of MoP. Initial tests in creating additional dictionaries with slightly random parameter estimates with a skewed distribution targeted in the direction of achieving an improvement in the accuracy of parameter estimates shows some promise, although at a much higher computational cost than using the base atom directly, or the above mentioned iterative decomposition using a normalised distribution. Further work is required for fully exploring the potential of this method. A metric for measuring the amount of accuracy gained in comparison to the computational cost of the extra processing incurred is left as a future improvement to the system.

The non-segmented system presented in Chapter 6 was shown to model transient components in an over-complete decomposition. Time and pitch scale modifications were successfully applied to these signals with very short attack times. Modelling transients as sinusoidal component in an over-complete decomposition, or excluding them from the sinusoidal part of the model requires further analysis and qualitative testing, leaving it as a future research directive.

A multi-resolution analysis system, including onset detection and alignment of analysis frames to transient events is an important future improvement. The addition of a pre-analysis stage and alignment of analysis and synthesis regions is a desirable future directive.

The current use of the second order phase difference measure for estimating frequency change has unique values until a certain amount. For a 1025 analysis frame, this is between -300 and $+300$ Hz. Extending the model to use higher frequency change estimates is left as a future directive.

The current system would benefit from a peak continuation algorithm which interpolates the amplitude and phase of tracked partials between frames, and so is left as a future improvement to the current implementation.

Finally, the workaround investigated to overcome the mismatch between a non-causal spectral modelling system and the odd frame size required for zero-phase padding, in conjunction with the UWT which requires a frame size divisible by 2^{level}, was acceptable, but not optimal. The extension of the inverse UWT in a non-causal framework is left as a current limitation and area for future improvement.

Appendix A

Plots

A.1 Audio Examples

A.1.1 Kick Examples

FIGURE A.1: Amplitude envelope of kick drum [8]

FIGURE A.2: Amplitude envelope of kick drum [8]

FIGURE A.3: Amplitude envelope of kick drum [8]

FIGURE A.4: Amplitude envelope of kick drum [8]

FIGURE A.5: Amplitude envelope of kick drum [8]

A.1.2 Bass Examples

(A) Amplitude Envelope

(B) Pitch Envelope

FIGURE A.6: Kick2 VST for shaping (A) Amplitude and (B) Pitch Envelope of synthesised Bass

(A) Amplitude Envelope

(B) Pitch Envelope

FIGURE A.7: Kick2 VST for shaping (A) Amplitude and (B) Pitch Envelope of synthesised Bass

A.1.3 Snare Examples

(A) Amplitude Envelope

(B) Pitch Envelope

FIGURE A.8: Kick2 VST for shaping (A) Amplitude and (B) Pitch Envelope of synthesised Snare Drum

(A) Amplitude Envelope

(B) Pitch Envelope

FIGURE A.9: Kick2 VST for shaping (A) Amplitude and (B) Pitch Envelope of synthesised Kick Drum

A.2 Synthetic Non-Stationary Testing

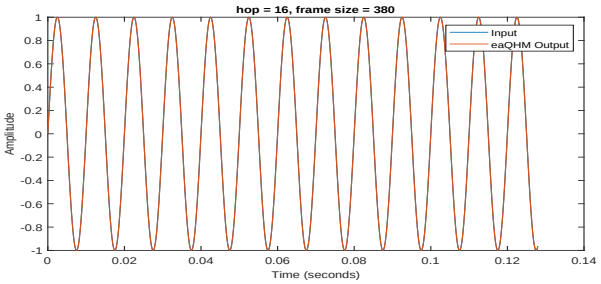
Explanations of the signals used for testing in the paper on "Adaptive Modeling of Synthetic Nonstationary Sinusoids" including the formulas are available in [142].

A.2.1 Expansion of signal abbreviations

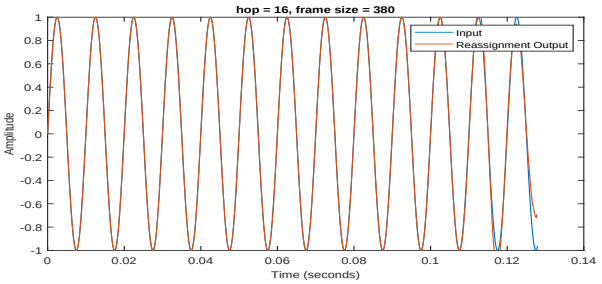
1. Constant Amplitude Linear Phase (CA-LP)
2. Exponential Amplitude Linear Phase (EA-LP)
3. Constant Amplitude Cubic Phase (CA-C3P)
4. Exponential Amplitude Cubic Phase (EA-C3P)
5. Linear Amplitude Cubic Phase (LA-C3P)
6. Cubic Amplitude Cubic Phase (C3A-C3P)
7. Sinusoidal Amplitude Sinusoidal Phase (SA-SP)
8. Exponential Amplitude Sinusoidal Phase (ESA-SP)
9. Exponential Amplitude Quadratic Phase (EA-QP)
10. Exponential Second Order Amplitude Constant Phase (EA-NM)
11. Linear Second Order Amplitude Constant Phase (LA-NM)
12. Exponential Amplitude Sawtooth (EA-SAW)

A.2.2 Plots of results using default setting

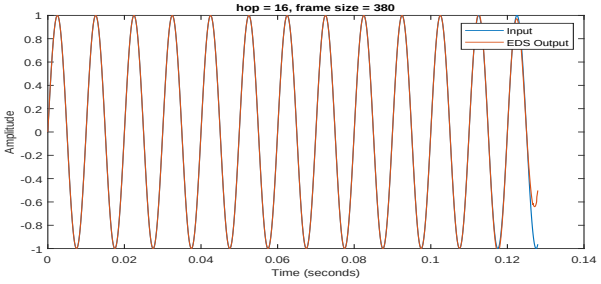
1. Constant Amplitude Linear Phase (CA-LP):



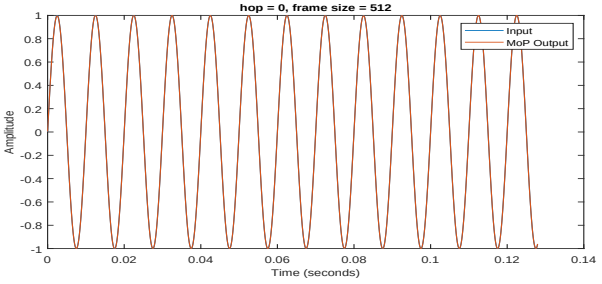
(A) Constant Amplitude Linear Phase eaQHM



(B) Constant Amplitude Linear Phase Reassignment



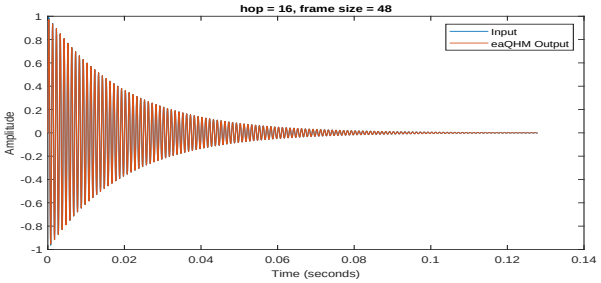
(C) Constant Amplitude Linear Phase EDS



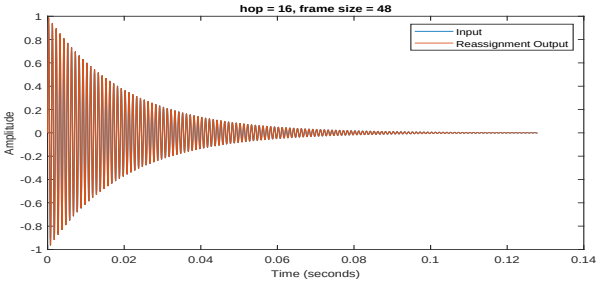
(D) Constant Amplitude Linear Phase MoP

FIGURE A.10: Constant Amplitude Linear Phase (CA-LP) defaults pitch related

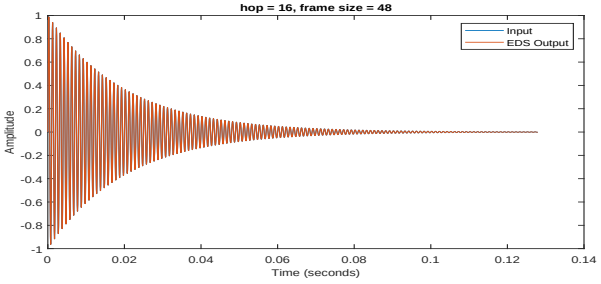
2. Exponential Amplitude Linear Phase (EA-LP):



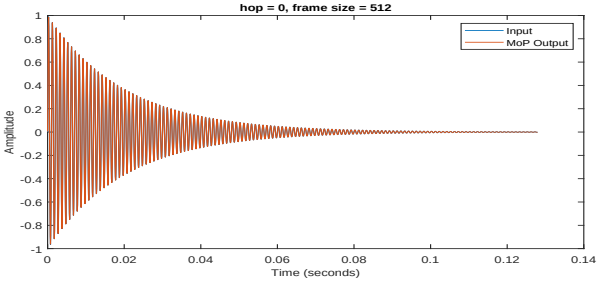
(A) Exponential Amplitude Linear Phase eaQHM



(B) Exponential Amplitude Linear Phase Reassignment



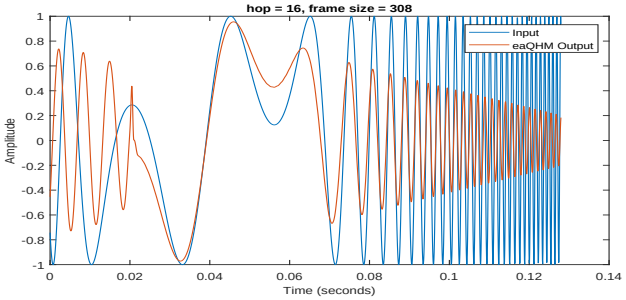
(C) Exponential Amplitude Linear Phase EDS



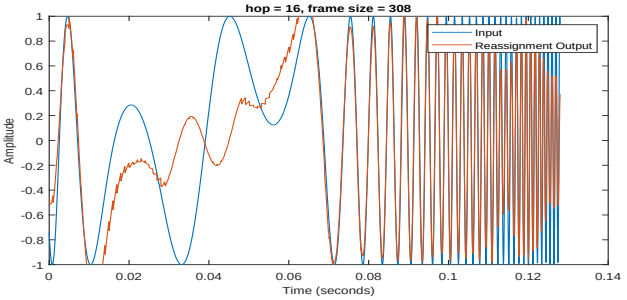
(D) Exponential Amplitude Linear Phase MoP

FIGURE A.11: Exponential Amplitude Linear Phase (EA-LP) defaults pitch related

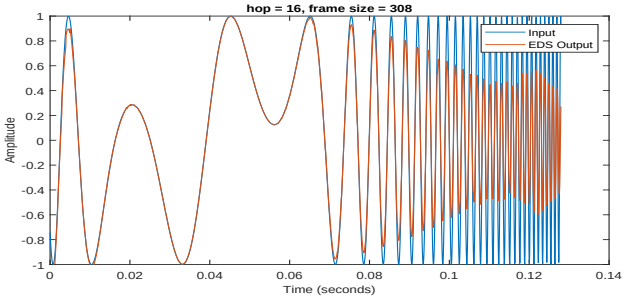
3. Constant Amplitude Cubic Phase (CA-C3P):



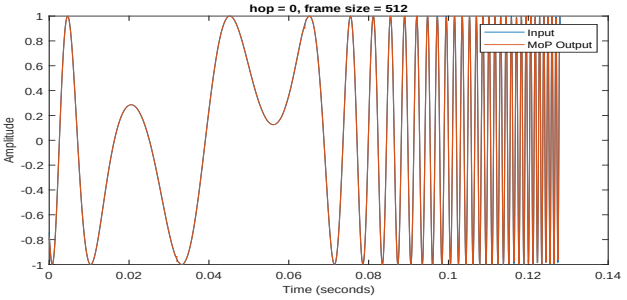
(A) Constant Amplitude Cubic Phase eaQHM



(B) Constant Amplitude Cubic Phase Reassignment



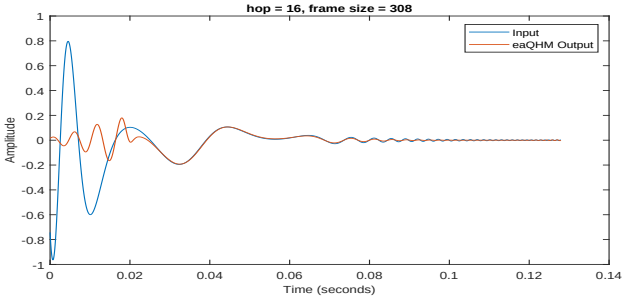
(C) Constant Amplitude Cubic Phase EDS



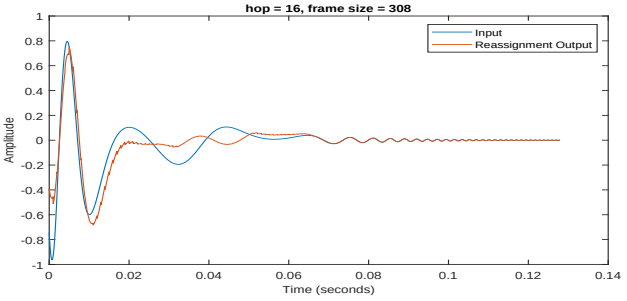
(D) Constant Amplitude Cubic Phase MoP

FIGURE A.12: Constant Amplitude Cubic Phase (CA-C3P) defaults pitch related

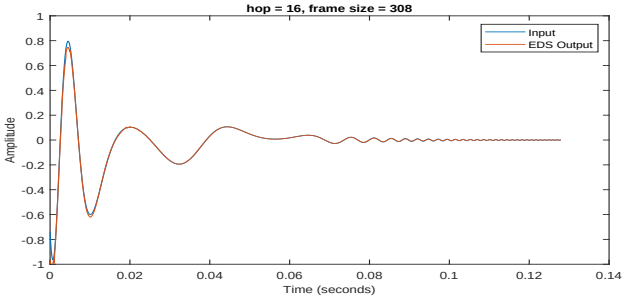
4. Exponential Amplitude Cubic Phase (EA-C3P):



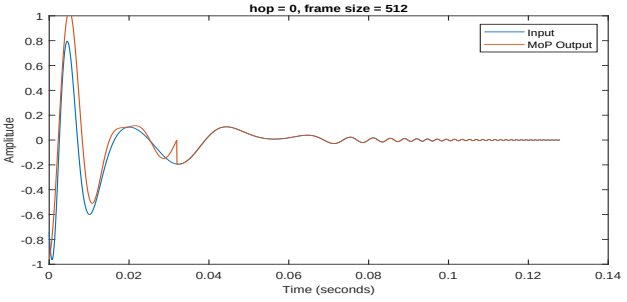
(A) Exponential Amplitude Cubic Phase eaQHM



(B) Exponential Amplitude Cubic Phase Reassignment



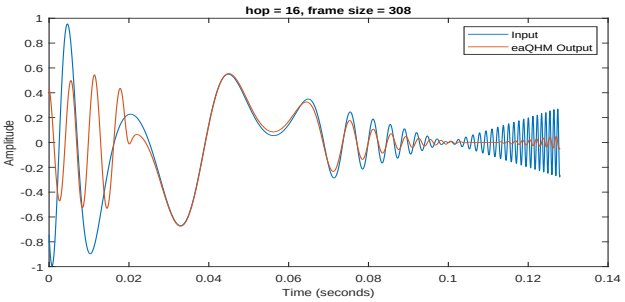
(C) Exponential Amplitude Cubic Phase EDS



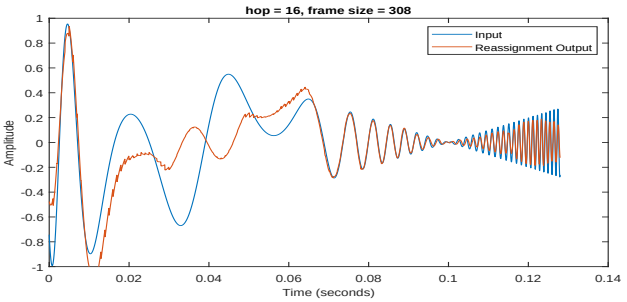
(D) Exponential Amplitude Cubic Phase MoP

FIGURE A.13: Exponential Amplitude Cubic Phase (EA-C3P) defaults pitch related

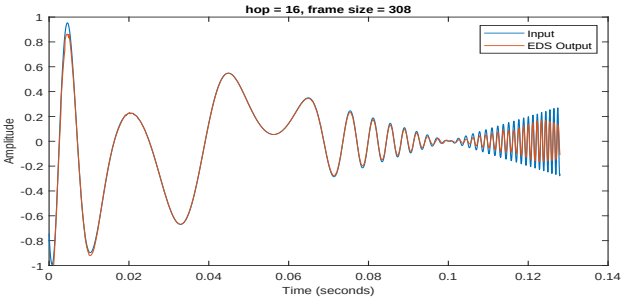
5. Linear Amplitude Cubic Phase (LA-C3P):



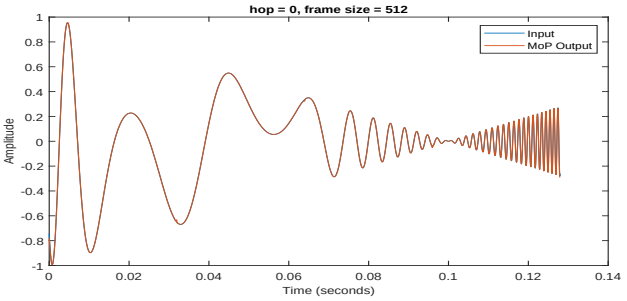
(A) Linear Amplitude Cubic Phase eaQHM



(B) Linear Amplitude Cubic Phase Reassignment



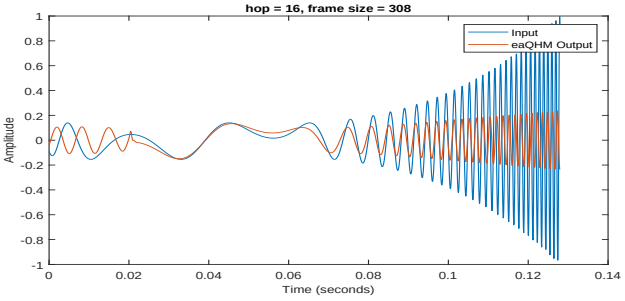
(C) Linear Amplitude Cubic Phase EDS



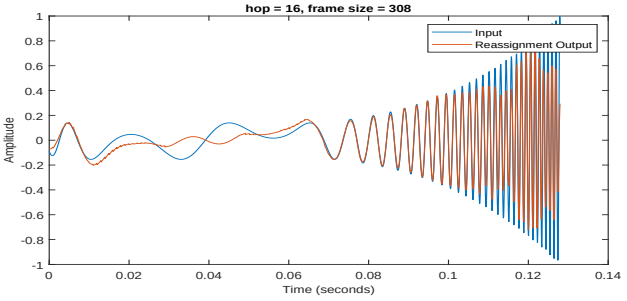
(D) Linear Amplitude Cubic Phase MoP

FIGURE A.14: Linear Amplitude Cubic Phase (LA-C3P) defaults pitch related

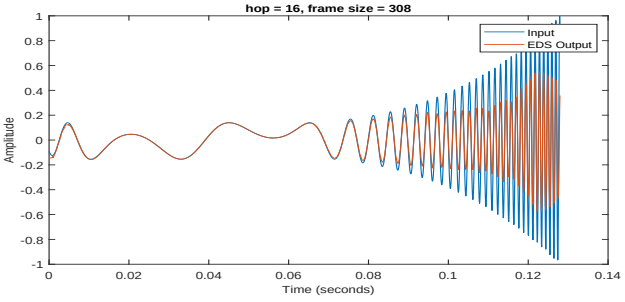
6. Cubic Amplitude Cubic Phase (C3A-C3P):



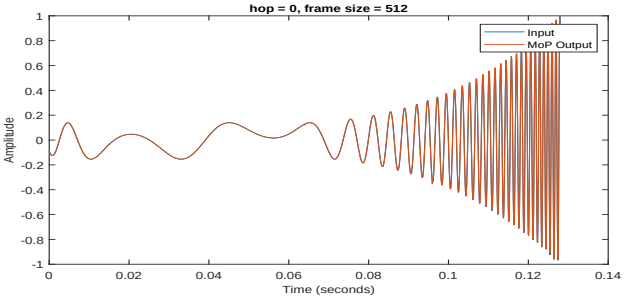
(A) Cubic Amplitude Cubic Phase eaQHM



(B) Cubic Amplitude Cubic Phase Reassignment



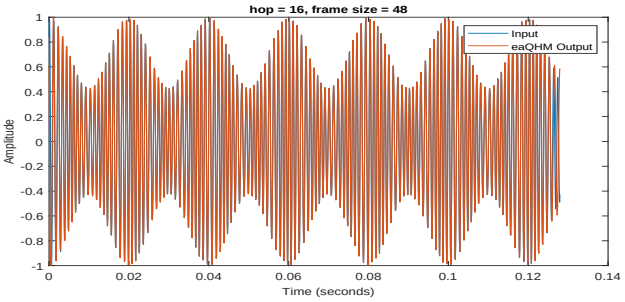
(C) Cubic Amplitude Cubic Phase EDS



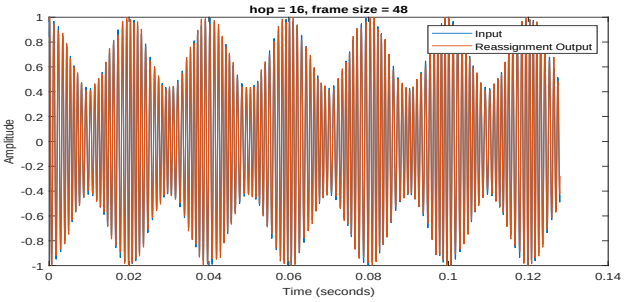
(D) Cubic Amplitude Cubic Phase MoP

FIGURE A.15: Cubic Amplitude Cubic Phase (C3A-C3P) defaults pitch related

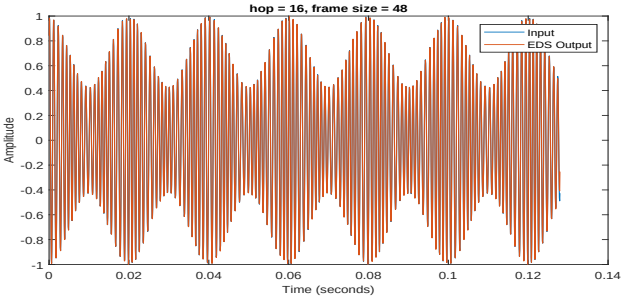
7. Sinusoidal Amplitude Sinusoidal Phase (SA-SP):



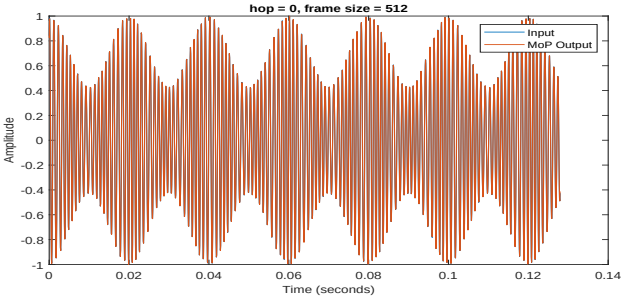
(A) Sinusoidal Amplitude Sinusoidal Phase eaQHM



(B) Sinusoidal Amplitude Sinusoidal Phase Reassignment



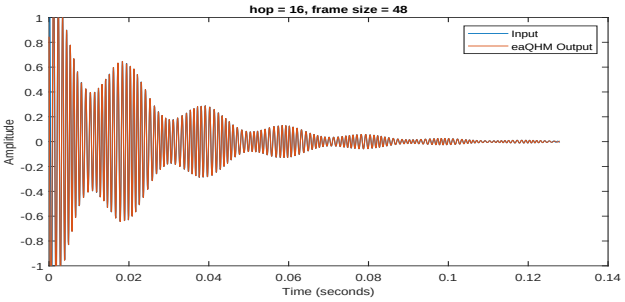
(C) Sinusoidal Amplitude Sinusoidal Phase EDS



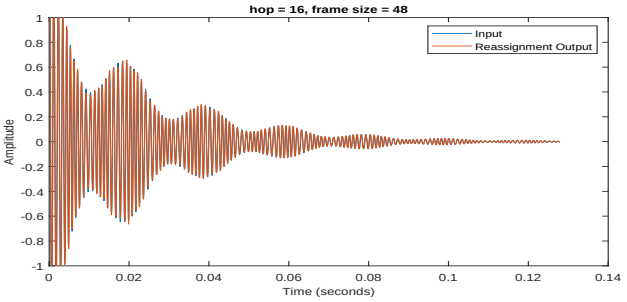
(D) Sinusoidal Amplitude Sinusoidal Phase MoP

FIGURE A.16: Sinusoidal Amplitude Sinusoidal Phase (SA-SP) defaults pitch related

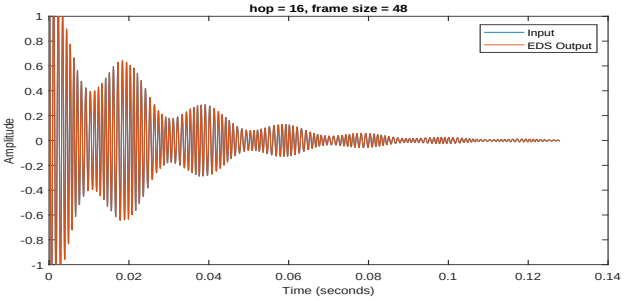
8. Exponentially Damped Sinusoidal Amplitude Sinusoidal Phase (ESA-SP):



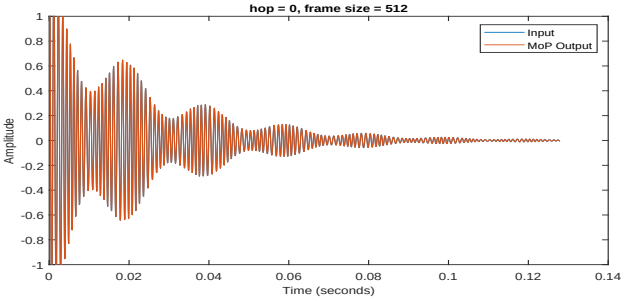
(A) EDS Amplitude Sinusoidal Phase eaQHM



(B) EDS Amplitude Sinusoidal Phase Reassignment



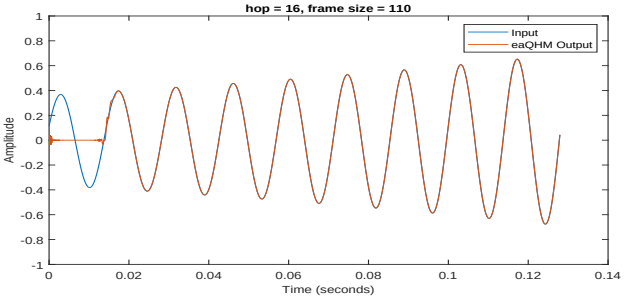
(C) EDS Amplitude Sinusoidal Phase EDS



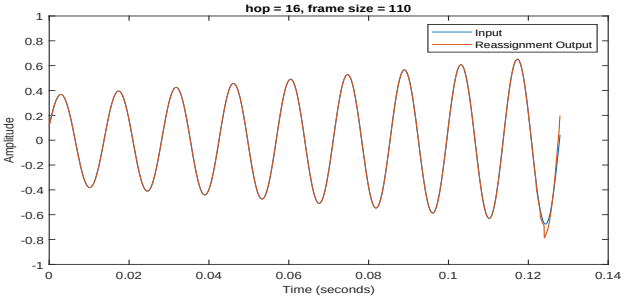
(D) EDS Amplitude Sinusoidal Phase MoP

FIGURE A.17: Exponentially Damped Sinusoidal Amplitude Sinusoidal Phase (ESA-SP) defaults pitch related

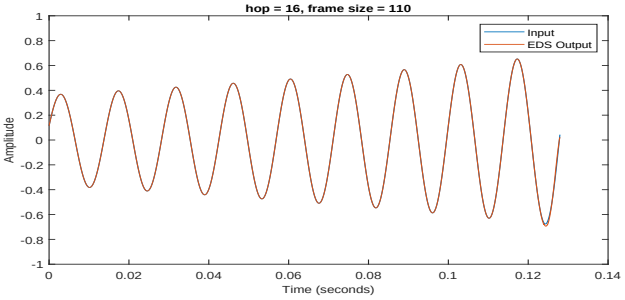
9. Exponential Amplitude Quadratic Phase (EA-QP):



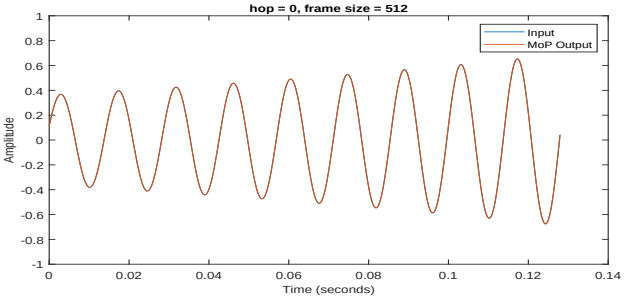
(A) Exponential Amplitude Quadratic Phase eaQHM



(B) Exponential Amplitude Quadratic Phase Reassignment



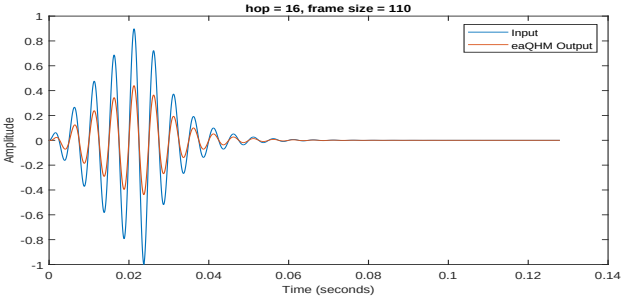
(C) Exponential Amplitude Quadratic Phase EDS



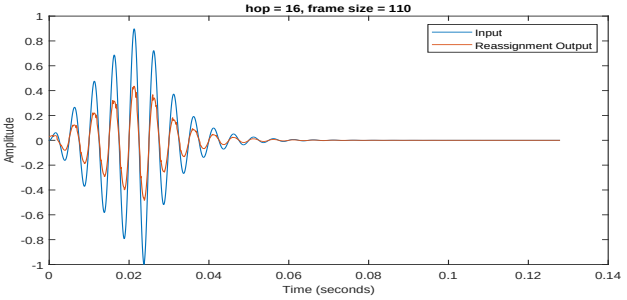
(D) Exponential Amplitude Quadratic Phase MoP

FIGURE A.18: Exponential Amplitude Quadratic Phase (EA-QP) defaults pitch related

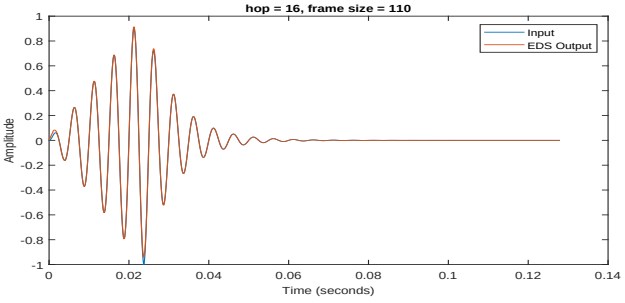
10. Exponential Second Order (non-monotonic) Amplitude Constant Phase (EA-NM):



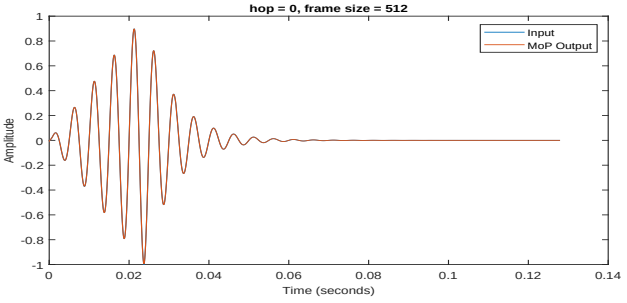
(A) Exponential 2nd Order Amplitude Constant Phase eaQHM



(B) Exponential 2nd Order Amplitude Constant Phase Reassignment



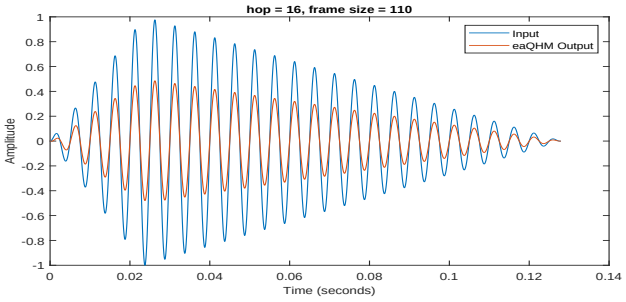
(C) Exponential 2nd Order Amplitude Constant Phase EDS



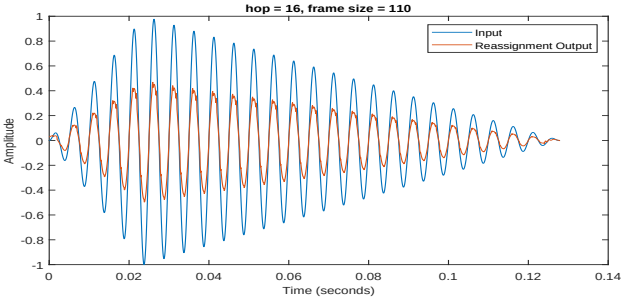
(D) Exponential 2nd Order Amplitude Constant Phase MoP

FIGURE A.19: Exponential Second Order Amplitude Constant Phase (EA-NM) defaults pitch related

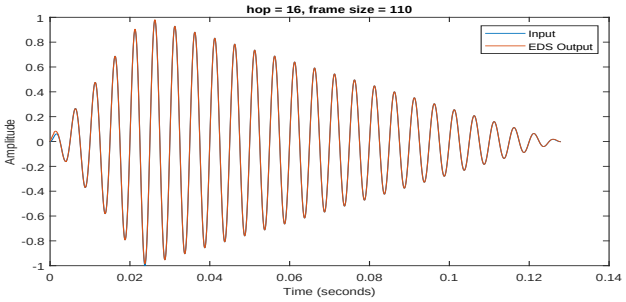
11. Linear Second Order (non-monotonic) Amplitude Constant Phase (LA-NM):



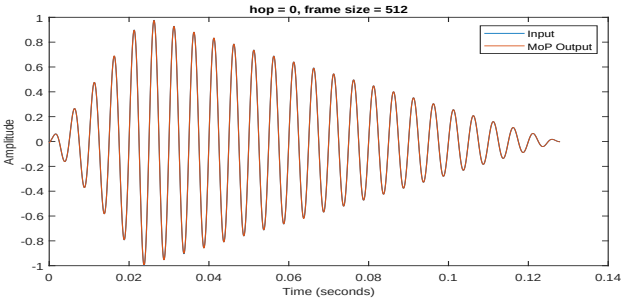
(A) Linear 2nd Order Amplitude Constant Phase eaQHM



(B) Linear 2nd Order Amplitude Constant Phase Reassignment



(C) Linear 2nd Order Amplitude Constant Phase EDS



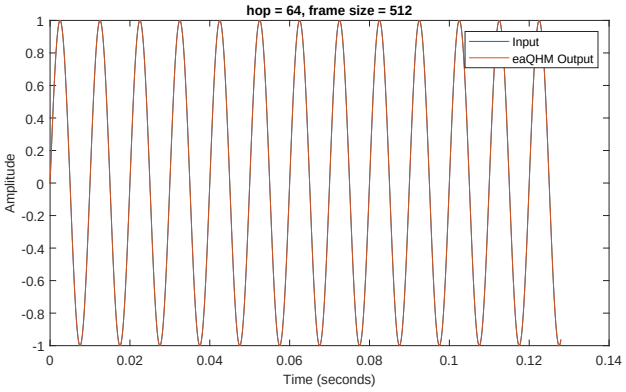
(D) Linear 2nd Order Amplitude Constant Phase MoP

FIGURE A.20: Linear Second Order Amplitude Constant Phase (LA-NM) defaults pitch related

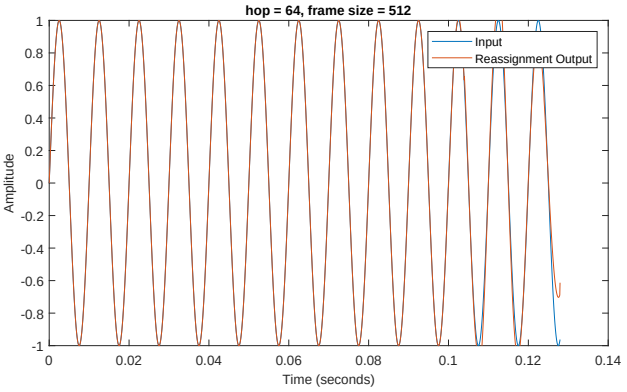
A.2.3 Constant Amplitude Linear Phase (CA-LP)

In general the results for Constant Amplitude Linear Phase (CA-LP) are good for all of the tests using different combinations of hop and frame sizes. The exception to this is in a non-realistic use case, where the frame size was set to match the frame size used by MoP (512 samples). It is known that eaQHM and EDS perform best using small frame sizes, and the quality deteriorates with increasing lengths. [142]

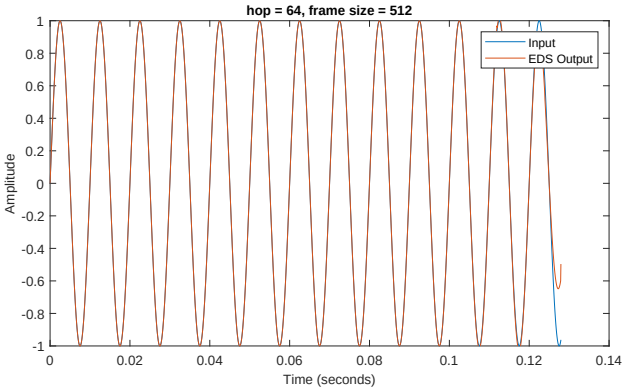
Figure A.21 shows how the model estimates deteriorate from too large a frame size.



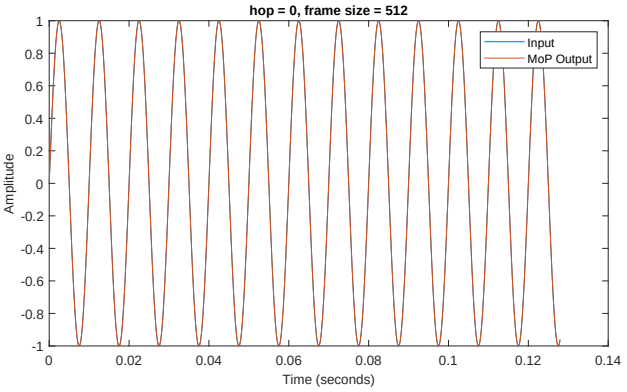
(A) Constant Amplitude Linear Phase eaQHM



(B) Constant Amplitude Linear Phase Reassignment



(C) Constant Amplitude Linear Phase EDS

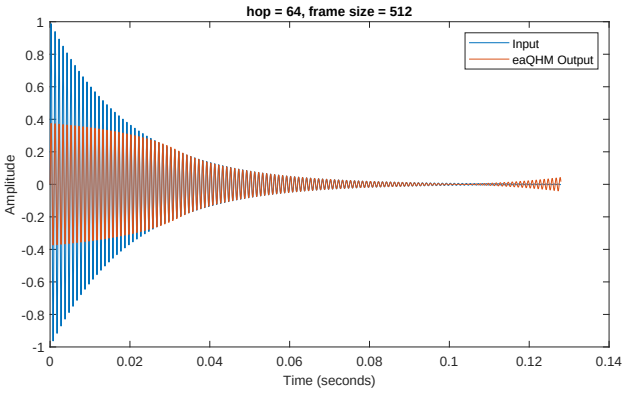


(D) Constant Amplitude Linear Phase MoP

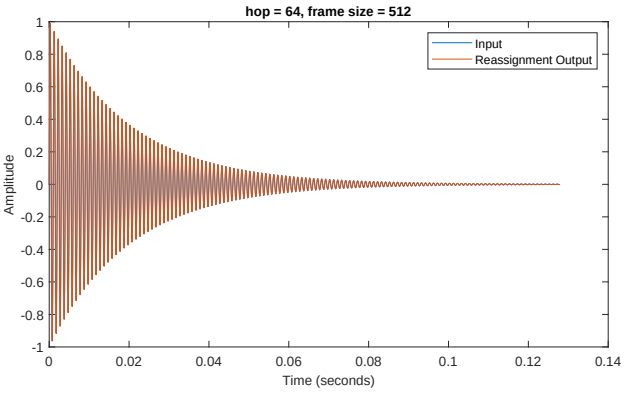
FIGURE A.21: Constant Amplitude Linear Phase (CA-LP) hop=64 frame=512

A.2.4 Exponential Amplitude Linear Phase (EA-LP)

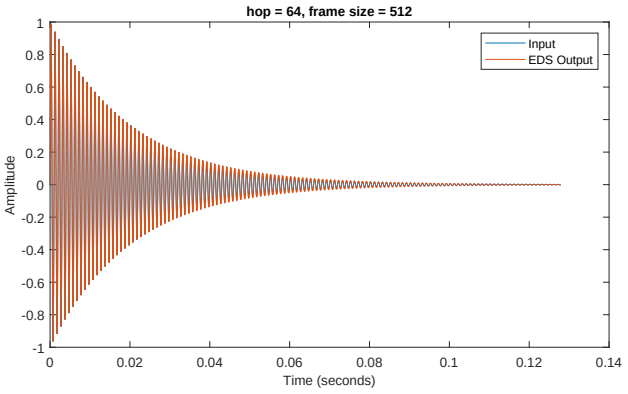
The same is true for Exponential Amplitude Linear Phase (EA-LP), which also performed well under all conditions with the exception of using too large a frame size (512 samples) Figure A.22 shows how the model estimates deteriorate from too large a frame size.



