UNIVERSITY OF CALIFORNIA, SAN DIEGO

Time Scale Modification Using a Sinusoidal Model

A Thesis submitted in partial satisfaction of the
requirements for the degree
Master of Arts

in

Music

by

Michelle Lee Daniels

Committee in charge:

Professor Shlomo Dubnov, Chair
Professor F. Richard Moore
Professor Miller Puckette

2009

The Thesis of Michelle Lee Daniels is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

Chair

University of California, San Diego

2009

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to acknowledge Dr. Kevin Short and Ricardo Garcia for introducing me to the sinusoidal analysis which is the foundation of much of the work in this thesis and for encouraging me to think outside the box. Many thanks to Ricardo especially for taking the time to give me very helpful feedback during the writing process.

I would also like to acknowledge the guidance of all of the members of my committee throughout the challenges of the past year.

Most importantly, I would like to acknowledge my parents, Karen and Murray Daniels, for all of their support and advice and for teaching me to love both computers and music in the first place.

ABSTRACT OF THE THESIS

Time Scale Modification Using a Sinusoidal Model

by

Michelle Lee Daniels

Master of Arts in Music

University of California, San Diego, 2009

Professor Shlomo Dubnov, Chair

Time-scale modification of audio signals is a classic digital signal processing problem. While many solutions have been implemented over the years, none work perfectly without audible artifacts for all kinds of input. In most applications, trade-offs must be made between computational efficiency and quality, resulting in different solutions being favored in different contexts and for different classes of input signals. Sinusoidal modeling is an approach which has shown significant promise for producing high-quality time-scaled audio. Leaving noise and transient signal components for future work in order to focus exclusively on sinusoidal components, this thesis describes a variation on traditional sinusoidal modeling which offers a unique combination of features including separating the process of identifying individual sinusoidal peaks from the process of combining those peaks into sinusoidal tracks, special treatment of phase during the peak tracking process, and analysis enhancements for sinusoidal tracks with modulating frequency and multichannel input signals.

Chapter 1

Introduction

1.1 Background

Time scaling is a popular audio effect with many applications including sound effect generation, electronic music, and in broadcast for fitting pre-recorded audio into time slots of slightly different duration. When an audio signal is time-scaled, its duration and rate of playback are altered based on a given modification factor, but both the spectral content of the original signal and any rhythmic content or temporal relationships should be retained. For example, simply playing back the original signal at a lower or higher sampling rate is not a correct approach to time scaling because, while it achieves the desired temporal modifications, this results in undesirable modifications to the spectral content of the signal. A perfect time scaling system would therefore satisfy all of the following requirements:

- Accurately modify the overall duration and playback rate of the input signal according to the given modification factor
- Preserve the spectral content of steady or sustained portions of the input signal, thereby preserving the perceived pitch
- Preserve the perceptual quality of attacks and transients in the input signal

Analog vs. digital

Time-scale modification methods which attempt to meet all of the criteria listed above are almost exclusively digital techniques. In the analog domain, primitive time scaling can be performed using variable speed tape playback, but this has the inevitable side effect of also modifying the spectral content of the signal, similar to the effect of changing sample rates in the digital domain. At one point in time, some highly complex machines [7] were developed to work around these artifacts, but the widespread availability of more practical digital techniques has made analog approaches essentially obsolete except in cases where the artifacts are actually desired for creative purposes. There are many digital approaches to time scaling, and these can generally be divided into two categories: time domain techniques and frequency domain techniques, although some more recent approaches are a hybrid of both [4]. The following sections describe some of the most common techniques in both domains.

1.1.1 Time-scale modification in the time domain

Synchronous overlap-and-add

Methods for time-scale modification in the time domain are typically based on overlapping and adding short segments of audio using various techniques for optimizing amplitude and phase continuity in overlap regions. One of the most basic of these methods is Synchronous Overlap-and-Add (SOLA), originally proposed for use with speech signals by Roucos and Wilgus in 1985 [13]. This efficient technique with low computational complexity uses the cross-correlation between consecutive time segments to determine the point of maximum similarity where overlap should occur to minimize artifacts [7].

Pitch-synchronous overlap-and-add

A variety of improvements to the basic SOLA algorithm have been made over the years, and one of the most popular of these is Pitch-Synchronous Overlap-Add (PSOLA), presented by Charpentier and Stella in 1986 [2] for use in speech

applications. PSOLA works well on pitched audio signals and uses an analysis pre-processing stage to detect the best places to segment the input audio signal based on the signal's periodic structure. Because of its dependence on periodicity however, PSOLA is not very effective for handling transients, non-pitched signals, and polyphonic sounds. However, some of these difficulties, such as the poor handling of transients and attacks, can be reduced by detecting problem areas in advance and, for example, not scaling note attacks while compensating by scaling sustained portions of the signal slightly more or less. This kind of custom handling of attacks and transients can also apply to other kinds of both time domain and frequency domain based time-scale modification methods.

1.1.2 Time-scale modification in the frequency domain

Phase vocoder

The phase vocoder, popular for performing a wide range of spectral audio modifications, is one tool used for time scaling in the frequency domain. Like other applications of the phase vocoder, time scale modification works best on steady monophonic signals and signals where partials are not very closely spaced in frequency and generally performs poorly when presented with transients, sharp attacks, and sinusoidal content which modulates over time across frequency bin boundaries. The phase vocoder consists of separate analysis and synthesis stages. In modern phase vocoder implementations, the analysis stage transforms audio signals into the frequency domain using the Short-Time Fourier Transform (STFT) with a particular analysis hop size, where the hop size defines the number of samples that are "hopped", or advanced, between the start of one analysis block (or "frame") and the next. Then, to perform time-scaled resynthesis, a different hop size is used for synthesis than for analysis, and the analysis phases of the STFT are modified to produce synthesis phases preserving phase continuity within a particular frequency bin cross consecutive frames [10]. An inverse STFT is then used to synthesize the modified time domain signal from the modified spectrum.

Sinusoidal models

Sinusoidal models are another approach to frequency domain time scaling. In such models, audio is decomposed into its component sinusoids, which, unlike in the phase vocoder, need not correspond to Discrete Fourier Transform (DFT) analysis bins. In many implementations of sinusoidal models, the analysis process not only detects sinusoidal components but also produces a residual signal which contains the non-sinusoidal elements of the input signal such as noise and transients. One of the strengths of sinusoidal modeling compared to other techniques such as the phase vocoder is that when applying effects and resynthesizing audio, these steady state (sinusoidal) and non-steady state (noise/transients) components can be processed independently. The resulting flexibility can greatly improve the quality of processing when the input signal is not purely sinusoidal, and can be quite useful in applications such as creating new audio effects or audio coding. While the sinusoidal modeling analysis process can be very computationally expensive, synthesis is generally highly optimized, and an input signal which has been analyzed once can then be synthesized in real-time with many transformations without repeating the analysis process.

The classic examples of sinusoidal models are a model for speech developed by McAuley and Quatieri [12] and Spectral Modeling Synthesis (SMS) for more general audio signals, pioneered by Serra and Smith [18]. Both approaches detect time-varying tracks of spectral peaks assumed to represent underlying sinusoidal components and describe these tracks at regular analysis time intervals using amplitude/frequency pairs along with, in some cases, phase. Once these tracks are identified, time scale modifications are relatively straightforward. Along a given trajectory, linear interpolation provides instantaneous amplitude, while frequency and (in some cases) phase are interpolated to compute the sample-by-sample phase advance according to the desired time scale. For each track, the use of interpolation ensures both amplitude and phase continuity in the reconstructed signal from frame to frame.

Most more recent incarnations of sinusoidal models not only model sinusoidal components but also incorporate models for the non-tonal (noisy) and tran-

sient parts of signals, which are not typically well-represented by using sinusoids alone. Serra describes a "stochastic plus deterministic" (sines + noise) model in his PhD thesis [14], while others, such as Verma, Meng, and Levine [19] have extended that further to include modeling of transients (sines + noise + transients). With these decompositions, sinusoidal components, noise, and transients can all be processed independently in ways which take advantage of their unique qualities. This can be particularly valuable for applications like time-scale modification, where such a system makes it much easier to preserve the perceptual quality of attacks and transients. Other variations of sinusoidal models include Dubnov's linear-prediction-based YASAS [5], a method by Depalle and Rodet [3] which uses a Hidden Markov Model, and Fitz and Haken's Lemur [9] which uses bandwidth-enhanced partials rather than pure sinusoids.

1.2 Goal of this thesis

This thesis will discuss an approach to time-scale modification using sinusoidal modeling assuming a "sines + noise" or "sines + noise + transients" decomposition. Although it has much in common with traditional sinusoidal modeling-based methods, this approach represents a unique combination of analysis and synthesis techniques which are each meant to improve various short-comings of time-scale modification using sinusoidal models.

To start with, rather than using more traditional methods of identifying sinusoidal components of a signal such as peak picking, this approach uses the Complex Spectral Phase Evolution method [16] to provide highly accurate detection of sinusoids and clean separation of sinusoidal and non-sinusoidal signal components. This process will be described in more detail in section 2.1. The residual which results from this analysis contains only noise and transient components. Because the residual can be easily processed completely independently of the sinusoidal components of the signal, this thesis does not discuss the various ways in which the noise and transients can be modeled and processed, but rather focuses on high-quality handling of the sinusoidal components themselves.

It is perhaps important to note that the CSPE analysis process provides only lists of peaks contained in each analysis frame. It does not provide any method for creating sinusoidal tracks from these individual peaks, and for many applications tracks are not a prerequisite for high-quality analysis and synthesis. However, for successful time-scale modification, it will be shown in chapter 3 that the use of sinusoidal tracks is very necessary. After describing the basic sinusoidal analysis/decomposition process, this thesis will discuss an algorithm for accurately identifying sinusoidal peak tracks, and it will describe how to use those tracks to perform effective time-scale modification.

Finally, some extensions to the basic analysis/synthesis methods described will be presented, including a transformation which improves the quality of results when time-scaling stereo audio signals and an iterative approach to signal analysis which can improve the quality of results for signals containing sinusoids whose frequency modulates across frame boundaries.

Chapter 2

Sinusoidal Model

All sinusoidal models decompose audio into sinusoidal components, but many signals contain noisy and/or transient components which cannot be accurately represented by sinusoids alone. As a result, the analysis stage of many sinusoidal modeling approaches includes both sinusoidal decomposition and computation of a residual signal containing non-sinusoidal elements of the signal which can then be modeled and manipulated independently. The model presented here is one such model, but the focus of this thesis will be on the handling of sinusoidal components and not ways in which the residual signal can be modeled or modified.

2.1 Analysis

The goal of the analysis process is to detect the sinusoidal components present in the input audio signal on a frame-by-frame basis, producing a data structure consisting of a list of frames, with each frame containing a list of its component sinusoidal peaks. The work presented in this thesis is implemented using the Complex Spectral Phase Evolution (CSPE) [16] method to identify the sinusoidal components of a signal and estimate each component's frequency, amplitude, and phase. This method has been very effective, especially because of its success in detecting sinusoids even in the presence of noise, but any approach which also accurately identifies component sinusoids and their parameters and separates them from the noisy components of the signal could also be used in its place.

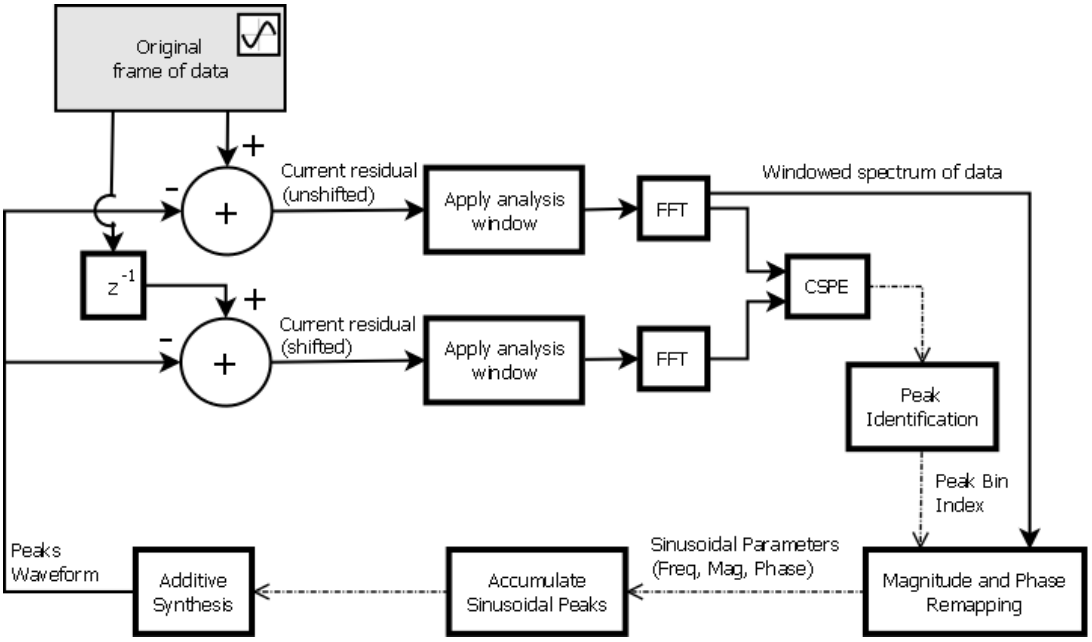

Figure 2.1: CSPE-based analysis process for a frame of data

Using the CSPE method to detect sinusoids is fairly straightforward. The first step in the sinusoidal decomposition is to divide the input signal into equal-sized overlapping blocks of samples. These blocks, or frames, will be analyzed individually to determine their sinusoidal content. If one sinusoid is identified and removed from the original signal at a time, the analysis process for each frame as seen in figure 2.1 would be as follows[1]:

1. Window the time domain data ($\vec{s_0}$) using analysis window $\vec{A}$ to produce the windowed signal $\vec{s}_{w0}$ and compute its DFT ($F_k(\vec{s}_{w0})$)

2. Window a time-shifted[2] version of the time domain data ($\vec{s_1}$) using analysis window $\vec{A}$ to produce the windowed signal $\vec{s}_{w1}$ and compute its DFT ($F_k(\vec{s}_{w1})$)

3. Compute the CSPE according to equation 2.1 (originally from [16]) where N is the analysis frame size, F denotes the DFT and F^* its complex conjugate,

[1]Various optimizations can be made, including identifying multiple peaks per CSPE computation and performing sinusoidal resynthesis in the frequency domain.

