University of Miami

An Adaptive Time-Frequency Distribution with Applications for Audio Signal Separation

By

Matthew Van Dyke Kotvis

A Research Project

Submitted to the faculty of the University of Miami in partial fulfillment of the requirements for the degree of Master of Science in Music Engineering Technology.

Coral Gables, Florida
April 25, 1997

Kotvis, Matthew Van Dyke (Master of Science, Music Engineering Technology)

An Adaptive Time-Frequency Distribution with Applications for Audio Signal Separation

Abstract of a Masters Research Project at the University of Miami.

Research project supervised by Kenneth C. Pohlmann

Abstract: An adaptive time-frequency distribution is developed to represent an audio signal in a more useful manner. The intended use of the adaptive time-frequency distribution is to aid digital signal processing systems in identifying specific components present in a signal. This distribution has applications to many facets of digital audio signal processing, including lossy audio compression and signal separation. As an example, the specific application of removing sound sources from two channel audio recordings is investigated.

Table of Contents

List of Figures

Chapter 1 - Introduction

One of the most widely used tools in digital signal processing is the Fourier transform, implemented as the Fast Fourier Transform, or FFT. It is useful for many reasons, the most important being the decomposition of any signal into its frequency components, providing magnitude and phase information at each frequency. In addition, an inverse transformation can convert back from the frequency domain into the time domain. The FFT can be computed relatively quickly, at or around real-time on today's digital signal processing hardware.

The FFT does have its disadvantages, however. The frequencies used to decompose a signal are a function of the sampling frequency of the signal and the number of frequency bins desired. Without modifying these two parameters, these frequencies are not selectable. A simple sine wave whose frequency does not fall on one of the frequencies of the transform will produce a spectrum with energy spread to many frequencies. Figure 1 illustrates this point. Figure 1a shows the 1024-point FFT of a 500Hz sinusoid sampled at a rate of 8000Hz. The 500Hz sinusoid falls exactly on one of the frequency bins of the FFT. As a result, significant energy is only present in the one frequency bin matching the frequency of the sinusoid. All other frequency bins contain energy which is more than 300dB below the peak frequency bin, which is essentially zero. Figure 1b shows the 1024-point FFT of a 503.9Hz sinusoid sampled at 8000Hz. Although there is only one component present, it does not appear as such in the computed spectrum. The frequency of 503.9Hz does not fall exactly on any of the bins of the FFT, therefore the transform must represent the single sinusoidal component using all available frequencies. Since there is

significant energy present in every bin, each bin must be used in the inverse transform in order to reconstruct the time-domain signal.

