# A matching filter and envelope system for timbral blending of the bass guitar

Michael Middleton

MA by Research

University of York

Music

January 2021

# Abstract

A method of intelligent filter curve estimation from signal spectra is investigated to assess its viability of blending the perceived timbres of two signals together. By influencing the spectrum of a source signal with that of a modifier, its magnitude spectrum will be reshaped to resemble the modifier signal and reflect some of its timbral characteristics more closely. A system of transplanting the time-domain signal envelope of a signal onto a host is also presented in a combined system. The intended purpose of such a system is in the development of a hybrid acoustic-electric instrument where the timbral products of an expressive performance may be used to manipulate the spectrum and envelope of musical signals. A bass guitar is studied as the source instrument given the wide range of expressive techniques that may be executed on the instrument. Further analysis of spectra gathered from bass guitar performance techniques are used to provide deeper insight into performance techniques that may be performed on the instrument.

# Contents

# Part I – Review of literature

Historically, there has been extensive research in the field of harmonics and timbre, often concerned with the analysis of speech (Zahorian and Hu 2008, Ying, Jamieson and Michell 1996, Talkin 1995). From a musical perspective, research into timbre has largely been concerned with data retrieval from musical signals (Fritz, Blackwell, Cross, Woodhouse and Moore 2012, Peeters, Giordano, Susini, Misdariis and McAdams 2011, Wake and Asahi 1998); instrument recognition and computer assisted transcription programs have been developed in response. Furthermore, instruments have been modelled digitally, allowing for some semi-realistic string timbres to be synthesized (Karplus and Strong 1983, Karjalainen, Valimaki and Tolonen 1998, Sullivan 1990). All these viewpoints are considered in the research summarised here, which forms the basis of analysing spectra and estimating an ideal filter curve using modified Yule-Walker equations. Being the subject instrument of this research, the electric bass guitar will be related to in examples concerning musical information retrieval in preparation for the original research to come in part II.

Exposing the spectral characteristics of bass guitar performance methods is important to musicians outside of a technological environment. A deconstruction of the timbral effects of common playing techniques can provide valuable insight into the phases a note will travel through in its lifetime. Musicians may come to a more intimate understanding of their instruments through uncovering the physical behaviours of the sounds it produces and how they may be manipulated. Within music technology, this understanding aids further work in synthesis and digital instrument modelling, as realistic synthesis of (electro) acoustic instruments requires extensive documentation of their spectra.

# 1. Development of the electric bass guitar

## 1.i. Instrument classification

To understand the future development of a hybrid instrument using timbral blending systems, an overview of the historical development of the bass guitar is beneficial. Many electronic instruments have been developed in modern times modelled on the contemporary electric guitar, borrowing aspects such as shape and means of performance. Over centuries of progressive development, the humble box-lute metamorphosised into the electroacoustic pickup-driven instrument that is familiar to musicians globally today (Jahnel 2000). Furthermore, as performance technique and timbral outcome of the instrument is crucial to the research presented in this paper, it would be helpful to refer to a system of instrument classification to assess where a hypothetical hybrid instrument would fit in amongst its predecessors. Guitars (and bass guitars by extension) are classified as chordophones by Hornbostel and Sachs (1914) in the widely used instrument categorisation system that bears their names. The authors define chordophones as instruments with one or more strings held taut between fixed points, placing guitars within the same greater family as most other stringed instruments. Hornbostel and Sachs greatly expand on their definition of chordophones by sub-categorising a guitar as a '[lute] whose body is built up in the shape of a bowl [and is] classified as bowl lute'. Each category of the Hornbostel-Sachs system is assigned a number for cataloguing purposes, derived from the steps that must be taken through the system to arrive at the category in question. Take the example of an acoustic guitar. From first to last digit, guitars are chordophones (assigned the category number 3) with the sound being amplified by resonating in its body (subcategory 2). The strings on a guitar are suspended above the body running in parallel with its surface (sub-subcategory 1). With this information, we can deduce that an acoustic guitar is indeed a form of lute and so the primary category, 321, is produced. These qualities are all shared with other stringed

instruments that are held by the performer such as lutes and violins but rules out larger stringed instruments like pianos. To further refine the definition, a guitar can be described as having a neck of simple construction which the strings contact when pressure is applied (3), shaped like a flat plane (2). Finally, its physical construction is box-like, with a flat front and back face (2). This new number, 322, is written after the first number separated by a period to arrive at the final catalogue number of 321.322. Other necked box lutes include violins and banjos, demonstrating that the system is not too granular as to separate every instrument into its own category. Further refinement may be specified; for example, guitars and banjos may be reduced even further to 321.322-5, the final digit expressing the conventional method of play (plucked strings using fingers or plectrums).

Hornbostel and Sachs' definition has been challenged recently, perhaps due to the questionable grouping of instruments with little practical relation, or the persistence of vague terminology within their classification system (Weisser and Quanten 2011, Kartomi 1990). The comparisons between guitars and banjos are immediately obvious as both are performed using similar techniques and share traits in construction, such as both instruments being fretted. However, the comparisons between a guitar and a violin are much less apparent. Much like the guitar and banjo, a violin can further be classified as 321.322-71, indicating that it is primarily a bowed instrument. A flaw with the Hornbostel-Sachs method becomes evident here. Given that the final digits (indicating playing methods in this instance) may be omitted, and that in the preceding six digits there is no indication as to the sound or performance style of the instrument, it may be argued that the system is overly concerned with the physical construction of the instrument and negates other important aspects of its being, including but not limited to its sound (or timbre), method of play and cultural heritage. Weisser and Quanten (2011) propose a modular approach to musical instrument classification in response to the strange groupings of instruments that the original system may be subject to. In their criticism of Hornbostel and Sachs' method, they note that the proposed framework of instrument classification is too restrictive by design and that it ignores much of the essential qualities of an instrument, resulting in erroneous or

culturally unacceptable classifications of instruments. Although instruments that produce sound via electronic means were developed long after the proposal of the Hornbostel-Sachs method, many revisions have included electrophones since the first revision of the system by Sachs (1940), who identified electrophones as a fifth group of musical instruments. In that time however, electrophones were rudimentary in their designs and instruments were limited to simple devices such as the Theremin or instruments amplified by electronic means (Glinsky 2000). It can be argued that this definition is haphazard and vague, clumping instruments that share almost every quality with their unamplified counterparts together into one disorganised group (Kartomi 1990). As any instrument may be amplified electronically in some capacity, the amplified electrophones may logically include a variation of every acoustic instrument in existence equipped with a microphone, rendering it effectively useless. More recent revisions of the Hornbostel-Sachs system provide clearer definitions for electrophonic instruments and amplified devices. In reaction, a revision was proposed by the organisation Musical Instrument Museums Online (MIMO, 2011) offering extensive sub-categorization for electronic and amplified instruments. To illustrate, Sachs' 1940 revision had one subcategory (53) for instruments that produce their sound via electrical means but specifically by using oscillators, leaving no clear means to place digital synthesizers in the system. Furthermore, all electrically amplified instruments are grouped under category 52 with no regard for the characteristics of the device being amplified. In the MIMO revision, electric guitars are placed in subcategory 513 (electroacoustic chordophones) and digital synthesizers are assigned subcategory 541, demonstrating the need for granularity as instrument design continues to evolve with technology.

## The modern guitar

Antonio de Torres Jurado, a Spanish luthier, created an instrument that is often recognised as the first modern acoustic guitar design around 1850 (Heck, 2001). Of note to musicologists is the fan-braced construction Torres Jurado built into the guitar's body, reinforcing it and altering how the body resonates. His design was largely popularised by a

wave of Spanish guitarists who rose to prominence during and succeeding his lifetime. Ultimately then, the apparent standardisation of guitar design was dictated by the instruments the musicians of the era chose. The implications of this are something to consider - it is hard not to wonder how different the guitar could be today had history favoured a slightly different method of construction or had a more Moorish design taken hold in middle-age Spain rather than the Latin style that prevailed. For the benefit of this paper, it also provokes thought on the future of guitar construction. Here, the purpose for creative applications of technological concepts becomes clear. Technology and musical instrument design are in a symbiotic relationship - a stride taken in one discipline pulls the other forward.

One such creative technological application, the electric guitar pickup, revolutionised the design of the instrument (O'Connor, 2016). A type of specialised magnetic pickup, the concept is simple; fine enamelled copper wire is wound around a magnetised core thousands of times. When metal guitar strings are plucked and vibrate adjacent to the pickup an electrical current is generated (Lawing 2017). This signal may then be amplified to artificially boost the sound produced by the guitar, allowing the performer to have greater control over the volume of their instrument by adjusting the amplifier gain. The electric guitar is no longer required to have a resonant body so it can be constructed of any solid material at an arbitrary size. Further experimentation using amplification technology led to creative uses of manipulation of sound using electronics, which in turn birthed new performance techniques informed by this newfound control over timbre (Herbst 2019). The cyclical nature of breakthrough from creative experimentation as previously introduced surfaces here once again as the musicians of the 20th century place technological development on a trajectory to suit their artistic needs.

Finally, the electric bass guitar, the subject instrument of this paper, can be discussed. Bass guitar construction is fundamentally related to guitar construction and as such, both instruments share a great deal of similarities (Brewer 2003). There are several construction hallmarks that one may expect to find exclusive to a bass guitar, however.

Perhaps most notably, a bass will usually have fewer strings than a guitar, with four-string designs being predominant. Four-string basses are typically tuned in fourths one octave below a standard guitar, with the lowest string being E and the highest being G. As a result, a four-string bass guitar in such a tuning will be able to produce notes with fundamental frequencies between approximately 40Hz to 400Hz. Any number of strings may be added above the typical four-string standard to extend the range of the instrument (Roberts 2019).

## 1.ii. The place of a timbral blending system in a progressive bass guitar design

Finally, the matching filter and envelope systems investigated in this paper can be married to an electric bass guitar to propose a hypothetical electrophonic instrument, progressing on the established design of the guitar. This instrument would utilise a timbral blending system to alter the output signal of the bass guitar informed by an excitation (or "modifier") signal. Digital signal processing occurs to reshape (time-domain) note and spectral envelopes present in the bass guitar signal. The modified signal is then sent to an output jack for amplification. Ideally, the matching system should influence the timbre of the bass guitar to resemble that of the modifier signal whilst retaining key characteristics of the source timbre, such as transients from the string being plucked represented as high-energy noise in the frequency domain. It should be noted that a modified bass guitar connected to a larger desktop computer for digital signal processing in this way can still be classified as a single instrument as the bass guitar would depend on the computer to render its processed sound.

Placing such an instrument in the MIMO-revised Hornbostel-Sachs system is slightly difficult, as the instrument retains key components of both electroacoustic chordophones and digital synthesizers. Categorising the guitar-like electronic instruments such as the Kitara (Misa Digital, n.d.) and Stepp DG-1 (Gilby 1987) is simple by comparison – they are both digital synthesizers assuming a different form to the more commonly seen keyboard

interface. Others such as the SynthAxe (Stansfield 2013) and Ztar series (StarrLabs, n.d) would not appear in the Hornbostel-Sachs method at all as the instruments do not produce their own sounds but control devices using MIDI instead. In these instances, the device (presumably a synthesizer of some description) would be the sound generator and therefore the instrument to be classified. For the bass guitar augmented with the matching filter, we are left with three options for classification, of which there appears to be no ideal candidate. Firstly, subcategory 541, a digital synthesizer, could be considered as the modifier signal sent to the matching system is sourced from a synthesizer in this example and is therefore integral to the final sound produced. However, no method of synthesis is involved in the actual rendering of the signal, rather it is just informed by a signal which happens to be synthesized. Secondly, it may be classified as an electric bass guitar, subcategory 321.322. This reasoning is based on conventional methods of processing bass guitar signals by means of amplification, distortion or any other method of effects processing. Essentially, the matching filter is an audio effect applied to the output bass guitar signal, so it could be argued that it does not meaningfully alter the construction of the bass guitar or the performance techniques used. Lastly, if the MIMO adaptation is being adhered to, the modified bass could be categorised as an electroacoustic chordophone, subcategory 513, upon the same basis that the bass guitar has not underwent enough alteration in its fundamental design to warrant recategorisation.

# 2. Timbre deconstructed

## 2.i. What is timbre?

Approaching timbre from a technical standpoint often makes for a challenging task. When someone who is assessing musical subjects from such an objective angle is asked about pitch or loudness, they will often insist that the discussion is focussed on *frequency* or *amplitude* instead. Such is the result of a disconnect in language between a creative source and an academic source, the former expressing their ideas through subjective language and the latter preferring the objective. Pitch is our perception of fundamental frequency in a sound (Plack, Oxenham, Fay and Popper 2005). Loudness is derived from our perception of sound pressure, which in turn is based upon the amplitude of the subject signal (Goldstein 2010, Raichel 2011). However, both parties are expressing the same general concepts of sound so despite this minor disconnect both can communicate their ideas clearly. Difficulty arises when *timbre*, an elusive property of sound, must be discussed from an objective standpoint. Adopted from modern French, there is no single word in English that can be used to express the concept of timbre accurately in a musical context. Therefore, the word is frequently in its original French form in English texts. The German word 'klängfarbe' (literally 'sound colour') was proposed as a German translation (Helmholtz, 1885); its use continues today as synonym of timbre. The English translation by Ellis (1885) interprets 'klängfarbe' as 'quality of tone' (or 'tone quality'), an understandable derivation from two possible translations from French, 'tone' and 'quality'.

Ellis objected to the usage of 'timbre', instead preferring to convey the idea of musical quality through 'tone'. Nonetheless, 'timbre' was adopted into the modern musical lexicon over the course of the early 20th century. A formal definition of 'timbre' was provided by the American National Standards Institute (ANSI, 1960):

> ‘Timbre is that attribute of auditory sensation in terms of which a listener can judge that two sounds similarly presented and having the same loudness and pitch are dissimilar’. (ANSI, 1960).

The ANSI definition is frequently referenced in texts on timbre and makes for an accurate, if evasive, description of the property. Timbre here is portrayed as an attribute of sound, distinct from perceived pitch and loudness, that contains the character or quality of the sound, the complex property that gives the emitter its unique sonic properties. The ANSI definition has faced criticism in more recent years. Houtsma (1997) takes issue with the lack of specificity surrounding the definition of timbre itself, highlighting that ANSI definitions of pitch and loudness refer to our perceptions of fundamental frequency and sound pressure respectively. In contrast, the definition of timbre is merely every aspect of a waveform apart from pitch and loudness. From this perspective, the ANSI definition can be critiqued as being inadequate as it leaves much of the subject’s actual definition for the reader to deduce by reduction. Bregman (1990) suggested that an alternate system must be developed to articulate timbre as our current language and definitions are too restrictive to properly articulate its complex nature.

Despite the back and forth between academics, the concept of timbre existing as a third property of sound perceptible by humans separate from pitch and loudness has been supported and upheld from early discussions on timbre (Helmholtz, 1885) to contemporary discourse in musicology (Kanno, 2001). Perhaps the clearest way to illustrate the meaning of timbre is to apply it to a musical scenario. Ask the participant to picture two different instruments, say a piano and a clarinet, playing the same note at the same loudness. Even though the perceived pitch and loudness of both instruments are equal to the observer, they may still distinguish between the two sounds based upon the sonic qualities possessed by each. The unique sounds produced by the piano, clarinet, or any instrument, as a result of its construction and form, can be defined as its timbre. By extension, any sound (musical or otherwise) can be described as having a certain timbre, which as Bregman critiqued, is often

defined using vague and colloquially understood terms. To illustrate, descriptors such as ‘muffled’ and ‘bright’ are commonly used to summon a reference to a sound in the head of the observer (Darke 2005). Even with little prior musical education, this observer can make an informed guess on what timbre a muffled sound may have as the adjective stimulates the imagination, evoking a general idea of its properties in the process. This simple method of translating sound to its quality as perceived by the listener is appreciated by scholars (Smalley 1994).

## The audio spectrum

Observations into what physical property of sound could produce timbre continued throughout the early 20th century. Seashore (1938) recognised timbre to be related to spectral content. He follows on to say:

> ’In general, we may say that, aside from accessory noises and inharmonic elements, the timbre of a tone depends upon (1) the number of harmonic partials present, (2) the relative location or locations of these partials in the range from the lowest to the highest, and (3) the relative strength or dominance of each partial.’

Fourier (1878) demonstrated the link between waveform shape and its harmonic content in the Fourier series. At last, we are given a physical indication as to what timbre is, therefore allowing it to be observed, measured and deconstructed. It can be assumed that the further a waveform diverges from a sinusoidal shape it will contain more apparent spectral content alongside the fundamental frequency. Therefore, timbre is informed by both time and frequency-domain information as defined by the shape of the waveform concerned. With this knowledge, it is possible to appreciate the sheer scale of what constitutes the timbre of a sound, perhaps contributing to what makes it so difficult to define. Given that the definition of timbre arrived upon previously is everything besides the perceived pitch and

loudness of a musical note, it can be deduced that timbre is defined by the presence of any other spectral components in the signal and their trajectories over time.

A common way to portray a signal in the frequency-domain, and a concept that will recur frequently in this paper, is to plot its magnitude spectrum. Fig. 2.1 shows the spectrum of a bass guitar note and presents a typical means of graphing spectral data, with the X and Y axes representing frequency and amplitude respectively. Individual harmonic elements and their place in series can be seen clearly across the length of the graph. Spectra are obtained from signals by using mathematical transforms on sampled waveforms, a concept explained further in chapter 3.

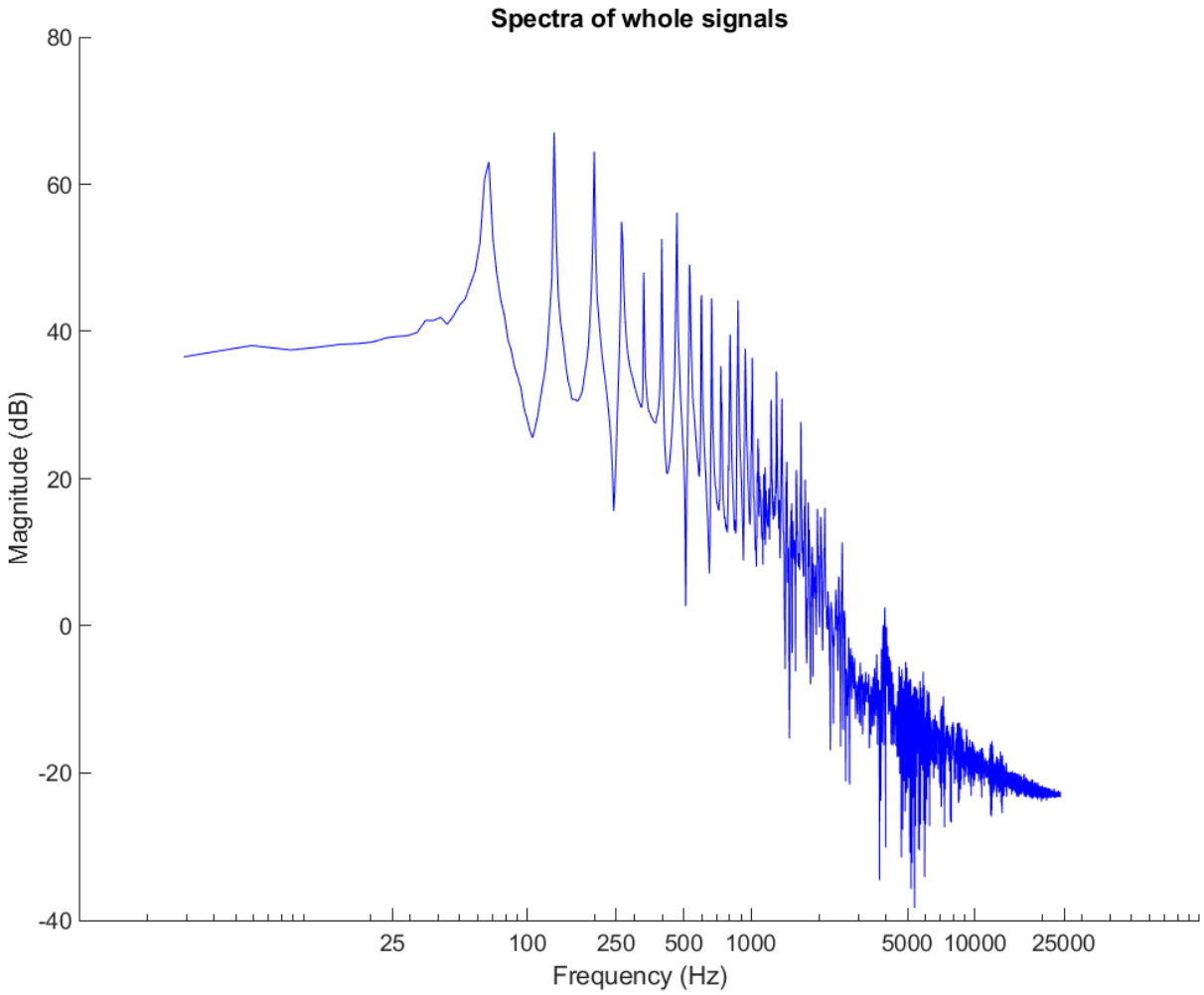


*Figure 2.1 - spectrum of a frame of audio. Its harmonic content is displayed as peaks at varying magnitudes on the X axis. Data was obtained using a 16,384-point FFT. The X axis is scaled logarithmically for readability.*

## 2.ii. Timbral results of bass guitar performance methods

The techniques analysed in this research constitutes fingerstyle playing, using a plectrum, using the thumb (or 'thumbing'), slapping and popping notes. Playing styles described here are amongst those noted and defined in other studies involving bass guitar timbre and spectra (Abesser, Lukashevich and Schuller 2010, Kramer, Abesser, Dittmar and Schuller 2012). Though there are limitless ways to perform on a bass guitar, as there is on any instrument, these techniques have been selected specifically due to their ubiquity in contemporary bass guitar recordings across a wide variety of musical genres (Brewer, 2003). Yasuda and Hama (2006) provide a detailed analysis of bass guitar timbre, detailing the spectra obtained from bass guitar signals. Their aim was to resynthesize the timbre of a bass guitar using their findings as a guideline; the outcomes of their experiments were promising. Significant contributions have been made in modelling the behaviour of plucked strings, allowing for a general understanding of how harmonic structures (and timbre by extension) are formed when a string is plucked (Karplus and Strong 1983, Sullivan 1990, Karjalainen, Valimaki and Tolonen 1998).

### Performance techniques

Fingerstyle playing is a typical plucking-hand performance technique on a bass guitar. A bassist will typically use the index and middle finger to pluck the strings at any point on the body. The thumb can rest on the pickup of the bass or on the string adjacent to the one being played to mute it, preventing unwanted resonance or notes being struck accidentally. The exact style of fingerstyle playing varies between performers, with some using just one finger to maintain a consistent timbre, whilst others incorporate their ring fingers to increase the speed of their playing. A common observation in playing techniques is the timbre largely depends on the surface contacting the string. Fingerstyle playing often produces a rounded, warm timbre due to the softness of a fingertip although callouses will likely form on the fingers after a while, making them tougher and producing more high

frequency content and inharmonic sinusoidal components. Softer surfaces such as fingertips will often see a reduction in high frequency content produced as the string can 'ease' back into its resting position as it rolls off the surface. Dynamically, there can be some variance between notes as it is more difficult to recreate this technique consistently, especially when playing quickly, as each finger inherently has differing physical strengths. Chords can be performed by 'raking' a finger up or down the strings whilst holding a shape on the neck, or by using multiple fingers to pluck more than one string at the same time in a technique like what may be performed on a guitar.

Just like a guitar, a plectrum may be used on the bass guitar in place of the performer's fingers. Playing using a plectrum can allow the bassist to play faster or perform complex rhythms with less physical strain when compared to fingerstyle playing. As a result, the dynamics produced by this method are often more even than what would be expected from a fingerstyle performance. As a demonstration of the importance of timbre to musicians, plectrums are sometimes favoured bassists who desire a different tone (Vega, 2020). It is therefore characterised as a dynamically consistent method of performance which produces a bright timbre.

Plucking the string using the thumb of was a technique developed early in the life of the bass guitar as it began to replace the upright bass in jazz bands (Brewer, 2003). Some performers would elect to use their thumb to replicate the warm, acoustic sound of an upright bass on the bass guitar. The technique is performed by resting the heel of the right hand on the bridge of the bass and plucking the string using the knuckle or pad of the thumb. The thumb is generally the softest finger on the hand as it is more resistant to hard callouses forming as a result of playing. In contrast to using a plectrum, which would increase the high frequency content produced by the instrument due to its hard surface, little inharmonic and high frequency content is produced by thumbing. The performer can elect to use the knuckle of their thumb to pluck rather than the pad which would counteract this effect slightly. Due to the relative strength of the thumb compared to the other fingers, it may be easier for the performer to produce notes with more accurate dynamics when compared to

playing fingerstyle, though this dynamic consistency may be lost should the bassist need to play quickly.

Slapping and popping are two techniques often used in conjunction with one another. The lower strings on the bass guitar are usually slapped, whilst the higher ones are popped. To perform a bass slap, the wrist is held perpendicular to the body of the bass with the thumb extended away from the other fingers. The wrist is then flicked or twisted to collide the bony part of the thumb near the knuckle against the string. A pop is performed by pulling the string away from the body of the bass before releasing it, causing the string to bounce against the neck (Oppenheim, 1981). Slapping and popping are two inherently loud techniques due to their percussive natures and both techniques are characterised by bright timbres. Bass popping is sometimes compared to the crack of a snare drum - just like how the metal snares snap against the skin of the drum, the string of the bass rattles against the neck producing a similar timbre. The sound of a bass slap could be described as sounding woody and resonant with a distinctive transient peak as the note is first played. Spectrally, these techniques carry a lot of mid and high frequency excitation and complex harmonic structures as the strings are treated so violently. The percussive qualities of these techniques also create inharmonic and noisy content in bass guitar signals, an effect that may also be produced through selectively muting strings.

# 3. Fourier analysis

The frequency-domain analysis of a series of data such as a musical signal can be described as Fourier analysis. The namesake is derived from the work of Fourier (1878) which detailed how a waveform could be represented as a sum of sine and cosine functions known as the Fourier series. When somebody talks about performing a 'Fourier transform', they likely refer to the discrete or fast Fourier transform (DFT and FFT), two frequently used mathematical transformations used to obtain spectral information from a signal (Oppenheim, Buck and Schafer, 1999). The DFT is simply the Fourier transform applied to a discrete (finite) series of evenly spaced samples from the time domain to retrieve a data series from the frequency domain. Eq. 3.1 describes the DFT algorithm on a dataset `x[n]` of length `N`.

$$X\left[k\right] = \sum_{n=0}^{N-1} x\left[n\right] \cdot e^{-\frac{i2\pi}{N} kn}$$

*Equation 3.1 - the DFT formula. The DFT is a slow algorithm due to the complex multiplication using* `e`.

Due to the DFT's lengthy computing time, it is used infrequently on larger datasets in this unmodified form. In fact, the DFT is so inefficient it will almost always be substituted for another derivative function capable of processing data far quicker. The number of calculations that must be performed by the DFT increases exponentially as the input sample size increases, making the DFT unsuitable for use with data sets aside from the very smallest in most applications.

The fast Fourier transform, or FFT, was developed in response to these needs (Cooley and Tukey, 1965). It serves the same function as the DFT but can be used more efficiently on larger sample sizes as the number of calculations that must be performed increases logarithmically rather than exponentially. The trade-off is a small degree of accuracy, although the errors produced by the FFT operation are negligible in most scenarios (Gentleman and Sande, 1966). There are two approaches to performing an FFT,

the most common of which is known as decimation in time, the other being decimation in frequency (Ramirez 1985). The main difference between both approaches is in their organisation and processing order of data. Decimation in time methods firstly separate odd and even indices of the input data set for calculation, whilst decimation in frequency methods handle half the data set first using progressively smaller DFTs, before computing the remaining half. Whilst both techniques manage their input data differently, the strategy of the FFT is exemplified in both decimation of time and frequency; that is to break the long data set down into 2-point calculations that can be processed easily by the DFT to avoid lengthy computation times. It must be stressed that there are countless approaches to the FFT, each best suited for a different (and possibly obscure) application based upon these two approaches to decimation.

## 3.i. Analog to digital conversion

The number of calculations required of the Fourier transform restricts it to being used practically solely on computers, so any waveform to be transformed must be provided in the form of a series of sampled points along its period. As can be expected of any analog to digital conversion procedure, there will be issues of noise and distortion to contend with due to data quantisation (Bennett, 1948). Crucially, the sample rate of the data, or the number of samples that are taken from the analog waveform during the length of the sampling period, must be appropriate for a musical signal. When a waveform is sampled, its amplitude is recorded at points spaced `s = t / r` along the time axis, where `s` is the spacing between points, `t` is the length of the sample in seconds and `r` is the sample rate. Lower sample rates mean fewer calculations must be performed by the Fourier transform as there is less data to process, although the highest frequency that can be resolved from the data produced will be artificially low in accordance with Nyquist sampling theorem (Nyquist 1928, Shannon 1949). It is generally accepted that the upper threshold of human hearing is around 20kHz (Plack et al. 2005). In musical signals, the sample rate is usually increased to resolve up to

this bandwidth. This comes at the expense of computational power, although raising it past a certain point (above 40kHz) could be redundant as the resolvable band width exceeds the frequency range of human hearing. The length of the sample must also be considered - whilst sampling at 44.1kHz for 0.5ms in an application may be acceptable, it would be too much to compute if the sample was 500ms long and the results were required quickly. When choosing a sampling rate, it is therefore necessary for the developer to balance the computational resources available to them with the highest sampling fidelity affordable to allow for workable frequency range.

## 3.ii. Effects of periodicity

In part II of this paper, the Fourier transform will be used to gather spectral data from sampled electric bass guitar signals for the purpose of interpreting timbral information (refer to the discussion on timbre in chapter 2). Therefore, in this application, there will be ever-changing, non-periodic signals to process. Regardless, the Fourier transform interprets the provided data set as being periodic - contrary to the data that must be transformed. The question of periodicity is therefore more complicated than what is suggested by its definition and sometimes one has little choice as to how the input data will be handled in an application. Ramirez (1985) illustrates the importance of periodicity by using a sine wave oscillator as his example:

> ‘When we turn on the oscillator and look at its output with an oscilloscope, we see something that certainly looks like a sine wave and keeps repeating itself in a periodic fashion. And when the oscillator is turned off, the output ceases. ... We have a circuit that generates a periodic waveform, a sine wave. Right? Wrong! Not if we are going to stay with the purely mathematical definition of periodicity. We turned the oscillator on, watched the output repeat itself for a while, then turned it off. The oscillator’s output didn’t repeat itself over all time from minus infinity to plus

> infinity. … “But,” you might say, “theoretical definitions aside, it’s periodic as far as I’m concerned - at least for the time I looked at it.” And that’s a good point of view. It’s a practical point of view.’

Ramirez’s observations on periodicity in this scenario makes perfect sense. His sine wave generator can be safely assumed to produce a waveform shape at a given amplitude and frequency consistently for as long as it remains powered. However, the sound produced by a bass guitar cannot be assumed to share this property. The essence of real-time timbral detection implies that the target waveform must be ever-changing in its composition, each cycle differing in spectral content. If the waveform were to be treated as if it is non-periodic, timbral detection becomes impossible in real-time because the whole waveform must be analysed from start to finish. The system being developed then changes in nature to become a recording analysis tool, a treatment to be applied after the moment the music is made and, therefore, finds little use in the development of a performable musical instrument such as the one detailed in chapter 1. Even if this long, non-periodic sample is Fourier transformed, it will still be treated as periodic by the function and the consequences of this assumed behaviour will still occur.

Spectral leakage is caused by a non-integer number of waveform cycles being sampled and provided to the Fourier transform (Harris, 1978). Unless the waveform phase is synchronised with the window position and range (that is, the window has captured a whole number of waveform cycles) then it will likely contain discontinuities caused by points at both ends of the time axis, each with different amplitudes, meeting as the cycle repeats.


*Figure 3.1 - a rendering of a discontinuity in a signal. The ripple at the peak of the wave is the Gibb's Phenomenon overshoot (Hewitt and Hewitt, 1979).*

## 3.iii. Windowing

One way of handling spectral leakage is by windowing the sampled data (Harris 1978). We have in fact already windowed the waveform once as data was sampled. The window we applied was rectangular in shape, as the waveform appears described in time with no tapering at either end. A specialised window function such as the Hamming or Hann window may be used to treat apparent spectral leakage. Window functions are applied to a sampled waveform by multiplying each point in the signal by the corresponding value returned by the window function. Values are tapered towards zero at either end of the time axis to reduce the impression of discontinuities (Smith III, 2010a). Like most other functions explored so far, there is no ideal windowing function. Windows fundamentally alter the shape of the target waveform, leaving its impression on the spectrum and introducing its own distortion whilst eliminating that caused by discontinuities. Even a rectangularly-windowed waveform, one that is merely sampled in time and left untreated by further windowing functions, introduces spectral distortion. The relative distortion is visible as lobes and side-

lobes when the windowing function is represented in the frequency domain (Smith III, 2010b).

Other trade-offs must be considered when selecting a windowing function. There will be some further degradation of the transformed data depending on the length of the tapers on either side of the window. Should the windowing function be too extreme, and the overall waveform be weighted towards zero to an excessive degree, the loss in data resolution may outweigh the damage caused by the spectral leakage it is intent on preventing (a problem that becomes quickly apparent if the sample rate is especially low). Salvatore and Trotta (1988) detail the results of the flat-top window on a pulse wave in their study, illustrating its effects on complex signals and noting its ability to prevent data degradation due to its wide main lobe. For the purposes of this research, the Hann window will be used to mitigate discontinuities; its mathematical expression is given below (eq. 3.2).

$$w_0\left[x\right] \triangleq \left\{ \begin{array}{c} \frac{1}{2}\left(1+\cos\left(\frac{2\pi x}{L}\right)\right)=\cos^2\left(\frac{\pi x}{L}\right), \left|x\right| \le \frac{L}{2} \\ 0, \left|x\right| > \frac{L}{2} \end{array} \right\}.$$

*Equation 3.2 – the Hann window (Harris 1978).*

# 4. Digital filter design

When we talk of filtering a waveform, we are referring to the process of modifying an input signal to produce an output more suited for the application it is intended for (Rader and Gold, 1967). This may involve the restriction of the frequency spectrum range to eliminate undesirable frequency content in the signal (Butterworth 1930), or to boost or attenuate the amplitude of a range of frequencies in the case of an equaliser (Massenburg 1972). One result of this action that is sometimes overlooked in filtering and equalisation is the manipulation of timbre, an expected occurrence whenever the frequency components in a signal are affected by some means.

There has already been plenty of discussion on the concept of a ‘matching filter’ in this paper, one that reacts to incoming spectral information by analysing the amplitude of the source frequency components. For illustrative purposes, matching filters may be more intuitively thought of as a method of equalisation, rather than a simple filter with a defined bandstop where frequencies above or below a threshold may be attenuated. To understand how the matching filter works, it is helpful to personify it as a studio technician, one who prefers a mix sounding a certain way based on a song they heard. The technician sets the equaliser parameters accordingly based on what they consider to be ideal. In the real matching filter system, the song heard by the technician is a musical signal and their thought processes leading to their preference is replaced by algorithmic calculation of an ideal filter curve based on its spectrum. Should a matching filter be applied to a playable musical instrument, a crucial aspect must be that it is functional in real-time; in Perez-Gonzalez and Reiss (2009) such an equalizer was proposed. A finite impulse response (FIR) filter was designed that estimates an ideal curve of the magnitude spectrum of a source musical signal. The filter is applied to a target signal in real-time with acceptable latency for musical performance purposes, whilst maintaining the average loudness of the pre-equalised signal. Their findings were that the system was most accurate in the upper frequency ranges and lost accuracy as the frequency decreased. The method was expanded upon by Ma, Reiss

and Black (2013) to improve on the method's accuracy, now using an infinite impulse response (IIR) filter.

Filters described in this section are linear and time-invariant (LTI). Linear filters introduce no additional sinusoids into the spectrum by means of distortion or modulation (Smith III 2007a). A final point to include is on the principle of convolution in the time domain affecting the spectra of a signal in the frequency domain. If signals `x[n]` and `y[n]` are convolved in the time domain, the spectrum of the resulting signal `S[f]` will be the product of spectrum `X[f]` multiplied by `Y[f]` (Zölzer and Dutilleux 2002). Crucially, convolution in the time domain is equivalent to multiplication in the frequency domain, and vice versa (Oppenheim et al. 1999).

## 4.i. FIR and IIR digital filter designs

As suggested, digital filters may be finite and infinite impulse response in design. Each approach exhibits properties that differentiate one from another which may be exploited to suit a specific purpose. Filter accuracy improves by increasing the filter order as the equation has more memory of the signal to refer to in cascaded buffers, allowing for more accurate prediction of the filter outcome (Oppenheim et al. 1999). Ideal filters cannot be created in the real world as it would require an infinitely long impulse response to remove all frequency components above the stopband, therefore requiring an infinitely long buffer (Smith III 2007b). Instead, an ideal filter is approximated from a reasonable filter order with the trade-off being calculation time against filter efficiency.

$$y[n] = a[0]x[n] + a[1]x[n-1] + \ldots a[N]x[n-N]$$

$$y[n] = \sum_{i=0}^{N} a[i]x[n-i]$$

*Equation 4.1 – FIR filter represented as a list of terms and simplified.*

FIR filters are less common in design for numerous reasons. Analog IIR filters are considerably easier to implement than their FIR counterparts, largely restricting their functionality to the digital domain, but more crucially a large filter order is required to achieve results close to the ideal filter (Rader and Gold 1967, Tabassum, Amin and Islam 2016).