[2]A shift of one sample is assumed here for simplicity, but larger shifts can be used.

to obtain an array of floating point frequency values (f_{CSPE}) indicating which frequency contributed significantly to the energy in a given DFT bin k

$$f_{CSPE(k)} = \frac{N\angle(F_k(\vec{s}_{w0})F_k^*(\vec{s}_{w1}))}{2\pi} \tag{2.1}$$

4. Identify the DFT bin containing the spectral peak with the most energy

5. If the frequency contributing to this energy is close to the center frequency of the DFT bin in question, then a sinusoidal component was likely present here in the original signal and we can identify this as a peak.

6. Estimate the magnitude and phase of the chosen peak using the spectrum computed in step 1 and the spectrum of the analysis window ($F(\vec{A})$). This estimation is done using equation 2.2[3] (also from [16]), where a is the magnitude and $\phi = \angle(e^{jb})$ is the phase.

$$ae^{jb} = \frac{2F_k(\vec{s}_{w0})}{F_{(k-f_{CSPE(k)})}(\vec{A})} \tag{2.2}$$

7. Resynthesize all sinusoids found in this frame so far

8. Subtract the resynthesized peaks from the input to compute a new residual containing one less sinusoidal component

9. Repeat steps 1 through 8 using the current residual as input until the desired number of peaks have been extracted or no more sinusoidal components have been detected

10. Store lists of the peaks detected in each frame along with their frequency, magnitude, and phase parameters.

A residual signal containing noise and transient components can then be computed by subtracting all synthesized sinusoids from the original signal. As shown in figure 2.2, signal processing, such as time-scale modification, can then be performed on the residual signal and detected sinusoids. The results of processing these two components are then combined to produce a final processed waveform.

[3]Note that for the most accurate results when dealing with low frequency peaks where there may be interference from negative frequencies, the technique described in [15] for estimating magnitude and phase should be used instead.

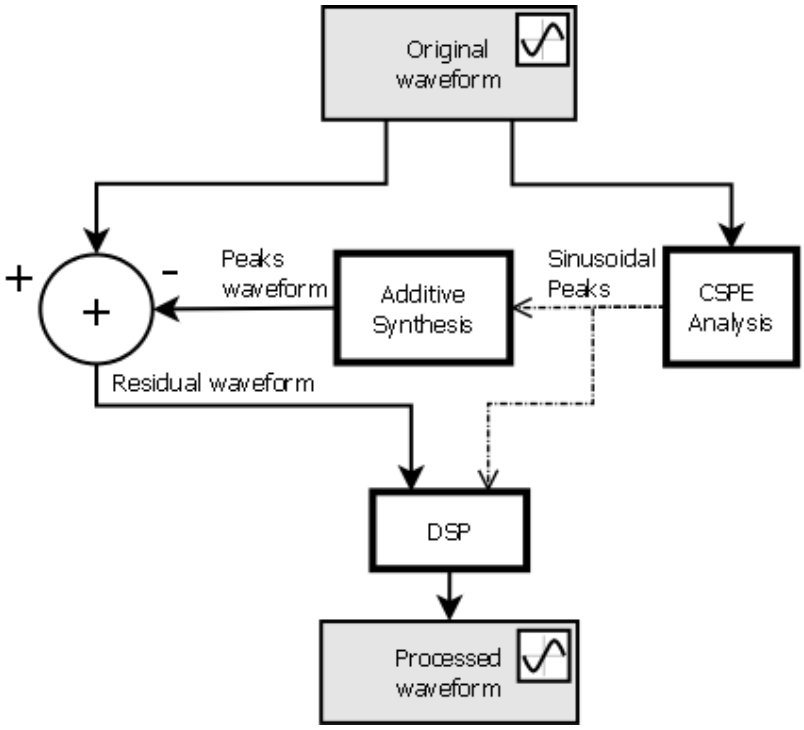

Figure 2.2: Analysis/synthesis process separating sinusoidal components from the original signal and producing a residual signal containing noise and transients

2.2 Synthesis

Reconstruction of the original input signal is achieved by synthesizing all of the sinusoidal peaks detected in each frame, concatenating frames using an overlap-add algorithm, and adding this combined sinusoidal signal to the (possibly modeled and reconstructed) residual signal containing noise and transients.

2.2.1 Sinusoidal peak synthesis

Using the same frame size and overlap as the analysis stage, all sinusoids in a particular frame are synthesized individually and added together, while any discontinuities between frames are minimized as a result of using a synthesis window to overlap and add frames. The additive synthesis of sinusoidal peaks can easily be performed in either the time domain or the frequency domain. In some cases, frequency domain synthesis may be more computationally efficient, but for simplicity this thesis will focus on time domain synthesis. To synthesize a peak in the time domain, the analysis amplitude[4] A, floating point frequency f, and initial phase ϕ are used as the parameters of the sinusoid according to equation 2.3, where t is the sample index and N is the analysis frame size.

$$\vec{s}(t) = A\cos(\frac{2\pi tf}{N} + \phi) \tag{2.3}$$

When reconstructed in the time domain, these sinusoids are not windowed (other than with the implicit rectangular window). Therefore, in order for the overlap-add process to work correctly, a synthesis window must be applied to each synthesized frame. Assuming that the same synthesis window is used for adjacent frames (a valid assumption unless a variable frame rate is used), the window must have the property that, overlapped with itself, it sums to 1.0 [1]. Equation 2.4 where w_s is the synthesis window, N is the frame size, and $n = 0, ...N-1$ is the sample index, summarizes this requirement for the case where there is 50% frame overlap.

[4]In order to use the frequency domain analysis magnitude directly in time domain reconstruction as an amplitude value, some re-scaling may be necessary during the analysis stage to compensate for implementation-dependent FFT scaling and the particular analysis window used. It is assumed throughout this thesis that any such necessary scaling has been performed.

Common windows that satisfy this property include triangular and hann windows.

$$w_s[n] + w_s[\frac{N}{2} + n] = 1.0 \tag{2.4}$$

2.2.2 Lossless reconstruction

By adding the noise and transients residual to the completed reconstructed sinusoids, the original input signal can be resynthesized with no error, resulting in a completely lossless reconstruction. However, it is common to also model the residual with some kind of approximation, in which case the original signal cannot be perfectly recovered. In a typical situation, the benefits of modeling both the sinusoidal components and the residual for purposes of transformations, audio coding, or other applications outweigh any artifacts introduced by the use of a not-quite-lossless residual.

Chapter 3

Time Scaling

With the results of analysis using a sinusoidal model such as that described in chapter 2, a variety of audio modifications can be performed, but this thesis focuses on one of the most common processes: time scale modification[1]. This chapter will introduce the challenge of handling the sinusoidal phase when performing time-scale modification, motivating the need for sinusoidal tracks. It will then describe the identification and synthesis of these sinusoidal tracks, present a way to improve the analysis process when dealing with tracks of modulating frequency, and describe how time-scale modification is performed during synthesis once sinusoidal tracks have been computed.

3.1 A phase challenge

Section 2.2 described how lossless or near-lossless reconstruction can be obtained using frame-by-frame synthesis of individual sinusoidal peaks and combining these reconstructed sinusoids with the residual signal containing noise and transients. When performing time-scale modification, however, this is approach is problematic. Because there is no concept of continuity across frame boundaries, each peak is in essence treated as though it is the beginning of a new event. When

[1]Time scale modification can be easily extended to pitch scale modification using sample rate conversion. This process is well-documented elsewhere, including [7] so it will not be described here.

time-scaling a sinusoidal peak, it is easy to make its duration shorter or longer than the original duration (which was one analysis frame), but one must give the peak an initial phase, and the only known phase is the original analysis phase.

This section describes a situation where using that original analysis phase while time scaling can be problematic.

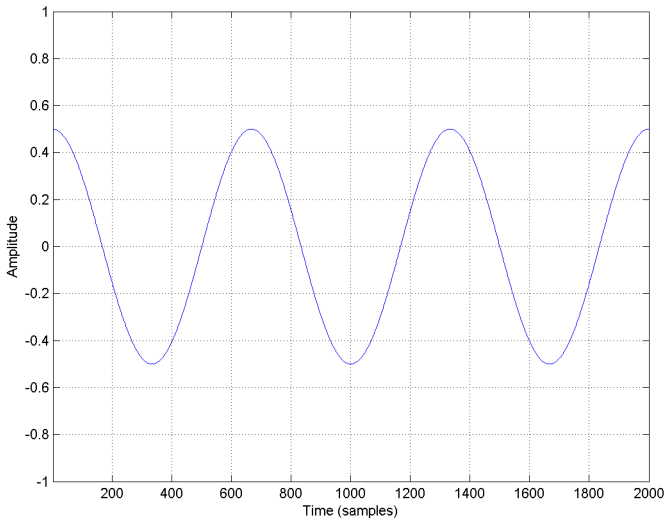

Figure 3.1: Original sinusoid

Figure 3.1 shows a two periods of a sinusoid with a frequency of 1.5 cycles per analysis frame. Assuming no overlap (for simplicity), the frame-by-frame sinusoidal decomposition of this signal is shown in figure 3.2. The analysis process from section 2.1 tells us that there is one sinusoidal peak in each frame, but the two peaks are completely independent; there is no concept of peak continuity from frame-to-frame.

Now, if we want to stretch the input signal in figure 3.1 to twice its original duration, it is necessary to stretch each of its component frames of audio to twice their original durations. Stretching each of the frames shown in figure 3.2 independently by simply extending their lengths results in the waveforms shown in figure 3.3.

This seems straightforward enough, but when these two waveforms are concatenated to generate the final time-scaled waveform, the result is the waveform

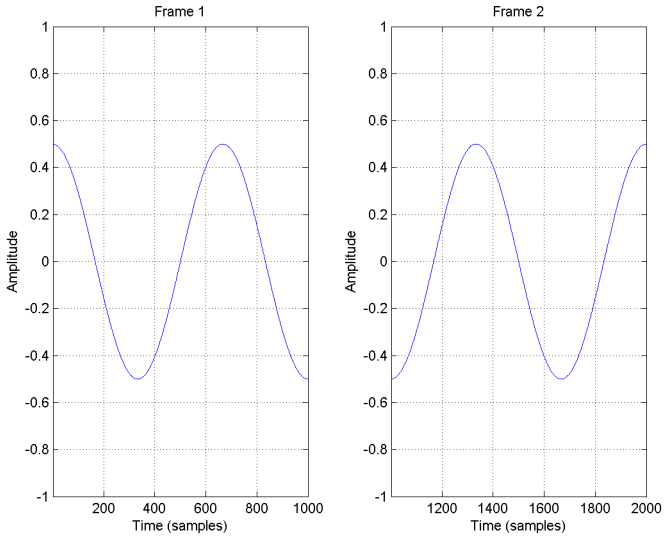

Figure 3.2: Frame-by-frame view of original sinusoid

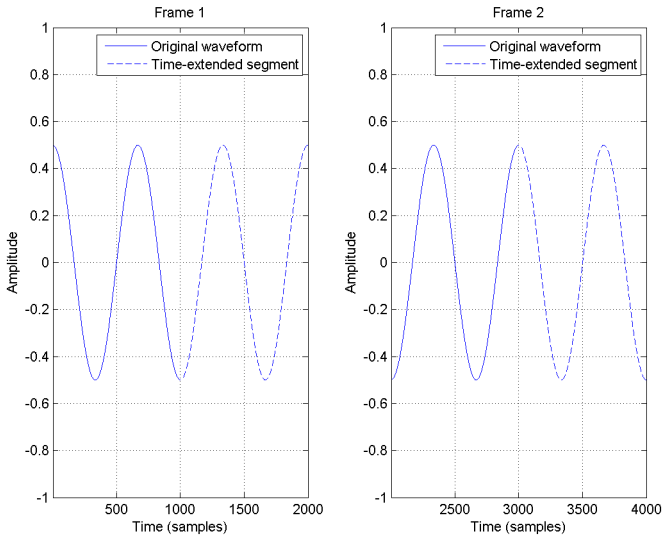

Figure 3.3: Frame-by-frame view of time-scaled sinusoid

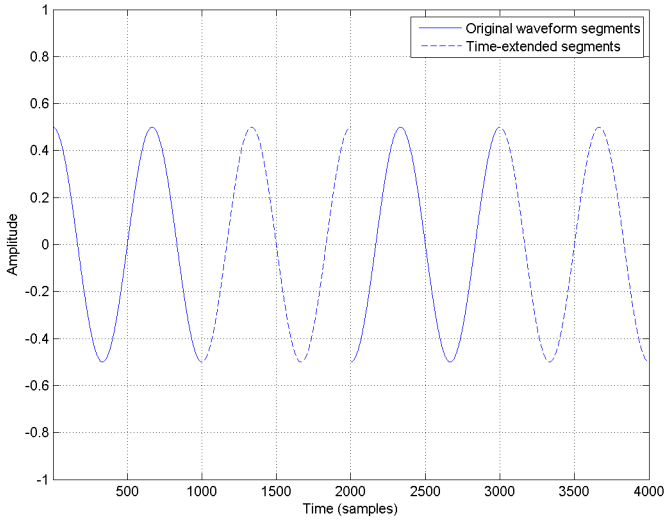

Figure 3.4: Sinusoid stretched by a factor of 2.0 on a per-peak basis without modifying phases