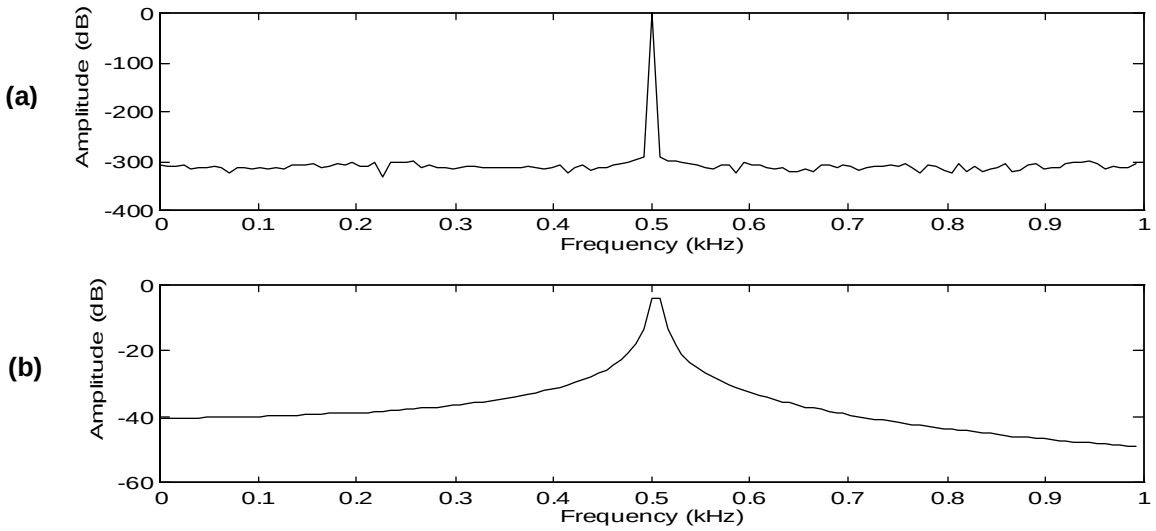

Figure 1 – FFT of sine waves of frequencies (a) 500Hz, (b) 503.9Hz

If changes occur to the signal within the block of time, the FFT will not represent these changes intuitively. As an example, Figures 2a and 2b show sinusoids with frequencies of 2000Hz and 500Hz respectively, sampled at 8000Hz. Both sinusoids fall exactly on FFT frequency bins when using a 1024-point transform. Figure 2c shows the FFT for the sum of the two sinusoids. In contrast, both Figures 3a and 3b are identical to Figures 2a and 2b respectively with the exception of the starting and ending times of each. The 2000Hz sinusoid is silenced during the last quarter of the signal while the 500Hz sinusoid is silenced during the first quarter of the signal. When the FFT is computed for the sum of the sinusoids the spectrum in Figure 3c is produced. Only two sinusoids are present, but energy is spread to nearly every frequency bin.

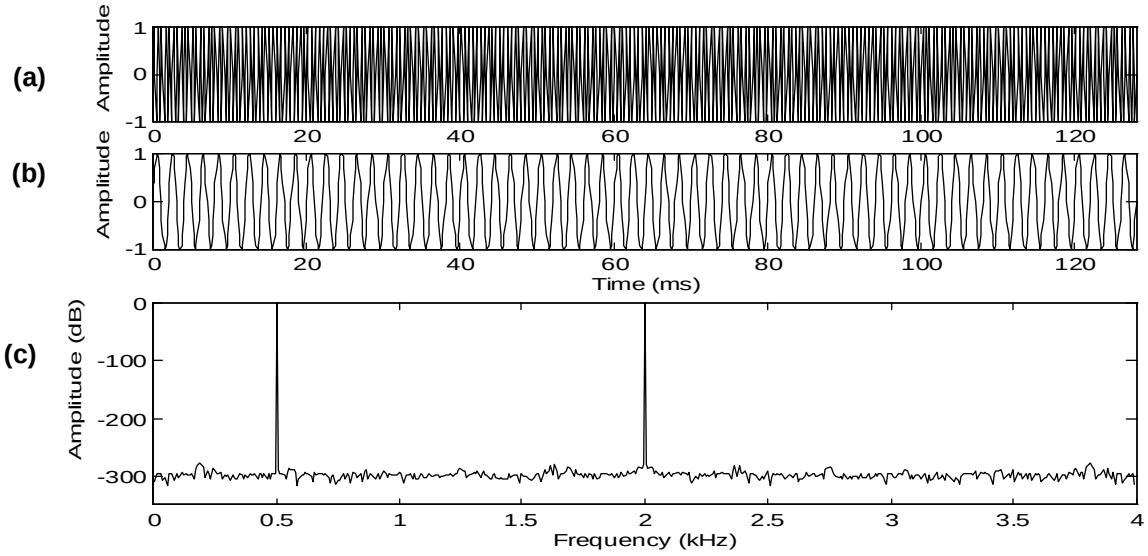

Figure 2 – Sine waves of frequencies (a) 2000Hz, (b) 500Hz, and (c) FFT of their sum