The FIR design can be transformed into an IIR filter by causing the impulse response to continue indefinitely (Oppenheim et al. 1999). IIR filters are not linear-phase; the filtered signal will be shifted forward in time. A practical method of negating phase distortion is by reversing the filtered signal and passing it through the filter again, undoing the phase offset produced by the first iteration of the filter by shifting the reversed samples back in time by the same degree (Kormylo and Jain, 1974). Other qualities produced by this behaviour are the effective doubling of the filter order as specified by its coefficients and the squaring of the magnitude response of the filter transfer function.

$$\sum_{i=0}^{N} a[i]x[n-1] = \sum_{j=0}^{M} b[j]y[n-j]$$

*Equation 4.2 – IIR filter expressed as a difference equation.*

It is left to define how these filter coefficients are produced. Approximating fitting the frequency response of the filter to an arbitrary curve in the time domain is made possible using an autoregressive moving average (ARMA) model (Friedlander and Porat 1984). ARMA models themselves are comprised of an autoregressive (AR) part and a moving average (MA) part that can be solved for the filter denominator and numerator respectively.

## AR and MA models

Moving average models are used to predict future values for a series `y[n]` purely from past values of a provided dataset `x[n]` (Wold, 1938). Eq. 4.3 represents the moving average model `y[n]` where `ω[n]` is Gaussian white noise representing stochastic error terms with a history of length `q` and a mean distribution of zero. White noise is used to

represent a dataset of random variables, stimulating different responses from the process as `i` increments (Friedlander and Porat, 1984). Its similarity to the FIR filter should be also be evident; the function is essentially a FIR filter applied to stochastic noise. MA models may be notated as MA(q), where q implies the order of the model.

$$y[n] = \sum_{i=0}^{q} b[i]\omega[n-i]$$

*Equation 4.3 – moving average model. Note its similarity to 4.1, the FIR filter, which is being applied to white noise* `ω[n]`.

Autoregressive models are also predictive functions that serve a similar purpose to their moving average counterparts. Just as MA processes are FIR filters by design, AR processes are a specialised type of IIR filter. Functionally, the AR and MA models reflect the differences between FIR and IIR filter designs. This fundamental difference can be exemplified if a 'shock' or sudden peak is supplied somewhere in the time domain data. The MA model has a recollection of a number of samples corresponding the length its order q so a shock will be 'forgotten' if it is at least q+1 samples from the present sample n. AR models refer to all samples provided to the system from zero time, so a shock anywhere in the dataset will have repercussions, great or small, for any value estimated in the future. It can be expected that shocks far enough back in time will affect future values so little that their effect can be almost indistinguishable but never zero. Like MA models, AR models can be specified as AR(p), where p refers to the order of the system. The autoregressive model is given in eq. 4.4, displaying the effective IIR filter applied to the noise error terms fulfilling the same criteria established previously.

$$y[n] + \sum_{i=1}^{p} a[i]x[n-i] = \omega[t]$$

*Formula 4.4 – autoregressive model used to produce white noise from an input sequence.*

## ARMA models and Yule-Walker equations

The ARMA model provides a powerful means of estimating coefficients (`b`, `a`) in the time domain from a limited dataset. Predictions of coefficients `a` and `b` can arise from solving modified Yule-Walker equations. The MATLAB function `yulewalk` is used for filter curve estimation from coefficients in later in this paper. This function is an implementation of a Yule-Walker method of ARMA spectral estimation proposed by Friedlander and Porat (1984). Eq. 4.5 portrays an ARMA process `y[n]` with orders of (`p`, `q`) where $p \geq q$. `ω[n]` represents the stochastic white noise process described previously. The autoregressive part is to the left of the addition sign.

$$y[n] = -\sum_{i=1}^{p} a[i]y[n-i] + \sum_{i=0}^{q} b[i]\omega[n-i]$$

*Equation 4.5 – ARMA process. The autoregressive and moving average parts can be seen on the left and right respectively. These parts resemble equations 4.3 and 4.4.*

Modified Yule-Walker (MYW) equations are often solved using a technique proposed by Prony (1795) to estimate coefficients (`b`, `a`) from a dataset of linearly spaced samples. Friedlander and Porat (1984) note that Prony's method is preferred for its direct handling of the dataset rather than sample correlation coefficients and that it can be adapted to produce sets of overdetermined equations. Mehra (1971) notes that using highly overdetermined sets yields more accurate estimations of coefficients. This property makes it a computationally efficient function at the potential cost of data accuracy. It is assumed that the condition number of the sample covariance matrix is equal to that of the data matrix squared. Should this requirement be unfulfilled, the quality of AR coefficient estimations will be significantly reduced.

### Exponential moving average filters

A final method of filtration is needed to smooth the produced filter curves in the whole matching filter system. Moving average filters, previously used in conjunction with an autoregressive function, are frequently used to smooth datasets of time-domain information, effectively reducing the apparent variance of the signal (Wold, 1938). Moving average filters may be also be applied in the frequency domain, as the smoothing process is effective on any data series that can be represented as points on a plane. Unlike the FIR moving average filter, the exponential moving average (EMA) filter described here is an IIR system with an order of one (Ma et al. 2013). Much like the AR function, the characteristic impulse response produced by the system is caused by the function referencing the product x to the right of the equals sign. The effect on a time-domain signal is analogous to that of a lowpass filter, smoothing out jagged ripples in the waveform which correspond to high frequency content in the frequency domain. In the frequency domain, where the points in the dataset represent the magnitude of the filter curve at a given frequency, the effect of the EMA filter applied to points within a single frame is a reduction in resonances caused by sudden peaks in the filter curve. Eq 4.6 is an example of an EMA filter as used in development of a matching filter system (Ma et al. 2013).

$$\alpha = e^{-1/(\tau \cdot fs)}$$

$$Y[n] = \alpha \cdot Y[n-1] + (1-\alpha) \cdot X[n]$$

*Equation 4.6 – exponential moving average filter (bottom) applied to a signal* `Y(n)`. *For calculating the value of* α, `e` *equals Euler's constant,* `fs` *equals signal sample rate and* τ *is a user-adjustable parameter to control the degree of smoothing.*

## 4.ii. Discussion on matching filters

As would be expected, the audible effects of the matching filter become more pronounced the further the estimated filter curve deviates from a flat line. Hence, filter

performance improves with filter order as the information provided to the ARMA process is allowed more historic information from the time-series sequence. A more general factor affecting filter performance is the structure of frequency components in the matched signal. Complex spectra consisting of partials dispersed across the frequency range will invariably result in smooth ideal curves as the spectrum concerned will resemble some sort of noise process. Consider the dataset provided to the matching filter implemented by Ma et al. (2013). Spectra were obtained from hundreds of commercial musical recordings from the UK and United States music charts over a span of decades and combined into one to compute the ideal filter curve. The combination of spectra from highly processed musical signals, themselves comprising of multiple musical instruments with their own partial structures, provides a great deal of data to the matching filter for the estimation of the ideal filter curve. In the application discussed here, where the matching filter is applied solely to the output signal of a bass guitar, it can be assumed that the performance of the filter will not be as accurate as the one implemented by Ma et al. The spectral information that can be gathered from a frame of audio from the bass guitar is markedly limited in comparison with far fewer sinusoidal components and providing less opportunity for deviation by extension. Ultimately, the research presented in this paper concerns the performance of such a filter on a particularly limited data source.

One further property that is presented by the matching filter presented here is its adaptation to the changing bass guitar signal. For the filter to function effectively, it should not only produce an accurate filter curve for each frame analysed but the curve of each frame should be congruent with the last. This factor is vital for the outcome signal to retain its musical qualities and sound like a convolution of signal timbres as a result. Should the ideal curve produced for each frame sound entirely independent from the last, the applied filter will sound “step-like” over the course of several frames and introduce undesirable non-musical traits to the signal. The dependent factors here are the magnitude and rate of spectral change over time and the effectiveness of the FFT to represent the spectrum accurately. Given a high rate of change for the ideal curve over time and the periodic

assumptions made by the FFT, the likelihood of false frequency components being reported by the transform is relatively high even with adequate windowing of the signal. Furthermore, little can be done to smooth the produced filter curve between frames excessively as smoothing inherently reduces the accuracy of the curve for the frame in question, deviating from the ideal curve of the modifier signal.

# Part II – A matching filter and envelope system

Discussed here is the development, implementation and evaluation of a matching filter and envelope system based upon principles and techniques discussed in part I.

## Method of research

Recordings were taken of a bass guitar to function as the various source signal states throughout testing. Single notes were recorded at pitches of C2 and C3 (where $f^0$ equals approximately 65.4Hz and 110.8Hz respectively). Notes were performed using numerous techniques, the details of which were outlined in chapter 2. These were fingerstyle, picked (using a plectrum), thumbing, slapping and popping. Additional samples were recorded of one octave G-major scales on the bass guitar, with root notes of G2 and G3 to assess system performance on signals consisting of multiple notes. Scales were recorded at 120bpm, with each note lasting one crotchet beat (equating to notes of 0.25 seconds in length). The bass guitar was recorded at a sample rate of 48kHz and a bit depth of 24. The signal was sent from the bass guitar pickups to a DI box, then into an audio interface for capture.

Corresponding samples were produced of an acoustic piano and a synthesized bass pluck to serve as the modifier signals, matching the pitch and duration of the bass guitar counterparts. Acoustic piano samples were taken from a Native Instruments Kontakt library. Although the samples were triggered using MIDI, the sound produced by the digital instrument relies on recordings of a physical grand piano. All post-processing available in the library was disabled so only the raw recordings of the piano were triggered. Kontakt was used to produce the piano modifier signals for several reasons. Crucially, facilities were unavailable to record an adequate piano signal independently at the time of research.

Moreover, having MIDI control over triggering the piano samples allowed for precision; each note is aligned to a grid spacing them evenly apart, with dynamics consistent between notes. It is not expected that the use of a sample library will have an adverse effect on the results of signal processing as the instrument in question is in no way synthesized. Instead, a 'best case' scenario is presented, where each note is performed perfectly on beat.

The synth pluck was created using XFer Serum, a VSTi synthesizer loaded inside Ableton Live. Fig. 5.1 depicts a screenshot of the synthesizer interface; the settings used for the synth patch are displayed. The patch is built using single square wave oscillator, modified by a lowpass filter with a low base cutoff frequency. Filter cutoff and resonance is modulated over the decay of the note to fall towards the base levels seen in fig. II.1. The acoustic result is a transient burst of high frequency content at note onsets, which is quickly muffled into a low sub-bass rumble as the note progresses. Note loudness is also modulated to fall to zero over time, following a similar trajectory to the filter cutoff and resonance envelope. Like the piano signals, no post-processing or effects were used to produce the synth plucks.


*Figure II.1 – Serum interface displaying the synth patch used for the bass pluck samples. Filter, waveform and loudness envelope settings can be seen in the top right, top left and bottom left areas respectively.*

The synth and piano signals were then edited to be the same length as the bass guitar signals. As the bass guitar was performed live in contrast to the MIDI triggering of the piano and synth notes, the signal had to be edited slightly to align each note onset with that of the modifiers. It was predicted that the signal processing chain employed in the timbral blending system would be sensitive to note onsets. In particular, the envelope matching function relies on source and modifier note onsets to be aligned for an accurate replication of modifier note envelopes to be produced in the processed signal. Editing was accomplished using Ableton Live to time-stretch the bass guitar notes into place. The time stretching algorithm was set to preserve note transients and notes were aligned to the grid. In the process, it was observed that some notes were shorter than a crotchet beat. Rather than

estimate new data to fill the remaining length of the note, the time stretching algorithm was set to fade to silence after the end of existing data. This was done to avoid any additional signal processing introduced by data interpolation that could be harmful to the performance of the system. All samples were finally exported at a sample rate of 48kHz and a bit depth of 24.

Results were gathered by loading the appropriate signals required for a test into the system and assessing the acoustic outcome of the processed signal. Initially, results were obtained using a set of default values for user adjustable parameters. These default values were found to be roughly appropriate for most musical signals. From there, parameters were altered and the outcome noted. This process was repeated for each adjustable setting until it was decided that the acoustic result could not be improved. Finally, the produced signal was analysed and presented in the following chapter.

# 5. MATLAB implementation

To review, the system being implemented is a dynamic filter designed to match an ideal curve obtained from magnitude spectra alongside a time-domain envelope matching function. The filter is applied to a source signal to attenuate frequency components in the source spectrum to shape it to fit that of the modifier. Reshaping the signal in the time domain is necessary to counter the distortion introduced to the waveform by the filter and to reproduce the envelopes of notes in the modifier signal. The techniques used for this implementation are intended to be transferrable to a real-time system with some slight modifications, however the prototype presented here will function offline to assess the viability of the system first. The timbral blending system was implemented as a MATLAB application to calculate the relevant coefficients for filter and time-domain envelope matching, apply these functions using the variables generated and produce charts for visualisation and analysis.

This section concerns the implementation and rationale behind signal processing techniques employed for the matching filter progress with appropriate code from the system provided for context. An example screenshot of the application interface can be seen in fig. 5.1. The data on the left-side of the application window can be changed between several layouts demonstrating the filter attributes, signal waveforms and spectral information as desired. The right side of the interface is for specification of variables to tailor the filter performance according to the signals loaded into the program. Changing these variables is necessary to attain optimal filter performance for different signals; the precise effect of these variables on the filter architecture will be detailed in the following section concerning filter results. Signal properties, such as sample rate and total number of samples are also displayed on the right-hand side of the window. It is assumed that all signals loaded into the system have a minimum sample rate of 44.1kHz to cover the whole spectrum of human hearing. Furthermore, the source and modifier signals must be the same length as the matching filter system was designed with no predictive functionality to estimate the spectral

and amplitude envelopes of signals into the future. A final requirement is that the signals loaded are normalised so no sample exceeds 1 or -1. Ensuring the signals loaded occupy as much dynamic range as possible without clipping provides a standard for comparison and parameter estimation so more accurate results can be produced from the system.

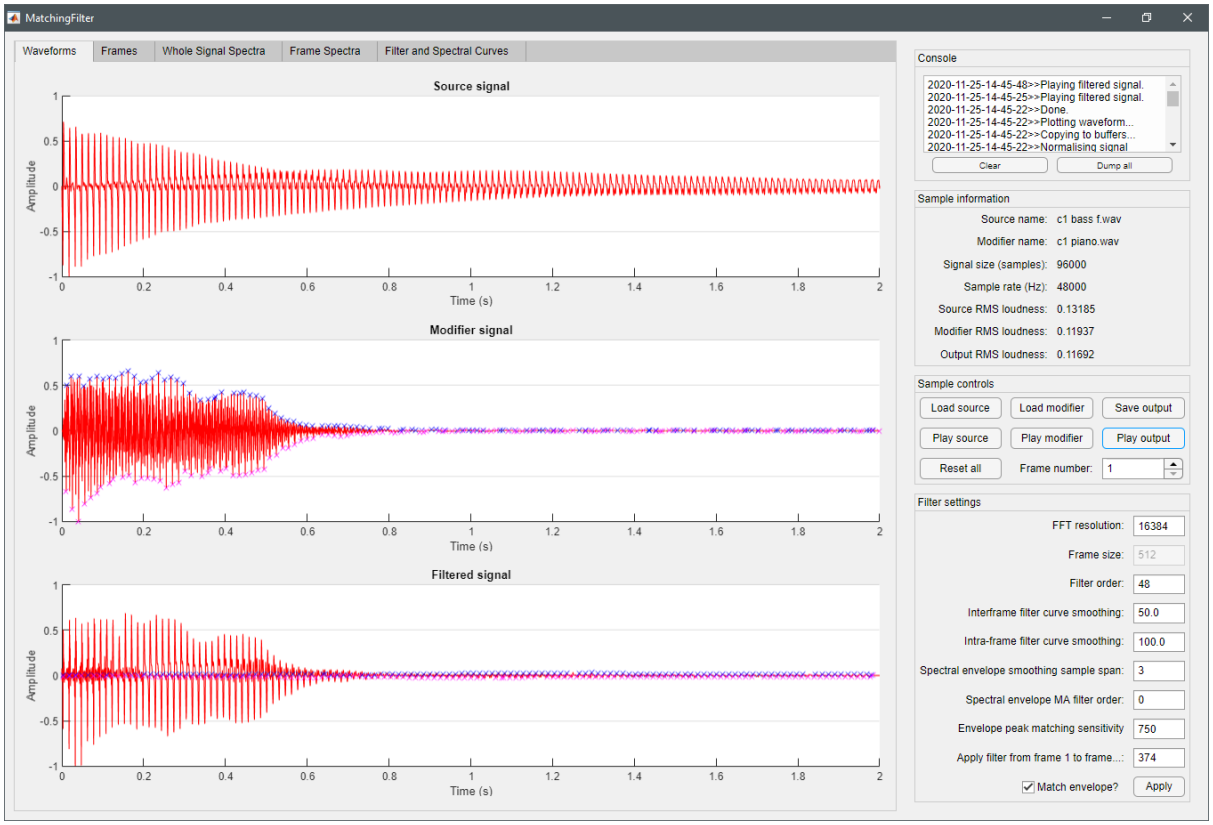


*Figure 5.1 – application waveform view. The source and modifier signals (top and middle) signals were loaded prior to signal processing. The outcome signal is displayed on the bottom. User controls and signal information are located in the right-hand panel. Signal envelopes are traced in blue and magenta crosses for the positive and negative parts of the signal respectively. The envelope reported for the filtered signal is before time-domain envelope matching.*

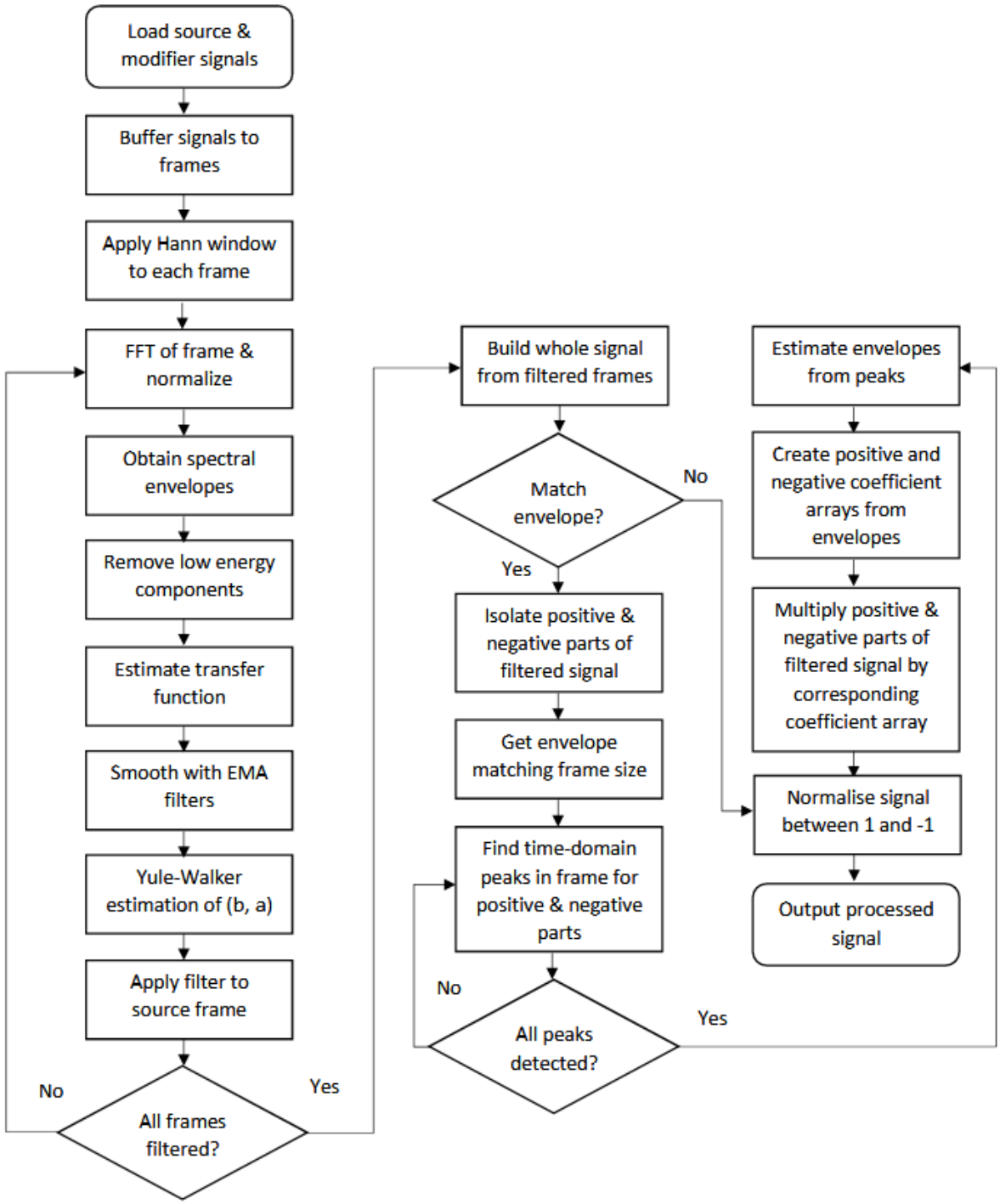


*Figure 5.2 – Simplified flow diagram representing signal processing in the matching filter and envelope system.*

## Loading signals

Signals are processed by the system using a frame-by-frame approach. As such, signals must be buffered into matrices with each frame consisting of a constant number of

samples `n`. Each frame is then windowed using an `n`-point Hann function to avoid spectral leakage in Fourier transforms. Amplitude peaks are also produced for the modifier signal when it is loaded; this function and its purpose will be described later in this section.

Signals are loaded and the right channel is discarded using the following MATLAB commands. Global variables for sample rate (`app.Fs`) and signal length in samples (`app.signalSize`) are also set, where `sig` equals the loaded source or modifier signal.

```
% load signals
[sig, app.Fs] = audioread(loc);
app.signalSize = length(sig);
sig = sig(:,1);
```

A single-channel signal is now stored in memory which can now be split into frames and buffered into global variables for processing, accessible by all functions in the program. The provided code executes when the modifier signal is being loaded. The loaded signal `sig` is stored in a global variable `app.modSignal`, which is then buffered into frames using the `buffer` command. This operation takes an input signal and a frame size in samples (defined by the user) and proceeds to portion the signal into frames of that size using a sliding window. An overlap of `n/2` is also specified, meaning sample 1 to `n/2` of each frame corresponds to samples `n/2+1` to `n` of the preceding frame. Global variable `app.numFrames` is set to equal the size of the second dimension of the frame matrix, which is equal to the signal length in samples divided by `n`, rounded up. The source signal is loaded in a similar way to the modifier; instances of `app.modSignal` are replaced by `app.srcSignal`, and `app.modFrame` is replaced with `app.srcFrame`.

```
% buffer
app.modSignal = sig;
n = app.frameSize.Value;
app.modFrame = buffer(app.modSignal,n,n/2,'nodelay');
buf = app.modFrame;
app.numFrames = size(buf,2);
```

Each frame can now be windowed using the n-point Hann function and stored in a separate set of buffers. Similarly, the code below is executed when the modifier signal is loaded and related global variables can be substituted for their source signal equivalents.

```
% window
w = hann(n);
for i = 1 : app.numFrames
      for j = 1 : app.frameSize.Value
            app.modWinFrame(j,i) = w(j) * app.modFrame(j,i);
      end
end
```

## Matching filter

Once the source and modifier signals are loaded into the system and buffered as above, the matching filter process can be executed. As previously discussed, the filter operates on a framewise basis, therefore the following procedure is applied to each frame successively. Firstly, the RMS loudness of the source and modifier frames is taken. If either returns an RMS value below 0.0001 the processed frame is assumed to be silent and the filter coefficients from the previous frame are reapplied to the active frame. This precaution was taken to avoid dividing by zero errors further on in the filtering process. A similar approach using a Hysteresis noise gate was utilised by Ma et al. (2013) for the real-time implementation of their matching filter system. Ultimately, by removing silent frames from processing, unnecessary spectral artefacts are mitigated and computing time is improved at virtually no acoustic cost.

Assuming the active frame is not silent, the filter will then obtain the normalised magnitude spectra of the source and modifier frames. This is done using an m-point FFT, where m is a user specified variable denoting the resolution of the transformed signals. The absolute value is taken from the spectra to return real numbers for processing. Transformed frames are then divided by the frame length n to return actual magnitude values for the transformed spectra. Normalisation occurs by dividing the whole dataset by the highest

magnitude value of each respective signal. The result of this normalisation is two spectra, one for the source frame and one for the modifier, with similar apparent magnitudes. Spectral normalisation is important here as relative loudness for each frame can be effectively ignored, providing the means for accurately estimating the difference in spectra in the form of a transfer function. Lastly, the transformed spectra must be made single-sided, as the FFT also returns redundant negative frequency information, assumed to be symmetrical around zero. As the negative frequency information is a mirror image of its positive counterpart, it serves no purpose in this application and can be trimmed to reduce processing time. The variable `res` in the provided MATLAB code represents FFT resolution, and `frameSize` is the total number of sample points in each frame buffer.

```
% get magnitude spectra
cs = abs(fft(app.winFrame(:,no), res))/frameSize;
cs = cs(1:res/2);
cm = abs(fft(app.modWinFrame(:,no), res))/frameSize;
cm = cm(1:res/2);

% normalise magnitude spectra
maxs = max(cs);
maxm = max(cm);
for i = 1 : res/2
      cs(i) = cs(i) / maxs;
      cm(i) = cm(i) / maxm;
end
```

The spectral envelopes of the frames are then estimated from their respective magnitude spectra. Spectral envelopes are representative of the whole spectrum at that point in time and describe its general shape. Were the raw spectra to be used for filter curve estimation as detailed in this section it can be assumed that the curve would be a poor representation of its ideal trajectory. Resonances and scalloping can interfere with transfer function estimation as frequency information is misrepresented by the FFT. These resonances can be removed from the spectra by detecting individual meaningful peaks in the signal and interpolating between the identified points in frequency space, producing a curve fitting the shape of the magnitude spectra (Zhivomirov, 2020). Piecewise cubic

Hermite interpolation polynomial (or PCHIP) smoothing was used for interpolation as the method avoids overshoots when fitting a non-oscillatory curve, preventing destructive interpolation of the spectrum. Should an inadequate number of peaks be detected for interpolation (the minimum number required being 2), the system will use the spectral envelope produced for the previous frame, or a set of zeroes if there is no other envelope that can be used in its place. The spectral envelopes are optionally smoothed using a moving average filter with a sample span of `s1` and an order of `s2`. Spectral envelopes are normalised within the range [0, 1] so the weakest and strongest magnitude responses for each spectrum are balanced. Normalising the spectral envelopes in this way accounts for rapid fluctuations in apparent signal energy caused by framewise processing.

```
% spectral envelope extraction
% adapted from Zhivomirov (2020)
f = linspace(0,app.Fs/2,res/2);
[pksCs, locsCs] = findpeaks(cs);
```

```
[pksCm, locsCm] = findpeaks(cm);
fpksCs = (locsCs-1)*(f(2) - f(1));
fpksCm = (locsCm-1)*(f(2) - f(1));


% use env of previous frame or zeros if not enough peaks
if length(pksCs) < 2
	if no == 1
		Cs = zeros(1,res/2);
	else
		Cs = app.envs{no-1};
	end
else
	Cs = interp1(fpksCs, pksCs, f, 'pchip');
	for i = 1 : s2
		Cs = smooth(Cs, s1);
	end
	Cs = rescale(Cs);
end

if length(pksCm) < 2
	if no == 1
		Cm = zeros(1,res/2);
	else
		Cm = app.envs{no-1};
	end
else
	Cm = interp1(fpksCm, pksCm, f, 'pchip');
	for i = 1 : s2
		Cm = smooth(Cm, s1);
	end
	Cm = rescale(Cm);
end
```

Lastly, low-energy components are removed from the spectral envelopes using a thresholding technique. These components are usually found in the high-frequency range of a spectrum and can be the product of signal noise. As the difference between these components can vary wildly between signals, erratic filter curves may be produced if they are not removed from the relevant spectra. By setting these values to zero for both the source and modifier spectral curves, the filter curve produced will level out, leading to some loss of transfer function accuracy with respect to the unmodified spectra. This loss of accuracy can be justified by the apparent reduction in signal distortion and consequently the more appealing acoustic properties of the final filtered signal.

```
% remove low energy components
for i = 1 : res/2
      if Cs(i) < 0.0001 && Cm(i) < 0.0001
            Cs(i) = realmin;
            Cm(i) = realmin;
      end
end
```

Transfer function estimation can now be performed. A function representing the difference between source and modifier spectral envelopes for every point in the matrices is returned. The MATLAB function `tfestimate` is given three arguments, with the first two comprising the modifier and source spectral envelopes, representing the transfer input and output signals respectively. A third argument f is also passed to the function. f is a linear scalar between 0 and 1 with `res/2` sample points acting as a normalised frequency vector. The transfer function is evaluated at each normalised frequency point, returning a dataset `txy` with a length of `res/2`. Absolute values are taken from the transfer function estimated to avoid processing complex numbers; the imaginary part returned from the transfer function estimation can be safely discarded.

```
% transfer function estimation
f = rescale(1:res/2)';
 [txy, ~] = tfestimate(Cm, Cs, [], [], f);
tf = abs(txy);
L = length(tf);
```

The transfer function itself (and by extension the produced filter curve) should then be smoothed, this time using exponential moving average (EMA) filter functions. Smoothing of the filter curve should occur to eliminate remaining apparent resonances that are reported by the transfer function and to provide a means to relate a predicted curve to the previous in its series. Hence, the curve is first smoothed internally to quash resonances before it is passed through the filter again to match the curve slightly to that of the preceding frame, the transfer functions of which are buffered to the cell array `app.Tf`. This intra-frame smoothing is required to avoid excess noise being introduced to the filtered signal. As each filter curve

produced is initially independent from the last, there can be a great deal of noise introduced by the filter if the curve for each frame varies wildly from the last. By smoothing the generated curve between frames, the filter becomes less responsive to sudden changes in spectral energy between frames but can also appear more sonically pleasing. Given the sensitivity of the EMA filters, variables `t1` and `t2` are left for the user to define to optimise filter performance according to the loaded signals. EMA filters perform more suitably here than the MA filter implemented for spectral envelope smoothing. Aside from being more computationally efficient, the EMA filter is self-referential and is receptive to shocks from the first supplied index onwards suggesting its ability to reliably predict future values.

```
% EMA filter
E = -1/t1;
al = exp(1)^E;

% apply EMA filter to transfer function within 1 active frame
for i = 2 : L
      tf(i) = al*tf(i-1)+(1-al)*tf(i);
end

% smooth overall filtering curves between frames
if no > 1
      E = -1/t2;
      al = exp(1)^E;
      tf = al*app.Tf{no-1}+(1-al)*tf;
end
```

Filter numerator coefficients `b` and denominator coefficients `a` can be estimated from the normalised transfer function `m` and the previously defined linear frequency vector f using modified Yule-Walker equations. The MATLAB function `yulewalk` estimates the denominator using modified Yule-Walker equations and the numerator from a least-squares fit of the calculated filter impulse response, an implementation of the methods proposed by Friedlander and Porat (1984). The final stage of the matching filter progress is applying the IIR filter to the active source frame using coefficients `a` and `b`. Filtering is implemented through the MATLAB `filtfilt` function to mitigate comb distortion in the final combined signal. Usually when filtering a signal frame-by-frame, the final filter conditions can be taken

as the initial conditions for the following frame, resulting in a continuous stream of points forming the complete signal. However, the filter curve in this matching filter system is dynamic, rendering the final filter conditions of the preceding frame largely meaningless to the active frame. Through experimentation through the development process, it was discovered that the shifting in time produced by typical IIR filtering resulted in significant amounts of distortion and comb artefacts after the whole signal was rebuilt from its overlapping frames. Therefore, an alternative system of filtering is required that leaves phase intact, which is typically shifted in time by the IIR filter. Phase can be preserved by reversing the filter outcome and passing it through the filter again, doubling the effective order of the filter and shifting the phase back in time, negating the initial shift. Comb artefacts are reduced considerably by using zero-phase filtering at the expense of filter accuracy. A continuous stream of filtered data buffered into frames cannot be returned by using the above technique as each filter curve produced from estimated coefficients is independent. There is no handling of final and initial filter conditions given the properties of dynamic filter curve generation, producing a discontinuous stream of data when reassembled frame by frame.

```
% yule-walker of transfer func
m = rescale(tf);
[b,a] = yulewalk(app.Order.Value,f,m);


% apply filter and copy to buffer
flt = filtfilt(b,a,app.frame(:,no));
for i = 1 : frameSize
      app.fltFrame(i,no) = flt(i);
end
```

## Note envelope matching

As filtered frames cannot be stitched neatly together to reassemble a whole continuous signal, another method of reassembly is required. Recall that signals were buffered so each frame overlaps the previous by half its length. By windowing each filtered

frame with the Hann function and piecing them together in an inverse manner to how the overlapping frames were buffered, a complete signal can be produced.

```
% assemble signal
w = hann(sz);
app.fltSignal = zeros(1,app.signalSize);
for i = 1 : lim
	% window
	for j = 1 : sz
		app.fltWinFrame(j,i) = w(j) * app.fltFrame(j,i);
	end
	for j = 1 : sz
		offset = (i-1)*(sz/2);
		if offset+j > app.signalSize
			break;
		 end
		if (i == 1 && j < sz/2+1) || (i == lim && j > sz/2)
			app.fltSignal(offset+j) = app.fltFrame(j,i);
		 else
			app.fltSignal(offset+j) = app.fltSignal(offset+j) +
		app.fltWinFrame(j,i);
		 end
	end
end
```

Time-domain envelope matching can now take place on the filtered signal. As will be detailed further in the results section of this paper, it was discovered that the amplitude envelope of a signal contributes significantly to its spectral characteristics. Should the notes constituting the assembled signal be left untreated in terms of their shape the filter will not appear to sufficiently blend the timbre of signals; in fact, the aggressive filter operations applied in the previous stage could render the signal unrecognisable and musically abstract. The time-domain envelope matching function attempts to rebuild note envelopes to reflect that of the modifier, manipulating the attack and decay of the filtered note as it proceeds through its lifespan. Zero crossings in the signal are preserved using the implemented method, allowing for great control over the envelope of the filtered signal with no opportunity for distortion to be introduced from a phenomenon such as amplitude modulation.