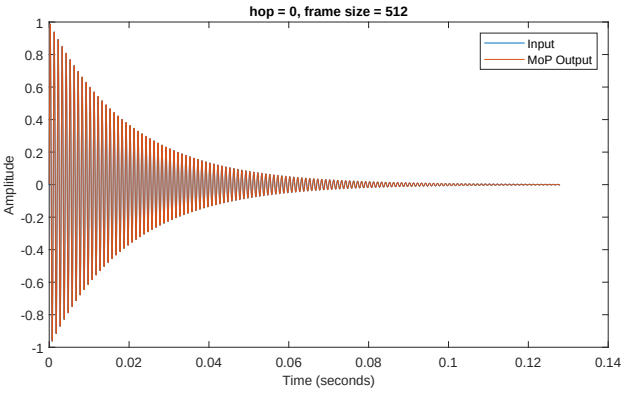
(A) Exponential Amplitude Linear Phase eaQHM



(B) Exponential Amplitude Linear Phase Reassignment



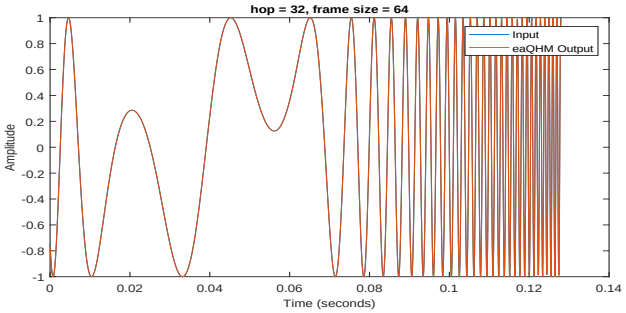
(C) Exponential Amplitude Linear Phase EDS



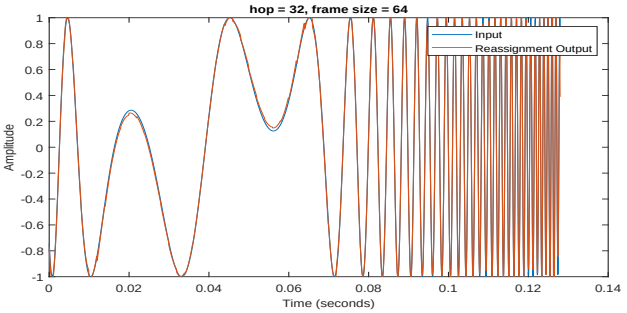
(D) Exponential Amplitude Linear Phase MoP

FIGURE A.22: Exponential Amplitude Linear Phase (EA-LP) hop=64 frame=512

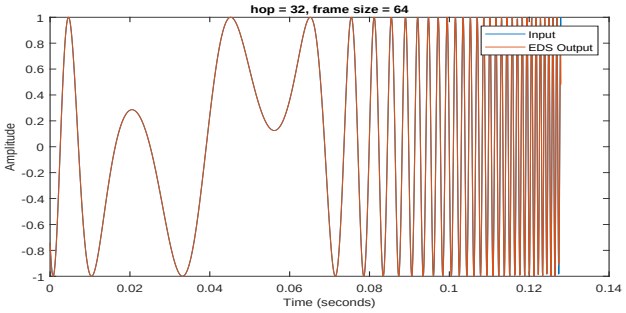
A.2.5 Constant Amplitude Cubic Phase (CA-C3P)



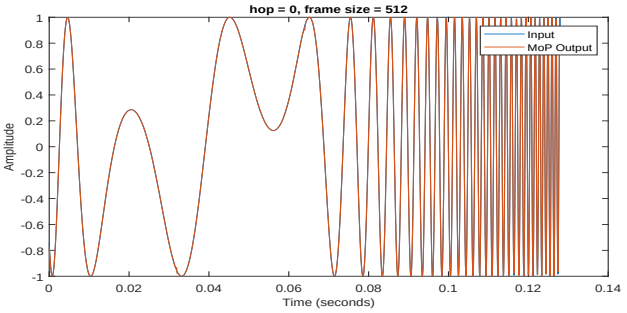
(A) Constant Amplitude Cubic Phase eaQHM



(B) Constant Amplitude Cubic Phase Reassignment

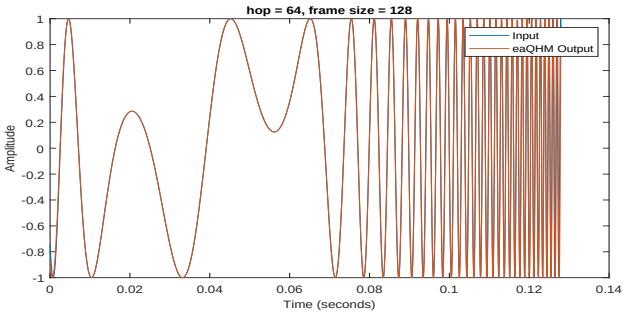


(C) Constant Amplitude Cubic Phase EDS

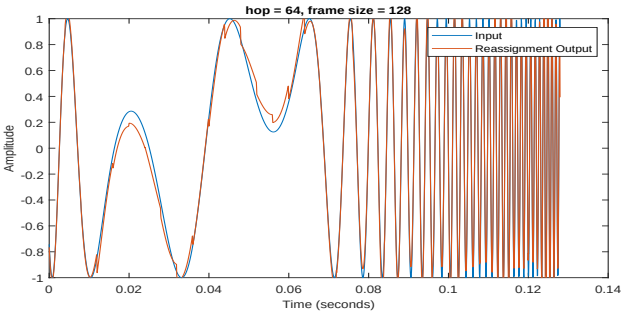


(D) Constant Amplitude Cubic Phase MoP

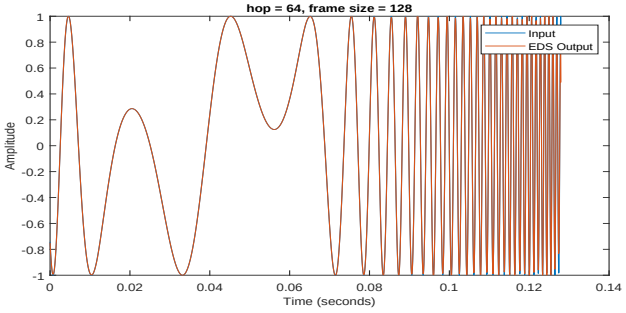
FIGURE A.23: Constant Amplitude Cubic Phase (CA-C3P) hop=32 frame=64



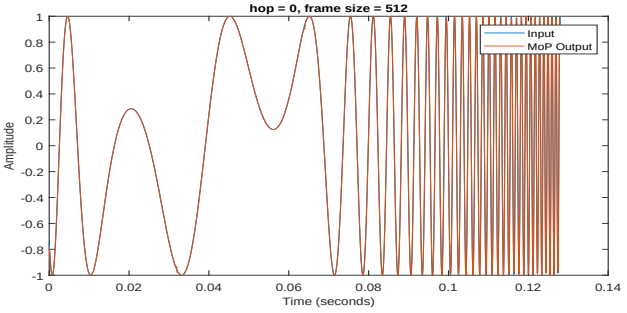
(A) Constant Amplitude Cubic Phase eaQHM



(B) Constant Amplitude Cubic Phase Reassignment

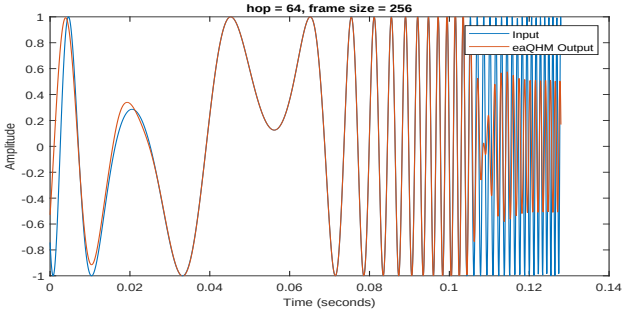


(C) Constant Amplitude Cubic Phase EDS

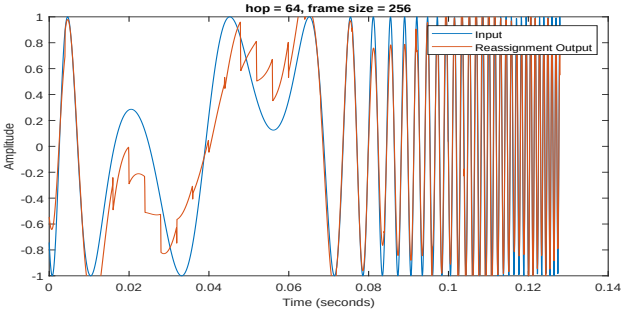


(D) Constant Amplitude Cubic Phase MoP

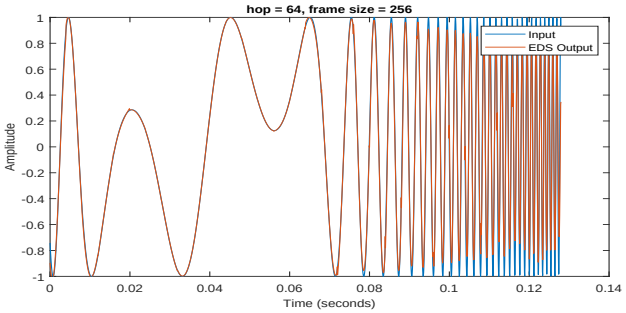
FIGURE A.24: Constant Amplitude Cubic Phase (CA-C3P) hop=64 frame=128



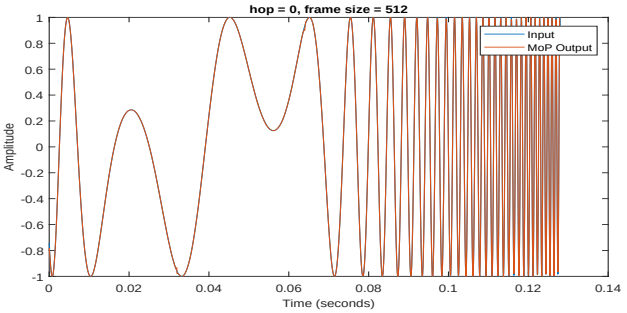
(A) Constant Amplitude Cubic Phase eaQHM



(B) Constant Amplitude Cubic Phase Reassignment

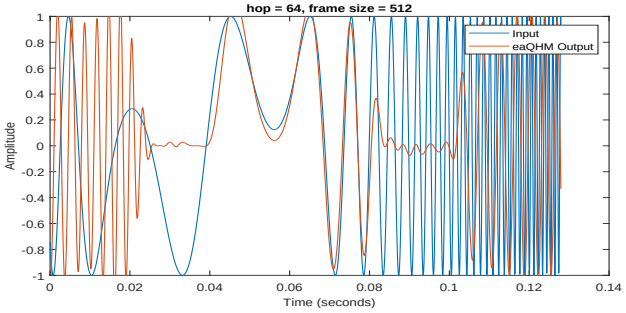


(C) Constant Amplitude Cubic Phase EDS

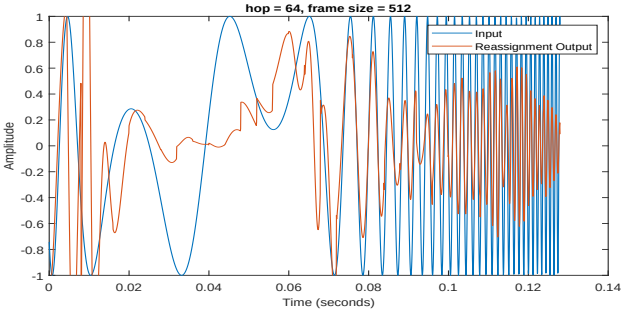


(D) Constant Amplitude Cubic Phase MoP

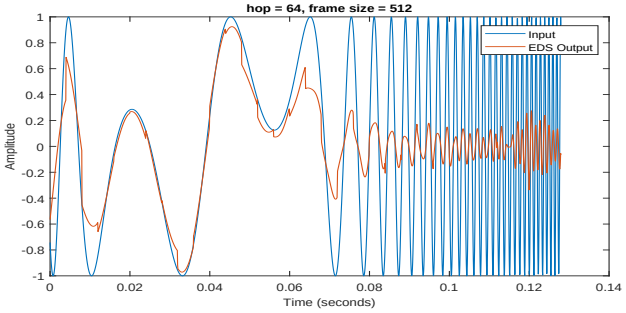
FIGURE A.25: Constant Amplitude Cubic Phase (CA-C3P) hop=64 frame=256



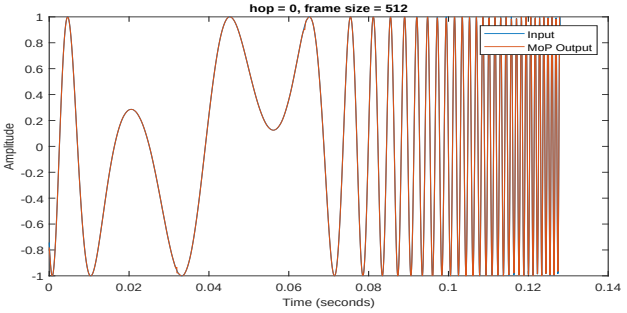
(A) Constant Amplitude Cubic Phase eaQHM



(B) Constant Amplitude Cubic Phase Reassignment



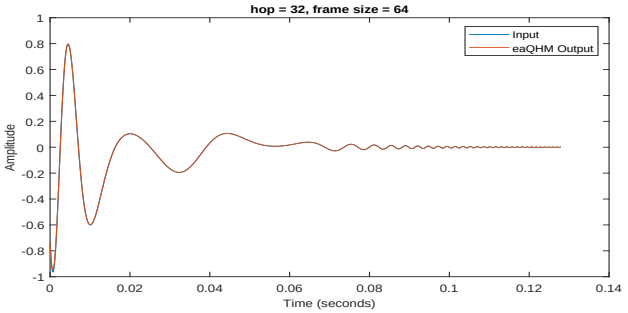
(C) Constant Amplitude Cubic Phase EDS



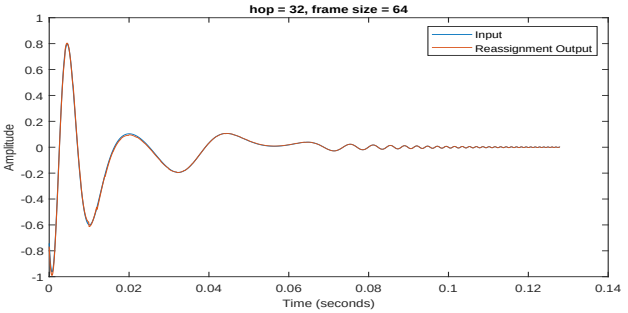
(D) Constant Amplitude Cubic Phase MoP

FIGURE A.26: Constant Amplitude Cubic Phase (CA-C3P) hop=64 frame=512

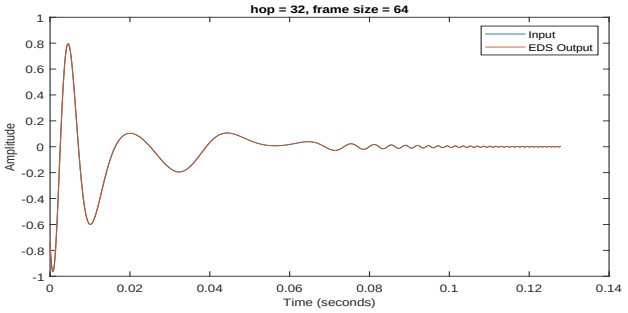
A.2.6 Exponential Amplitude Cubic Phase (EA-C3P)



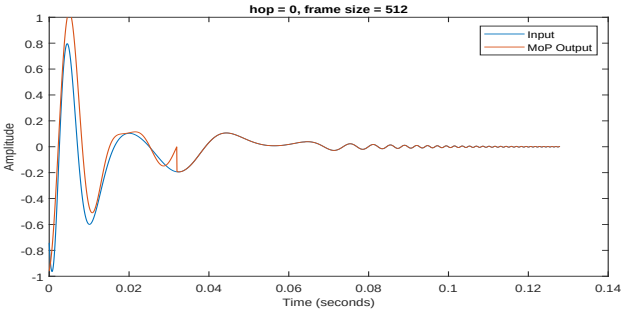
(A) Exponential Amplitude Cubic Phase eaQHM



(B) Exponential Amplitude Cubic Phase Reassignment

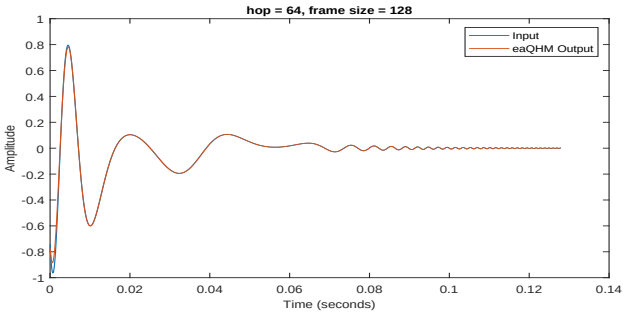


(C) Exponential Amplitude Cubic Phase Reassignment

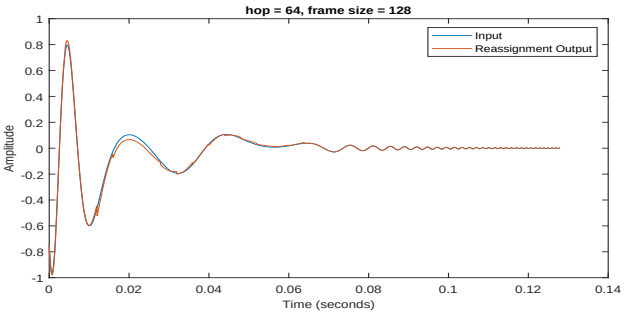


(D) Exponential Amplitude Cubic Phase MoP

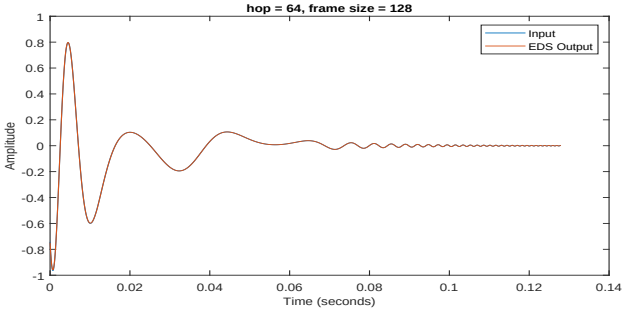
FIGURE A.27: Exponential Amplitude Cubic Phase (EA-C3P) hop=32 frame=64



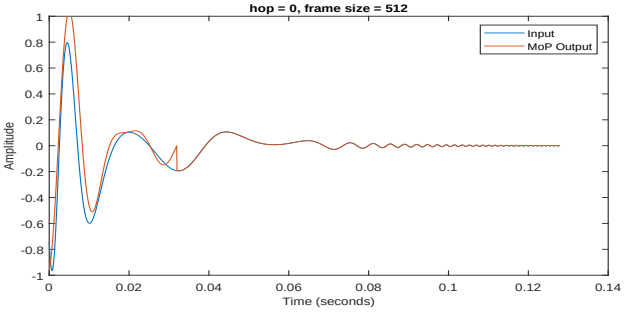
(A) Exponential Amplitude Cubic Phase eaQHM



(B) Exponential Amplitude Cubic Phase Reassignment

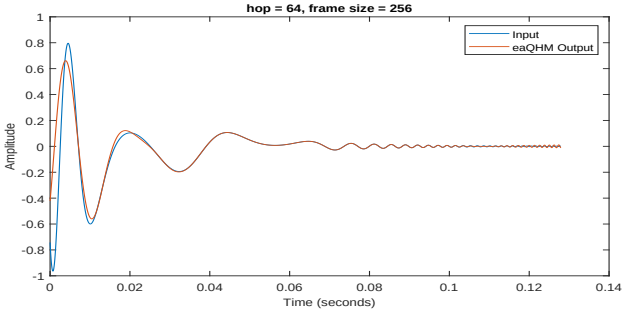


(C) Exponential Amplitude Cubic Phase Reassignment

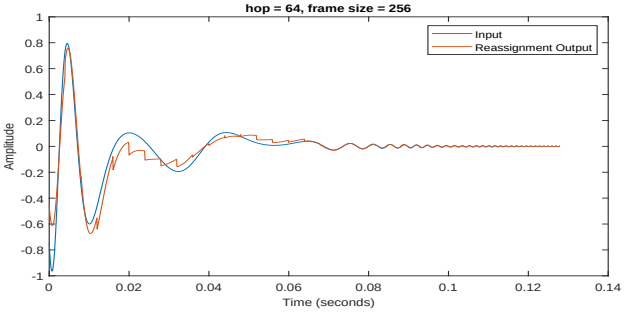


(D) Exponential Amplitude Cubic Phase MoP

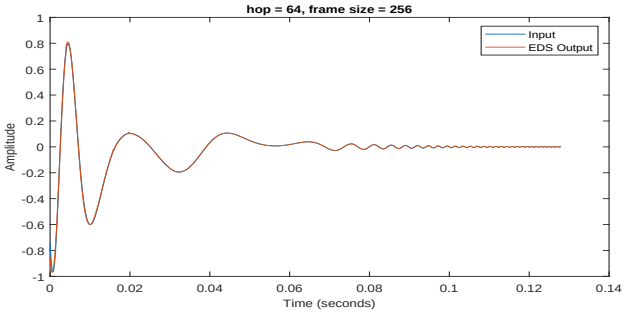
FIGURE A.28: Exponential Amplitude Cubic Phase (EA-C3P) hop=64 frame=128



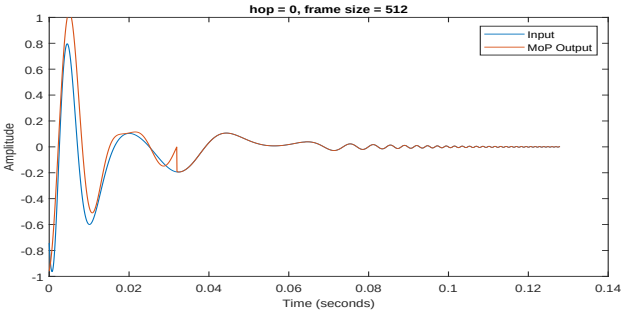
(A) Exponential Amplitude Cubic Phase eaQHM



(B) Exponential Amplitude Cubic Phase Reassignment

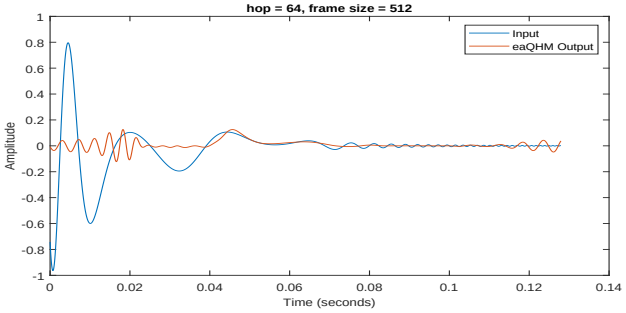


(C) Exponential Amplitude Cubic Phase Reassignment

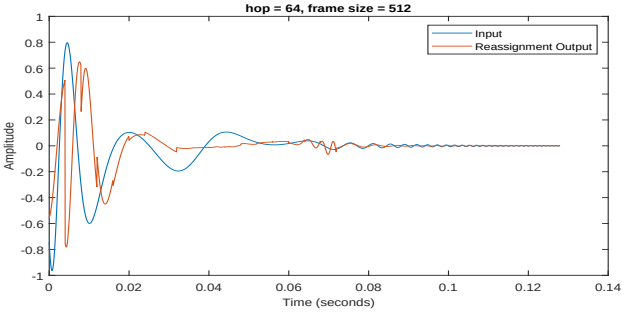


(D) Exponential Amplitude Cubic Phase MoP

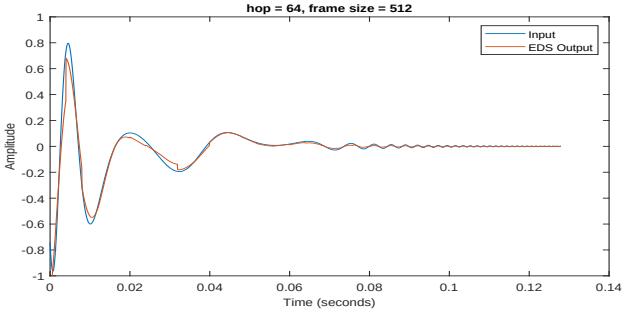
FIGURE A.29: Exponential Amplitude Cubic Phase (EA-C3P) hop=64 frame=256



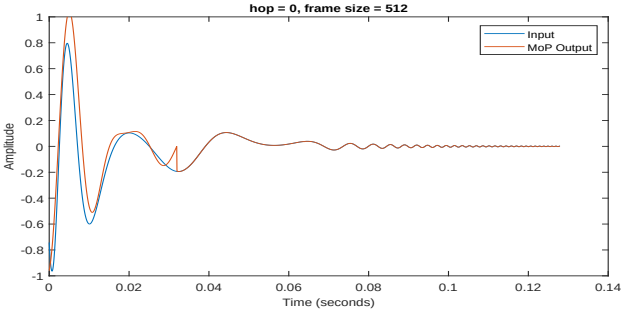
(A) Exponential Amplitude Cubic Phase eaQHM



(B) Exponential Amplitude Cubic Phase Reassignment



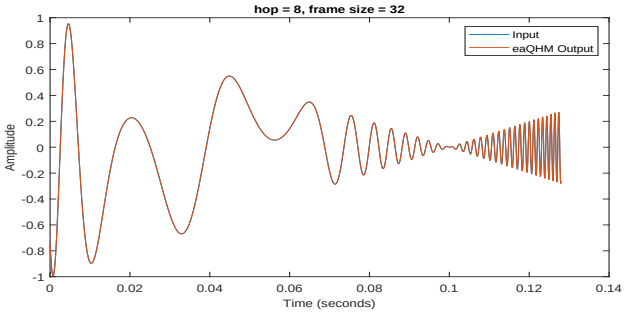
(C) Exponential Amplitude Cubic Phase Reassignment



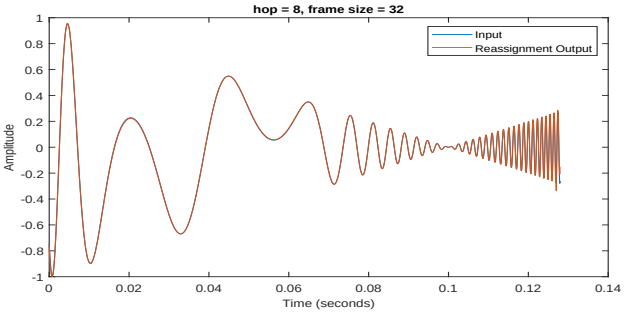
(D) Exponential Amplitude Cubic Phase MoP

FIGURE A.30: Exponential Amplitude Cubic Phase (EA-C3P) hop=64 frame=512

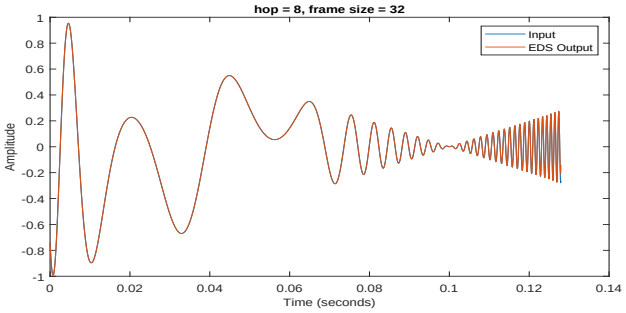
A.2.7 Linear Amplitude Cubic Phase (LA-C3P)



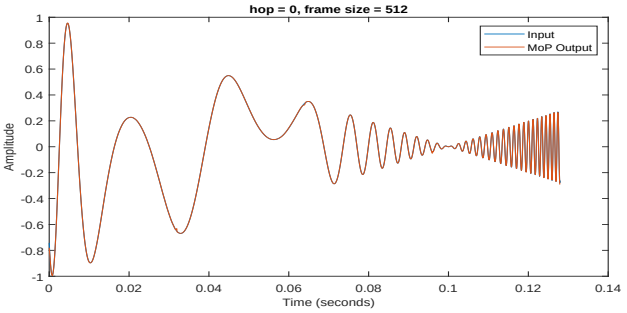
(A) Linear Amplitude Cubic Phase eaQHM



(B) Linear Amplitude Cubic Phase Reassignment

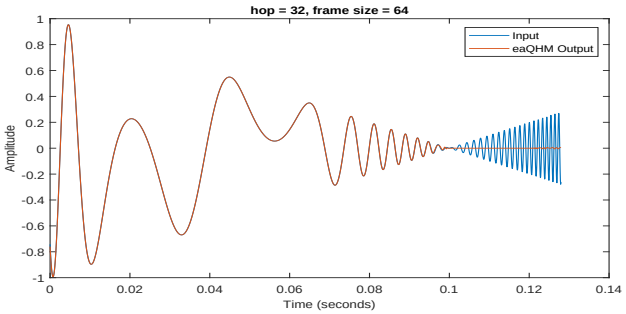


(C) Linear Amplitude Cubic Phase EDS

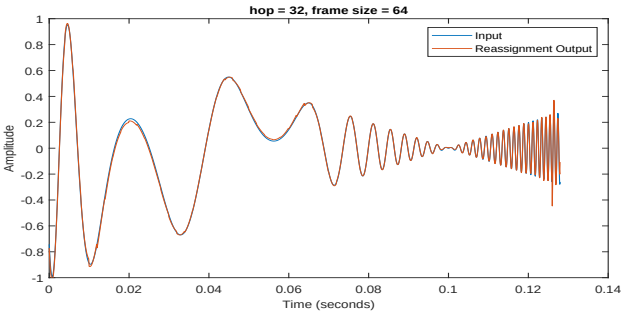


(D) Linear Amplitude Cubic Phase MoP

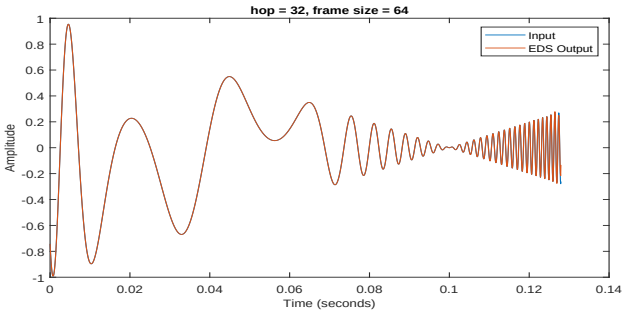
FIGURE A.31: Linear Amplitude Cubic Phase (LA-C3P) hop=8 frame=32



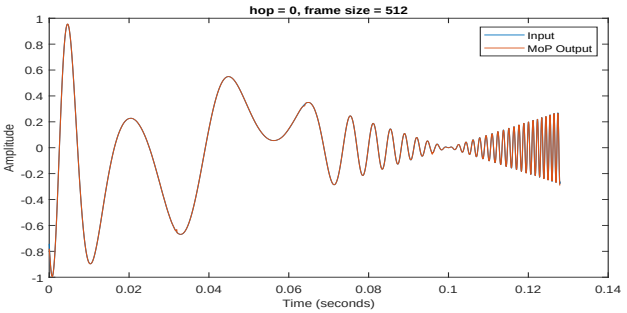
(A) Linear Amplitude Cubic Phase eaQHM



(B) Linear Amplitude Cubic Phase Reassignment

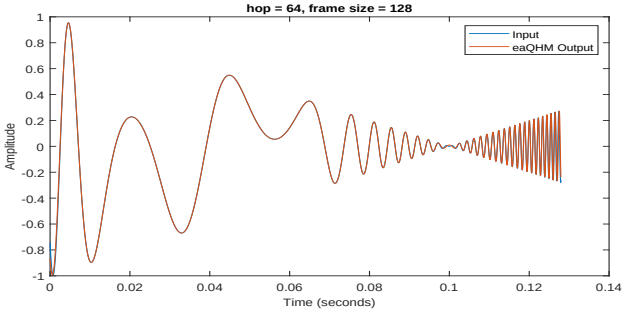


(C) Linear Amplitude Cubic Phase EDS

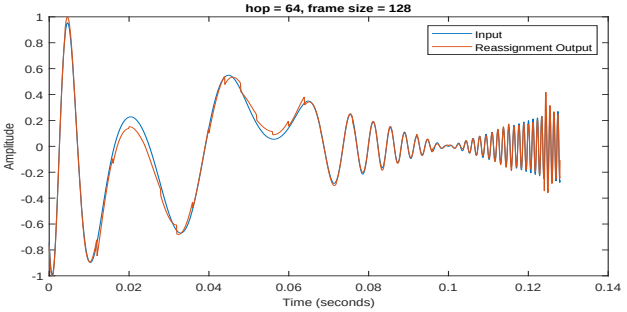


(D) Linear Amplitude Cubic Phase MoP

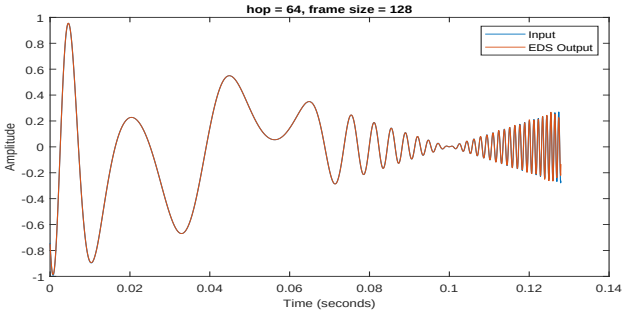
FIGURE A.32: Linear Amplitude Cubic Phase (LA-C3P) hop=32 frame=64



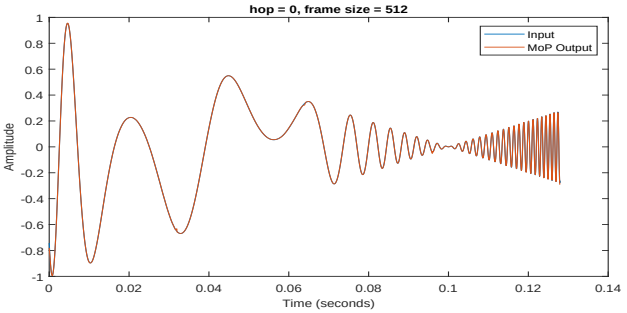
(A) Linear Amplitude Cubic Phase eaQHM



(B) Linear Amplitude Cubic Phase Reassignment

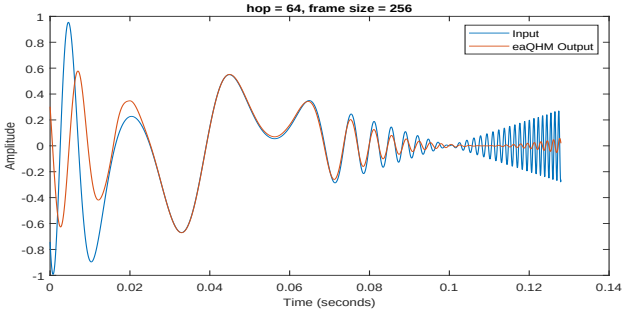


(C) Linear Amplitude Cubic Phase EDS

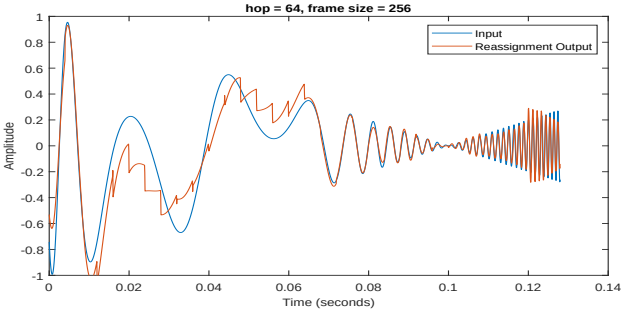


(D) Linear Amplitude Cubic Phase MoP

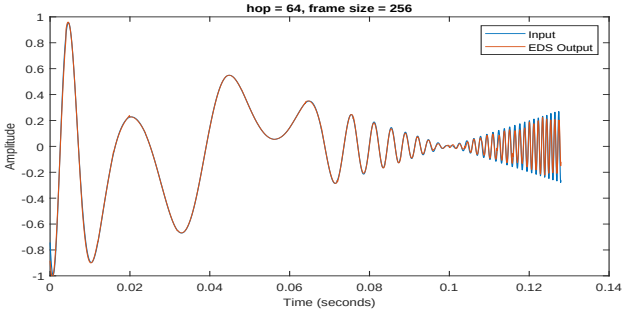
FIGURE A.33: Linear Amplitude Cubic Phase (LA-C3P) hop=64 frame=128



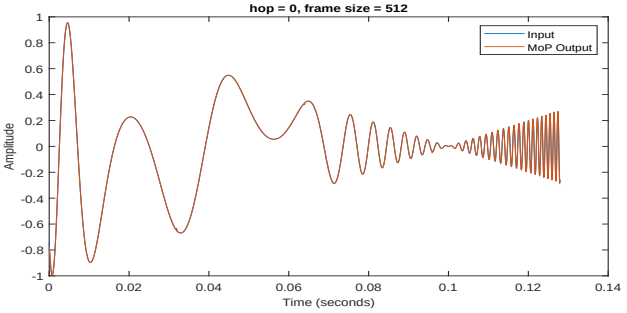
(A) Linear Amplitude Cubic Phase eaQHM



(B) Linear Amplitude Cubic Phase Reassignment

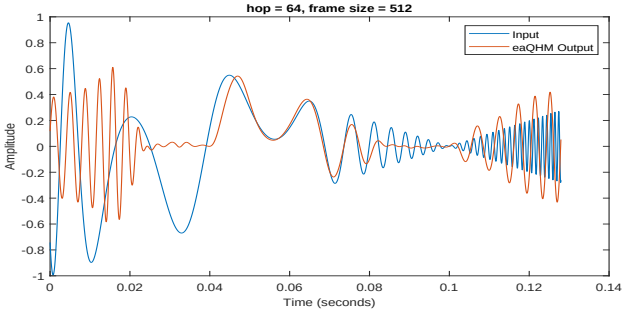


(C) Linear Amplitude Cubic Phase EDS

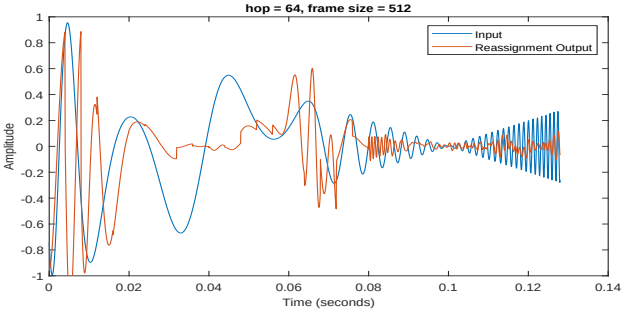


(D) Linear Amplitude Cubic Phase MoP

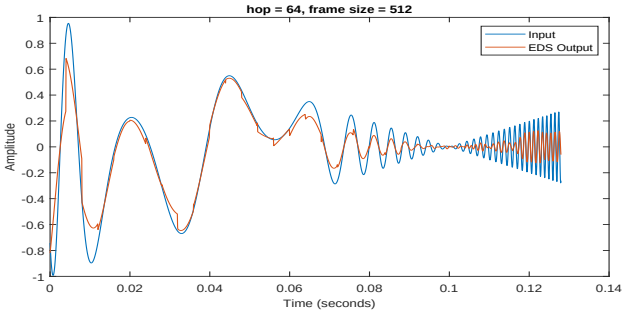
FIGURE A.34: Linear Amplitude Cubic Phase (LA-C3P) hop=64 frame=256



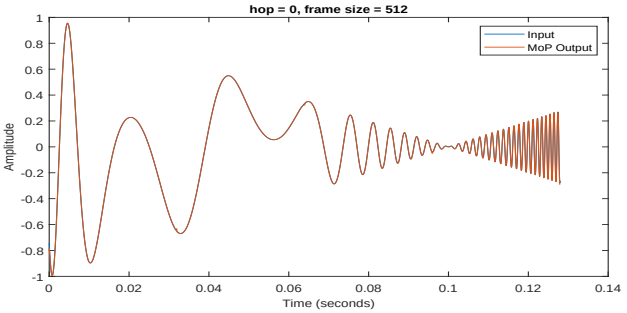
(A) Linear Amplitude Cubic Phase eaQHM



(B) Linear Amplitude Cubic Phase Reassignment



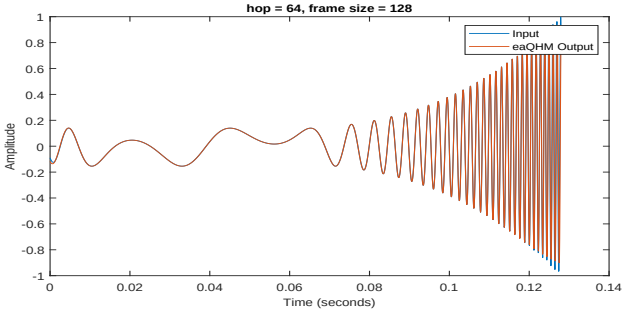
(C) Linear Amplitude Cubic Phase EDS



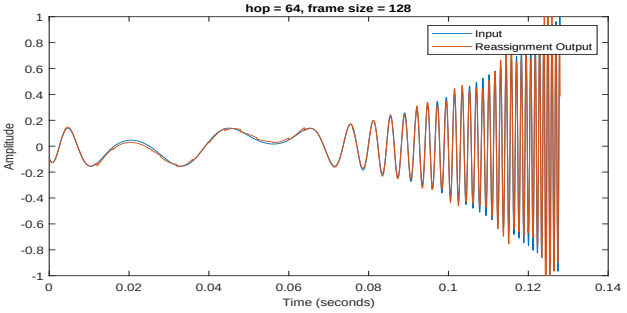
(D) Linear Amplitude Cubic Phase MoP

FIGURE A.35: Linear Amplitude Cubic Phase (LA-C3P) hop=64 frame=512

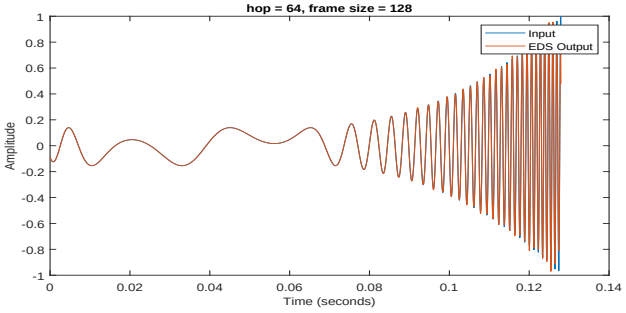
A.2.8 Cubic Amplitude Cubic Phase (C3A-C3P)



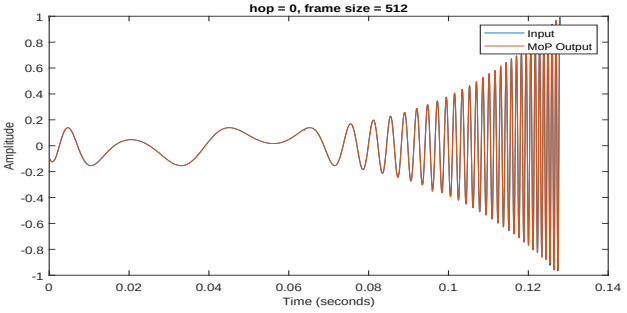
(A) Cubic Amplitude Cubic Phase eaQHM



(B) Cubic Amplitude Cubic Phase Reassignment

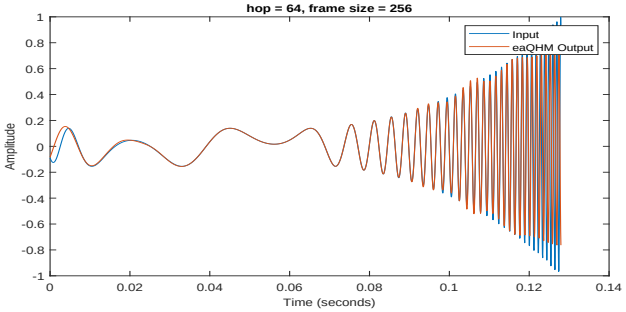


(C) Cubic Amplitude Cubic Phase EDS

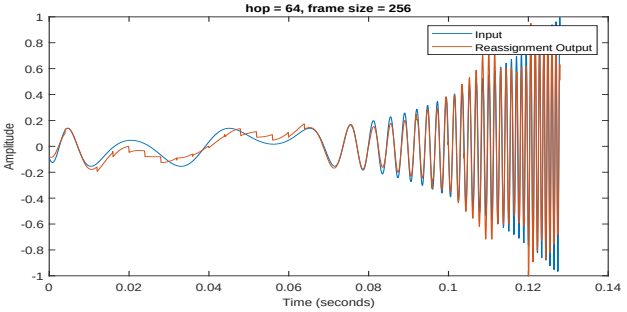


(D) Cubic Amplitude Cubic Phase MoP

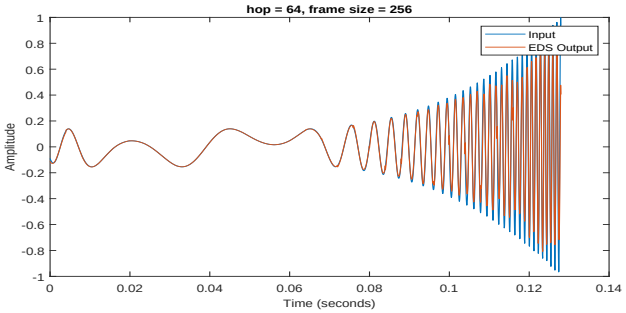
FIGURE A.36: Cubic Amplitude Cubic Phase (C3A-C3P) hop=64 frame=128



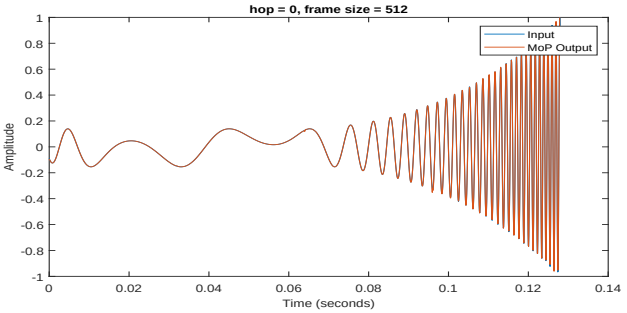
(A) Cubic Amplitude Cubic Phase eaQHM



(B) Cubic Amplitude Cubic Phase Reassignment

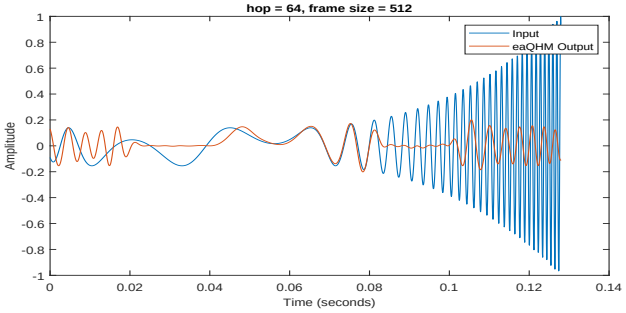


(C) Cubic Amplitude Cubic Phase EDS

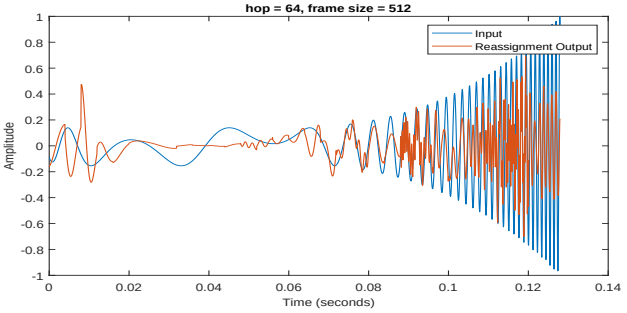


(D) Cubic Amplitude Cubic Phase MoP

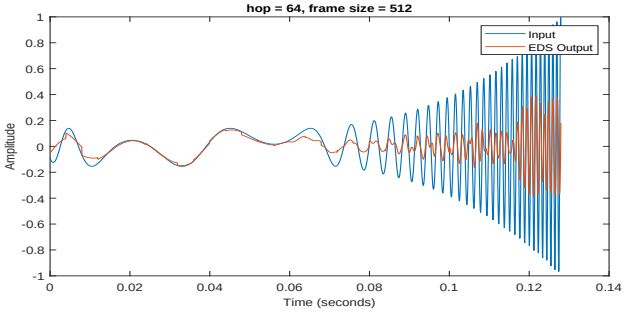
FIGURE A.37: Cubic Amplitude Cubic Phase (C3A-C3P) hop=64 frame=256



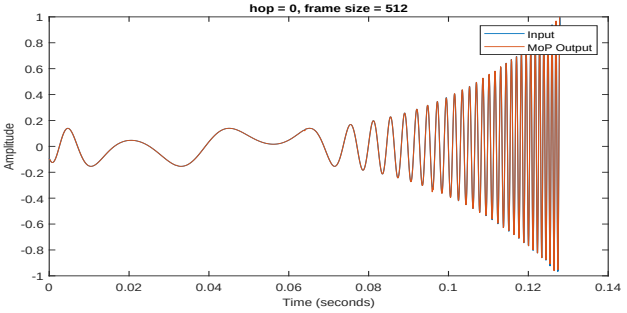
(A) Cubic Amplitude Cubic Phase eaQHM



(B) Cubic Amplitude Cubic Phase Reassignment



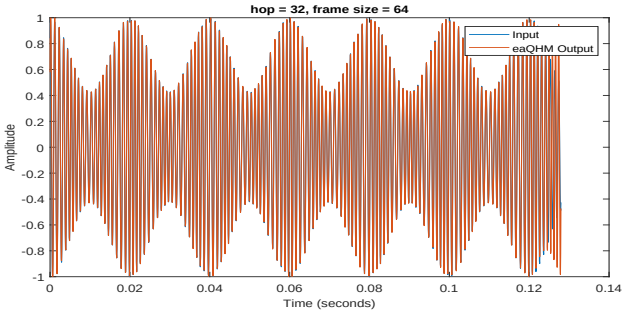
(C) Cubic Amplitude Cubic Phase EDS



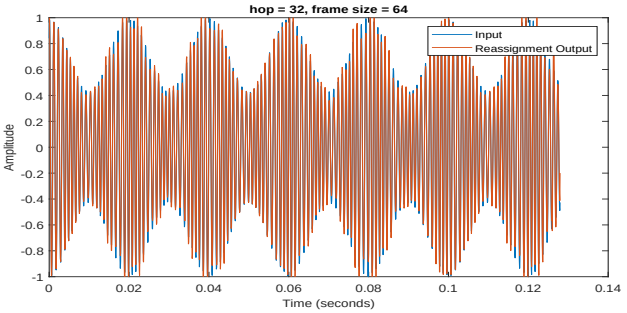
(D) Cubic Amplitude Cubic Phase MoP

FIGURE A.38: Cubic Amplitude Cubic Phase (C3A-C3P) hop=64 frame=512

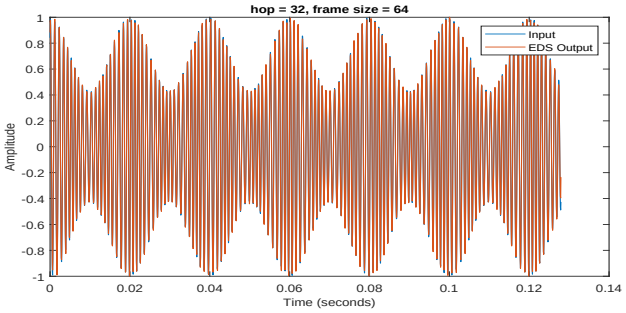
A.2.9 Sinusoidal Amplitude Sinusoidal Phase (SA-SP)



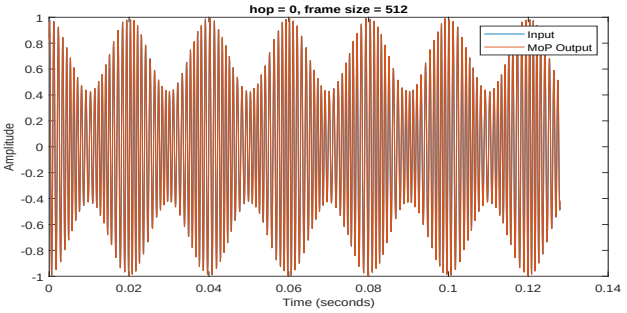
(A) Sinusoidal Amplitude Sinusoidal Phase eaQHM