shown in figure 3.4, which has a very large discontinuity between the independently reconstructed sinusoids rather than being a continuous sinusoid as would be expected and desired. While using an overlap-add algorithm during synthesis can help to smooth these kinds of discontinuities, they are still quite noticeable. Clearly, using the initial analysis phase for each time-scaled peak was not correct, but how do we know what other phase to use? If instead of treating each frame independently of every other frame, we knew at synthesis time that the peak in frame two was in reality a continuation of the peak in frame one, then we could adjust the phase of the peak in frame two during reconstruction to match the ending phase of the peak in frame one. This is where the concept of a peak "track" becomes relevant.

A peak track consists of a series of sinusoidal peaks in consecutive frames which, in the original input signal, most likely represented a single sinusoid, varying slowly (or not at all) over time. Once peak tracks have been identified from a set of peaks, rather than time scaling each peak independently, each peak track can be synthesized and scaled as a unit, enforcing internal phase continuity. Therefore, even though the concept of a track is not necessary for unmodified synthesis or

for some applications of sinusoidal modeling, tracks are very necessary for time-scale modification. As a result, before performing time-scale modification on the analysis results from the sinusoidal model described in chapter 2, it is necessary to first generate peak tracks using a peak tracking algorithm which converts the lists of peaks in each frame into lists of peak tracks.

3.2 Sinusoidal tracks

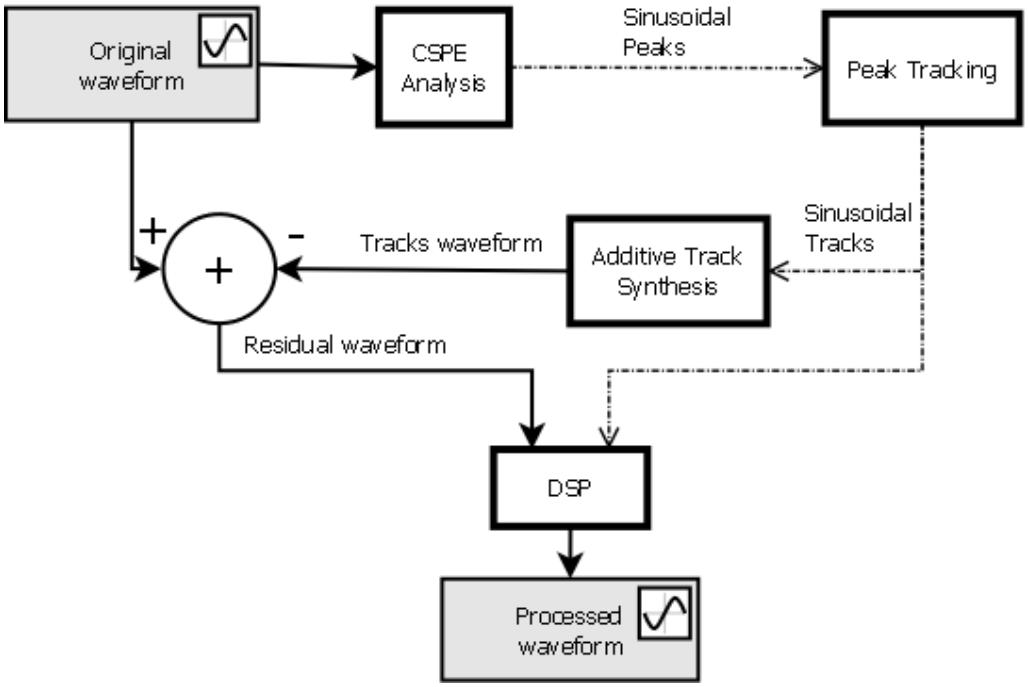

Figure 3.5: Analysis/synthesis process with sinusoidal peak tracking

The analysis/synthesis process described in chapter 2 can be extended to include the use of sinusoidal peak tracks as shown in figure 3.5. This section describes the process of identifying peak tracks from the lists of sinusoids which result from a frame-by-frame analysis as well as the sinusoidal track synthesis process and a method for improving the original signal analysis after the initial identification of tracks.

3.2.1 Tracking sinusoidal peaks

The goal of a peak tracking algorithm is to identify peaks in consecutive analysis frames which represent sinusoids that were continuous across frame boundaries in the original input signal. McAulay and Quatieri [12] and Serra [14] describe a recursive method of tracking peaks which iterates through each peak in a frame and finds the peak in the previous frame which most closely matches it. In the event that two peaks both match the same previous peak, the closest match wins, and the loser must then redo the search, looking for the next best match and possibly displacing another already-matched peak. If a peak does not match an existing track, it becomes the start of a new track. Similarly, if a track is not continued by any peaks, it ends.

This algorithm was used to track spectral peaks that did not necessary represent sinusoidal/tonal elements of the input signal. However, for the purposes of this thesis, because the analysis process described in chapter 2 separates sinusoidal from noisy signal components before peak tracking begins, the tracking algorithm can be modified to help distinguish between sinusoidal components which were truly continuous across frame in the original input signal and those which, even though they are of similar frequency, may in fact have originated from different sources in a polyphonic signal. This distinction is very important for time-scaling applications, because in the case of truly continuous sinusoidal tracks, the phase of the time-scaled peaks in that track should be continuous and free of discontinuities, whereas if there was no continuous track in the original signal, it would be incorrect to assume that the phase should be continuous across frame boundaries in the time-scaled signal.

Therefore, while the general structure of the peak-tracking algorithm used in this thesis is the same as McAulay and Quatieri and Serra's, the criteria that must be met for a peak to be considered a match to an existing peak track have been made quite strict. The resulting peak-tracking process takes as input a frame-by-frame list of sinusoidal peaks identified in the original input signal and returns a frame-by-frame list of the sinusoidal tracks which start in each frame. Each track in turn contains a list of its component peaks.

For each peak, the criteria for be labeled as a match for an existing peak track are as follows:

- Frequency continuity: As shown in equation 3.1, the change in frequency between the last peak in the track to be continued and the current peak must be less than a certain frequency change threshold (ϵ_{freq}). This is the most obvious requirement and has been used in all peak tracking algorithms, such as those by McAulay and Quatieri [12] and Serra and Smith [18].

$$|f_1 - f_2| < \epsilon_{freq} \tag{3.1}$$

- Amplitude continuity: As shown in equation 3.2, the variation in amplitude (in dB) between the last peak in the track to be continued and the current peak must be less than a certain amplitude change threshold (ϵ_{amp}). This requirement is used in some more recent sinusoidal models, such as the work by Levine [11].

$$|dB(A_1) - dB(A_2)| < \epsilon_{amp} \tag{3.2}$$

- Phase continuity: As described next, the difference in phase between two consecutive sinusoids where they intersect must be less than a certain phase change threshold.

Phase continuity

Because of the way in which a peak track is synthesized, even though it may span multiple analysis and/or synthesis frames in time, there are no discontinuities in the phase of the synthesized track even when it is time-scaled. This property can be labeled as horizontal phase coherence, where the word *horizontal* refers to the fact that time is generally the horizontal axis in graphical plots [10]. To accurately reproduce a signal, however, another form of phase coherence should also be considered: vertical phase coherence. Vertical phase coherence traditionally refers to maintaining the phase relationships across frequency channels within a single synthesis frame of a phase vocoder [10]. However, the same concept applies to sinusoidal components, where it is desirable to maintain the phase relationships

between sinusoids across different frequencies rather than across time. In a time-scaled signal, unless the partials of the signal share harmonic relationships (such as in [8]), it is typically not mathematically possible to perfectly preserve both horizontal and vertical phase coherence. Even when phase coupling does occur in real-world signals, it is typically a non-linear effect related to the family of instrument being analyzed [6], further complicating this scenario. The result is that trade-offs must be made between preserving horizontal phase coherence and vertical phase coherence.

In general, it fairly obvious that horizontal phase coherence is more important for sustained sounds and vertical phase coherence is more important for attacks and transients. This is one source of artifacts in both phase vocoder-based and typical sinusoidal model-based time-scale modification, because both approaches favor horizontal over vertical phase coherence. In the case of the phase vocoder, phase continuity is being emphasized across frame boundaries within a particular frequency bin, and in the case of sinusoidal models, phase continuity is being emphasized across a particular sinusoidal track in time.

In this thesis, the choice between horizontal or vertical phase coherence is made during the peak tracking process. If two sinusoidal peaks in consecutive analysis frames are close in frequency and the phase of the second peak matches the first at the point of overlap, then the two peaks will be considered part of the same track, ensuring that when they are time-scaled the phase continuity they had in the original signal is preserved. If, however, the two peaks do not have a similar phase at the point of overlap, it can be assumed that they did not represent a continuous sinusoid in the original signal, but rather it is more likely that the second peak represents the onset of a new sinusoid. As such, these two peaks will not be considered part of the same track, and when synthesized at a modified time scale, the initial analysis phase of the second sinusoid will be preserved. When there is an attack or transient in the original signal, typically many sinusoidal components will turn on or off or experience phase discontinuities at the same instant. If each is identified as the start of a new track rather than the continuation of an old track, then because the initial analysis phases of each

sinusoid are preserved, vertical phase coherence will be maintained and the attack or onset will maintain the same time domain envelope it had in the original signal.

This trade-off between the two kinds of phase coherence thus becomes an important part of the peak-tracking algorithm, and it is the check for peak continuity when determining whether or not a given peak continues an existing track which decides whether horizontal or vertical phase coherence will be favored. If the phase continuity check indicates that the phase in the original signal between the two sinusoids in question was fairly continuous, then a track exists, and whenever a track is resynthesized, it will preserve horizontal phase continuity internally. However, if the phase continuity check determines that there was a significant discontinuity in phase in the original signal between the two sinusoids, they will not be treated as a single track, meaning that the initial phase of the second sinusoid will be preserved. In the case of an onset or attack, the phase continuity check will likely identify discontinuities along most tracks, resulting in vertical phase coherence as the initial phases at the onset of each sinusoid are preserved.

Computationally, phase continuity is more complicated to evaluate than frequency and amplitude continuity. One reason for this is that any distance measure must consider phase values modulo 2π and continuous around the unit circle. Also, unlike with analysis frequency and analysis amplitude, comparing the analysis phase (initial phase) from frame to frame is meaningless in terms of continuity. The actual phase of interest is the phase where the sinusoid in the previous frame in question intersects the sinusoid in the current frame. In the case of frame overlap, there is a region of intersection rather than simply one point. The phase continuity measure used here looks at the point half-way through the overlap region (where the two overlapping sinusoids will be averaged during reconstruction), and checks that the phase of each sinusoid at that point is within a certain phase change threshold (ϵ_{phase}) as shown in equation 3.3, where wrap(ϕ) is a function which wraps the given phase value ϕ into a $[-\pi, \pi]$ range by adding or subtracting integer multiples of 2π.

$$|\text{wrap}(\phi_1 - \phi_2)| < \epsilon_{phase} \tag{3.3}$$

3.2.2 Track synthesis

Like the analysis process, the sinusoidal model synthesis process must also be extended to handle sinusoidal tracks rather than simply synthesizing isolated peaks in each frame. Just as individual sinusoidal peaks can be synthesized on a frame-by-frame basis, sinusoidal tracks can also be synthesized in this manner, but this involves some additional computations to derive envelopes for parameters that may vary over the course of a track, such as amplitude and frequency. Using the same frame size and overlap as the analysis stage, each frame is synthesized by reconstructing and adding together the current section of each track which is "alive" during that frame.