Firstly, its envelope must be estimated from local maxima detected over the length of the signal. A function was produced to return arrays containing signal peaks and their

locations in a provided signal. Signal envelope estimation is performed after signal reassembly in the filtering process, but also when the modifier signal is initially loaded to estimate the target envelope. Positive and negative signal envelopes are estimated independently, so four datasets are returned from the function for positive and negative peaks and locations respectively. The function first isolates each pole of the signal and stores the two divided signals in variables `pos` and `neg`. Peak value estimation is performed on a framewise basis, however there is no buffering of frames as in previous examples. Rather, the function searches for the highest value in the user defined sample span `peakVal`. The number of frames `nF` is calculated from the signal length and peakVal to loop through pos and neg to find the local maxima in the sliding window. If the portion of the signal is entirely silent, a value of 0 is returned with the corresponding location being `sig(((i-1)*peakVal)+1`, where sig represents either `pos` or `neg`. The code for the `getPeaks` function is provided below.

```
function [pksP, locsP, pksN, locsN] = getPeaks(sig)
% isolate positive and negative parts of signal
pos = sig;
for i = 1 : app.signalSize
      if pos(i) < 0
            pos(i) = 0;
      end
end
neg = sig;
for i = 1 : app.signalSize
```

```
        if neg(i) > 0
                neg(i) = 0;
        end
end

% get envelope matching frame size
peakVal = app.peak.Value;
nF = floor(app.signalSize/peakVal);
pksP = zeros(1,nF);
pksN = zeros(1,nF);
locsP = zeros(1,nF);
locsN = zeros(1,nF);

% find peaks in range
for i = 1 : nF
        [pksP(i), I] = max(pos(((i-1)*peakVal)+1:((i-1)*peakVal)+peakVal));
        locsP(i) = (i-1)*peakVal+I;
        [pksN(i), I] = max(abs(neg(((i-1)*peakVal)+1:((i-
1)*peakVal)+peakVal)));
        locsN(i) = (i-1)*peakVal+I;
end

% set return values
for count = 1 : 2
        if count == 1
                pks = pksP;
        else
                pks = pksN;
        end

        if count == 1
                pksP = pks;
        else
                pksN = -pks;
        end
end
end
```

Datasets containing an equal number of peak values and their locations across the signal have now been produced for the modifier signal and the filtered signal. The time-domain signal envelopes can now be estimated by interpolating between the positive and negative points reported by the `getPeaks` function, returning sets of peaks and locations that extend the length of the signal. PCHIP interpolation is used once again to avoid overshooting. Coefficients are then calculated for the positive and negative halves for every point in the signal from the envelopes representing the modifier and filtered signals (`difMP`, `difMN`, `difFP` and `difFN`). When each sample in the filtered signal is multiplied by the

corresponding coefficient from the relevant array for its pole, the waveform is reshaped to roughly fit the envelope of the modifier. Coefficients greater than 2 (or less than -2) are restrained to 2 (or -2) to prevent extreme spikes in amplitude caused by errors in the estimation process. The code for interpolating the peaks and location arrays, estimating and applying amplitude coefficients to the filtered signal is provided below.

```
% get envelopes of pos and neg parts of modifier and filtered signals
vec = 1:app.signalSize;
difMP = pchip(app.locsMP, app.pksMP, vec);
difMN = pchip(app.locsMN, app.pksMN, vec);
difFP = pchip(app.locsFP, app.pksFP, vec);
difFN = pchip(app.locsFN, app.pksFN, vec);

% estimate positive amplitude coefficients
coeffP = zeros(1,app.signalSize);
for i = 1 : app.signalSize
```

```
        coeffP(i) = 1 + (difMP(i) / difFP(i));
        if isnan(coeffP(i)) || isinf(coeffP(i))
                coeffP(i) = 0;
        end
end

% estimate negative amplitude coefficients
coeffN = zeros(1,app.signalSize);
for i = 1 : app.signalSize
        coeffN(i) = 1 + (difMN(i) / difFN(i));
        if isnan(coeffN(i)) || isinf(coeffN(i))
                coeffN(i) = 0;
        end
end

% match signal envelope
for i = 1 : app.signalSize
        if app.fltSignal(i) > 0
                coeff = coeffP(i);
        elseif app.fltSignal(i) < 0
                coeff = coeffN(i);
        end

        if app.fltSignal(i) * coeff > 2
                app.fltSignal(i) = 0;
        elseif app.fltSignal(i) * coeff < -2
                app.fltSignal(i) = 0;
        else
                app.fltSignal(i) = app.fltSignal(i) * coeff;
        end
end
```

The final stage in the signal processing chain is to normalise the filtered and reshaped waveform. The code below amplifies the signal around zero so its highest or lowest sample does not exceed 1 or -1 respectively. This stage is required as the input samples are expected to be normalised in the same manner, so the processed signal should abide by the same rules for meaningful comparisons between the signals to be made.

```
% maximise range without clipping
if abs(min(app.fltSignal)) > max(app.fltSignal)
        mx = abs(min(app.fltSignal));
        mn = min(app.fltSignal);
else
        mx = max(app.fltSignal);
        mn = -max(app.fltSignal);
end
for i = 1 : length(app.fltSignal)
        app.fltSignal(i) = 2 .* app.fltSignal(i) ./ (mx - mn);
```

```
end
```

At this point, the various plots and information panels around the application are updated to include data from the processed signal, which may also be exported as a .wav file for further investigation in external applications.

# 6. Analysis of results

## 6.i. Investigation into spectra of bass guitar signals

### General observations and trends in bass guitar signals

In chapter 2, various methods of performance on the bass guitar were detailed, each characterised by a distinctive timbral outcome. It is therefore necessary to investigate the spectral composure of signals produced using commonly utilised performance techniques to evaluate how they may be affected by the matching filter system. Fig. 6.1 illustrates the magnitude spectrum of a C2 note played on the bass guitar fingerstyle. It was previously established that fingerstyle performance produces a relatively soft, round timbre when compared to many other techniques typically utilised by a bassist, an observation that is reflected in the magnitude spectrum of the signal. Much of the spectral information of the signal is condensed into the low to middle range frequency bands whilst spectral information in the higher range is largely limited to noise-like partials produced by the note transient.

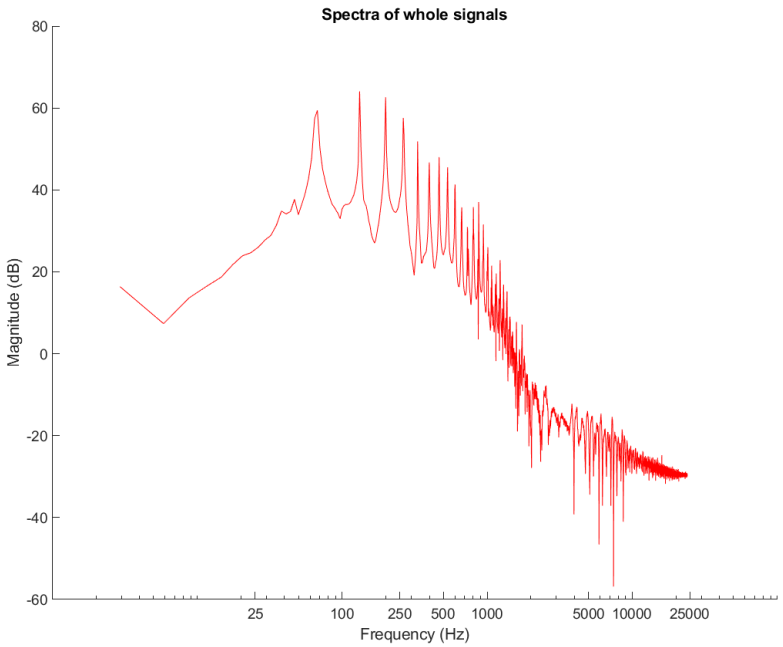


*Figure 6.1 – the magnitude spectrum a single C2 note played on the electric bass guitar using the finger-style performance technique.*

In contrast, plucking the strings of the bass guitar with a plectrum can be expected to produce a harder, cutting sound characterised by the striking medium being constructed of a tougher material (plastic rather than skin). Fig. 6.2 portrays the spectrum of the same note sounded using a plectrum rather than fingers. When compared to the spectrum of the note produced using the fingerstyle technique, differences can be observed in its partial structure that are responsible for its apparent difference in timbre. In the higher frequency ranges, the fingerstyle spectrum appears less erratic and noise-like in its construction, reflecting the relative softness of its timbre against the plectrum-sounded signal. Harmonic components also appear more pronounced and clearer with deeper troughs between them as can be seen in the lower frequency ranges. Again, this can be attributed to differences in striking mediums; a finger, being thicker than the plectrum, contacts the string for a longer period of time as the string rolls around its surface.

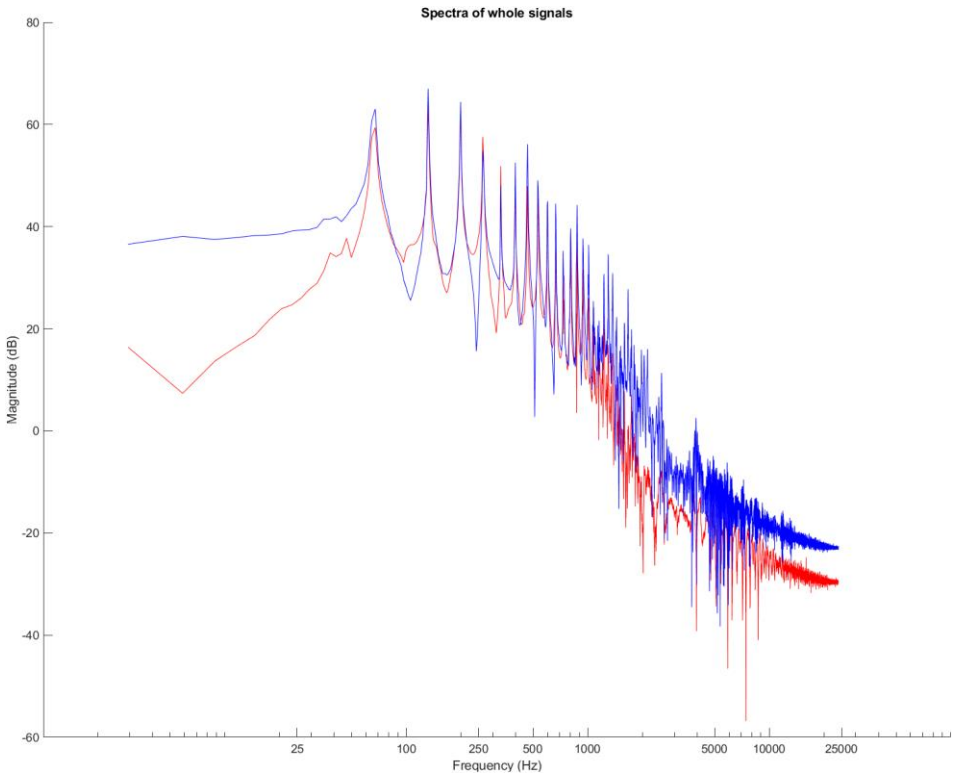


*Figure 6.2 – spectrum of a C2 note played using a plectrum (blue). The fingerstyle example from 6.1 is in red.*

Such an observation may also be made when the spectrum of a note sounded with the pad of the thumb as is shown in fig. 6.3. If the three performance styles introduced so far are compared directly on a scale of timbral “softness”, it can be surmised that fingerstyle playing falls between thumbing and using a plectrum. This observation is reinforced by the trends detailed in the spectra so far with regards to the sounding mediums used for each example. The thumb, when used to stroke down on the string, sounding the note using the flesh and knuckle on the digit, exaggerates the timbral roundness perceived when using the fingerstyle technique. This is due to the larger, softer pad on the thumb making even more contact with the string as the note is played, dampening the string. High frequencies reported in its spectra are far less pronounced.

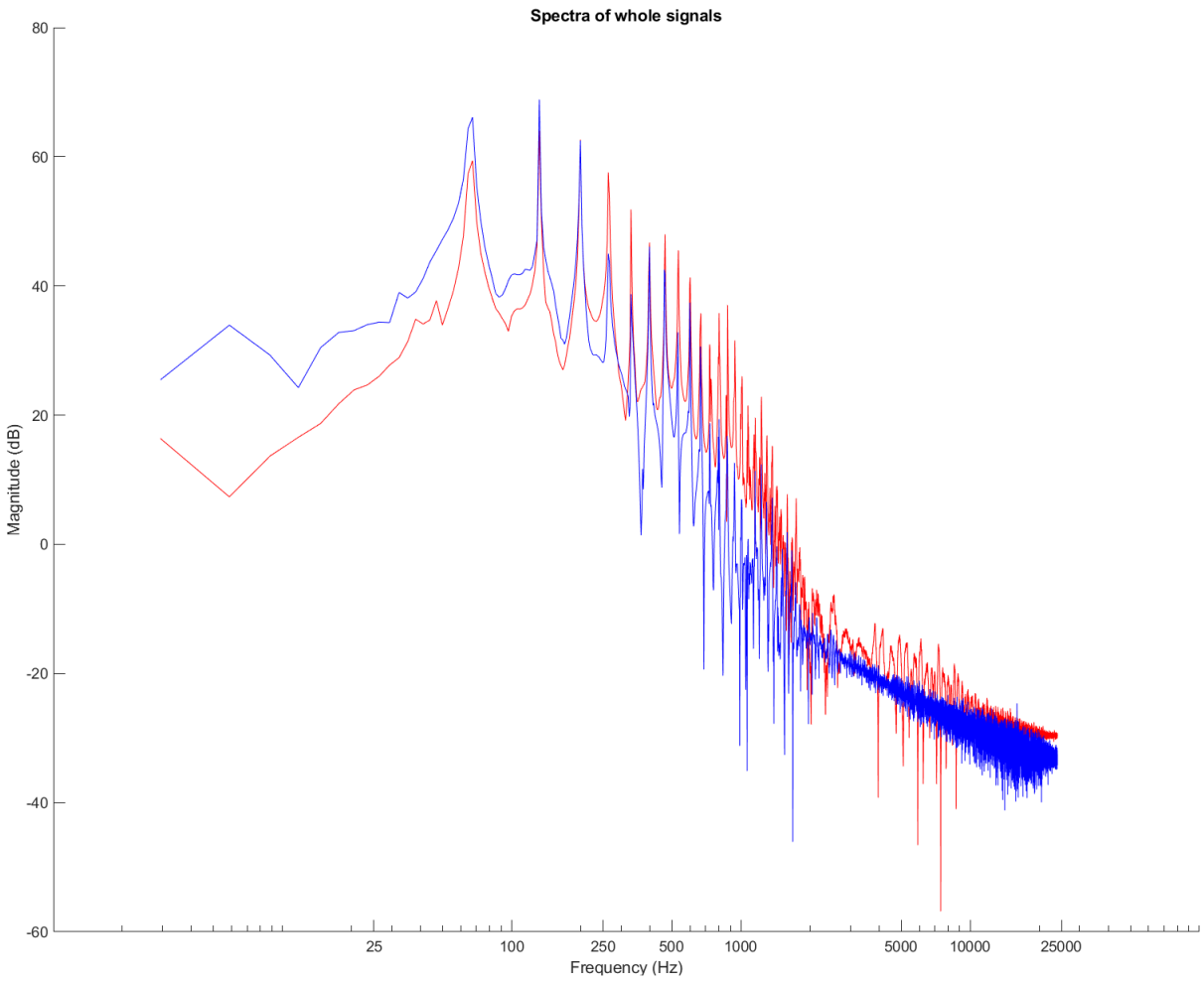


*Figure 6.3 – spectrum of a C2 note sounded using the thumb (blue). 6.1 is again pictured in red.*

Striking the string with the thumb in a percussive manner, known as slapping, produces a note with a timbre distinct from typical thumbing. When compared to fingerstyle playing, a higher concentration of partial components in the high frequency range are reported indicating the note transient, like the spectrum of the plectrum-sounded note (fig. 6.4). The lower frequency components are relatively high in magnitude, indicating a powerful note transient occupying the same frequency range one would typically expect of a kick or snare drum (fig. 6.5).

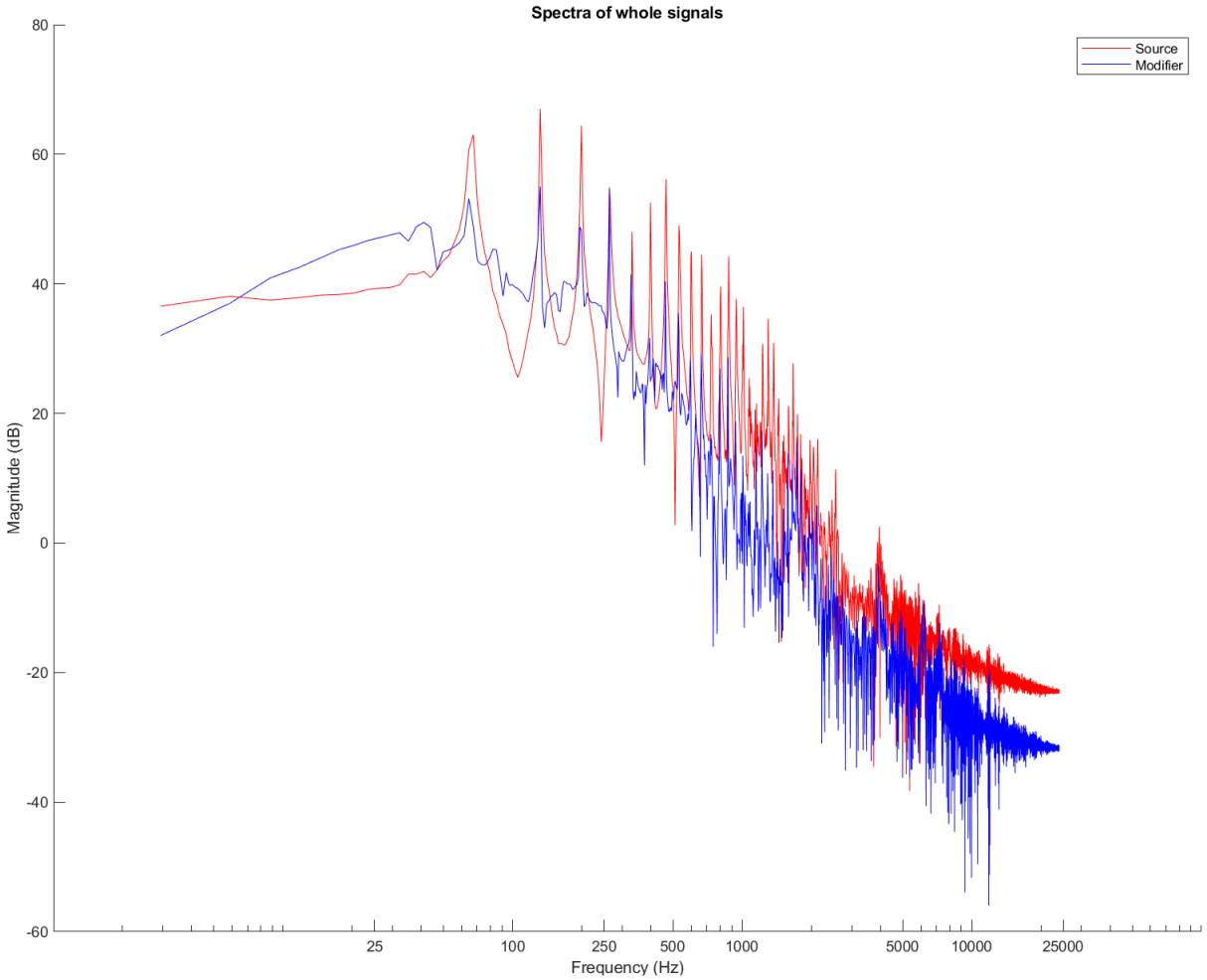


*Figure 6.4 – spectrum of a slapped C2 note, sounded by the thumb impacting on the string (blue). The spectra of 6.2 is pictured in red, showing a similar erratic partial structure in the high frequency band.*

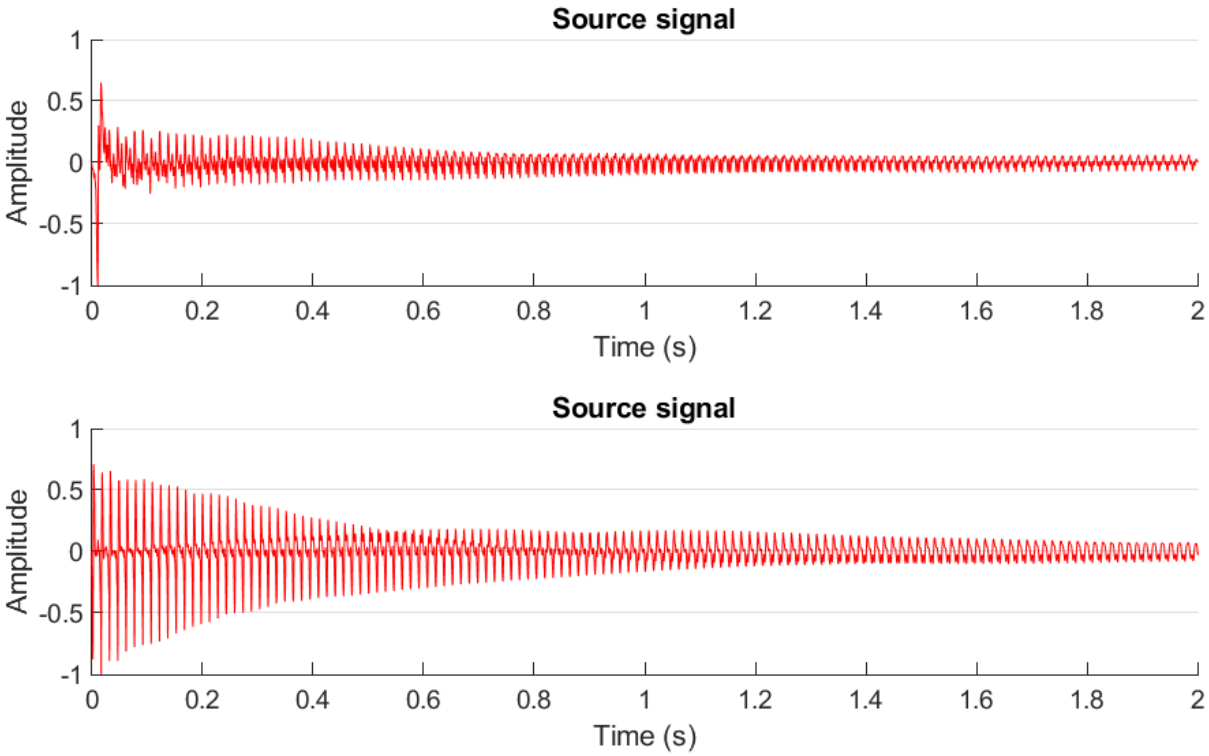


*Figure 6.5 – waveforms of a slapped note (top) and a note played fingerstyle (bottom). Both signals have been peak normalised. The apparent transient and subsequent complex dispersion of sinusoidal energy in the frequency domain are identifiable as differences in waveform shape in the time domain.*

## Observations concerning performance pitch and velocity

Similar observations are made regardless of the note played on the bass guitar, with some minute fluctuations at the highest note registers when string tension and pickup response are considered. Fig. 6.6 represents the magnitude spectra of fingerstyle notes played one and two octaves higher than the previous examples on the bass guitar. Similar spectral behaviour can be observed regardless of the left-hand position on the instrument, suggesting that the spectra produced by the bass guitar, and by extension its timbre, are heavily influenced by the manner of excitation applied to the strings.

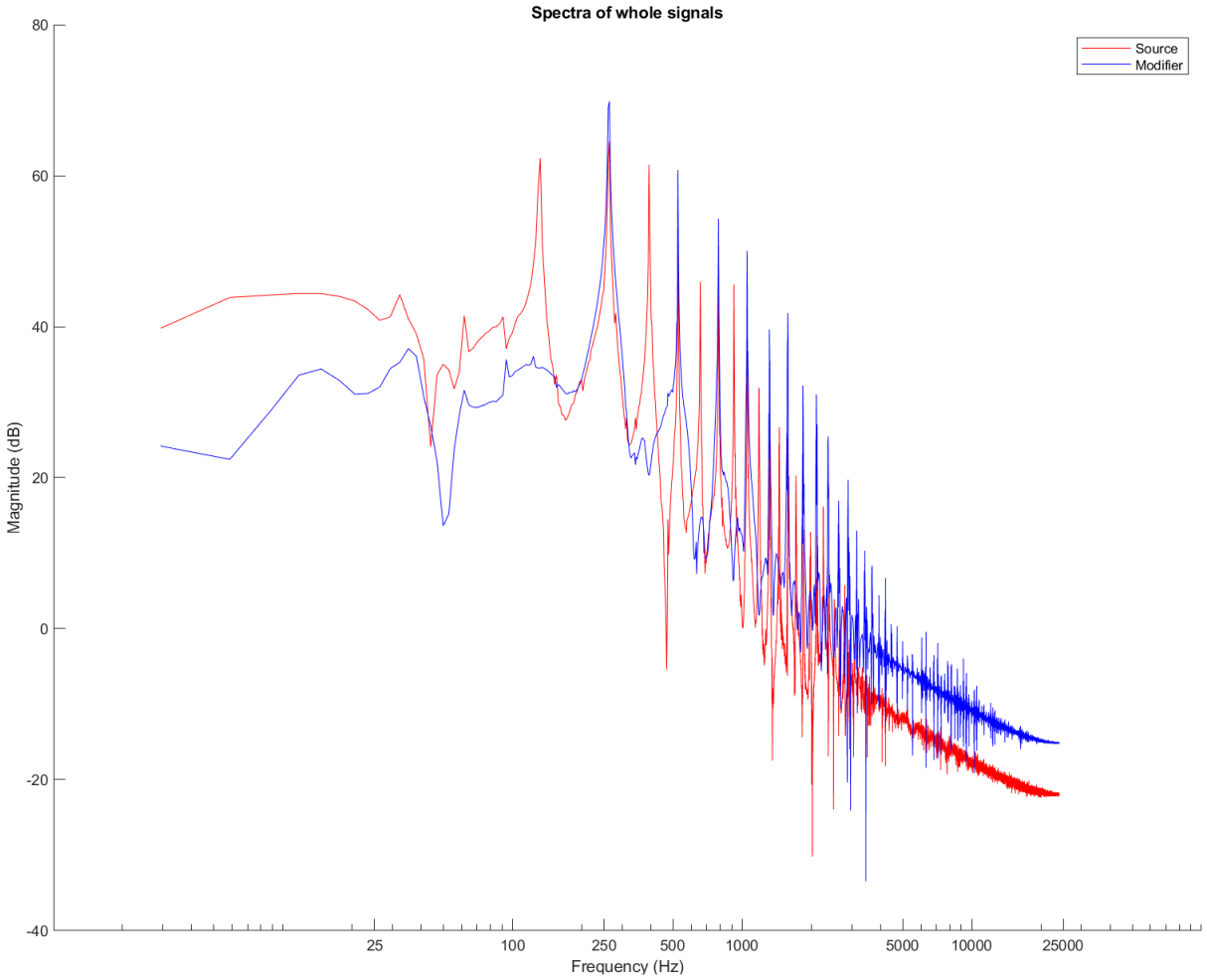


*Figure 6.6 – spectra of C3 and C4 bass guitar notes in red and blue respectively, sounded fingerstyle. Similar harmonic trends are observable between both examples and in figure 6.1, with variations attributable to string tension, gauge, the "shifting" of harmonic components further up the frequency axis and signal noise below the fundamental frequency.*

Striking the string softly produces similar partial structures in the magnitude spectrum as a hard strike would but with less apparent intensity in the high frequency ranges. The timbre of the softer note retains the quality of the striking medium but sounds less bright; this is reflected in the magnitude spectra presented in fig. 6.7. It can be deduced that a less apparent transient at the onset of the note, responsible for much of the high frequency noise-like partials, results in a warmer timbre through the reduction of these spectral elements in signals.

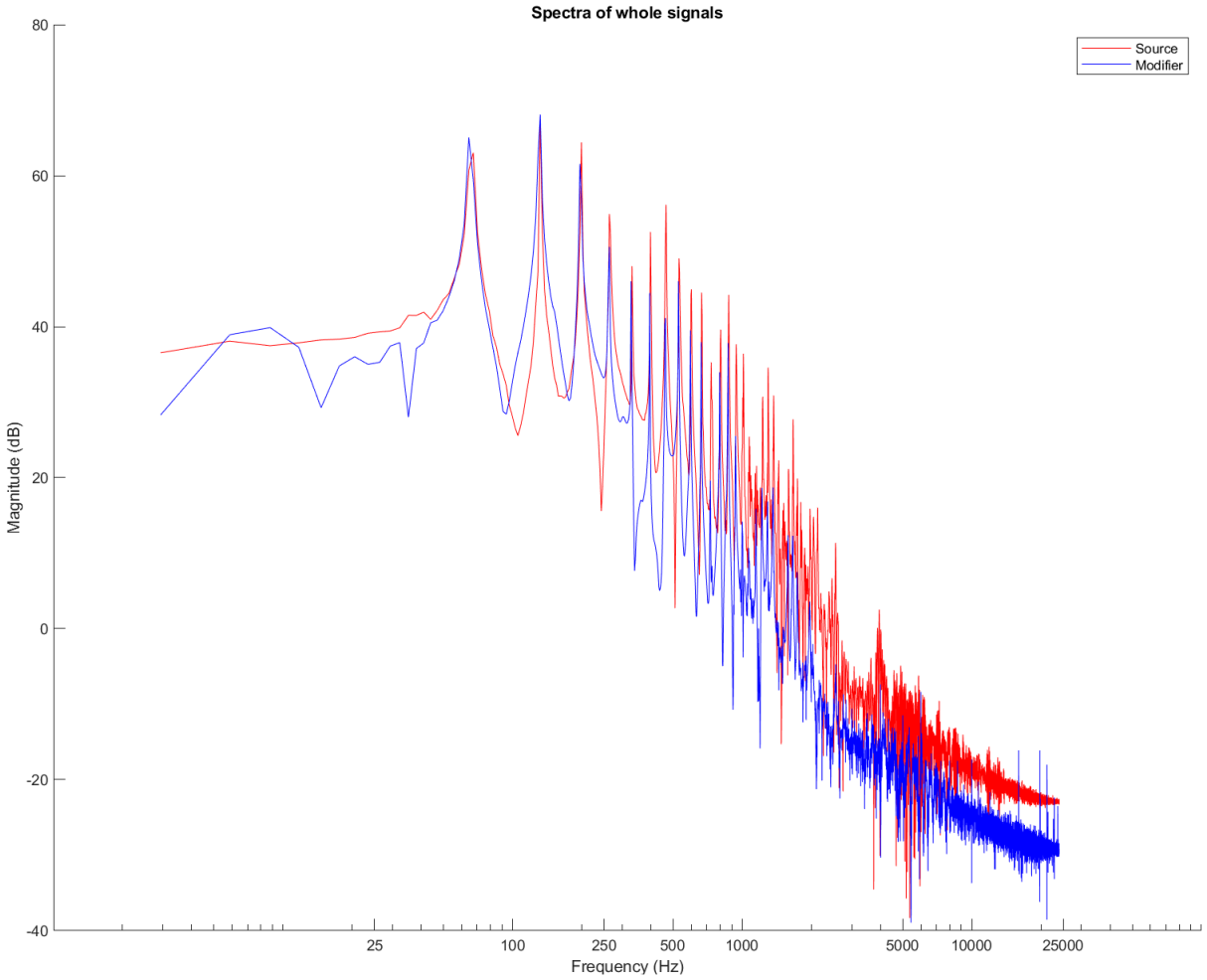


*Fig. 6.7 – spectra of a C2 note sounded with a plectrum softly (blue) and the high-velocity sample from 6.2 (red). The noise-like spectral components in the high frequency range present in 6.2 are here as well, reflecting the striking medium. However, they appear much less chaotic and prominent than in the previous example.*

It may be concluded that the transient of the note heavily shapes its acoustic properties whilst the steady sinusoids defines the perceived pitch of the note. This indicates the importance of not only spectra but note envelopes in timbral manipulation. It may also be remarked that the series of harmonics produced by the vibration of the string shape the 'body' of the perceived timbre of the note, providing the bulk of its lasting resonant properties.

# 6.ii. Effects of variables on filter performance

For the following investigation into filter performance, two zero-mean stochastic white noise processes were used for the source and modifier signals unless specified otherwise. The source signal utilised the unmodified white noise process, whilst a pre-filtered copy of the noise process was used for the modifier. White noise was filtered using a lowpass filter at 1000Hz with a 96dB/octave roll-off, producing a noise signal with a sharp slope after the cutoff frequency. These signals were chosen for filter testing as complex amplitude envelopes and spectral information are absent to allow for testing in an idealised environment. System parameters were initially set as follows, with alterations depending on the function being described:

- FFT resolution: 16,384 points
- Frame size: 256 samples
- Envelope matching peak detection window size: 750 samples
- Envelope matching on
- Filter order: 48 (effectively 96)
- Interframe filter curve smoothing: 100
- Intra-frame filter curve smoothing: 148
- Spectral envelope MA filter sample size: 80 samples
- Spectral envelope MA filter order: 5

## Envelope matching

As the filter curve is dynamic, changing frame by frame, it is highly likely for the envelope of the purely filtered signal to be altered beyond recognition in the process. This can only be avoided by using a mostly static filter, forced into effect by using a large degree of smoothing between frames and largely defeating the purpose of the matching filter

system. Therefore, it is recommended that envelope matching is used in every application of the system unless troubleshooting. Fig. 6.8 illustrates the effect of envelope matching on a filtered musical signal. The top example, with no envelope shaping, has an erratic note envelope leading to a poor acoustic representation of the modifier signal. This can be controlled by manipulating the envelope of the signal to match that of the modifier as seen in the bottom example. Here, the filtered signal was manipulated to resemble the modifier signal the system was provided with, restoring musical quality to the affected signal.

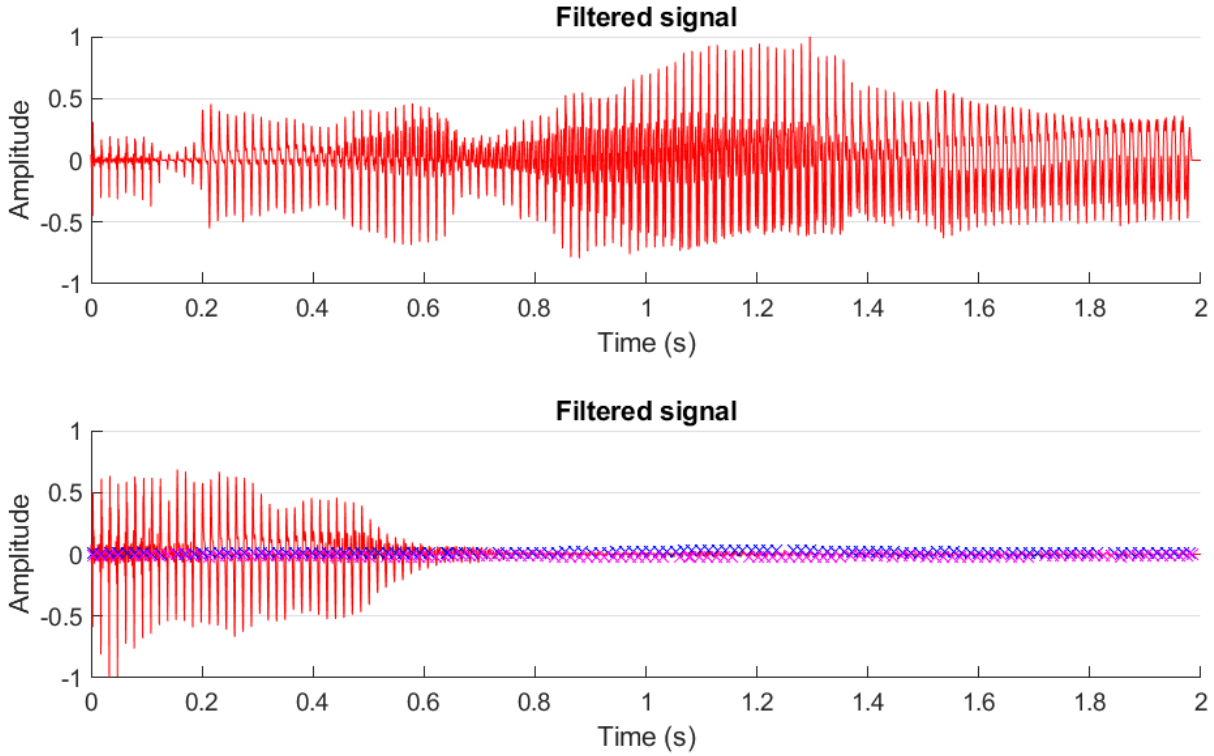


*Figure 6.8 – the effects of envelope shaping on a filtered signal. Top is without shaping post-filtering; bottom is the same signal with envelope matching applied.*

The envelope detection system functions using a sliding window. The sample range of this window is defined by the user. Two peak values are reported per chunk for the positive and negative poles of the system respectively. Therefore, decreasing this value causes more peak values to be reported, improving the resolution of the envelope vectors used for amplitude coefficient estimation. However, specifying too low of a value will result in erroneous peak values being reported, resulting in an improper estimation of the signal envelope. This problem will be explored in further detail later in this chapter.

## Frame size

To arrive at a sensible estimation when determining the optimal length in samples for frame-based processing, some vital considerations must be made. Firstly, the quasi-real-time functionality of the system should be reflected by selecting a frame length that will minimise apparent latency and react to changing spectral information speedily. Choosing a reasonably small frame length should be prioritised in instances where the timbral quality of the signal fluctuates rapidly or when multiple notes are triggered in succession. Causing the system to operate too quickly limits the accuracy of mathematical processes that estimate the dynamic filter curve by reducing the information available to the system. Specifying too high of a frame size causes the filter to operate too slowly to replicate intricate timbral details on the source signal.

## FFT resolution

Invariably, the resolution of the magnitude spectra used for the calculation of filter parameters has the greatest effect on the signal quality produced by the system. Using a high number of samples per frame increases the data available to the system, which improved spectral matching accuracy even with a smaller-point FFT. It was discovered that using a larger-point FFT will always produce more acoustically pleasing results at the expense of processing time. This property remains true even in cases of extreme zero padding, where the signal frame length is transformed using an exceptionally high point FFT. No further information can be obtained from the frame without increasing its length, so there is no reporting of unseen or hidden sinusoidal components in the magnitude spectrum. Rather, there is more room for interpolation between the existing spectral components, producing a finely sampled spectral curve. From here, the principle is simple; the longer FFT bins allow for more points of analysis when obtaining filter coefficients, increasing the number of calculations to be performed but improving filter accuracy. Fig. 6.9 portrays a processed signal in magenta against the input signals used to produce it. The higher FFT

resolution used in the second example provided an improved fit to the modifier signal spectrum in blue. For all experiments performed in this chapter, an FFT resolution of 16,384 was used unless specified otherwise. Increasing the FFT points beyond this value produced little noticeable improvement.