(B) Sinusoidal Amplitude Sinusoidal Reassignment

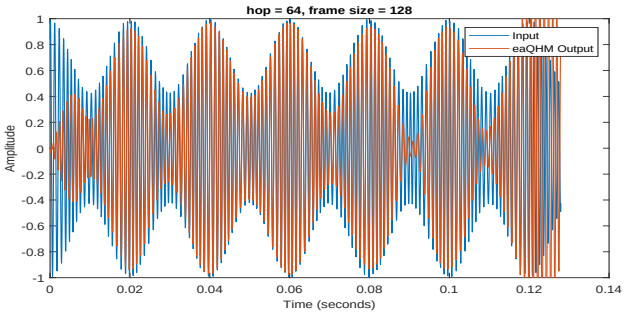


(C) Sinusoidal Amplitude Sinusoidal Phase EDS

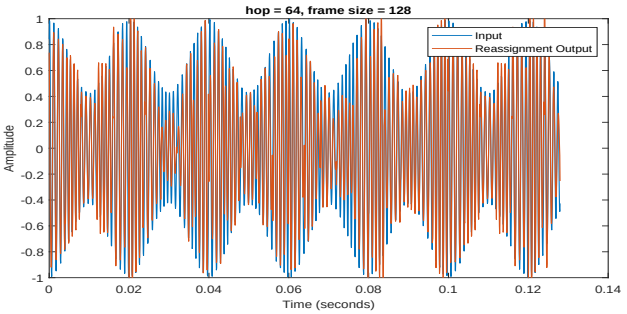


(D) Sinusoidal Amplitude Sinusoidal Phase MoP

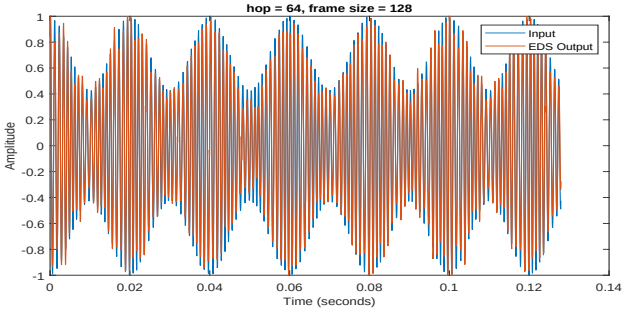
FIGURE A.39: Sinusoidal Amplitude Sinusoidal Phase (SA-SP) hop=32 frame=64



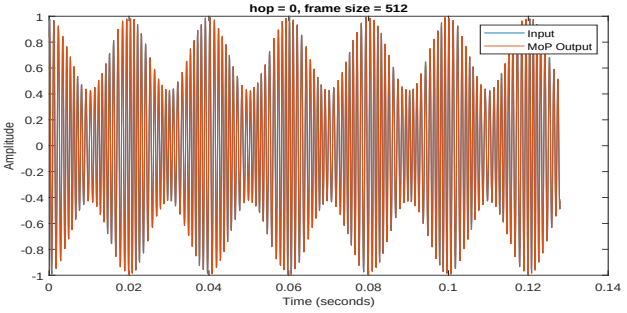
(A) Sinusoidal Amplitude Sinusoidal Phase eaQHM



(B) Sinusoidal Amplitude Sinusoidal Reassignment

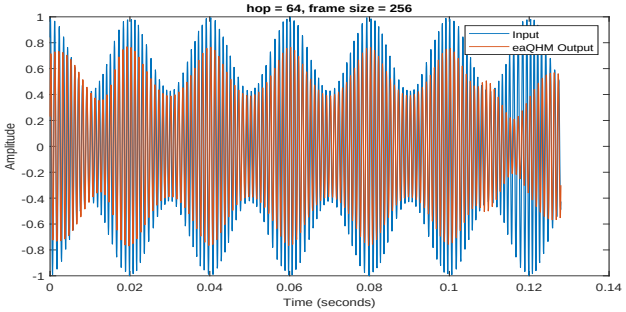


(C) Sinusoidal Amplitude Sinusoidal Phase EDS

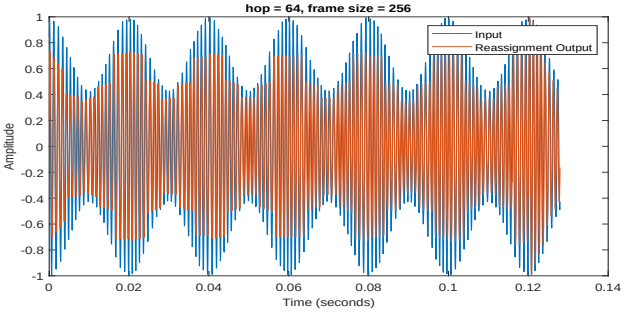


(D) Sinusoidal Amplitude Sinusoidal Phase MoP

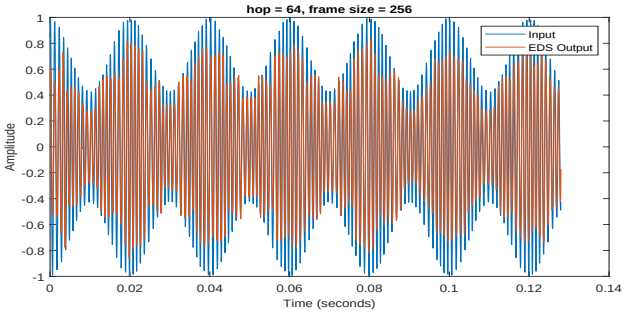
FIGURE A.40: Sinusoidal Amplitude Sinusoidal Phase (SA-SP) hop=64 frame=128



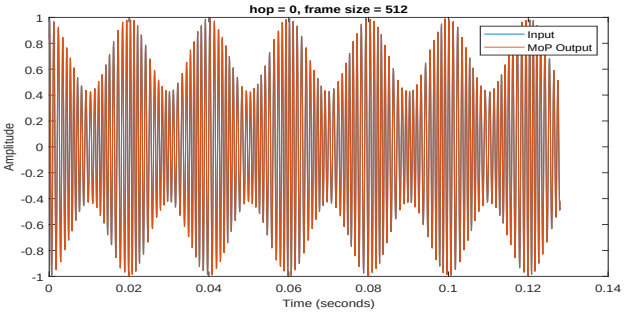
(A) Sinusoidal Amplitude Sinusoidal Phase eaQHM



(B) Sinusoidal Amplitude Sinusoidal Reassignment

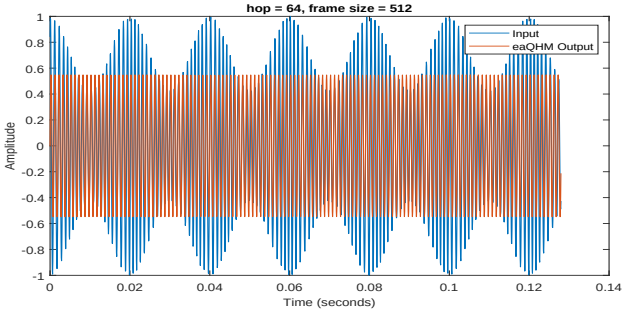


(C) Sinusoidal Amplitude Sinusoidal Phase EDS

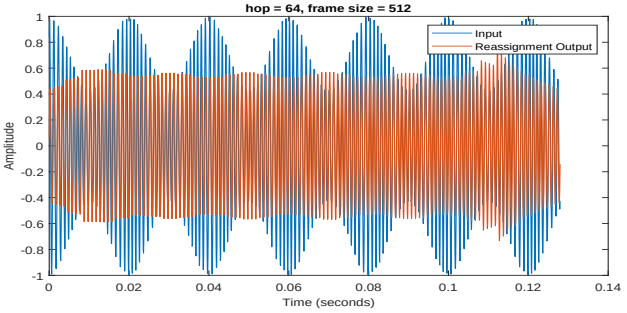


(D) Sinusoidal Amplitude Sinusoidal Phase MoP

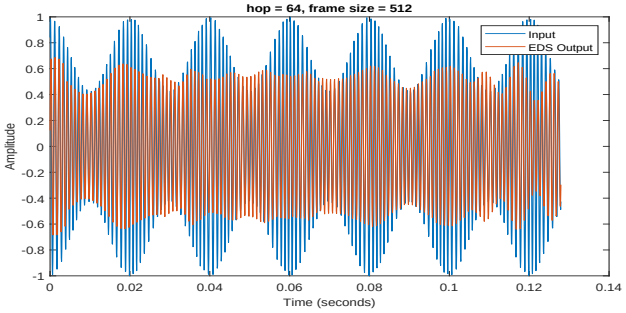
FIGURE A.41: Sinusoidal Amplitude Sinusoidal Phase (SA-SP) hop=64 frame=256



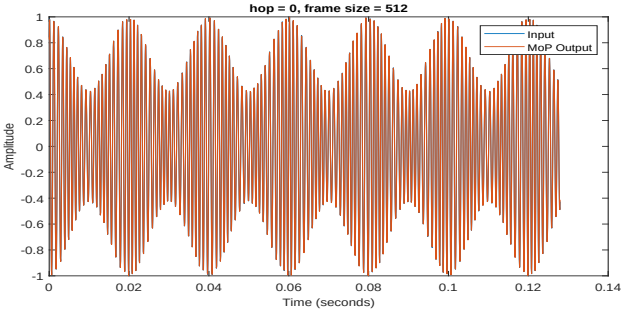
(A) Sinusoidal Amplitude Sinusoidal Phase eaQHM



(B) Sinusoidal Amplitude Sinusoidal Reassignment



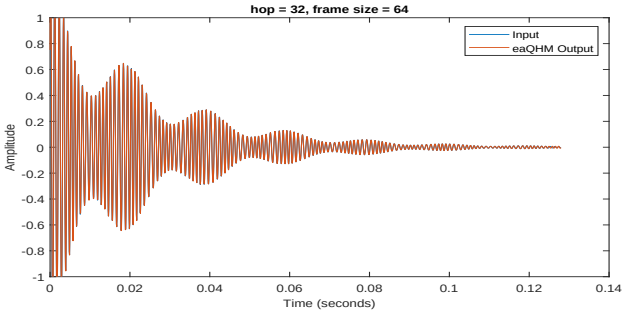
(C) Sinusoidal Amplitude Sinusoidal Phase EDS



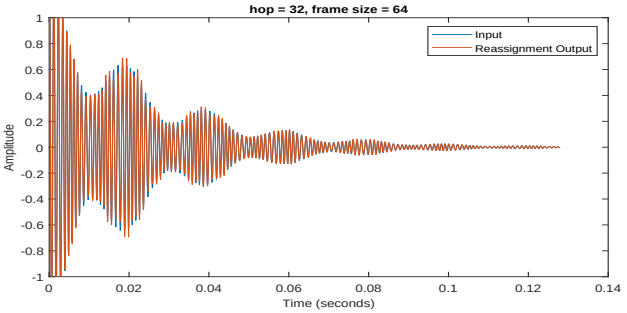
(D) Sinusoidal Amplitude Sinusoidal Phase MoP

FIGURE A.42: Sinusoidal Amplitude Sinusoidal Phase (SA-SP) hop=64 frame=512

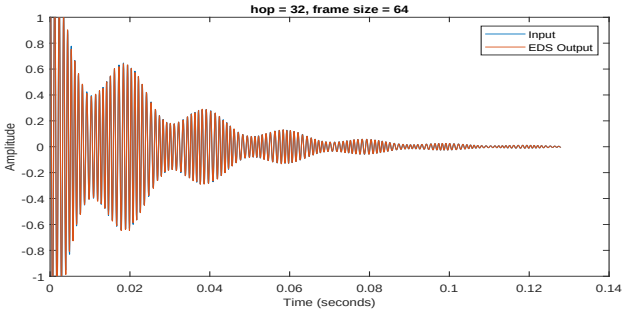
A.2.10 Exponentially Damped Sinusoidal Amplitude Sinusoidal Phase (ESA-SP)



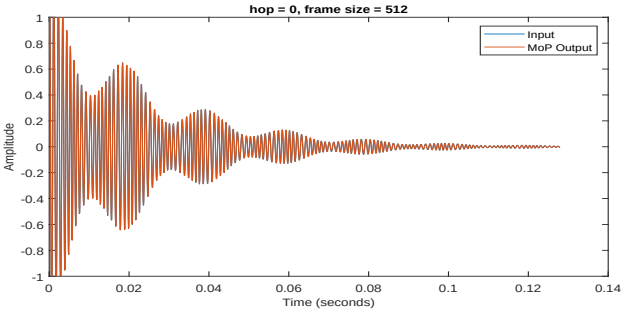
(A) EDS Amplitude Sinusoidal Phase eaQHM



(B) EDS Amplitude Sinusoidal Phase Reassignment

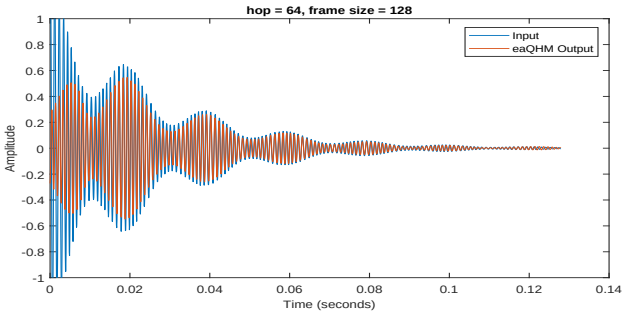


(C) EDS Amplitude Sinusoidal Phase EDS

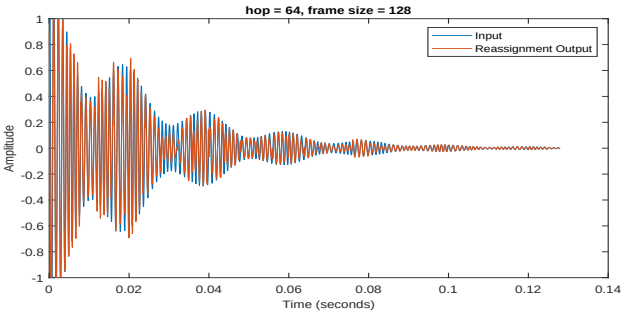


(D) EDS Amplitude Sinusoidal Phase MoP

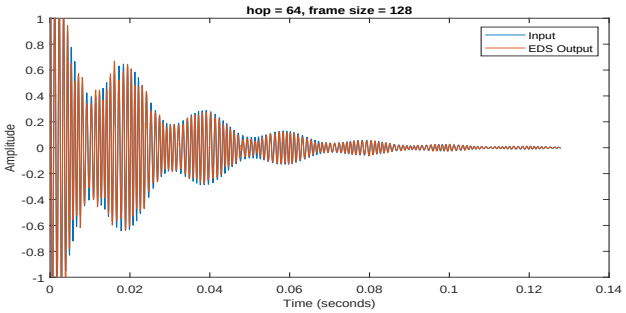
FIGURE A.43: EDS Amplitude Sinusoidal Phase (ESA-SP) hop=32 frame=64



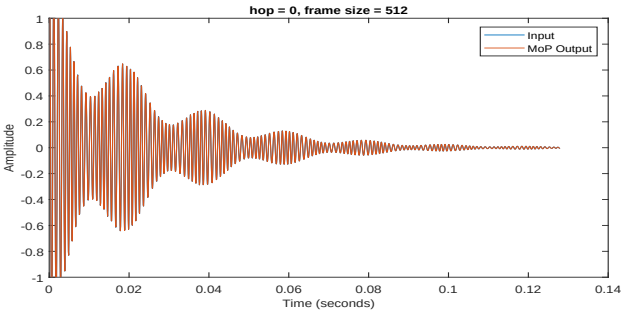
(A) EDS Amplitude Sinusoidal Phase eaQHM



(B) EDS Amplitude Sinusoidal Phase Reassignment

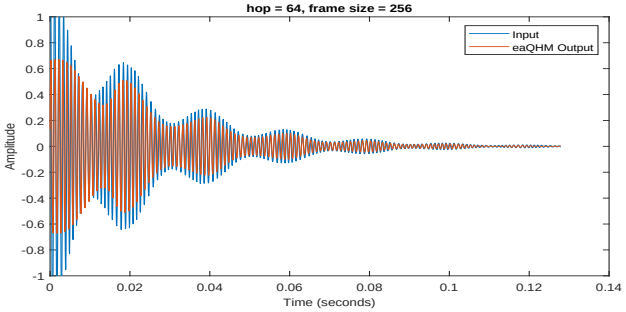


(C) EDS Amplitude Sinusoidal Phase EDS

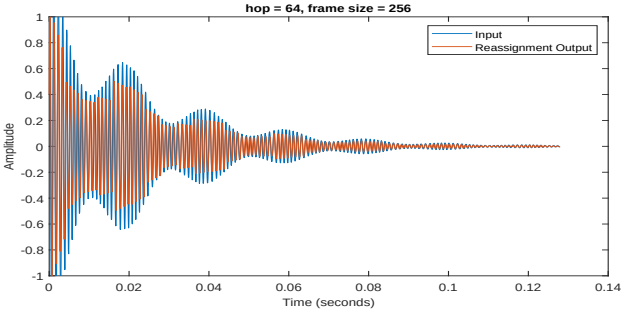


(D) EDS Amplitude Sinusoidal Phase MoP

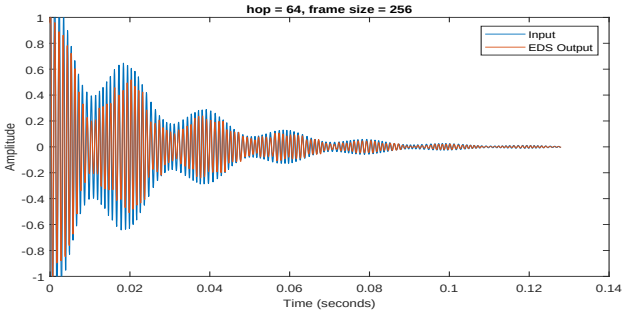
FIGURE A.44: EDS Amplitude Sinusoidal Phase (ESA-SP) hop=64 frame=128



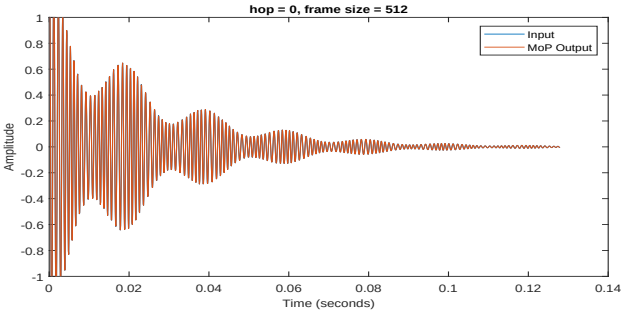
(A) EDS Amplitude Sinusoidal Phase eaQHM



(B) EDS Amplitude Sinusoidal Phase Reassignment

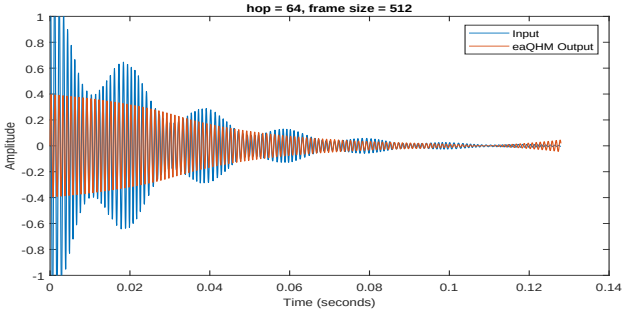


(C) EDS Amplitude Sinusoidal Phase EDS

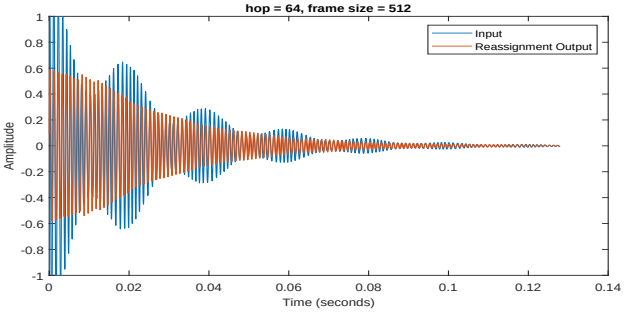


(D) EDS Amplitude Sinusoidal Phase MoP

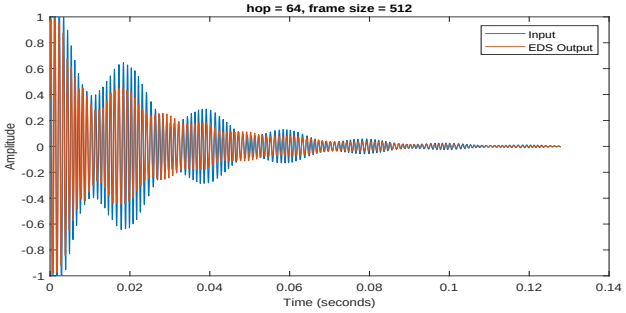
FIGURE A.45: EDS Amplitude Sinusoidal Phase (ESA-SP) hop=64 frame=256



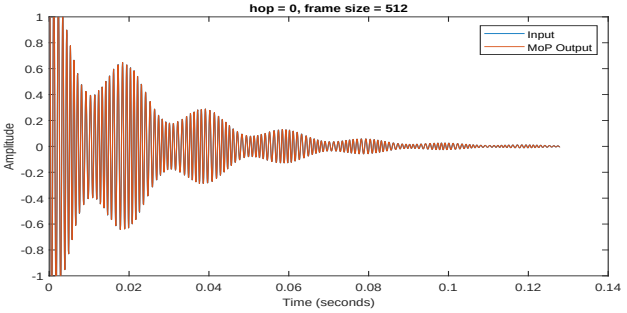
(A) EDS Amplitude Sinusoidal Phase eaQHM



(B) EDS Amplitude Sinusoidal Phase Reassignment



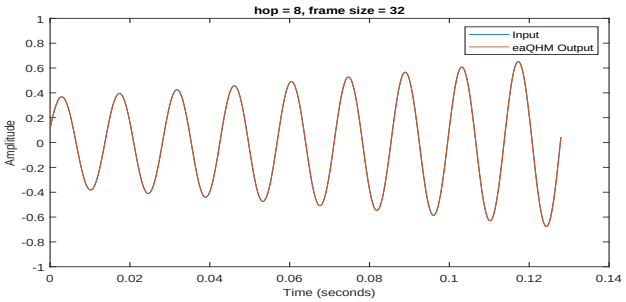
(C) EDS Amplitude Sinusoidal Phase EDS



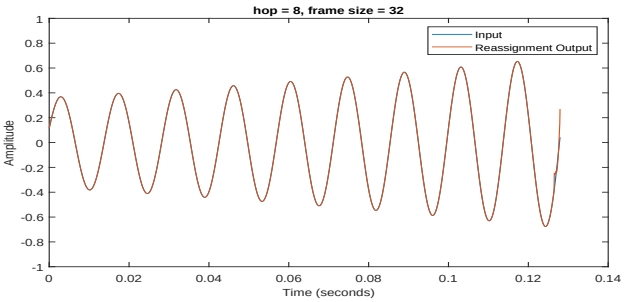
(D) EDS Amplitude Sinusoidal Phase MoP

FIGURE A.46: EDS Amplitude Sinusoidal Phase (ESA-SP) hop=64 frame=512

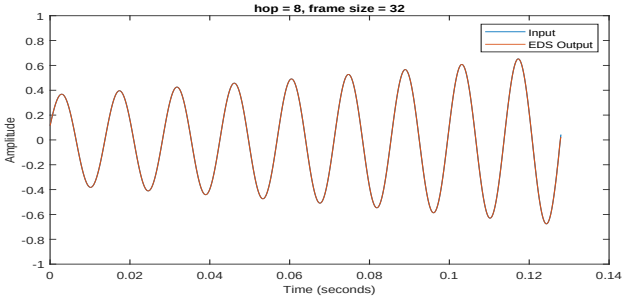
A.2.11 Exponential Amplitude Quadratic Phase (EA-QP)



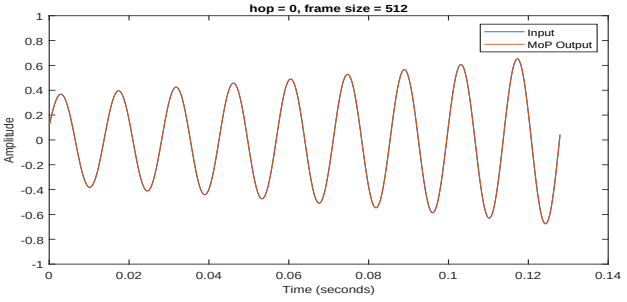
(A) Exponential Amplitude Quadratic Phase eaQHM



(B) Exponential Amplitude Quadratic Phase Reassignment

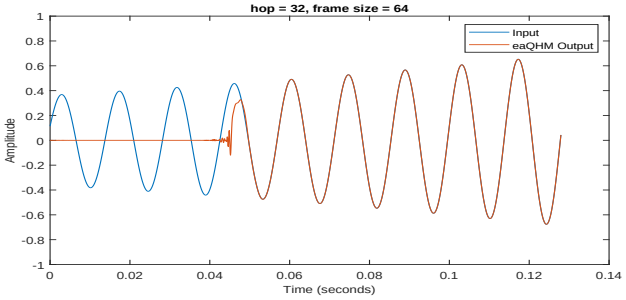


(C) Exponential Amplitude Quadratic Phase EDS

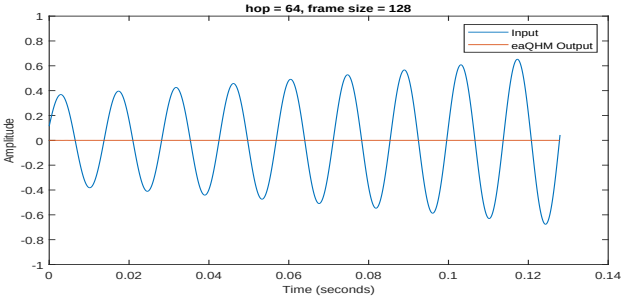


(D) Exponential Amplitude Quadratic Phase MoP

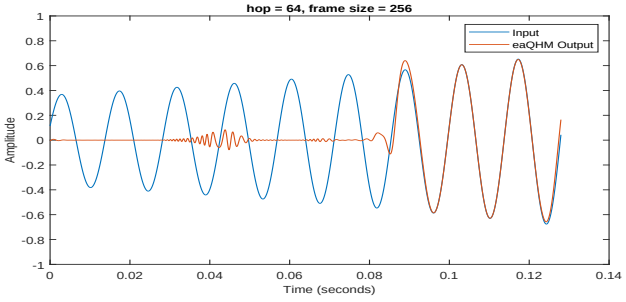
FIGURE A.47: Exponential Amplitude Quadratic Phase (EA-QP) hop=8 frame=32



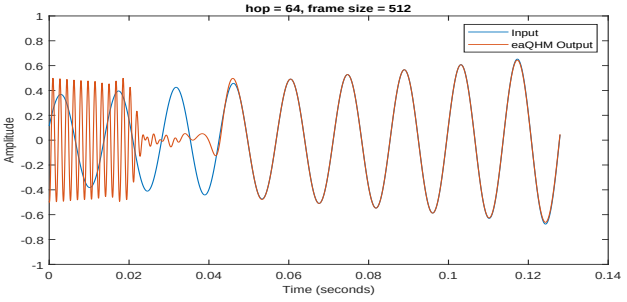
(A) Exponential Amplitude Quadratic Phase eaQHM



(B) Exponential Amplitude Quadratic Phase eaQHM

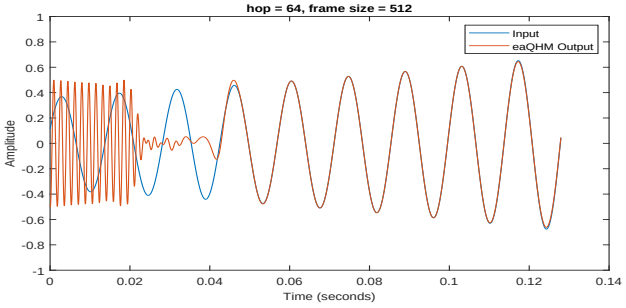


(C) Exponential Amplitude Quadratic Phase eaQHM

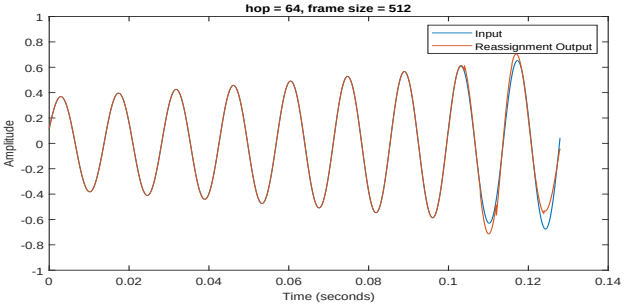


(D) Exponential Amplitude Quadratic Phase eaQHM

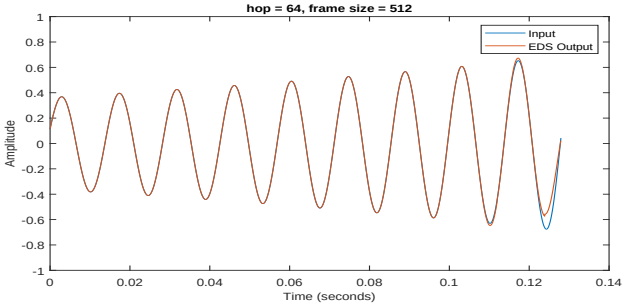
FIGURE A.48: eaQHM struggles with Exponential Amplitude Quadratic Phase, possibly due to the hop size not being small enough in conjunction with the window length being too large



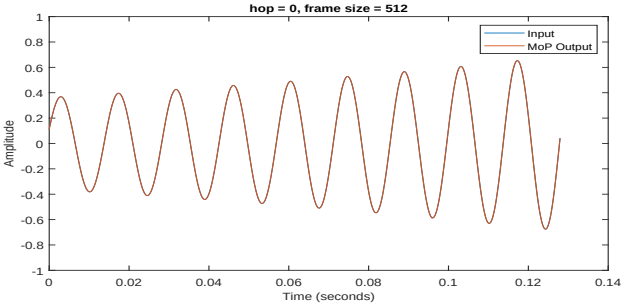
(A) Exponential Amplitude Quadratic Phase eaQHM



(B) Exponential Amplitude Quadratic Phase Reassignment



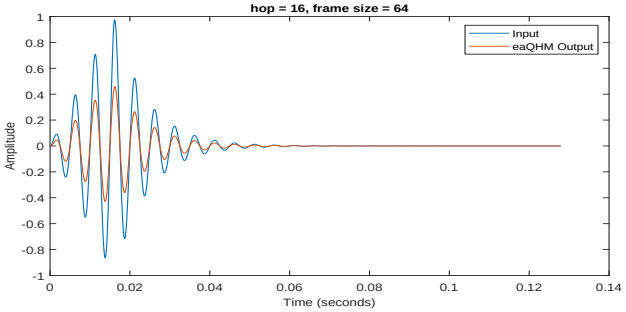
(C) Exponential Amplitude Quadratic Phase EDS



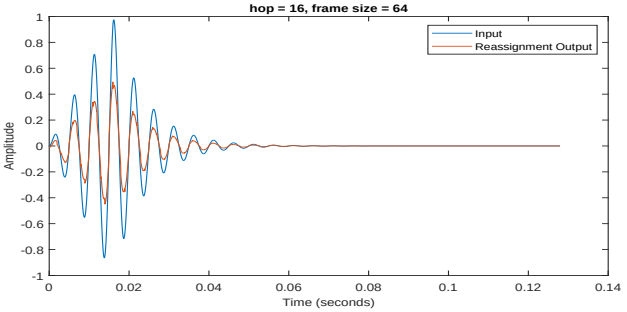
(D) Exponential Amplitude Quadratic Phase MoP

FIGURE A.49: Exponential Amplitude Quadratic Phase (EA-QP) hop=64 frame=512

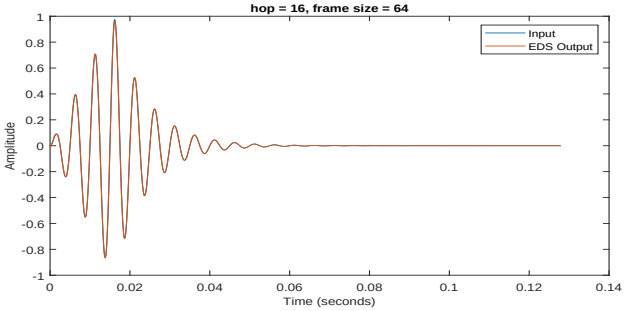
A.2.12 Exponential Second Order Amplitude Constant Phase (EA-NM)



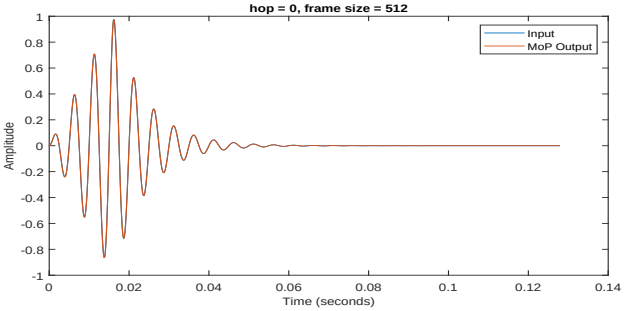
(A) Exponential 2nd Order Amplitude Constant Phase eaQHM



(B) Exponential 2nd Order Amplitude Constant Phase Reassignment

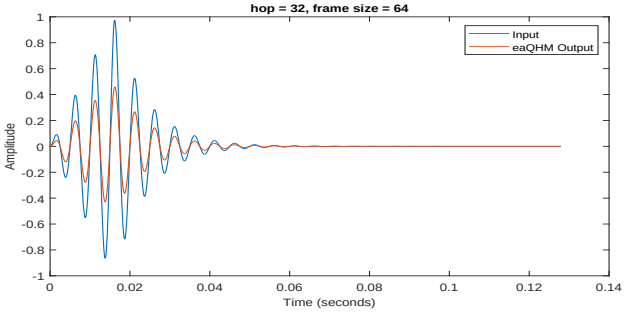


(C) Exponential 2nd Order Amplitude Constant Phase EDS

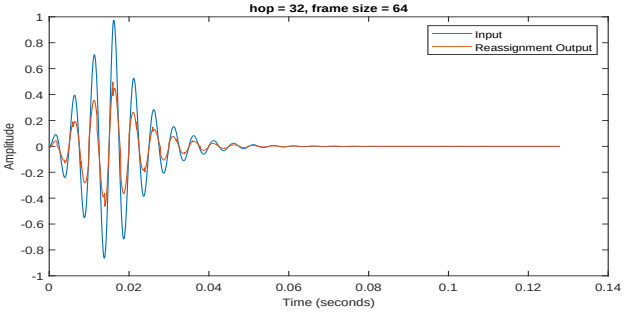


(D) Exponential 2nd Order Amplitude Constant Phase MoP

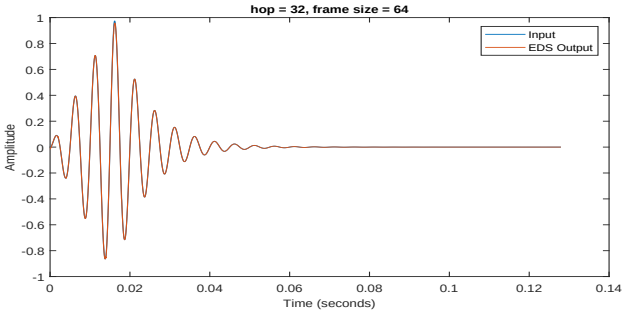
FIGURE A.50: Exponential Second Order Amplitude Constant Phase (EA-NM) hop=16 frame=64



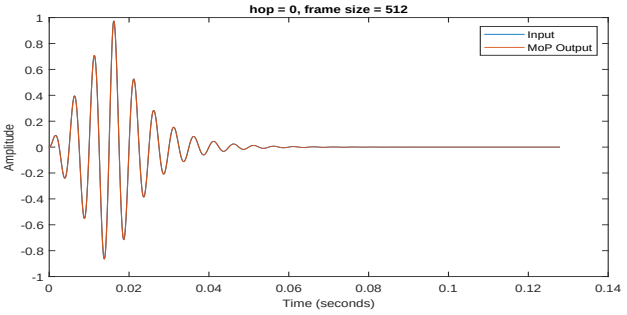
(A) Exponential 2nd Order Amplitude Constant Phase eaQHM



(B) Exponential 2nd Order Amplitude Constant Phase Reassignment

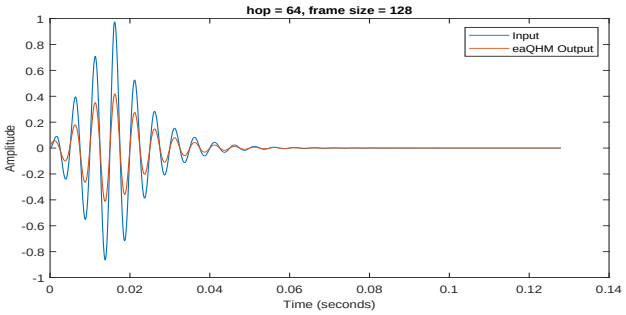


(C) Exponential 2nd Order Amplitude Constant Phase EDS

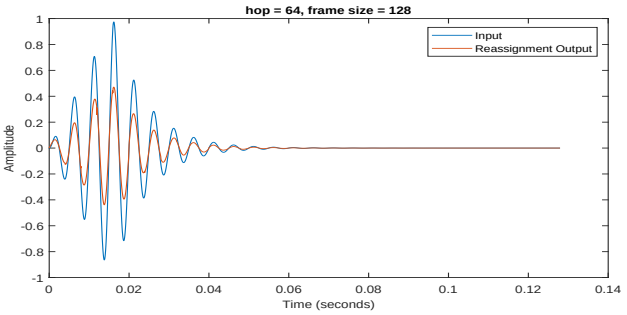


(D) Exponential 2nd Order Amplitude Constant Phase MoP

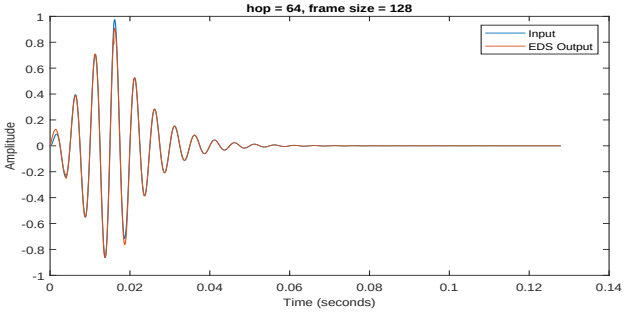
FIGURE A.51: Exponential Second Order Amplitude Constant Phase (EA-NM) hop=32 frame=64



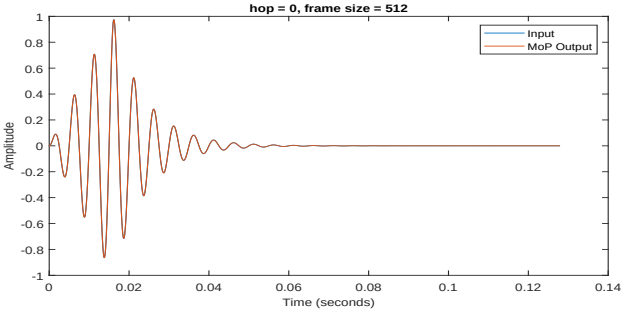
(A) Exponential 2nd Order Amplitude Constant Phase eaQHM



(B) Exponential 2nd Order Amplitude Constant Phase Reassignment

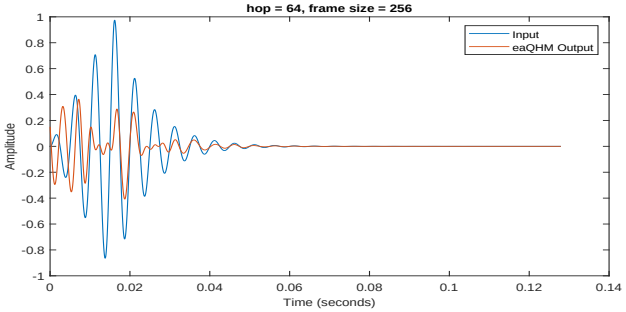


(C) Exponential 2nd Order Amplitude Constant Phase EDS

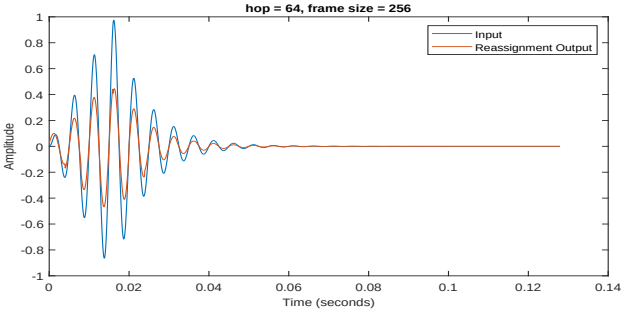


(D) Exponential 2nd Order Amplitude Constant Phase MoP

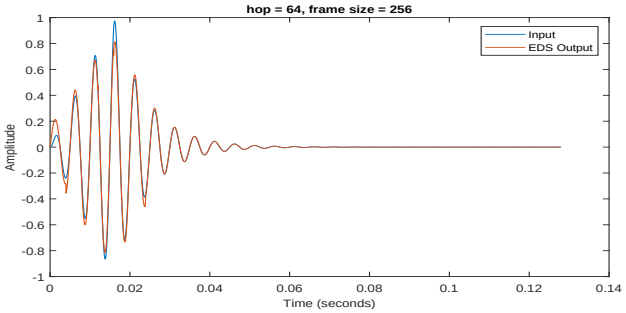
FIGURE A.52: Exponential Second Order Amplitude Constant Phase (EA-NM) hop=64 frame=128



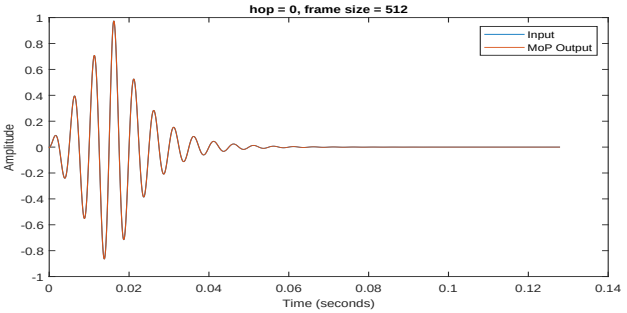
(A) Exponential 2nd Order Amplitude Constant Phase eaQHM



(B) Exponential 2nd Order Amplitude Constant Phase Reassignment

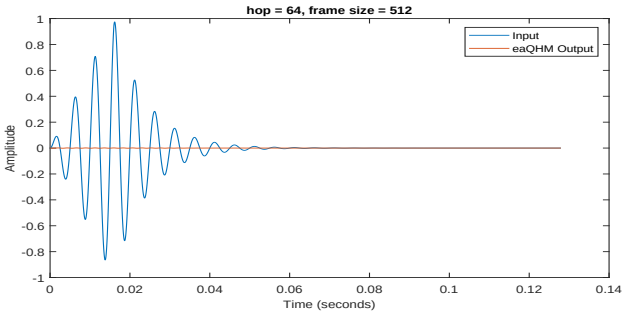


(C) Exponential 2nd Order Amplitude Constant Phase EDS

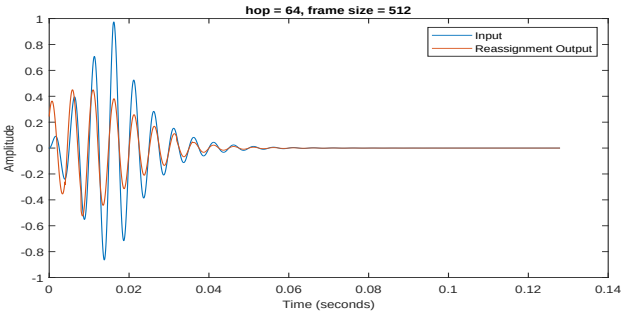


(D) Exponential 2nd Order Amplitude Constant Phase MoP

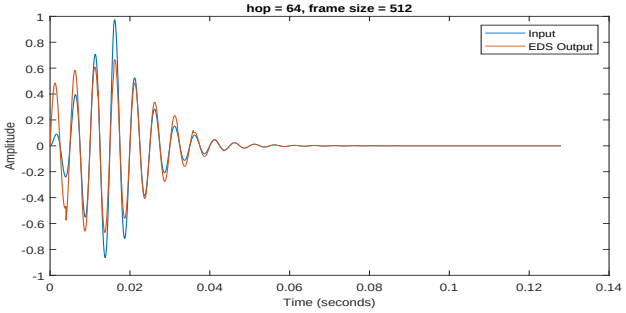
FIGURE A.53: Exponential Second Order Amplitude Constant Phase (EA-NM) hop=64 frame=256



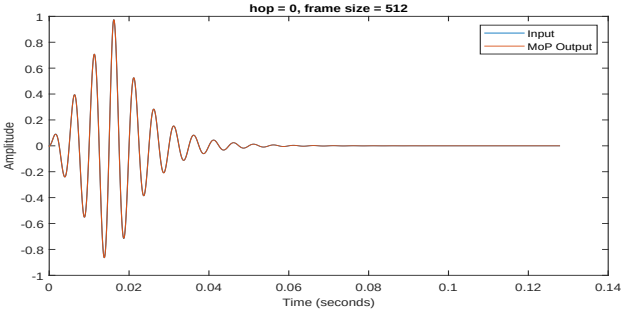
(A) Exponential 2nd Order Amplitude Constant Phase eaQHM



(B) Exponential 2nd Order Amplitude Constant Phase Reassignment



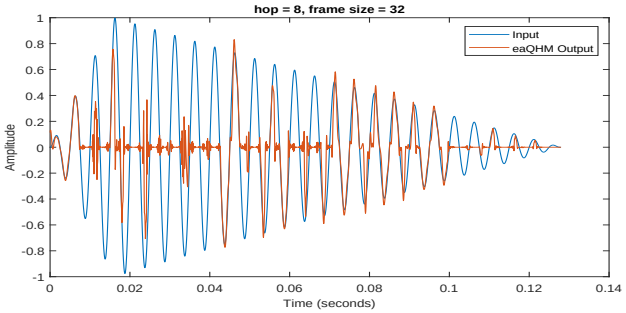
(C) Exponential 2nd Order Amplitude Constant Phase EDS



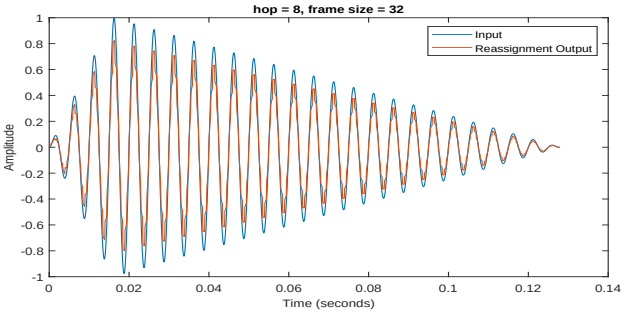
(D) Exponential 2nd Order Amplitude Constant Phase MoP

FIGURE A.54: Exponential Second Order Amplitude Constant Phase (EA-NM) hop=64 frame=512

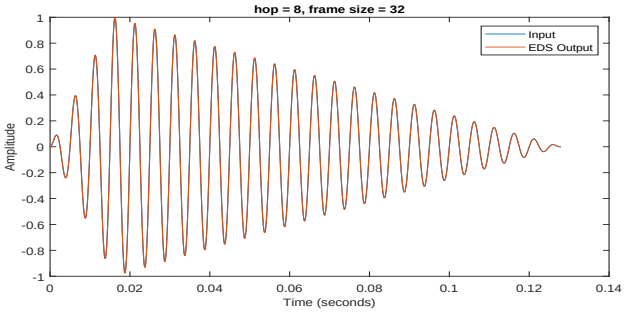
A.2.13 Linear Second Order Amplitude Constant Phase (LA-NM)