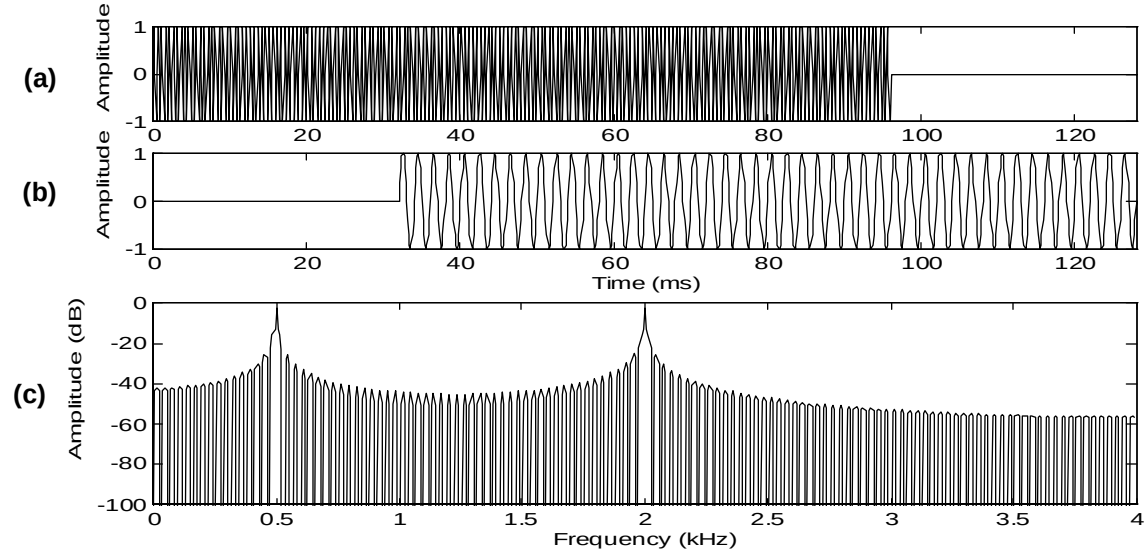

Figure 3 – Sine waves of frequencies (a) 2000Hz, (b) 500Hz with different start and end times, and (c) the FFT of their sum

Although the FFT is a powerful transform, it is not well suited to some signals, such as audio signals, which change rapidly with time relative to block size. A transform is needed which will accurately identify the components present in a signal. In this research an adaptive time-frequency distribution is developed which adapts to the signal being analyzed. Since the intended use is for audio applications, the distribution breaks the signal into octaves in order to distribute

the time-frequency information in a manner more suitable to audio. The reason for this is illustrated in Figure 4. Figure 4a shows an audio signal in the time domain, while Figure 4b shows the spectrum of that signal. Although there are many details of the signal which the FFT does not reveal, the spectrum does show that there is virtually no spectral content in the upper 75% of the frequencies computed. A logarithmic spacing of frequencies would essentially gather more information on lower frequency content at the expense of higher frequency content. Another reason for using octaves to process audio is the fact that the fundamental frequencies for the notes of a musical scale are spaced logarithmically in frequency. It should be noted that the signal in Figure 4 is by no means representative of all audio signals, but only an indicator of some of their properties.

One of many applications of the adaptive time-frequency distribution is the blind separation of sound sources on two-channel recordings, which is discussed in detail in later chapters. The concept behind using the adaptive transform to separate audio signals has already been addressed in this chapter. Both Figures 1b and 3c illustrate the property of the FFT which spreads energy from one or more sinusoidal components to many frequencies. Using an adaptive transform allows specific sinusoidal components to be identified, and plots similar to Figures 1a and 2c can be obtained. The process of separating individual components of signals becomes much easier.