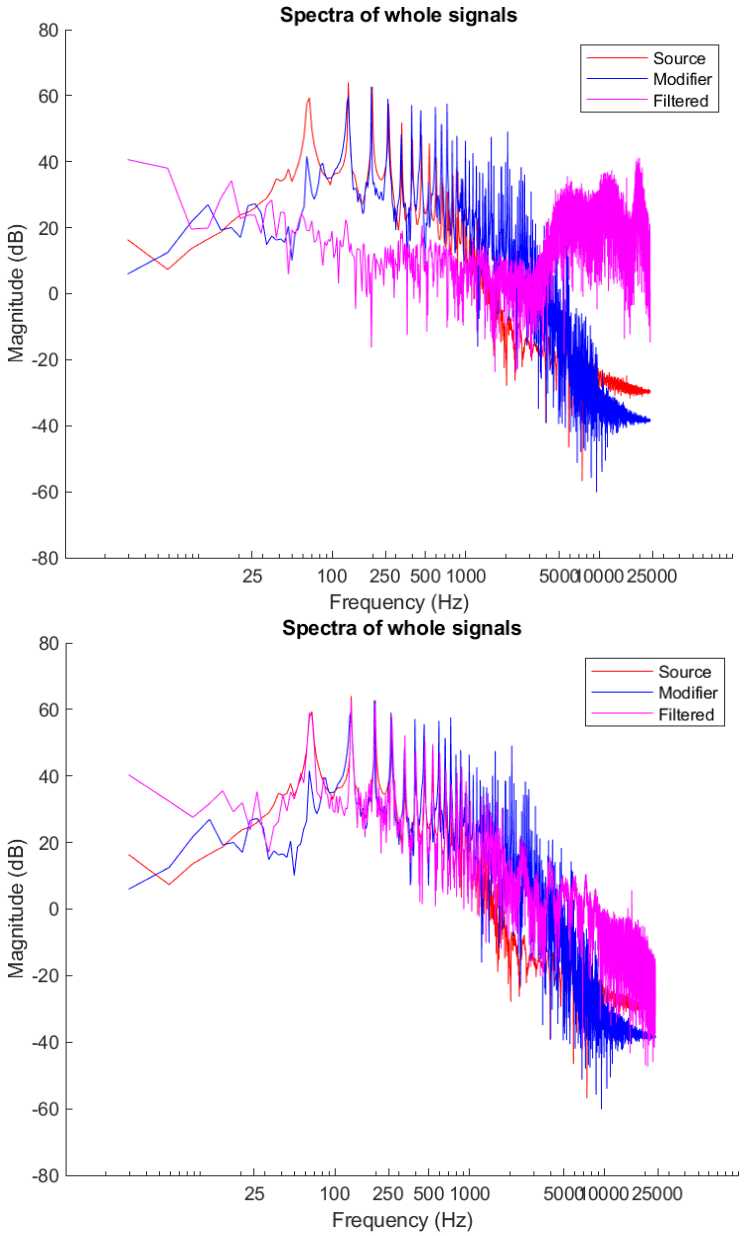


*Figure 6.9 – the results of variating the FFT resolution in signal processing. The first example used a 512-point FFT, whilst the second used a 16,384-point transformation.*

## Spectral envelope estimation

Spectral envelopes are extracted from the magnitude spectra of frames using a moving average filter applied to a vector of apparent peaks, removing resonances in the spectra in the process whilst retaining the general shape of the underlying magnitude spectrum. The MATLAB function `findPeaks` was used to identify high-magnitude samples in the signal spectra. It was then smoothed using the MA filter function `smooth`. The MA filter takes two parameters; the sample size for the sliding window and the MA filter order, specifying the number of iterations the filter must perform. Increasing these parameters will increase the degree of smoothing applied to the dataset. By lengthening the window sample size, the filter becomes more efficient at evening out resonances spaced further apart as the algorithm is allowed a greater scope for averaging, demonstrated in fig. 6.10. Increasing the filter order amplifies the effect of the filter but does not necessarily smooth the peaks if the sampling range is not wide enough to capture significant spectral peaks within its boundaries. Fig. 6.11 portrays the $10^{th}$ order MA filter applied to the same frame of audio with variable sampling ranges. Despite the high filter order, the MA filter is unable to eliminate resonant peaks from the spectral envelope if the sampling range is too small. It can be concluded that the length of the sampling window is the key factor to producing an acceptable model of a spectral envelope in this application. Furthermore, the user can estimate a sensible sampling range from their knowledge of the harmonic structure of the signals loaded and scale their prediction according to the specified FFT resolution.

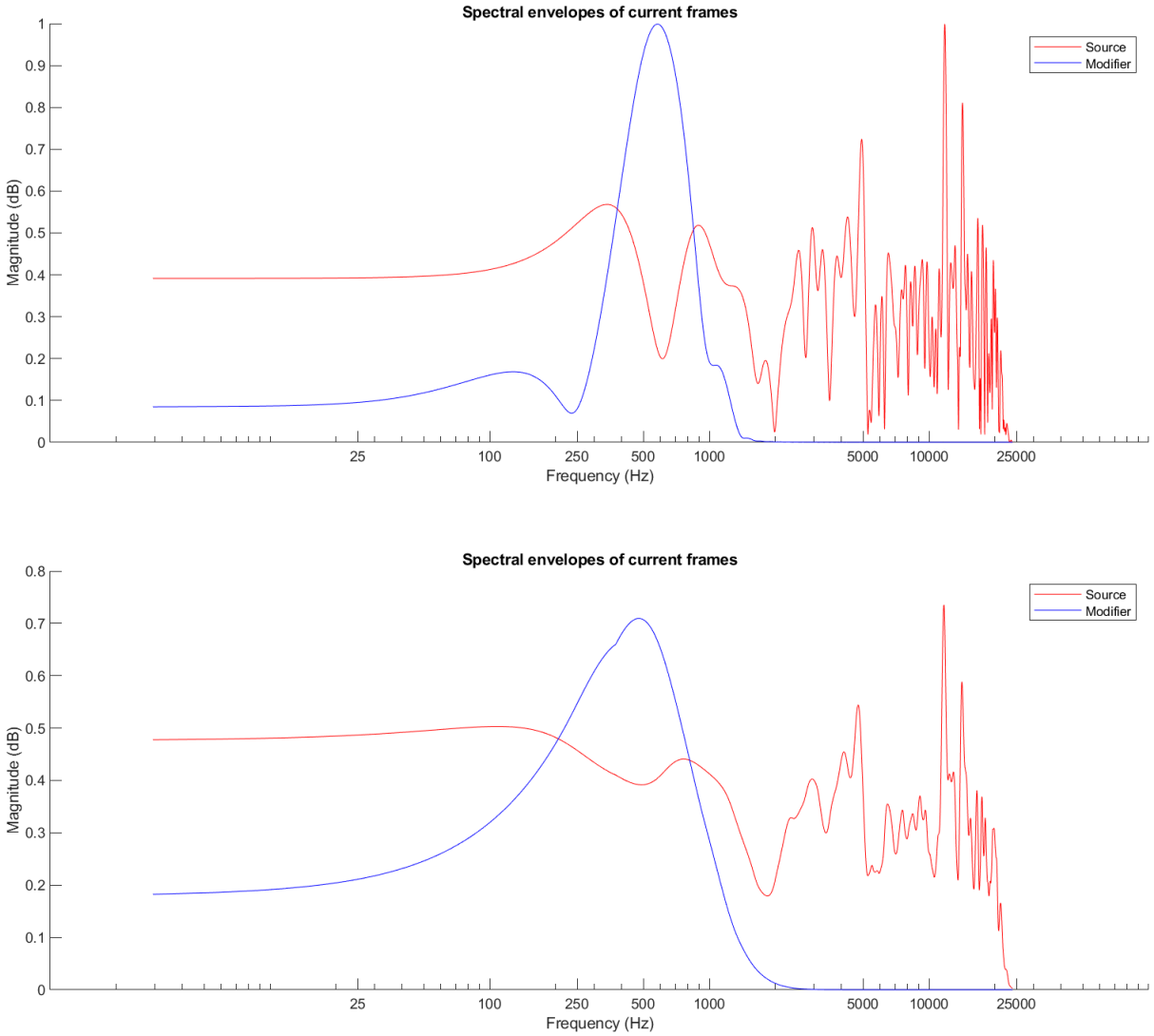


*Figure 6.10 – spectral envelopes affected by a first-order MA filter. The first used a window sample length of 8 samples, whilst the second was assigned a length of 256.*

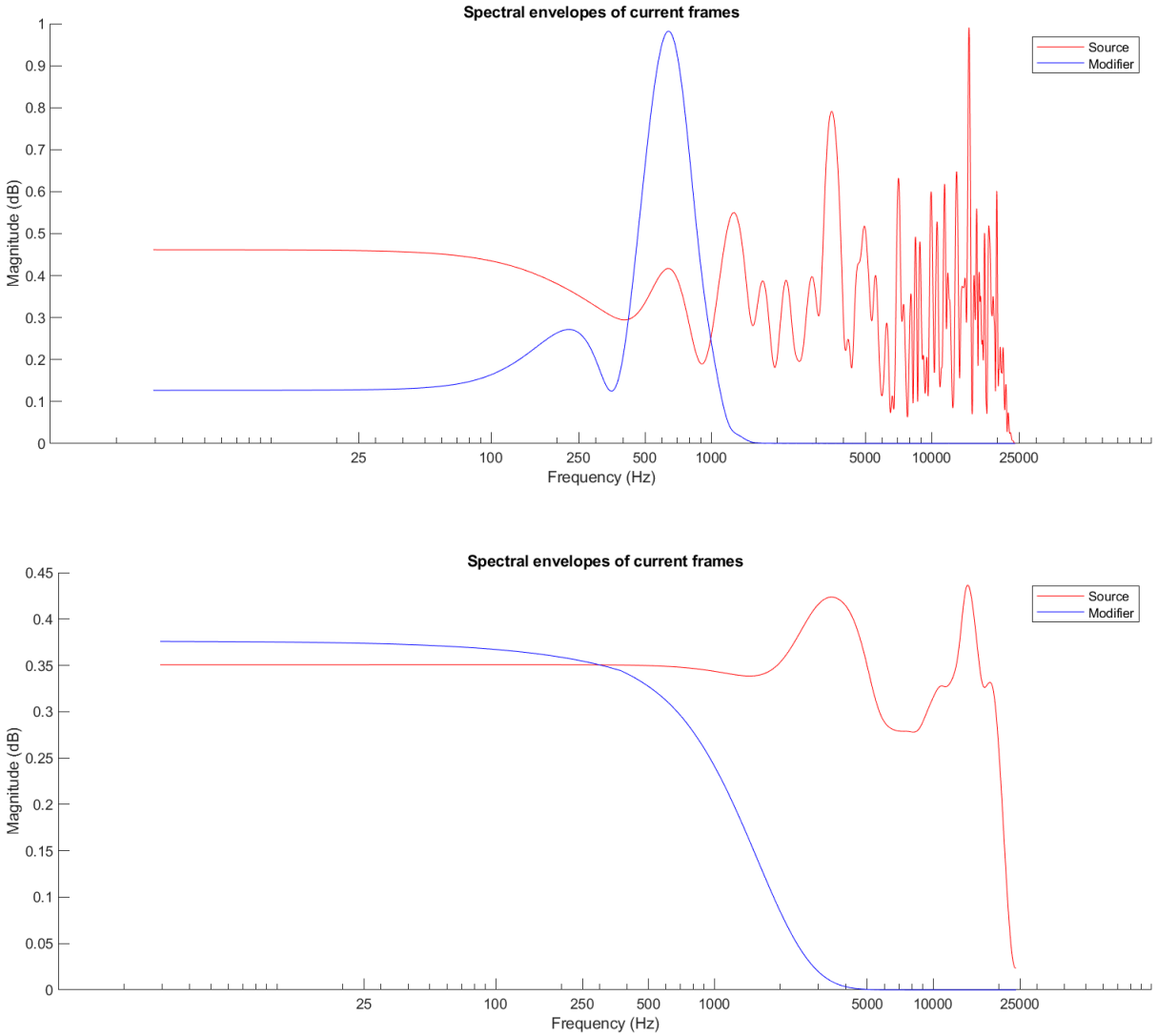


*Figure 6.11 – 10th order MA filter applied to frame spectra with variable sampling range. The first demonstrates a range of 8 samples, and the second 256 samples.*

For the white noise signals, it was found that a window length of 80 samples and an order of 5 was enough to produce an accurate estimation of spectral envelopes. A well-estimated envelope should fit the general shape of the signal provided. In the lower example of fig. 6.11, the spectral envelope of the lowpassed noise signal (blue) was extracted very accurately. The estimation of the source signal spectral envelope (red) was poorer, owing to the concentration of erratic spectral components in the white noise signal. The sampling range of 80 was ideal for de-noising the spectra enough to avoid artefacts in the estimated filter curve whilst being restricted enough to retain key spectral information.

Increasing the order beyond 5 began to erode the remaining peaks too far to estimate an accurate filter curve. Fig. 6.12 illustrates the estimated spectral envelopes, the filter curve produced as informed by these estimates and the effect of the filter on a frame of audio.

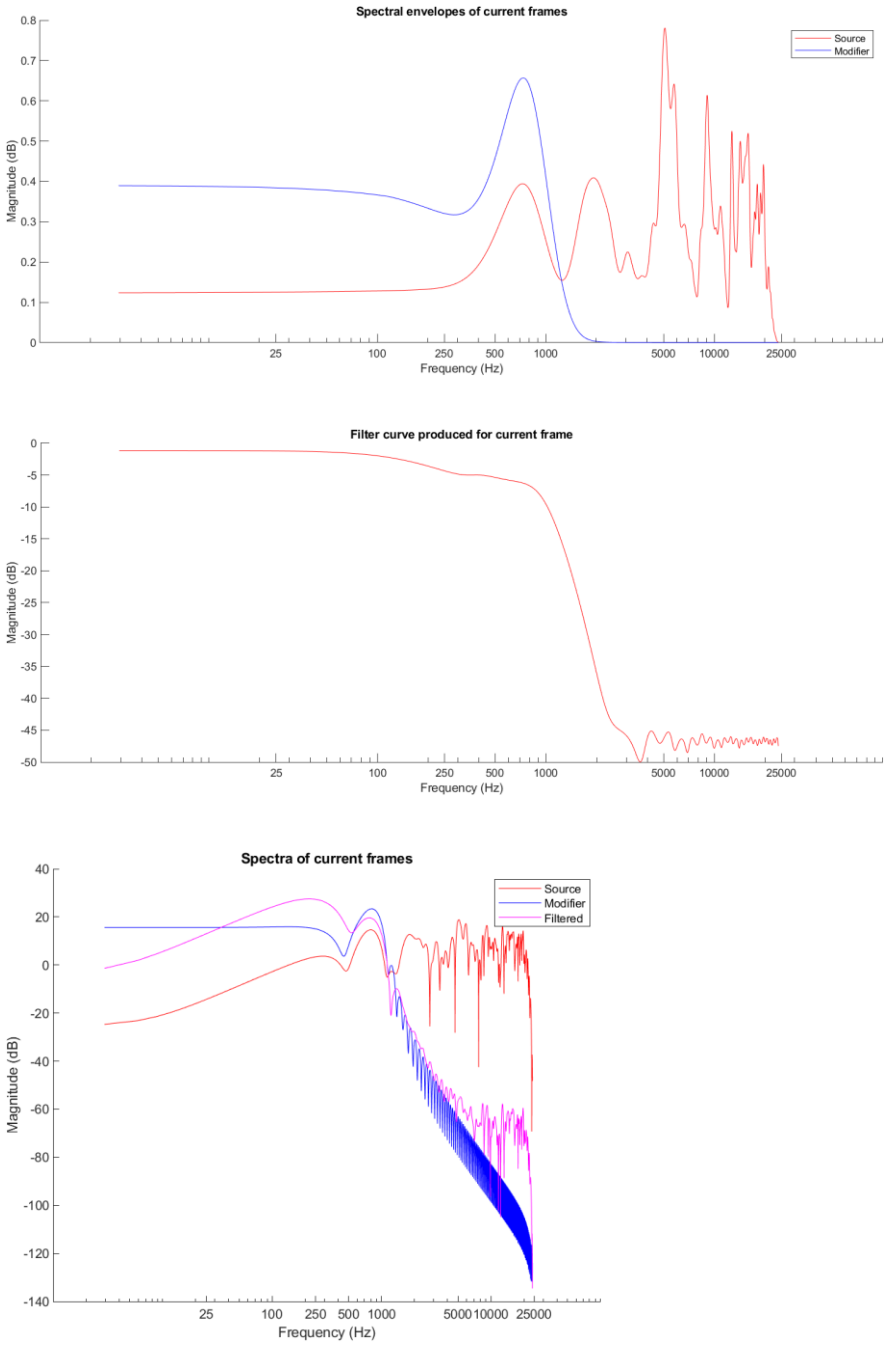


*Figure 6.12 – effect of the matching filter on a frame of audio using optimal settings for spectral envelope estimation.*

## Matching filter order and smoothing

Recall that the native zero-phase filtering function in MATLAB functions by passing the signal through the filter once, then reverses the filtered signal before passing it through again to negate the phase shifting introduced by IIR filters. Therefore, the filter order specified by the user is half that of the effective order. Almost invariably, increasing the filter order improves the performance of the filter, with no significant sonic improvements if the order exceeds 64 (effectively 128). Likewise, reducing the order of the filter below 32 hinders filter performance, with especially low values failing to accurately replicate the spectral curve of the modifier signal. Fig. 6.13 illustrates the spectral performance of the filter with an order of 4.

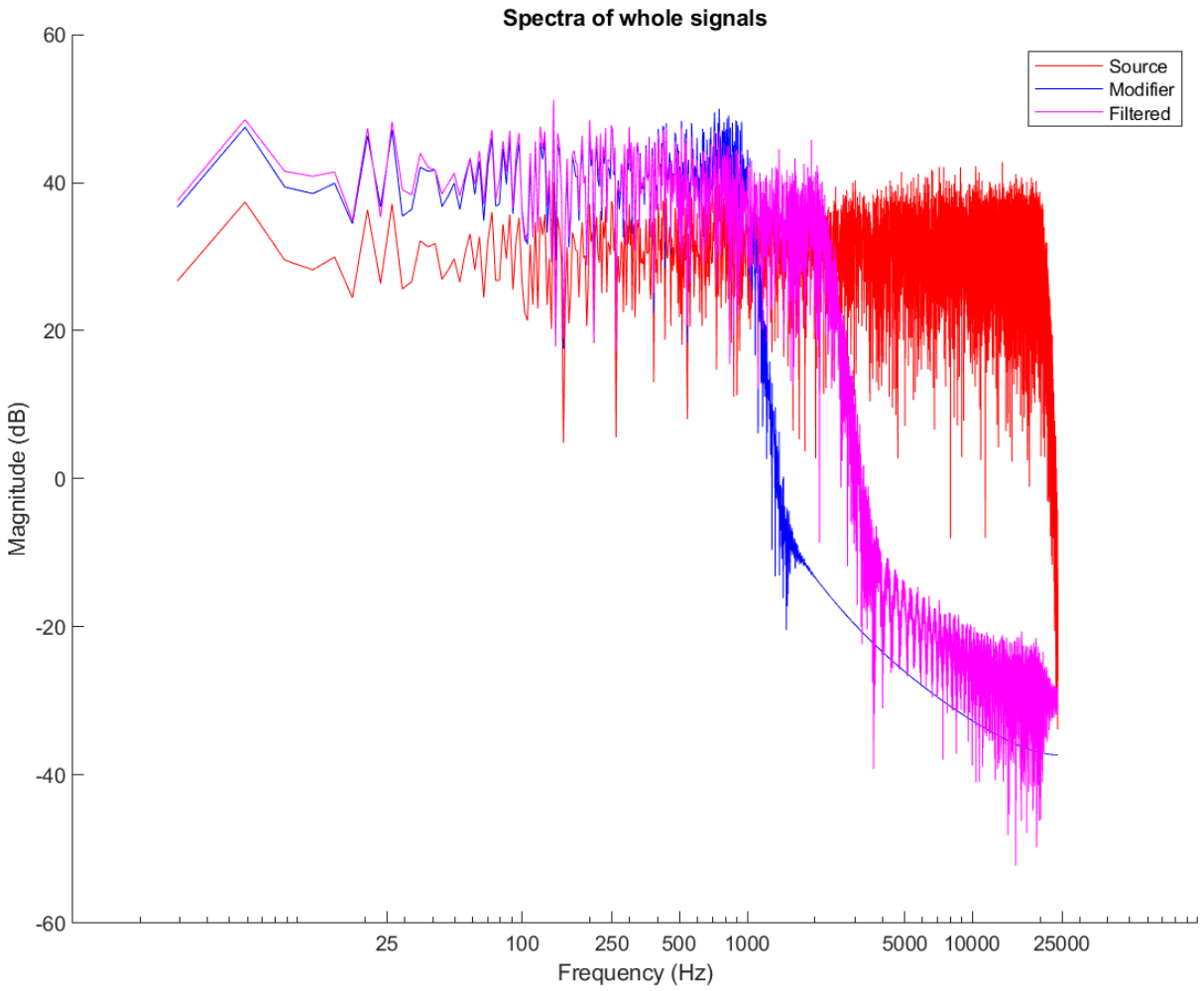


*Figure 6.13 – effect of the matching filter on white noise signals with an order of 4. The modifier spectrum is modelled poorly when using a lower filter order.*

Setting the filter order too high can often be detrimental to the output signal and will require different degrees of interframe filter curve smoothing to temper the extra detail introduced to the curve. Fig. 6.14 illustrates this behaviour; the apparent scalloping in the filter curve is reduced as the smoothing parameter is adjusted to be higher. The effect of altering this sole smoothing parameter to compensate for the higher filter order is profound and leads to impressive matching filter results as illustrated in fig. 6.15. The second example with the compensated smoothing setting approaches the modifier spectral curve to a considerably greater degree. Acoustically, the filtered sound is more pleasing to listen to as the filter curves produced are less erratic overall with a reduced fluctuation in filter curve shape over time.

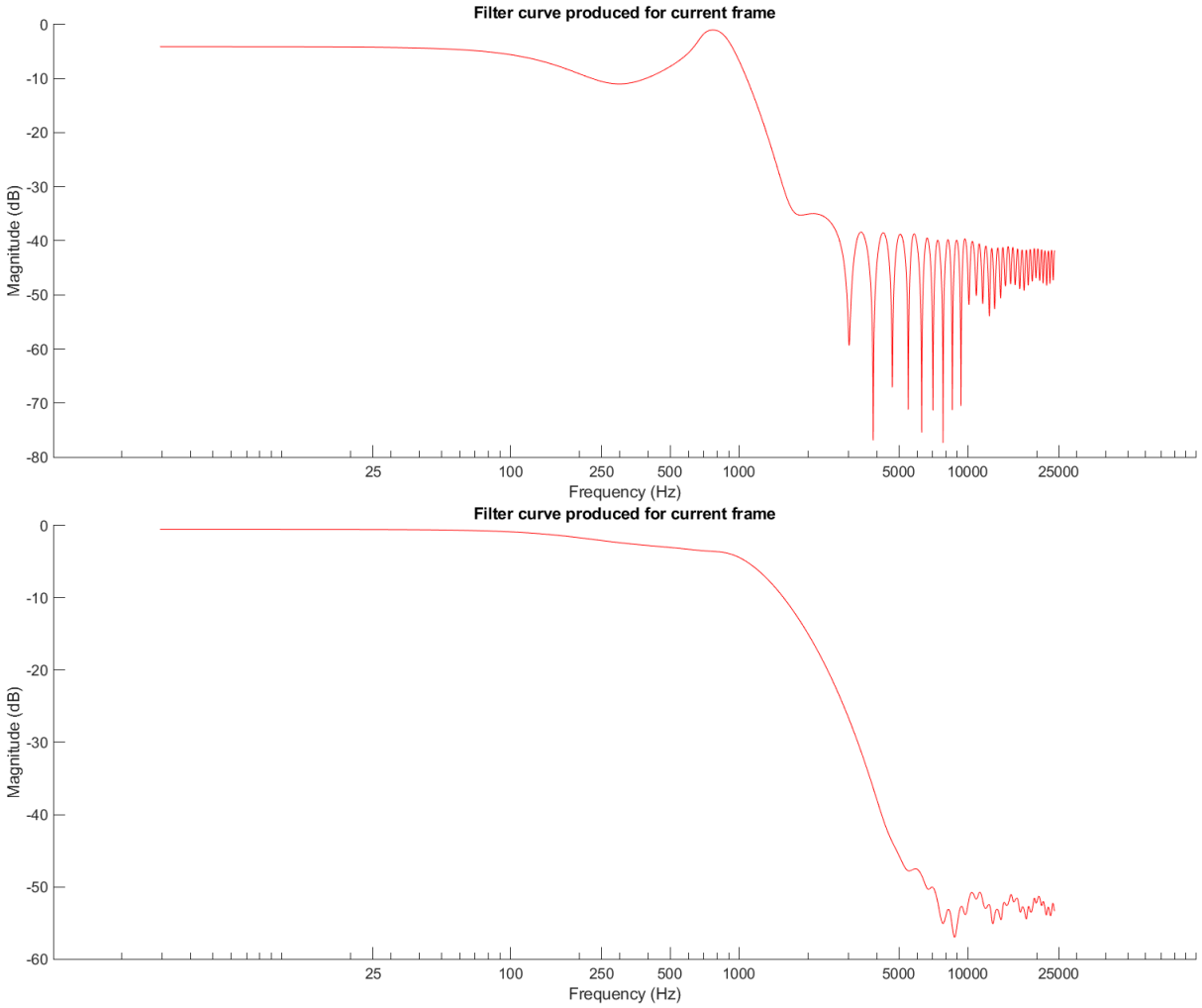


*Figure 6.14 – filter curves produced for a frame of audio. The first was produced with an order of 64 and an interframe smoothing setting of 64 samples. The second example uses a smoothing value of 256.*

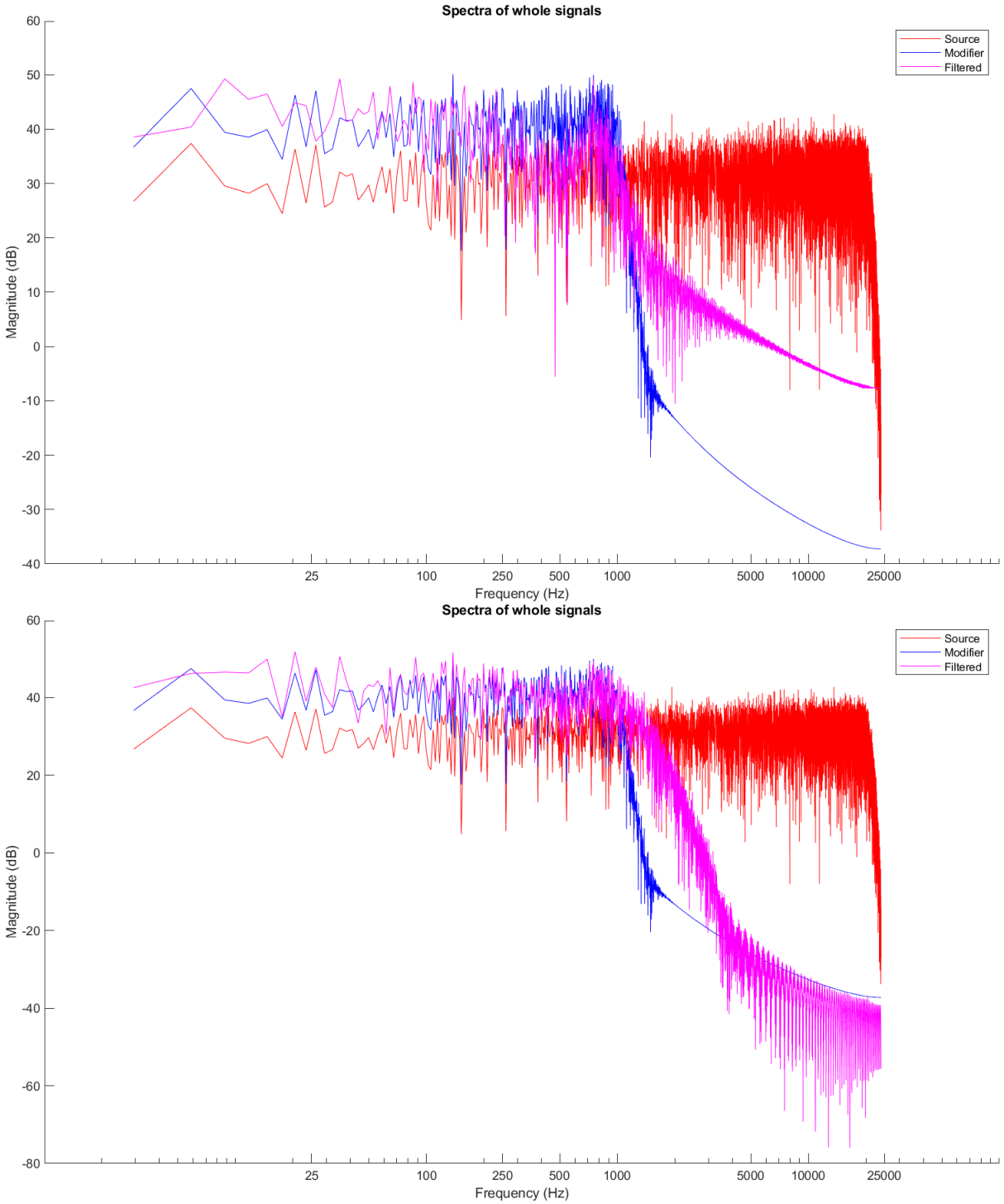


*Figure 6.15 – spectra produced for the noise signals. Similarly, the first example took a filter order of 64 and an interframe smoothing setting of 64 whilst the second uses a smoothing setting of 256.*

It can be surmised that a balance must be found between the amount of interframe filter curve smoothing and filter order to suit the needs of the source and modifier signals. For the white noise signal and its low-passed counterpart, an order of 48 (effectively 96) and

an interframe smoothing value of 128 was found to be ideal. Fig. 6.16 depicts the spectra of the filtered signal in magenta, fitting the curve of the modifier signal (blue) snugly. The filtered sound is comparable to the modifier signal in its character with slight discernible differences between the two. Ultimately, balancing the relationship between filter order and interframe smoothing is crucial to producing a clearly filtered musical signal retaining properties of both the source and modifier samples.

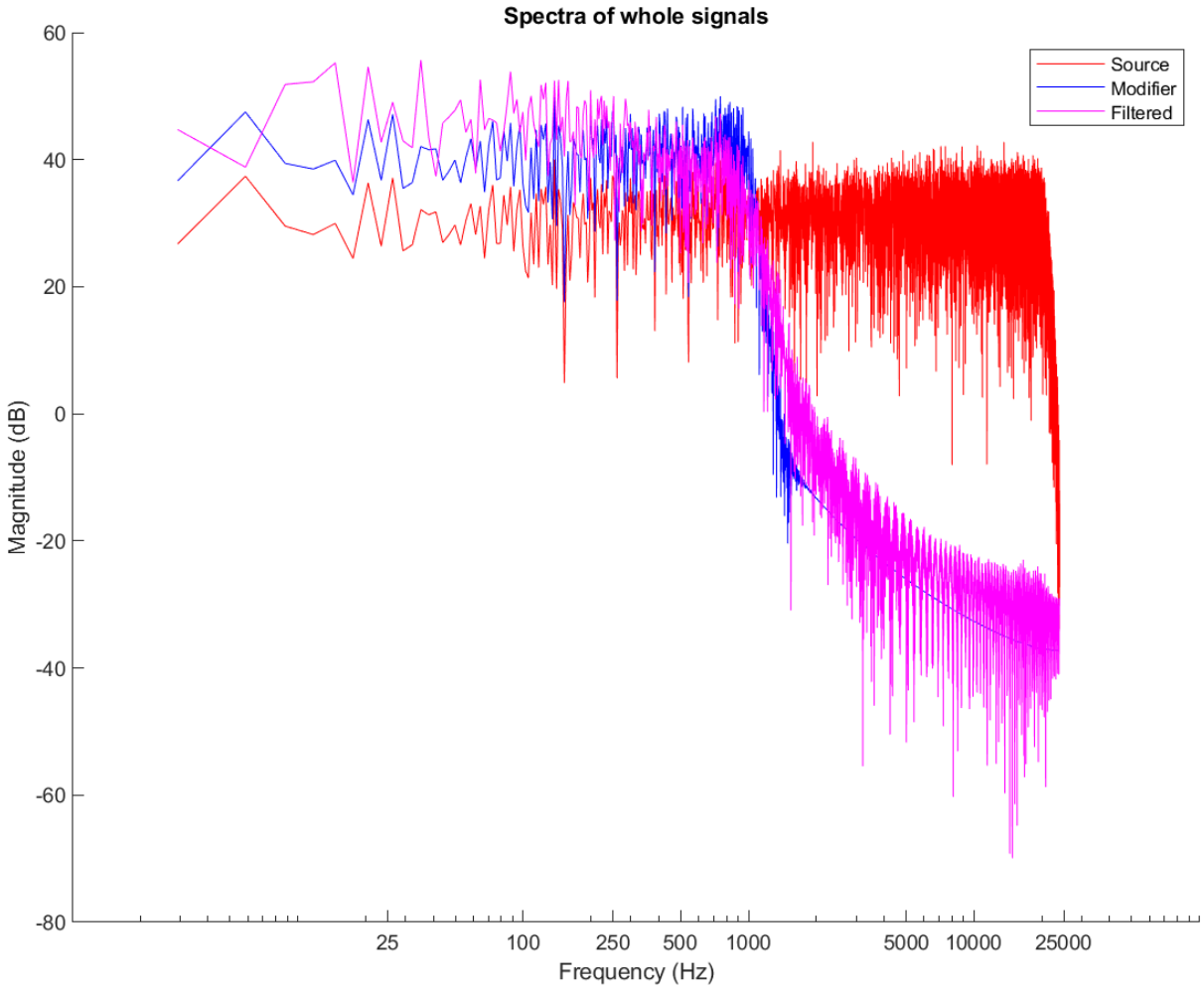


*Figure 6.16 – spectra produced for the noise signals using optimal interframe smoothing and filter order. The cutoff frequency (around 1000Hz) of the lowpassed noise sample is matched effectively.*

A second degree of transfer filter smoothing, dubbed intra-frame smoothing, is also required to further control the dynamic filter curve generation. As a unique filter curve is produced for each frame, it can be expected that their shapes will fluctuate wildly between

frames as there is no handling of final or initial filter conditions in the system. Should each filter curve deviate significantly from the curve of the preceding frame, it is highly likely that the filtered signal will appear distorted as partials in the spectra are aggressively over-manipulated. By smoothing filter curves between frames, the persistence of the generated filter curves can be controlled, effectively reducing the distortion introduced by the dynamic filter curve system. A hypothetical “distortion free” scenario would be to use a static filter curve which would fail to react to the spectral envelope of the modifier signal. In contrast, a system with no intra-frame smoothing at all would most accurately model the required filter curve for each frame but introduce an unacceptable level of distortion to the signal. Therefore, the user must choose to trade off system reflexes for increasingly acoustically pleasing results. Fig. 6.17 portrays the spectrum of the processed signal with no intra-frame smoothing applied. Distortion and excess noise introduced to the signal are clearly visible above 1000Hz, above the lowpass cutoff of the modifier signal.

Finding an appropriate value to use for intra-frame smoothing is largely dependent on the eccentricity of spectra as they appear for every frame. Should the spectra of frames deviate significantly from one another over the length of the entire buffer, a higher degree of smoothing will most likely be needed. This is a common occurrence in musical signals as will be investigated later in this section. For this example of a lowpassed white noise signal, little intra-frame smoothing is required as the filter cutoff applied to the modifier remains constant throughout the sample. Real musical signals can be expected to have more complex signal envelopes and spectral shapes, hence the requirement for greater intra-frame filter curve smoothing.