Amplitude

For each of these tracks, an amplitude envelope is generated by taking into account the amplitudes of any peaks in the track whose duration overlapped with the current frame during analysis. Using the fact that a peak's analysis amplitude is the average across the frame, an amplitude envelope for the entire track can be generated from the amplitudes of the individual peaks. At the beginning and end of a track, extrapolation is required to obtain the endpoints of the amplitude envelope, and in these cases the amplitude changes from the last known segment of the envelope are continued linearly either forward or backward according to equations 3.4 for the first breakpoint and 3.5 for the last to guarantee that the average amplitude in the current frame is indeed the analysis amplitude.

$$A_0 = A_1 - (A_2 - A_1) \tag{3.4}$$

$$A_N = A_{N-1} + (A_{N-1} - A_{N-2}) \tag{3.5}$$

Figure 3.6 shows an example of an amplitude envelope with endpoints extrapolated from existing data points. This example assumes 50% overlap, so each segment of the envelope is of equal length (including the end segments).

In the case of a track whose duration is limited to a single frame, there is only one data point from which to create the envelope, resulting in constant

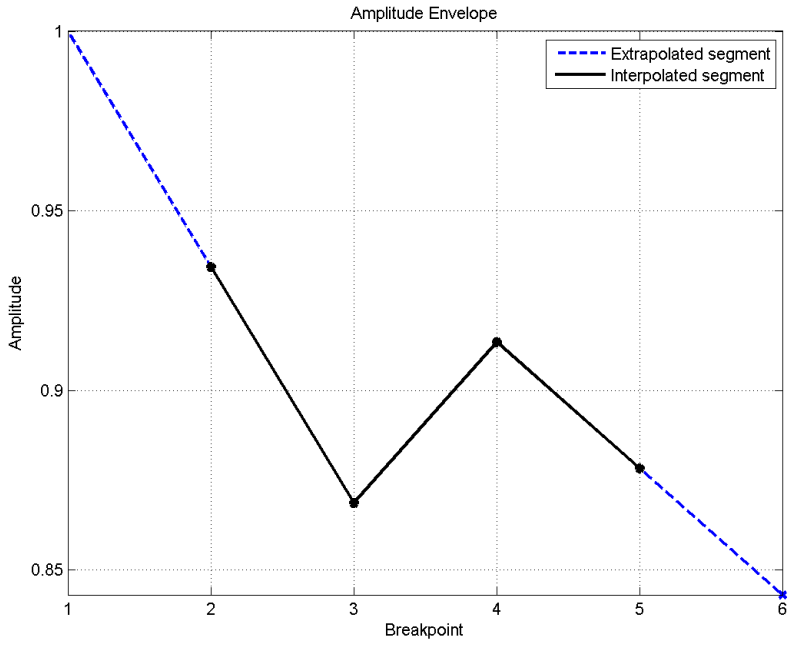

Figure 3.6: Amplitude envelope with linear interpolation between breakpoints

amplitude throughout the frame, and in the cases where amplitude is not constant, the standard approach of linear interpolation is used to obtain the instantaneous amplitude for each sample. Figure 3.6 also shows this linear interpolation between breakpoints. In all of these situations, the overlap-add process will result in the smooth fade in/out of each peak, so the amplitude envelope itself is not responsible for providing that effect.

Frequency

A frequency envelope for the track is generated in an identical manner to the amplitude envelope and is used to determine the phase advance of the sinusoid to be synthesized at each sample in the frame. The question of how to best interpolate frequency and phase from frame to frame using the given frequency envelope is a complicated one. There are two commonly used approaches. In the simplest approach, linear interpolation is used to determine the instantaneous frequency at a given sample, and the phase advance for that sample is computed

accordingly. This method is highly accurate for a "true" sinusoidal track - one in which there was phase continuity in the original signal. However, if there was not phase continuity in the original signal from frame to frame, linear interpolation results in phase errors compared to the original signal (and potentially compared to other parallel sinusoidal tracks) [11]. In those cases, a higher-order interpolation such as the cubic polynomial used by McAulay and Quatieri [12] can be used to ensure continuity of both frequency and phase between frames.

The model described in this thesis assumes a certain degree of phase continuity in the original signal in order for sinusoidal peaks to be considered part of the same track. This forces the start of a new track when phase continuity is not present, and new tracks always begin with the original analysis phase. This is the desired behavior because a lack of phase continuity in the original signal most likely signals the beginning of a new event - either a new note attack or the addition of another "voice" contributing to that particular frequency in a polyphonic signal. Using the initial analysis phase at these moments is intended to improve the quality of attacks and transients. This phase continuity requirement for tracks has an additional benefit, which is that linear interpolation can be used with a reasonable amount of accuracy. As a result, the more computationally complex cubic polynomial interpolation is not used at this time, although it could potentially provide an improvement in audio quality in certain situations such as when tracks are very long due to accumulation of phase errors at each frame boundary.

Initial phase

Once an amplitude envelope has been computed (of which a segment is used for each frame) and sample-by-sample phase advances have been derived from a segment of the frequency envelope for the track, the initial phase is the last parameter which must be calculated for each frame. The analysis phase of the first peak in a track is always used as the initial phase for the first frame in the track, and to ensure continuity in subsequent frames, the initial phase for each segment of the sinusoid is computed based on the sample-by-sample phase advances leading up to the current starting sample.

Synthesis

Just as it is used when resynthesizing frames of peaks, additive synthesis is also used to compute the composite of all sinusoidal tracks which are alive in a given frame. As each frame is synthesized, an overlap-add algorithm is used to combine it with previously-synthesized frames to create the final reconstruction of the sinusoidal portion of the original audio.

3.3 Time-scaled synthesis

3.3.1 Constant scaling factor

Once sinusoidal tracks have been identified, time scale modification of the decomposed audio signal can be performed. This section presents an algorithm for synthesizing the time-scaled version of an input signal given a constant modification factor.

The amount of time scaling to be performed is defined by a parameter α specifying the desired ratio of output signal duration to input signal duration[2]. For example, a scaling factor of $\alpha = 2.0$ stretches the input signal to double its original duration, a scaling factor of $\alpha = 1.0$ retains the original duration, and a factor of $\alpha = 0.5$ condenses the signal to half its original duration. It is important to note that with the approach described here there is no requirement that the modification factor be a factor of two or have any particular relationship to the original analysis frame size used.

Once a time-scale modification factor is specified, the sinusoidal content of the input signal is synthesized such that the onset time and duration of each track are scaled by the given factor. While in the non-scaled synthesis example above, sinusoids are reconstructed on a frame-by-frame basis, for simplicity the time-scaled reconstruction is performed on a track-by-track basis with no concept of frame boundaries or overlap. This reduces the complexity of the synthesis algorithm by eliminating the need to track modified phases from previously reconstructed peaks

[2]Some time-scaling systems define the modification parameter as a scaling factor for playback speed. The two approaches are interchangeable, as one is simply the inverse of the other.

in a given track between frames and making it easy to handle scaling factors which result in tracks starting and/or ending at samples which are not frame boundaries.

The input to the synthesis process is then a list of sinusoidal tracks. While the tracks are by default ordered by the analysis frame in which they start, the order in which they are reconstructed can be arbitrary if desired; the lack of synthesis frames means that tracks starting in earlier frames do not necessarily have to be constructed first, although this is certainly the easiest way to proceed. Iterating through this list in the desired order, each track is reconstructed in its entirety before the next track is handled. As in the non-time-scaled case, amplitude and frequency envelopes along with an initial track phase control the instantaneous parameters of the synthesized sinusoid as described below.

Amplitude

Time-scaled synthesis of a particular track begins with generating an amplitude envelope from the amplitudes of each peak in the track. In the frame-by-frame track reconstruction, the ends of each amplitude are always smoothly faded in and out as a result of the synthesis window used in the overlap-add reconstruction algorithm. However, in the case of the track-by-track synthesis used here, there is no overlap-add and therefore, to ensure smooth track fade-in and fade-out, the start and end points of the envelope are set to zero. Linear interpolation is performed to compute the amplitude of each sample between breakpoints, but when approaching the endpoints, interpolation using a synthesis window can be used to more closely simulate the fade in and out of the overlap-add algorithm in frame-by-frame synthesis.

Frequency

After the amplitude envelope has been computed, a frequency envelope is created from the frequencies of each peak in the track, with start and end point linearly extrapolated from peak frequencies as described in section 3.2.2 on non-scaled synthesis. As described in that section, there are multiple possible ways to compute sample-by-sample phase advances for a synthesized track based

on analysis frequency and phase information, but as in the non-time-scaled case, because phase continuity is already a prerequisite for two sinusoids to be labeled as a track, linear interpolation of instantaneous frequency is a reasonably accurate way to obtain sample-by-sample phase advances, so this approach is also used in the time-scaled case.

Initial Phase

Once an amplitude envelope has been computed and sample-by-sample phase advances have been derived from the frequency envelope for the track, the initial phase is the last parameter which must be calculated for each track. As in non-time-scaled synthesis, the analysis phase of the first peak in a track is always used as the initial phase.

Synthesis

As each track is synthesized from the amplitude envelope, phase advances, and initial phase computed as described above, its sample values are added to the final buffer of output audio to create the synthesized time-scaled waveform.

3.3.2 Variable scaling factor

While this time-scaling algorithm has currently only been implemented using a constant time-scale modification factor throughout a given signal, there is nothing to prevent the same procedure from working for variable scaling factors with only some minor modifications. The output audio buffer would of course require some additional computations to pre-allocated the correct number of audio samples. Then the amplitude and frequency envelopes for each track would have to be recomputed using non-evenly-spaced breakpoints. With the use of track-based instead of frame-based synthesis, there is no need to handle variable frame sizes or other such complicating factors.

Chapter 4

Sinusoidal Modeling Enhancements

4.1 Improved analysis using sinusoidal tracks

The addition of peak tracking to the sinusoidal model described in chapter 2 is very powerful and can more accurately model certain kinds of signals than a frame-by-frame isolated peak model can. However, synthesizing tracks typically also creates some artifacts in the resulting synthesized audio compared to the same content synthesized using individual sinusoidal peaks. These artifacts arise from a difference in the way that each peak is reconstructed during the analysis process compared to the way that the same peak is reconstructed as part of a sinusoidal track. The analysis process has no concept of tracks, as its job is merely to detect individual sinusoids on a frame-by-frame basis. Consequently, it makes the assumption that the frequency of every sinusoid is constant throughout the duration of a frame. Therefore, when the analysis process reconstructs a peak, even if that peak was actually part of a track with modulating frequency in the original audio, the analysis has no way of knowing about the modulation and therefore reconstructs the peak with a constant frequency.

As figure 2.1 showed, sinusoidal peaks are not identified all at once. Rather, they are extracted using an iterative process one-by-one or in small groups, starting

with the peaks whose magnitude is largest. If a peak is incorrectly extracted (which is exactly what occurs when a modulating-frequency sinusoid is extracted as a constant-frequency one), error is introduced into the residual signal that remains. Figure 4.1 shows one frame of a linear chirp signal. After a constant-frequency sinusoid approximating the frequency of the chirp is extracted, the residual (error signal) is as shown in figure 4.2.

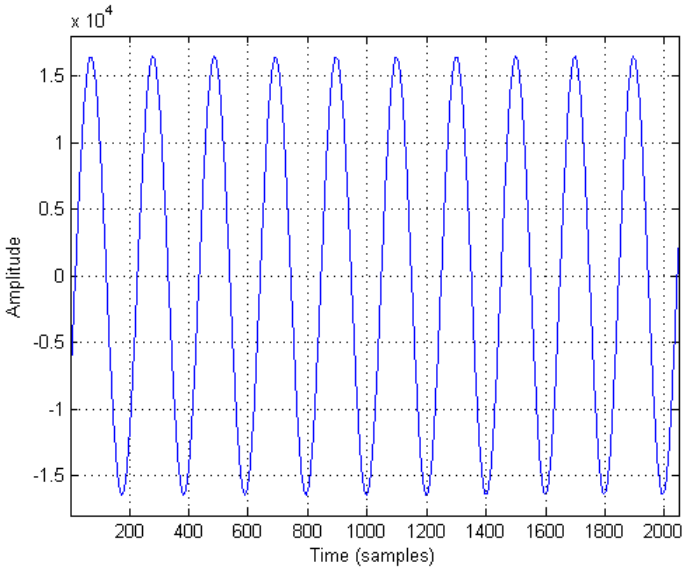

Figure 4.1: One frame of a linear chirp signal (original)

The parameters computed for sinusoids identified during a later iteration in the same frame may be influenced by this error, since each analysis iteration attempts to accurately model whatever residual signal it is given, and therefore in some cases, it will attempt to model the error signal. When track reconstruction is later performed, and the modulating frequency is accurately synthesized, that error is gone but the sinusoids that were modeling it are still present. As a result, some of the track's energy may then be synthesized twice, introducing artifacts into the final synthesized signal.