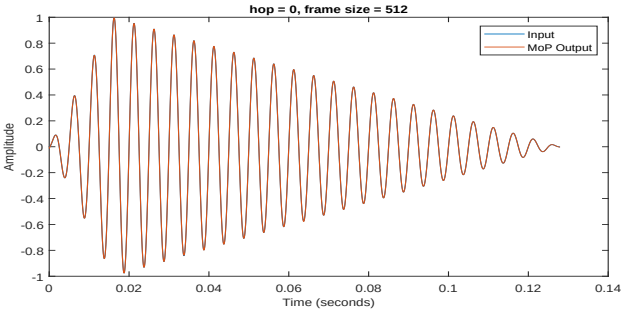
(A) Linear 2nd Order Amplitude Constant Phase eaQHM



(B) Linear 2nd Order Amplitude Reassignment

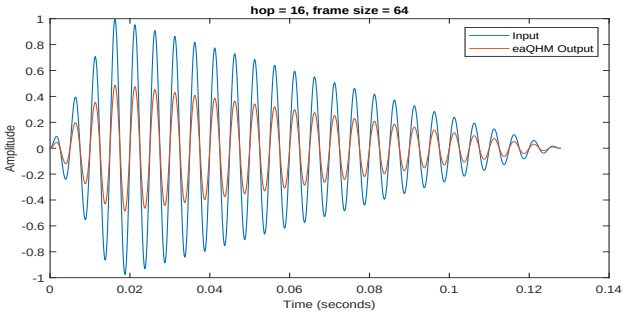


(C) Linear 2nd Order Amplitude EDS

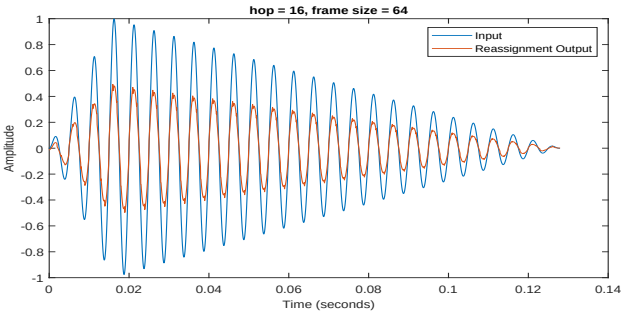


(D) Linear 2nd Order Amplitude MoP

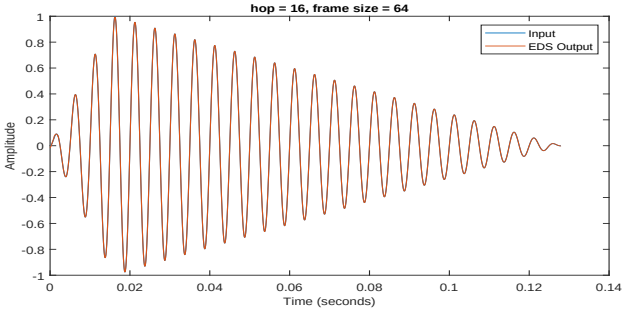
FIGURE A.55: Linear Second Order Amplitude Constant Phase (LA-NM) hop=8 frame=32



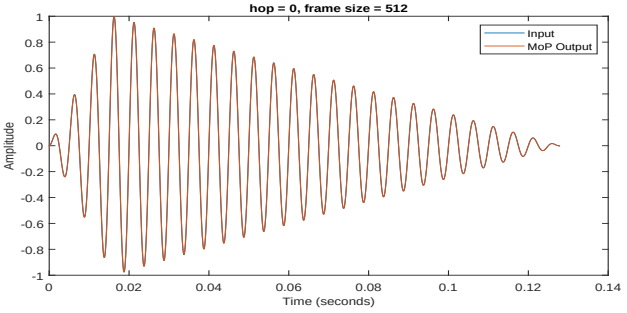
(A) Linear Second Order Amplitude eaQHM



(B) Linear Second Order Amplitude Reassignment

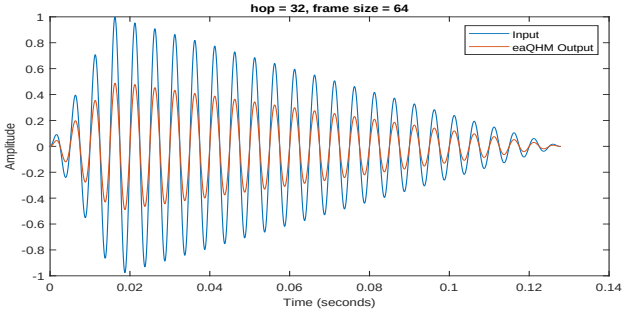


(C) Linear Second Order Amplitude Constant Phase EDS

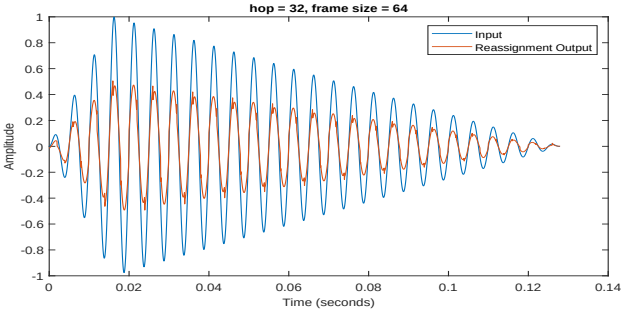


(D) Linear Second Order Amplitude MoP

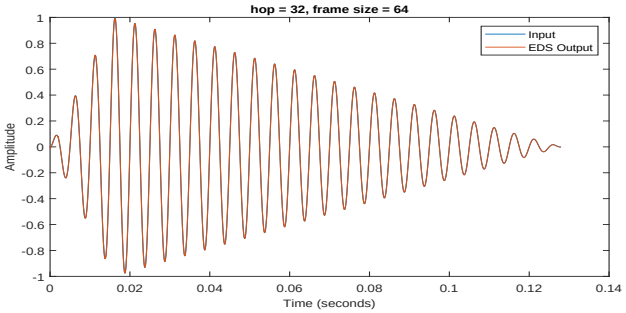
FIGURE A.56: Linear Second Order Amplitude Constant Phase (LA-NM) hop=16 frame=64



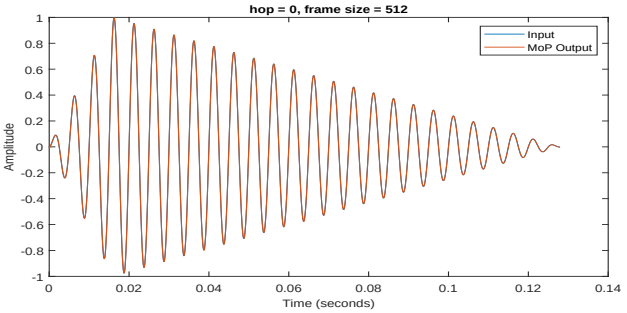
(A) Linear Second Order Amplitude eaQHM



(B) Linear Second Order Amplitude Reassignment

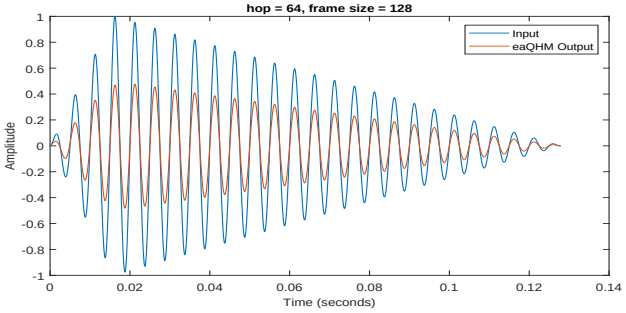


(C) Linear Second Order Amplitude EDS

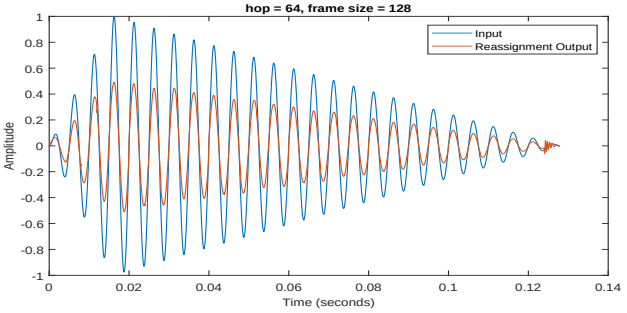


(D) Linear Second Order Amplitude MoP

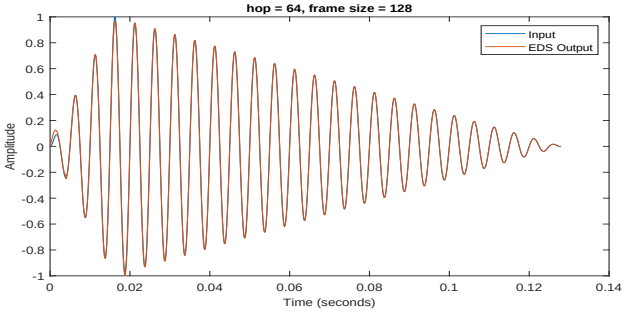
FIGURE A.57: Linear Second Order Amplitude Constant Phase (LA-NM) hop=32 frame=64



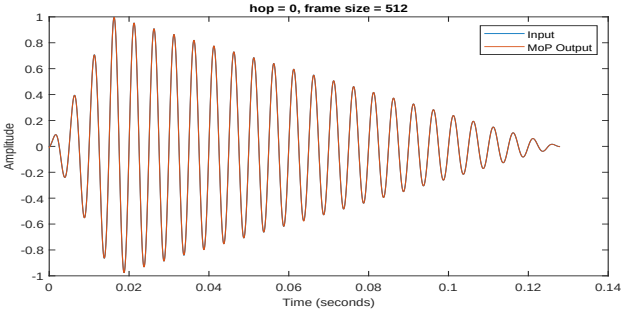
(A) Linear Second Order Amplitude eaQHM



(B) Linear Second Order Amplitude Reassignment

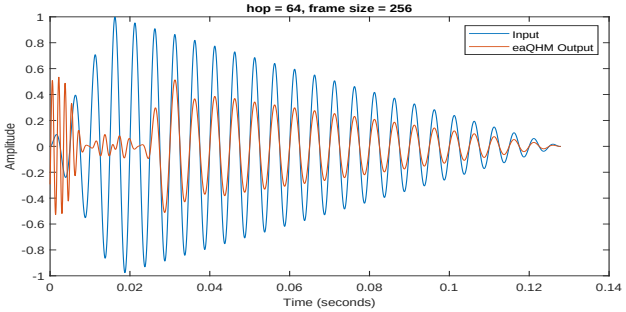


(C) Linear Second Order Amplitude EDS

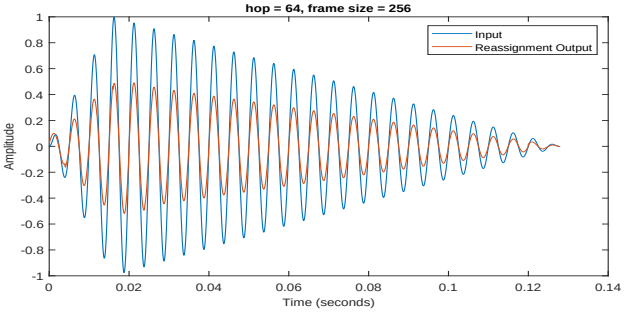


(D) Linear Second Order Amplitude MoP

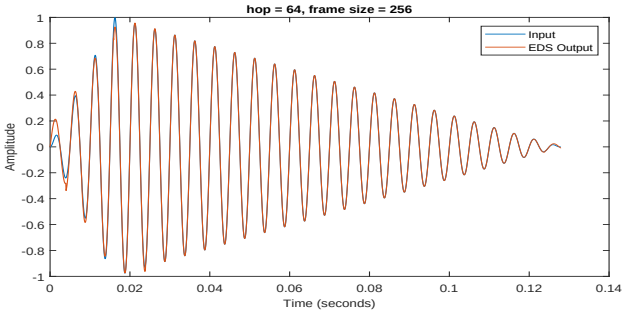
FIGURE A.58: Linear Second Order Amplitude Constant Phase (LA-NM) hop=64 frame=128



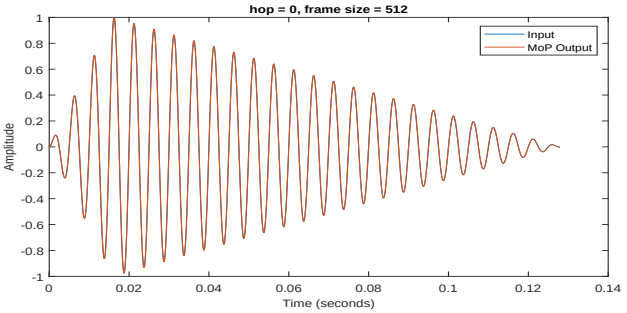
(A) Linear Second Order Amplitude eaQHM



(B) Linear Second Order Amplitude Reassignment

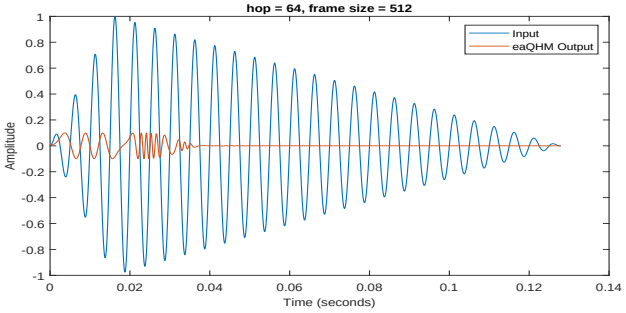


(C) Linear Second Order Amplitudee EDS

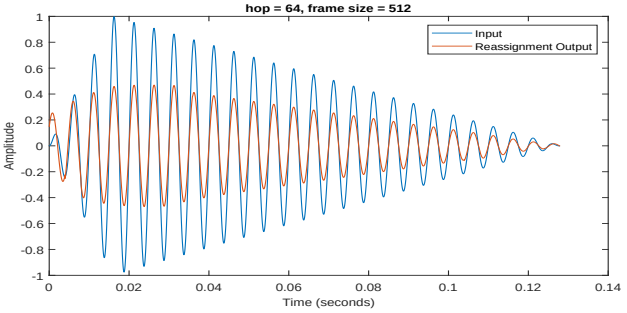


(D) Linear Second Order Amplitude MoP

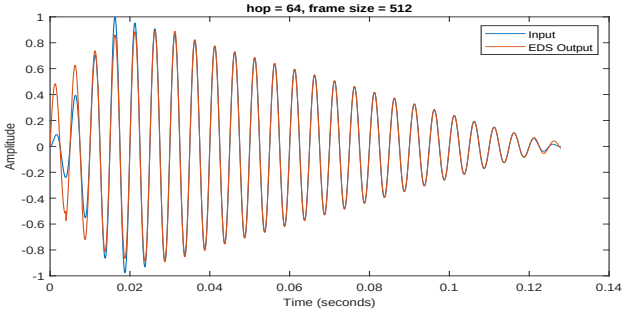
FIGURE A.59: Linear Second Order Amplitude Constant Phase (LA-NM) hop=64 frame=256



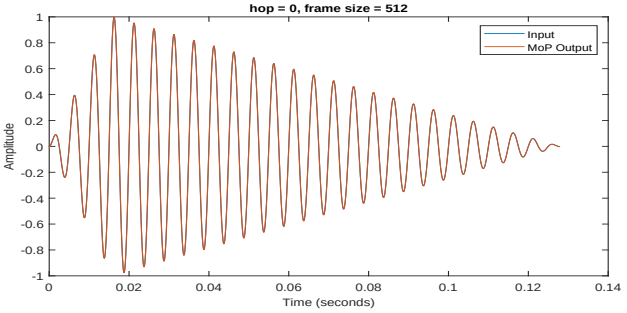
(A) Linear Second Order Amplitude eaQHM



(B) Linear Second Order Amplitude Reassignment



(C) Linear Second Order Amplitude EDS



(D) Linear Second Order Amplitude MoP

FIGURE A.60: Linear Second Order Amplitude Constant Phase (LA-NM) hop=64 frame=512

A.2.14 Tables

Parameter	Atom1	Atom2	Atom3	Atom4	Atom5	Atom6	Atom7	Atom8
Fest Original	1000	1055	948	1096	897	835	1228	1036
Fest Calculated	500	527	474	548	448	417	614	518
Fest Measured	500	555	448	598	395	330	730	537
Difference	0	-28	26	-50	53	87	-116	-19

TABLE A.1: Estimated Frequency values compared between reference and the pitch shifted signals decomposition's

Parameter	Atom1	Atom2	Atom3	Atom4	Atom5	Atom6	Atom7	Atom8
Aest Original	0.7679	0.3338	0.3736	0.1210	0.0648	0.0290	0.0451	0.0083
Aest Measured	0.7722	0.3616	0.3978	0.1063	0.0603	0.0367	0.0437	0.0099
Difference	-0.0043	-0.0278	-0.0242	0.0147	0.0045	-0.0077	0.0014	-0.0016

TABLE A.2: Estimated Amplitude values compared between reference and the pitch shifted signals decomposition's

Parameter	Atom1	Atom2	Atom3	Atom4	Atom5	Atom6	Atom7	Atom8
daEst Original	-13.66	-21.88	-22.95	-17.08	-11.04	-10.75	-27.62	4.00
daEst Measured	-13.81	-24.07	-24.82	-16.43	-11.24	-16.83	-25.85	3.80
Difference	0.1500	2.1900	1.8700	-0.6500	0.2000	6.0800	-1.7700	0.2000

TABLE A.3: Estimated Amplitude Change values compared between reference and the pitch shifted signals decomposition's

Parameter	Atom1	Atom2	Atom3	Atom4	Atom5	Atom6	Atom7	Atom8
Phase Original	-1.57	1.88	0.97	1.70	1.92	-2.11	-3.13	-2.63
Phase Measured	-1.57	1.91	0.99	1.68	2.01	-1.79	-3.11	-2.62
Difference	0	-0.0300	-0.0200	0.0200	-0.0900	-0.3200	-0.0200	-0.0100

TABLE A.4: Estimated Phase values compared between reference and the pitch shifted signals decomposition's

A.2.15 Timing Measurements (MIPS)

TABLE A.5: SRER (dB) for each Hop and Frame Size (samples)

Hop	Size	def	8	15	16	16	32	64	64	64	def hop
Frame	Size	def	32	32	64	380	64	128	256	512	def frame
CA-LP	eaQHM	290.12	290.58	290.58	290.50	290.12	290.43	289.74	289.30	289.34	H=16,F=190
	RSM	24.47	31.83	31.09	29.92	24.47	27.74	25.86	25.87	22.38	H=16,F=380
	EDSM	26.56	37.21	34.95	34.89	26.56	36.17	34.09	29.62	25.67	H=16,F=380
	MoP	53.86	53.86	53.86	53.86	53.86	53.86	53.86	53.86	53.86	H=0,F=512
EA-LP	eaQHM	84.70	97.14	73.13	82.87	49.51	80.29	65.72	54.71	45.02	H=16,F=24
	RSM	69.43	61.06	59.62	73.04	69.90	70.65	68.92	69.42	67.63	H=16,F=40
	EDSM	81.36	81.76	79.46	79.45	71.33	80.77	78.74	74.28	70.60	H=16,F=48
	MoP	52.25	52.25	52.25	52.25	52.25	52.25	52.25	52.25	52.25	H=0,F=512
CA-C3P	eaQHM	60.91	118.01	101.50	97.47	50.57	76.51	55.18	55.15	12.60	H=16,F=154
	RSM	4.84	32.17	33.46	27.85	4.89	25.16	18.57	8.46	0.15	H=16,F=308
	EDSM	11.63	37.10	34.84	34.99	6.64	36.06	33.16	18.08	3.96	H=16,F=308
	MoP	41.08	41.08	41.08	41.08	41.08	41.08	41.08	41.08	41.08	H=0,F=512
EA-C3P	eaQHM	48.94	96.88	72.92	82.61	38.30	80.11	65.37	53.46	22.31	H=16,F=154
	RSM	6.70	41.42	41.16	31.49	4.98	31.19	23.09	9.86	-2.43	H=16,F=308
	EDSM	25.54	81.62	79.31	75.22	19.15	71.51	47.74	28.11	10.89	H=16,F=308
	MoP	6.80	6.80	6.80	6.80	6.80	6.80	6.80	6.80	6.80	H=0,F=512
LA-C3P	eaQHM	43.68	43.72	67.97	28.68	37.87	17.15	50.28	46.00	18.20	H=16,F=154
	RSM	2.82	36.07	36.88	29.45	3.20	27.88	19.20	7.15	-1.43	H=16,F=308
	EDSM	22.01	42.32	40.08	40.18	16.96	41.24	38.69	25.97	9.27	H=16,F=308
	MoP	46.06	46.06	46.06	46.06	46.06	46.06	46.06	46.06	46.06	H=0,F=512
C3A-C3P	eaQHM	52.49	101.97	92.62	84.73	46.72	65.12	46.01	46.39	23.94	H=16,F=154
	RSM	7.75	23.91	25.33	20.86	6.22	17.62	14.29	10.85	1.33	H=16,F=308
	EDSM	7.07	28.19	25.95	26.05	4.08	27.12	24.75	12.47	2.73	H=16,F=308
	MoP	37.03	37.03	37.03	37.03	37.03	37.03	37.03	37.03	37.03	H=0,F=512
SA-SP	eaQHM	56.29	77.06	49.77	50.37	4.80	29.43	11.48	7.68	3.95	H=16,F=24
	RSM	28.33	34.81	34.76	24.26	3.26	20.21	8.65	4.21	3.33	H=16,F=48
	EDSM	39.60	41.21	39.13	33.92	3.22	31.87	14.30	6.40	4.07	H=16,F=48
	MoP	50.18	50.18	50.18	50.18	50.18	50.18	50.18	50.18	50.18	H=0,F=512
ESA-SP	eaQHM	46.30	73.32	40.23	48.93	4.83	42.19	11.63	5.40	4.21	H=16,F=24
	RSM	27.12	35.81	35.53	22.15	3.18	20.47	8.30	4.56	3.29	H=16,F=48
	EDSM	42.21	56.26	54.18	32.71	4.59	30.03	14.97	6.88	5.06	H=16,F=48
	MoP	50.16	50.16	50.16	50.16	50.16	50.16	50.16	50.16	50.16	H=0,F=512
EA-QP	eaQHM	70.28	138.36	126.57	122.56	8.20	104.28	4.18	0.00	2.04	H=16,F=55
	RSM	23.31	27.78	27.88	27.63	16.65	27.01	21.99	18.02	11.84	H=16,F=440
	EDSM	29.44	34.72	32.46	32.39	24.10	33.68	31.59	27.15	23.20	H=16,F=110
	MoP	51.85	51.85	51.85	51.85	51.85	51.85	51.85	51.85	51.85	H=0,F=512
EA-NM	eaQHM	7.13	5.62	7.14	5.21	2.83	4.09	2.94	3.03	2.04	H=16,F=55
	RSM	6.23	6.44	6.41	5.62	2.00	5.59	3.96	3.04	-1.61	H=16,F=440
	EDSM	31.99	39.09	38.83	30.62	15.10	28.24	20.99	17.87	8.85	H=16,F=110
	MoP	48.90	50.11	50.11	50.56	48.90	50.56	50.23	50.07	50.07	H=0,F=512
LA-NM	eaQHM	6.54	1.95	0.99	1.89	2.93	2.68	3.00	3.02	2.11	H=16,F=55
	RSM	6.18	6.19	6.18	6.05	6.56	6.04	5.97	6.19	6.75	H=16,F=440
	EDSM	42.08	48.83	48.68	40.29	24.08	38.44	30.51	27.23	16.40	H=16,F=110
	MoP	50.75	50.38	50.38	50.48	50.75	50.48	50.78	51.03	51.03	H=0,F=512

TABLE A.6: Speed (MIPS) for each Hop and Frame Size (samples)

Hop	Size	def	8	15	16	16	32	64	64	64	def hop
Frame	Size	def	32	32	64	380	64	128	256	512	def frame
CA-LP	eaQHM	0.199	0.180	0.135	0.119	0.201	0.145	0.154	0.158	0.123	H=16,F=190
	RSM	0.218	0.428	0.222	0.259	0.245	0.211	0.136	0.099	0.115	H=16,F=380
	EDSM	1.697	0.151	0.107	0.219	1.818	0.111	0.287	0.424	0.730	H=16,F=380
	MoP	0.216	0.189	0.188	0.255	0.192	0.130	0.192	0.266	0.149	H=0,F=512
EA-LP	eaQHM	0.068	0.093	0.075	0.105	0.155	0.074	0.068	0.061	0.089	H=16,F=24
	RSM	0.204	0.335	0.122	0.179	0.270	0.068	0.070	0.041	0.089	H=16,F=40
	EDSM	0.087	0.103	0.060	0.096	2.073	0.062	0.257	0.376	0.990	H=16,F=48
	MoP	0.083	0.100	0.097	0.111	0.081	0.088	0.085	0.118	0.104	H=0,F=512
CA-C3P	eaQHM	0.148	0.169	0.067	0.059	0.140	0.036	0.047	0.038	0.087	H=16,F=154
	RSM	0.133	0.313	0.204	0.206	0.306	0.094	0.071	0.078	0.071	H=16,F=308
	EDSM	1.953	0.889	0.642	0.705	2.839	0.473	0.488	0.693	1.423	H=16,F=308
	MoP	0.576	0.602	0.590	0.701	0.613	0.499	0.580	0.543	0.486	H=0,F=512
EA-C3P	eaQHM	0.076	0.074	0.080	0.055	0.086	0.037	0.035	0.058	0.071	H=16,F=154
	RSM	0.221	0.327	0.128	0.194	0.354	0.095	0.058	0.052	0.122	H=16,F=308
	EDSM	1.091	0.098	0.037	0.083	1.824	0.028	0.115	0.494	1.481	H=16,F=308
	MoP	0.413	0.634	0.479	0.541	0.420	0.499	0.644	0.655	0.496	H=0,F=512
LA-C3P	eaQHM	0.125	0.238	0.080	0.089	0.153	0.064	0.054	0.038	0.149	H=16,F=154
	RSM	0.438	0.337	0.199	0.166	0.234	0.116	0.073	0.048	0.058	H=16,F=308
	EDSM	1.186	0.103	0.057	0.080	2.284	0.034	0.116	0.431	1.258	H=16,F=308
	MoP	0.610	0.554	0.492	0.493	0.588	0.488	0.606	0.645	0.510	H=0,F=512
C3A-C3P	eaQHM	0.123	0.071	0.047	0.079	0.089	0.027	0.018	0.040	0.196	H=16,F=154
	RSM	0.171	0.328	0.191	0.161	0.314	0.104	0.052	0.073	0.079	H=16,F=308
	EDSM	1.656	0.061	0.038	0.077	1.849	0.027	0.132	0.620	1.294	H=16,F=308
	MoP	0.417	0.432	0.429	0.538	0.466	0.526	0.527	0.670	0.501	H=0,F=512
SA-SP	eaQHM	0.076	0.076	0.051	0.056	0.148	0.023	0.023	0.078	0.060	H=16,F=24
	RSM	0.110	0.333	0.131	0.170	0.331	0.094	0.050	0.063	0.069	H=16,F=48
	EDSM	0.039	0.078	0.032	0.082	2.150	0.030	0.053	0.176	1.447	H=16,F=48
	MoP	0.481	0.587	0.639	0.633	0.519	0.679	0.406	0.390	0.652	H=0,F=512
ESA-SP	eaQHM	0.089	0.083	0.032	0.063	0.232	0.037	0.024	0.033	0.082	H=16,F=24
	RSM	0.167	0.299	0.196	0.130	0.222	0.065	0.031	0.034	0.190	H=16,F=48
	EDSM	0.040	0.054	0.053	0.078	1.998	0.057	0.300	0.158	1.467	H=16,F=48
	MoP	0.756	0.413	0.315	0.528	0.447	0.667	0.724	0.401	0.541	H=0,F=512
EA-QP	eaQHM	0.214	0.077	0.053	0.036	0.090	0.019	0.017	0.032	0.058	H=16,F=55
	RSM	0.304	0.272	0.121	0.175	0.231	0.090	0.059	0.036	0.041	H=16,F=440
	EDSM	0.183	0.111	0.050	0.074	1.849	0.028	0.106	0.178	1.170	H=16,F=110
	MoP	0.139	0.101	0.062	0.079	0.096	0.094	0.065	0.121	0.076	H=0,F=512
EA-NM	eaQHM	0.554	0.150	0.050	0.090	0.092	0.041	0.028	0.030	0.199	H=16,F=55
	RSM	0.168	0.387	0.128	0.164	0.165	0.111	0.048	0.078	0.131	H=16,F=440
	EDSM	0.124	0.071	0.061	0.085	1.416	0.059	0.272	0.193	0.820	H=16,F=110
	MoP	0.320	0.143	0.099	0.241	0.202	0.121	0.213	0.096	0.124	H=0,F=512
LA-NM	eaQHM	0.480	0.043	0.042	0.314	0.038	0.084	0.010	0.019	0.119	H=16,F=55
	RSM	0.208	0.375	0.128	0.144	0.365	0.058	0.060	0.037	0.108	H=16,F=440
	EDSM	0.100	0.088	0.044	0.061	1.325	0.050	0.033	0.105	0.722	H=16,F=110
	MoP	0.178	0.201	0.147	0.144	0.121	0.121	0.186	0.278	0.119	H=0,F=512

A.2.16 Monotonic dA = -16 dB and dF = 500 Hz

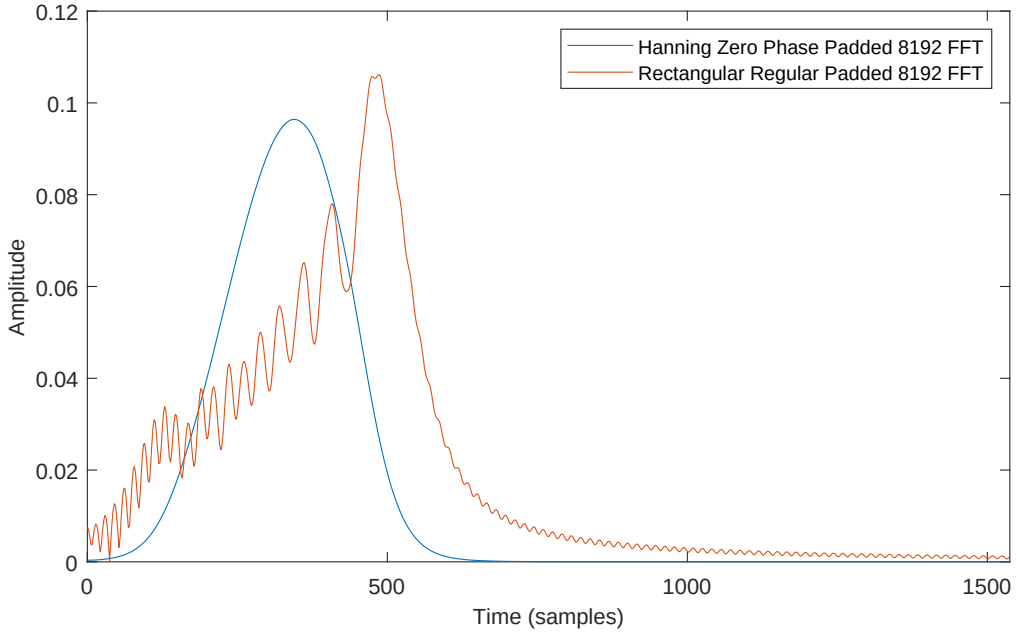
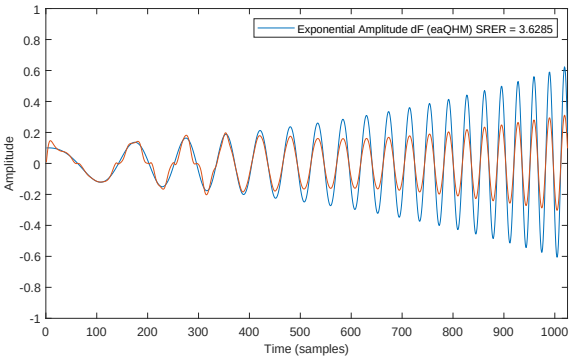
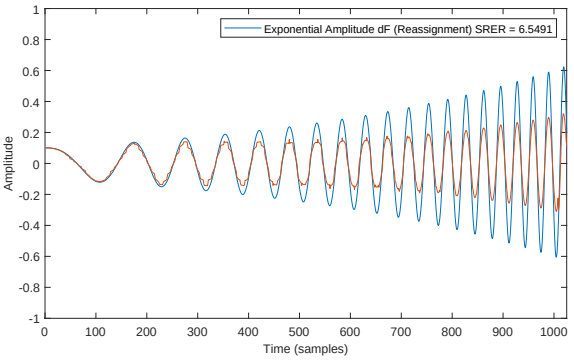


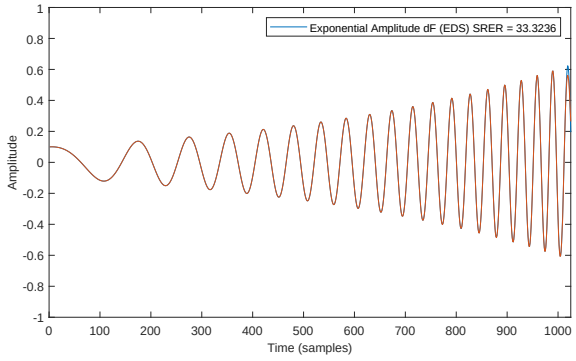
FIGURE A.61: Exponential Amplitude dF FFT



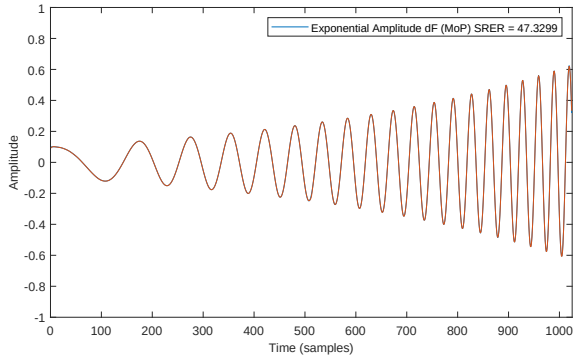
(A) Exponential Amplitude dF eaQHM 8 32



(B) Exponential Amplitude dF Reassignment 8 32

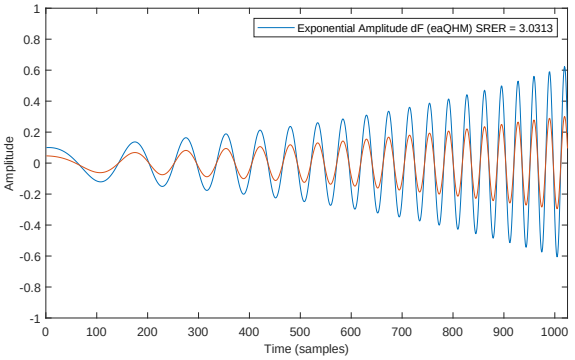


(C) Exponential Amplitude dF EDS 8 32

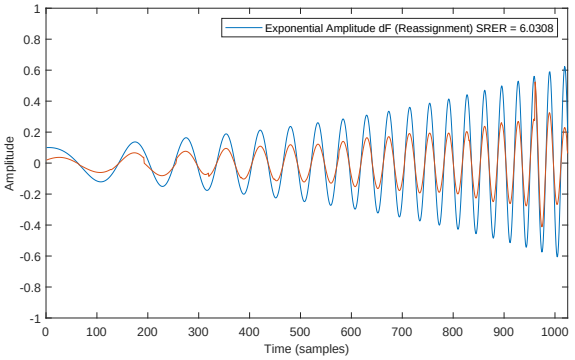


(D) Exponential Amplitude dF MoP 8 32

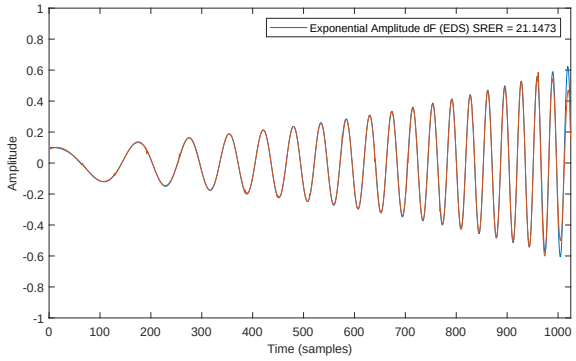
FIGURE A.62: Monotonic Exponential hop=8, window=32



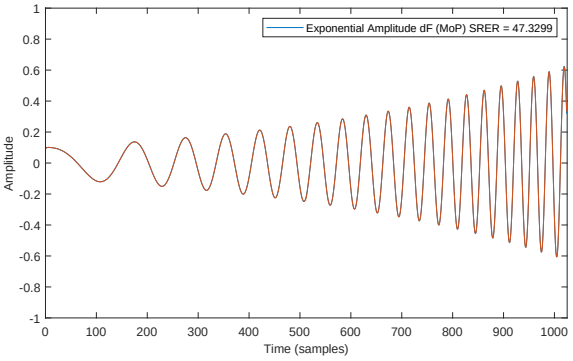
(A) Exponential Amplitude dF eaQHM 64 128



(B) Exponential Amplitude dF Reassignment 64 128



(C) Exponential Amplitude dF EDS 64 128



(D) Exponential Amplitude dF MoP 64 128

FIGURE A.63: Monotonic Exponential hop=64, window=128

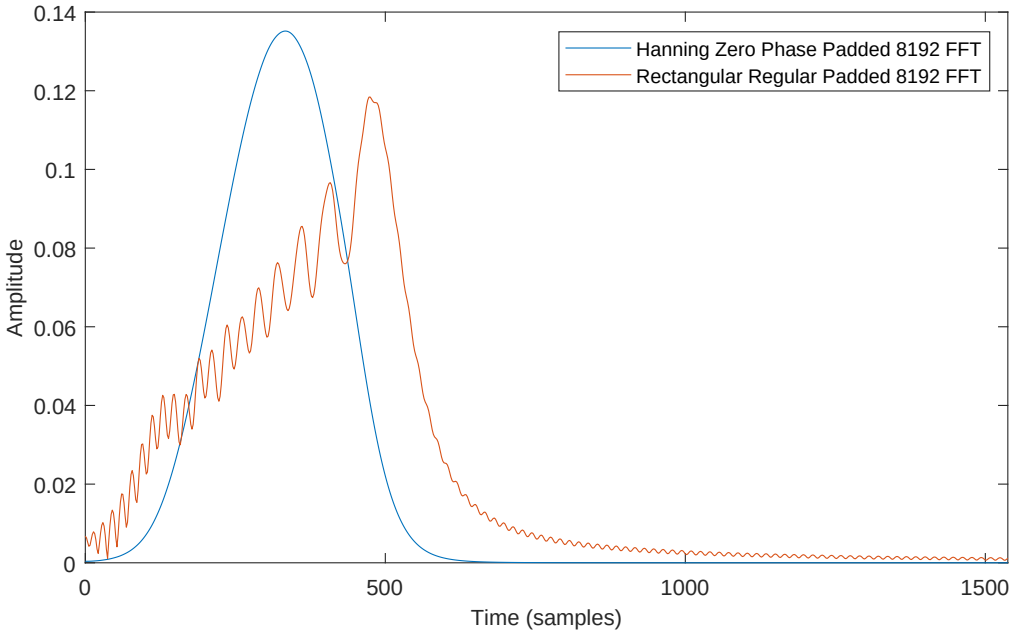
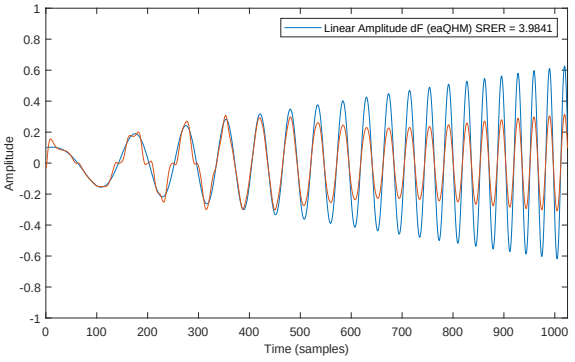
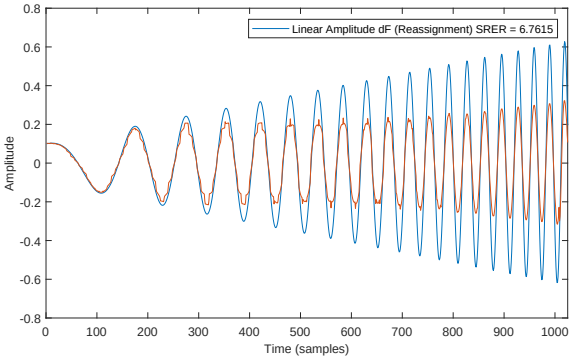


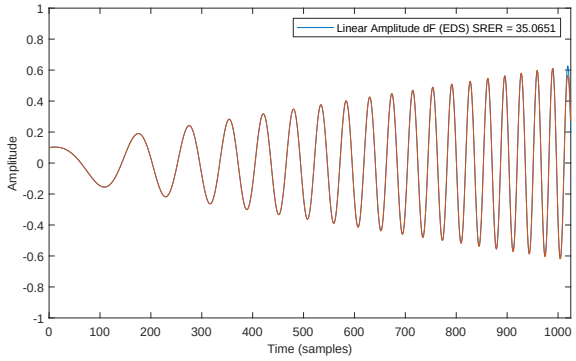
FIGURE A.64: Linear Amplitude dF FFT



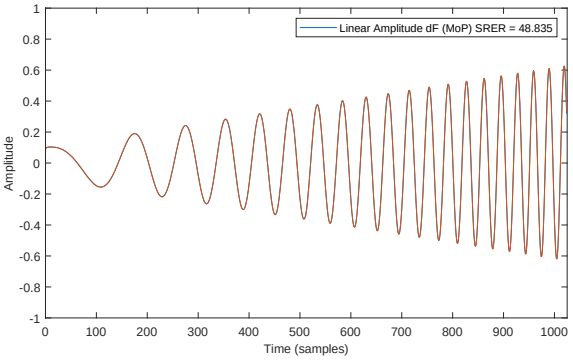
(A) Linear Amplitude dF eaQHM 8 32



(B) Linear Amplitude dF Reassignment 8 32

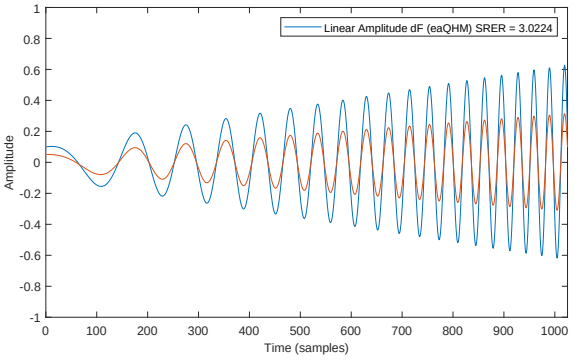


(C) Linear Amplitude dF EDS 8 32

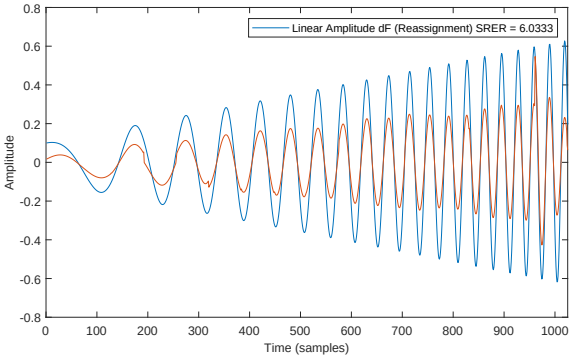


(D) Linear Amplitude dF MoP 8 32

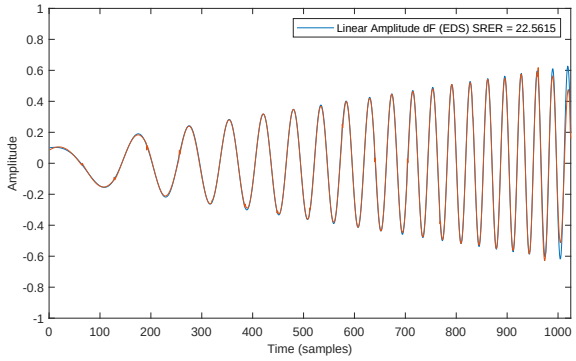
FIGURE A.65: Monotonic Linear hop=8, window=32



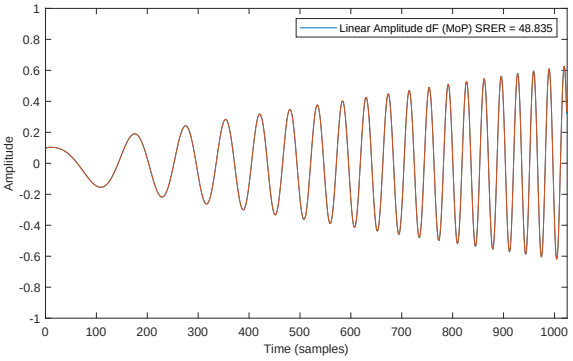
(A) Linear Amplitude dF eaQHM 64 128



(B) Linear Amplitude dF Reassignment 64 128



(C) Linear Amplitude dF EDS 64 128



(D) Linear Amplitude dF MoP 64 128

FIGURE A.66: Monotonic Linear hop=64, window=128

Amplitude Curve	Hop Size	Window Size	eaQHM	Reassignment	EDSM	MoP
Exponential	8	32	3.628479	6.549085	33.323619	47.329877
Exponential	64	128	3.031349	6.030819	21.147327	47.329877
Linear	8	32	3.984062	6.761536	35.065089	48.835016
Linear	64	128	3.022397	6.033307	22.561468	48.835016

TABLE A.7: SRER Monotonic dA and Large dF

Amplitude Curve	Hop Size	Window Size	eaQHM	Reassignment	EDSM	MoP
Exponential	8	32	0.061766	0.120591	0.039137	0.193360
Exponential	64	128	0.028919	0.020961	0.025808	0.184694
Linear	8	32	0.161514	0.181353	0.062378	0.273954
Linear	64	128	0.089347	0.067447	0.058280	0.255371