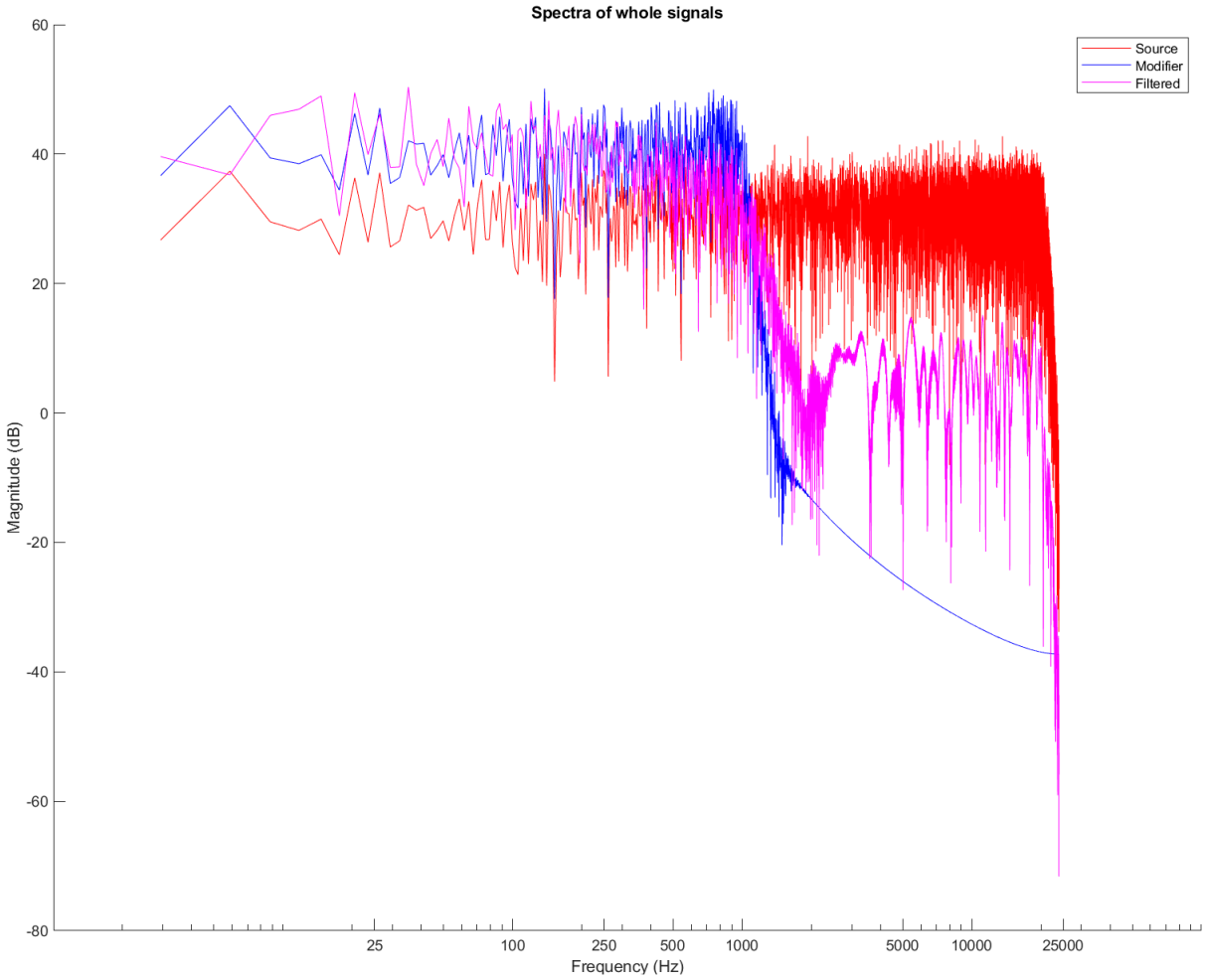


*Figure 6.17 – Matching filter applied with no intraframe smoothing. The acoustic results are poor, owing to the noise-like dispersion of frequencies reported in the high frequency range.*

## 6.iii. Filter performance concerning musical signals

The performance of the filter has been demonstrated on simple noise signals, which provide optimal conditions for the filter to operate. Musical signals are notably more complex however; unlike the noise samples used in section 7.ii, musical notes are subject to spectral changes over time as sinusoidal components in the signal decay. It is known that the matching filter can estimate a filter curve to reliably fit the spectrum of a subject signal to that of another when provided with acceptable parameters by the user. To be established in this section is the performance of the filter when complex variables such note envelopes and complex spectral fluctuations are concerned.

For the following examples, one of two musical signals were used as the modifier signal in the matching filter system to produce the filtered output, whilst the bass guitar is defined as the source signal. The modifier signals created for these demonstrations consist of a synthesized square-wave bass note and a grand piano, each possessing spectral and temporal qualities quite different from bass guitar signals.

## Bass guitar with piano modifier signal

Piano signals were used as modifier signals to influence the time and frequency domain qualities of source bass signals. Fig. 6.18 illustrates three signals in the time domain loaded by the system, each representing a single note. The fingerstyle bass guitar note is pictured in the top plot, used as the source signal in processing. The piano note is pictured in the middle plot with its envelope outlined in blue and magenta crosses. The processed signal occupies the bottom plot with its envelope outlined in crosses before it was reshaped to fit the modifier. Time-domain envelope matching performance of the system on these samples was excellent as the filtered signal matches the shape of the modifier very closely. Upon closer inspection, the system has attempted to match the envelope of the filtered signal to the noise floor of the modifier signal after the note itself has died off after roughly 0.9 seconds. This behaviour is encouraging as it signifies the envelope matching algorithm can manipulate the shape of a musical note extensively and adjust to extreme changes in envelope shape reliably. However, it also indicates that the system begs for more granular control as the portion of the filtered signal being matched to the signal noise is part of a sustained note. The acoustic result is not residual background noise after the note is released but an extreme alteration in the trajectory of the note decay. Slow-moving sinusoidal components comprising pitch and timbral information are retained in the signal when they should be absent for a truly accurate match in signal envelopes. Despite this observation, the acoustic performance of the envelope matching system is good as the unique attack and note envelope of the piano is effectively replicated on the filtered signal whilst musical qualities of the bass guitar are retained. Sliding window size local maxima

detection was set to 750 samples in this example, so one peak and its corresponding location is reported for the modifier and processed signals every 750 samples which is in turn used for envelope estimation. It can be observed that there are few noticeable errata in the location of crosses on the modifier signal, indicating that the envelope estimated for the signal is relatively true to the actual waveform shape. It can also be observed that the zero crossings of the signal have been preserved and amplitude modulation has been avoided entirely, producing an acoustically clear processed signal. By checking the RMS loudness of the three signals, it can be confirmed that the envelope matching algorithm has matched the energy of the modifier signal well. The source signal returns an RMS value of 0.312, the modifier returns 0.119 and the processed signal returns 0.117; for comparison, the filtered signal without envelope matching (fig. 6.19) returns an estimated RMS loudness of 0.264.

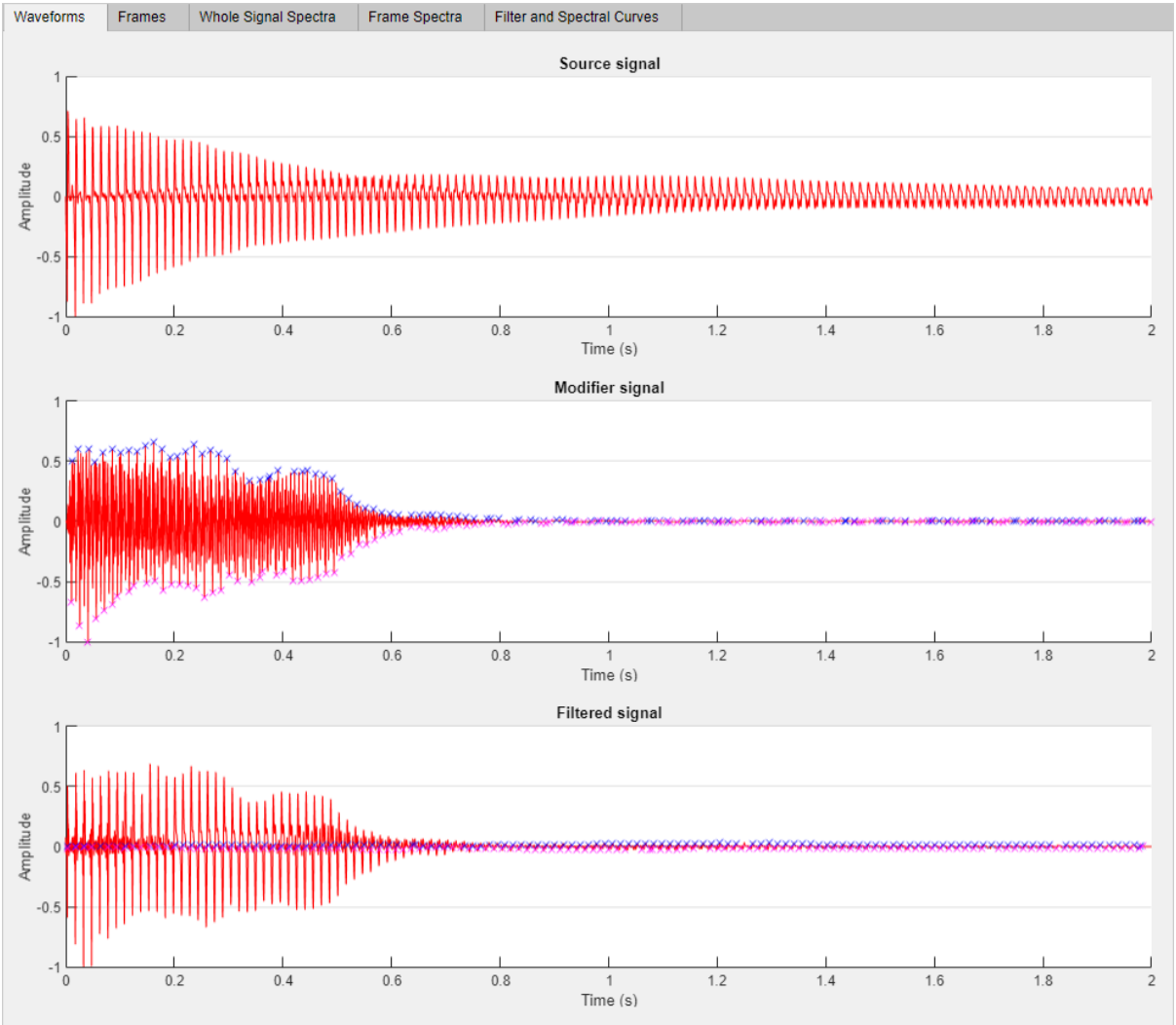


*Figure 6.18 – C2 bass guitar note (source) filtered and shaped to take spectral and temporal qualities from a C2 piano note (modifier).*

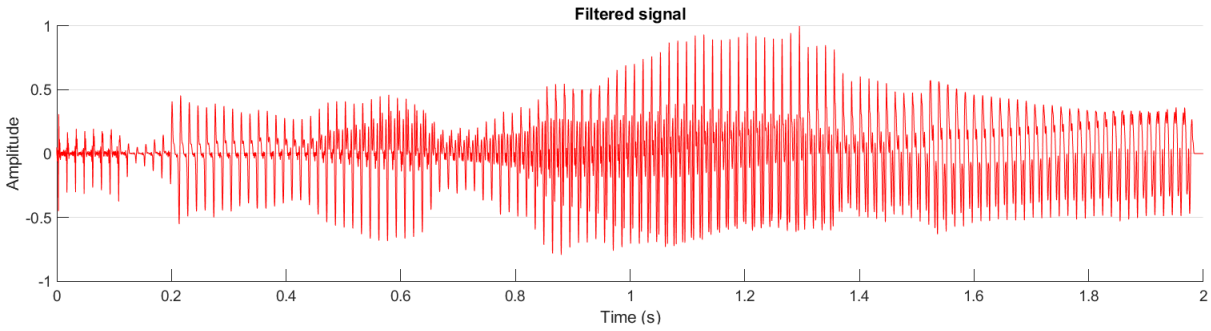


*Figure 6.19 – filtered note processed in the same manner as 6.19 but without envelope matching. The shape of the waveform is greatly distorted by the filtering process and bears little resemblance to the source or modifier signals.*

Spectral results of the system are also encouraging. In fig. 6.20, the shape of the source waveform has been heavily reshaped to fit the shape of the modifier. The most noticeable change in waveform shape can be observed beyond 1000Hz, where the source spectrum has been altered quite dramatically, bringing it roughly in line with the modifier curve. In general, the shape of the source spectrum has been matched to the modifier well until around 5kHz, where the processed signal appears to deviate wildly and report high frequency content surpassing both unprocessed signals. This can largely be attributed to smoothing of the filter curve culling high-end detail, omission of low-energy points during filter calculation (typically found in the highest end of the frequency spectrum) or distortion introduced from reassembling the signal from independently filtered frames. The processed signal matches the timbral qualities of the piano note reasonably well overall, as much of the low-end is substituted for mid-range “bite”, although a noticeable buzzy quality is introduced to the signal by the filter. The effect is incredibly slight, but present in neither the source or modifier signals and is assumed to be distortion. Fig. 6.21 portrays an average filter curve produced for this example and pictures little attenuation in the frequencies concerned, therefore it can be surmised that the filter is operating poorly in the highest frequency ranges of the signal. Altering user definable settings did not significantly change the performance of the filter in this area, nor did disabling the omission of low-energy frequency components from filter coefficient calculation.

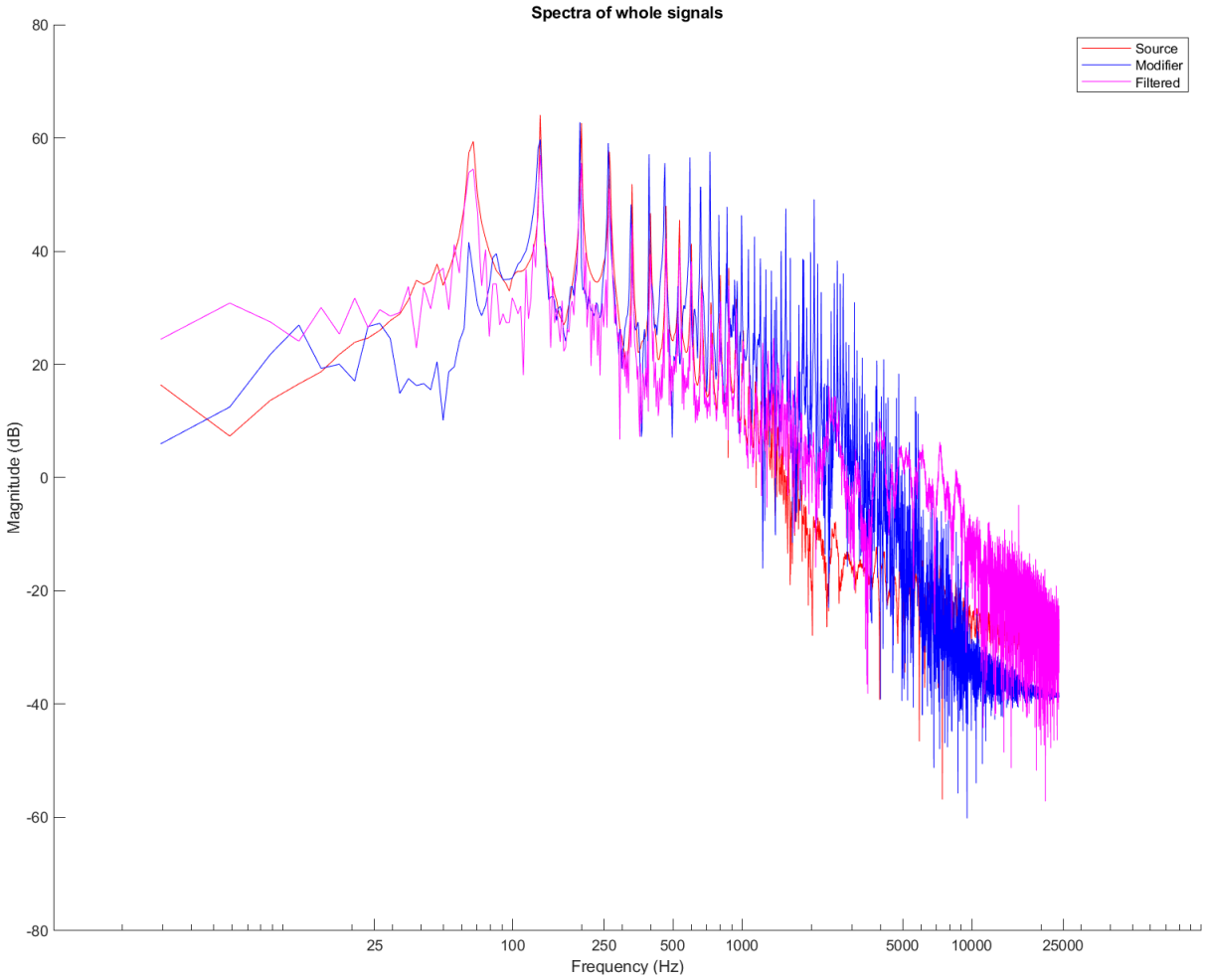


*Figure 6.20 – spectra of the C2 bass guitar note (red), C2 piano note (blue) and processed signal (magenta).*

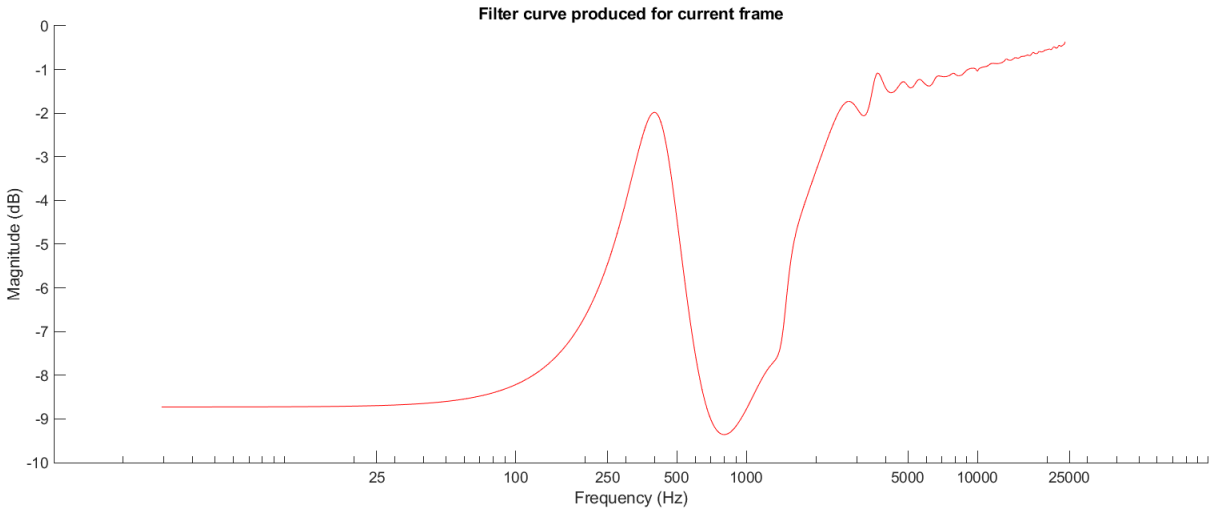


*Figure 6.21 – average filter curve produced for a frame from 6.21, taken from roughly halfway through the signal.*

The fingerstyle bass guitar signal defined as the source was then swapped for a note sounded using a plectrum and the signal processing was reapplied. System performance was similar in both the time and frequency domain, although it can be noted that the acoustic result appeared closer to that of the piano modifier signal. This could be due to the spectrum of the plectrum-sounded source signal being closer in shape to the modifier before signal processing than the fingerstyle sample was. Fig. 6.22 illustrates the time-domain plot of the processed signal, which fits the envelope of the modifier well. One noticeable error can be observed just after 1.4 seconds where a slight bump has formed in the time domain representation of the signal that is not present in the modifier, although the bulk of the sample before the piano note is released is matched accurately.

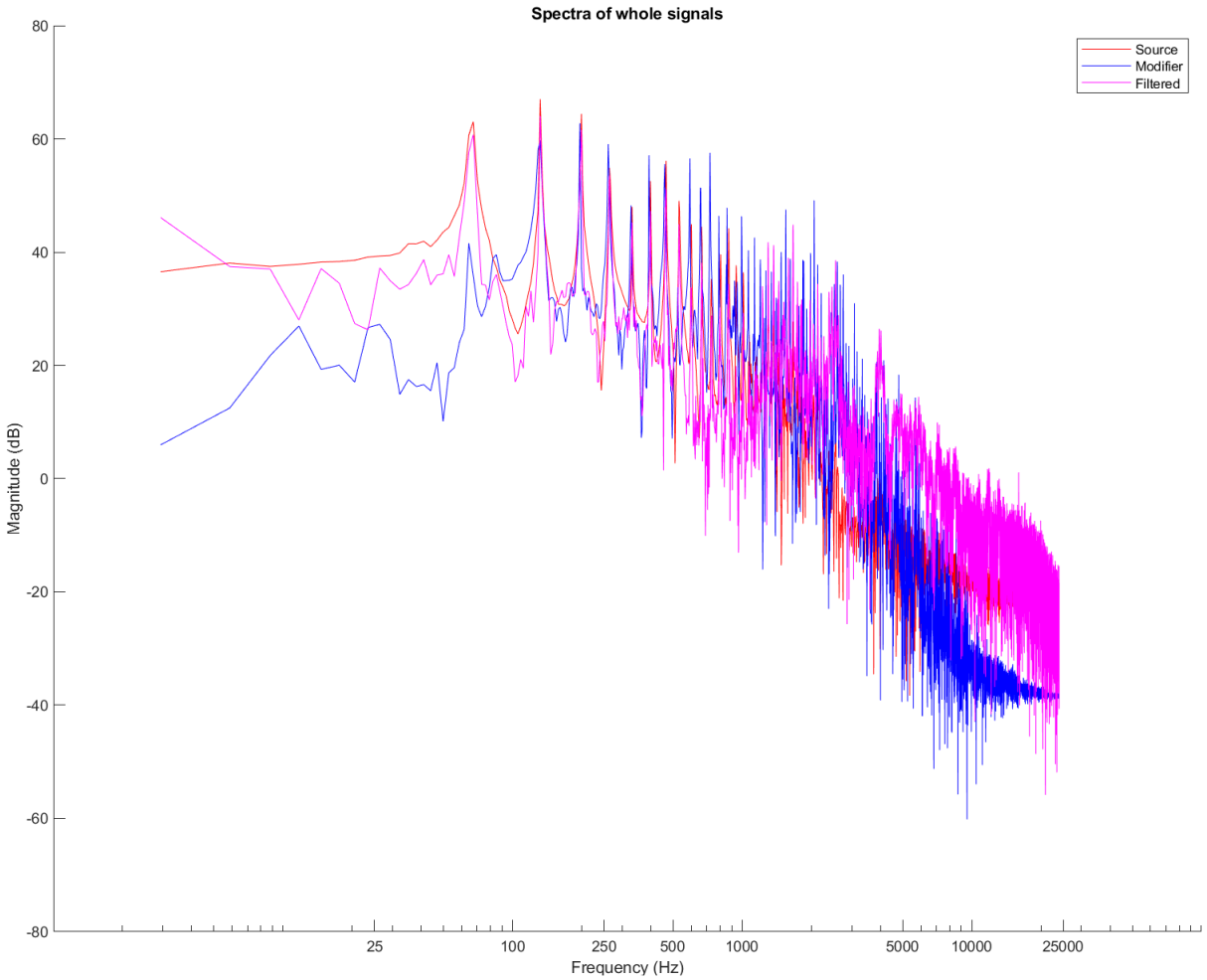


*Figure 6.22 – signal processing applied to a C2 bass note sounded with a plectrum. Similar high frequency distortion is observable here as in 6.20.*

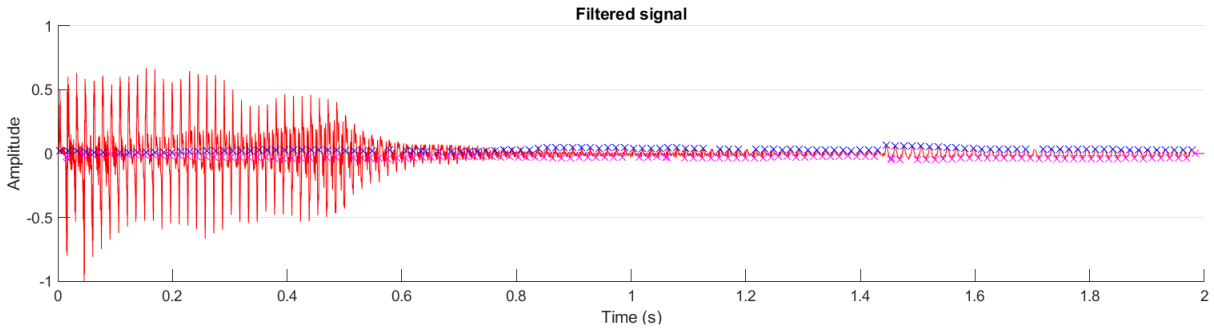


*Figure 6.23 – time-domain plot of the processed signal, using a C2 plectrum-sounded bass guitar note as the source and a piano C2 note as the modifier. Note the "bump" around 1.4 seconds.*

This process was repeated for a slapped bass guitar note to trial the processing on a note with a pronounced transient. Some tweaking of system parameters was required to produce an acoustically acceptable signal, namely reducing the degree of filter curve smoothing. With these adjustments made, the system produced a signal of comparable quality to the previous examples, with some notable quirks. In fig. 6.24, it can be observed that once again the system effectively managed to reshape the waveform to match the modifier signal envelope. However, it can also be seen that the pronounced transient of the source signal is crushed during the envelope shaping process resulting in a noise-like burst of sound at the beginning of the processed signal rather than a percussive hit. That is not to say all acoustic properties contained in the bass guitar transient are lost, but dramatically altered to the point where it is not immediately clear that the unprocessed signal was a slapped bass guitar note. Furthermore, qualities of the piano transient and note envelope were imprinted on the timbre of the signal just as in the previous examples. Overall, the effect of the signal processing on the slapped bass guitar note has provided an intriguing example where the acoustic properties of the processed signal lie somewhere between that of the source and modifier. Although the processed signal still bears considerable resemblance to the source signal and it could not be mistaken for a piano, this particular example stands as a promising proof of concept of sorts, demonstrating that significant and

meaningful replication of timbral qualities can be executed through manipulation of a signal in the time and frequency domains.

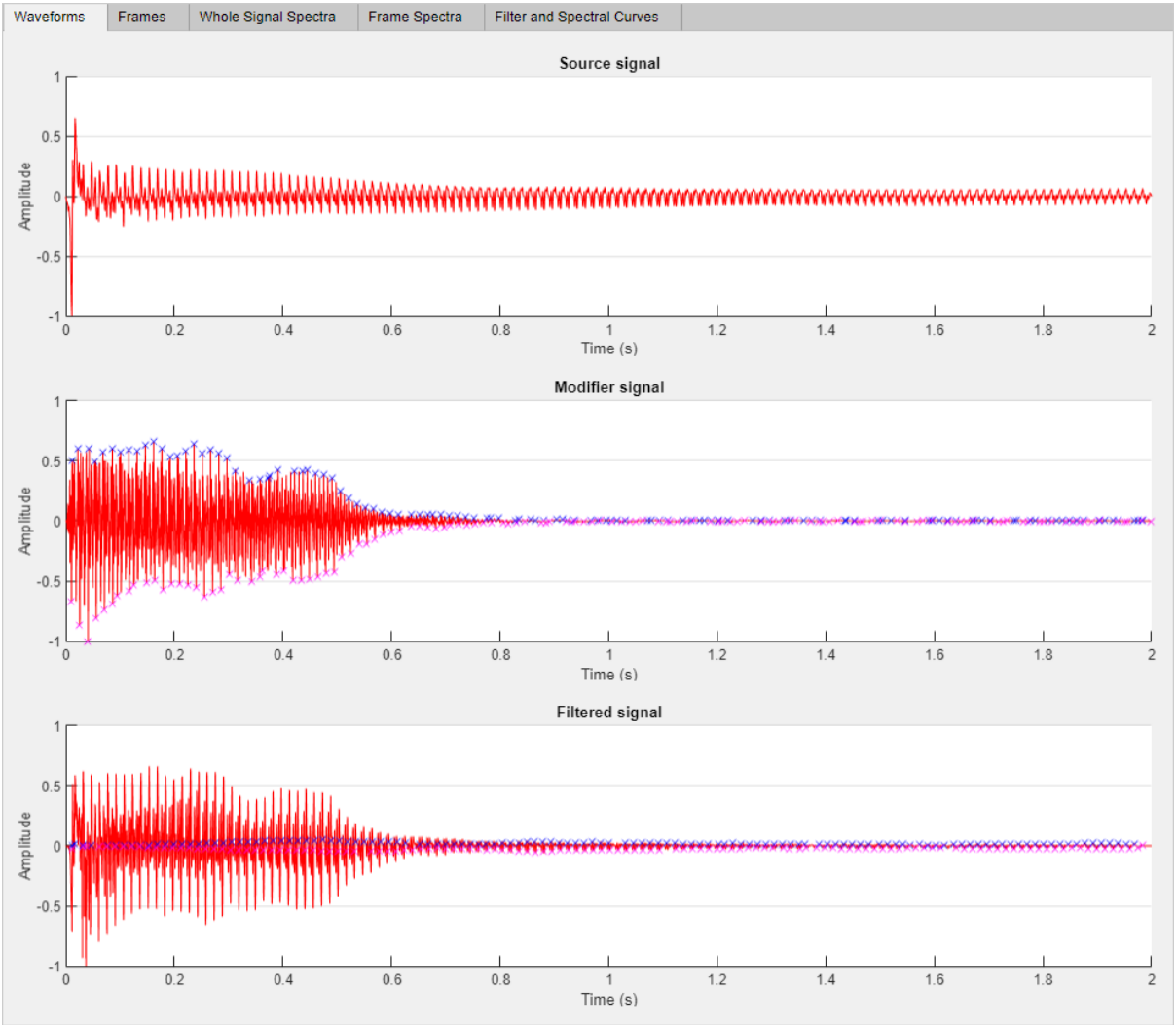


*Figure 6.24 – time-domain plots of the slapped C2 bass guitar note, the C2 piano note and the outcome signal. The "crushing" of the slapped bass transient can be observed at the start of the processed waveform.*

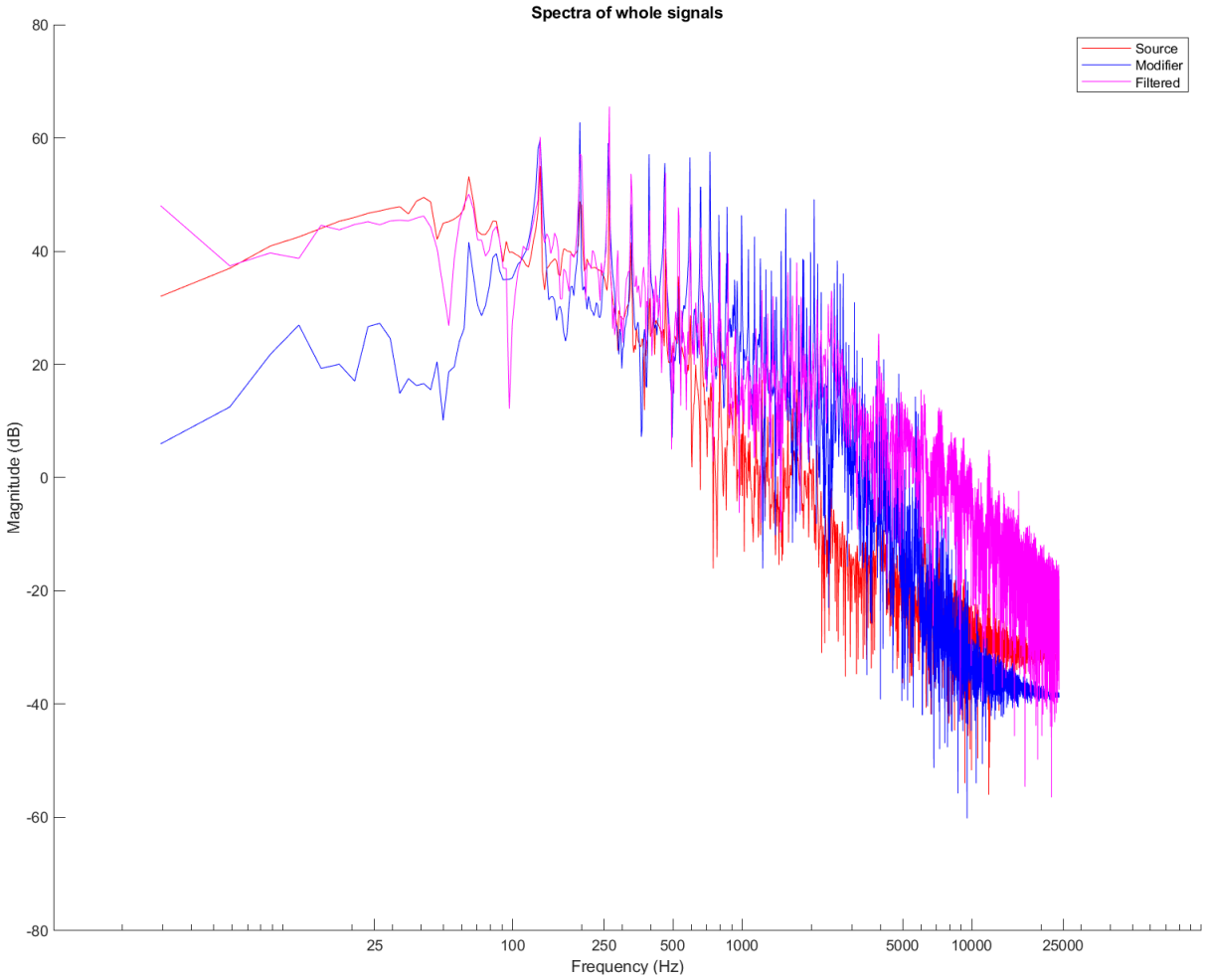


*Figure 6.25 – magnitude spectra of the signals in 6.25. Characteristic high-frequency distortion is present here as in all other previous examples, whilst the rest of the spectra is matched to the modifier with reasonable accuracy.*

Processing was repeated, this time using samples recorded an octave higher than the previous examples. Unlike the previous examples, results here were less encouraging and subject to more extreme acoustic artefacts. For an acoustically pleasing signal to be produced, filter curve smoothing between frames needed to be drastically reduced to combat pops and crackles orders of magnitude higher than the musical content of the signal. However, by decreasing the degree of intra-frame smoothing, each successive filter curve generated by the system deviates more in its shape from that of its predecessor. Consequently, when the filtered signal is reassembled, a greater amount of distortion is introduced to the signal caused by the overlap-add formulation used for signal assembly

used in conjunction with discontinuous frames of data. Fig. 6.26 illustrates the relatively poor effect of the envelope matching system on the higher note; not only have several audible pops been introduced towards the end of the signal, the overall shape of the processed signal envelope is only faintly reminiscent of the modifier envelope. This was remedied somewhat by making the sampling range for envelope estimation finer, as portrayed in fig. 6.27, where local maxima were located within a frame size of 375 samples rather than the 750-sample span used for 6.26. Audible pops were also eliminated by refining the sampling span, although the overall shape of the note is still lacking in comparison to the modifier signal envelope. The RMS loudness of the processed signal was also closer to that of the source, with the source RMS being 0.167, the modifier RMS being 0.128 and the processed RMS being 0.151.

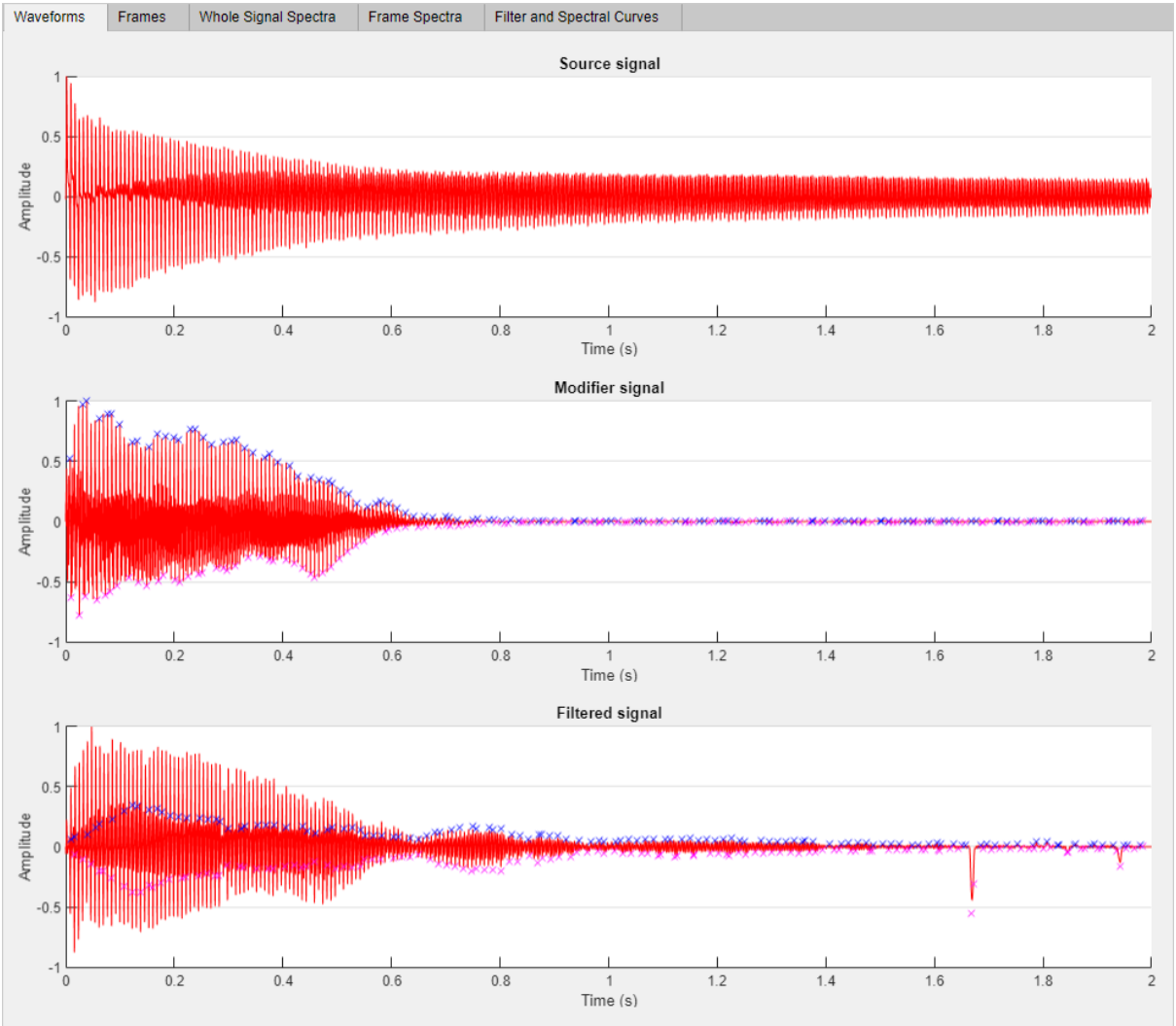


*Figure 6.26 – time domain representations of a C3 bass note sounded fingerstyle, a C3 piano note and the outcome processed signal. Audible pops can be observed towards the end of the processed signal.*

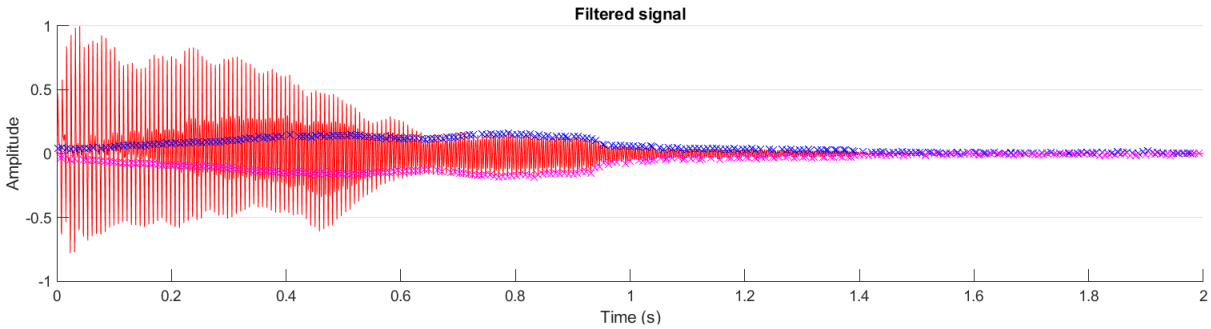


*Figure 6.27 – processed signal produced in the same means as 6.27, but with a finer sampling range for peak detection to occur.*

Spectrally, the results of the system on the higher notes appears quite impressive in comparison to the earlier examples. It can be observed in fig. 6.28 that the processed signal

spectra fit that of the modifier across the entire length of the frequency spectrum with less apparent high-frequency distortion. Acoustically however, the processed signal appears duller than it was when applied to samples recorded an octave lower. Very few timbral qualities of the piano have been carried over to the bass guitar overall. The note attack has been replicated quite faithfully although the manner of decay sounds little like a note sounded by a piano, but more like a heavily processed bass guitar. What is of note here is that the performance of the filter and envelope matching algorithms were reversed when compared to the previous C2 examples – where they struggled to replicate spectra accurately, the system appears to handle the spectra of higher notes with greater accuracy. Likewise, whilst the C3 examples managed to replicate note envelopes faithfully, the system is struggling to do the same with notes an octave higher. When the acoustic outcome of the signal is considered however, it is the lower examples that bear more resemblance to their modifier signals rather than the C3 examples. This suggests that the envelope of a note is of greater importance than spectral detail when it comes to human perception of timbre.