These artifacts have their roots in the analysis process, but they are purely a track-resynthesis artifact; they do not occur when peak tracking is not used. However, the same problem of reconstructing modulating-frequency sinusoids as

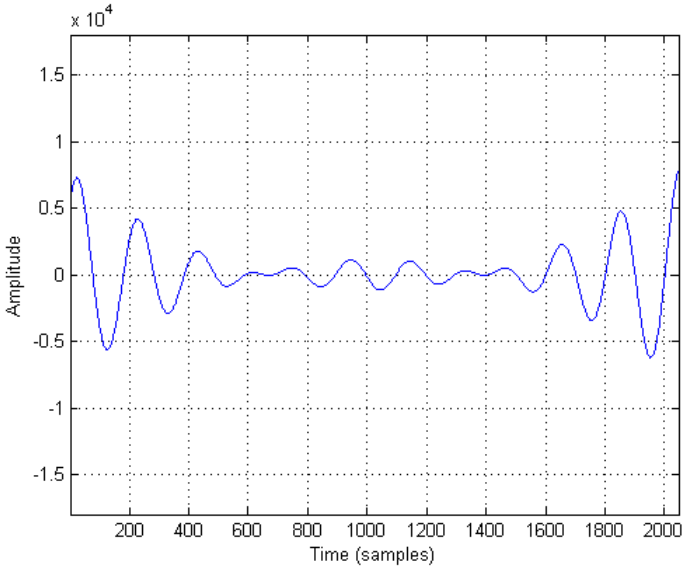

Figure 4.2: One frame of a linear chirp signal after one constant-frequency sinusoid has been extracted (error signal)

constant-frequency ones spawns a second kind of artifact which was already alluded to. In the case above, where the analysis attempts to model an erroneous residual, it may not succeed. Instead, the errors may cascade, since as more peaks are identified and removed from that residual signal, some may be correct, "true" peaks, but some may simply be artifacts of that error previously introduced. Even the correct peaks may have errors in their parameter estimation as a result of the influence of the error in the residual. The artifacts that result from a failed attempt to model this error with additional constant frequency sinusoids are present in the frame-by-frame individual peak reconstruction even before tracks are identified. Proper track identification and synthesis using the resulting correct modulating frequencies may alleviate some of these artifacts, but this is not guaranteed.

The result is two kinds of artifacts - one which occurs in both peak and track resynthesis as a result of analysis error, and one which occurs only when using the tracks which try to compensate for this analysis error. Fortunately, both sets of artifacts can be reduced, and even nearly eliminated, by performing a second iteration of the analysis process, taking into account the varying frequency of tracks according to the following steps:

- Once sinusoidal tracks have been identified from the original set of peaks, filter out and reconstruct only those tracks which have a duration of at least two frames[1]. If any of these tracks represent sinusoids with varying frequency, they will be reconstructed with an interpolated approximation of that variation, which is more accurate than the assumption of constant frequency used in the original decomposition stage.

- When all tracks with $length \geq 2$ frames have been reconstructed, subtract them from the original input signal to produce a new residual which contains everything except for the long sinusoidal tracks (including noise and transient components)

- Repeat the sinusoidal decomposition described in section 2.1, identifying sinusoidal peaks from this new residual rather than the original signal

- Derive new sinusoidal tracks from the newly-identified peaks

- Combine this new list of tracks with the tracks having $length \geq 2$ frames from the original decomposition stage to create a final, complete list of tracks[2].

The resulting revised analysis process is shown in figure 4.3. As will be demonstrated in section 5.2, the addition of this second round of analysis leads to higher accuracy since the residual used as input was calculated by synthesizing varying-frequency sinusoids with less error than during the original analysis process, where sinusoids were reconstructed assuming a constant frequency across each frame. A method for correctly identifying modulating sinusoids during the initial sinusoidal peak detection process without requiring knowledge of future frames would be the ideal way to address these problems, eliminating them completely,

[1] Any track whose duration is only one frame has a constant frequency and is therefore already handled correctly in the original analysis/decomposition

[2] Note that in the event that some of the original tracks were artifacts of analysis error in the initial analysis and not part of the original signal, their inverse will be present in the residual used for the repeated analysis. As a result, an erroneous track and its inverse may both be present in the final list of tracks. While these will be automatically canceled out during the resynthesis process, an additional step can be introduced which identifies and removes such track pairs, thereby reducing the number of tracks that must be synthesized.

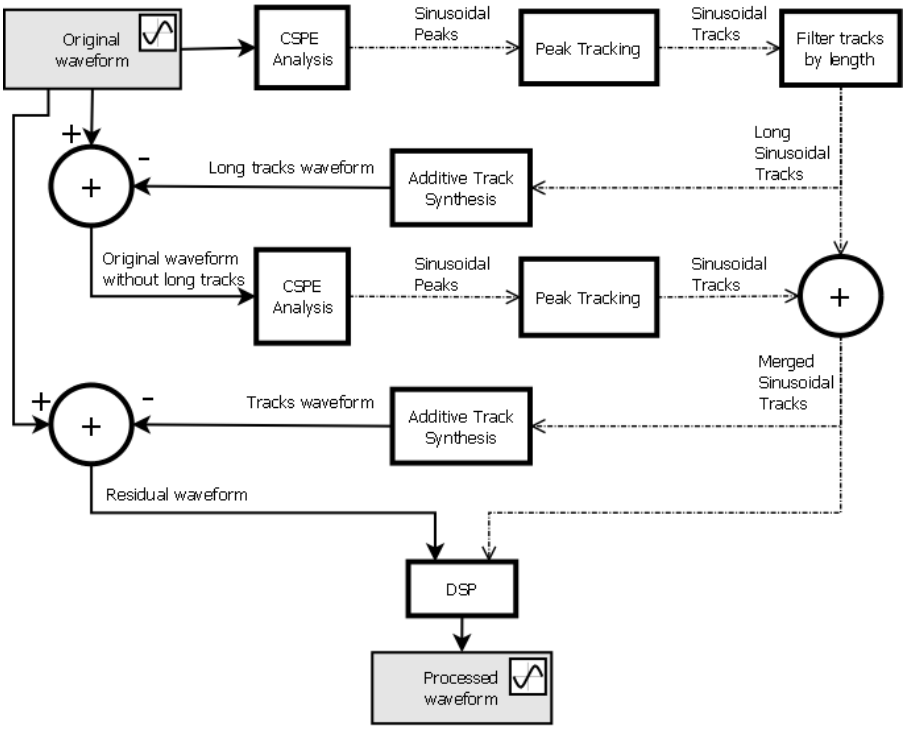

Figure 4.3: Repeated analysis/synthesis process with sinusoidal peak tracking

but when such a method is unavailable, this second round of analysis does a reasonable job of compensating for the shortcomings of a constant-frequency analysis process.

4.2 The Unified Domain: advanced handling of stereo signals

Another enhancement to the sinusoidal model already described in this thesis is the use of the Unified Domain (UD) transformation [17] for improved handling of multichannel audio input. The UD transformation is a lossless and invertible transformation which makes it possible to process all channels of a multichannel signal simultaneously, replacing independent frequency domain magnitudes for each channel with one magnitude value combined with "spatial"[3] angle informa-

[3]"Spatial" here refers to the mathematical space where, in polar coordinates, the left and right channels are separated by $\frac{\pi}{2}$ radians rather than physical space, where perception of the

tion which encodes the distribution of that magnitude between channels. While the UD transformation can also be utilized with larger numbers of channels, this section will focus on the most common and simplified case of stereo signals. For more mathematically rigorous definitions and extensions to higher channel counts, the interested reader is referred to [17].

4.2.1 Unified Domain peaks

We can define a traditional monophonic sinusoidal peak using three parameters: frequency (f), magnitude (A), and phase (ϕ). A stereo peak can be identified when there are peaks in both the left and right channels with very similar frequencies (magnitude and phase need not necessarily be similar). Such a stereo peak can then be described by the frequencies for each channel (f_L and f_R), independent magnitudes for both channels (A_L and A_R), and independent phases for both channels (ϕ_L and ϕ_R). In the Unified Domain, that same stereo peak is instead defined by a single frequency (f_{UD}), a single magnitude (A_{UD}), a single spatial magnitude angle (σ), and independent phases for each channel (ϕ_L and ϕ_R), where A_{UD} is given by equation 4.1, and σ is given by equation 4.2.

$$A_{UD} = \sqrt{A_L^2 + A_R^2} \tag{4.1}$$

$$\sigma = \arctan(\frac{A_R}{A_L}) \tag{4.2}$$

For example, a peak whose energy is entirely in the left channel will have $\sigma = 0$, while a peak whose energy is shared equally between channels will have $\sigma = \frac{\pi}{4}$, and a peak whose energy is entirely in the right channel will have $\sigma = \frac{\pi}{2}$ (σ is always in the range $[0, \frac{\pi}{2}]$).

This UD peak representation is especially useful when identifying and extracting sinusoids from an input signal during sinusoidal modeling. Consider the following scenario: a sinusoid in a stereo signal is panned mostly to the left channel using amplitude panning but it still has some small amplitude in the right

actual physical location of a sound depends on such factors as the relative phase of the channels and the location of loudspeakers

channel. An algorithm which detects and extracts sinusoids on a per-channel basis may detect the sinusoid in the left channel with no difficulty, but the one in the right channel, while part of the same stereo peak, may be ignored due to its low amplitude. When the two channels are resynthesized, this peak will appear only in the left channel, since its right channel component will be missing. As a result, it will have a lower amplitude than in the original and its position in the stereo field will not be accurate. With the UD approach, however, the peak will either be detected in both channels or in neither channel, depending on its UD magnitude. This can greatly reduce stereo image artifacts that may occur in a resynthesized signal when only one channel of a sinusoid's energy is reproduced.

4.2.2 Sinusoidal decomposition

The UD transformation can be used throughout the analysis, synthesis, peak tracking, and time-scaling processes for improved results with stereo signals. In the initial analysis stage, where the signal is being decomposed into sinusoidal components, the UD can be used to extract stereo UD peaks at each iteration rather than monophonic ones. The analysis process described in section 2.1 then becomes the following:

1. Window the time domain data for both channels and compute each channel's DFT

2. For each channel, compute the CSPE according to equation 2.1 to obtain an array of frequency values indicating which frequency contributed significantly to the energy in a given DFT bin

3. Compute the UD magnitude for each DFT bin according to equation 4.1

4. Using the UD magnitude vector computed in step 3, identify the DFT bin containing the spectral peak with the most energy

5. If the frequencies contributing to this energy in the CSPE vector for each channel are close to the center frequency of the DFT bin in question, a

sinusoidal component was likely present here in the original signal, so we identify this as a UD peak.

6. Estimate the UD frequency (f_{UD}) of the sinusoid using equation 4.3 which computes a magnitude-weighted average of the frequencies in the left and right channels (f_L and f_R) which, especially in a polyphonic signal, may be slightly different due to the interference of other components.

$$f_{UD} = \frac{(A_L f_L) + (A_R f_R)}{A_L + A_R} \tag{4.3}$$

7. Estimate the magnitude and phase for both channels of the chosen peak using the spectra computed in step 1 and the spectrum of the analysis window, according to equation 2.2

8. Compute the UD magnitude and spatial angle σ from the left and right channel magnitudes using equations 4.1 and 4.2

9. Resynthesize both channels of all sinusoids that have been detected in the current frame so far

10. Subtract the resynthesized peaks from the input to compute a new residual containing one less sinusoidal component.

11. Repeat steps 1 through 10 until the desired number of peaks have been extracted or no more sinusoidal components have been detected

12. Store lists of the UD peaks detected in each frame including their frequency, UD magnitude, spatial angle, and phase parameters as well as the residual signal containing non-sinusoidal elements.

4.2.3 Peak synthesis

Synthesis of UD peaks is performed in the time domain in the same manner as synthesis of monophonic peaks. The only difference is that left and right channel amplitudes[4] are derived from the UD magnitude A_{UD} and the spatial magnitude

[4]Recall from section 2.2 that the frequency domain magnitudes are scaled to correspond to time domain amplitudes during the analysis process - the same scaling occurs in the UD case

angle σ, resulting in equations 4.4 and 4.5 for synthesizing the left channel ($\vec{s_L}$) and right channel ($\vec{s_R}$) samples respectively.

$$\vec{s_L}(t) = A_{UD}\cos(\sigma)\cos(\frac{2\pi t f_{UD}}{N} + \phi_L) \tag{4.4}$$

$$\vec{s_R}(t) = A_{UD}\sin(\sigma)\cos(\frac{2\pi t f_{UD}}{N} + \phi_R) \tag{4.5}$$

Once the left and right channels for a given frame have been synthesized, an overlap-add algorithm is used (as in the monophonic case) to concatenate adjacent frames.

4.2.4 Peak tracking

When tracking Unified Domain peaks, the criteria used to determine continuity in a monophonic signal must be slightly altered to accommodate the parameters of a UD peak, so the list in section 3.2.1 becomes the following:

- Frequency continuity: the change in frequency between the last peak in the track to be continued and the current peak must be less than a certain frequency change threshold (as shown in equation 3.1 for a non-UD peak)

- UD amplitude continuity: the variation in UD amplitude (in dB) between the last peak in the track to be continued and the current peak must be less than a certain amplitude change threshold (as shown in equation 3.2 for non-UD amplitude values)

- Phase continuity: A UD peak contains initial phases for both input channels, so both channels' phases must satisfy the constraint of phase continuity shown in equation 3.3 for non-UD peaks.

- Spatial continuity: Spatial angle continuity is a unique requirement for UD peaks which does not exist in the monophonic case. As shown in equation 4.6, the change in UD spatial angle from peak to peak must be below a certain threshold (ϵ_σ).

$$|\sigma_1 - \sigma_2| < \epsilon_\sigma \tag{4.6}$$

For example, one could require that a peak which was entirely in one channel in one frame but was amplitude panned entirely to the other channel in the next frame be treated as two independent peaks, while a peak which was amplitude panned slowly over a period of multiple frames might be treated as a single continuous track.