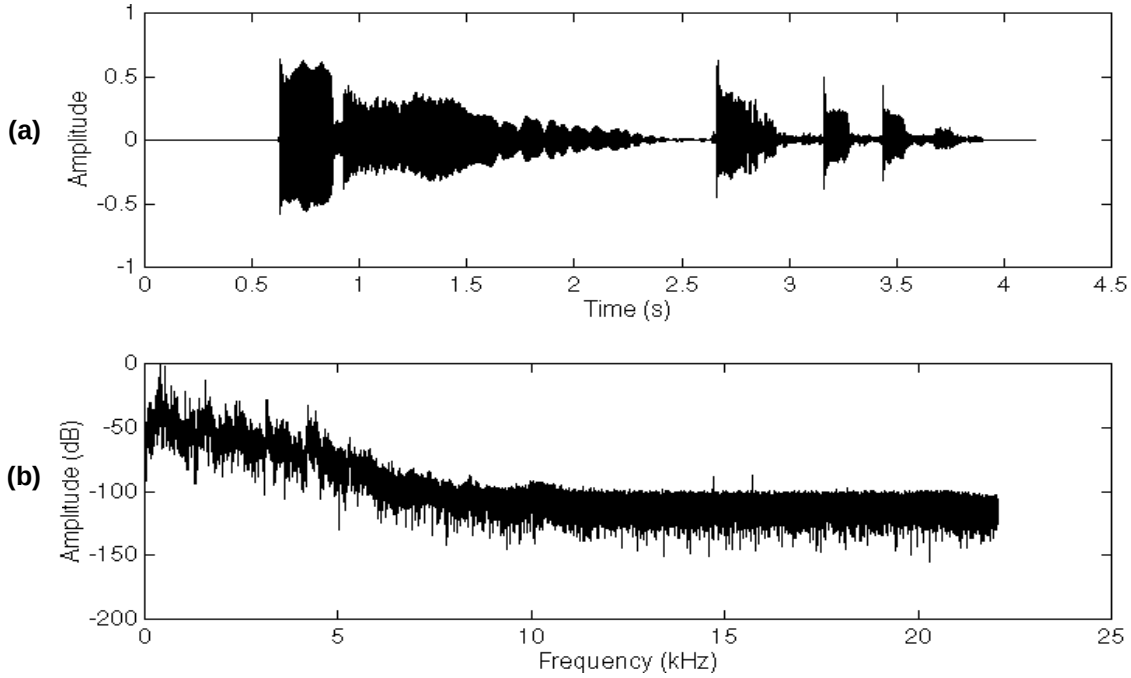

Figure 4 – (a) Plot of an audio signal in time and (b) its Fourier transform

The paper is essentially divided into two parts. The first part includes Chapters Two through Five and focuses on the adaptive time-frequency distribution. The second part starting with Chapter Six deals with the problem of separating signals out of two-channel recordings. Specifically, Chapter Two outlines the properties of some current time-frequency distributions. Chapter Three details the specifics of an adaptive method of signal decomposition using non-orthogonal sinusoids of any frequency. Chapter Four combines elements of both of the previous chapters and presents the adaptive time-frequency distribution and its properties. Chapter Five gives some examples of adaptive time-frequency distributions applied to audio signals. Chapter Six investigates the advantages and disadvantages of blind source separation from two-channel audio recordings. Chapter Seven highlights current related research in the field of blind signal separation. Chapter Eight gives examples of blind signal separation, lossy audio compression, and time scaling using the adaptive time-frequency distribution. Chapter Nine points out the

need for further research and places for improvement in the current algorithm for computing the adaptive time-frequency distribution. Chapter Ten concludes the paper.

Chapter 2 - Time-Frequency Analysis

In this chapter, current methods for generating time-frequency distributions will be discussed. This material is presented in order to emphasize the fact that time-frequency distributions do exist, but the distributions have properties which make them difficult to use in some applications. Understanding these properties will shed light on the value of the adaptive time-frequency distribution presented in Chapter 4. The focus of the discussion will not be the theory behind time-frequency analysis but the method of computation and the pros and cons of each distribution.