TABLE A.8: MIPS Monotonic dA and Large dF

A.2.17 Non-Monotonic dA = -16 dB and dF = 500 Hz

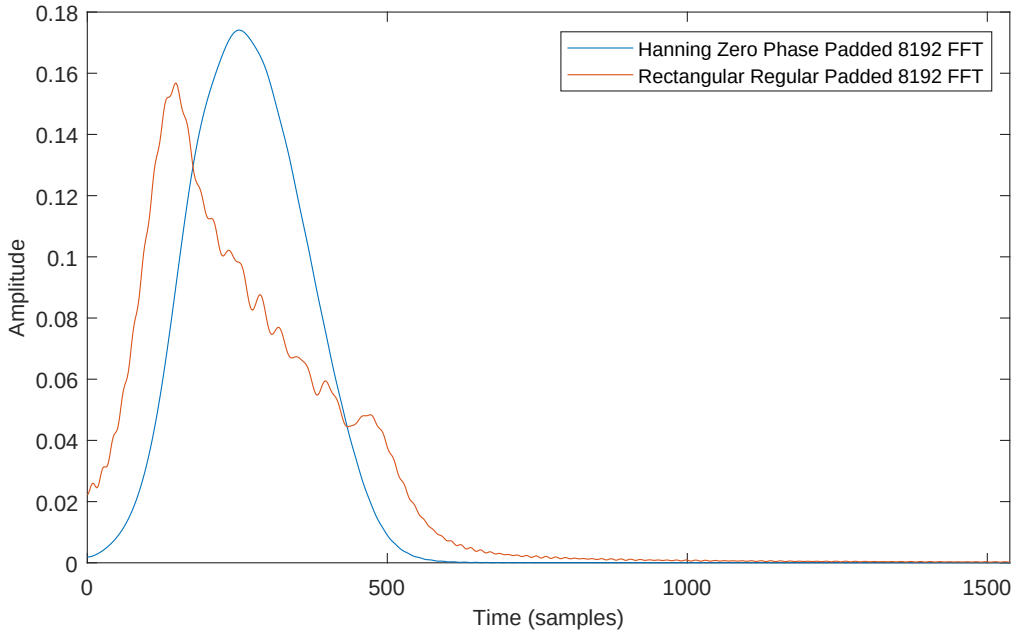
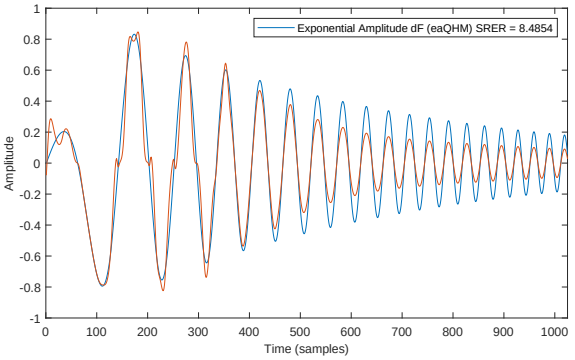
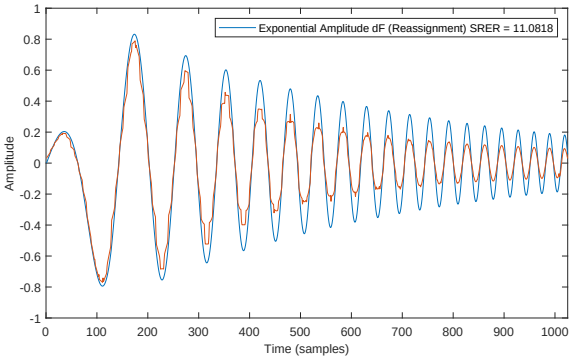


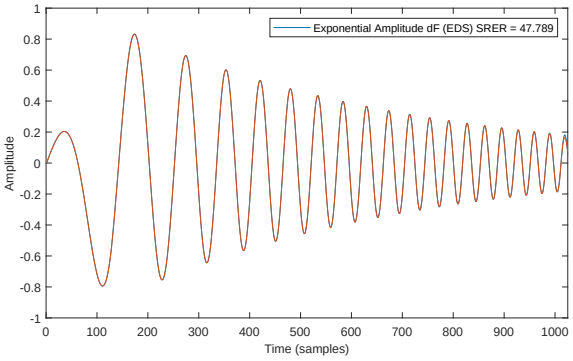
FIGURE A.67: Exponential nm dA dF FFT



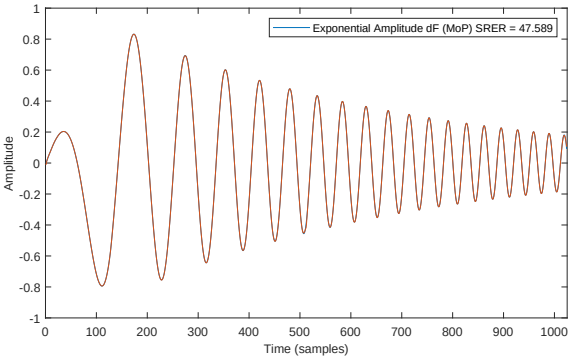
(A) Exponential nm dA dF eaQHM 8 32



(B) Exponential nm dA df Reassignment 8 32

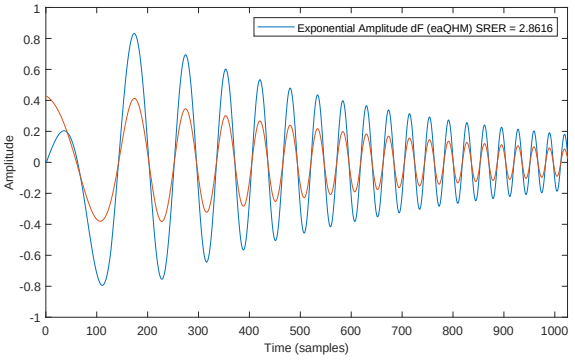


(C) Exponential nm dA dF EDS 8 32

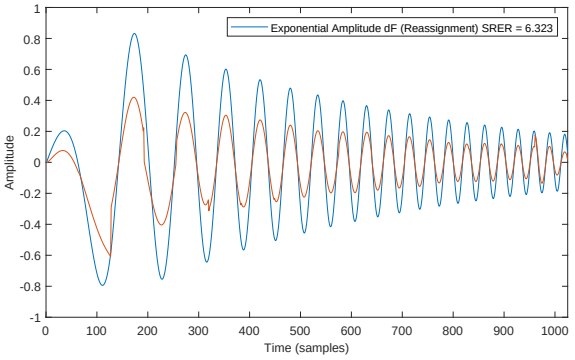


(D) Exponential nm dA dF MoP 8 32

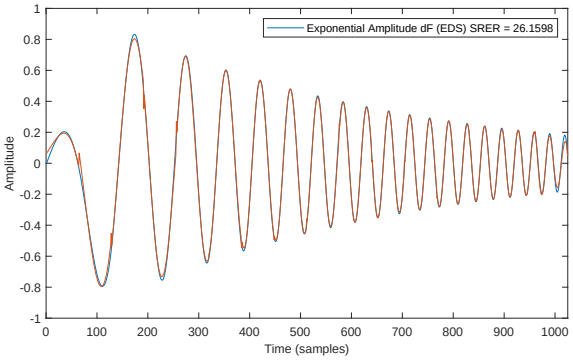
FIGURE A.68: Non-Monotonic Exponential hop=8, window=32



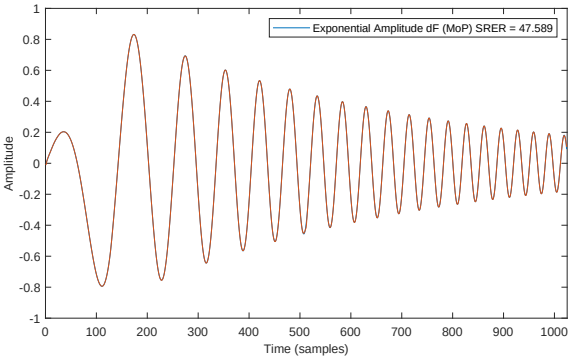
(A) Exponential nm dA dF eaQHM 64 128



(B) Exponential nm dA dF Reassignment 64 128



(C) Exponential nm dA dF EDS 64 128



(D) Exponential nm dA dF MoP 64 128

FIGURE A.69: Non-Monotonic Exponential hop=64, window=128

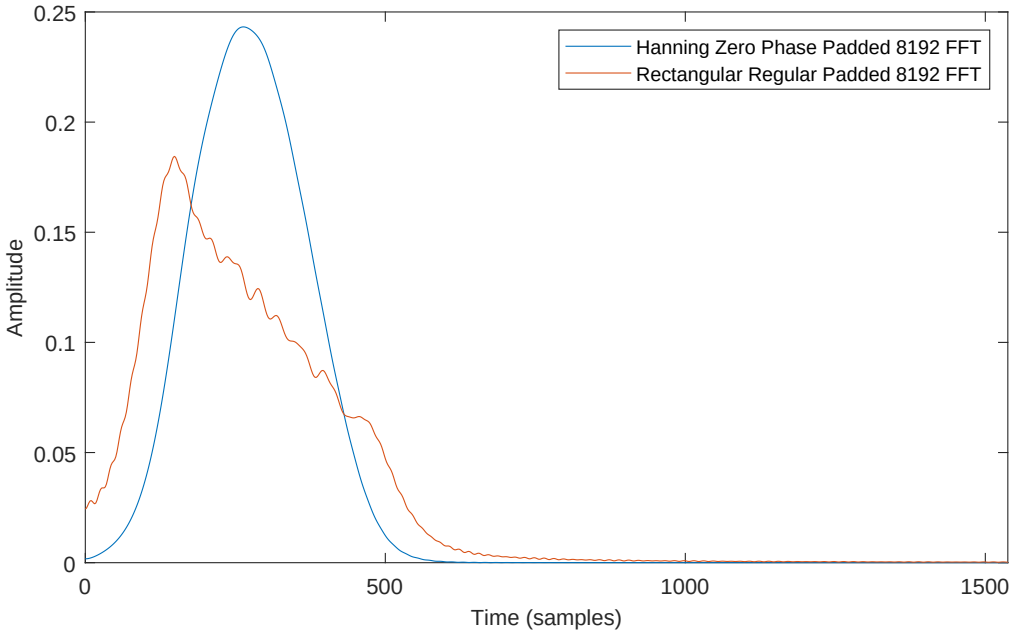
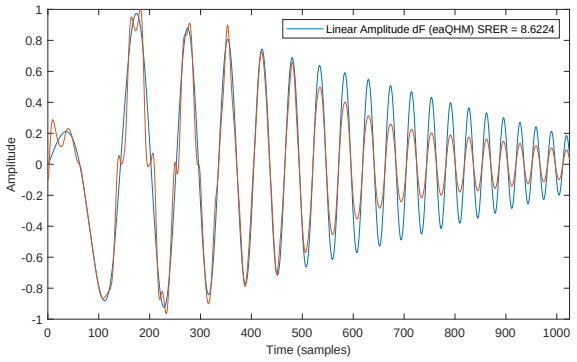
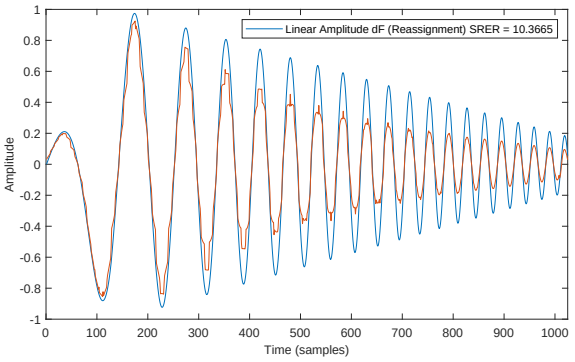


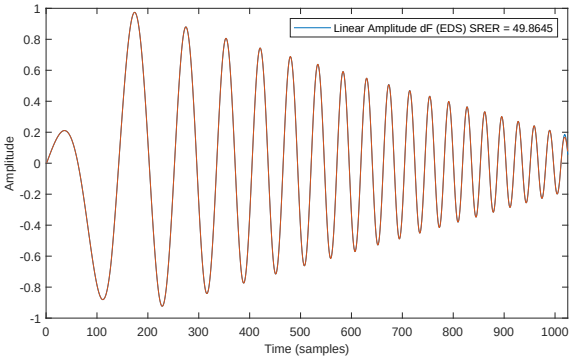
FIGURE A.70: Linear nm dA dF FFT



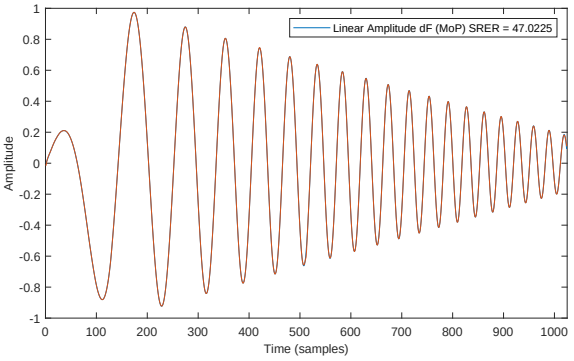
(A) Linear nm dA dF eaQHM 8 32



(B) Linear nm dA dF Reassignment 8 32

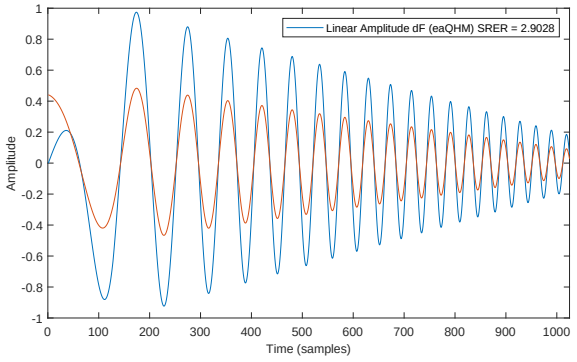


(C) Linear nm dA dF EDS 8 32

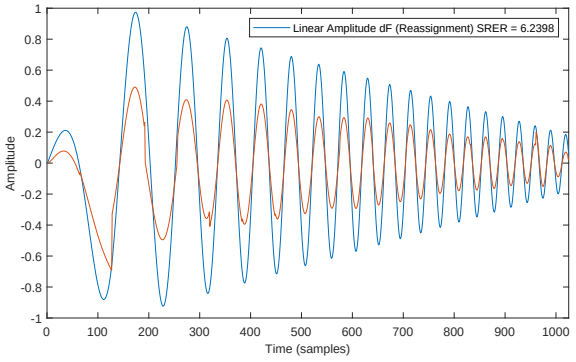


(D) Linear nm dA dF MoP 8 32

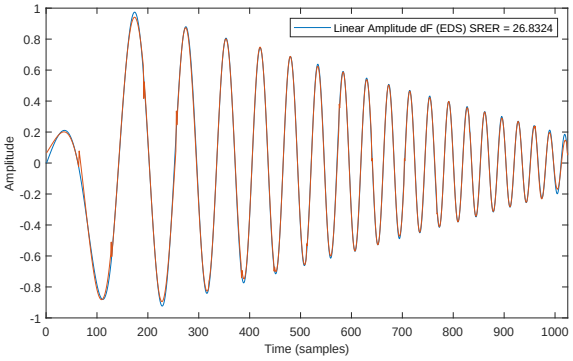
FIGURE A.71: Non-Monotonic Linear hop=8, window=32



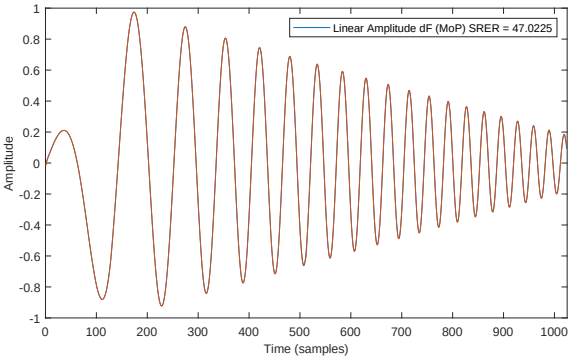
(A) Linear nm dA dF eaQHM 64 128



(B) Linear nm dA dF Reassignment 64 128



(C) Linear nm dA dF EDS 64 128



(D) Linear nm dA dF MoP 64 128

FIGURE A.72: Non-Monotonic Linear hop=64, window=128

Amplitude Curve	Hop Size	Window Size	eaQHM	Reassignment	EDSM	MoP
Exponential	8	32	8.485436	11.081811	47.789034	47.588994
Exponential	64	128	2.861611	6.323050	26.159758	47.588994
Linear	8	32	8.622400	10.366513	49.864514	47.022511
Linear	64	128	2.902772	6.239761	26.832407	7.022511

TABLE A.9: SRER Non-Monotonic dA and Large dF

Amplitude Curve	Hop Size	Window Size	eaQHM	Reassignment	EDSM	MoP
Exponential	8	32	0.071885	0.133287	0.031617	0.217553
Exponential	64	128	0.031369	0.020885	0.029844	0.168090
Linear	8	32	0.305141	0.302936	0.116105	0.321516
Linear	64	128	0.098539	0.075704	0.073089	0.244399

TABLE A.10: MIPS Non-Monotonic dA and Large dF

Appendix B

Sound Examples

B.1 Example of Mastering and Compression on Kick and Bass

This section displays the process of creating a kick and bass line. A kick drum and a bass line, including the combined output are displayed in Figure B.1. It can is also displayed that the end of the kick drum does overlap with the bass line. A common technique for improving the low end separation of this is to use side-chain compression sop that the volume of the kick drum or bass is reduced when overlapping. This example does not include any side chain compression, ultimately the use of these tools comes down to the producer preference and years of experience in achieving the desired sound. In this case the lack of side-chain compression has the effect of adding a lot of punch and more of an attack to the bass.

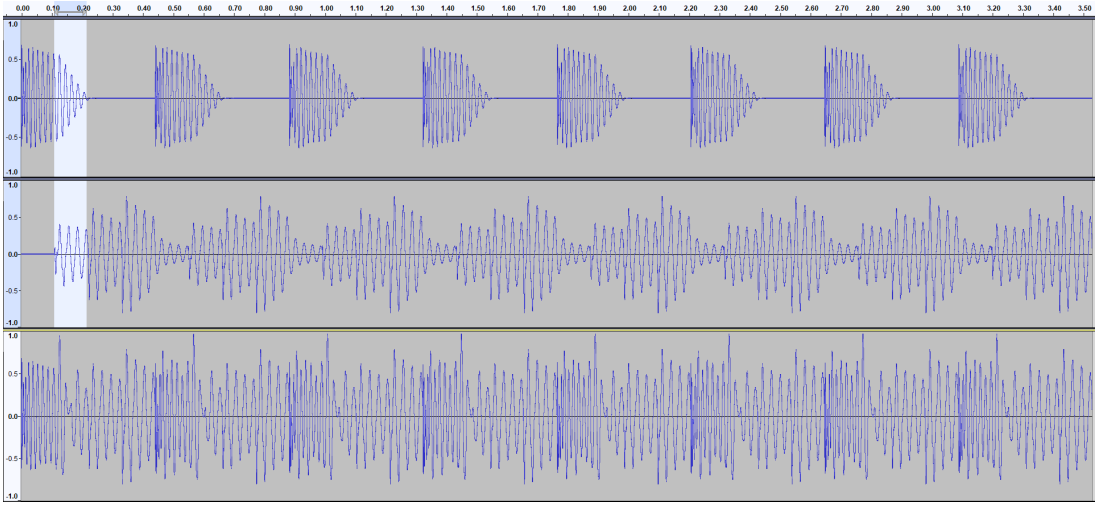


FIGURE B.1: Example of creating a Kick and Bass

Some of the mastering plugins used for processing the kick and bass after reducing it by 6 dB for extra headroom include: the Manley Massive Passive Equaliser, Elysia Compressor, Shadow Hills Mastering Compressor, and Precision Multiband Compressor from UAD [266]. Numerous other equalisers, transient effect processors and other mastering tools could also be used.

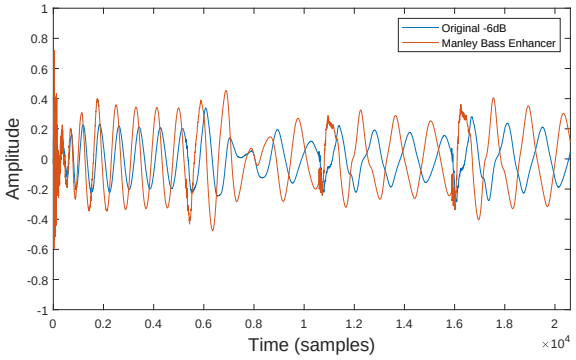
(A) Precision Multiband Compressor

(B) Shadow Hills Mastering Compressor

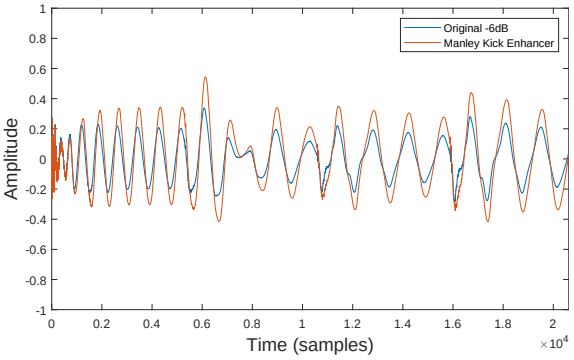
(C) Elysia Compressor

(D) Manley Massive Passive Equaliser

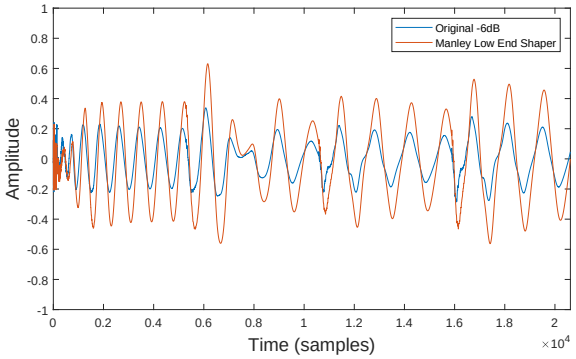
FIGURE B.2: Examples of Mastering Equalisers and Compressors



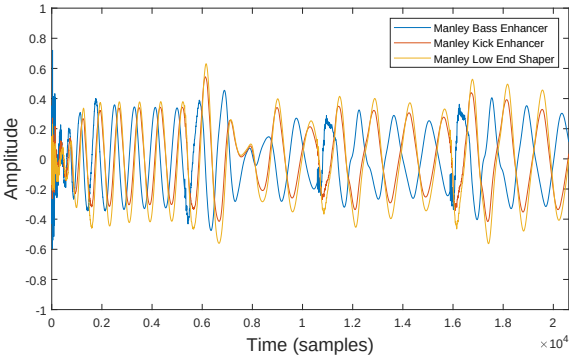
(A) Massive Passive Equaliser Bass Enhancer



(B) Massive Passive Equaliser Kick Enhancer

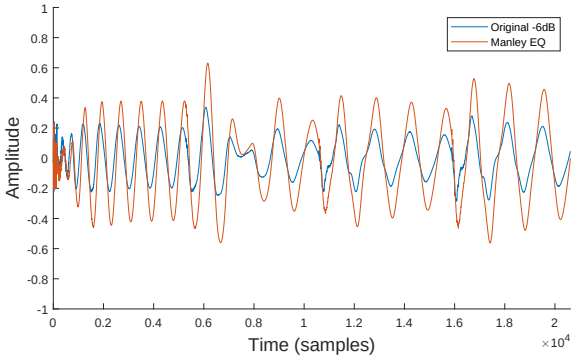


(C) Massive Passive Equaliser Low End Shaper

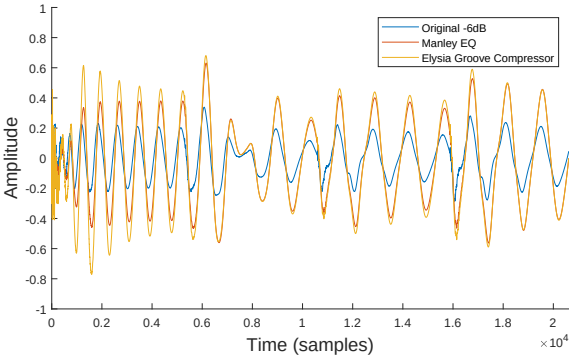


(D) Different Compressor Settings

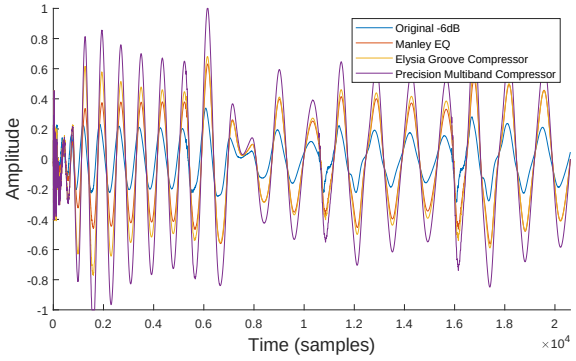
FIGURE B.3: Effect of different Manley Massive Passive Equaliser presets on Kick and Bass



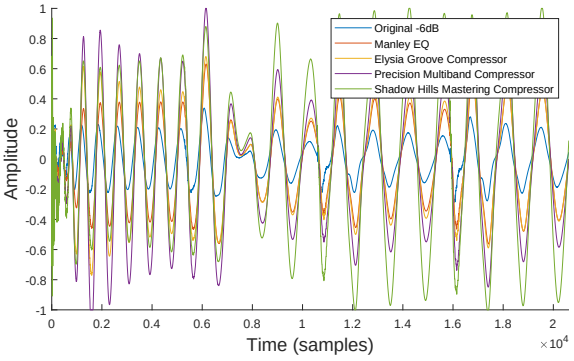
(A) Original vs Manley Bass Enhancer



(B) Original Manley Elysia

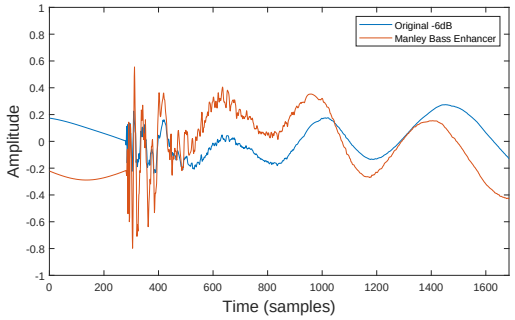


(C) Original Manley Elysia Precision Multiband

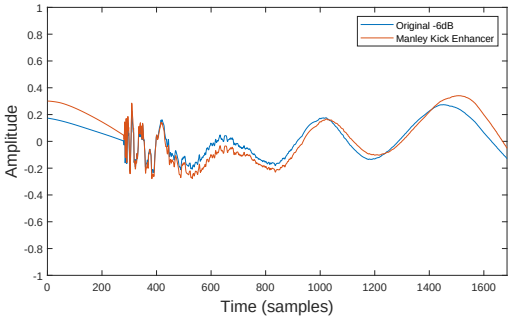


(D) Original Manley Elysia Precision Shadowhills

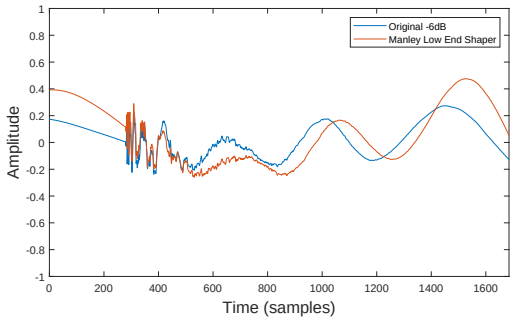
FIGURE B.4: Effect of different compressors on kick and bass



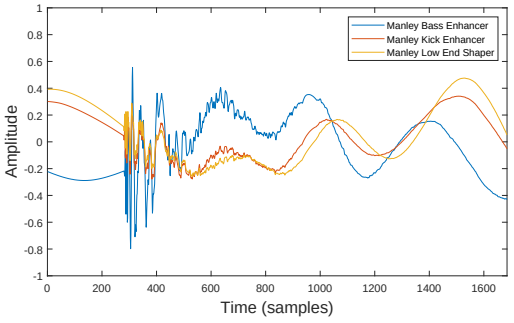
(A) Massive Passive Equaliser Bass Enhancer



(B) Massive Passive Equaliser Kick Enhancer

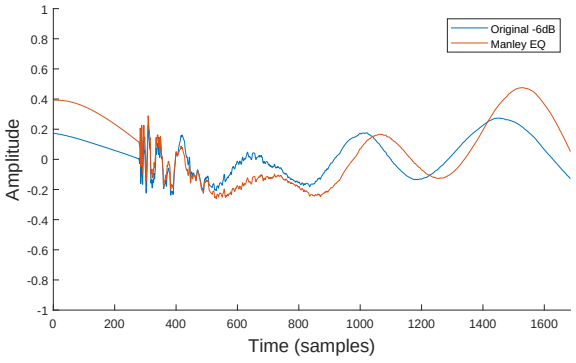


(C) Massive Passive Equaliser Low End Shaper

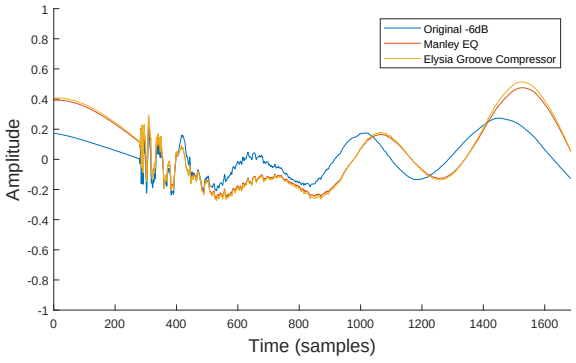


(D) Comparison of different Manley Compressor presets and the effect on kick and bass

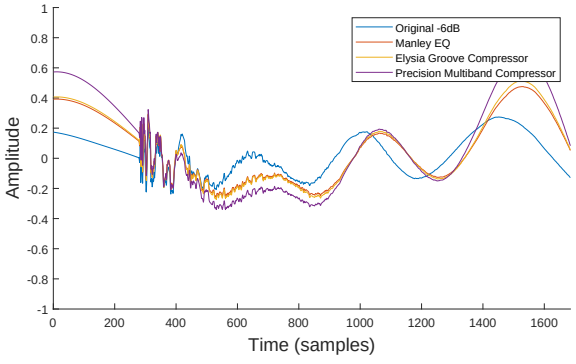
FIGURE B.5: Effect of Massive Passive Equaliser



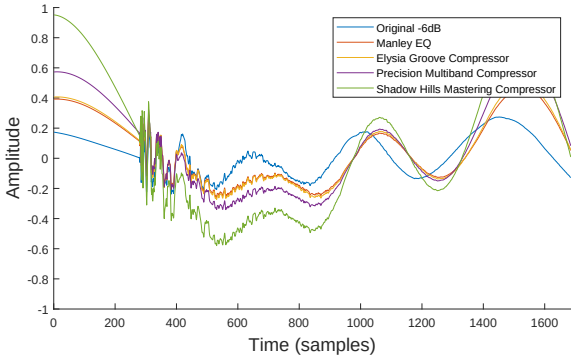
(A) Manley Massive Passive Equaliser Bass Enhancer



(B) Manley and Elysia Groove Compressor



(C) Manley, Elysia and Precision Multiband



(D) Manley, Elysia and Precision and Shadow Hills

FIGURE B.6: Effect of different compressors on Kick Drum

B.2 MoP Examples

In the current section a number of audio snippets extracted from various EDM musical genres have been selected. Each of the 14 examples has been analysed and re-synthesised using the causal implementation of MoP which uses a rectangular window for analysis. Exponential amplitude is presumed as the current implementation has net yet been extended to take linear phase difference measurements into account. Changes in frequency are also omitted from the current implementation when re-synthesising sinusoidal atoms. The measurements for 3 tests were taken with the maximum number of sinusoidal partials set to 128, 256 and 512 respectively, and a frame size of 1024 samples with no overlap. Amplitude and Phase coherence is not implemented between successive frames as no modifications are applied and the reconstructed signal with this method is in general free of any artifacts between frame boundaries. The number of sinusoidal partials used by the model has a direct impact on this.

The results of the original audio are compared to the output of the synthesised audio. Residual components from the different number of sinusoidal partials used to model the sound are presented.

The full list of audio examples as 48kHz WAV files is available with the full submission of the thesis. MP3 versions of the input and resulting output files are embedded below, and are available for playback within the document as MPEG files.

B.2.1 Shadow Fx and Interpulse - Reflexion:

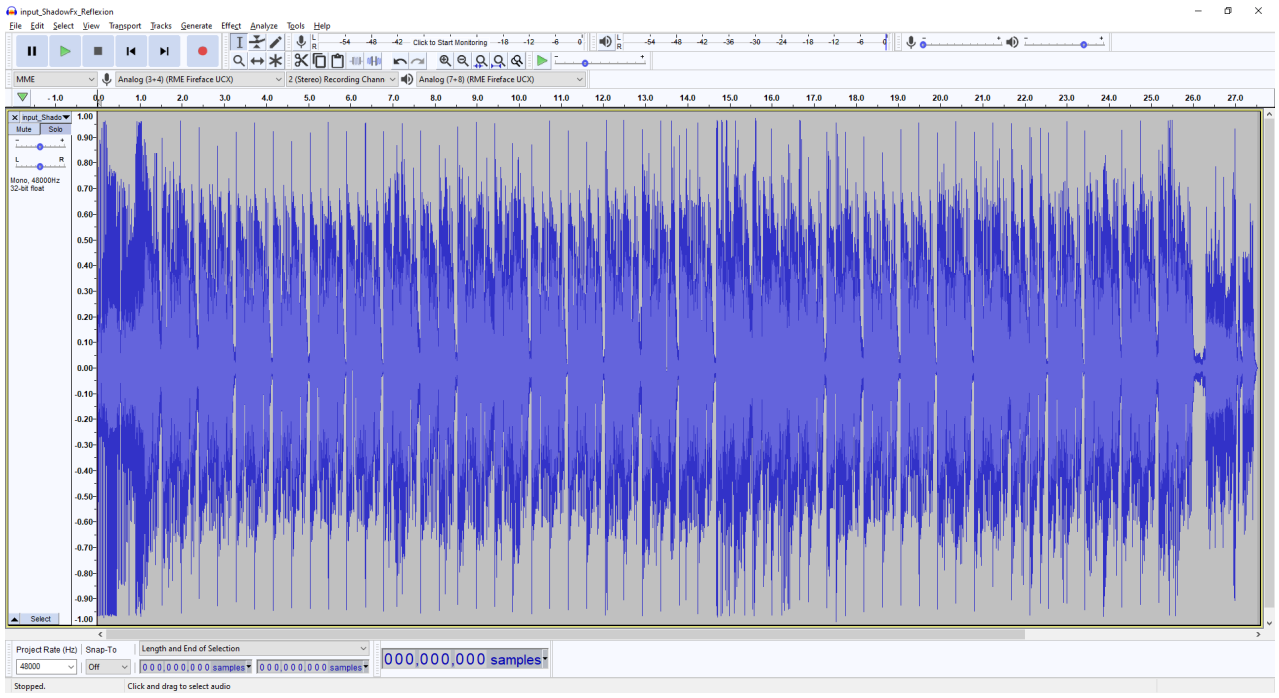
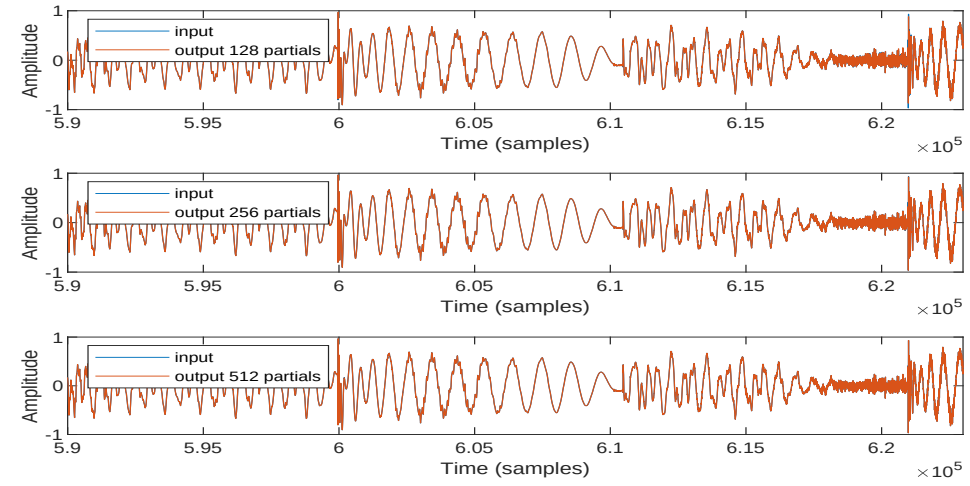
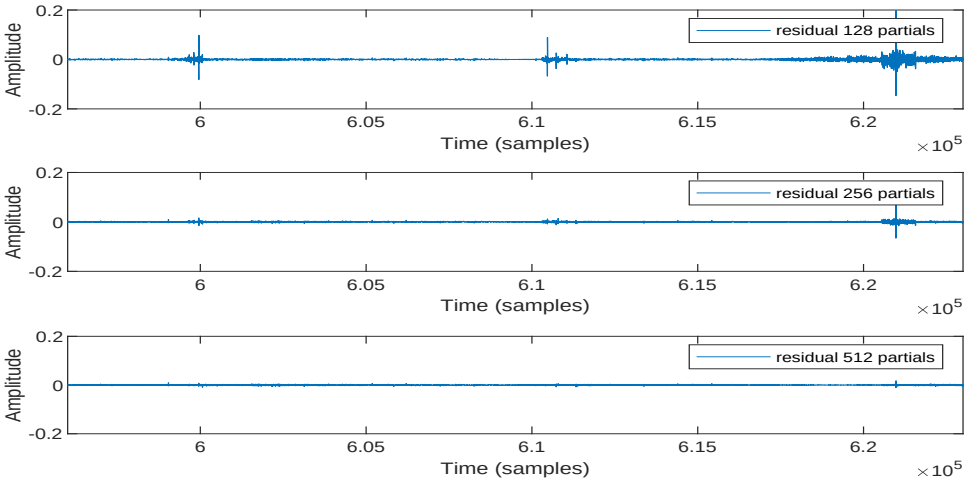


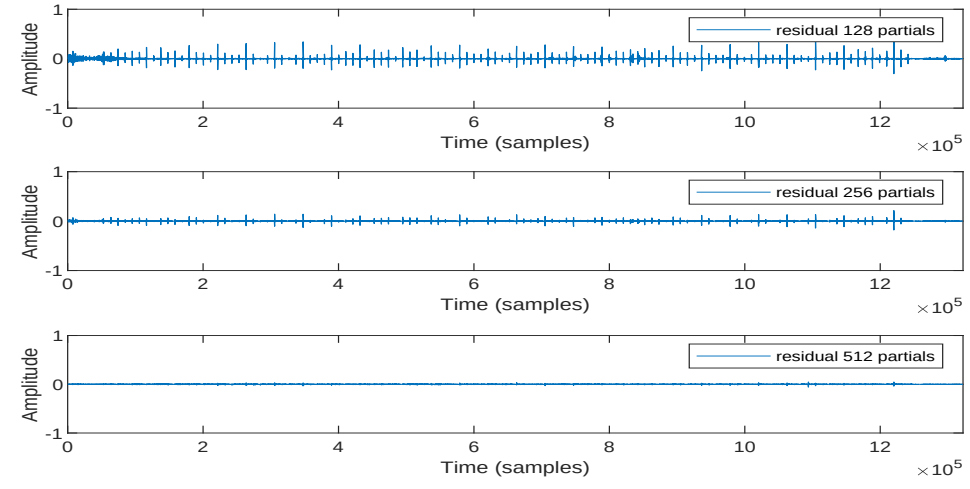
FIGURE B.7: Excerpt from Shadow Fx and Interpulse - Reflexion [267]



(A) Input vs Output



(B) Zoomed in section of Residual Signal



(C) Entire Residual Signal

FIGURE B.8: Output of ShadowFx MoP Modelling using different number of maximum allowed partials

B.2.2 Lumen - Gruntled:

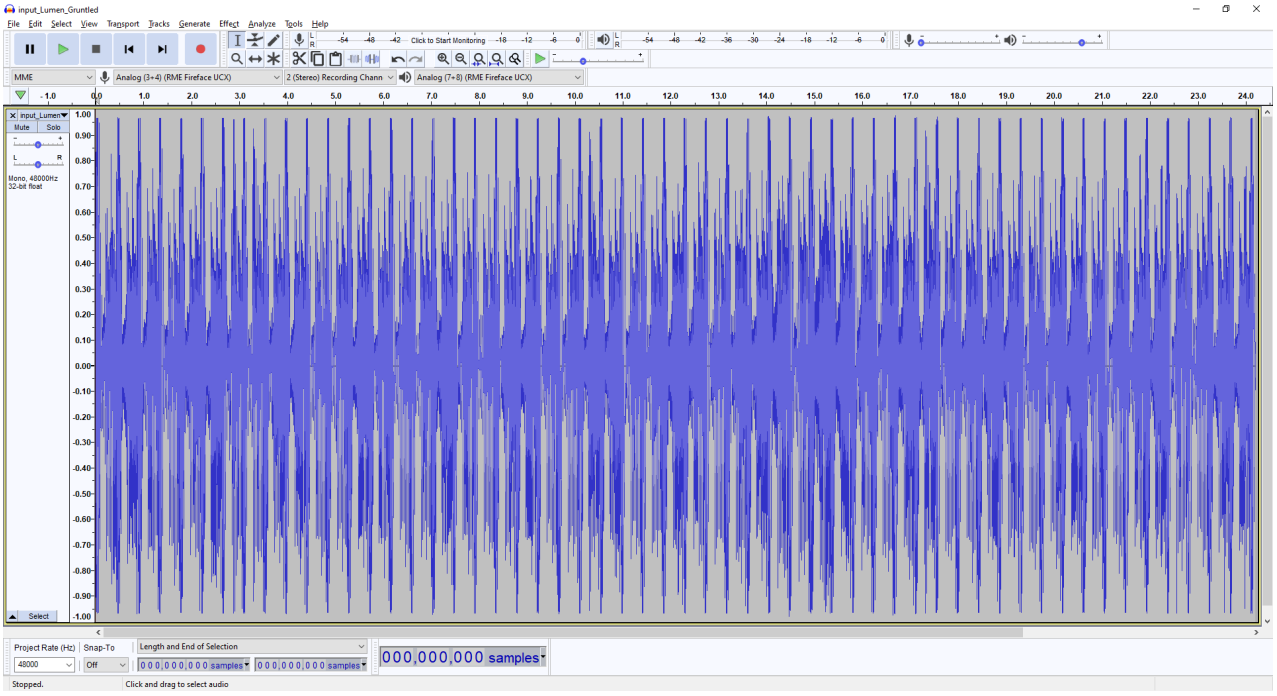
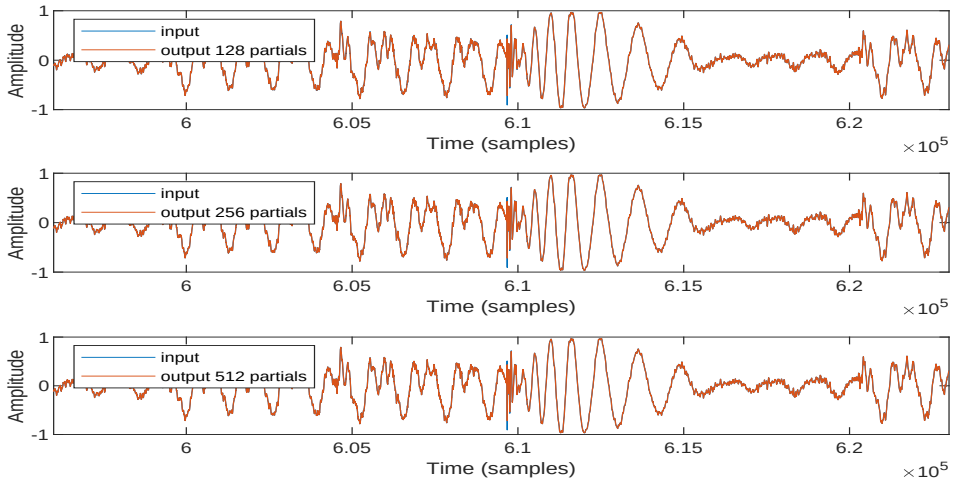
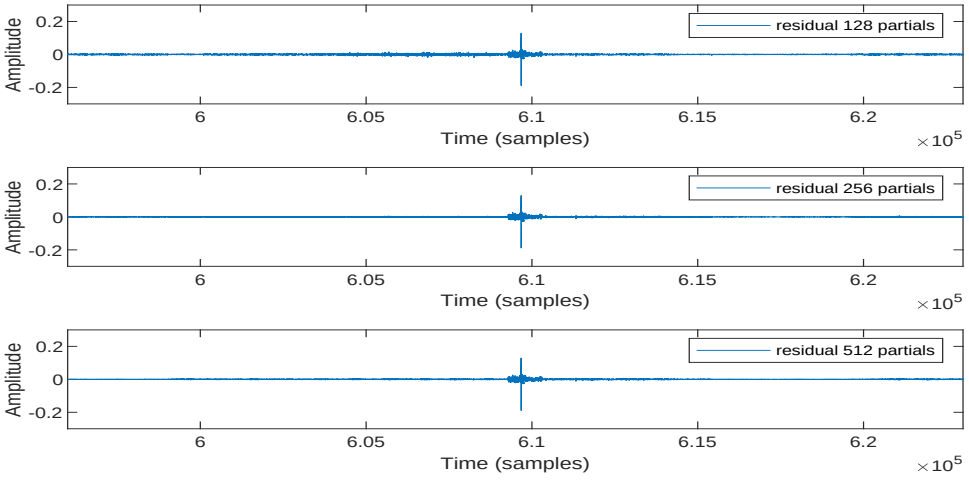


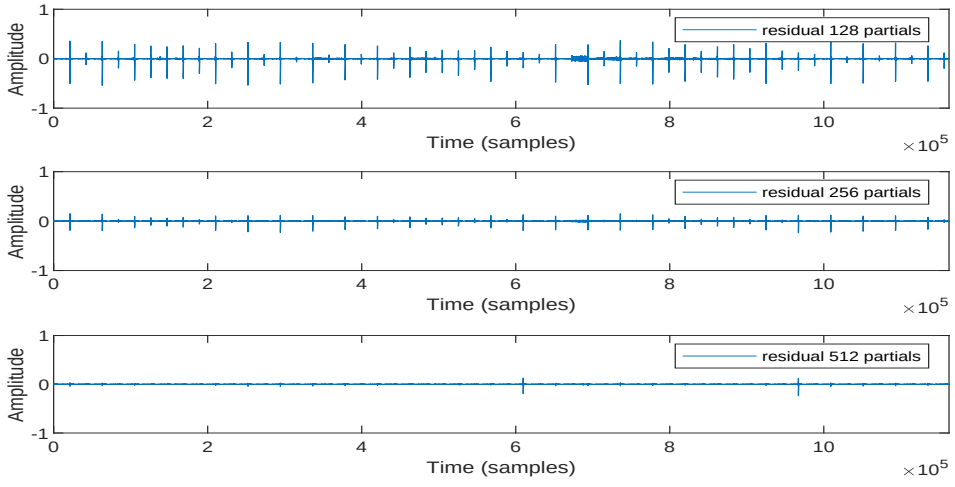
FIGURE B.9: Excerpt from Lumen - Gruntled [268]



(A) Input vs Output



(B) Zoomed in section of Residual Signal



(C) Entire Residual Signal

FIGURE B.10: Output of Lumen MoP Modelling using different number of maximum allowed partials

B.2.3 Pspiralife - Macro Micro:

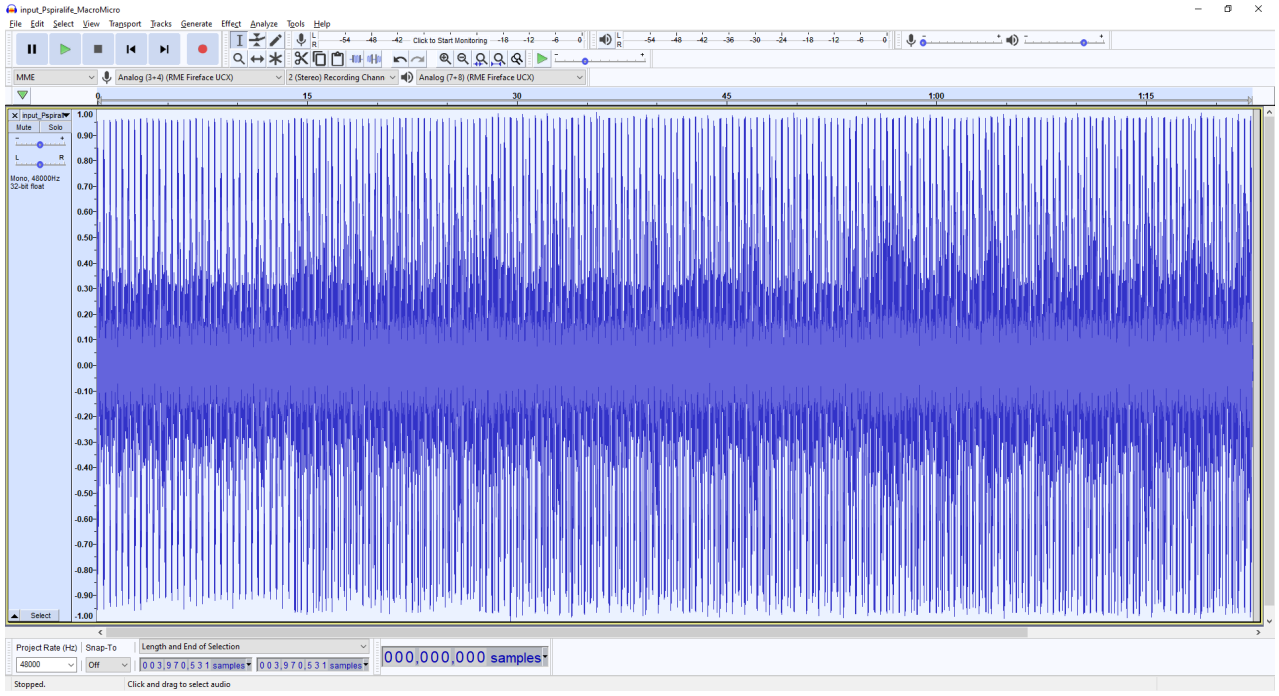
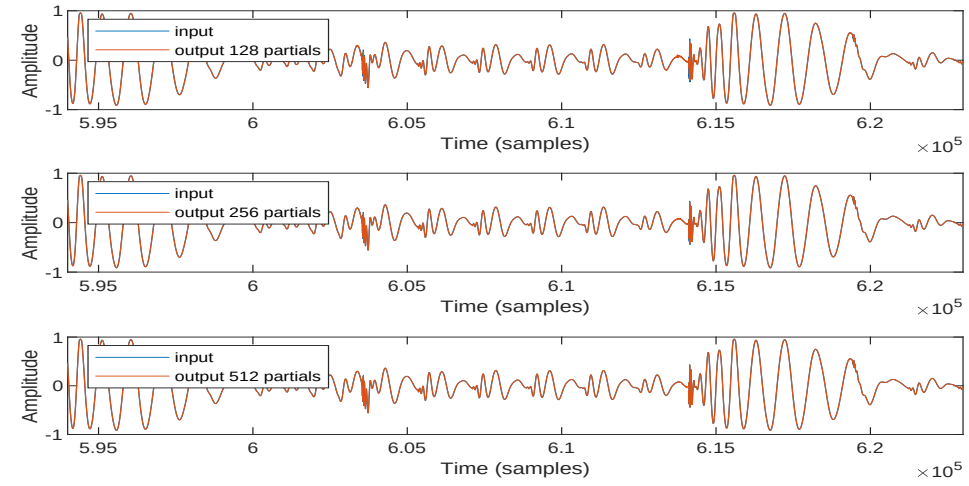
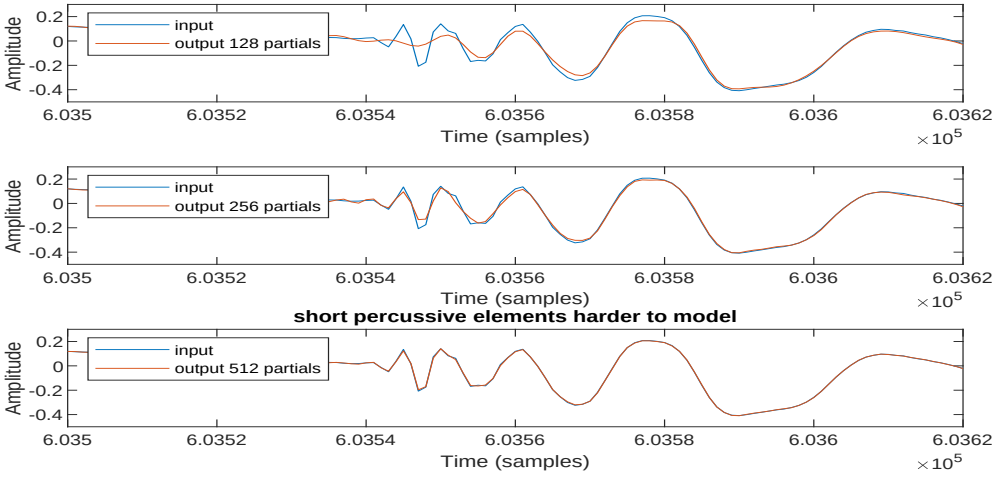


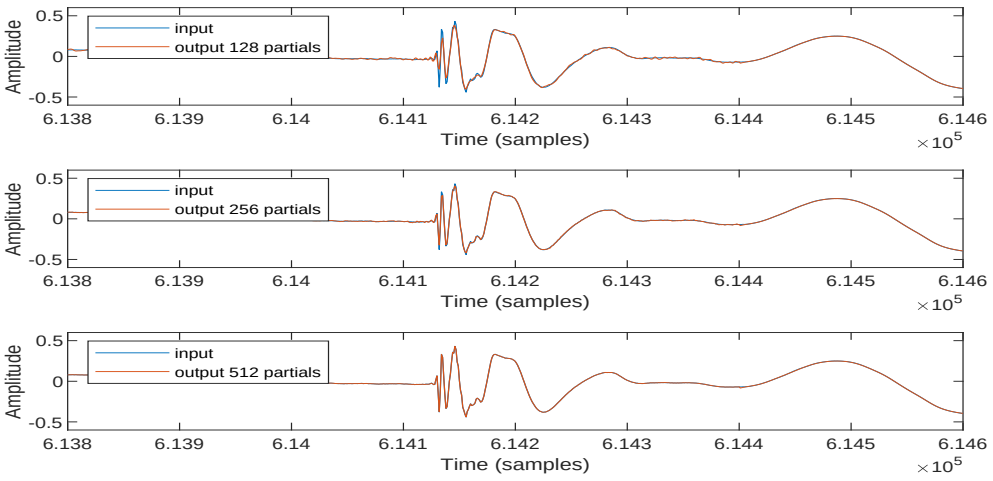
FIGURE B.11: Excerpt from Lumen - Gruntled [269]



(A) Input vs Output

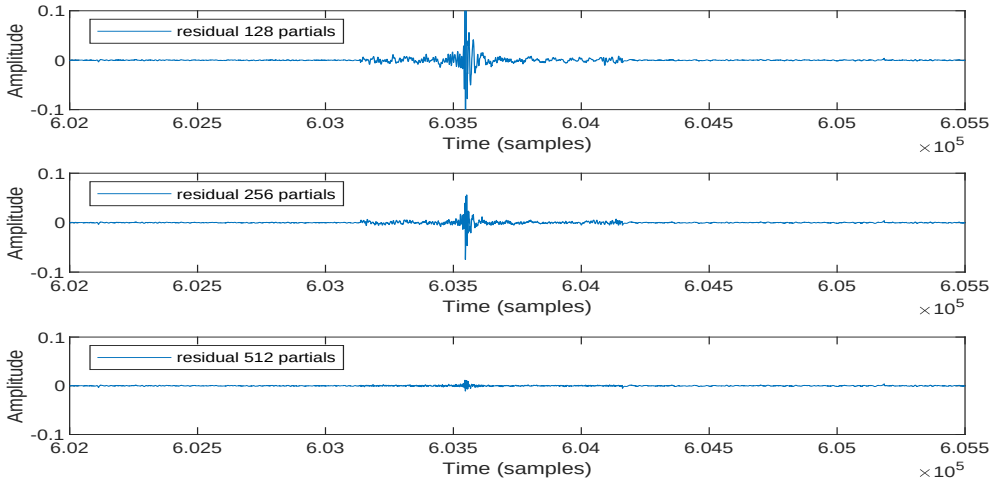


(B) Zoomed in section of Input and Output Signals

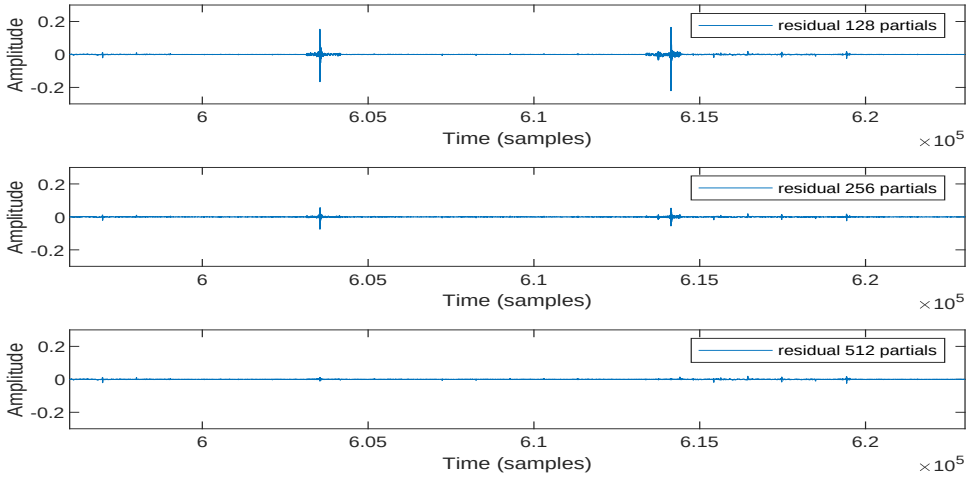


(C) Zoomed in section of Input and Output Signals

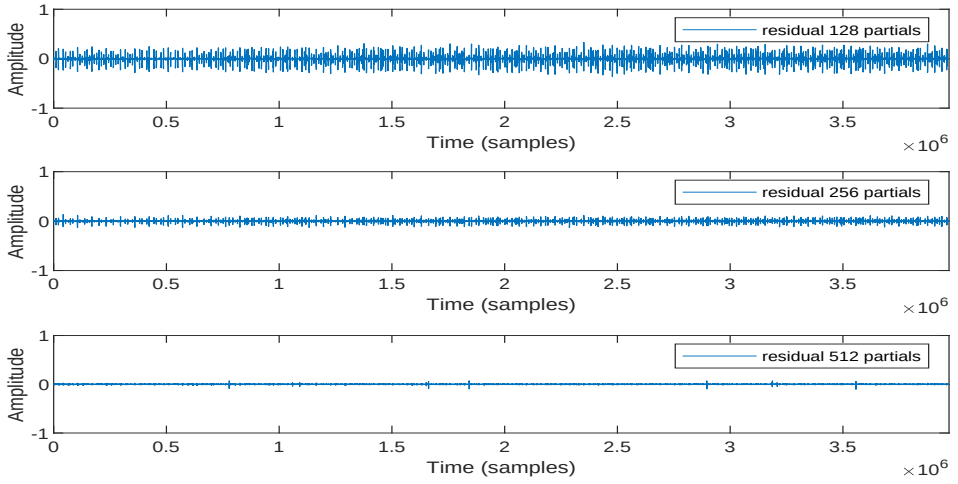
FIGURE B.12: Output of Pspiralife MoP Modelling using different number of maximum allowed partials



(A) Zoomed in section of Residual Signal



(B) Zoomed in section of Residual Signal



(C) Entire Residual Signal

FIGURE B.13: Residuals of Pspiralife MoP Modelling using different number of maximum allowed partials

B.2.4 Sébastien Léger - Son Of Sun:

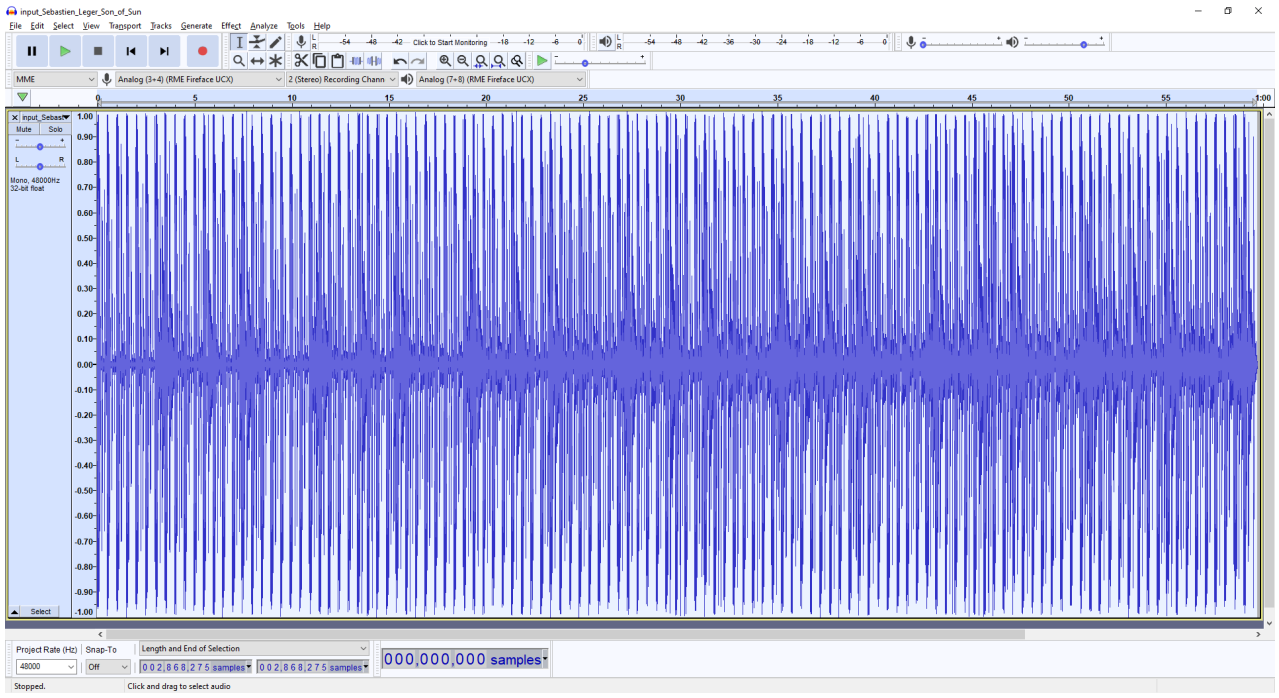
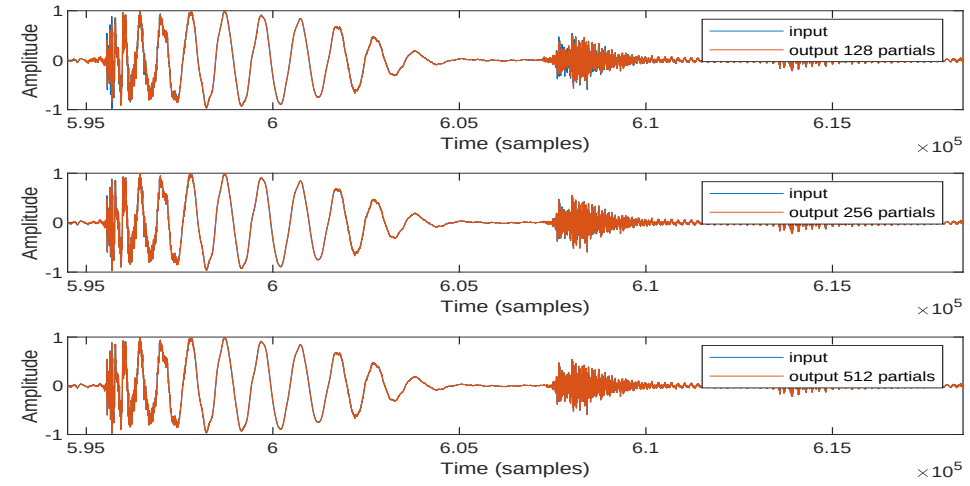
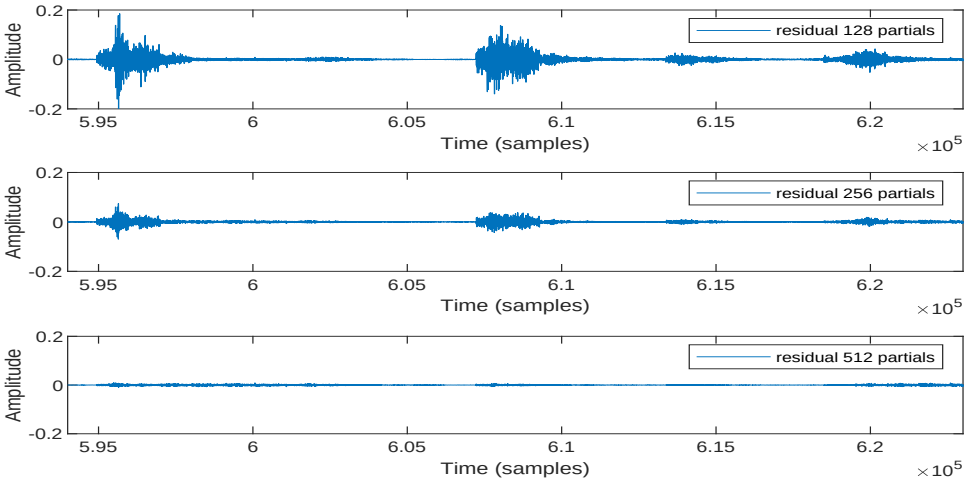


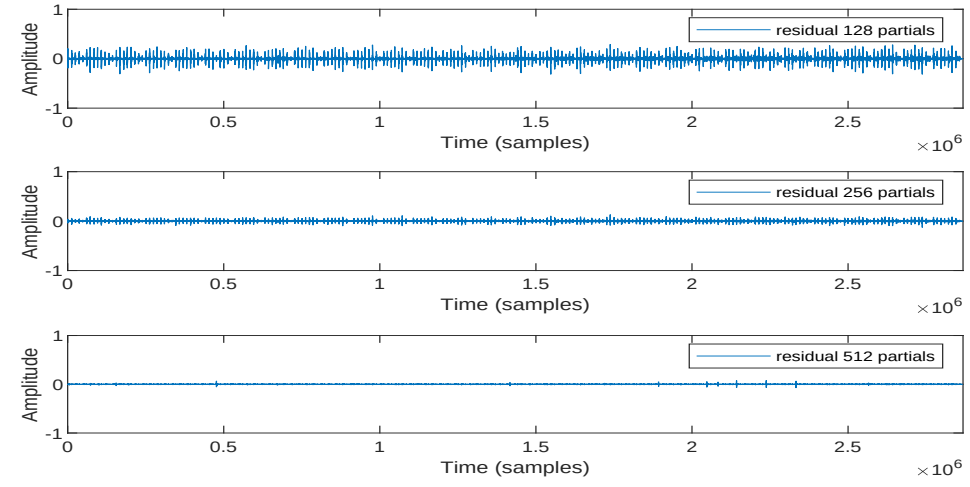
FIGURE B.14: Excerpt from Sébastien Léger - Son Of Sun [270]



(A) Input vs Output



(B) Zoomed in section of Residual Signal



(C) Entire Residual Signal

FIGURE B.15: Output of Sebastien Leger MoP Modelling using different number of maximum allowed partials

B.2.5 Hermanez - Tale of the Unexpected:

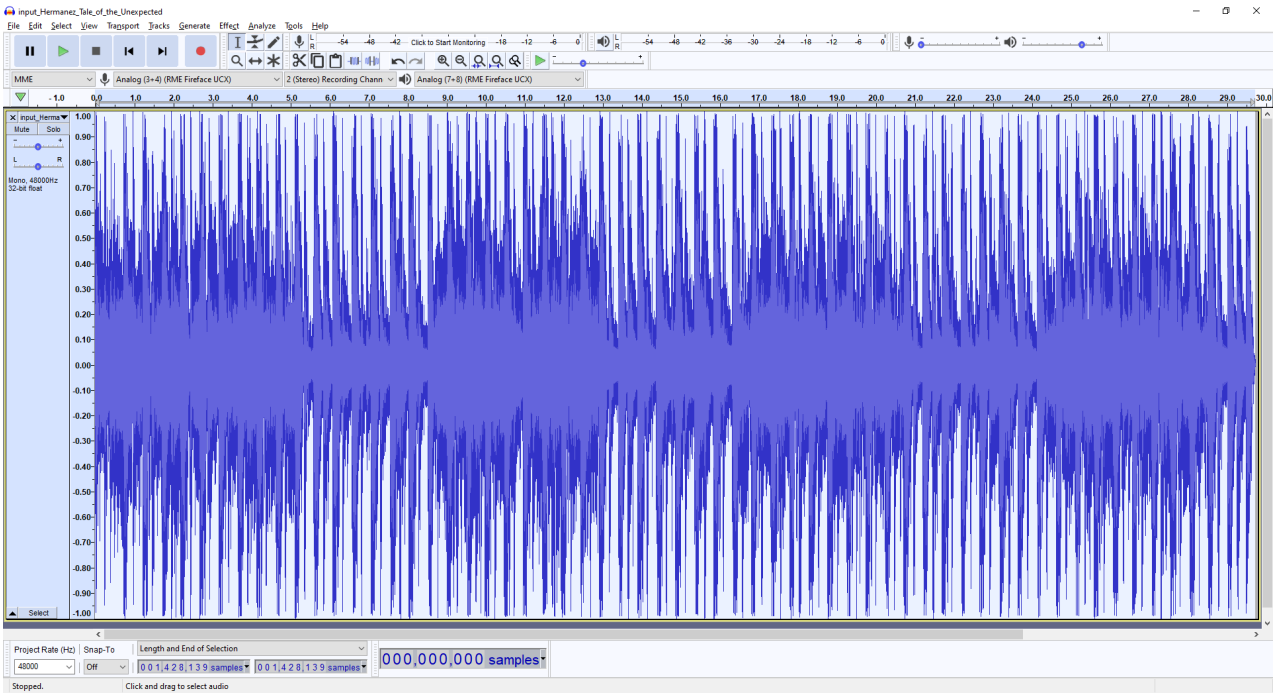
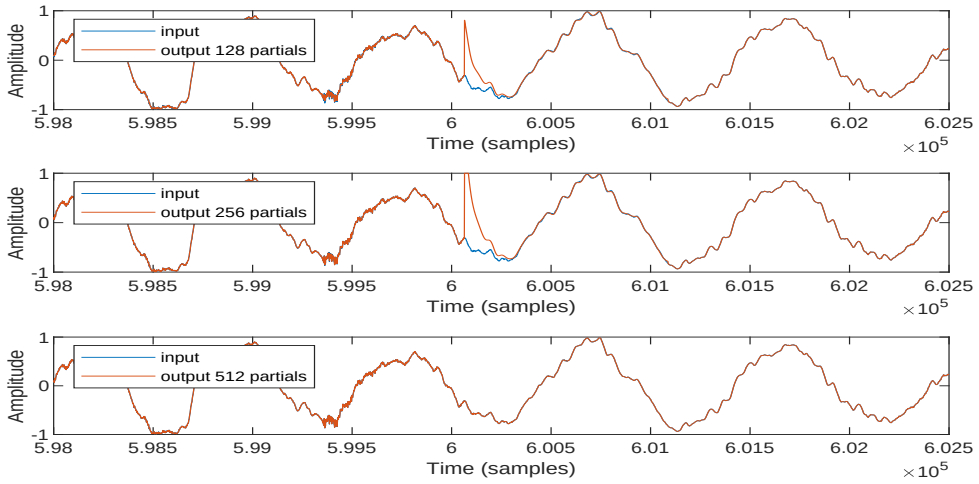
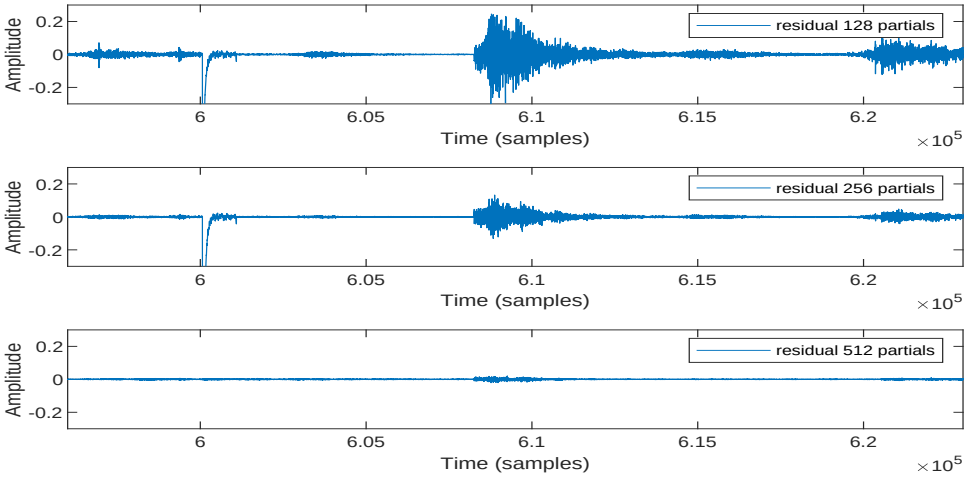


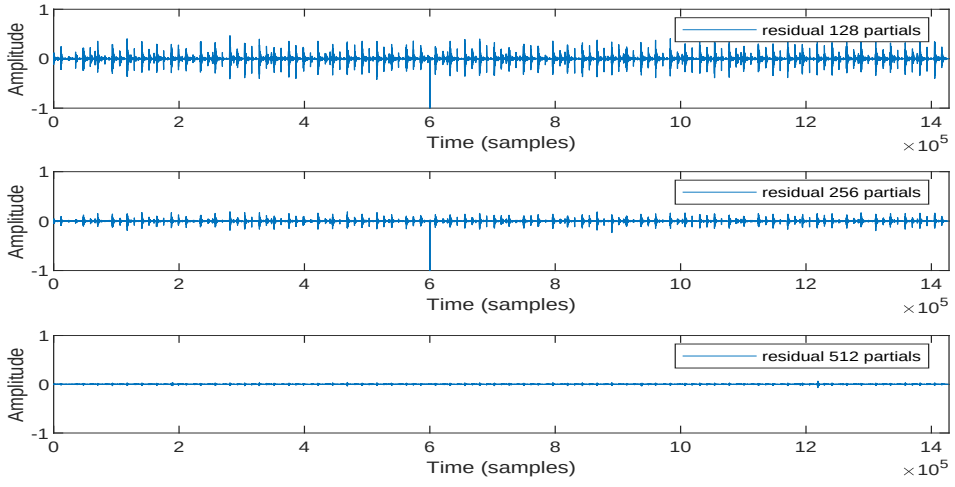
FIGURE B.16: Excerpt from Hermanez - Tale of the Unexpected [271]



(A) Zoomed in section of Input vs Output



(B) Zoomed in section of Residual Signal



(C) Entire Residual Signal

FIGURE B.17: Output of HermanezT MoP Modelling using different number of maximum allowed partials

B.2.6 Pippi Ciez featuring Sabrina and Sabrina - Baohum:

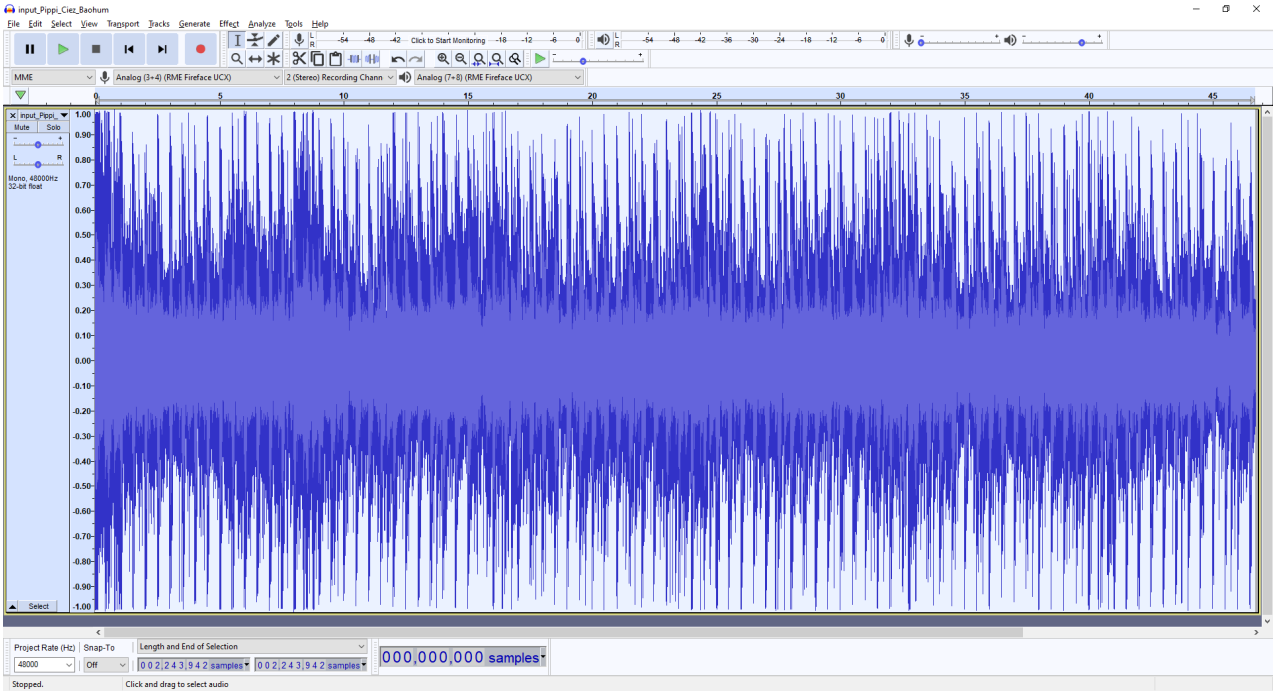
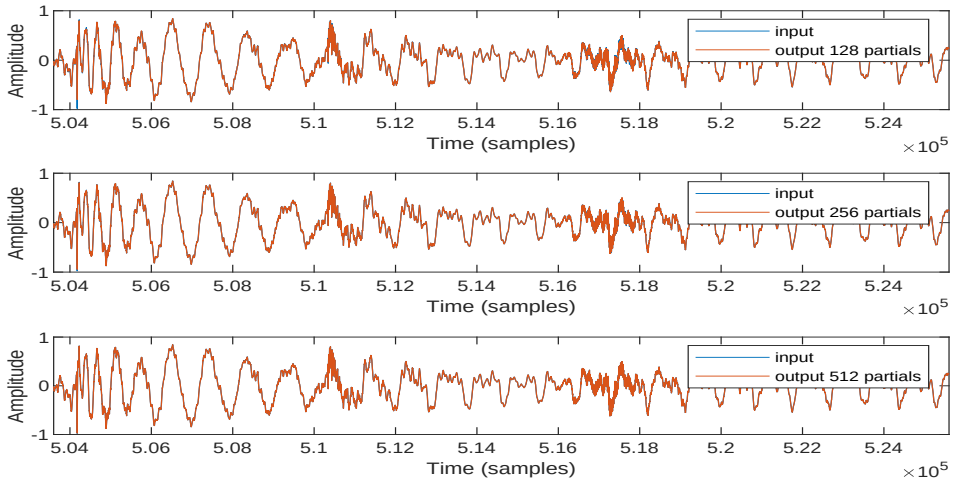
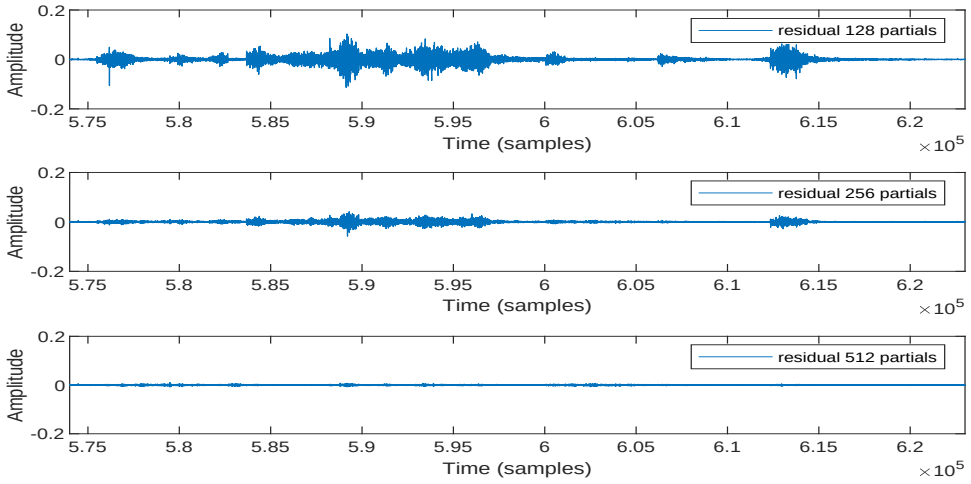


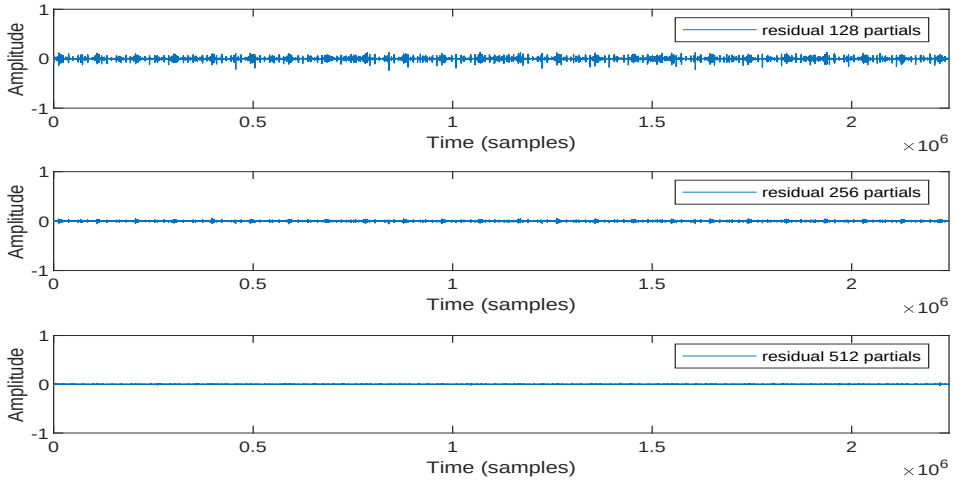
FIGURE B.18: Excerpt from Lumen - Gruntled [272]



(A) Input vs Output



(B) Zoomed in section of Residual Signal



(C) Entire Residual Signal

FIGURE B.19: Output of PippiCiez MoP Modelling using different number of maximum allowed partials

B.2.7 Gary Normal - Faireley Forrest:

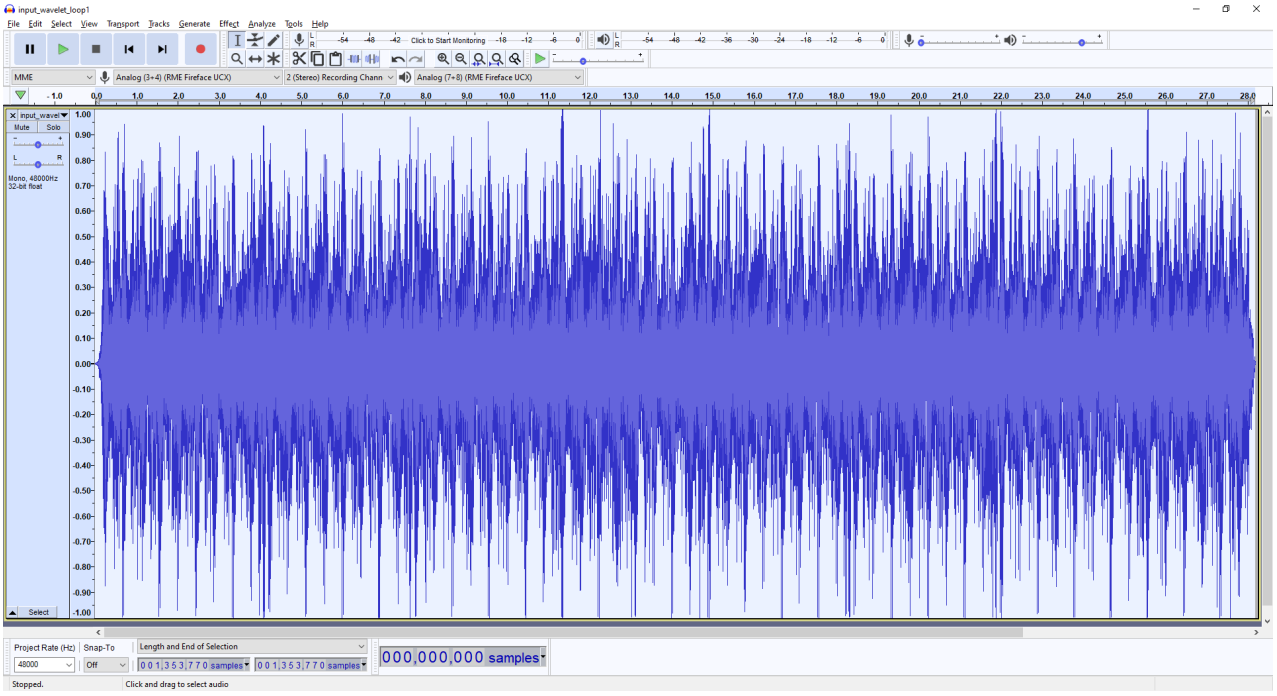
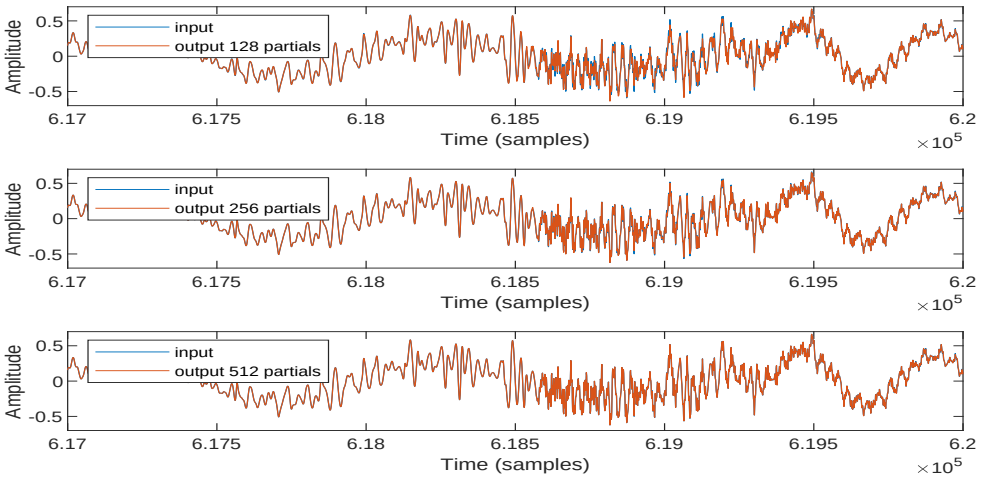
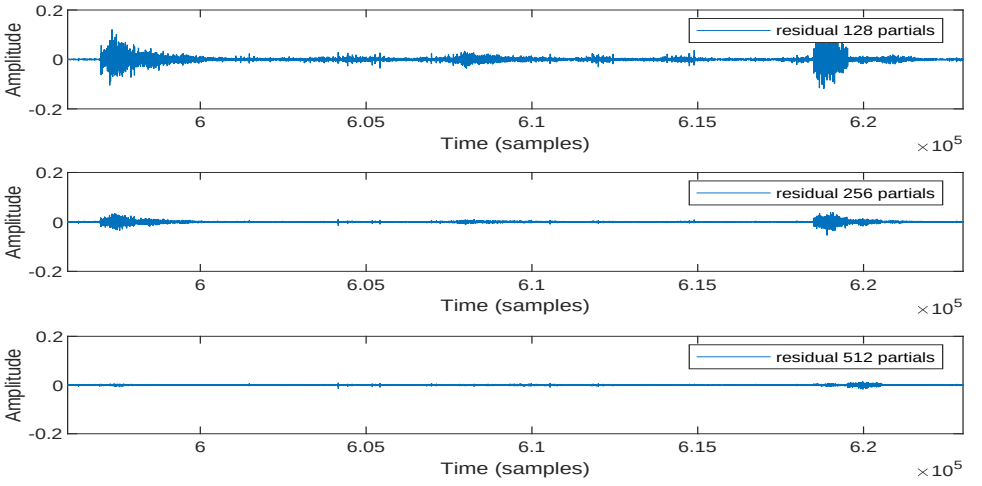


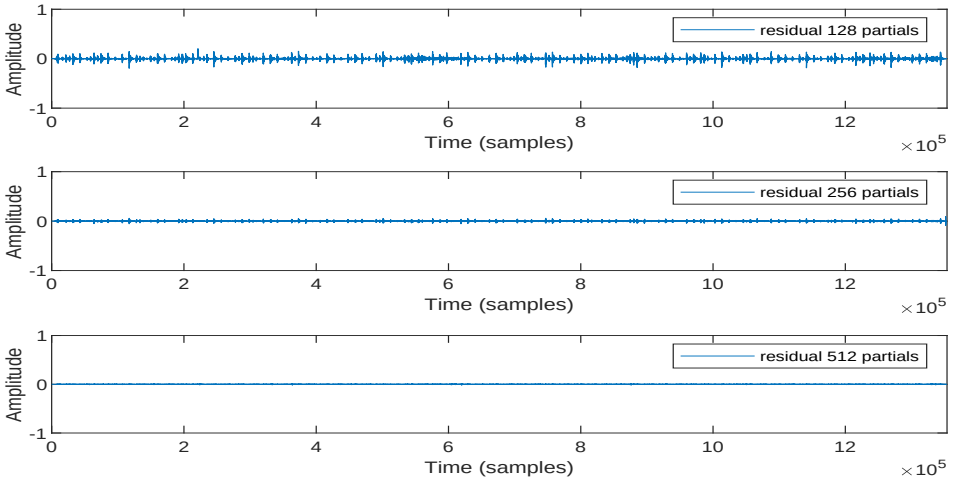
FIGURE B.20: Excerpt from Gary Normal - Faireley Forrest [273]



(A) Input vs Output



(B) Zoomed in section of Residual Signal



(C) Entire Residual Signal

FIGURE B.21: Output of Wavelets MoP Modelling using different number of maximum allowed partials

B.2.8 Petran - AumDelux:

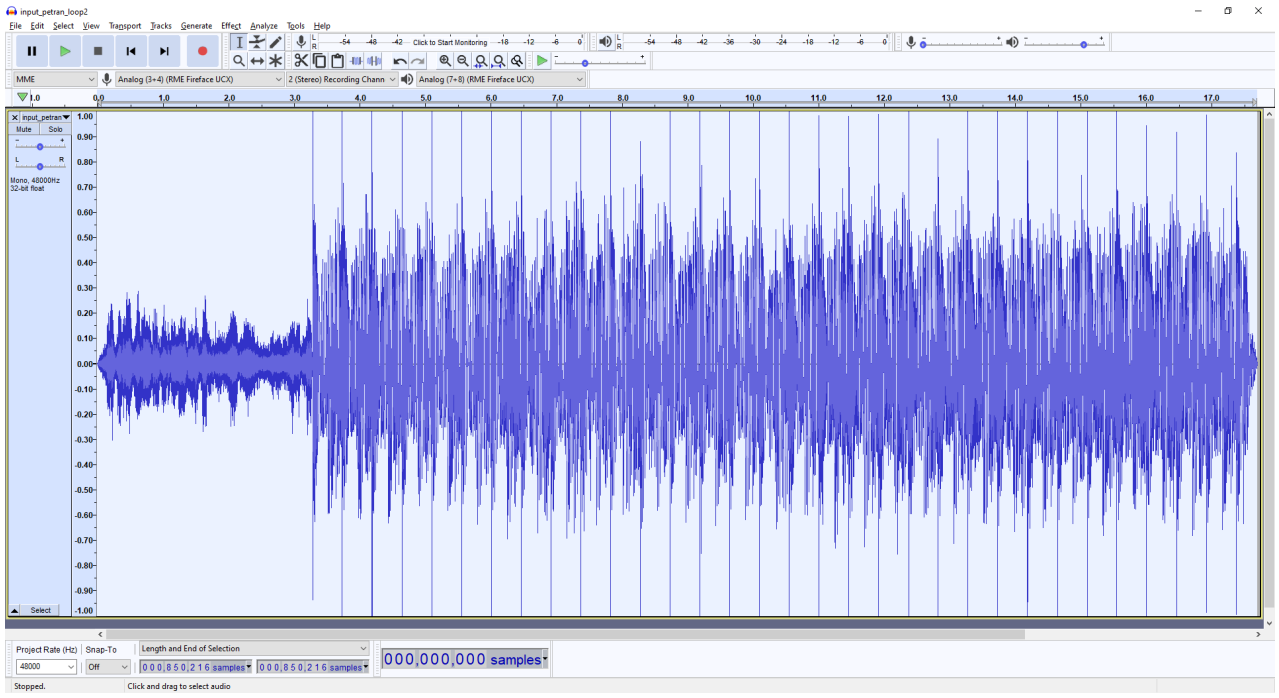
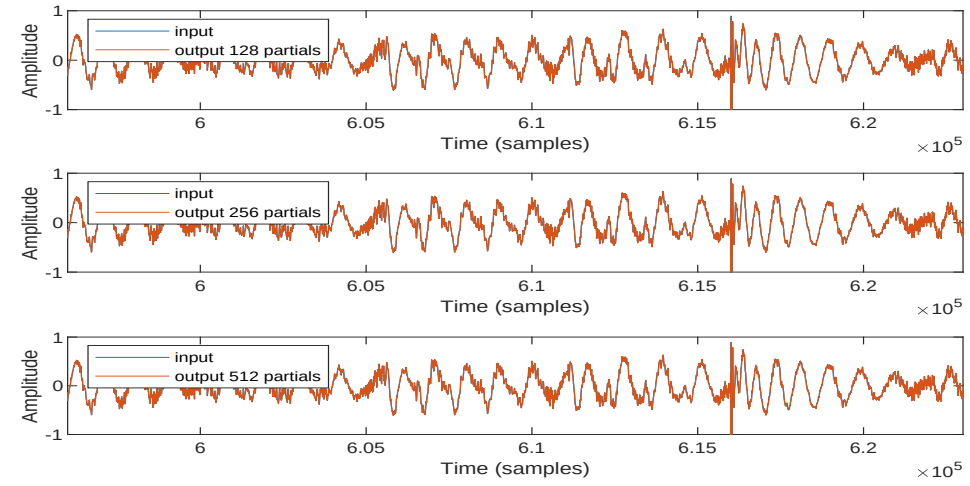
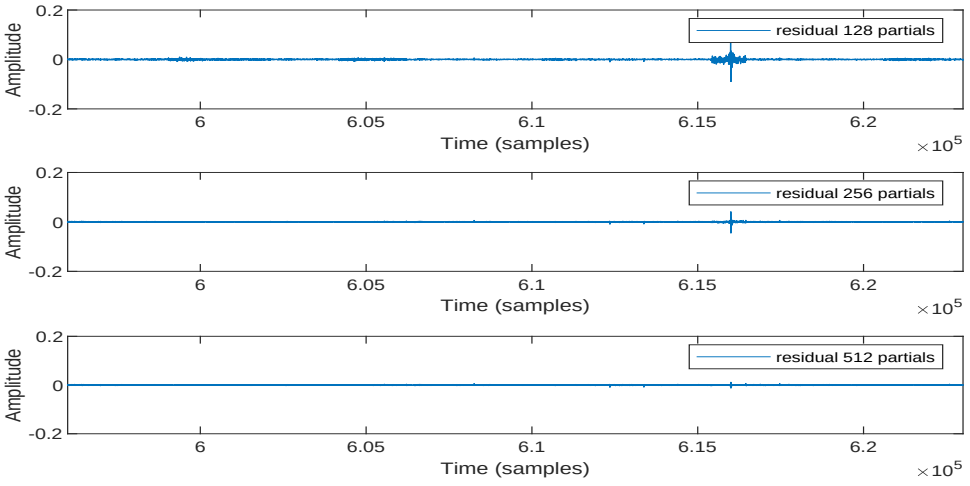