4.2.5 Track synthesis and time scaling

Synthesizing UD sinusoidal tracks with or without time-scale modification is nearly identical to the process of synthesizing monophonic tracks, so the benefit of UD usage is primarily during the analysis process and not the synthesis process. The only difference is that, as when synthesizing individual peaks, left and right channel track amplitudes must be derived from the UD magnitude and spatial magnitude angle before an individual peak can be synthesized. Like in non-UD track synthesis, separate amplitude envelopes can be computed for each channel of the track separately, or, alternatively, a single UD magnitude envelope can be computed and used on the fly along with a spatial magnitude angle envelope to derive left and right channel amplitudes as needed.

Chapter 5

Discussion

Evaluating the quality of a sinusoidal model and time-scale modification algorithm is a challenging problem. Audio quality measurements can be highly subjective, and in most cases there is no single mathematically correct answer against which one can compare the accuracy of results. Despite these challenges, some clear comparisons can be made between different approaches, and this chapter will attempt to illustrate the success of certain aspects or parameters of the sinusoidal model and time scaling process by comparing results with and without these features enabled.

5.1 Success of the phase continuity measure

One of the most significant differences between the sinusoidal model described in this thesis and more traditional models is the consideration of phase during the peak-tracking process. When deciding whether a sinusoid is a continuation of a previous peak, not only must the sinusoid's frequency and amplitude be similar to the frequency and amplitude of the previous peak in the track, but there must also be continuity in the phase between the two adjacent peaks. A parameter which controls the amount of phase error allowed between two consecutive peaks in a track can be introduced in the sinusoidal model implementation (ϵ_{phase} as described in section 3.2.1). This section will discuss the effect of modifying that parameter on the quality of both non-time-scaled and time-scaled results.

5.1.1 No time scaling

Without time-scale modification, the results of the sinusoidal model analysis and synthesis can always be directly compared to the original input audio signal for accuracy, so we will be begin by discussing the non-time-scaled case. In general, unless there are few phase discontinuities in the original signal, the synthesis of tracks identified using the phase continuity restriction will result in a waveform much closer to the original signal than a waveform synthesized from tracks identified with no phase restrictions. The following example illustrates this by comparing the reconstructed waveform with no phase continuity restriction (e.g. $\epsilon_{phase} = 2\pi$) to the reconstructed waveform of a sample of audio using a very restrictive phase continuity factor (e.g. $\epsilon_{phase} = 0.05$ radians).

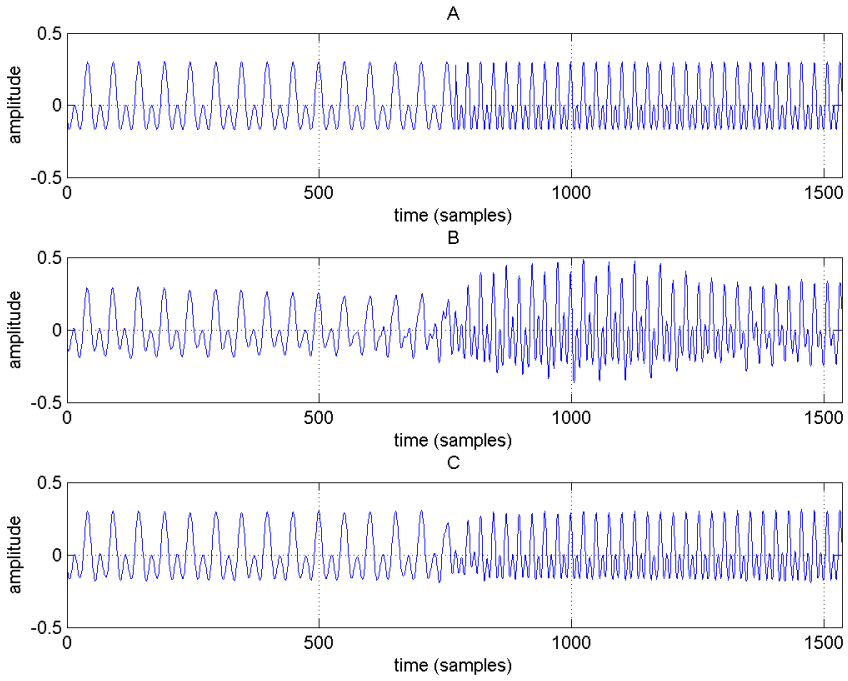

Figure 5.1: A) Original waveform B) Synthesized tracks with no phase continuity requirements C) Synthesized tracks with strict phase continuity requirements

Plot A in figure 5.1 shows a waveform consisting of multiple sinusoids. In the first half of the segment shown, two sinusoids with frequencies f_1 and f_2 are present. Both turn off half-way through the segment. At the same time that these

two sinusoids turn off, two new sinusoids turn on: one with the same frequency f_2 and one with a new frequency f_3 as shown in figure 5.2. As shown in figure 5.3, the initial phase of the sinusoid with frequency f_2 in the second half is not the same as the ending phase of the sinusoid with the same frequency in the first half, because these represent two distinct events and not one continuous sound (notice the broadband spectral content introduced in figure 5.2 as a result of this discontinuity in the phase of f_2).

With no phase continuity restriction (equivalent to a restriction of $\epsilon_{phase} = 2\pi$) during the peak tracking process, the two sinusoids with frequency f_2 are incorrectly identified as being part of the same sinusoidal track. In figure 5.1, plot B, we see the result of synthesizing these sinusoids as a single continuous peak track based on the loose phase continuity restriction, resulting in the wrong phase across the onset of the second note. There is a significant difference between this waveform and the original shown in plot A. While this example uses linear interpolation, even using cubic polynomial phase interpolation would not solve this problem completely since there is no way it could accurately reproduce the discontinuity shown in figure 5.3.

However, with a strict phase continuity restriction of $\epsilon_{phase} = 0.05$ radians, these two sinusoids are correctly identified as being two separate sinusoidal tracks. This makes a big difference in the accuracy of the synthesized time domain waveform compared to the original signal. For example, plot C in figure 5.1 shows the resynthesized waveform when phase continuity is enforced as a condition for being identified as a peak track. There is a small difference between this result and the original shown in plot A, but it is very minor compared to the significant distortion when no phase continuity was required, as shown in plot B.

This is an example of a situation which is not uncommon in polyphonic music: two distinct "voices" with frequency components of the same or very similar frequency either superimposed in time or played consecutively. The example above demonstrated how the phase continuity measure described in this thesis helps to ensure that onsets of sinusoids will retain the correct phase at synthesis time by not assuming a sinusoidal track is present when there was no phase continuity in

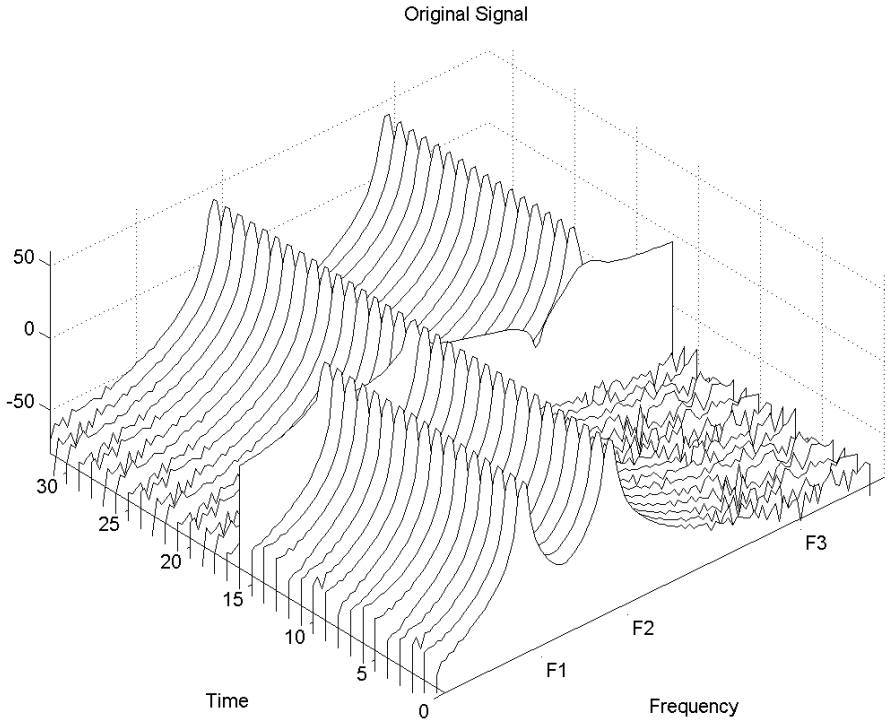

Figure 5.2: Waterfall plot of original signal's spectrum

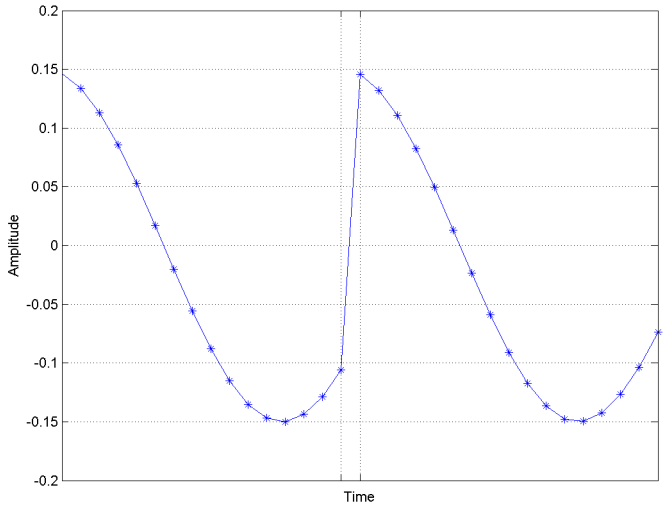

Figure 5.3: Discontinuity between sinusoids with frequency f_2

the original signal.

5.1.2 Time scaling

While the example above illustrated the non-time-scaled case, significant improvements in audio quality also result from using the stricter phase continuity measure with various time-scale modification factors. The following example uses a test signal similar to that shown in 5.2 but extended to include a third "note" that includes the same frequency f_2 that is present in the first two notes in addition to a new frequency component, $f4$. Just as there is a phase discontinuity in f_2 between the first two notes, there is also a phase discontinuity between the second and third notes. The spectrogram of the resulting three-note test signal is shown in 5.4.

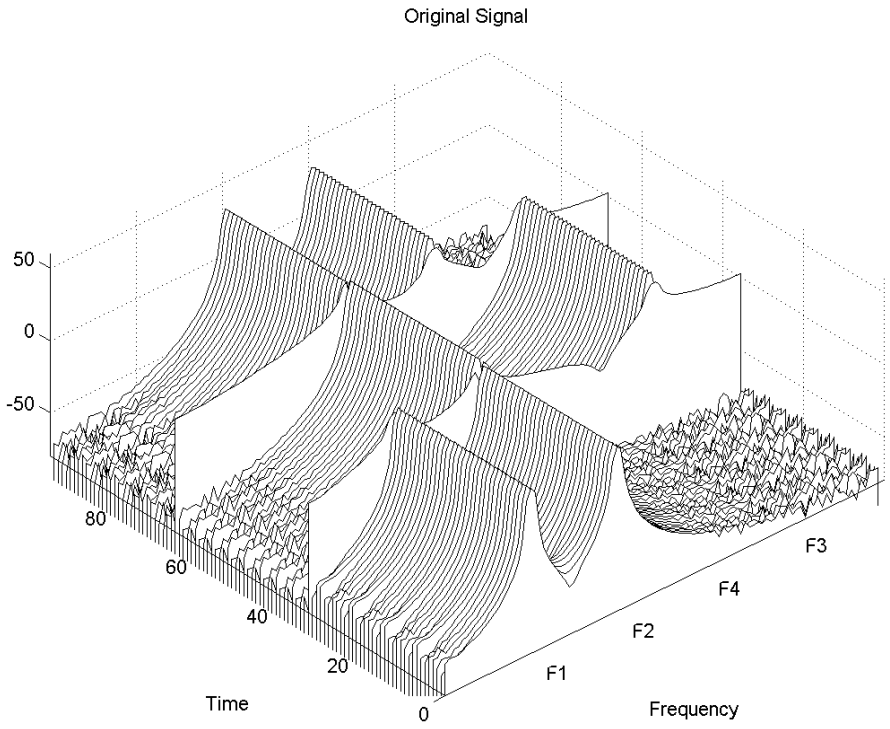

Figure 5.4: Waterfall plot of original three-note signal's spectrum

Because of this signal's simplicity, it was also possible to construct ideal

time-scaled versions of the signal by simply synthesizing each of the three notes for a scaled duration compared to the original. This gave a mathematically perfect time-scaled version of the signal which could then be used to compare the accuracy of time-scaled results with different phase continuity factors. When this three-note signal is time-scaled using a range of modification factors, the resulting Mean Squared Error (MSE) for each version compared to the perfect time-scaled version (sample-by-sample in the time domain) is listed in tables 5.1 and 5.2. In table 5.1, the MSE was computed across the entire time-scaled signals, whereas in table 5.2, the MSE was computed for a constant number of samples around the phase discontinuities, independent of scale factor. This allows us to compare the effect of the phase continuity measure both across the entire signal (table 5.1) and locally around the discontinuities (table 5.2).