2.1 Introduction

The need for time-frequency analysis of audio signals is straightforward; audio signals typically change with respect to both time and frequency. A simple example is the chirp signal, a sinusoid whose frequency increases linearly with time. If a Fourier transform was performed on the entire duration of a chirp signal, a near-flat spectrum would be produced. Figure 5 shows both the chirp signal in time and its Fourier transform. This is the proper behavior for the Fourier transform; its purpose is to reveal what frequency content was present during the time analyzed. Since the chirp signal spends an equal amount of time at each frequency, the Fourier transform produces a spectrum with equal energy at all frequencies. The Fourier transform of an impulse signal will also produce a flat spectrum.

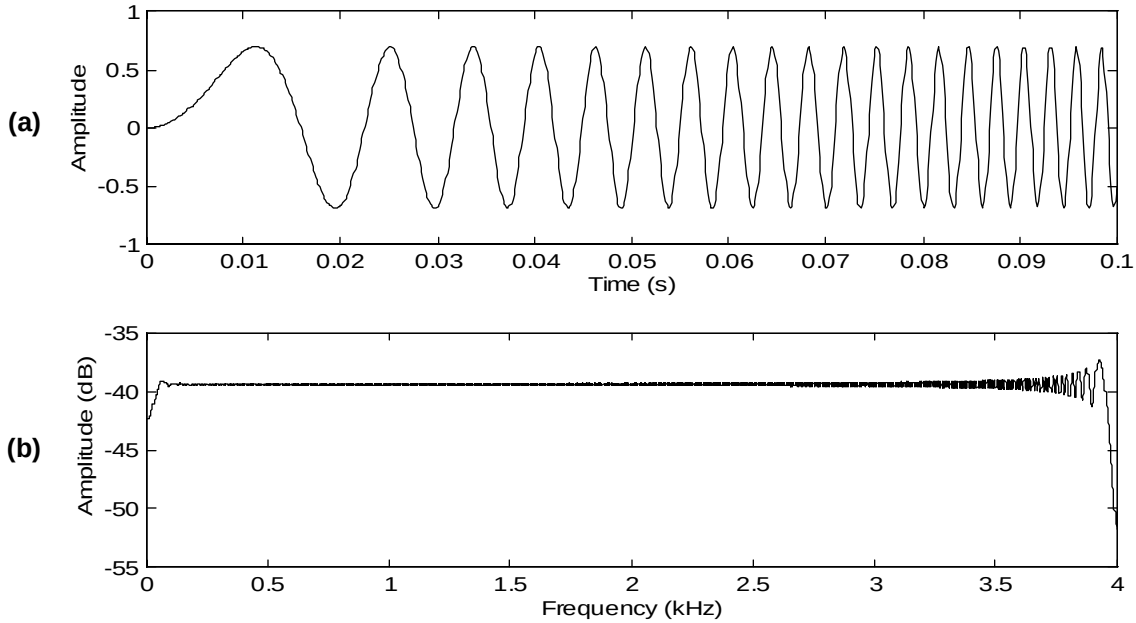

Figure 5 – (a) Portion of chirp signal and (b) its Fourier transform

Many signals can create similar spectra, so frequency-domain analysis alone is not adequate. A more desirable form of signal analysis is one that shows how a signal changes with time and frequency simultaneously. Time-frequency distributions compute the amount of energy present for points on a time-frequency plane, which can then be displayed with a three-dimensional plot. The ideal time-frequency distribution for a chirp signal is shown in Figure 6. For any given point in time, the signal has exactly one frequency component. The distribution correctly indicates that the signal increases in frequency as time increases.