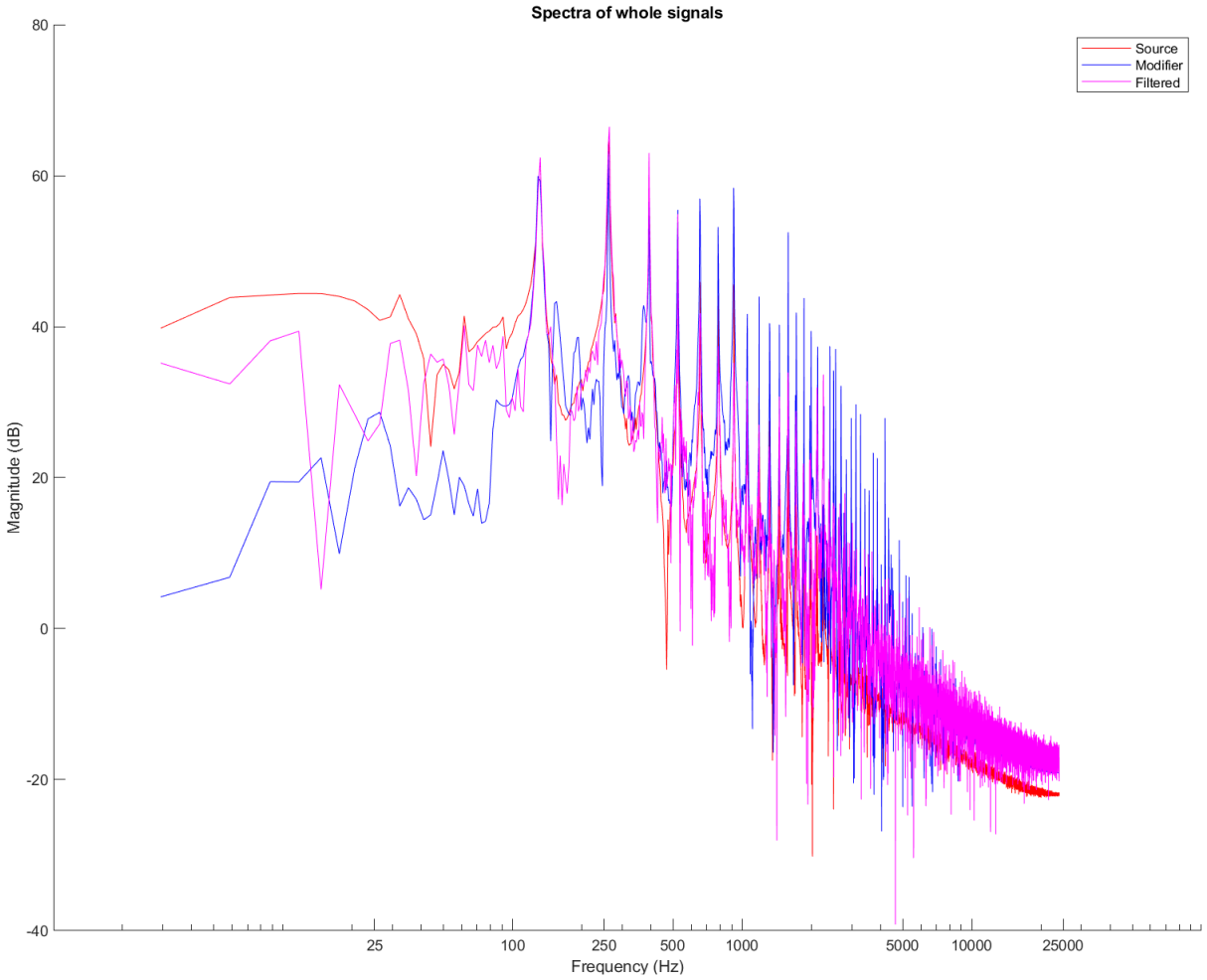


*Figure 6.28 – spectra of a C3 fingerstyle bass guitar note (source), C3 piano note (modifier) and the processed outcome. The processed signal can be seen matching the shape of the modifier spectrum much more closely than in previous examples.*

Once again, the fingerstyle bass sample was substituted for that of the same note sounded with a plectrum and processing was reapplied. Just like the example pictured in fig. 6.28, there was little difference in the quality of results between the higher-register fingerstyle and plectrum-sounded samples. There was little to no discernible difference in acoustic plausibility between the C3 fingerstyle and plectrum samples when affected by the piano modifier signal, so the source signal was exchanged for a note sounded by popping. When the signals are viewed in the time domain, the popped note can be compared to the slapped example by the presence of a powerful transient defining much of its timbral character. System performance on the popped note was perhaps the strongest of all the

higher-register examples. As illustrated in fig. 6.29, the envelope matching process produced accurate results as the processed signal has taken on the envelope of the modifier reasonably well, although not as well as in the lower C2 examples. A noticeable spike can be seen in the negative pole of the processed signal just before 1 second elapses and another can be observed just after the transient has dispersed. Like the slapped example, the transient of the pop has been crushed to fit the modifier envelope although the acoustic effect here is much less desirable. The timbral quality of the pop is still largely present but has been exaggerated, making the processed note sound noisy and unpleasant. Despite this aesthetic consequence, it can still be argued that the system performed adequately in this instance in comparison to other samples auditioned in this register. Signal spectra for the popped example are portrayed in fig. 6.30. It can be observed that spectral performance was poorer here than in previous tests, although the overall timbral blending effect intended by the system is more pronounced in this experiment than other C2 examples. Again, this is an allusion to the importance of note envelopes in perception of timbre over spectral detail in a situation such as the one demonstrated here.

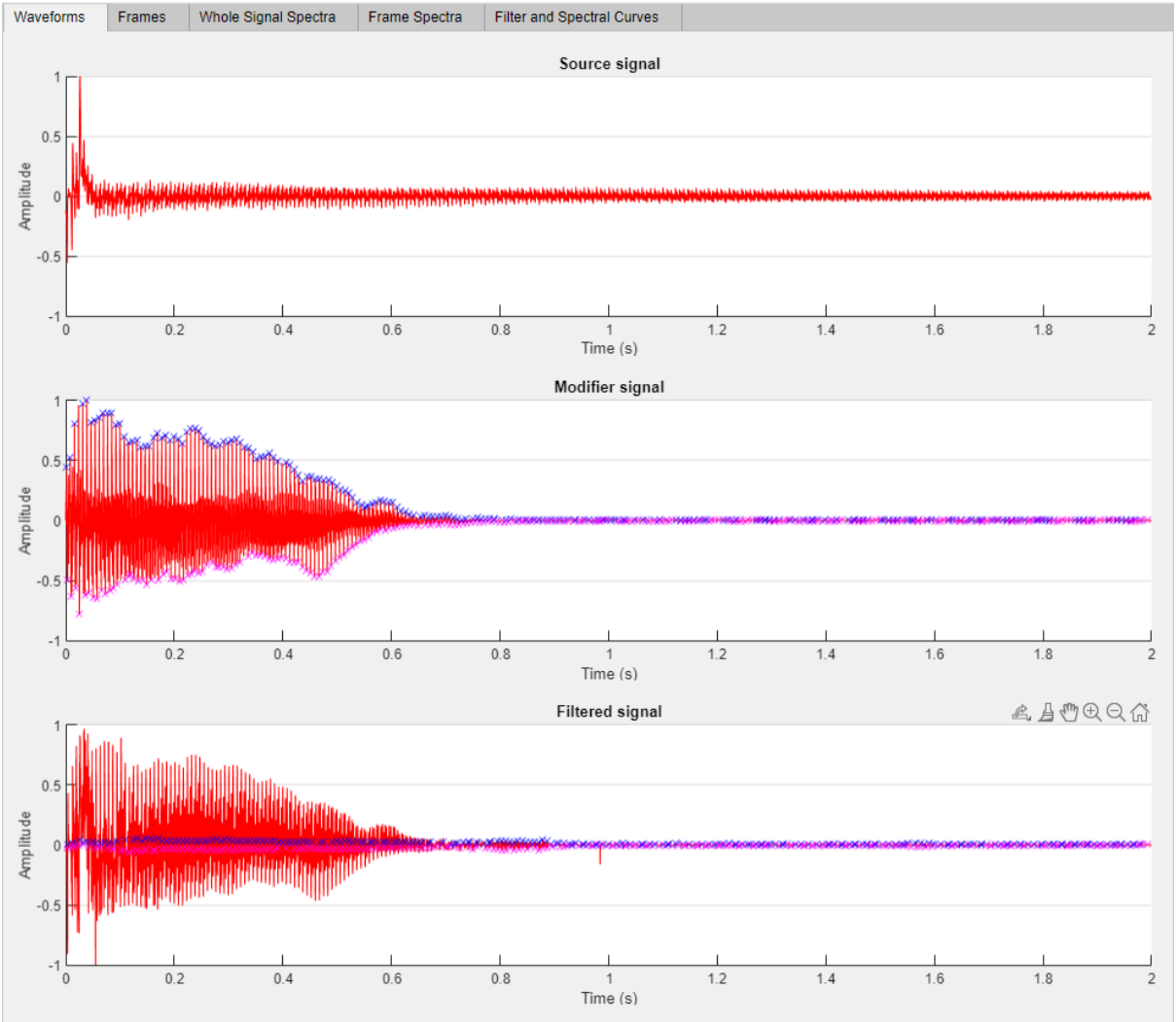


*Figure 6.29 – time-domain representations of a popped C2 bass note (source), a C2 piano note (modifier) and the resulting processed signal.*

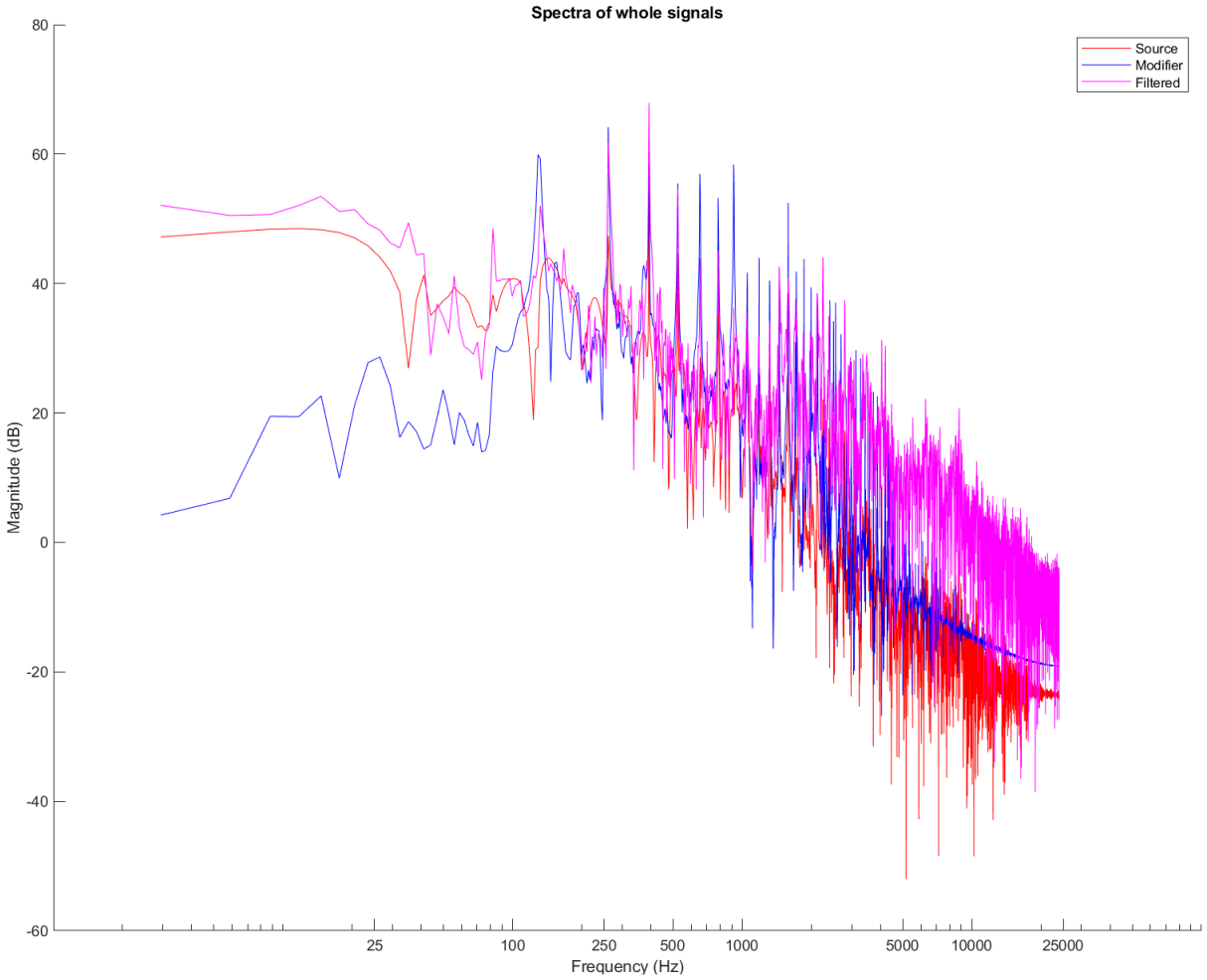


*Figure 6.30 – spectra of a popped C2 bass note (source), C2 piano note (modifier) and the processed signal produced. Spectral matching here is far less accurate than other examples in this register.*

Longer musical passages of bass and guitar signals were then processed and analysed. A recording of a one-octave G major scale played on the bass guitar was loaded into the system as the modifier whilst a piano playing the same scale was used as the modifier. The root note for each of these scales was G2. The challenge faced by the system here is processing notes in series, rather than an isolated single note, which makes for a more challenging endeavour as note onsets may arrive at any point in the signal and present variable fundamental frequencies. This greatly increases the complexity of the signals to be analysed and processed and can indicate system performance over an extended period of musical expression.

Envelope matching performance was found to be acceptable when applied to the example of the G major scale. In fig. 6.31, it can be observed that each note in the source signal, occupying roughly half a second per note, have been shaped individually by the algorithm to mirror the shape of the modifier envelope post filtration. Several discrepancies can be observed in the processed signal however; note transients appear to be exaggerated, perhaps caused by misalignment of envelope matching coefficients. By observing the reported RMS loudness for each signal, it can be confirmed that the envelope matching process has nonetheless matched the modifier signal energy well. RMS loudness reported for the source signal was 0.19, the modifier was 0.211 and the processed signal was reported as 0.215.

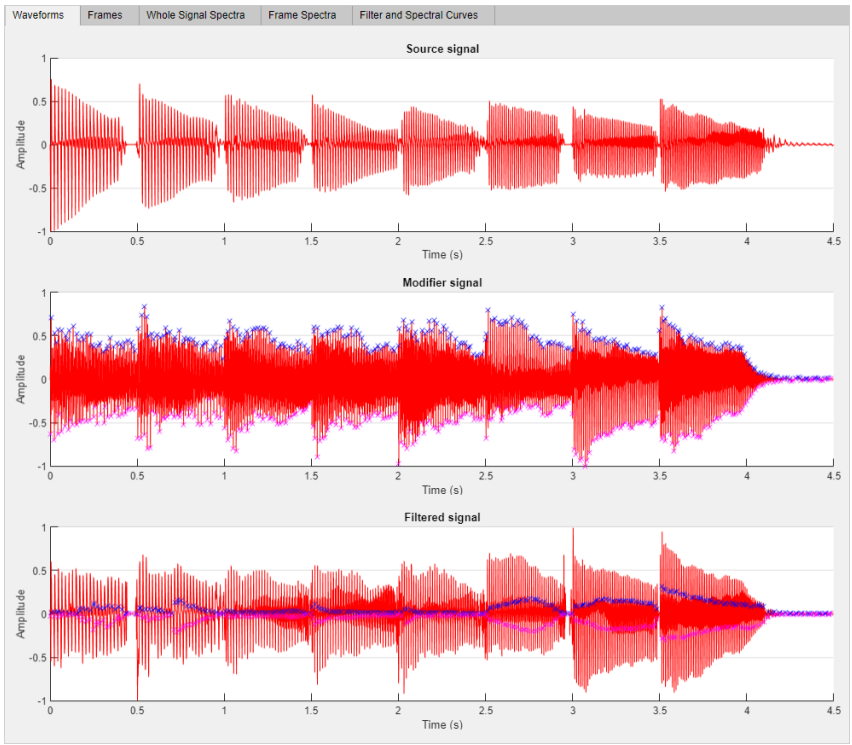


*Figure 6.31 – G major scale, one octave, performed fingerstyle on bass guitar (source) and piano (modifier). The processed signal is also pictured.*

Performance of the matching filter on the spectral components of the signal was peculiar. Spectral results, pictured in fig. 6.32, are encouraging upon first sight. Conforming with results obtained from prior investigations, the processed signal has managed to match the curve of the modifier signal reasonably well, with areas for improvement identifiable around and past 2kHz as the spectrum deviates from the ideal curve. However, when the audio sample is replayed, the dynamic filter can be heard “lagging”, producing a morphing timbre that bulges and swells in its spectrum over time. The effect is a warm, rounded timbre towards the beginning of the sample which becomes more brittle sounding as time progresses. This quality cannot be credited to a change in timbre of the modifier signal over time, which remains consistent throughout the whole sample. Instead, this behaviour can be explained by the frame size used for matching filter estimation and the effects of smoothing the produced filter curves. Adjusting each of these variables independently has little effect on system performance. When both are lowered significantly however, spectral results worsen despite the theoretically quicker system response time. Acoustically, the signal becomes muddied as the filter curve changes drastically between frames, losing much of the musical information of the source signal in the process. This behaviour suggests the system cannot be expected to perform exceptionally no matter the extent to which parameters are adjusted. There appears to be a limit to the quality of the outcome signal. Attempting to compensate for unsatisfactory acoustic results by forcing the system to operate on smaller datasets or reducing filter curve deformation from smoothing destroys the musical characteristics of the source signal entirely in the process. Even poorer spectral results were observed when the source and modifier inputs were transposed an octave higher as illustrated in fig. 6.33. This is in line with the middling results observed for the lengthier musical signal and the degradation in output signal quality consistent with using higher pitched samples.

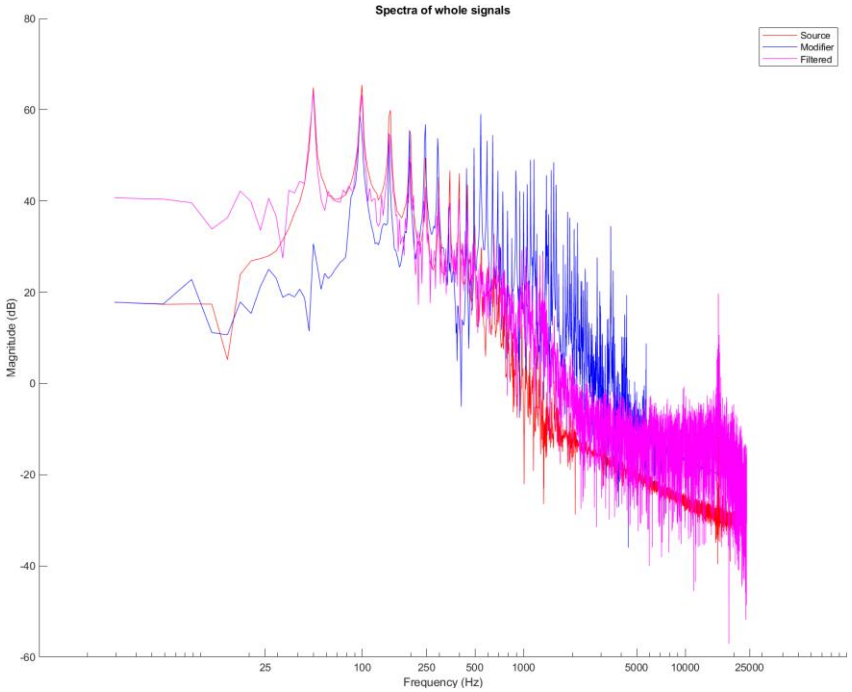


*Figure 6.32 – spectral results of 6.31.*

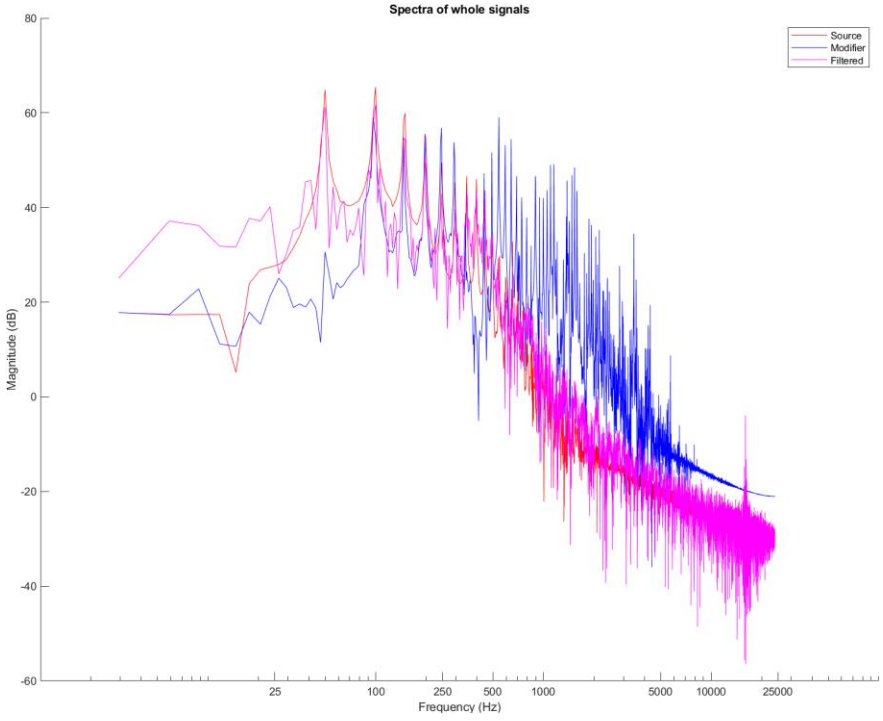


*Figure 6.33 – same as 6.32, except all signals are transposed one octave higher.*

## Bass guitar with synthesizer modifier signal

The experiments were then repeated using a synthesized signal as the modifier, in place of the piano signals used for the previous examples. Using synthesized signals

presents some new challenges for the system to overcome. In comparison to the piano signals, the synthesized “pluck” sounds being used as the modifier signals here have much simpler timbres. Comparatively, the piano signals used previously are spectrally rich with a great deal of harmonic activity that changes dynamically as the note decays. The synthesized notes here use a single square wave oscillator in contrast to the complex acoustic systems that generated the piano waveform. A stark difference in spectral composition is the near absence of even-numbered harmonics in the synthesized signal as a perfect square wave is composed solely of odd harmonics. Secondly, the envelope of the waveform is more uniform than that of the piano signals. Recall the longer piano sample used as the source signal in fig. 6.31. It can be observed that each note is asymmetric in shape and has a differing local maximum, whilst the synthesized signals used here will have consistently symmetrical note envelopes and local maxima. Logically, this suggests that the envelope matching system may perform well on the synthesized signals given their simpler envelopes. In practice however, this quality may be negated by the simpler shape of the waveforms. As the envelope detection system is in no way predictive, simpler envelope shapes are no more difficult for the system to estimate than the most complex. The relatively simple waveform shape could detract from the envelope detection system as there are fewer prominent peaks for the algorithm to detect. If the envelope detection sensitivity is set to a number that does not reduce to an integer number when divided by the fundamental frequency of the note, the algorithm is subject to detecting multiple or no peaks per oscillation. This property does not present a problem when the subject signal is sufficiently complex in its shape, such as the piano signals. For a relatively simple shape, such as a square wave-derived synthesized note, the system begins to lose accuracy, increasing in severity as the signal approaches sinusoidal. Fig. 6.34 illustrates the same synthesized signal with detected peaks marked with crosses. The bottom example, with a smaller sample range for peak detection, produces a considerable number of errata, inaccurately representing the signal envelope.

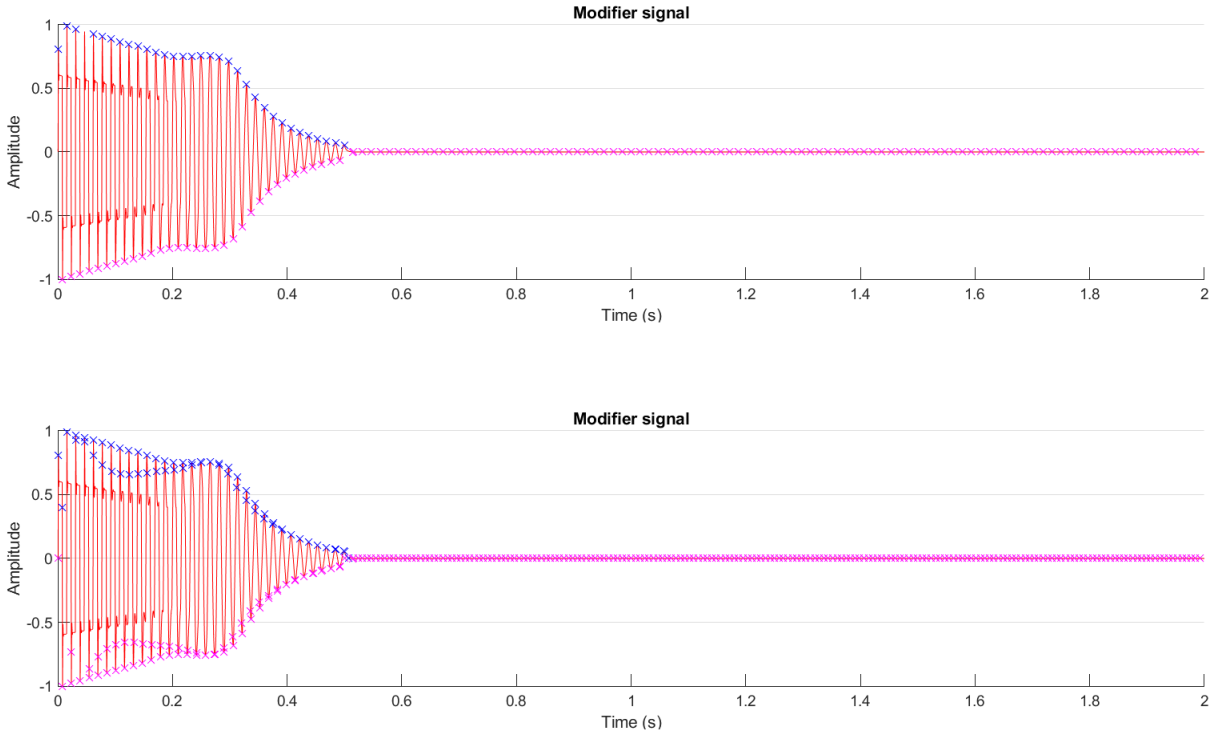


*Figure 6.34 – peak detection on a synthesized pluck waveform. The top example uses a peak detection sample width of 750 whilst the bottom uses a width of 375.*

Finally, the synthesized notes have a clear, technically definable timbre that should be replicated in the processed signal. The lowpass filter applied to the square wave provides a knocking, resonant quality to the synthesized notes as the cutoff is reduced over the decay of the note. This behaviour can be observed in the signals themselves as the lowpass filter causes the signal to become more sinusoidal as the note decays. This is represented in fig. 6.35.

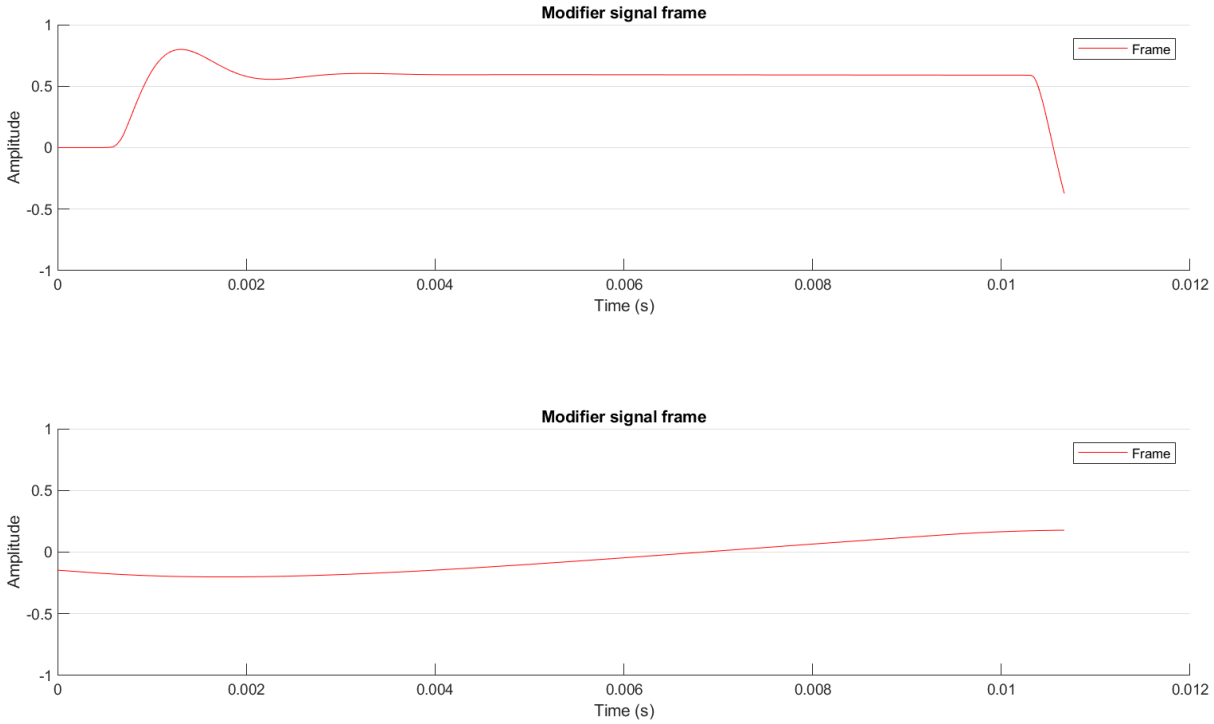


*Figure 6.35 – 512-sample frames of audio from the synthesized pluck signal. The top frame is taken from the start of the signal, capturing the instant the note is initiated. The bottom frame is taken several hundred frames later, after the note has decayed considerably.*

Like previous experiments, a C2 fingerstyle bass note was loaded as the source signal, this time with an accompanying C2 synthesized bass pluck and processing was applied. The lack of spectral richness in the modifier signals is immediately evident in the results produced by the system. Spectral results (fig. 6.36) were reasonable in this instance, with the processed signal spectrum lying roughly between the source and modifier spectra. Likewise, the results of envelope matching to the modifier signal (fig. 6.37) were acceptable. The system appears to have taken the shape of the modifier signal well until around 0.4s, when the synthesizer note had mostly decayed. The envelope matching system appeared to struggle with the particularly low-energy portion of the signal, a trend that will establish itself in the following examples. Despite scepticism about envelope detection on the modifier signal, the algorithm detected the peaks well with a sample width of 750 points. The actual shape of the processed signal is lacking in comparison to that of the modifier. This is likely due to the arrangement of the peak locations being misaligned; during peak detection,

locations for modifier and processed signal peaks are detected independently and may not necessarily align. During amplitude coefficient estimation, it is assumed that peaks do align, therefore if either signal produces a set of regularly spaced peaks (such as the synth pluck) it is likely that the accuracy of envelope estimation and matching will be reduced. This behaviour can also be mitigated by using a tighter sample range for peak detection; however, such an option is inappropriate for simple waveform shapes given the behaviour detailed previously.

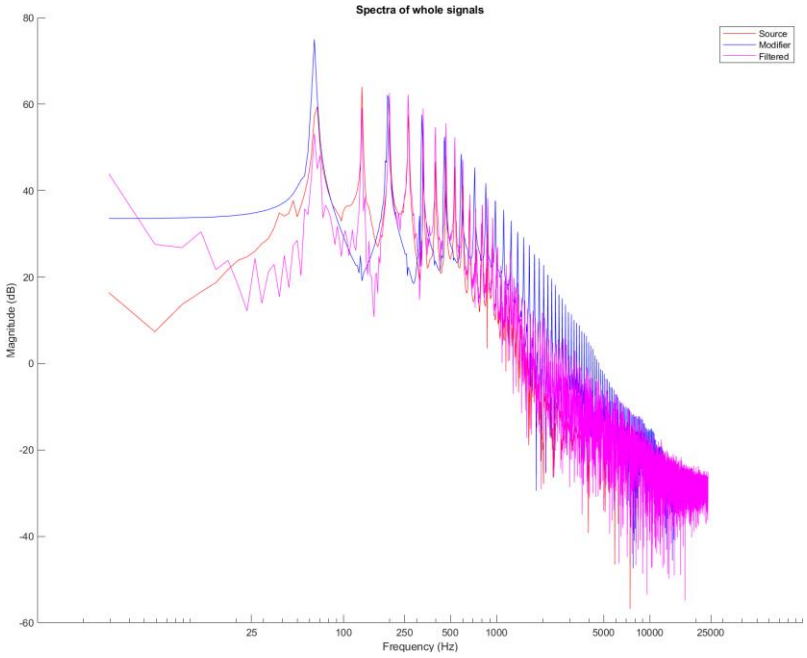


*Figure 6.36 – spectra of a fingerstyle C2 bass note (source), a C2 synthesized pluck (modifier) and the output signal produced.*

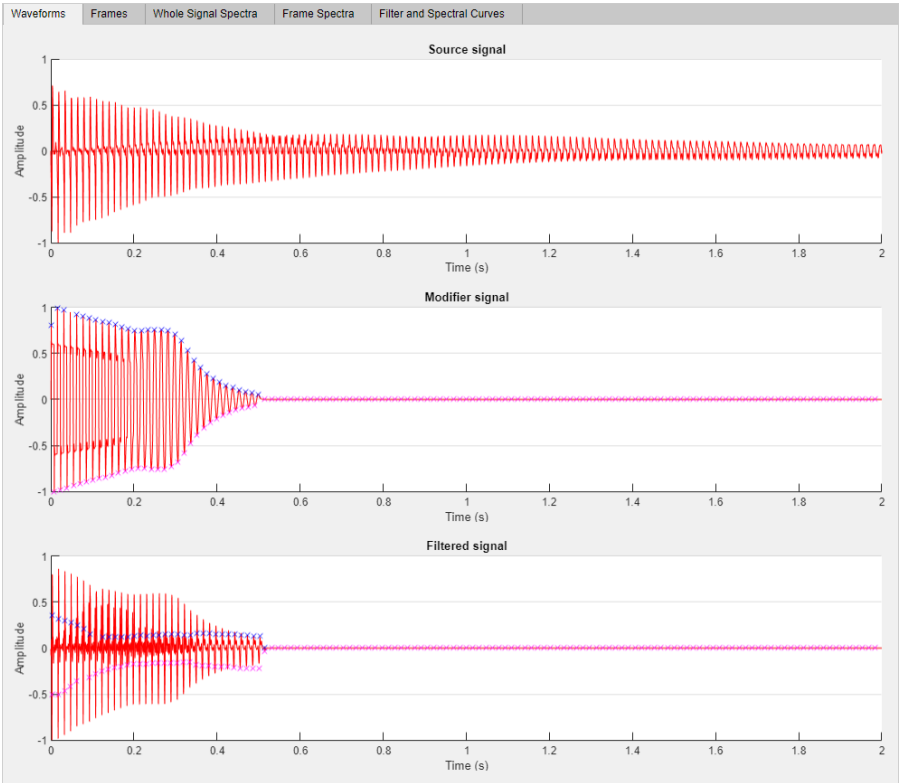


*Figure 6.37 – waveforms of the signals specified in 6.36.*

The fingerstyle bass note, loaded as the source signal, was exchanged for a note sounded with a plectrum. Results here were markedly poorer in both the time and frequency domains. Once again, the last moments of the note was not shaped properly by the envelope matching process leading to a steep (but not discontinuous) drop in signal power as the note is released. Fig. 6.38 pictures the waveform of the processed signal. Compared to the processed waveform in 6.36, it can be observed that the note envelope was generally matched to the modifier signal better than the fingerstyle sample was. On the other hand, spectral results were poor. Theoretically, the system should have performed much better than it did as the spectra of both the source and modifier signals are similar. In fig. 6.39, the processed spectrum deviates significantly from both the source and modifier spectra, ultimately resembling neither. This is most likely caused by the few source spectrum peaks that surpass the modifier peaks in the same area (around 250 to 2,000Hz). Therefore, despite the partials in the modifier signal being denser in that band than in the source signal, it is nonetheless recognised as having more effective band power than the modifier. Whilst

estimating filter coefficients, the system will produce an ideal filter curve based on the difference between these spectral envelopes and will incorrectly assume the source signal should be attenuated around that band, rather than boosted.  Fig. 6.40 portrays a typical filter curve produced for this plectrum-sounded example.