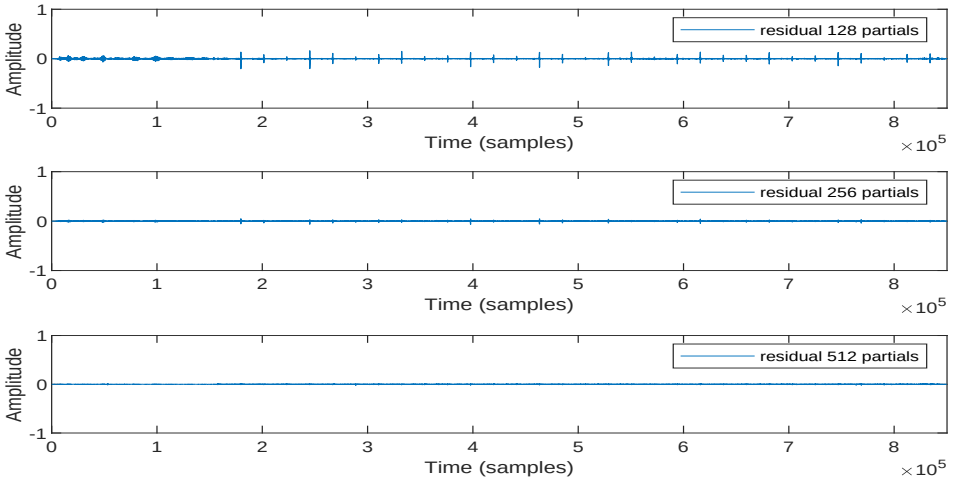
FIGURE B.22: Excerpt from Petran - AumDelux [274]



(A) Input vs Output



(B) Zoomed in section of Residual Signal



(C) Entire Residual Signal

FIGURE B.23: Output of AumDelux MoP Modelling using different number of maximum allowed partials

B.2.9 Elowinz - Granjurema:

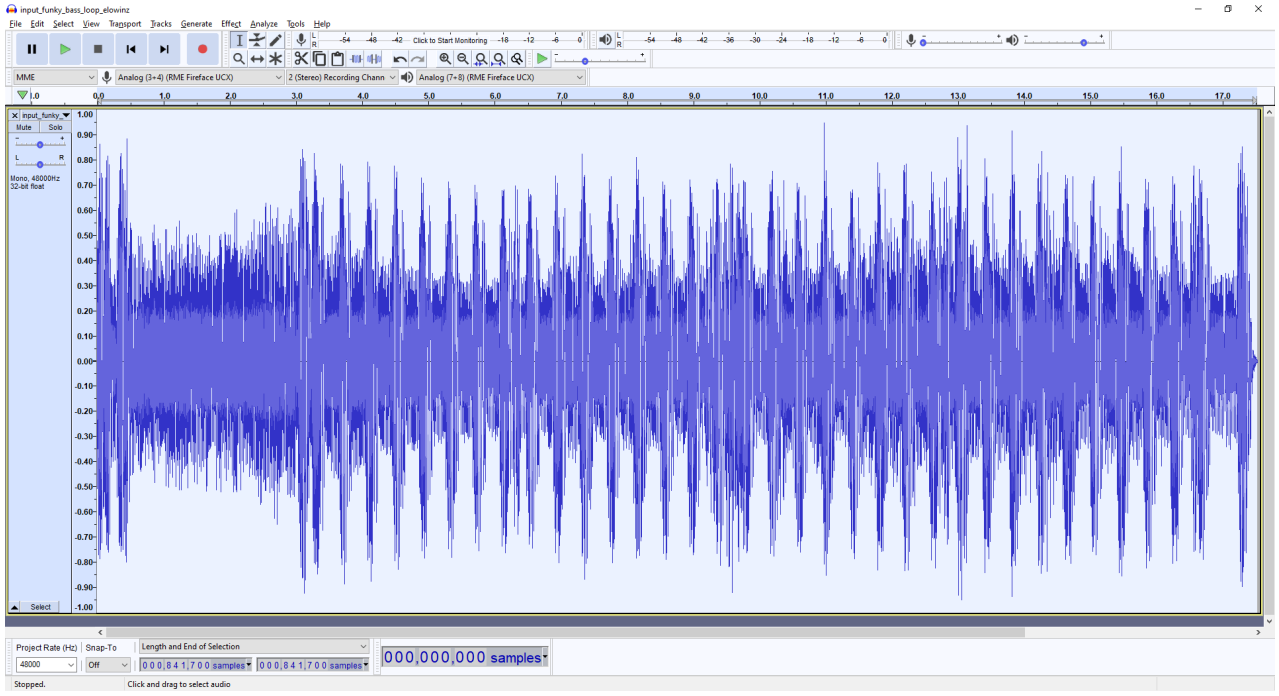
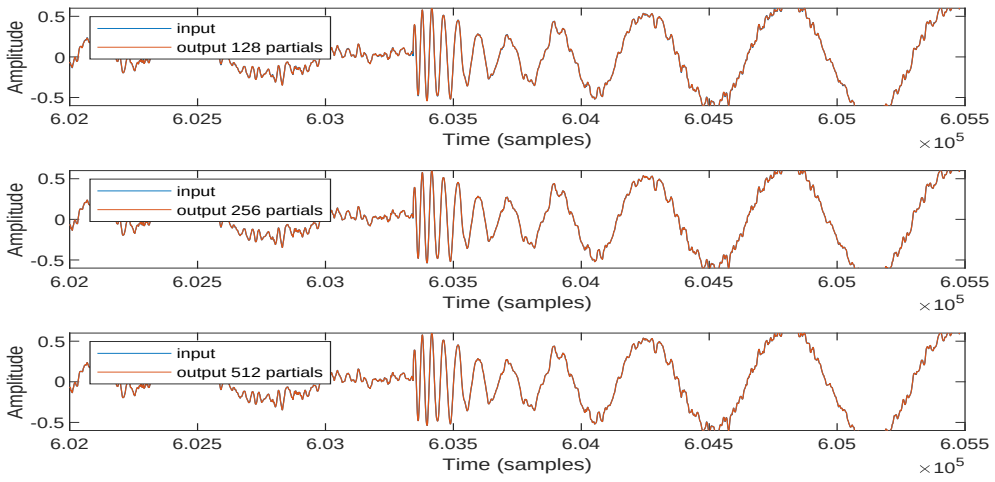
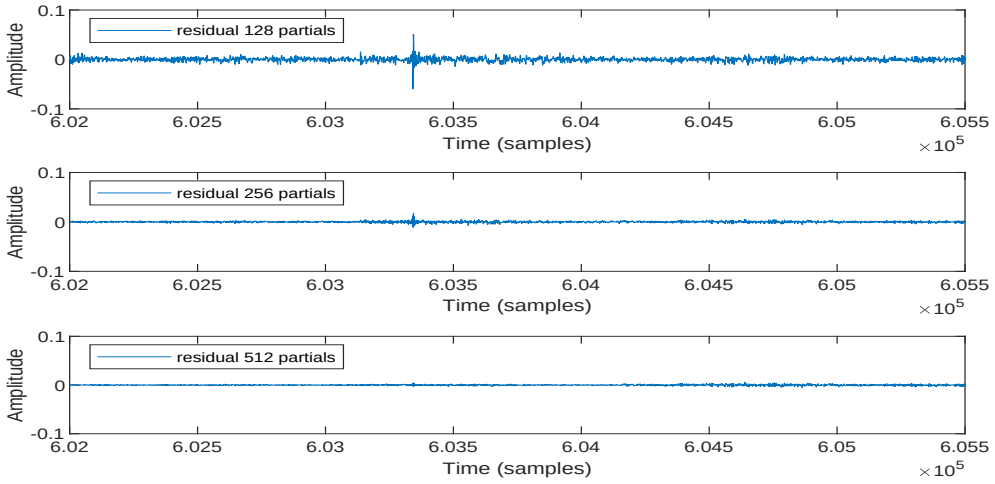


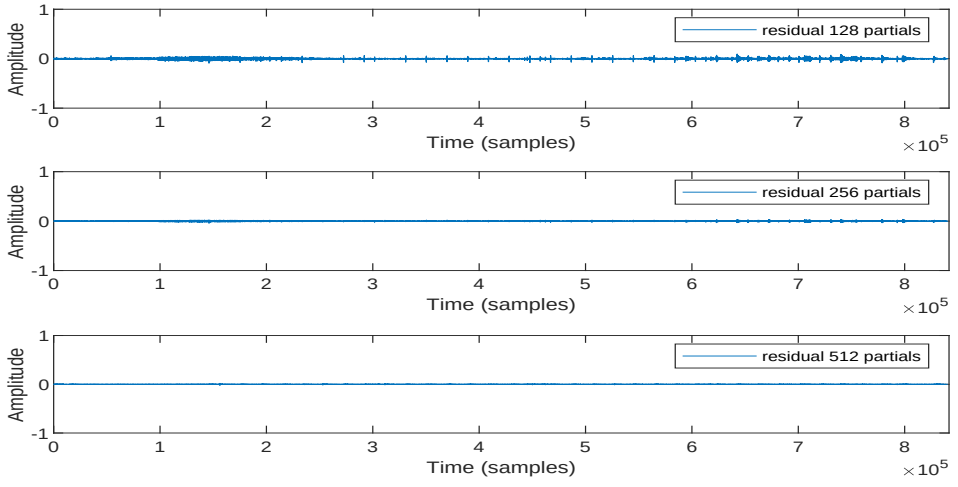
FIGURE B.24: Excerpt from Elowinz - Granjurema [275]



(A) Input vs Output



(B) Zoomed in section of Residual Signal



(C) Entire Residual Signal

FIGURE B.25: Output of Elowinz MoP Modelling using different number of maximum allowed partials

B.2.10 Elowinz - Granjurema:

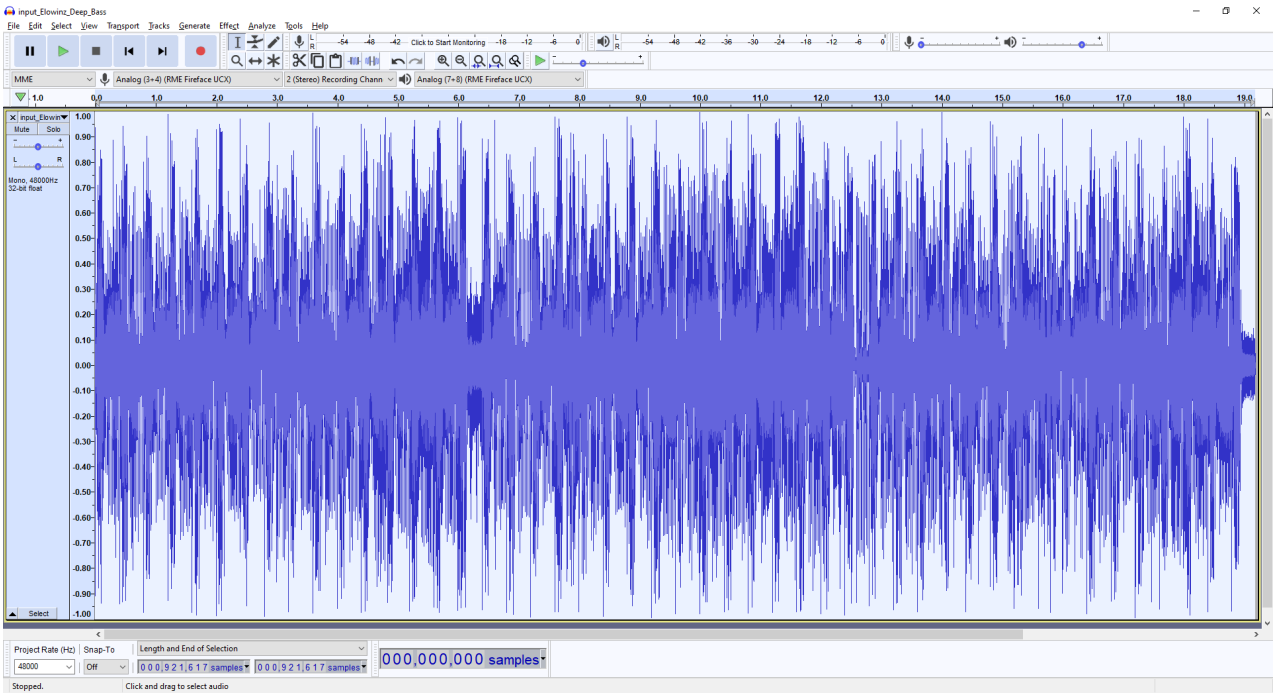
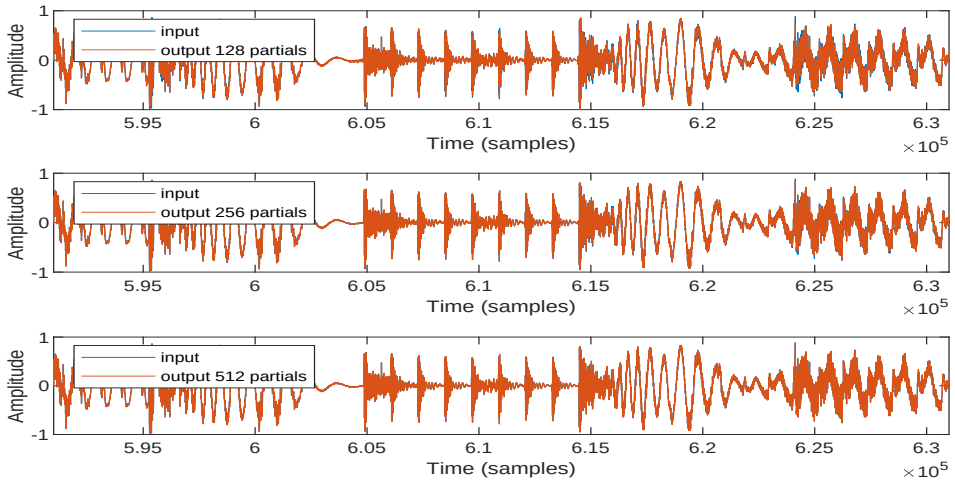
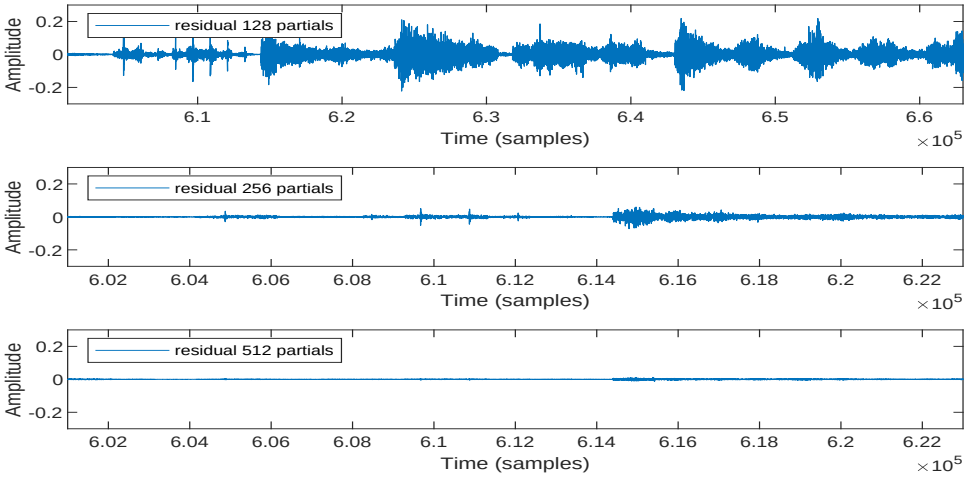


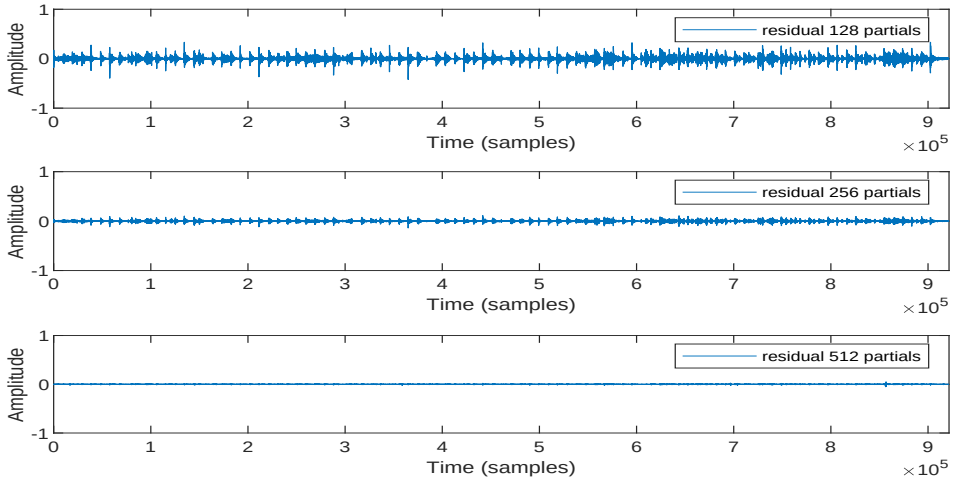
FIGURE B.26: Excerpt from Elowinz - Granjurema [276]



(A) Input vs Output



(B) Zoomed in section of Residual Signal



(C) Entire Residual Signal

FIGURE B.27: Output of Elowinz MoP Modelling using different number of maximum allowed partials

B.3 Residual Output Comparison between MoP and eaQHM

B.3.1 Hermanez - Tale of the Unexpected:

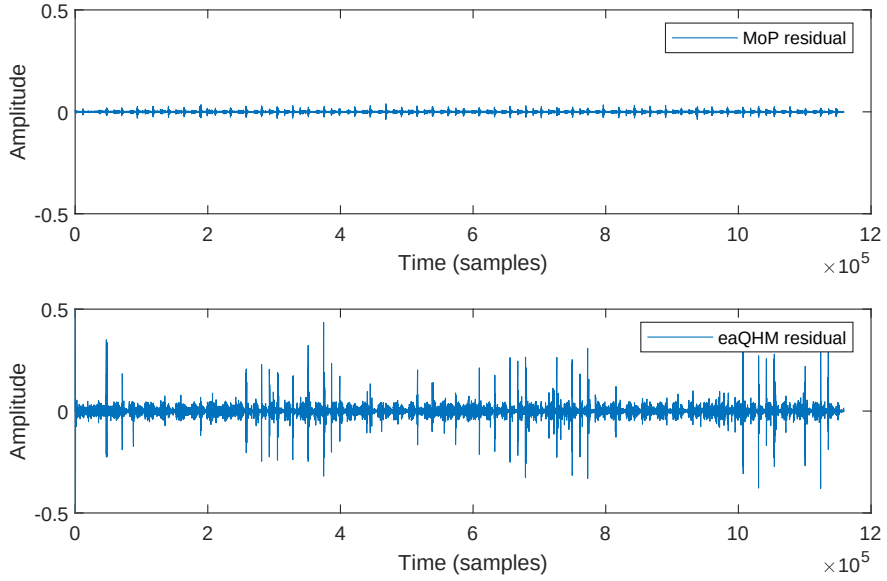


FIGURE B.28: Comparison of MoP (1024 samples, no overlap) and eaQHM (32 samples, 15 sample overlap)

Appendix C

Equations

C.1 Equations:

C.1.1 Reassignment and Generalized Derivative Method Equations:

Reassignment and the Derivatives methods model a signal as a complex exponential with polynomial arguments, given by:

$$s(t) = \exp(\underbrace{(\lambda_0 + \mu_0 t)}_{\lambda(t)=\log(a(t))} + j\underbrace{\left(\phi_0 + \omega_0 t + \frac{\psi_0}{2}t^2\right)}_{\phi(t)}) \tag{C.1}$$

The formula for calculating reassigned frequency $\hat{\omega}$ and amplitude modulation $\hat{\mu}$ are given by:

$$\hat{\omega}(t,\omega) = \frac{\partial}{\partial t}\Im\left(\log\left(S_w(t,\omega)\right)\right) = \omega - \underbrace{\Im\left(\frac{S_{w'}(t,\omega)}{S_w(t,\omega)}\right)}_{-\Delta_\omega} \tag{C.2}$$

$$\hat{\mu}(t,\omega) = \frac{\partial}{\partial t}\Re\left(\log\left(S_w(t,\omega)\right)\right) = -\Re\left(\frac{S_{w'}(t,\omega)}{S_w(t,\omega)}\right) \tag{C.3}$$

The formula for $\hat{t}$ is given by:

$$\hat{t}(t,\omega) = t - \frac{\partial}{\partial\omega}\phi(t,\omega) = t - \Im\left(\frac{\frac{\partial S_w}{\partial\omega}}{S_w}\right) \tag{C.4}$$

The formula for frequency derivative $\hat{\psi}$ is:

$$\hat{\psi} = \frac{\partial\hat{\omega}}{\partial\hat{t}} = \frac{\partial\hat{\omega}}{\partial t}/\frac{\partial\hat{t}}{\partial t} \tag{C.5}$$

Finally the formula for reassigned phase $\hat{\phi}$ and amplitude *hata* are:

$$\hat{\phi}_0 = \angle\left(\frac{S_w\left(\omega_m\right)}{\Gamma_w\left(\Delta_\omega, \hat{\mu}_0, \hat{\psi}_0\right)}\right) \tag{C.6}$$

$$\hat{a}_0 = \left|\frac{S_w}{\Gamma_w\left(\omega_\Delta, \hat{\mu}_0, \hat{\psi}_0\right)}\right| \tag{C.7}$$

where the function Γ_w is given by [277]:

$$\Gamma_w\left(\omega, \mu_0, \psi_0\right) = \int_{-\infty}^{+\infty} w(t)\exp\left(\mu_0 t + j\left(\omega t + \frac{\psi_0}{2}t^2\right)\right)dt \tag{C.8}$$

and where w' denotes the derivative of w.

C.1.2 Phase Distortion Plots for Linear and Exponential Amplitude Change:

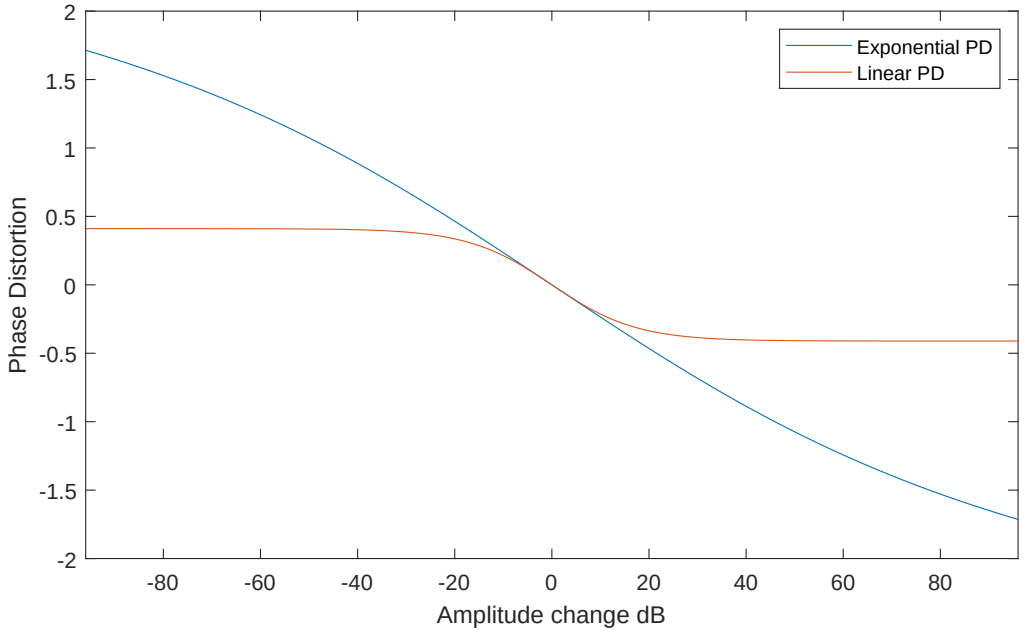


FIGURE C.1: Exponential Analytical and Measured phase difference compared

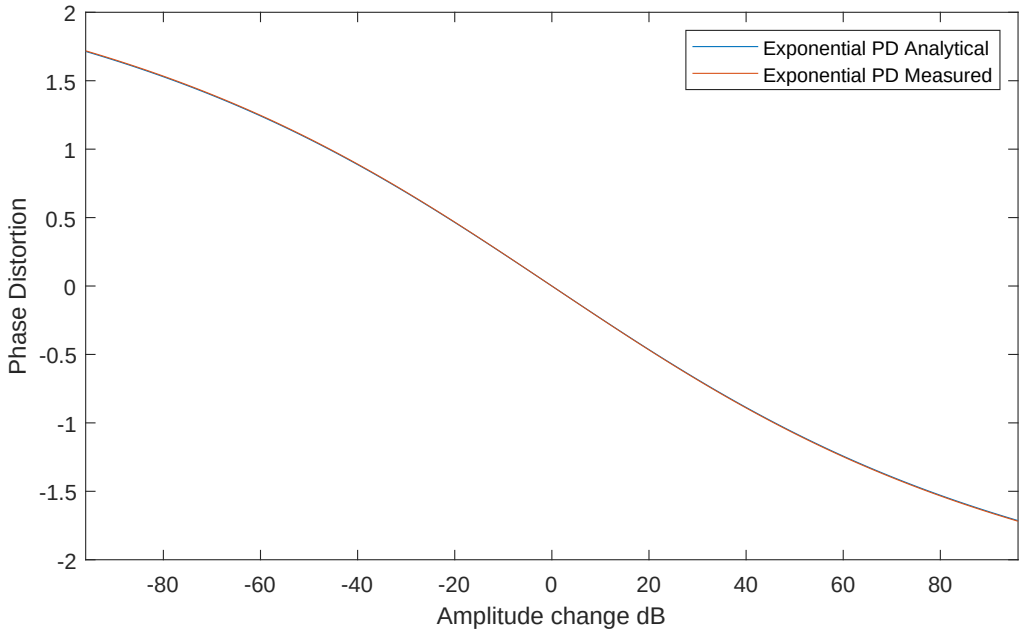


FIGURE C.2: Exponential Analytical and Measured phase difference compared

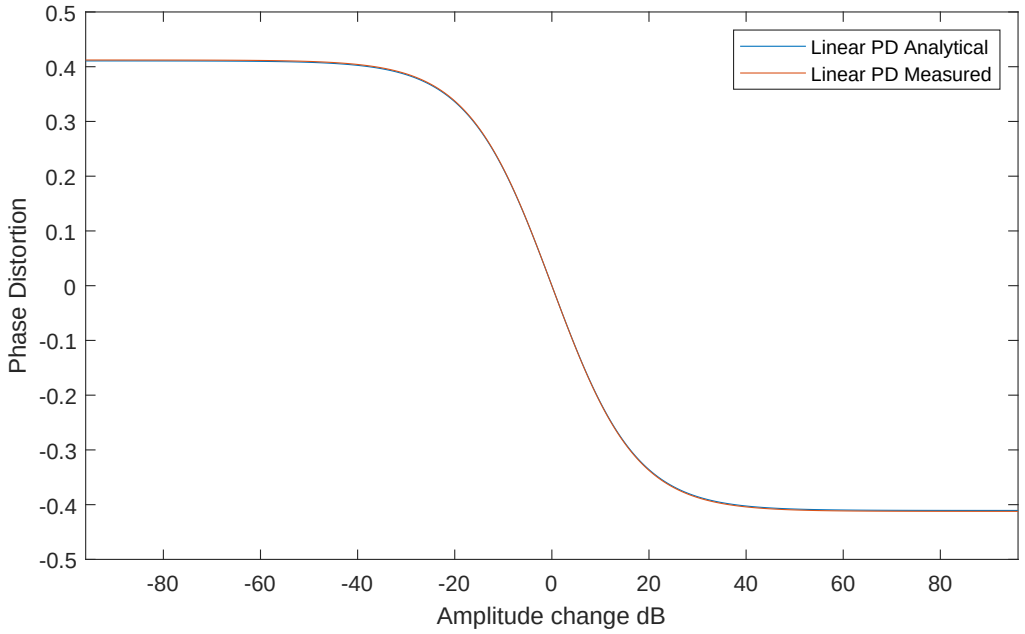


FIGURE C.3: Linear Analytical and Measured phase difference compared

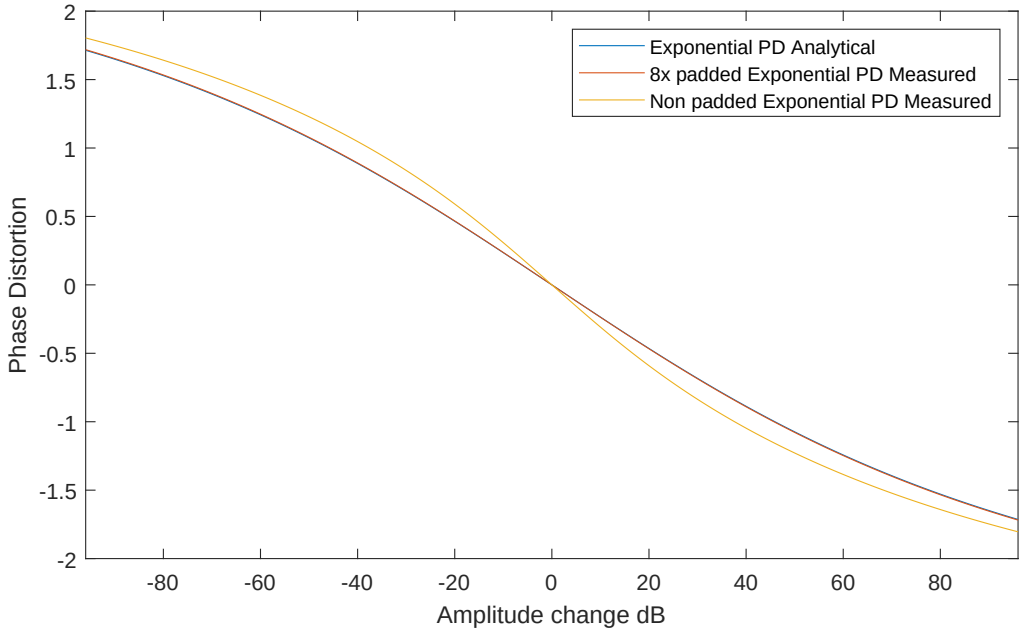


FIGURE C.4: Linear Analytical and Measured phase difference compared

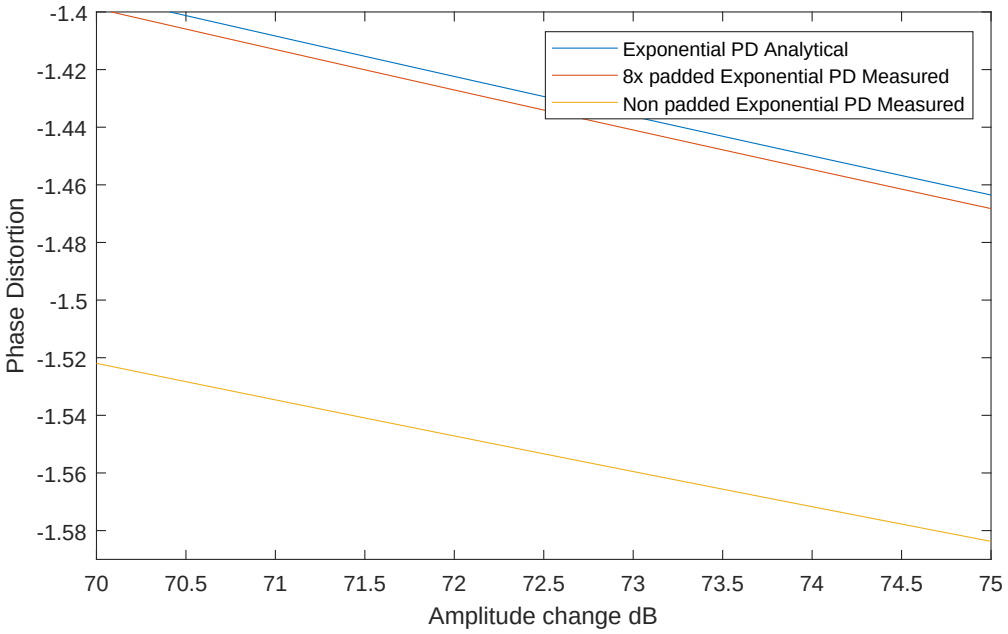


FIGURE C.5: Linear Analytical and Measured phase difference compared

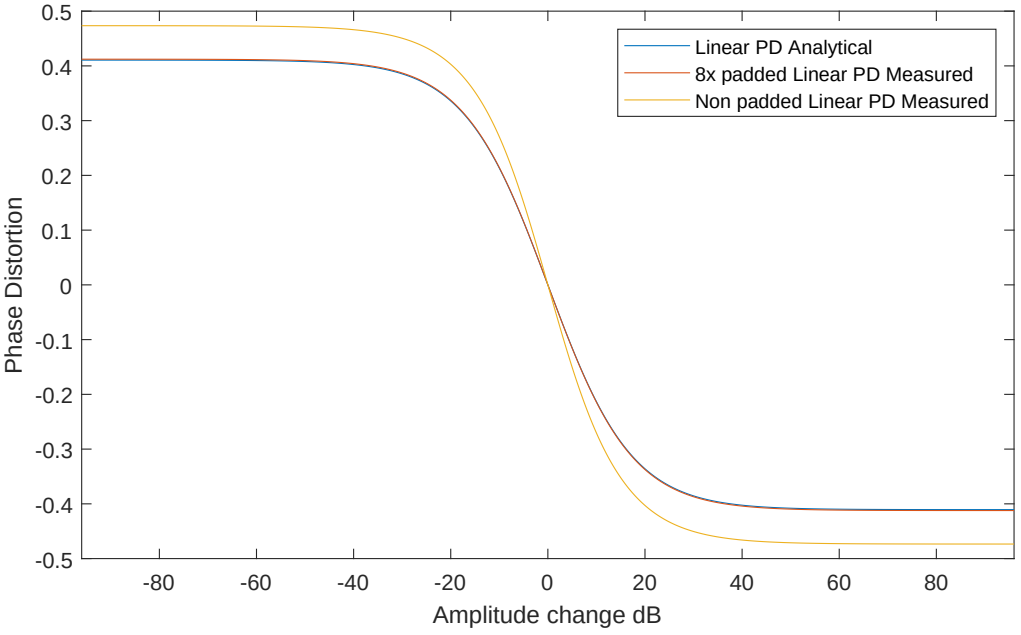


FIGURE C.6: Linear Analytical and Measured phase difference compared

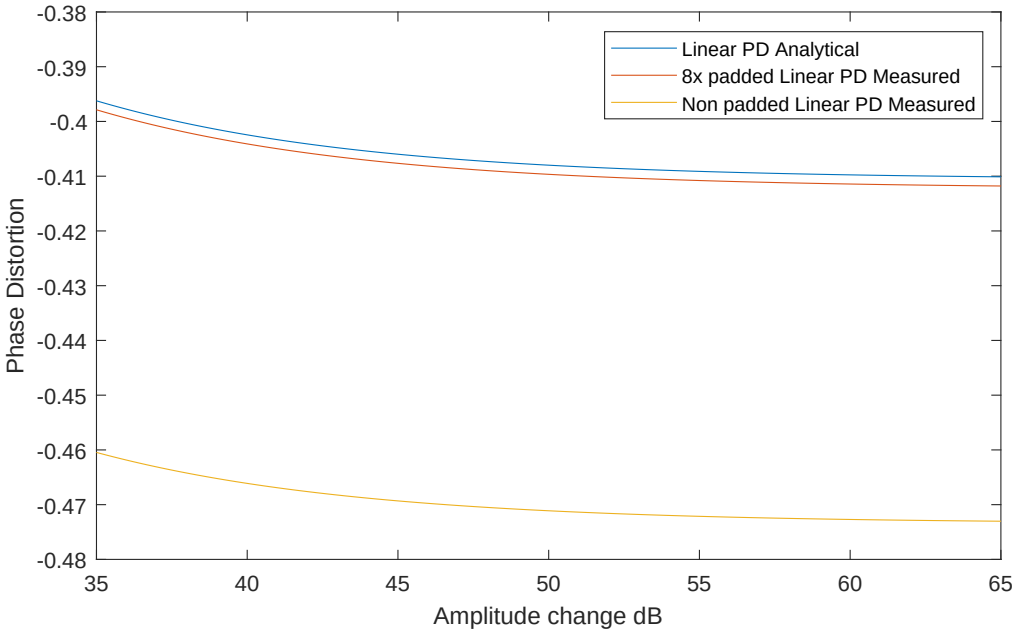


FIGURE C.7: Linear Analytical and Measured phase difference compared

C.1.3 Causal Exponential Amplitude Change Rectangular Window

The equation for deriving causal estimates of the derivative of the phase using a rectangular window were independently derived. However, the original derivation of equation C.1.3 was subsequently found in [278] where correction of the amplitude estimate is given by using information from the Fourier Transform of the window function.

Fourier integral, f is frequency, t is time, a is amount of exponential amplitude change

$$\int \left(e\text{^}((a-(I\times 2\times \pi \times f))\times t)\right), \{t, 0, 1\} \tag{C.9}$$

$$= -\frac{1-e^{a-2if\pi}}{a-2if\pi} \tag{C.10}$$

find derivative at peak, i.e. when f == 0:

$$2\left(-1+\frac{1}{a}+\frac{1}{1-e^{a}}\right)\pi, f=0 \tag{C.11}$$

$$\log\left[\frac{\mathrm{M}}{\mathrm{Abs}\left[\frac{-1c^{a}}{\mathrm{a}}\right]}\right] \tag{C.12}$$

C.1.4 Numerical Representations of Amplitude

The volume of a signal in a floating point system is normalised to range from -1.0 to 1.0.

The signal energy (magnitude) of the Fourier spectrum is usually converted to Decibels (dB) which is a logarithmic expression of ratios. The decibel uses the base 10 and is defined as:

$$\mathrm{dB} = 10log_{10}(\frac{\mathrm{Amplitude})}{\mathrm{Amplitude}_{\mathrm{ref}}}) \tag{C.13}$$

where Amplitude is the measured amplitude and $\mathrm{Amplitude}_{\mathrm{ref}}$ is the reference amplitude.

The Neper, is also a logarithmic ratio of two numbers but uses the natural logarithm and is defined as

$$\mathrm{Np} = \ln(\frac{\mathrm{Amplitude})}{\mathrm{Amplitude}_{\mathrm{ref}}}) \tag{C.14}$$

The Neper is used less frequently in audio representations but is useful for certain calculations because the derivative of the natural logarithm at $x = 1$ is 1.

$$\begin{aligned}\frac{d}{dx}\ln(x) &= \frac{1}{x} \\ &= 1, \quad x = 1\end{aligned} \tag{C.15}$$

C.1.5 Measure Accuracy of Linear and Exponential Discrimination

Signal-to-Noise Ratio (SNR) Equation:

$$\mathrm{SNR} = 10 \log \left(\frac{\sum_{n=1}^{N} f(n)^2}{\sum_{n=1}^{N} [f(n) - \widehat{f}(n)]^2} \right) \tag{C.16}$$

SNR is a measure in dB that compares the level of a signal to the level of background noise.

Signal-to-Residual ration (SRR) Equation:

$$SRR = \frac{\langle s, ws \rangle}{\langle s - \hat{s}, w(s - \hat{s}) \rangle} \tag{C.17}$$

where $s, \hat{s}$ are the original signal (without noise) and the estimated signal respectively, and w the Hann window.

Signal-to-Reconstruction-Error ratio (SRER) Equation:

$$\mathrm{SRER} = 20 \log_{10} \frac{\mathrm{RMS}[s(t)]}{\mathrm{RMS}[\hat{y}(t)]} \tag{C.18}$$

where $s(t)$ is the original signal, $y(t)$ is the re-synthesised signal from model parameters, and $\hat{y}(t)$ is the residual signal obtained by subtracting $y(t)$ from $s(t)$. SRER is the ratio in dB of the energy in the original signal $s(t)$ and the residual $\hat{y}(t)$.

C.2 Models used for Synthetic Nonstationary Sinusoids

In [142] the short-term signal model $s(t)$ is described for the Exponentially Damped Sinusoidal Model (EDSM), Reassigned Sinusoidal Model (RSM) and The extended adaptive Quasi-Harmonic Model (eaQHM) as follows:

C.2.1 Exponentially Damped Sinusoidal Model (EDSM)

EDSM assumes that $x(t)$ can be approximated by the underlying signal model

$$x(t) = \exp(\lambda + \mu t)\cos(\omega t + \theta) \tag{C.19}$$

where $A(t) = \exp(\lambda + \mu t)$ is the temporal envelope and $\Phi(t) = \omega t + \theta$ is the time-varying phase. The short-term frame $x(t)$ in EDSM is simply modeled as a stationary sinusoid with constant frequency ω modulated in amplitude by an exponential envelope controlled by λ and μ. $A(t)$ grows exponentially when $\mu > 0$, decays when $\mu < 0$, and is constant if $\mu = 0$. The literature has shown [171, 172, 279, 280] that subspace methods render accurate parameter estimation for EDSM. This work uses ESPRIT to fit the parameters of EDSM [280].

C.2.2 Reassigned Sinusoidal Model (RSM)

$$x(t) = \exp(\lambda + \mu t)\cos\left(\psi t^2 + \omega t + \theta\right) \tag{C.20}$$

where $A(t) = \exp(\lambda + \mu t)$ is the temporal envelope and $\Phi(t) = \psi t^2 + \omega t + \theta$ is the time-varying phase. The short-term frame $x(t)$ in RSM is approximated as a sinusoid with quadratic phase (quadratic frequency ψ, linear frequency ω, and phase shift θ) modulated in amplitude by an exponential envelope controlled by λ and μ similarly to EDSM.

The parameters of the model are estimated using the time-frequency reassignment method [153, 159, 281, 282].

C.2.3 The extended adaptive Quasi-Harmonic Model (eaQHM)

The assumption behind eaQHM is that speech and musical sounds can be approximated by a sum of M quasi-harmonic, highly nonstationary, AM-FM modulated partials $s_m(t)$. Each partial is further modeled inside the analysis frame as a short-term $x(t)$ which can be approximated by the underlying signal model

$$x(t) = (\lambda + \mu t)\cos\left(\psi t^2 + \omega t + \theta\right) \tag{C.21}$$

where $A(t) = (\lambda + \mu t)$ is the temporal envelope and $\Phi(t) = \psi t^2 + \omega t + \theta$ is the time-varying phase. The short-term frame $x(t)$ in eaQHM is implicitly modeled as a sinusoid with quadratic phase (quadratic frequency ψ, linear frequency ω, and phase shift θ) modulated in amplitude by a *linear* envelope controlled by λ and μ. $A(t)$ grows linearly when $\mu > 0$, decays when $\mu < 0$, and is constant when $\mu = 0$ [168].

In eaQHM the parameters λ, μ, ψ, ω and π are adapted from successive steps of parameter re-estimation using least squares by iterating over the entire audio file from beginning to end numerous times.

"To estimate the AM-FM components, we use QHM's parameters $[ak, bk]$ and a set of f_ks.

The AM and FM components do not have a closed form description, like $A \times \cos(2 \times \pi_k \times +\phi)$.

It's more like $A(n) \times \cos(\theta_k(n))$, where $A(n)$ and $\theta_k(n)$ do not have a closed form expression. In a way, this is a non-parametric representation." [283]

Appendix D

Mathematical Proofs

D.1 Mathematica Proofs

D.1.1 Non-Causal Linear Derivative of the Phase

$$\text{Integrate}\left[(1/L)\times((\text{Cos}[2\times\text{Pi}\times\ \text{L/L}]+1)/2)\times\left(\text{e}\text{\^{}}(-(\text{I}\times 2\times\text{Pi}\times\text{f})\times\text{t})\right)\times\right.$$
$$\left.\left(\left(\text{e}\text{\^{}}(-\text{a}/2)\right)+\left(\text{t}\times\left(\left(\text{e}\text{\^{}}(\text{a}/2)\right)-\left(\text{e}\text{\^{}}(-\text{a}/2)\right)\right)\right)+\left(\left(\left(\text{e}\text{\^{}}(\text{a}/2)\right)-\left(\text{e}\text{\^{}}(-\text{a}/2)\right)\right)/2\right)\right),\quad\{\text{t},(-\text{L}/2),(\text{L}/2)\}\right] \tag{D.1}$$

Setting $f=0$ results in an infinite expression where $1/0$ is encountered.

Limiting $f->0$ however results in:

$$\frac{d(\arg(W))}{df}|\lim_{f\to 0}=\frac{(-1+e^{\alpha})(-6+\pi^2)}{3(1+e^{\alpha})\pi} \tag{D.2}$$

D.1.2 Non-Causal Exponential Derivative of the Phase

From [10] with supporting materials available online: [284]

Integrate[$(1/L) \times ((\text{Cos}[2 \times \text{Pi} \times /L] + 1)/2) \times (E\text{^}((a - (I \times 2 \times \text{Pi})) \times (t))), \{t, (-L/2), (L/2)\}$]

$$= \frac{4\pi^2 \,\text{Sinh}\left[\frac{1}{2}L(a - 2if\pi)\right]}{L(a - 2if\pi)\left(a^2L^2 - 4iafL^2\pi - 4\left(-1 + f^2L^2\right)\pi^2\right)} \tag{D.3}$$

$$\text{Manipulate}\left[\frac{4\pi^2 \,\text{Sinh}\left[\frac{1}{2}L(a - 2if\pi)\right]}{L(a - 2if\pi)\left(a^2L^2 - 4iafL^2\pi - 4\left(-1 + f^2L^2\right)\pi^2\right)}, \{L, \{1\}\}\right] \tag{D.4}$$

$$\text{ComplexExpand}\left[\frac{4\pi^2 \,\text{Sinh}\left[\frac{1}{2}(a - 2if\pi)\right]}{(a - 2if\pi)\left(a^2 - 4iaf\pi - 4\left(-1 + f^2\right)\pi^2\right)}\right] \tag{D.5}$$

$$= \pi\left(\frac{2}{a} + \frac{4a}{a^2 + 4\pi^2} - \text{Coth}\left[\frac{a}{2}\right]\right) \tag{D.6}$$

Appendix E

Accompanying Material

E.1 List of accompanying material

Attached with submission of the thesis is a `Murray_107037547_AccompanyingMaterial.zip`

In the .zip file are 3 Main Folders

1. eaQHM-analysis-and-synthesis-in-Python: Contains the eaQHM Python code
2. Mathematica: Contains the Mathematica Proofs
3. Matlab: Contains the Matlab code and Audio Samples

The Matlab root folder contains most of the scripts used, wit the exception of the `da_and_df_tests` and `Env_Type_Discrimination_Test_Results` Subfolders which contain some separate test scripts.