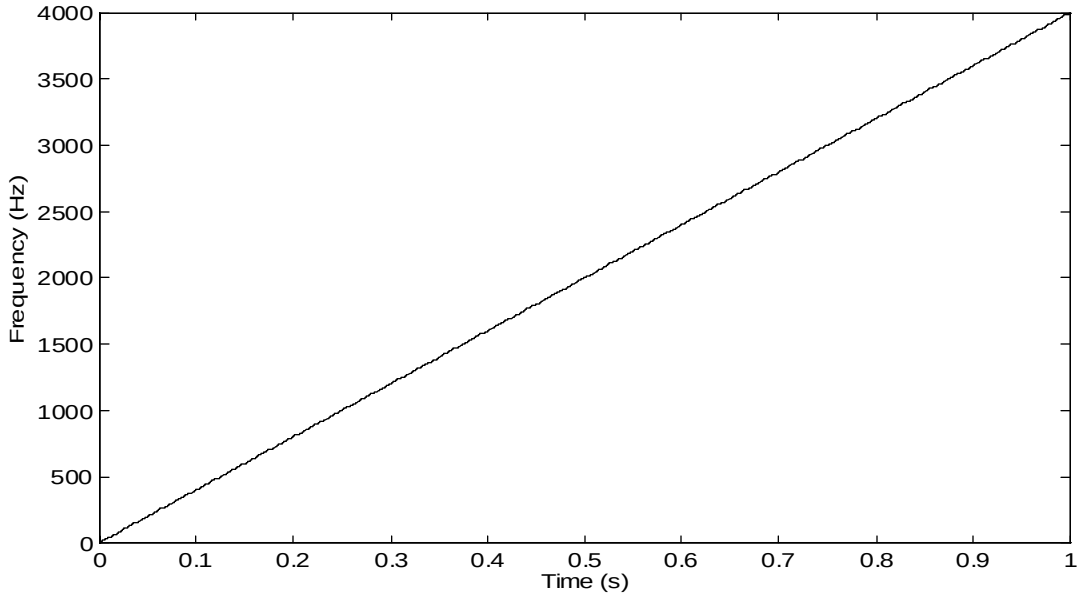

Figure 6 – Contour plot for the ideal time-frequency distribution of a chirp signal

Note that Figure 6 depicts the ideal time-frequency distribution of the chirp signal. It is not possible to obtain the ideal time-frequency distribution for any signal due to the uncertainty principle, which essentially states that any attempt to improve the time resolution of a system will degrade the frequency resolution. As stated by Skolnik, "both the time waveform and frequency spectrum cannot be made arbitrarily small simultaneously."[1] The product of these two resolutions, the time-bandwidth product, remains constant for any system. Figure 6 cannot be produced by any time-frequency distribution, but there are many different time-frequency distributions which approach the ideal case. The following sections provide specifics on a number of different time-frequency distributions.

2.2 Spectrogram

The spectrogram is the most basic of all time-frequency distributions. It is computed by performing a Fourier transform on short time segments of a signal. The spectrogram shown in

Figure 7 is computed by breaking up the chirp signal into smaller equal-sized sections and analyzing each section with the Fourier transform.