With both error measurement approaches, the version where strict phase continuity was enforced during peak tracking does noticeably better with every scale factor up to a factor of 2.0, where the two performed nearly identically. In the case of this particular signal, it was a coincidence that given the frequency of the sinusoid at f_2, the scaling factor of 2.0 happened to be precisely the right value such that the phase was actually continuous in the ideal, or correct, time-scaled version. Since there was no phase discontinuity, the use of the phase continuity factor was irrelevant and the two versions performed nearly identically.

As the time scaling factor is increased beyond 2.0, the benefit of the phase continuity measure for this particular signal becomes much less significant, and in fact, based on MSE, the version of the time-scaled signal without enforcing phase continuity actually outperforms the strict phase continuity version for scale factors 3.0 and 4.5. The improvement in these cases is more significant when measured across the entire time-scaled signal rather than simply around the discontinuities. This occurs because, while there is still a local improvement at the point of discontinuities, the error measurement in the longer time-scaled signals gives greater weight to the continuous segments of the signal. If the analysis process perfectly detects the correct analysis phase for the sinusoid at f_2 at the onsets of the second and third notes, then there will be no error in the time-scaled signals using the

phase continuity measure, while there will be error in the signals not using the phase continuity measure as a result of continuing the phase from the first note. However, if the analysis process does not detect the correct phase for the sinusoid at f_2, there will be error in both signals throughout the sustained portion of the signal. Depending on the scaling factor, the phase error of the continued sinusoid (such as with the scaling factor of 2.0 as described above) may actually be less than that of the non-continued sinusoid, resulting in greater MSE across the duration of the continuous notes.

Another factor is at work here, making the MSE greater for larger scale factors than it is for smaller ones, and that is the general lack of handling of transient components in the current sinusoidal model. Because the sinusoidal components alone cannot perfectly recreate the phase discontinuities between notes, error is introduced in the synthesized signal around the points of discontinuity. When reconstructed with time scaling factors greater than 1.0, the sinusoidal components are stretched in time as desired, but the errors around the discontinuity are also stretched over a greater number of samples, and as a result they effect a larger proportion of the signal than they do when the scaling factor is less than 1.0. So while in general the use of the phase continuity measure performs quite well, it would perform best when integrated with an approach which provides a cleaner analysis at the points where sharp attacks and onsets occur.

While this simple signal performs very well with the strict phase continuity requirement during time-scale modification, more complex polyphonic signals do not always perform as well. Having no phase continuity requirement is not acceptable because it introduces artifacts even into non-scaled results. However, a very strict continuity requirement such as $\epsilon_{phase} = 0.05$ radians causes its own problems. In polyphonic signals it is not uncommon to have continuous sinusoidal tracks either overlapped by other tracks of similar frequency or blurred by noise. When that happens, even though the underlying sinusoid was mostly continuous in the original signal, errors in frequency and/or phase estimation can cause the phase continuity measure to miss cases where tracks should have been identified. While these cases typically sound fine with no time-scale modification, they do not scale

Table 5.1: Mean Squared Error for different time scale modification factors with and without phase continuity requirement (across the entire scaled signal)

Scale Factor	**MSE No Phase Continuity**	**MSE Strict Phase Continuity**	**Improvement With Strict Continuity**
0.25	3.87×10^{-2}	2.38×10^{-2}	2.11 dB
0.4	3.73×10^{-2}	1.18×10^{-2}	5.00 dB
0.5	4.64×10^{-2}	7.15×10^{-3}	8.12 dB
1.0	5.61×10^{-4}	1.47×10^{-5}	15.81 dB
1.1	1.78×10^{-2}	3.01×10^{-3}	7.73 dB
1.5	4.82×10^{-2}	7.43×10^{-3}	8.12 dB
1.8	3.89×10^{-2}	1.97×10^{-2}	2.95 dB
2.0	2.49×10^{-2}	2.47×10^{-2}	0.04 dB
2.5	7.55×10^{-2}	4.47×10^{-2}	2.28 dB
3.0	4.34×10^{-2}	5.91×10^{-2}	-1.34 dB
3.5	6.80×10^{-2}	6.33×10^{-2}	0.31 dB
4.5	3.81×10^{-2}	5.16×10^{-2}	-1.31 dB

Table 5.2: Mean Squared Error for different time scale modification factors with and without phase continuity requirement (across a constant number of samples)

Scale Factor	**MSE No Phase Continuity**	**MSE Strict Phase Continuity**	**Improvement With Strict Continuity**
0.25	4.57×10^{-2}	2.65×10^{-2}	2.36 dB
0.4	5.05×10^{-2}	1.45×10^{-2}	5.43 dB
0.5	5.84×10^{-2}	9.68×10^{-3}	7.81 dB
1.0	1.11×10^{-3}	5.62×10^{-5}	12.96 dB
1.1	1.92×10^{-2}	7.03×10^{-3}	4.38 dB
1.5	6.01×10^{-2}	1.17×10^{-2}	7.10 dB
1.8	4.17×10^{-2}	3.46×10^{-2}	0.81 dB
2.0	3.45×10^{-2}	3.36×10^{-2}	0.12 dB
2.5	8.58×10^{-2}	5.77×10^{-2}	1.72 dB
3.0	6.47×10^{-2}	7.24×10^{-2}	-0.49 dB
3.5	7.35×10^{-2}	6.90×10^{-2}	0.28 dB
4.5	3.75×10^{-2}	3.93×10^{-2}	-0.21 dB

very well because of a lack of horizontal phase coherence in the correct locations. In such situations, significant improvement in the quality of time-scaled results can be obtained by somewhat relaxing the phase continuity requirement during the track identification stage. Transient and attack smearing becomes more problematic as a result, but the increased quality due to better horizontal phase coherence seems to perceptually outweigh such artifacts.

5.2 Success of redoing the analysis

5.2.1 No time scaling

Section 4.1 described performing a second iteration of the analysis process in order to "clean up" signals containing tracks with modulating frequency. One example of a case where this process of repeating the analysis has resulted in improved quality is a test signal consisting of a linear chirp. The chirp is a single modulating sinusoidal track, and it is very poorly modeled by the assumption of constant peak frequency in the original analysis stage. In this case, repeating the analysis using the correct reconstruction of the track which models the chirp dramatically reduces or eliminates many of the artifacts which were present in the originally synthesized tracks, as will be described in this section.

The spectra of two different segments of the original linear chirp signal are depicted in plot A of figures 5.5 and 5.6. In these plots, the chirp itself is clearly visible just above frequency bin 40 as the only signal component present.

Plot B in both figures shows the same segments of the chirp synthesized from sinusoidal tracks after the first round of analysis. Note that artifacts in the form of significant energy have been introduced just above bin 140 in both cases along with noise in the range of bins 60 to 90 in figure 5.5. While the lower amplitude and lower frequency artifacts are less objectionable, the artifact around bin 140 does not occur continuously and its frequency is far away from that of the true chirp, making it very audible and objectionable.

Fortunately, this is exactly the kind of artifact which the re-analysis process described in section 4.1 is intended to address, because it is caused primarily by

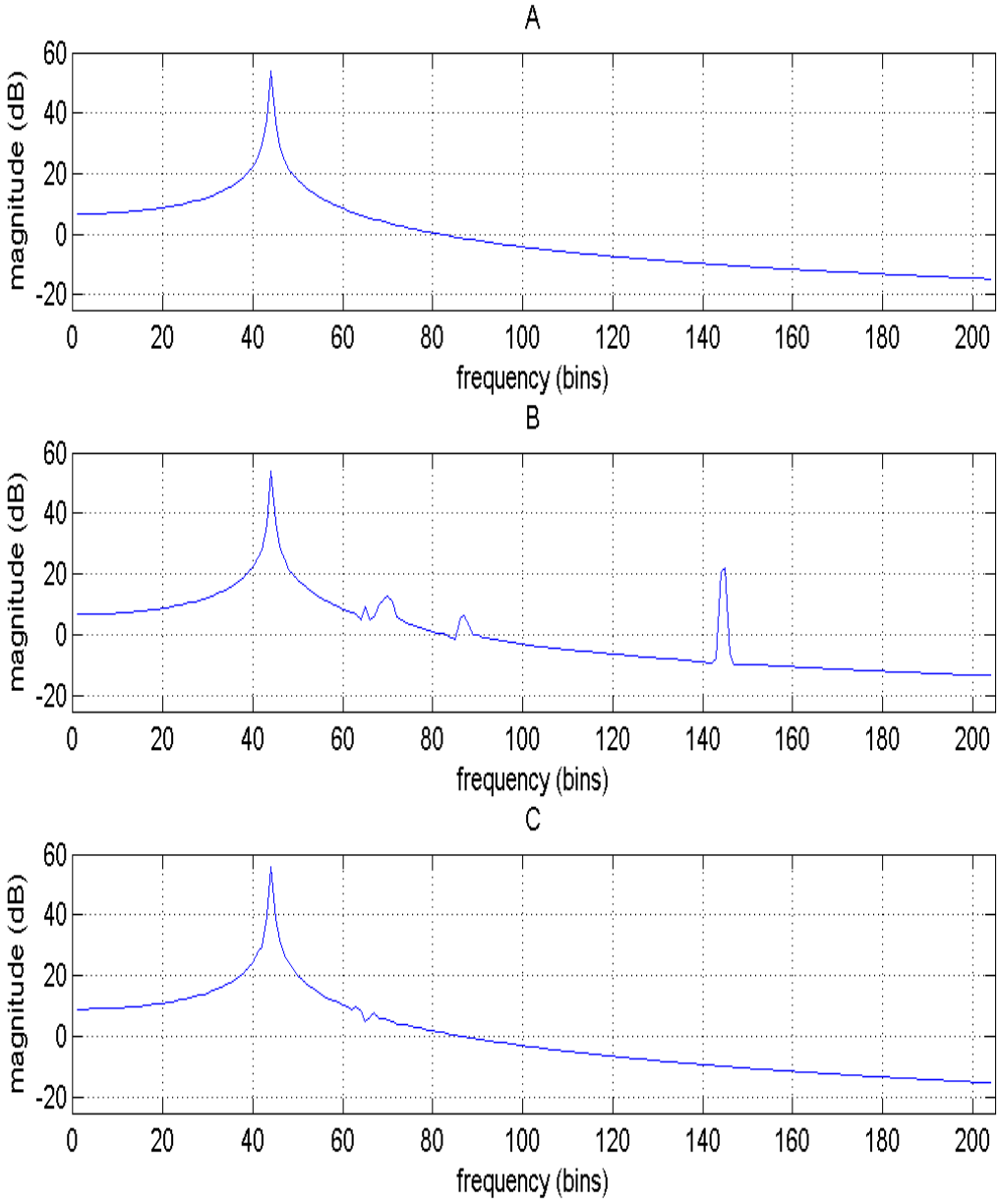

Figure 5.5: A) Spectrum of the original chirp signal B) Spectrum of the chirp signal resynthesized with tracks C) Spectrum of the chirp signal resynthesized with tracks after second analysis stage (significant improvement)