Within the Matlab Folder are a number of Subfolders:

1. Audio: contains audio test files and outputs
2. `da_and_df_tests`: Contains a script testing both dA and dF
3. `DesamToolbox_v1.1`, eaQHM, EDS, MoP: Contain scripts used to test synthetic nonstationary signals.
4. `Env_Type_Discrimination_Test_Results`: Contains a script to test envelope type discrimination and estimation of non-causal dA for linear and exponential amplitude change.
5. segSWT: Contains scripts testing forward segSWT and inverse segSWT.
6. MoP contains causal MoP test script using rectangular window. Some additional scripts can be found here for testing time and pitch scale modifications.
7. Ch6 contains the segmented and non-segmented scripts used to test the system implementation in Chapter 6. This also includes the non-segmented system implementation used in Chapters 5 and 6.

Bibliography

[1] T. Mckenna, "The archaic revival: Speculations on psychedelic mushrooms, the amazon, virtual reality, ufos, evolut," 1992. [Online]. Available: http://www.youtube.com/watch?v=uJjR3aUhsOk

[2] M. Winkelman, *Biogenetic structural perspectives on shamanism and raves: The origins of collective ritual dance*, 01 2015, pp. 1–36.

[3] P. Kirn, "Keyboard presents the evolution of electronic dance music," 2011.

[4] N. Om, "Kick & bass mixing in electronic music," https://www.udemy.com/course/kick-bass-complete/, 2017.

[5] J. Phillips-Silver, C. Aktipis, and G. Bryant, "The ecology of entrainment: Foundations of coordinated rhythmic movement," *Music perception*, vol. 28, pp. 3–14, 09 2010.

[6] T. Lenc, P. Keller, M. Varlet, and S. Nozaradan, "Neural tracking of the musical beat is enhanced by low-frequency sounds," *Proceedings of the National Academy of Sciences*, vol. 115, p. 201801421, 07 2018.

[7] G. Jackson, "Modern approaches: Processing bass," https://daily.redbullmusicacademy.com/2016/02/modern-approaches-processing-bass/, 2016.

[8] S. Academy, "Kick2 vst." [Online]. Available: https://www.sonicacademy.com/products/kick-2

[9] J. Wells, "Real-time spectral modelling of audio for creative sound transformation," Ph.D. dissertation, University of York, 2006. [Online]. Available: http://etheses.whiterose.ac.uk/14084/

[10] ——, "Methods for separation of amplitude and frequency modulation in fourier transformed signals," in *Proceedings of the 13th International Conference on Digital Audio Effects (DAFx10)*, 2010.

[11] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[12] D. Zantalis and J. Wells, "Semi-blind audio source separation of linearly mixed two-channel recordings via guided matching pursuit," in *Proceedings of the 17th International Conference on Digital Audio Effects (DAFx-14)*, 2014.

[13] D. Zantalis, "Guided matching pursuit and its application to sound source separation," Ph.D. dissertation, Department of Electronic Engineering, University of York, 2016. [Online]. Available: http://etheses.whiterose.ac.uk/13204/

[14] J. Wells, "Modal decompositions of impulse responses for parametric interaction," *journal of the audio engineering society*, vol. 69, no. 7/8, july 2021.

[15] l. Savioja, V. Välimäki, and J. Smith, "Real-time additive synthesis with one million sinusoids using a gpu," *journal of the audio engineering society*, 2010.

[16] "Matlab," https://uk.mathworks.com/.

[17] "Nondecimated discrete stationary wavelet transforms (swts)," https://uk.mathworks.com/help/wavelet/ug/discrete-stationary-wavelet-transform-swt.html, 2023.

[18] J.-C. Pesquet, H. Krim, and H. Carfantan, "Time-invariant orthonormal wavelet representations," *IEEE Transactions on Signal Processing*, vol. 44, no. 8, pp. 1964–1970, 1996.

[19] R. R. Coifman and D. L. Donoho, *Translation-Invariant De-Noising*. New York, NY: Springer New York, 1995, pp. 125–150. [Online]. Available: https://doi.org/10.1007/978-1-4612-2544-7_9

[20] G. P. Nason and B. W. Silverman, *The Stationary Wavelet Transform and some Statistical Applications*. New York, NY: Springer New York, 1995, pp. 281–299. [Online]. Available: https://doi.org/10.1007/978-1-4612-2544-7_17

[21] T. Stanley, *The History of Philosophy*, 1655.

[22] J. James, *The Music of the Spheres: Music, Science, and the Natural Order of the Universe*, ser. Copernicus Series. Springer, 1995. [Online]. Available: https://books.google.co.za/books?id=sVDqE3Qsd20C

[23] T. Davis, "The harmony of the spheres: what modern physics can tell us," 2018. [Online]. Available: https://www.abc.net.au/classic/features/music-of-the-spheres-what-modern-physics-can-tell-us/10124000

[24] S. Helmreich, "Gravity's reverb: Listening to space-time, or articulating the sounds of gravitational-wave detection," *Cultural Anthropology*, vol. 31, 2016.

[25] T. Rossing, P. Wheeler, and F. Moore, *The Science of Sound*, ser. Addison-Wesley series in physics. Addison Wesley, 2002. [Online]. Available: https://books.google.co.za/books?id=kLDvAAAAMAAJ

[26] J. O. Smith, *Mathematics of the Discrete Fourier Transform (DFT)*. http://www.w3k.org/books/: W3K Publishing, 2007.

[27] Y. Landman, "The nodes of a vibrating string are harmonics," https://en.wikipedia.org/wiki/Harmonic#/media/File:Moodswingerscale.svg, 2010.

[28] J. Smith, "Physical modeling using digital waveguides," *Computer Music Journal*, vol. 16, p. 74, 1992.

[29] ——, "Efficient synthesis of stringed musical instruments," 1993.

[30] K. Karplus and A. Strong, "Digital synthesis of plucked-string and drum timbres," *Computer Music Journal*, vol. 7, no. 2, 1983. [Online]. Available: http://www.jstor.org/stable/3680062

[31] D. A. Jaffe and J. O. Smith, "Extensions of the karplus-strong plucked-string algorithm," *Computer Music Journal*, vol. 7, p. 56, 1983.

[32] S. Bilbao, "Energy-conserving finite difference schemes for tension-modulated strings," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2004.

[33] ——, "Fast modal synthesis by digital waveguide extraction," *IEEE Signal Processing Letters*, vol. 13, no. 1, pp. 1–4, 2006.

[34] W. Sabine, *Collected Papers on Acoustics*. Peninsula Publishing, 1993. [Online]. Available: https://books.google.co.za/books?id=3B8_PQAACAAJ

[35] J. Balint, "Evaluating the decay of sound," Ph.D. dissertation, Graz University of Technology, 2020.

[36] J. Moorer, *About This Reverberation Business*, 1985, vol. 3.

[37] G-Sonique, "Ultrabass vst." [Online]. Available: https://www.g-sonique.com/ultrabassmx.html

[38] E. Krantzberg, "Eliminate competition between the kick and bass." [Online]. Available: https://www.sonarworks.com/soundid-reference/blog/learn/eliminate-competition-between-the-kick-and-bass/

[39] musicianonamission, "Adsr envelopes." [Online]. Available: https://www.musicianonamission.com/adsr/

[40] Bitwig, "Bitwig studio." [Online]. Available: https://www.bitwig.com/

[41] masteringthemix, "How to get the right amount of punch in your master." [Online]. Available: https://www.masteringthemix.com/blogs/learn/how-to-get-the-right-amount-of-punch-in-your-master

[42] K. Hofbauer, "Estimating frequency and amplitude of sinusoids in harmonic signals," Ph.D. dissertation, Graz University of Technology, 2004.

[43] J. Murray and E. Miranda, "Real-time granular synthesis with spiking neurons," *Advances in Networks, Computing and Communications 4*, p. 278, 2006.

[44] K. McCracken, J. Matthias, and E. Miranda, "Neurogranular synthesis: Granular synthesis controlled by a pulse-coupled network of spiking neurons," 04 2011, pp. 354–363.

[45] J. Smith, *Physical audio signal processing*, 01 2004.

[46] S. Bilbao, A. Torin, P. Graham, J. Perry, and G. Delap, "Modular physical modeling synthesis environments on gpu," 09 2014.

[47] S. Bilbao and C. J. Webb, "Physical modeling of timpani drums in 3d on gpgpus," *Journal of The Audio Engineering Society*, vol. 61, pp. 737–748, 2013.

[48] A. Oppenheim, R. Schafer, J. Buck, and L. Lee, *Discrete-time Signal Processing*, ser. Prentice Hall international editions. Prentice Hall, 1999. [Online]. Available: https://books.google.co.za/books?id=Bv1SAAAAMAAJ

[49] J. Reiss and A. McPherson, *Audio Effects: Theory, Implementation and Application.* CRC Press, 2014. [Online]. Available: https://books.google.co.za/books?id=mlHSBQAAQBAJ

[50] U. Zölzer, X. Amatriain, D. Arfib, J. Bonada, G. De Poli, P. Dutilleux, G. Evangelista, F. Keiler, A. Loscos, D. Rocchesso *et al.*, *DAFX - Digital Audio Effects.* John Wiley & Sons, 2002. [Online]. Available: https://books.google.co.za/books?id=h90HIV0uwVsC

[51] J. Smith, *Spectral Audio Signal Processing.* W3K Publishing, 2007. [Online]. Available: https://books.google.co.za/books?id=qQa8swEACAAJ

[52] J. Eggerrmont, "1Introduction," in *Auditory Temporal Processing and its Disorders.* Oxford University Press, 2015.

[53] A. Oxenham, "Questions and controversies surrounding the perception and neural coding of pitch," *Frontiers in Neuroscience*, vol. 16, p. 1074752, 01 2023.

[54] M. Rutherford, H. von Gersdorff, and J. Goutman, "Encoding sound in the cochlea: from receptor potential to afferent discharge," *The Journal of Physiology*, vol. 599, 03 2021.

[55] S. Shamma and K. Dutta, "Spectro-temporal templates unify the pitch of resolved and unresolved harmonics," *The Journal of the Acoustical Society of America*, vol. 145, pp. 1783–1783, 03 2019.

[56] P. Joris, C. Schreiner, and A. Rees, "Neural processing of amplitude-modulated sounds," *Physiological reviews*, vol. 84, pp. 541–77, 05 2004.

[57] M. Khalil, *The Evolution of Auditory Perception.* Cham: Springer International Publishing, 2018, pp. 1–5. [Online]. Available: https://doi.org/10.1007/978-3-319-16999-6_982-1

[58] J. Schnupp, I. Nelken, and A. J. King, *Auditory Neuroscience: Making Sense of Sound.* The MIT Press, 11 2010. [Online]. Available: https://doi.org/10.7551/mitpress/7942.001.0001

[59] R. Bhatia, *A History of Fourier Series.* Mathematical Association of America, 2005.

[60] A. V. Oppenheim and A. S. Willsky, *Signals and Systems.* Prentice Hall, 1997.

[61] D. Gabor, *Theory of Communication.* Institution of Electrical Engineering, 1946. [Online]. Available: https://books.google.co.za/books?id=kvJSAAAAMAAJ

[62] B. A. Schreiber, "The uncertainty principle," 2023.

[63] "Constant overlap-add (cola) constraint," https://uk.mathworks.com/help/signal/ref/iscola.html.

[64] S. Roucos and A. Wilgus, "High quality time-scale modification for speech," in *ICASSP '85. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 10, 1985, pp. 493–496.

[65] T. Royer, "Pitch-shifting algorithm design and applications in music," Master's thesis, 2019.

[66] J. Bonada, "Wide-band harmonic sinusoidal modeling," 2008. [Online]. Available: https://api.semanticscholar.org/CorpusID:17711131

[67] W. Verhelst and M. Roelands, "An overlap-add technique based on waveform similarity (wsola) for high quality time-scale modification of speech," in *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, 1993, pp. 554–557 vol.2.

[68] S. Toma, G. Târşa, E. Oancea, D. Munteanu, F. Totir, and L. Anton, "A td-psola based method for speech synthesis and compression," in *2010 8th International Conference on Communications*, 2010, pp. 123–126.

[69] P. Cook, "Real sound synthesis for interactive applications," 01 2002.

[70] J. Moorer, "Signal processing aspects of computer music: A survey," *Proceedings of the IEEE*, vol. 65, no. 8, pp. 1108–1137, 1977.

[71] X. Serra and J. Smith, "Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition," *Computer Music Journal*, vol. 14, no. 4, 1990. [Online]. Available: http://www.jstor.org/stable/3680788

[72] J. L. Flanagan and R. M. Golden, "Phase vocoder," *The Bell System Technical Journal*, vol. 45, no. 9, 1966.

[73] M. Portnoff, "Implementation of the digital phase vocoder using the fast fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 3, pp. 243–248, 1976.

[74] J. Moorer, "The use of the phase vocoder in computer music applications," *AES: Journal of the Audio Engineering Society*, vol. 26, 01 1978.

[75] J. Smith and X. Serra, "Parshl: An analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation," *Proceedings of the International Computer Music Conference*, 01 1987.

[76] J. Laroche and M. Dolson, "Improved phase vocoder time-scale modification of audio," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 323–332, 1999.

[77] ——, "About this phasiness business," in *Proceedings of the 1997 International Computer Music Conference, ICMC 1997, Thessaloniki, Greece, September 25-30, 1997*. Michigan Publishing, 1997. [Online]. Available: https://hdl.handle.net/2027/spo.bbp2372.1997.019

[78] ——, "Phase-vocoder: about this phasiness business," 1997.

[79] ——, "New phase-vocoder techniques for pitch-shifting, harmonizing and other exotic effects," in *Proceedings of the 1999 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics. WASPAA'99 (Cat. No.99TH8452)*, 1999, pp. 91–94.

[80] N. Juillerat, "Audio time stretching with controllable phase coherence," in *Audio Engineering Society Convention 142*. Audio Engineering Society, 2017.

[81] C. Duxbury, M. Davies, and M. Sandler, "Improved time-scaling of musical audio using phase locking at transients," 01 2012.

[82] M. Puckette, "Phase-locked vocoder," 11 1995, pp. 222 – 225.

[83] A. Moinet and T. Dutoit, "Pvsola: A phase vocoder with synchronized overlap-add," 09 2011.

[84] J. Bonada, "Wide-band harmonic sinusoidal modeling," 01 2008.

[85] J. Allen and L. Rabiner, "A unified approach to short-time fourier analysis and synthesis," *Proceedings of the IEEE*, vol. 65, no. 11, pp. 1558–1564, 1977.

[86] M. Goodwin, "Multiscale overlap-add sinusoidal modeling using matching pursuit and refinements," in *Proceedings of the 2001 IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics (Cat. No.01TH8575)*, 2001, pp. 207–210.

[87] M. Portnoff, "Time-scale modification of speech based on short-time fourier analysis," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 3, pp. 374–390, 1981.

[88] R. McAulay and T. Quatieri, "Magnitude-only reconstruction using a sinusoidal speech modelmagnitude-only reconstruction using a sinusoidal speech model," in *ICASSP '84. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 9, 1984, pp. 441–444.

[89] G. Peeters and X. Rodet, "Sinola: A new analysis/synthesis method using spectrum peak shape distortion, phase and reassigned spectrum," in *International Conference on Mathematics and Computing*, 1999. [Online]. Available: https://api.semanticscholar.org/CorpusID:4987635

[90] F. Hammer, "Timescale modification using the phase vocoder," Ph.D. dissertation, 09 2001.

[91] T. Karrer, E. Lee, and J. Borchers, "Phavorit: A phase vocoder for real-time interactive time-stretching," 01 2006.

[92] A. Roebel, "Shape-invariant speech transformation with the phase vocoder," 09 2010, pp. 2146–2149.

[93] M. Liuni and A. Roebel, "Phase vocoder and beyond," *Music/Technology*, vol. VII, pp. 73–89, 08 2013.

[94] J. Driedger and M. Müller, "A review of time-scale modification of music signals," *Applied Sciences*, vol. 6, p. 57, 02 2016.

[95] M. Caetano and P. Depalle, "On the estimation of sinusoidal parameters via parabolic interpolation of scaled magnitude spectra," in *2021 24th International Conference on Digital Audio Effects (DAFx)*, 2021, pp. 81–88.

[96] A. Roebel, "A new approach to transient processing in the phase vocoder," 09 2003.

[97] X. Serra, "A system for sound analysis/transformation/synthesis based on a deterministic plus stochastic decomposition," 1989. [Online]. Available: https://api.semanticscholar.org/CorpusID:60454701

[98] R. McAulay and T. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 4, pp. 744–754, 1986.

[99] A. Nuttall, "Some windows with very good sidelobe behavior," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 1, pp. 84–91, 1981.

[100] M. Goodwin, "Adaptive signal models: Theory, algorithms, and audio applications," Ph.D. dissertation, University of California, Berkeley, 1998.

[101] J. C. Glover, "Sinusoids, noise and transients: spectral analysis, feature detection and real-time transformations of audio signals for musical applications," Ph.D. dissertation, National University of Ireland Maynooth, 2012.

[102] M. Ali, "Adaptive signal representation with application in audio coding," Ph.D. dissertation, 1996.

[103] K. Hamdy, "Audio modeling for coding and time scaling applications," Ph.D. dissertation, 2000.

[104] T. S. Verma, S. N. Levine, and T. H. Y. Meng, "Transient modeling synthesis: a flexible analysis/synthesis tool for transient signals," in *Proceedings of the ICMC*, 1997.

[105] T. Verma and T. Meng, "An analysis/synthesis tool for transient signals that allows a flexible sines+transients+noise model for audio," in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98 (Cat. No.98CH36181)*, 1998.

[106] T. S. Verma, "An analysis/synthesis tool for transient signals," 1998.

[107] T. S. Verma and T. H. Y. Meng, "Extending spectral modeling synthesis with transient modeling synthesis," *Computer Music Journal*, vol. 24, no. 2, pp. 47–59, 2000. [Online]. Available: http://www.jstor.org/stable/3681927

[108] T. Verma, "Perceptually Based Audio Signal Model With Application to Scalable Audio Compression," Ph.D. dissertation, Elec. Engineering Dept., Stanford University (CCRMA), 2000.

[109] D. P. W. Ellis, "Sinewave and sinusoid+noise analysis/synthesis in matlab," 2003. [Online]. Available: http://www.ee.columbia.edu/~dpwe/resources/matlab/sinemodel/

[110] F. Nsabimana and U. Zölzer, "Transient encoding of audio signals using dyadic approximations," 2022.

[111] J. O. Smith and S. N. Levine, "Audio representations for data compression and compressed domain processing," 1998.

[112] M. Bosi and R. E. Goldberg, *Introduction to Digital Audio Coding and Standards.* USA: Kluwer Academic Publishers, 2002.

[113] A. den Brinker, J. Breebaart, P. Ekstrand, J. Engdegård, F. Henn, K. Kjörling, W. Oomen, and H. Purnhagen, "An overview of the coding standard mpeg-4 audio amendments 1 and 2: He-aac, ssc, and he-aac v2," *EURASIP J. Audio, Speech and Music Processing*, vol. 2009, 01 2009.

[114] L. Daudet, "A review on techniques for the extraction of transients in musical signals," in *Computer Music Modeling and Retrieval*, R. Kronland-Martinet, T. Voinier, and S. Ystad, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 219–232.

[115] S. N. Levine and J. O. S. III, "A sines+transients+noise audio representation for data compression and time/pitch scale modifications," 1998.

[116] S. N. Levine, T. S. Verma, and J. O. Smith, "Multiresolution sinusoidal modeling for wideband audio with modifications," *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98 (Cat. No.98CH36181)*, vol. 6, pp. 3585–3588 vol.6, 1998.

[117] K. Brandenburg, "Mp3 and aac explained," *journal of the audio engineering society*, 1999.

[118] A. den Brinker, J. Breebaart, P. Ekstrand, J. Engdegård, F. Henn, K. Kjörling, W. Oomen, and H. Purnhagen, "An overview of the coding standard mpeg-4 audio amendments 1 and 2: He-aac, ssc, and he-aac v2," *EURASIP J. Audio, Speech and Music Processing*, 2009.

[119] J. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. Sandler, "A tutorial on onset detection in music signals," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 5, pp. 1035–1047, 2005.

[120] S. Dixon, "Onset detection revisited," 09 2006.

[121] ——, "Simple spectrum-based onset detection," 01 2006.

[122] R. Zhou and J. Reiss, "Music onset detection combining energy-based and pitch-based approaches," 09 2007.

[123] W. Wang, *Machine Audition: Principles, Algorithms, and Systems*, ser. Premier Reference Source. Information Science Reference, 2011. [Online]. Available: https://books.google.co.za/books?id=4GsVQwAACAAJ

[124] B. Thoshkahna, F. Nsabimana, and K. Ramakrishnan, "A transient detection algorithm for audio using iterative analysis of stft." 01 2011, pp. 203–208.

[125] N. Ono, K. Miyamoto, J. Le Roux, H. Kameoka, and S. Sagayama, "Separation of a monaural audio signal into harmonic/percussive components by complementary diffusion on spectrogram," 2008.

[126] N. Ono, K. Miyamoto, H. Kameoka, and S. Sagayama, "A real-time equalizer of harmonic and percussive components in music signals." 01 2008, pp. 139–144.

[127] A. Roebel and X. Rodet, "Efficient spectral envelope estimation and its application to pitch shifting and envelope preservation," 09 2005, pp. 30–35.

[128] M. Caetano, J. J. Burred, and X. Rodet, "Automatic segmentation of the temporal evolution of isolated acoustic musical instrument sounds using spectro-temporal cues," 09 2010.

[129] A. Roebel, "Onset detection by means of transient peak classification," 2011. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01161440

[130] ——, "Onset detection in polyphonic signals by means of transient peak classification," 09 2005.

[131] B. Hamilton, "Non-stationary sinusoidal parameter estimation," Ph.D. dissertation, McGill Univ, McGill University, 2012.

[132] K. J. Werner and F. G. Germain, "Sinusoidal parameter estimation using quadratic interpolation around power-scaled magnitude spectrum peaks," *Applied Sciences*, vol. 6, no. 10, 2016. [Online]. Available: https://www.mdpi.com/2076-3417/6/10/306

[133] K. Werner, "The xqifft: Increasing the accuracy of quadratic interpolation of spectral peaks via exponential magnitude spectrum weighting," 01 2015.

[134] M. Abe and J. Smith, "Cqifft: Correcting bias in a sinusoidal parameter estimator based on quadratic interpolation of fft magnitude peaks," no. STAN-M-117, 2004. [Online]. Available: http://ccrma.stanford.edu/files/papers/stanm117.pdf

[135] ——, "Design criteria for the quadratically interpolated fft method (i): Bias due to interpolation," no. STAN-M-114, 2004. [Online]. Available: https://ccrma.stanford.edu/files/papers/stanm114.pdf

[136] J. Wells, "Reading the sines: Sinusoidal identification and description using the short time fourier transform."

[137] G. Strang, *Introduction to Linear Algebra*. Wellesley-Cambridge Press, 2021. [Online]. Available: https://books.google.co.za/books?id=c-1kzgEACAAJ

[138] ——, "Linear algebra mit opencourseware," https://ocw.mit.edu/courses/18-06-linear-algebra-spring-2010/, 2010.

[139] P. Grinfeld, "Math the beautiful," http://www.youtube.com/channel/UCr22xikWUK2yUW4YxOKXclQ, 2020.

[140] G. Sanderson, "3blue1brown - essence of linear algebra," https://www.youtube.com/watch?v=fNk_zzaMoSs&list=PLZHQObOWTQDPD3MizzM2xVFitgF8hE_ab.

[141] M. Goodwin and M. Vetterli, "Matching pursuit and atomic signal models based on recursive filter banks," *IEEE Transactions on Signal Processing*, vol. 47, no. 7, pp. 1890–1902, 1999.

[142] M. Caetano, G. Kafentzis, and A. Mouchtaris, "Adaptive modeling of synthetic nonstationary sinusoids," 11 2015.

[143] R. Roy, A. Paulraj, and T. Kailath, "Esprit-a subspace rotation approach to estimation of parameters of cisoids in noise," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 5, pp. 1340–1342, 1986.

[144] R. Badeau, "High resolution spectral analysis and non-negative decompositions applied to music signal processing," Ph.D. dissertation, 11 2010.

[145] A. L'vov, A. Seranova, R. Ermakov, A. Sytnik, and A. Muchkaev, "Comparison of methods for parameter estimating of superimposed sinusoids," in *Recent Research in Control Engineering and Decision Making*, O. Dolinina, I. Bessmertny, A. Brovko, V. Kreinovich, V. Pechenkin, A. Lvov, and V. Zhmud, Eds. Springer International Publishing, 2021.

[146] R. Badeau, G. Richard, and B. David, "Performance of esprit for estimating mixtures of complex exponentials modulated by polynomials," *IEEE Transactions on Signal Processing*, vol. 56, no. 2, pp. 492–504, 2008.

[147] O. Das, J. Abel, and J. Smith, "Fast music-an efficient implementation of the music algorithm for frequency estimation of approximately periodic signals," 09 2018.

[148] P. Masri and A. Bateman, "Improved modeling of attack transients in music analysis-resynthesis," 1996, pp. 100–103.

[149] 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, no. 5, 1995.

[150] S. Hainsworth and M. Macleod, "Time frequency reassignment: A review and analysis," 2003.

[151] B. Hamilton, P. Depalle, and S. Marchand, "Theoretical and practical comparisons of the reassignment method and the derivative method for the estimation of the frequency slope," in *2009 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2009.

[152] B. Hamilton and P. Depalle, "A unified view of non-stationary sinusoidal parameter estimation methods using signal derivatives," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012.

[153] S. Muševič and J. Bonada, "Comparison of non-stationary sinusoid estimation methods using reassignment and derivatives," 2010.

[154] S. Marchand and P. Depalle, "Generalization of the derivative analysis method to non-stationary sinusoidal modeling," in *Digital Audio Effects (DAFx) Conference*, Espoo, Finland, Sep. 2008, pp. 281–288. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00351950

[155] S. Muševič, "Non-stationary sinusoidal analysis," Ph.D. dissertation, Pompeu Fabra University, 2013.

[156] P. Flandrin, F. Auger, and E. Chassande-Mottin, *Time-Frequency Reassignment: From Principles to Algorithms*, 01 2002, pp. 179–204.

[157] K. Fitz and L. Haken, "On the use of time-frequency reassignment in additive sound modeling," *Advances in Engineering Software - AES*, vol. 50, 11 2002.

[158] B. Hamilton, "Non-stationary sinusoidal parameter estimation," Ph.D. dissertation, McGill University, 2012.

[159] J. J. Wells and D. T. Murphy, "High accuracy frame-by-frame non-stationary sinusoidal modelling," in *Proc. International Conference on Digital Audio Effects (DAFx)*, 2006.

[160] ——, "Single-frame discrimination of non-stationary sinusoids," in *2007 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2007.

[161] ——, "A comparative evaluation of techniques for single-frame discrimination of nonstationary sinusoids," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, 2010.

[162] F. Keiler and S. Marchand, "Survey on extraction of sinusoids in stationary sounds," 11 2002.

[163] M. Abe and J. Smith, "Am/fm rate estimation for time-varying sinusoidal modeling," in *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, vol. 3, 2005, pp. iii/201–iii/204 Vol. 3.

[164] P. Masri, "Computer modeling of sound for transformation and synthesis of musical signal," Ph.D. dissertation, 1996.

[165] P. Masri and C. Canagarajah, "Extracting more details from spectrum with phase distortion analysis," 1998.

[166] S. Mallat, *A Wavelet Tour of Signal Processing.* Academic Press, 1999.

[167] P. Addison, *The Illustrated Wavelet Transform Handbook: Introductory Theory and Applications in Science, Engineering, Medicine and Finance.* Taylor & Francis, 2002. [Online]. Available: https://books.google.co.za/books?id=RUSjIMQACQQC

[168] G. Kafentzis, Y. Pantazis, O. Rosec, and Stylianou, "An extension of the adaptive quasi-harmonic model," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 03 2012.

[169] M. Goodwin, "Matching pursuit with damped sinusoids," in *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, 1997, pp. 2037–2040 vol.3.

[170] S. Muševič and J. Bonada, "Derivative analysis of complex polynomial amplitude, complex exponential with exponential damping," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 488–492.

[171] K. Hermus, W. Verhelst, P. Lemmerling, P. Wambacq, and S. Van Huffel, "Perceptual audio modeling with exponentially damped sinusoids," *Signal Processing*, vol. 85, no. 1, pp. 163–176, 2005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165168404002361

[172] J. Jensen, R. Heusdens, and S. Jensen, "A perceptual subspace approach for modeling of speech and audio signals with damped sinusoids," *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 2, pp. 121–132, 2004.

[173] Z.-S. Liu, J. Li, and P. Stoica, "Relax-based estimation of damped sinusoidal signal parameters," *Signal Process.*, vol. 62, pp. 311–321, 1997.

[174] R. Boyer and K. Abed-Meraim, "Damped and delayed sinusoidal model for transient signals," *IEEE Transactions on Signal Processing*, vol. 53, no. 5, pp. 1720–1730, 2005.

[175] N. Ruiz Reyes, P. Vera Candeas, and F. López Ferreras, "Wavelet-based approach for transient modeling with application to parametric audio coding," *Digital Signal Processing*, vol. 20, no. 1, pp. 123–132, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1051200409000840

[176] P. Masri and A. Bateman, "Identification of nonstationary audio signals using the fft, with application to analysis-based synthesis of sound," in *Proc. IEE Colloquium on Audio Engineering. pp*, 1995.

[177] M. Lagrange, S. Marchand, and J. Rault, "Sinusoidal parameter extraction and component selection in a non stationary model," in *Proceedings of the Digital Audio Effects (DAFx02) Conference*, 2002.

[178] M. Lagrange, R. Badeau, B. David, N. Bertin, J. Echeveste, O. Derrien, S. Marchand, and L. Daudet, "The DESAM toolbox: spectral analysis of musical audio," in *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, 09 2010, pp. 254–261.

[179] M. Sprevak and M. Colombo, *The Routledge Handbook of the Computational Mind*, ser. Routledge Handbooks in Philosophy, 2018. [Online]. Available: https://books.google.co.za/books?id=ZiBtDwAAQBAJ

[180] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Algorithms for simultaneous sparse approximation. part i: Greedy pursuit," *Signal Processing*, vol. 86, no. 3, pp. 572–588, 2006, sparse Approximations in Signal and Image Processing. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165168405002227

[181] G. Davis, S. Mallat, , and Z. Zhang, "Adaptive time-frequency decompositions with matching pursuit," in *Wavelet Applications*, ser. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, H. H. Szu, Ed., vol. 2242, Mar. 1994, pp. 402–413.

[182] D. Wen, P. Jia, Q. Lian, Y. Zhou, and C. Lu, "Review of sparse representation-based classification methods on eeg signal processing for epilepsy detection, brain-computer interface and cognitive impairment," *Frontiers in Aging Neuroscience*, vol. 8, 2016.

[183] R. Gribonval, E. Bacry, S. Mallat, P. Depalle, and X. Rodet, "Analysis of sound signals with high resolution matching pursuit," 07 1996, pp. 125 – 128.

[184] R. Gribonval, P. Depalle, X. Rodet, E. Bacry, and S. Mallat, "Sound Signals Decomposition Using a High Resolution Matching Pursuit," in *Proc. Int. Computer Music Conf. (ICMC'96)*, Hong-Kong, Hong Kong SAR China, Aug. 1996, pp. 293–296. [Online]. Available: https://hal.inria.fr/inria-00576655

[185] M. Goodwin and M. Vetterli, "Atomic decompositions of audio signals," in *Proceedings of 1997 Workshop on Applications of Signal Processing to Audio and Acoustics*, 1997.

[186] n. ruiz reyes and p. vera candeas, "a sinusoidal modeling approach based on perceptual matching pursuits for parametric audio coding," *journal of the audio engineering society*, 2005.

[187] P. Vera-Candeas, N. Ruiz Reyes, J. Cuevas-Martínez, M. Rosa-Zurera, and F. Lopez-Ferreras, "Sinusoidal modelling using perceptual matching pursuits in the bark scale for parametric audio coding," *Vision, Image and Signal Processing, IEE Proceedings* -, vol. 153, pp. 431– 435, 09 2006.

[188] N. Ruiz Reyes and P. Vera Candeas, "Adaptive signal modeling based on sparse approximations for scalable parametric audio coding," *IEEE Transactions on Audio, Speech, and Language Processing*, 2010.

[189] P. Vera-Candeas, N. Ruiz-Reyes, M. Rosa-Zurera, F. Lopez-Ferreras, and J. Curpian-Alonso, "New matching pursuit based sinusoidal modelling method for audio coding," 2004.

[190] A. Petrovsky, V. Herasimovich, and A. Petrovsky, "Audio/speech coding using the matching pursuit with frame-based psychoacoustic optimized time-frequency dictionaries and its performance evaluation," 09 2016, pp. 225–229.

[191] M. Rosa-Zurera, E. A. Cortizo, and A. A. Petrovsky, "Matching pursuit with frame-based psychoacoustic optimized wp-dictionary," *Signal Processing Algorithms, Architectures, Arrangements, and Applications SPA 2007*, pp. 169–174, 2007.

[192] P. Vera-Candeas, N. Ruiz Reyes, M. Rosa-Zurera, D. Muñoz, and F. Lopez-Ferreras, "Transient modeling by matching pursuits with a wavelet dictionary for parametric audio coding," *Signal Processing Letters, IEEE*, vol. 11, pp. 349 – 352, 04 2004.

[193] N. R. Reyes and P. V. Candeas, "Optimizing a wavelet-based dictionary for transient modelling with application to parametric audio coding," 2006.

[194] H. E. Bass, L. C. Sutherland, A. J. Zuckerwar, D. T. Blackstock, and D. M. Hester, "Atmospheric absorption of sound: Further developments," *Journal of the Acoustical Society of America*, vol. 97, pp. 680–683, 1995.

[195] J. Smith, "Fir example," https://ccrma.stanford.edu/~jos/filters/FIR_Example.html.

[196] P. Welch, "The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms," *IEEE Transactions on Audio and Electroacoustics*, vol. 15, no. 2, pp. 70–73, 1967.

[197] A. Master and Y.-W. Liu, "Nonstationary sinusoidal modeling with efficient estimation of linear frequency chirp parameters," vol. 5, 05 2003, pp. V – 656.

[198] P. Leveau and L. Daudet, "Multi-resolution partial tracking with modified matching pursuit," *2006 14th European Signal Processing Conference*, pp. 1–4, 2006.

[199] H. W. M., "The effect of amplitude envelope on the pitch of sine wave tones," in *The Journal of the Acoustical Society of America*, 1978.

[200] J. Driedger, "Time-scale modification algorithms for music audio signals," Ph.D. dissertation, Saarland University, Saarbrücken, Germany, 2011.

[201] Y. Pantazis, O. Rosec, and Y. Stylianou, "Adaptive am–fm signal decomposition with application to speech analysis," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 2, pp. 290–300, 2011.

[202] M. Caetano, G. P. Kafentzis, A. Mouchtaris, and Y. Stylianou, "Adaptive sinusoidal modeling of percussive musical instrument sounds," in *21st European Signal Processing Conference (EUSIPCO 2013)*, 2013.

[203] S. Musevic and J. Bonada, "Distribution derivative method for generalised sinusoid with complex amplitude modulation," 2015.

[204] M. Betser, "Sinusoidal polynomial parameter estimation using the distribution derivative," *IEEE Transactions on Signal Processing*, no. 12, pp. 4633–4645, 2009.

[205] M. Koutsogiannaki, Y. Pantazis, and Y. Stylianou, "A novel method for the extraction of vocal tremor," 01 2009.

[206] Y. Pantazis, O. Rosec, and Y. Stylianou, "Adaptive am–fm signal decomposition with application to speech analysis," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, pp. 290 – 300, 03 2011.

[207] M. Caetano, G. Kafentzis, A. Mouchtaris, and Y. Stylianou, "Full-band quasi-harmonic analysis and synthesis of musical instrument sounds with adaptive sinusoids," *Applied Sciences*, vol. 6, p. 127, 05 2016.

[208] P. Antivasis, "eaqhm analysis and synthesis in python," https://github.com/Antibas/eaQHM-analysis-and-synthesis-in-Python.

[209] Mathworks, "Dynamic memory allocation and performance," https://www.mathworks.com/help/coder/ug/minimize-dynamic-memory-allocation.html.

[210] V. Lazzarini, J. ffitch, J. Timoney, and R. Bradford, "Streaming spectral processing with consumer-level graphics processing units," 2014. [Online]. Available: https://mural.maynoothuniversity.ie/5853/

[211] F. Nsabimana and U. Zölzer, "Transients detection and segmentation in audio signals," in *The Journal of the Acoustical Society of America*, 2007.

[212] H. Thornburg, "Detection and modeling of transient audio signals with prior information," Ph.D. dissertation, Stanford, CA, USA, 2005, aAI3186405.

[213] R. Gray and J. Berger, "Detection and modelling of transient audio signals with prior information," 2005.

[214] J. Glover, V. Lazzarini, and J. Timoney, "Real-time detection of musical onsets with linear prediction and sinusoidal modeling," *EURASIP Journal on Advances in Signal Processing*, 2011.

[215] W. Russell and T. Binder, *In the Wave Lies the Secret of Creation.* University of Science & Philosophy, 1995. [Online]. Available: https://books.google.co.za/books?id=OONHPQAACAAJ

[216] B. S. Atal and M. R. Schroeder, "Linear prediction analysis of speech based on a pole-zero representation," 1978.

[217] J. Wells, D. Murphy, and B.-S. WAVELETS, "Short-time wavelet analysis of analytic residuals for real-time spectral modelling," 2007.

[218] M. Goodwin, "Residual modeling in music analysis-synthesis," in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, vol. 2, 1996, pp. 1005–1008 vol. 2.

[219] J. Wells, "Personal conversations regarding sinusoidal modelling," 2016.

[220] D. Anderson, "Speech analysis and coding using a multi-resolution sinusoidal transform," in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, vol. 2, 1996, pp. 1037–1040 vol. 2.

[221] S. Levine, T. Verma, and J. Smith, "Alias-free, multiresolution sinusoidal modeling for polyphonic, wideband audio," in *Proceedings of 1997 Workshop on Applications of Signal Processing to Audio and Acoustics*, 1997.

[222] e. b. george and m. j. smith, "analysis-by-synthesis/overlap-add sinusoidal modeling applied to the analysis and synthesis of musical tones," *journal of the audio engineering society*, 1992.

[223] E. George and M. Smith, "Speech analysis/synthesis and modification using an analysis-by-synthesis/overlap-add sinusoidal model," *IEEE Transactions on Speech and Audio Processing*, vol. 5, no. 5, pp. 389–406, 1997.

[224] J. Glover, V. Lazzarini, and J. Timoney, "Real-time segmentation of the temporal evolution of musical sounds," *Journal of the Acoustical Society of America*, 2012.

[225] J. C. Glover, V. Lazzarini, and J. Timoney, "Metamorph: Real-time high-level sound transformations based on a sinusoids plus noise plus transients model," in *DAFX-12 the 15th Int. Conference on Digital Audio Effects*, September 2012. [Online]. Available: https://mural.maynoothuniversity.ie/4122/

[226] Y. Masuyama, T. Kusano, K. Yatabe, and Y. Oikawa, "Modal decomposition of musical instrument sounds via optimization-based non-linear filtering," *Acoustical Science and Technology*, vol. 40, pp. 186–197, 05 2019.

[227] G. Kafentzis, "Adaptive sinusoidal models for speech with applications in speech modifications and audio analysis," Ph.D. dissertation, 2014.

[228] P. Vera-Candeas, N. Ruiz-Reyes, M. Rosa-Zurera, D. Martinez-Munoz, and F. Lopez-Ferreras, "Transient modeling by matching pursuits with a wavelet dictionary for parametric audio coding," *IEEE Signal Processing Letters*, vol. 11, no. 3, pp. 349–352, 2004.

[229] J. An and O. An, "Additive synthesis based on the continuous wavelet transform: A sinusoidal plus transient model," 08 2003.

[230] C. Duxbury, M. E. Davies, and M. B. Sandler, "Separation of transient information in musical audio using multiresolution analysis techniques," 2001.

[231] L. DAUDET, S. Molla, and B. Torrésani, "Transient detection and encoding using wavelet coefficient trees," 2001.

[232] W. Ahmad, H. Hacihabiboglu, and A. Kondoz, "Analysis-synthesis model for tyransient impact sounds by stationary wavelet transform and singular value decomposition," 2008.

[233] D. Mackenzie, "Wavelets: Seeing the forest and the trees," 2001.

[234] K. Soman, K. Ramachandran, and N. Resmi, *Insight Into Wavelets : from Theory to Practice*. PHI Learning, 2010. [Online]. Available: https://books.google.co.za/books?id=V7DgqDL_ZuAC

[235] L. Bruce, A. Cheriyadat, and M. Burns, "Wavelets: getting perspective," *IEEE Potentials*, 2003.

[236] A. Graps, "An introduction to wavelets," *IEEE Computational Science and Engineering*, 1995.

[237] M. Holschneider, R. Kronland-Martinet, J. Morlet, and P. Tchamitchian, "A real-time algorithm for signal analysis with the help of the wavelet transform," in *Wavelets*, J.-M. Combes, A. Grossmann, and P. Tchamitchian, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1990, pp. 286–297.

[238] A. P. Nason and B. W. Silverman, "The stationary wavelet transform andsome statistical," 1995.

[239] T. Liu and A. Fraser-Smith, "Detection of transients in 1/f noise with the undecimated discrete wavelet transform," *IEEE Transactions on Signal Processing*, 2000.

[240] M. Lang, H. Guo, J. Odegard, C. Burrus, and R. Wells, "Noise reduction using an undecimated discrete wavelet transform," *IEEE Signal Processing Letters*, 1996.

[241] B. Leslie and M. Sandler, "A wavelet packet algorithm for 1d data with no block end effects," in *1999 IEEE International Symposium on Circuits and Systems (ISCAS)*, 1999.

[242] J. Nealand, A. Bradley, and M. Lech, "Overlap-save convolution applied to wavelet analysis," *IEEE Signal Processing Letters*, 2003.

[243] P. Rajmic, "Exploitation of the wavelet transform and mathematical statistics for separation signals and noise, (in czech)," Ph.D. dissertation, Brno University of Technology, 2004.

[244] ——, "Method for real-time signal processing via wavelet transform," 01 2005, pp. 368–378.

[245] M. segmentované, P. Rajmic, and J. Vlach, "Method of segmented wavelet transform for real-time signal processing," 2005.

[246] Z. Prusa, "Segmentwise discrete wavelet transform," Ph.D. dissertation, Brno University of Technology, 2004.

[247] P. Rajmic and J. Vlach, "Real-time audio processing via segmented wavelet transform," 2007.

[248] P. Rajmic, Z. Prusa, and R. Konczi, "Vst plug-in module performing wavelet transform in real-time," 2012.

[249] G. Strang and T. Nguyen, *Wavelets and Filter Banks*. Wellesley-Cambridge Press, 1996. [Online]. Available: https://books.google.co.za/books?id=Z76N_Ab5pp8C

[250] E. Battenberg and R. Aviûienis, "Implementing real-time partitioned convolution algorithms on conventional operating systems," 2011.

[251] F. Wefers, "Partitioned convolution algorithms for real-time auralization," Ph.D. dissertation, 09 2014.

[252] D. Darlington, L. Daudet, and M. Sandler, "Digital audio effects in the wavelet domain," 2002.

[253] J. Fowler, "The redundant discrete wavelet transform and additive noise," *IEEE Signal Processing Letters*, vol. 12, no. 9, pp. 629–632, 2005.

[254] J. Smith, "Overlap save," https://ccrma.stanford.edu/~jos/OLA/Overlap_Save_Method.html.

[255] S. Paquelet and V. Savaux, "On the symmetry of fir filter with linear phase," *Digital Signal Processing*, 2018.

[256] W. Ahmad, H. Hacihabiboglu, and A. Kondoz, "Analysis-synthesis model for transient impact sounds by stationary wavelet transform and singular value decomposition," 2008.

[257] V. Lombardi, "Packers in their first team meeting in 1959," 1959.

[258] X. Rodet and F. Jaillet, "Detection and modeling of fast attack transients," in *ICMC*, 2001.

[259] K. Fitz and P. Christensen, "Transient preservation under transformation in an additive sound model," 06 2001.

[260] G. P. Kafentzis, G. Degottex, O. Rosec, and Y. Stylianou, "Pitch modifications of speech based on an adaptive harmonic model," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 7924–7928.

[261] ——, "Time-scale modifications based on a full-band adaptive harmonic model," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 8193–8197.

[262] K. H. Nielsen, "Practical linear and exponential frequency modulation for digital music synthesis," 2020.

[263] M. Desainte-Catherine and S. Marchand, "High precision fourier analysis of sounds using signal derivatives," *J. Audio Eng. Soc.*, vol. 48, 04 2002.

[264] A. Gnutti, F. Guerrini, N. Adami, P. Migliorati, and R. Leonardi, "A wavelet filter comparison on multiple datasets for signal compression and denoising," *Multidimensional Systems and Signal Processing*, vol. 32, 01 2021.

[265] J. Hopkins and D. Sugerman, *No One Here Gets Out Alive*. Warner Books, 2006. [Online]. Available: https://books.google.co.za/books?id=LwwhPwAACAAJ

[266] "Uad plugins." [Online]. Available: https://www.uaudio.com/uad-plugins.html

[267] ShadowFx, "Reflexion," https://zenonrecords.bandcamp.com/track/shadow-fx-interpulse-reflexion.

[268] Lumen, "Gruntled," https://zenonrecords.bandcamp.com/track/lumen-gruntled.

[269] Pspiralife, "Macro micro," https://zenonrecords.bandcamp.com/track/pspiralife-macro-micro.

[270] S. Léger, "Son of sun," https://alldayidream.bandcamp.com/track/son-of-sun-2.

[271] Hermanez, "Tale of the unexpected," https://alldayidream.bandcamp.com/track/tale-of-the-unexpected.

[272] P. Ciez, "Baohum," https://alldayidream.bandcamp.com/track/baohum.

[273] J. Murray, "Wavelets."

[274] Petran, "Petran loop2."

[275] Elowinz, "Granjurema," https://beatspace-parvati.bandcamp.com/album/granjurema.

[276] ——, "Granjurema," https://beatspace-parvati.bandcamp.com/album/granjurema.

[277] M. Betser, P. Collen, G. Richard, and B. David, "Estimation of frequency for am/fm models using the phase vocoder framework," *IEEE Transactions on Signal Processing*, vol. 56, no. 2, pp. 505–517, 2008.

[278] J. Wells, "Interactive reverberation modelling - supporting materials," http://www.jezwells.org/publications/reverberation_modelling.

[279] J. Nieuwenhuijse, R. Heusens, and E. Deprettere, "Robust exponential modeling of audio signals," in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98 (Cat. No.98CH36181)*, vol. 6, 1998, pp. 3581–3584 vol.6.

[280] R. Badeau, B. David, and G. Richard, "A new perturbation analysis for signal enumeration in rotational invariance techniques," *IEEE Transactions on Signal Processing*, vol. 54, no. 2, pp. 450–458, 2006.

[281] S. Muševič and J. Bonada, "Generalized reassignment with an adaptive polynomial-phase fourier kernel for the estimation of non-stationary sinusoidal parameters," 09 2011, pp. 371–374.

[282] S. Marchand, "The simplest analysis method for non-stationary sinusoidal modeling."

[283] G. P. Kafentzis, "Personal conversations regarding eaqhm," 2022.

[284] Y. Landman, "The nodes of a vibrating string are harmonics," http://www.jezwells.org/research/dafx10/, 2010.