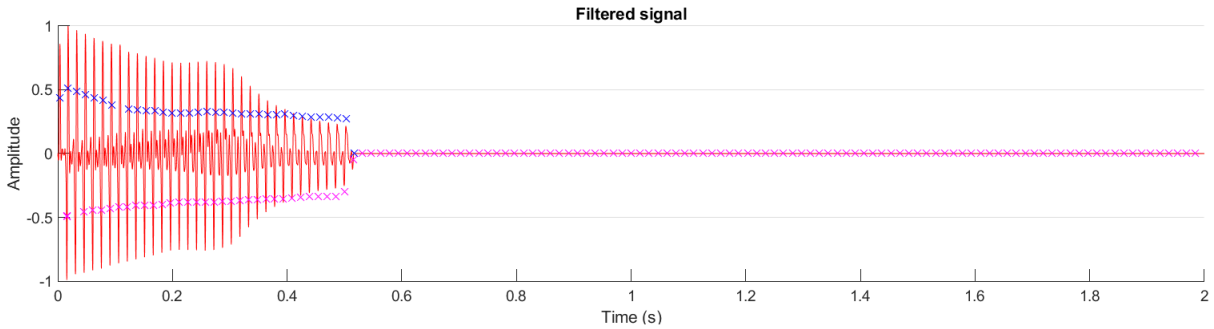


*Figure 6.38 – waveform of signal produced using a plectrum-sounded bass guitar note as the modifier.*

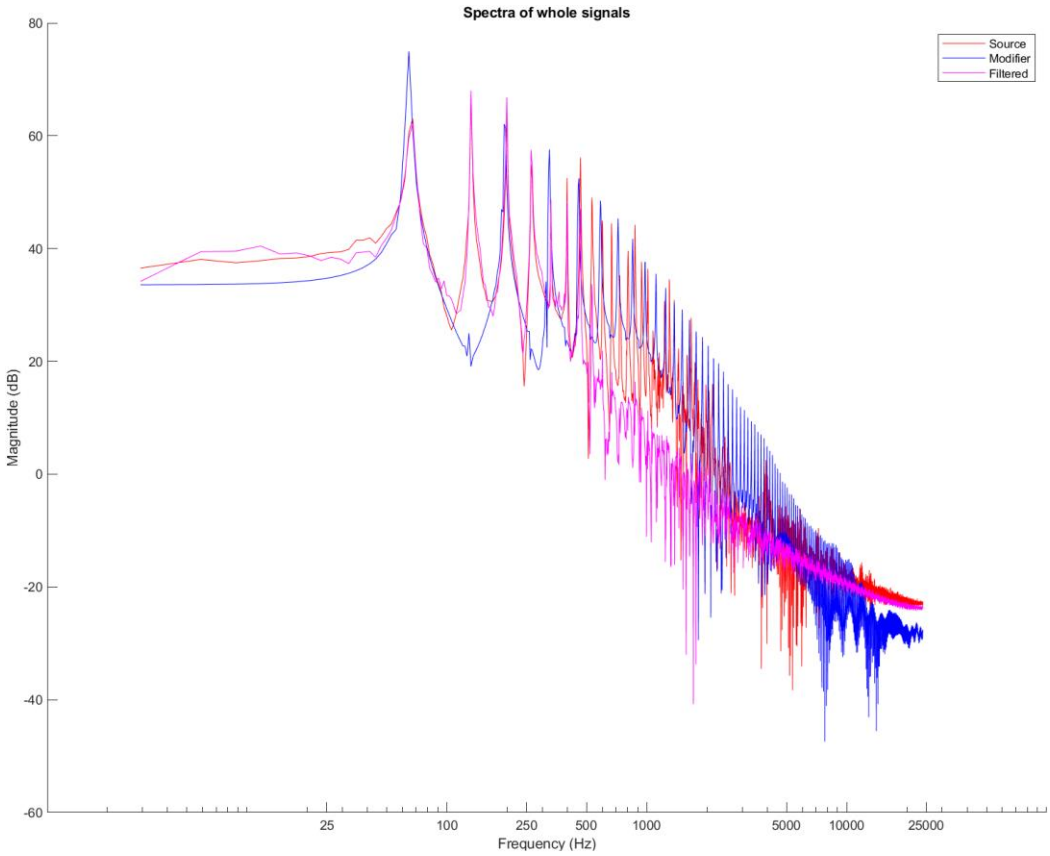


*Figure 6.39 – spectral results of the waveform produced in 6.38 overlaid with the plectrum-sounded C2 bass note (source) and the C2 synthesized pluck.*

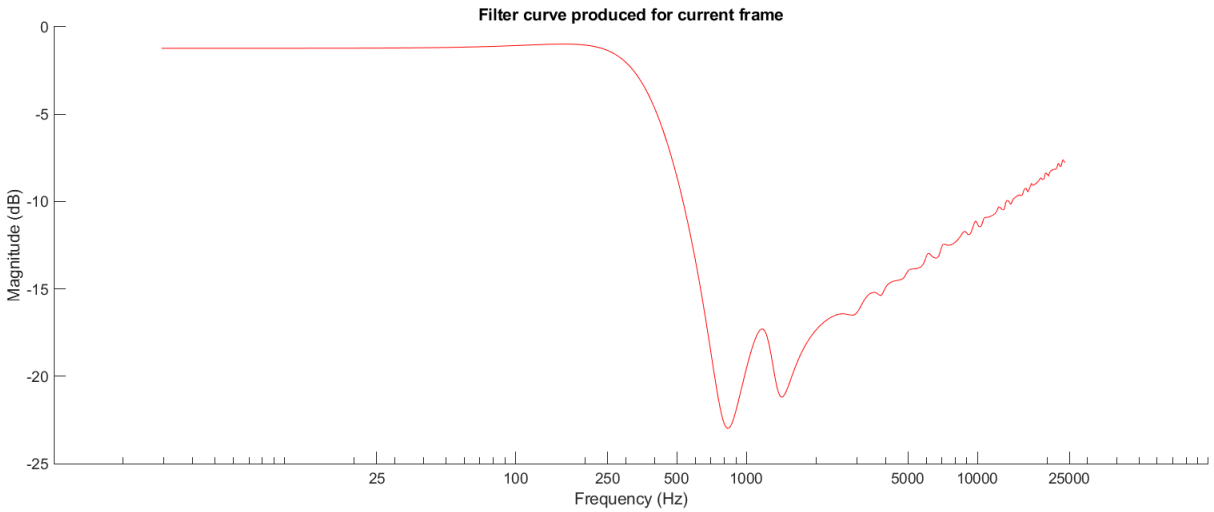


*Figure 6.40 – typical filter curve produced for the signals used in 6.38 and 6.39.*

Acoustically, both the fingerstyle and plectrum examples failed to replicate the unique timbral properties of the modifier signal onto the source. There was a general notion of timbral blending caused by the envelope shaping, although the dynamic filter failed to match the evolving timbre of the synthesized notes. As previously detailed, the modifier signal has a knocking timbral quality caused by the lowpass filter cutoff falling as the note decays. No comparable resonant quality was replicated by the filtering process when using a signal frame size of 512 samples and an intra-frame smoothing value of 75. The system was forced to react more quickly to signal data by reducing the frame size to 256 samples and the smoothing value to 30. As expected, acoustic results were worse as computational accuracy fell off due to the limited data that can be retrieved from the smaller frame size. Reducing the degree of intra-frame smoothing, forcing the filter curve to deviate more between frames, also served to further deform the waveform shape making envelope matching difficult. The waveform of this signal is pictured in fig. 6.41, whilst its spectrum is represented in 6.42.

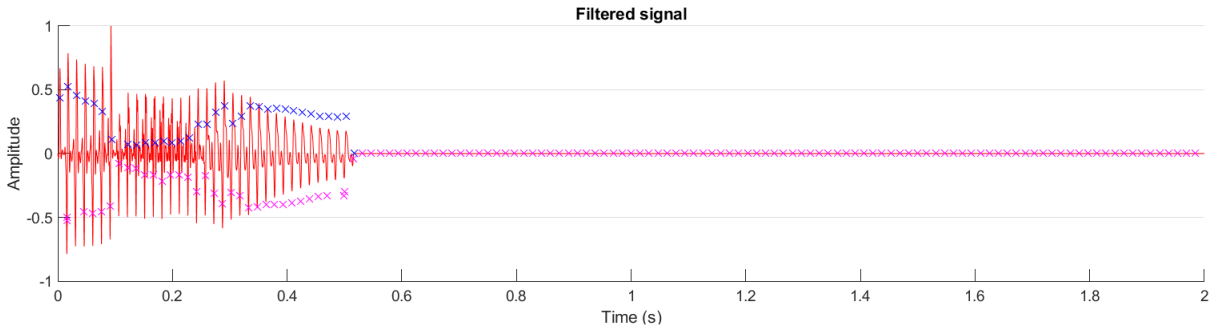


*Figure 6.41 – processed waveform produced using a frame size of 256 samples and an intra-frame smoothing size of 30.*

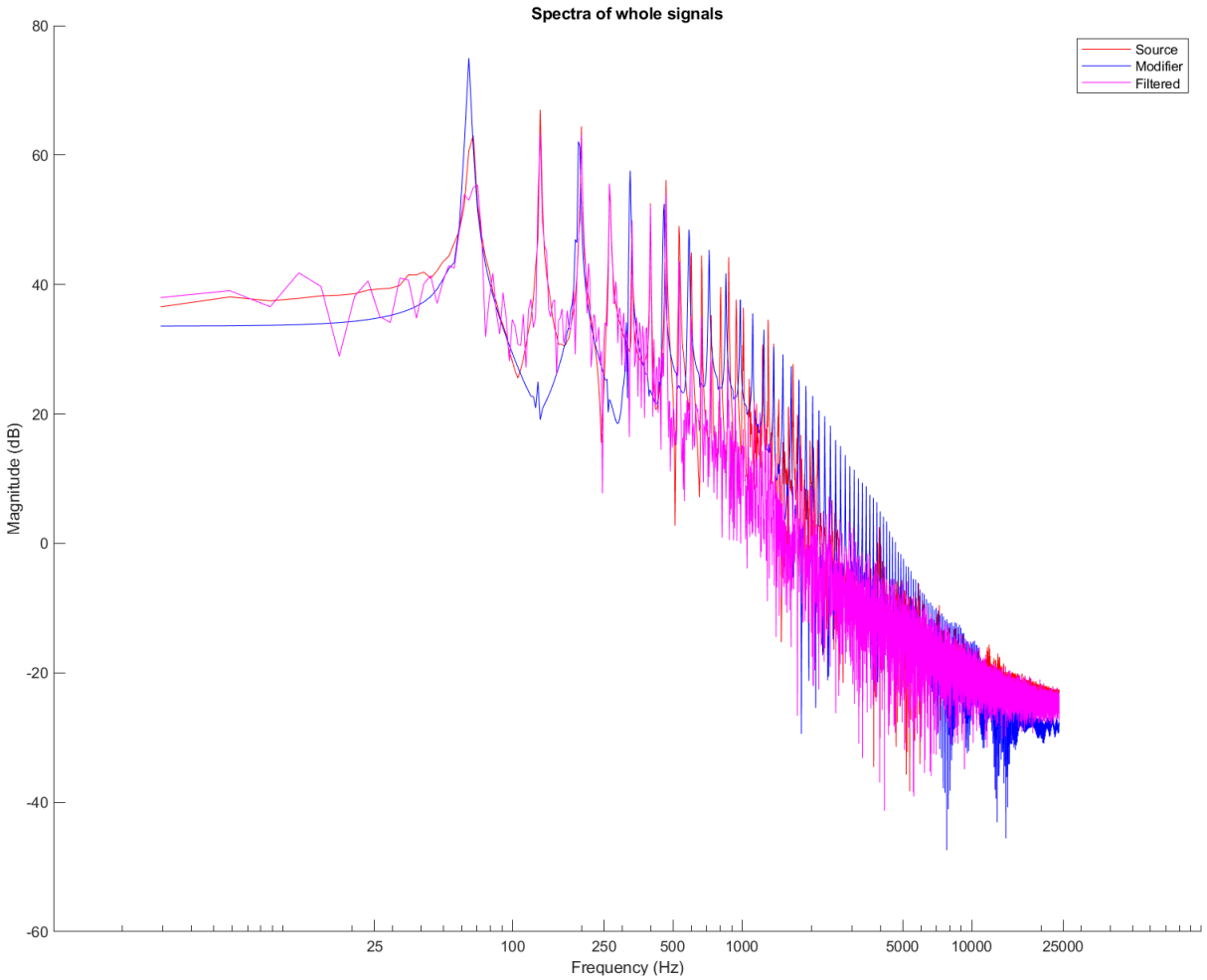


*Figure 6.42 – spectrum of the waveform pictured in 6.42, overlaid with the spectra of a C2 plectrum-sounded bass guitar note (source) and C2 synthesized pluck (modifier).*

A slapped C2 bass guitar note was then loaded into the system as the source signal and processing was reapplied. As illustrated in fig. 6.43, the matching envelope system

performed on par with the previous examples in this section. However, the transient of the slapped note was destroyed in the process, reflected in its spectrum (fig. 6.44). The loud, noise-like impulse forming the transient of the slapped note is forcibly reshaped to the point of distortion. Fig. 6.45 represents the same spectra, but with no envelope shaping applied to the output signal after filtering. Whilst the lack of envelope matching provides an inherently poor representation of timbral blending, this plot does confirm that the distortion present in fig. 6.44 is caused by the matching envelope process.

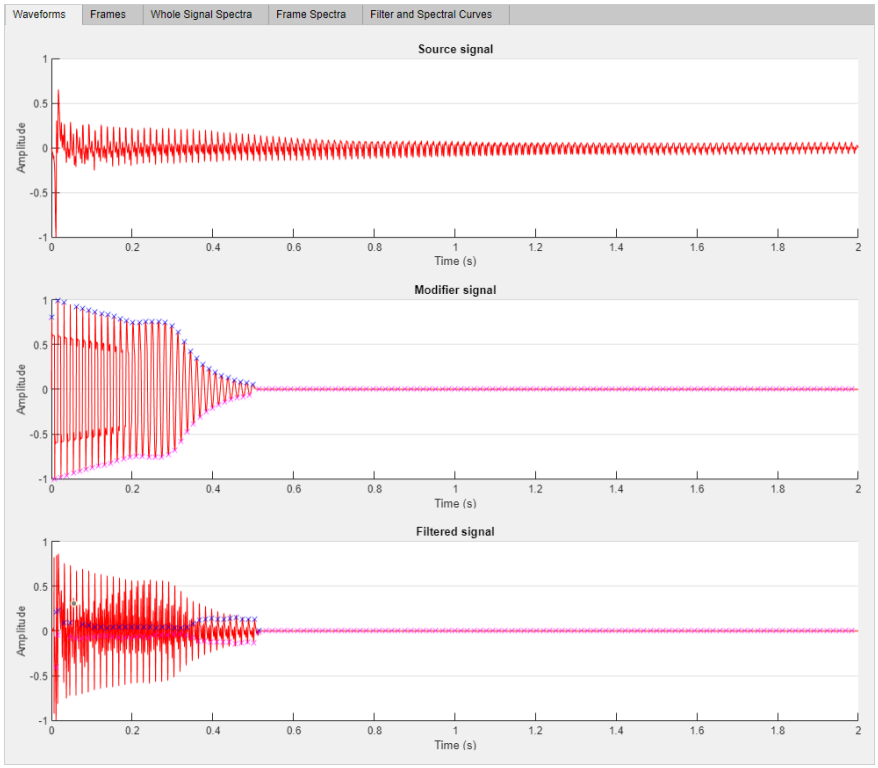


*Figure 6.43 – waveforms of a slapped C2 bass note (source), C2 synthesized pluck (modifier) and the processed signal.*

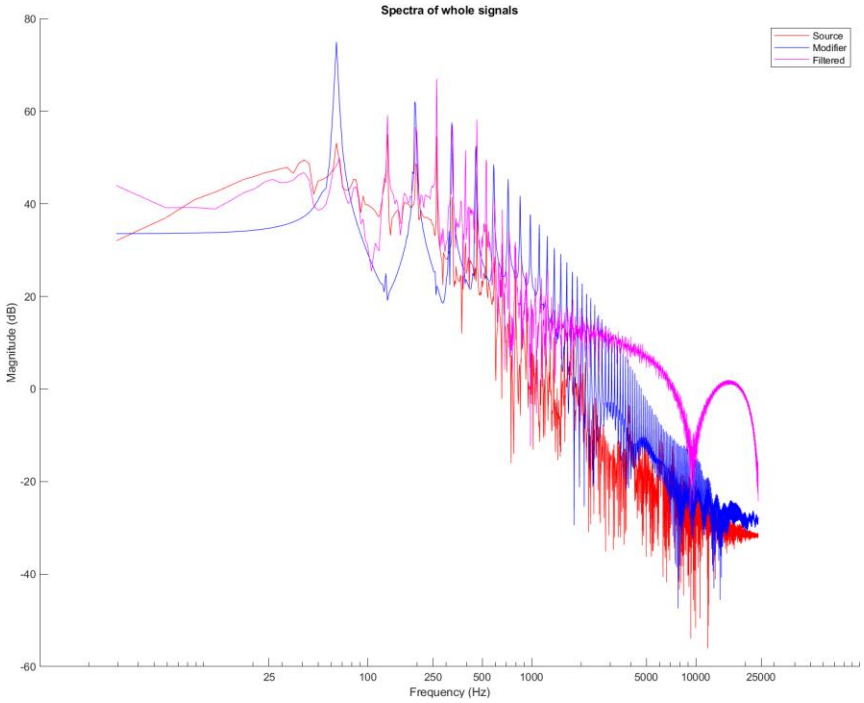


*Figure 6.44 – overlaid spectra of the signals detailed in 6.43. The spectrum of the filtered signal is greatly deformed by the aggressive time-domain envelope shaping.*

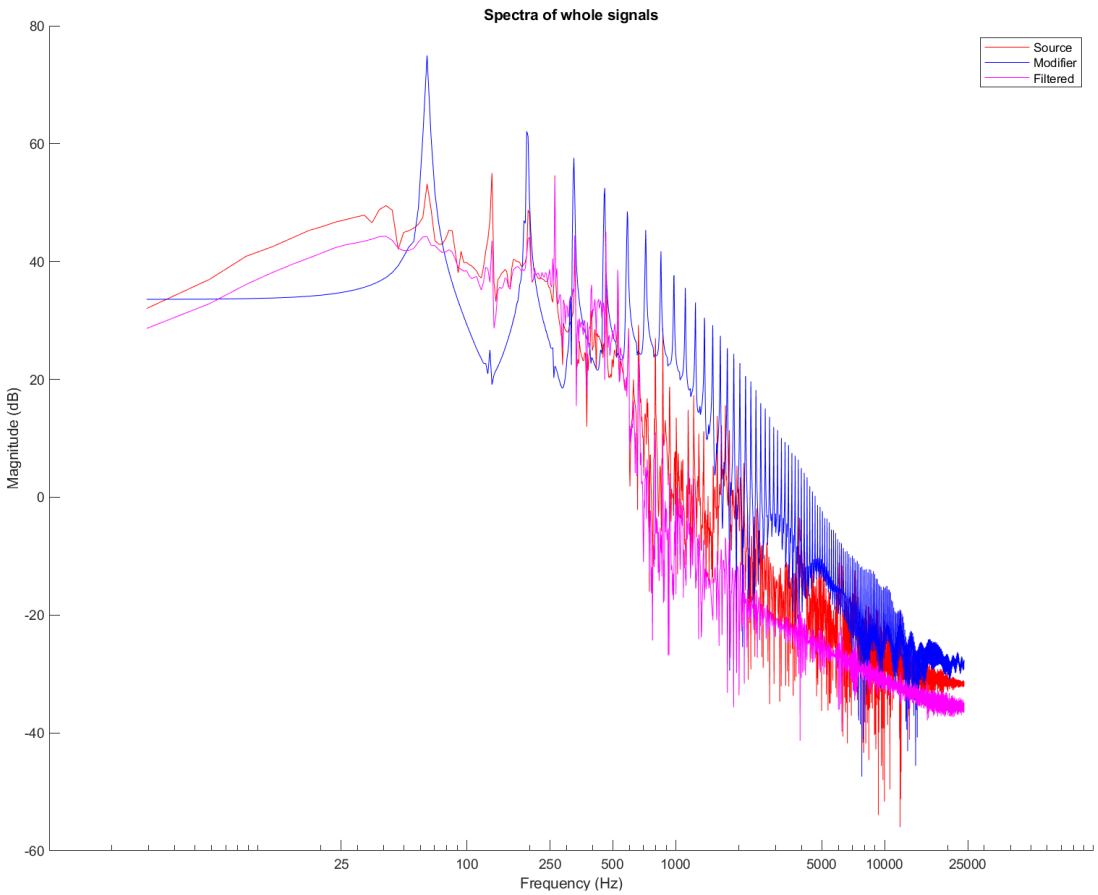


*Figure 6.45 – spectra produced using the same processes as 6.43, but with envelope matching disabled. The filtered signal no longer appears deformed.*

Acoustically, both the fingerstyle and plectrum examples failed to replicate the unique timbral properties of the modifier on the source signal. The vague notion of timbral blending is present given the decent performance of the envelope matching algorithm. However, the system was unable to act fast enough to replicate the travelling filter cutoff of the synthesized signal. The slapped signal provided the closest representation of the modifier signal timbre, despite the distortion introduced by the envelope matching algorithm. Upon closer analysis, this could be due to the source signal already containing a prominent transient which, when distorted in the process of envelope matching, somewhat replicates the timbre of the synth pluck. As this behaviour is dependent on the source signal and, given the performance of the system on the other examples here, it is fair to say that the system produced consistent, middling results on all the examples here. These observations appear to reinforce the earlier speculation on system performance concerning simple synthesized signals. Furthermore, it supports the observation made when analysing the results concerning piano signals, in that there is a limit to the speed at which the filter can operate. Reducing the sample size of frames and increasing the precision of filter curve estimation beyond 512 and 40 samples respectively appears to degrade results to the point where any improvements in system response time are negated.

The experiments were repeated using samples one octave higher (C3). Contrary to the previous results of testing higher-pitched samples, the system performed better on average than it did on the C2 samples. As can be seen in fig. 6.46, the envelope matching system performed well on the higher-pitched samples, capturing the shape of the waveform well. Some errors in estimation can be seen towards the end of the processed note, likely due to the possibility of peak misalignment when estimating envelope amplification coefficients. These errata occur around wide spaces in the peaks, lending credence to this theory. Unlike the previous C2 examples, the envelope matching process has captured the final decaying moments of the note accurately, tapering it to a point just like the modifier signal. Spectral results (fig. 6.47) were mixed; the system appeared to affect the spectrum

of the source signal very little at zero, with matching accuracy improving as the axis approaches the Nyquist frequency. For consistency, the C3 fingerstyle bass note was exchanged for one sounded with a plectrum and processing was repeated. System performance was consistent with established trends – the quality of the results pictured in figures 6.48 and 6.49 were of a similar standard to the fingerstyle examples.

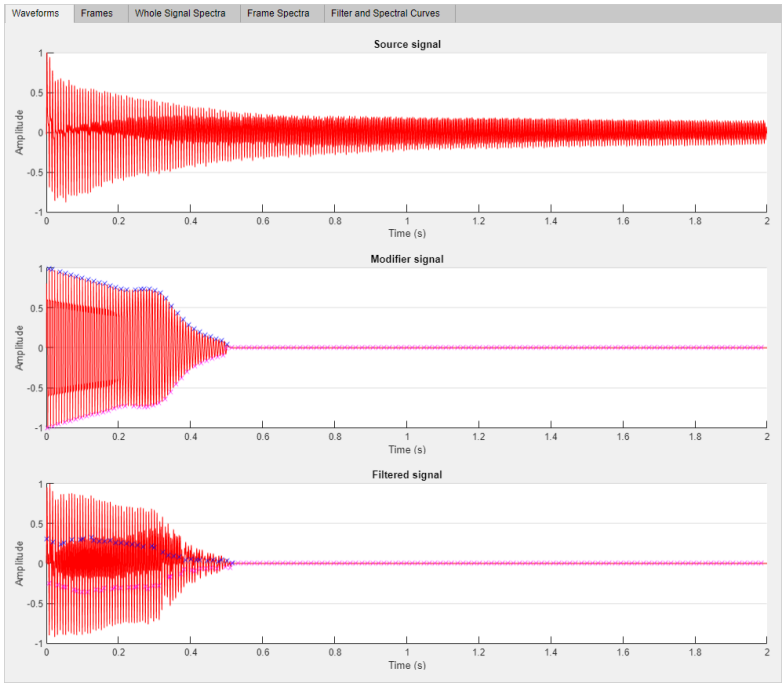


*Figure 6.46 – waveforms of a C3 fingerstyle bass note (source), C3 synthesized pluck (modifier) and the outcome signal.*

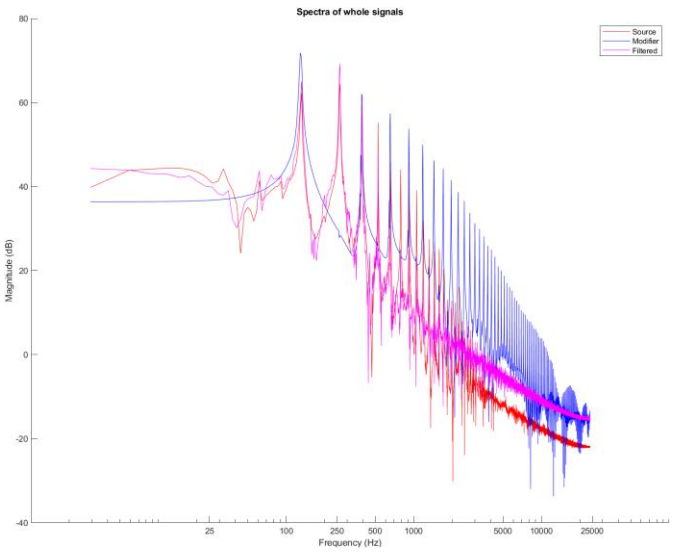


*Figure 6.47 – spectra produced for the signals described in 6.45.*

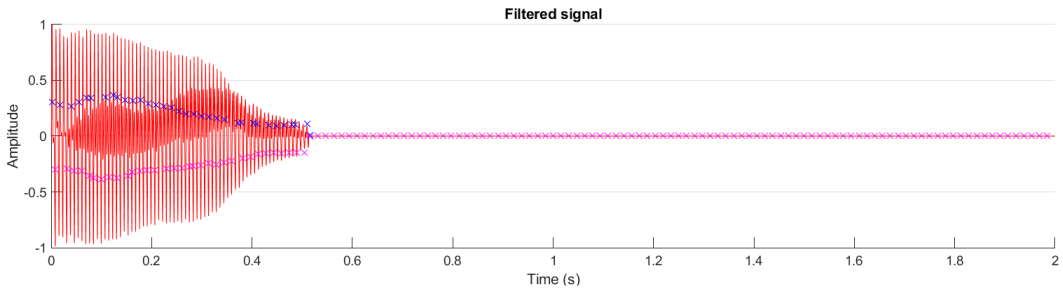


*Figure 6.48 – waveform of processed signal produced using a C3 plectrum-sounded bass note as the source signal.*

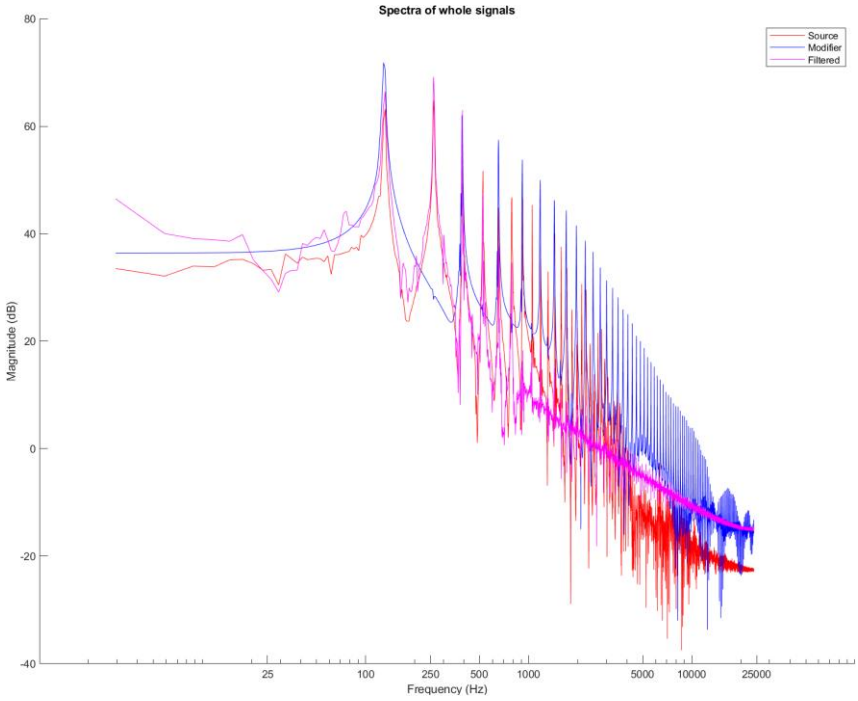


*Figure 6.49 – spectra of the C3 plectrum-sounded bass note (source), the C3 synthesized signal (modifier) and the processed signal.*

Finally, the source signal was replaced with a popped C3 note and processing was reapplied. The performance of the system here was poor, again caused by aggressive reshaping of a waveform that carries a prominent transient. In the time domain, the waveform appears distorted over its lifespan with a sharp peak immediately as the note initiates (fig. 6.50). This spike at the start of the note contributes significantly to the distortion present in its magnitude spectrum (fig. 6.51). Just as in 6.45, this distortion is absent if

envelope matching is disabled. The actual shape of the note envelope is accurate aside from the initial spike, which causes the rest of the signal to be reduced in amplitude as the signal is normalised.

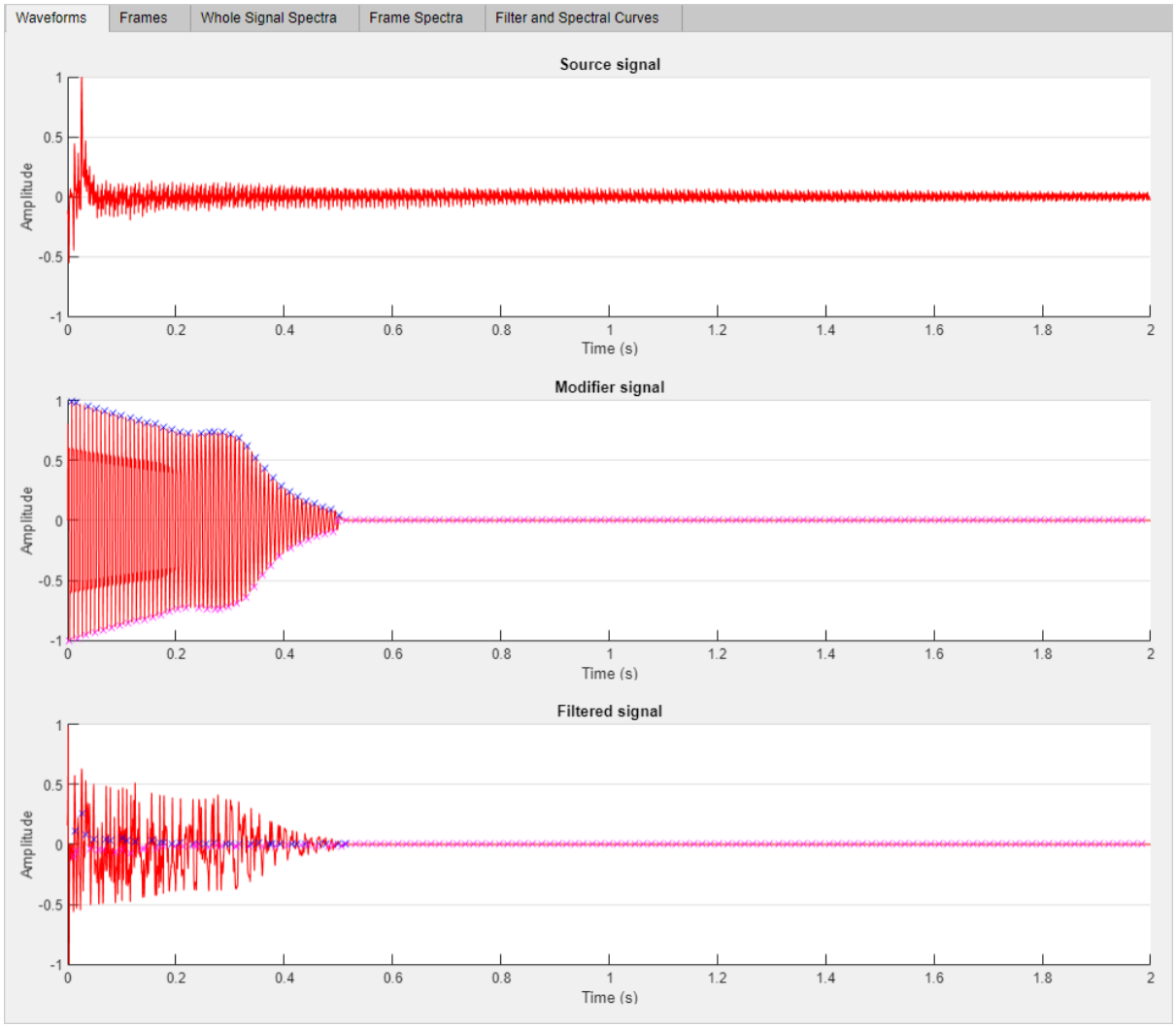


*Figure 6.50 – waveforms of a popped C3 bass note (source), the C3 synthesized pluck note (modifier) and the resulting processed signal.*

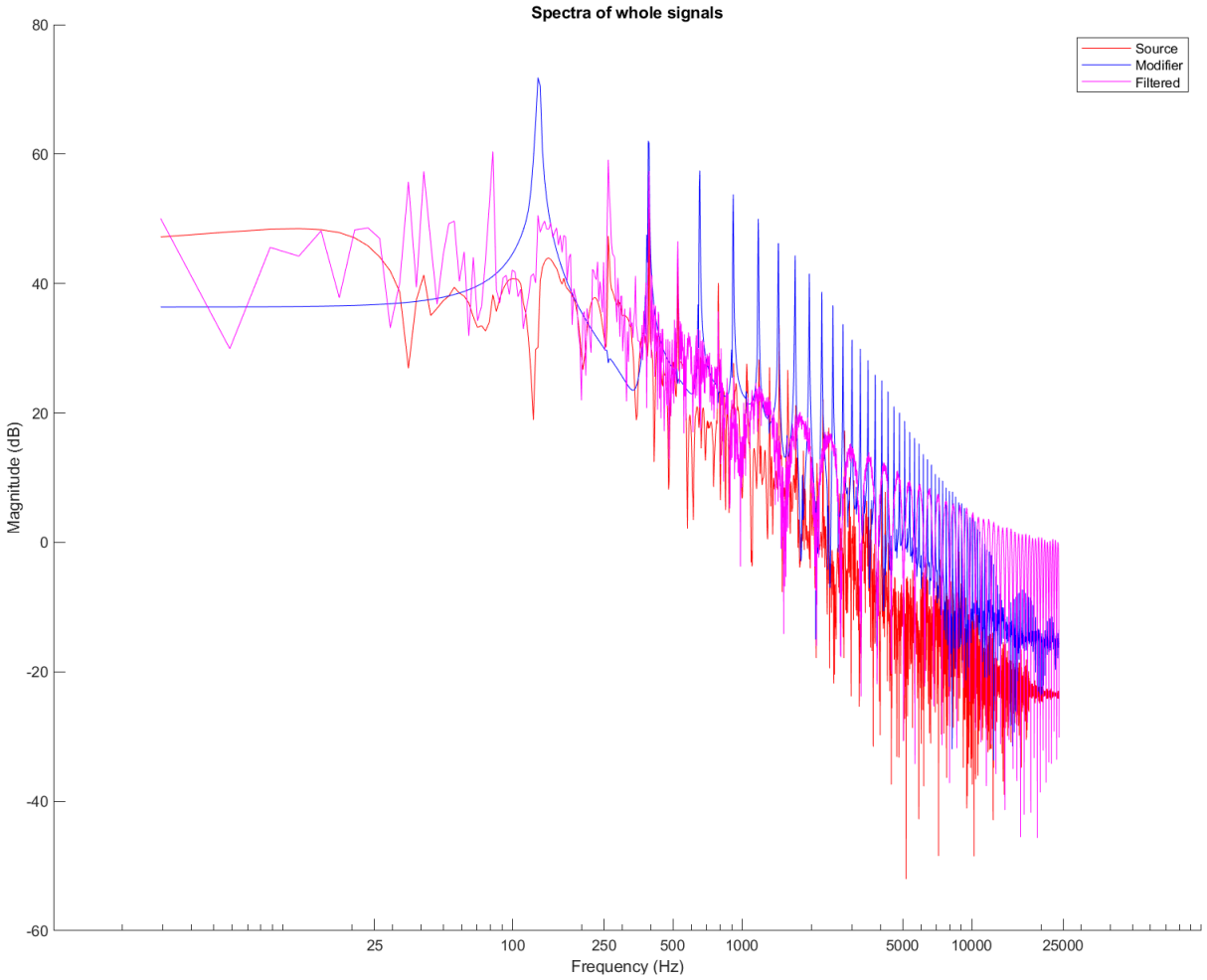


*Figure 6.51 – spectra of the signals detailed in 6.50. Similar deformation to that in 6.44 is apparent.*

Acoustically, the C3 examples with the synthesized pluck were an improvement over their lower-pitched counterparts but ultimately failed to produce a plausible instance of timbral blending. As in previous experiments where the filter had failed to perform adequately, any timbral blending occurring is mostly the product of the envelope shaping process capturing the shape of the note somewhat accurately. The actual effect of the filter was also unpredictable, sometimes attenuating high frequencies effectively to match the rounded thump of the synth note. In other situations, the lower frequencies would appear to be smothered by the filter producing a raspy, brittle timbre that would subside as the note decays. This could be an attempt to reproduce the falling filter cutoff of the synth note, however the filter lacks the precision and reaction time to generate a convincing acoustic result.