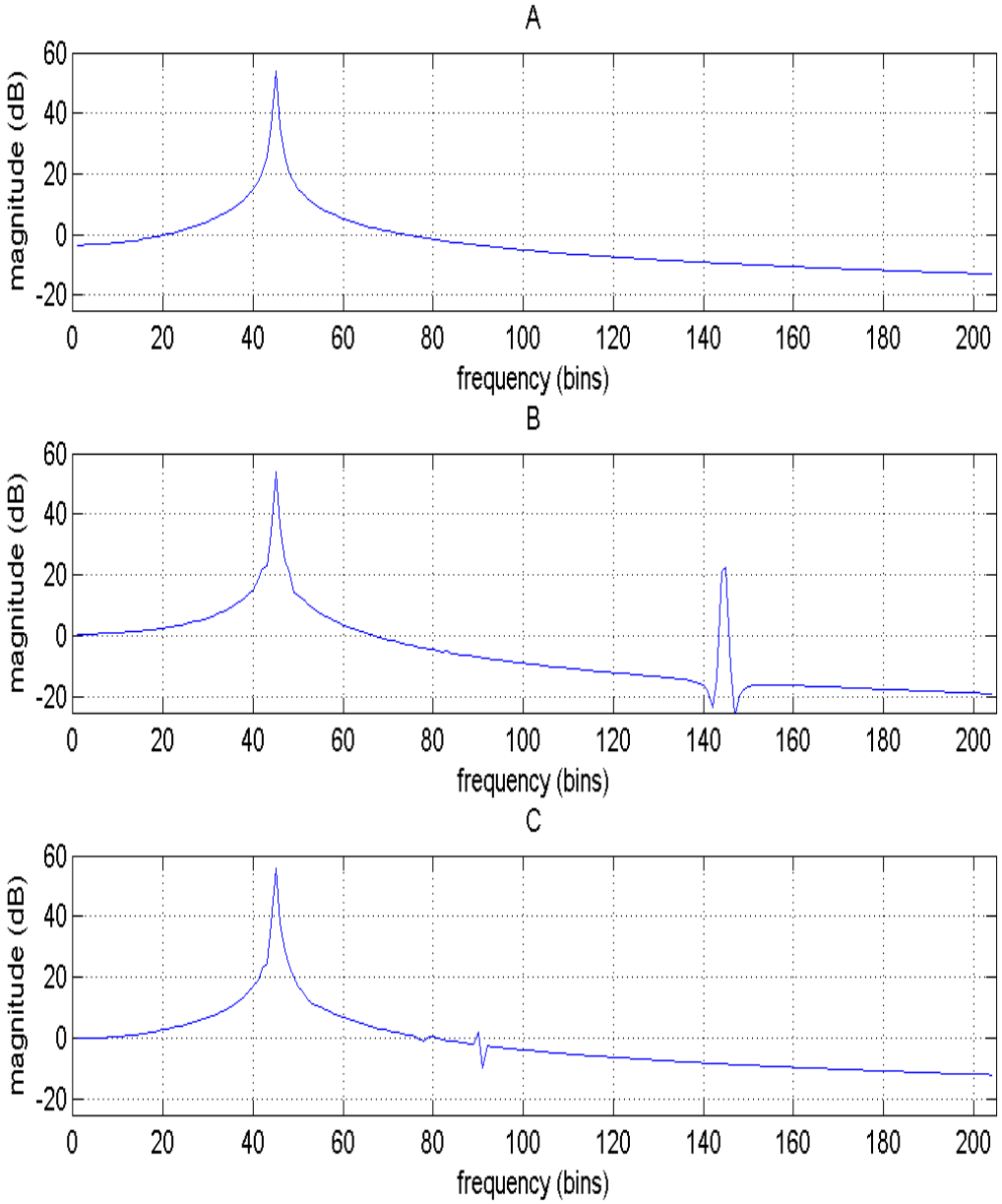

Figure 5.6: A) Spectrum of the original chirp signal B) Spectrum of the chirp signal resynthesized with tracks C) Spectrum of the chirp signal resynthesized with tracks after second analysis stage (good improvement)

Table 5.3: Mean Squared Error for linear chirp with different time scale modification factors with and without the repeated analysis

Scale Factor	**MSE Original Analysis**	**MSE Repeated Analysis**	**Improvement With Repeated Analysis**
0.25	5.04×10^{-4}	3.85×10^{-5}	11.17 dB
0.5	5.01×10^{-4}	1.82×10^{-5}	14.40 dB
1.0	4.40×10^{-4}	2.02×10^{-5}	13.38 dB
1.5	7.07×10^{-4}	5.75×10^{-5}	10.90 dB
2.0	7.30×10^{-4}	2.74×10^{-4}	4.27 dB
3.0	7.09×10^{-4}	1.74×10^{-4}	6.09 dB

side-effects of the fact that the chirp is modulating rather than maintaining a constant frequency from bin to bin. As expected, after performing the repeated analysis, the magnitude of this artifact is significantly reduced (by 30dB or more in places), shown in plot C in both figures. Other artifacts still remain, and in figure 5.6 some error is actually introduced around bin 90 where there was none in the first round of analysis, so the repeated analysis is not a cure-all (at least not with one repetition). However, overall the artifacts in the second round of tracks are a significant improvement over the first set of tracks identified.

5.2.2 Time scaling

The improvements described in section 5.2.1, due to the repeated analysis, carry over from the non-time-scaled case to the use of various time scale modification factors. Given a linear chirp signal, it is possible to synthesize "correct" time-scaled versions of the original signal by creating a chirp which linearly sweeps the same frequency range as the original but over a scaled duration. Using these time-scaled versions of the original chirp as a reference, the Mean Squared Error (MSE) can be computed for the linear chirp scaled after the original analysis and the same chirp scaled after repeating the analysis as described in section 4.1. Similar to tables 5.1 and 5.2, table 5.3 lists the MSE for this linear chirp scaled by different modification factors in both cases. There is a noticeable improvement with all scaling factors when the repeated analysis is used.

Table 5.4: Mean Squared Error for linear chirp with different time scale modification factors with and without the repeated analysis (doubled frequency slope)

Scale Factor	**MSE Original Analysis**	**MSE Repeated Analysis**	**Improvement With Repeated Analysis**
0.25	2.34×10^{-3}	8.86×10^{-5}	14.21 dB
0.5	2.00×10^{-3}	6.95×10^{-5}	14.59 dB
1.0	1.72×10^{-3}	4.85×10^{-5}	15.51 dB
1.5	3.00×10^{-3}	9.66×10^{-5}	14.92 dB
2.0	3.16×10^{-3}	1.59×10^{-4}	13.00 dB
3.0	3.00×10^{-3}	1.79×10^{-4}	12.14 dB

These improvements to the chirp signal are even more pronounced when the slope of the chirp is increased so that the frequency changes more rapidly. This is to be expected, since in this case, the errors due to estimating a constant frequency during the original analysis process are greater than they are when the frequency slope is smaller. This case is illustrated by the results shown in table 5.4, which were computed using a linear chirp signal whose frequency slope was twice that of the chirp used to obtain the results shown in table 5.3.

As we have seen, the chirp benefits greatly from the use of the repeated analysis process. However, the artifacts caused by incorrect reconstruction of modulating sinusoids are less perceptible in more complex polyphonic signals. This is likely because they can be masked or prevented by the presence of additional higher-magnitude sinusoids occurring in the same frame which prevent the analysis process from wasting sinusoids trying to reconstruct an error signal that was not present in the original. Therefore there is less of an audible benefit to repeating the analysis for this more complex signals. However, with properly identified tracks, the repeated analysis is still more mathematically accurate than the original analysis, so while extra computations must be performed, it may be beneficial to repeat the analysis of every signal in this manner unless computing resources are very limited.

5.3 Conclusions

There are many approaches to audio time-scale modification. Both time domain and frequency domain techniques can be effective for different kinds of audio signals. In the frequency domain, fundamental limitations of the phase vocoder make it ineffective for noisy signals and those with transient components. Sinusoidal modeling, while more computationally expensive than the phase vocoder, offers more flexibility for handling more diverse audio signals. One of the most promising approaches to sinusoidal modeling involves the decomposition of input audio signals into three kinds of components: sinusoids, noise, and transients. This three-way decomposition allows each of the component types to be processed in ways which take advantage of their particular characteristics. This can be a particularly valuable scheme for time-scale modification, where it is often desirable to scale sinusoidal and noise components while preserving the original attacks and transients by shifting them in time but not modifying their time domain envelopes. Therefore, to best time-scale audio using a sinusoidal model, three distinct procedures must be implemented: one to time-scale the sinusoidal components, one to time-scale the noise components, and one to time-scale the transient components.

The goal of this thesis has been to address the first of these procedures by demonstrating an effective way of time-scaling the sinusoidal components of a signal. This process begins with the use of the CSPE method to separate the sinusoidal components from noise and transient components. Once these sinusoidal components have been extracted, sinusoidal tracks are identified from the lists of sinusoidal peaks in each analysis frame. Here, a new phase continuity measure is used to help prevent tracks from being created out of sinusoids which were not continuous in the original audio signal. This approach helps to ensure better vertical phase coherence at transients and attacks while still providing horizontal phase coherence during the steady and sustained segments of the signal. Finally, enhancements to the basic analysis/synthesis method were introduced. These included a method for reducing artifacts when analyzing signals with modulating frequency over sinusoidal tracks using an iterative analysis approach, and the use of a transformation which helps improve the quality of results for multichannel in-

put signals. All of these factors combine to create an effective way to time scale the sinusoidal part of a sinusoidal model decomposition. Combined with high-quality approaches to time scaling the noise and transient components, this would make an excellent foundation for a complete sinusoidal model-based time-scale modification system.

Appendix A

Implementation Notes

The sinusoidal analysis/synthesis system used to obtain the results presented in this thesis was implemented by the author using Matlab R2008b Student Version. It has been tested successfully on Windows XP and Mac OS X 10.5.

Functionality

In addition to many low-level reusable utility functions, the Matlab code includes the following functionality:

- CSPE analysis: Generate a .MAT data file containing a list of the sinusoidal peaks present in each analysis frame of a given input wavefile.
- Peak synthesis: Synthesize the sinusoidal peaks in a given .MAT data file and write a wavefile containing the result.
- Residual computation: Given the original wavefile used in the CSPE analysis and the resynthesized sinusoidal peak wavefile, compute the residual waveform containing noise and transients and write this to a wavefile
- Identify peak tracks: Given a .MAT data file containing sinusoidal peaks, generate a .MAT data file containing lists of the sinusoidal tracks which begin in each frame

- Track synthesis: Given a .MAT data file containing sinusoidal tracks, synthesize the peak tracks with no time-scale modification and write the resulting waveform to a wavefile
- Time-scale modification: Given a .MAT data file containing sinusoidal tracks, synthesize the peak tracks with a given constant time-scale modification factor and write the resulting waveform to a wavefile
- Redo analysis: Given a .MAT data file containing sinusoidal tracks, and the original input wavefile used in the CSPE analysis, perform the improved analysis process described in section 4.1
- Plot peak tracks: Given a .MAT data file containing sinusoidal tracks, generate a plot which displays the evolution of track frequency over time

In addition to the binary .MAT data files, human-readable .TXT file versions of all data files can be written at each stage to assist debugging and comparisons between different approaches.

Parameters

The parameters used in all of the Matlab code are highly customizable and can be coordinated through use of an XML parameter file which contains all of the parameters which will be used for a given round of analysis/synthesis. Customizable parameters include:

- Analysis frame size
- Number of sinusoidal peaks to be extracted per frame
- Minimum frequency of a sinusoidal peak
- Continuity threshold and range parameters for peak tracking, including frequency, amplitude, phase, and, when using the Unified Domain, spatial angle continuity

References

[1] Marina Bosi and Richard E. Goldberg. *Introduction to Digital Audio Coding and Standards*. Kluwer Academic Publishers, 2003.

[2] F. Charpentier and M. Stella. Diphone synthesis using an overlap-add technique for speech waveforms concatenation. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '86.*, volume 11, pages 2015–2018, Apr 1986.

[3] P. Depalle, G. Garcia, and X. Rodet. Tracking of partials for additive sound synthesis using hidden markov models. In *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*, volume 1, pages 225–228 vol.1, Apr 1993.

[4] David Dorran, Robert Lawlor, and Eugene Coyle. A hybrid time-frequency domain approach to audio time-scale modification. *Journal of the Audio Engineering Society*, 54(1/2):21–31, January/February 2006.

[5] Shlomo Dubnov. YASAS -yet another sound analysis-synthesis method. In *Proceedings of International Computer Music Conference*, 2006.

[6] Shlomo Dubnov and Xavier Rodet. Investigation of phase coupling phenomena in sustained portion of musical instruments sound. *The Journal of the Acoustical Society of America*, 113(1):348–359, 2003.

[7] P. Dutilleux, G. De Poli, and U. Zolzer. Time-segment processing. In Udo Zolzer, editor, *DAFX: Digital Audio Effects*, pages 205–214. John Wiley & Sons, Ltd., 2002.

[8] Riccardo Di Federico. Waveform preserving time stretching and pitch shifting for sinusoidal models of sound. In *Proceedings of the COST-G6 Digital Audio Effects Workshop*, 1998.

[9] Kelly Fitz and Lippold Haken. Bandwidth enhanced sinusoidal modeling in lemur. In *International Computer Music Conference*, September 1995.

[10] J. Laroche and M. Dolson. Improved phase vocoder time-scale modification of audio. *IEEE Transactions on Speech and Audio Processing*, 7(3):323–332, May 1999.

[11] Scott N. Levine. *Audio Representation for Data Compression and Compressed Domain Processing*. PhD thesis, Stanford University, December 1998.

[12] R. McAulay and T. Quatieri. Speech analysis/synthesis based on a sinusoidal representation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(4):744–754, Aug 1986.

[13] S. Roucos and A. Wilgus. High quality time-scale modification for speech. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '85.*, volume 10, pages 493–496, Apr 1985.

[14] Xavier Serra. *A System for Sound Analysis/Transformation/Synthesis Based on a Deterministic Plus Stochastic Decomposition*. PhD thesis, Stanford University, 1989.

[15] Kevin M. Short and Ricardo A. Garcia. Accurate low-frequency magnitude and phase estimation in the presence of DC and near-DC aliasing. In *Proceedings of the 121st Convention of the Audio Engineering Society*, 2006.

[16] Kevin M. Short and Ricardo A. Garcia. Signal analysis using the complex spectral phase evolution (CSPE) method. In *Proceedings of the 120th Convention of the Audio Engineering Society*, 2006.

[17] Kevin M. Short, Ricardo A. Garcia, and Michelle L. Daniels. Multichannel audio processing using a unified domain representation. *Journal of the Audio Engineering Society*, 55(3):156–165, March 2007.

[18] J. Smith and X. Serra. PARSHL: An analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation. In *Proceedings of the International Computer Music Conference*, 1987.

[19] Tony S. Verma, Scott N. Levine, and Teresa H. Y. Meng. Transient modeling synthesis: a flexible analysis/synthesis tool for transient signals. In *Proc. International Computer Music Conference*, pages 164–167, Sept. 1997.