The last experiments performed with the system concerned longer musical signals, just as performed with the earlier piano samples. A fingerstyle G-major one octave scale was loaded into the system as the source signal, whilst an equivalent synthesized signal was loaded as the modifier. Signal processing was then applied. As illustrated in fig. 6.52, the envelope matching system performed well generally but stumbled when matching note transients. Clear spikes in the time-domain plot can be seen at the onset of several notes, caused by faulty multiplication of points in the signal to match the modifier signal envelope. Again, it is likely that these distortions were caused by misalignments in peak location detection prior to amplitude coefficient estimation. The spectra of the signals portrayed in 6.52 is given in fig. 6.53. Immediately, the results of the process look poor and the general acoustic result left much to be desired. Interestingly however, the timbre of each note in the processed signal appeared to change as the signal played through. The bass frequency band appeared to swell as the signal progressed and the high-end bite of the bass guitar was dulled. This suggests that the system was failing to perform fast enough to provide an accurate real-time depiction of timbral blending, which was not improved by forcing the system to operate more regularly using smaller frame sizes. This is consistent with other results obtained by limiting the frame size of signals in this section. Processing was repeated using samples transposed an octave higher and similar results were observed (figs. 6.54 and 6.55). Like the other higher pitched C3 samples covered in this section, the system appeared to perform marginally better on the higher G-major scales analytically. The timbre of notes appeared to morph as the signal progressed, just like the lower-pitched scales auditioned previously.

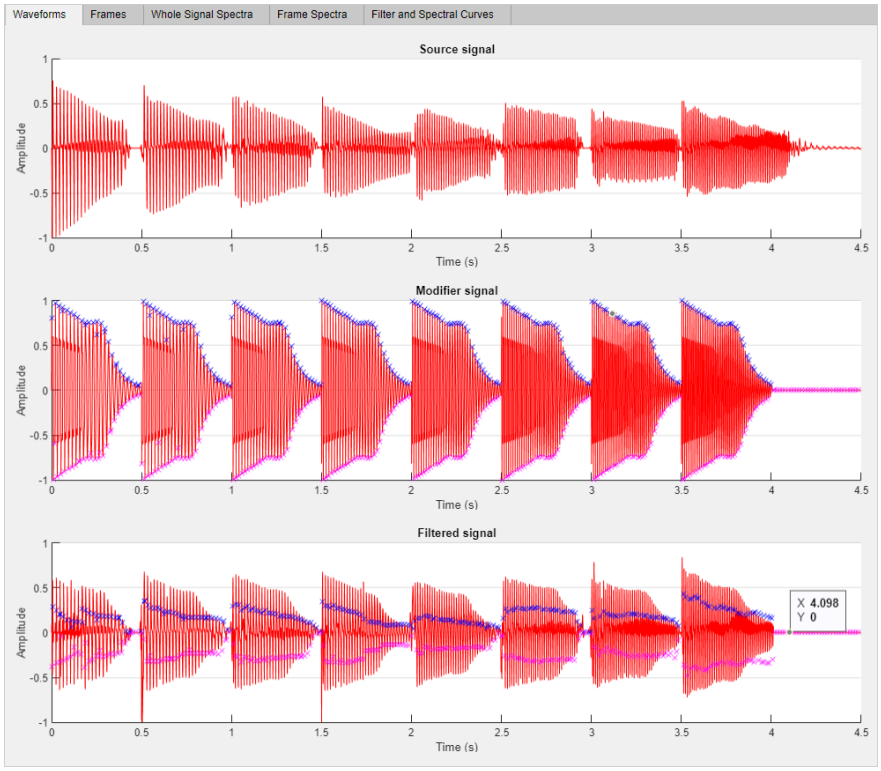


*Figure 6.52 – waveforms of G-major scale signals (root note G2). Source is the bass guitar signal; modifier is the synthesized pluck.*

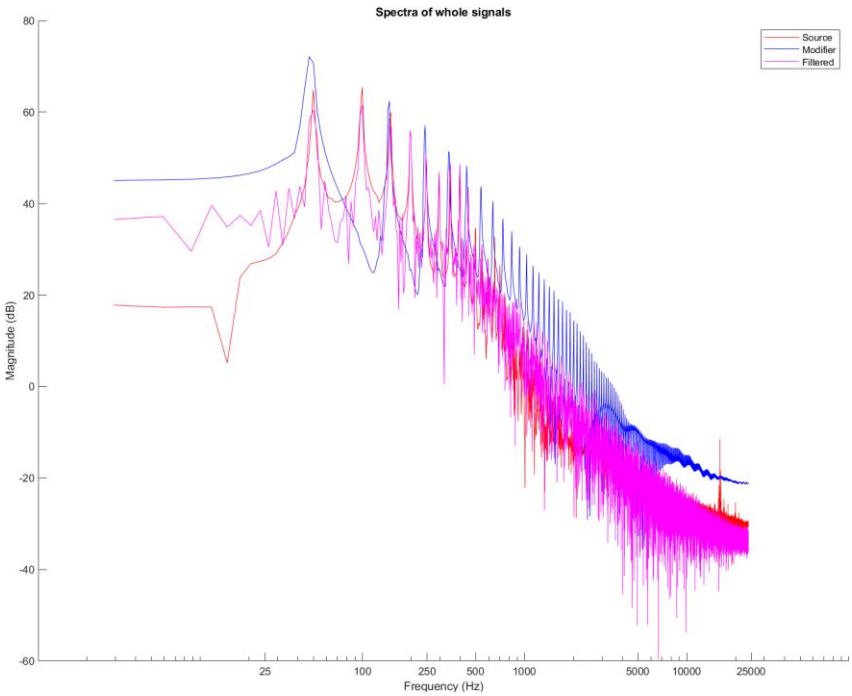


*Figure 6.53 – spectra of the signals detailed in 6.52. Some matching appears to be achieved around 1000Hz.*

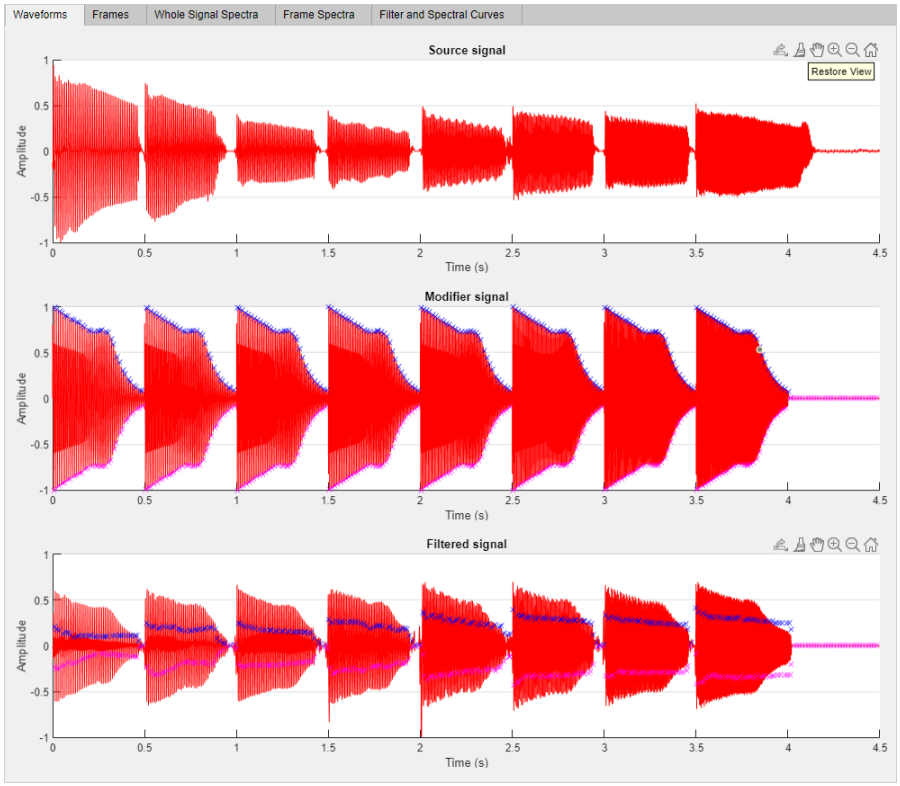


*Figure 6.54 – waveforms of signals used in a similar experiment to 6.52, with a root note of G3.*

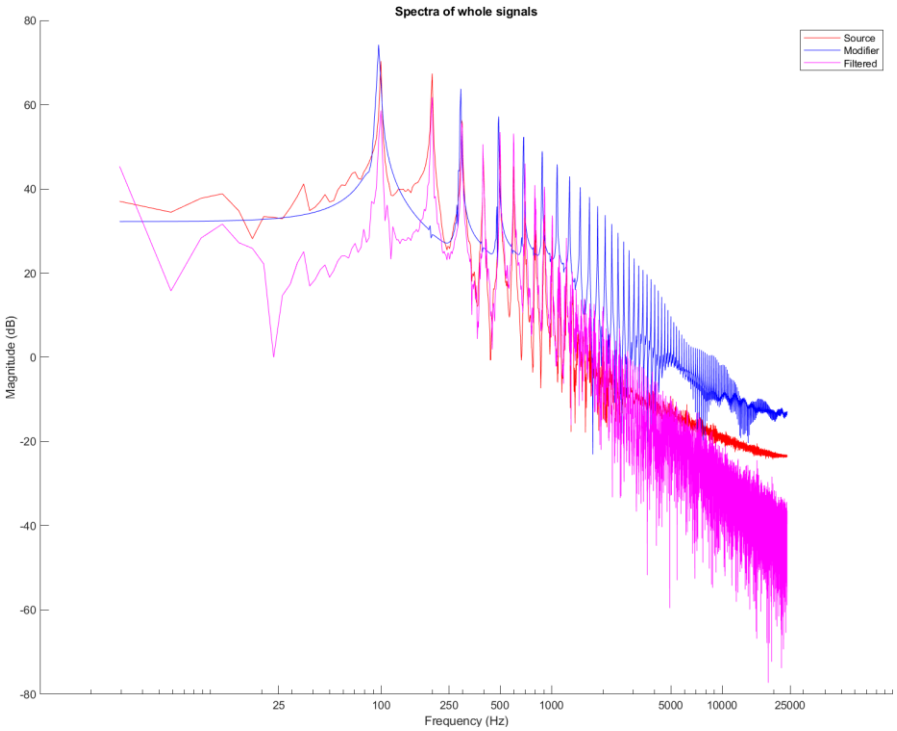


*Figure 6.55 – spectra of the signals detailed in 6.54. Spectral matching was poor overall.*

In summary, the matching filter and envelope systems appeared to struggle with processing the synthesized signals, more so than it did with the piano examples. These

results lend credibility to the speculation on system performance with simpler musical signals earlier in this chapter. Furthermore, it is suggested that the performance of the filter is largely dependent on how spectrally rich both the source and modifier signals are. White noise examples, being the most spectrally dense of any examples auditioned in this section, had their spectra matched well, whilst the square-wave synthesizer was frequently misrepresented in its spectral behaviour. It was also discovered that such a system of filtering using a dynamic filter curve was subject to numerous trade-offs, each in turn affecting the acoustic quality of the outcome signal. The filter could not be made to operate fast enough to capture minute timbral detail and replicate it on a host signal, nor had it the accuracy to model such fluctuations in timbre over time. This is exemplified by the tests using synthesized modifier signals; the defining resonant pluck of the synth notes was impossible to replicate accurately no matter the tweaking of user parameters. Further supporting evidence exists in the results of processing longer musical signals comprising of several notes. The evolving timbre as the signal progressed is a smeared representation of the rapidly changing apparent timbre caused by the triggering of new notes. The system had no means of identifying the onset of a new note, causing individual notes and musical passages to be treated the same. The system did perform well on single notes with little timbral evolution as they decay, such as the C3 piano note examples which were matched well in both the time and frequency-domains.

Signal envelope matching performed well overall on most signals but was subject to similar flaws as the signal. Most crucially, as illustrated by numerous examples in this section, note onsets are subject to distortion caused by erroneous multiplication of signal amplitudes, greatly hampering the acoustic plausibility of the signal. It also appears that the method of tracking peak locations in signals was inadequate and could lead to misshapen note envelopes in the case of one signal containing regularly occurring peaks, or if the signal subject to envelope shaping contains multiple notes. Envelope matching worked the best on signals consisting of single notes with irregularly spaced peaks, such as the earlier C2 and C3 piano examples. It also managed to match asymmetrical signal envelopes and did not

introduce any amplitude modulation in the process. This indicates that the concept behind the envelope shaping algorithm is functionally sound, but in need of refinements to handle more types of modifier signals.

# 7. Conclusions and further work

Having tested the matching filter and envelope algorithms on a wide variety of bass guitar samples with corresponding modifier signals, it can be concluded that the matching filter is inadequate for replicating timbre in this manner. This result can be largely attributed to a key fault in its working, that being a window length too small for effective analysis and, subsequently, estimation of filter coefficients quickly enough to produce the illusion of timbral blending. In contrast, the window length was also too large to create the illusion of replicating aspects of an ever-changing signal, instead producing step-like artefacts at extreme values that only somewhat reflect the modifier signal. Previously, the matching filter system implemented by Ma et al. (2013) was used as an example on the importance of sample data quality. Now, it is more evident as to why the filter presented here struggled in comparison. Firstly, the frame size of the signal to be filtered is relatively small – 64 samples long in the original design as opposed to the optimal value of 256 used to test the system in chapter 6.

Compared to the datasets retrieved used in the original implementation, the data obtained from the bass guitar was miniscule. Ma et al. (2013) produced an ideal curve determined by amalgamating spectra obtained from nearly 800 songs that appeared in the UK and US charts from 1950 onwards and smoothing the combined curve. This produced a smooth curve that covered the entire frequency spectrum with no sudden or isolated peaks, demonstrating the scaling effectiveness of the FFT operation when applied to longer datasets. In their work, the matching filter was also intended for use on other fully mixed musical signals comprising of multiple instruments resulting in a greater density of detail in the magnitude spectrum. The combination of short window lengths and limited spectral information in the source signals resulted in poor and inaccurate estimations of filter parameters, ultimately failing to capture meaningful timbral qualities of the modifier signal.

The performance of the matching envelope system was more encouraging, perhaps due to the relatively simple techniques employed in its development. It was discovered that

there is still significant room for improvement, largely concerning the misalignment of detected peaks resulting in erratic amplitude coefficient estimations. This was most evident when signals containing multiple notes were processed. The performance of the matching envelope system on individual notes was more promising, although the issue of erroneous estimations was rarely avoided entirely. The matching envelope system also appeared to struggle on the simpler synthesized waveforms, suggesting that a more intelligent peak-finding system is required. Effects on higher-pitched notes were inconsistent as some envelopes were detected and replicated well whilst other signals were destroyed in the process. This inconsistency suggests the algorithm is sensitive to minute changes in signal composition. It can be concluded that the matching envelope system works in an ideal environment and functions as a proof of concept but requires further development for practical use as a musical tool.

On a fundamental level however, it could be argued that filters themselves are not appropriate for combining the timbral characters of two signals as a filter only modifies an existing signal, as opposed to synthesizing a signal. Filters can be thought of as devices that alter a sound from the top down; they manipulate spectra after the signal has already been produced. This contradicts how sounds are produced in the real world, from vibrating mediums creating variations in pressure which propagate as waves through the surrounding material until they reach our ear drums (Goldstein 2010). Synthesizing a signal is a substitute for the vibrating medium in this model, whereas a filter would merely affect the wave shape of the signal after it has already been created.

## 7.i. Sinusoidal modelling synthesis

An ideal timbral blending system should produce a signal that sounds convincing to a listener, as if a physical instrument could have created the sound even if it had been synthesized artificially. Take for example the Stroviol, an acoustic instrument shaped constructed similarly to a violin, with four strings running perpendicular to a fretless

fingerboard (University of Edinburgh, n.d.). Instead of sound being amplified through the bridge and hollow body of the instrument, it is projected through a metal horn, much like a trumpet in its design. The sound produced by the Stroviol can be described as a curious mixture of a violin and a trumpet, as if a conventional violin had inherited the typical brassy overtones of a horn. The Stroviol makes for an excellent test subject for future research into timbral blending as a violin source signal influenced by a trumpet modifier signal should equate to a product sound comparable in its timbre to the Stroviol.

Another existing timbral blending device is the vocoder, a tool used to synthesize a signal based upon the spectral characteristics of two inputs. Prior to synthesis, the source signal is decomposed into at least two components – the steady-state sinusoids and the noise parts (Flanagan and Golden 1966). These components can be expected to exhibit differing spectral behaviours. Sinusoids define the tonal information of a signal and change frequency and amplitude slowly over time. Tonal information for speech is produced by controlled vibrations of the vocal cords in the larynx and is important to determine the tone or implication of a speaker. This information is removed in the synthesis step by some early vocoders which simply modulated the amplitude of a buzzer to replace the complex sinusoids produced by the larynx, making them comparatively difficult to understand (Dudley, 1940). Even as vocoder technology improved and modifier input signals became accepted for modulation over primitive buzzers, the recognisable human trait of voice modulation is often obliterated in favour of the timbral details of the modifier as defined by the behaviour of its slow-moving sinusoids.

Noise components are much the opposite and are chaotic in frequency and amplitude. Naturally, noise cannot be traced as a peak in the frequency domain, the product of a single sinusoid moving in frequency and amplitude over time. Rather, the noise component defines the spectral details of the signal (Taylor, 2009). In speech, noise as produced by our mouths can constitute the plosives of hard consonants and hisses of soft consonants alike. Further distinctions may be made between the impulsive plosive sounds of “T” and “P” from softer noisy sounds “s” and “c”. These impulsive dispersions of

sinusoidal energy can be identified as transients with distinct behaviour from other noise in a signal, perhaps produced by external factors from the instrument such as microphone hiss or feedback. Therefore, a musical signal can consist of three identifiable parts – its sinusoids, transients and noise.

By this reasoning, a sinusoidal modelling synthesis technique could be more suitable for the blending of signal timbres. Based on the performance of the matching filter in this application, filters do not offer an acceptable level of accuracy for such a purpose, so a sensible path to take this research down is to investigate the feasibility of constructing an acoustically plausible signal from the ground up. One such method of signal analysis and decomposition using the sines-transient-noise (STN) model has been proposed (Verma and Meng, 1998). A musical signal is taken and is first decomposed frame by frame for segmented processing like other techniques detailed in this paper. Sinusoidal modelling can be achieved using a matching pursuits approach (Mallat and Zhang 1993, Pati et al. 2002).

## 7.ii. Application of matching pursuit-based cross-synthesis for creative purposes

Decomposition of the source signal into its STN components yields three distinct signals that faithfully recreate the source signal when combined. Multiple signals, when decomposed in such a manner, could theoretically be convolved and reassembled to blend the timbres of existing samples. Alternatively, the decomposed components could be analysed and parameterised to alter their acoustic properties. Any manner of signal processing could be applied to the components before they are reassembled. Time-stretching, envelope shaping and simple gain control could be used to drastically alter the timbre of musical signals at a more granular level than the timbral blending system presented in chapters 5 and 6.

Perhaps the greatest benefit of decoding spectral information this way would be to gather information on a handful of parameters that can be expected to be in most musical

signals – note attack, decay, vibrato and so on. Any number of musical characteristics and the degree to which they occur can be estimated from a signal with accuracy increasing alongside the length of the input signal (effectively, the dataset for analysis). From there, the system could ‘learn’ how an instrument sounds based on its spectral activity and how performance variables, such as dynamics and attack, affect the changing spectrum of a note over its lifetime. Given an input signal, note onsets may be extracted by transient detection and the same musical characteristics may be determined of it just like the modifier signal. Ultimately, as the final signal is up for synthesis based upon the two inputs, the synthesizer will refer to the expected behaviour of the modifier signal and estimate how the note for resynthesis would take on this new behaviour. This is an example of the promises of sinusoidal modelling synthesis for creative timbral blending and manipulation as the precise spectral qualities of an instrument, as defined by its construction and performance, may be decoded and used to synthesize an acoustically plausible but physically absent instrument.

# Bibliography

**Abesser, J., Lukashevich, H., and Schuller, G. (2010).** *Feature-based extraction of plucking and expression styles of the electric bass guitar. 2010 IEEE International Conference On Acoustics, Speech And Signal Processing. doi: 10.1109/icassp.2010.5495945*

**ANSI (1960).** *Psychoacoustic Terminology: Timbre. New York, NY: American National Standards Institute.*

**Aravanditis, T. (2014).** *Spectral Modelling for Transformation and Separation of Audio Signals. MSc. University of York.*

**Bregman, A. (1990).** *Auditory scene analysis. Cambridge, Mass: MIT Press.*

**Brewer, R. (2003).** *The Appearance of the Electric Bass Guitar: A Rockabilly Perspective. Popular Music And Society, 26(3), 351-366. doi: 10.1080/0300776032000116996*

**Cooley, J. and Tukey, J. (1965).** *An Algorithm for the Machine Calculation of Complex Fourier Series. Mathematics of Computation, 19(90), p.297.*

**Darke, G. (2005).** *Assessment of Timbre Using Verbal Attributes. Proceedings Of The Conference On Interdisciplinary Musicology (CIM05). Retrieved from https://www.researchgate.net/profile/Graham_Darke/publication/228675696_Assessment_of_timbre_using_verbal_attributes/links/5af01394aca2727bc0065c61/Assessment-of-timbre-using-verbal-attributes.pdf*

**Flanagan, J. and Golden, R. (1966).** *Phase Vocoder. Bell System Technical Journal, 45(9), 1493-1509. doi: 10.1002/j.1538-7305.1966.tb01706.x*

**Fourier, J. B. J. (1878).** *The Analytical Theory of Heat. Nature, 18(451). Trans. Alexander Freeman.*

**Friedlander, B., and Porat, B. (1984).** *The Modified Yule-Walker Method of ARMA Spectral Estimation. IEEE Transactions on Aerospace and Electronic Systems, AES-20(2), 158-173.* doi: 10.1109/taes.1984.310437

**Fritz, C., Blackwell, A., Cross, I., Woodhouse, J. and Moore, B. (2012).** *Exploring Violin Sound Quality: Investigating English Timbre Descriptors and Correlating Resynthesized Acoustical Modifications with Perceptual Properties. The Journal of the Acoustical Society of America, 131(1), pp.783-794.*

**Gentleman, W., and Sande, G. (1966).** *Fast Fourier Transforms. Proceedings Of The November 7-10, 1966, Fall Joint Computer Conference On XX - AFIPS '66 (Fall). doi: 10.1145/1464291.1464352*

**Gilby, I. (1987).** *Stepp DG1 Digital Guitar (SOS Feb 1987). Retrieved 26 December 2020, from http://www.muzines.co.uk/articles/stepp-dg1-digital-guitar/1508*

**Glinsky, A. (2000).** *Theremin: Either music and espionage (p. 26). Urbana: University of Illinois Press.*

**Goldstein, E. (2010).** *Encyclopedia of perception (p. 147). Los Angeles: SAGE.*

**Harris, F. (1978).** *On the use of windows for harmonic analysis with the discrete Fourier transform. Proceedings of the IEEE, 66(1), 51-83. doi: 10.1109/proc.1978.10837*

**Heck, T. (2001).** *Torres Jurado, Antonio de. Oxford Music Online. [online] Available at: https://www-oxfordmusiconline-com.libproxy.york.ac.uk/grovemusic/view/10.1093/gmo/9781561592630.001.0001/omo-9781561592630-e-0000044517 [Accessed 1 Feb. 2020].*

**Helmholtz, H. and Ellis, A. (1885).** *On the sensations of tone as a physiological basis for the theory of music. 2nd ed.*

**Herbst, J. P. (2019).** *Empirical Explorations of Guitar Players' Attitudes Towards Their Equipment and the Role of Distortion in Rock Music. Current Musicology, (105). https://doi.org/10.7916/cm.v0i105.5404*

**Hewitt, E., & Hewitt, R. (1979).** *The Gibbs-Wilbraham phenomenon: An episode in fourier analysis. Archive For History Of Exact Sciences, 21(2), 129-160. doi: 10.1007/bf00330404*

**Jahnel, F. (2000).** *Manual of Guitar Technology*. *Westport, CT: Bold Strummer, pp.24-35.*

**Kanno, M. (2001).** *Timbre as Discourse: Contemporary Performance Practice on the Violin. D. Phil. University of York.*

**Karjalainen, M., Valimaki, V. and Tolonen, T. (1998).** *Plucked-String Models: From the Karplus-Strong Algorithm to Digital Waveguides and beyond. Computer Music Journal, 22(3), p.17.*

**Karplus, K. and Strong, A. (1983).** *Digital Synthesis of Plucked-String and Drum Timbres. Computer Music Journal, 7(2), p.43.*

**Kartomi, M. J. (1990).** *On Concepts and Classifications of Musical Instruments*. *Chicago. University of Chicago Press.*

**Kormylo, J. and Jain, V. (1974)***. Two-pass recursive digital filter with zero phase shift. IEEE Transactions On Acoustics, Speech, And Signal Processing, 22(5), 384-387. doi: 10.1109/tassp.1974.1162602*

**Kramer, P., Abesser, J., Dittmar, C. and Schuller, G. (2012).** *A digitalwaveguide model of the electric bass guitar including different playing techniques. 2012 IEEE International Conference On Acoustics, Speech And Signal Processing (ICASSP). doi: 10.1109/icassp.2012.6287889*

**Lawing, S. (2017).** *How Does a Pickup Really Work? — Lawing Musical Products. Retrieved 25 December 2020, from https://lawingmusicalproducts.com/dr-lawings-blog/how-does-a-pickup-really-work*

**Lighthill, M. J. (1958).** *An Introduction to Fourier Analysis and Generalised Functions. Cambridge: Cambridge University Press.*

**Ma, Z., Reiss, J. and Black, D. A. A. (2013).** *Implementation of an intelligent equalization tool using Yule-Walker for music mixing and mastering. 134th Audio Engineering Society Convention 2013. 173-182.*

**Martin, K.D. (1999).** *Sound-Source Recognition: A Theory and Computational Model. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA*

**Massenburg, G. (1972).** *Parametric Equalization. Journal Of The Audio Engineering Society. Retrieved from http://www.aes.org/e-lib/browse.cfm?elib=16171*

**Mehra, R. (1971).** *On-line identification of linear dynamic systems with applications to Kalman filtering. IEEE Transactions on Automatic Control, 16(1), pp.12-21.*

**Misa Digital (n.d.).** *Kitara. [online] Available at: https://misadigital.com/products/kitara [Accessed 1 Feb. 2020].*

**Nyquist, H. (1928).** *Certain Topics in Telegraph Transmission Theory. Transactions Of The American Institute Of Electrical Engineers, 47(2), 617-644. doi: 10.1109/t-aiee.1928.5055024*

**O'Connor, S. (2016).** *Patented electric guitar pickups and the creation of modern music genres. George Mason University, 23(4), 1007-1044.*

**Oppenheim, A., Buck, J. and Schafer, R. (1999).** *Discrete-time signal processing (2nd ed., pp. 8-93, 240-540, 541-774). Upper Saddle River, N.J.: Prentice Hall.*

**Oppenheim, T. (1981).** *Slap it! (pp. 1-2). Bryn Mawr, Pennsylvania: Theodore Presser.*

**Peeters, G., Giordano, B., Susini, P., Misdariis, N. and McAdams, S. (2011).** *The Timbre Toolbox: Extracting audio descriptors from musical signals. The Journal of the Acoustical Society of America, 130(5).*

**Perez-Gonzalez, E. and Reiss, J. (2009).** *Automatic Equalization of Multichannel Audio Using Cross-Adaptive Methods. [online] Aes.org. Available at: http://www.aes.org/e-lib/browse.cfm?elib=15026 [Accessed 26 Jan. 2020].*

**Plack, C., Oxenham, A., Fay, R. and Popper, A. (2005).** *Pitch: Neural Coding and Perception (p. 99). New York: Springer Science & Business Media Inc.*

which constitute the scientific basis of the sound being observed

**Pinkus, A. and Zafrany, S. (1999).** *Fourier Series and Integral Transforms. Cambridge: Cambridge University Press.*

**Proakis, J. and Manolakis, D. (1996).** *Digital signal processing (3rd ed., pp. 855-857). London [etc.]: Prentice Hall.*

**Prony, R. (1795).** *Essai Experimental et Analytique. Journal de l'école Polytechnique de Paris, 1, 24-76.*

**Rader, C. and Gold, B. (1967).** *Digital filter design techniques in the frequency domain. Proceedings Of The IEEE, 55(2), 149-171. doi: 10.1109/proc.1967.5434*

**Raichel, D. (2011).** *The science and applications of acoustics (pp. 13-30). New York: Springer.*

**Ramirez, R. (1985).** *The FFT, Fundamentals and Concepts. Englewood Cliffs, N.J.: Prentice-Hall.*

**Roberts, J. (2019).** *Partners: Anthony Jackson & Fodera Guitars. Retrieved 25 December 2020, from https://bassmagazine.com/artists/partners-anthony-jackson-fodera-guitars*

**Sachs, C. (1940).** *The History of Musical Instruments. New York: W.W. Norton, pp.455.*

**Salvatore, L. and Trotta, A. (1988).** *Flat-top Windows for PWM Waveform Processing via DFT. IEE Proceedings B Electric Power Applications, 135(6), p.346.*

**Seashore, C. (1938).** *The Psychology of Music. Music Educators Journal, 25(3), pp.23-23.*

**Slawson, W. and Erickson, R. (1978).** *Sound Structure in Music. Journal of Music Theory, 22(1), p.105.*

**Shannon, C. (1949).** *Communication in the Presence of Noise. Proceedings Of The IRE, 37(1), 10-21. doi: 10.1109/jrproc.1949.232969*

**Smalley, D. (1994).** *Defining timbre - Refining timbre. Contemporary Music Review, 10(2), pp.35-48.*

**Smith III, J. (2010a).** *Spectral Analysis Windows: Spectral Audio Signal Processing. Retrieved 23 December 2020, from https://www.dsprelated.com/freebooks/sasp/Spectrum_Analysis_Windows.html*

**Smith III, J. (2010b).** *Rectangular Window Side Lobes: Spectral Audio Signal Processing. Retrieved 23 December 2020, from https://www.dsprelated.com/freebooks/sasp/Rectangular_Window_Side_Lobes.html*

**Smith III, J. (2007a).** *Linear Time-Invariant Filters | Introduction to Digital Filters with Audio Applications. Retrieved 30 December 2020, from https://ccrma.stanford.edu/~jos/fp/Linear_Time_Invariant_Digital_Filters.html*

**Smith III, J. (2007b).** *The Ideal Lowpass Filter | Introduction to Digital Filters with Audio Applications. Retrieved 30 December 2020, from https://ccrma.stanford.edu/~jos/sasp/Ideal_Lowpass_Filter.html*

**Stansfield, A. (2013).** *SynthAxe. [online] Alendi.co.uk. Available at: http://www.alendi.co.uk/synthaxe.html [Accessed 1 Feb. 2020].*

**StarrLabs (n.d.).** *StarrLabs Ztar Z7S. [online] Available at: https://www.starrlabs.com/product/z7s/ [Accessed 1 Feb. 2020].*

**University of Edinburgh (n.d.).** *Retrieved 1 January 2021, from https://collections.ed.ac.uk/stcecilias/record/96119*

**Sullivan, C. (1990).** *Extending the Karplus-Strong Algorithm to Synthesize Electric Guitar Timbres with Distortion and Feedback. Computer Music Journal, 14(3), p.26.*

**Tabassum, F., Amin, M. and Islam, M. (2016).** *Comparison of FIR and IIR Filter Bank in Reconstruction of Speech Signal. International Journal of Computer Science and Information Security,. 14. 864-872.*

**Taylor, P. (2009).** *Text-to-Speech Synthesis (pp. 426-429). Cambridge: Cambridge University Press.*

**Vega, B. (2020).** *Bobby Vega: "I wasn't that consistent with my fingers, so I got dexterity and stamina from the pick." Retrieved 29 December 2020, from https://www.guitarworld.com/features/bobby-vega-i-wasnt-that-consistent-with-my-fingers-so-i-got-dexterity-and-stamina-from-the-pick*

**Verma, T. and Meng, T. (1998).** *Transient modelling synthesis: an analysis/synthesis tool for transient signals. Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98 (Cat. No.98CH36181).*

**von Hornbostel, E. and Sachs, C. (1914).** *Systematik der Musikinstrumente. Ein Versuch. Zeitschrift für Ethnologie, pp.20-23.*

**Wake, S. and Asahi, T. (1998).** *Sound Retrieval with Intuitive Verbal Expressions.*

**Weisser, S. and Quanten, M. (2011).** *Rethinking Musical Instrument Classification: Towards a Modular Approach to the Hornbostel-Sachs System. Yearbook for Traditional Music, 43, pp.122-123.*

**Wold, H. (1938).** *A study in the analysis of stationary time series (pp. 93-133). Stockholm: Almqvist & Wiksell.*

**Yasuda, K. and Hama, H. (2007).** *Formant structure for timbre of stringed instruments. International Journal of Innovative Computing, Information and Control. 3. 1369-1378.*

**Zacharakis, A., Pastiadis, K., Reiss, J. and Papadelis, G. (2012).** *Analysis of Musical Timbre Semantics Through Metric and Non-Metric Data Reduction Techniques. In: International Conference on Music Perception.*

**Zhivomirov, H. (2020).** *Spectral Envelope Extraction with Matlab Implementation (https://www.mathworks.com/matlabcentral/fileexchange/66199-spectral-envelope-extraction-with-matlab-implementation), MATLAB Central File Exchange. [Accessed 25 Nov. 2020].*

**Zölzer, U. and Dutilleux, P. (2002).** *DAFX: Digital Audio Effects (1st ed., pp. 48-49). Chichester: John Wiley & Sons, Ltd.*