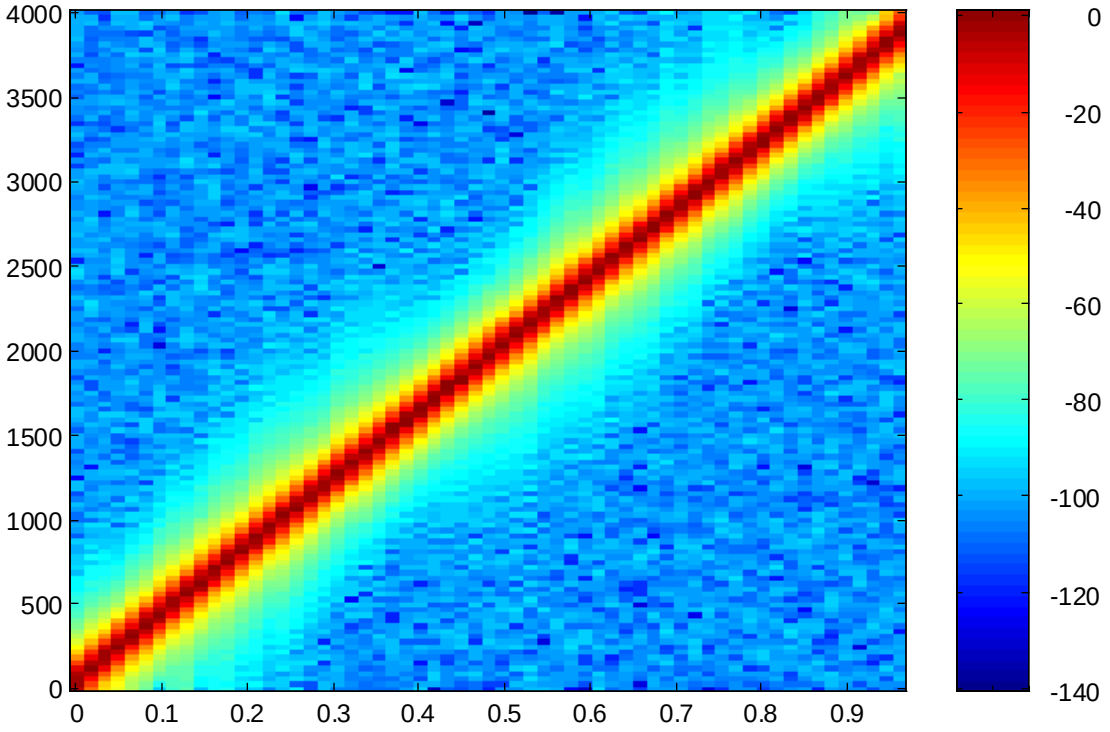

Figure 7 – Spectrogram of a chirp signal (magnitude is grayscale)

Figure 7 reveals that the spectrogram indicates the presence of a signal with frequency increasing with time, yet the bandwidth of the signal at any given time does not match that of the ideal case shown in Figure 6. This result is related to a drawback of the discrete Fourier transform. Since the discrete Fourier transform treats each signal as a periodic signal any discontinuity between the first and last samples of a signal will cause the transform to include spectral content representing the discontinuity. The use of time-domain windowing reduces the effects of discontinuities. However, for each block of time the spectrogram computes the Fourier transform of a signal multiplied by a window function, not just a signal. The resulting transform is equivalent to the spectrum of the signal convolved with the spectrum of the window.

Therefore the increased frequency spread of the spectrogram is a result of windowing of the signal.

In addition to windowing, another common method for improving the response of the spectrogram is overlapping. When using overlap, a signal is not broken up into separate time blocks but overlapping blocks. Typical overlap for a spectrogram (including the one in Figure 7) is fifty percent, meaning that a time block used in the spectrogram is comprised of all the samples from the last half of the previous block and the first half of the next block. Overlapping is used to counteract the effects of time-domain windowing, which attenuates the spectral content of the signal near the ends of a time block.

2.3 Wigner Distribution

The Wigner distribution is fundamentally quite different from the spectrogram [2]. The most notable characteristic of the Wigner distribution is that it is highly non-local, which means signal energy for all points in time is used to compute the frequency content for the current time. When computing the Wigner distribution at a specific point in time, the time samples for future times are multiplied by the time samples for past times. In essence, when evaluating a signal s at time t, the signal is folded around time t and each overlapping pair of samples $s(t+n)$ and $s(t-n)$ is multiplied together. This folded signal is then used by the Fourier transform to determine the frequency content of the signal at time t.

Similar to the spectrogram, overlapping can be used to improve the response of the Wigner distribution. In fact the Wigner distribution can be computed for every point in time. This results in smoother transitions between time-frequency blocks but has the disadvantage of creating a

large number of time-frequency points. A signal of length N_t analyzed with N_f frequencies creates a time frequency distribution of $N_t \cdot N_f$ points.

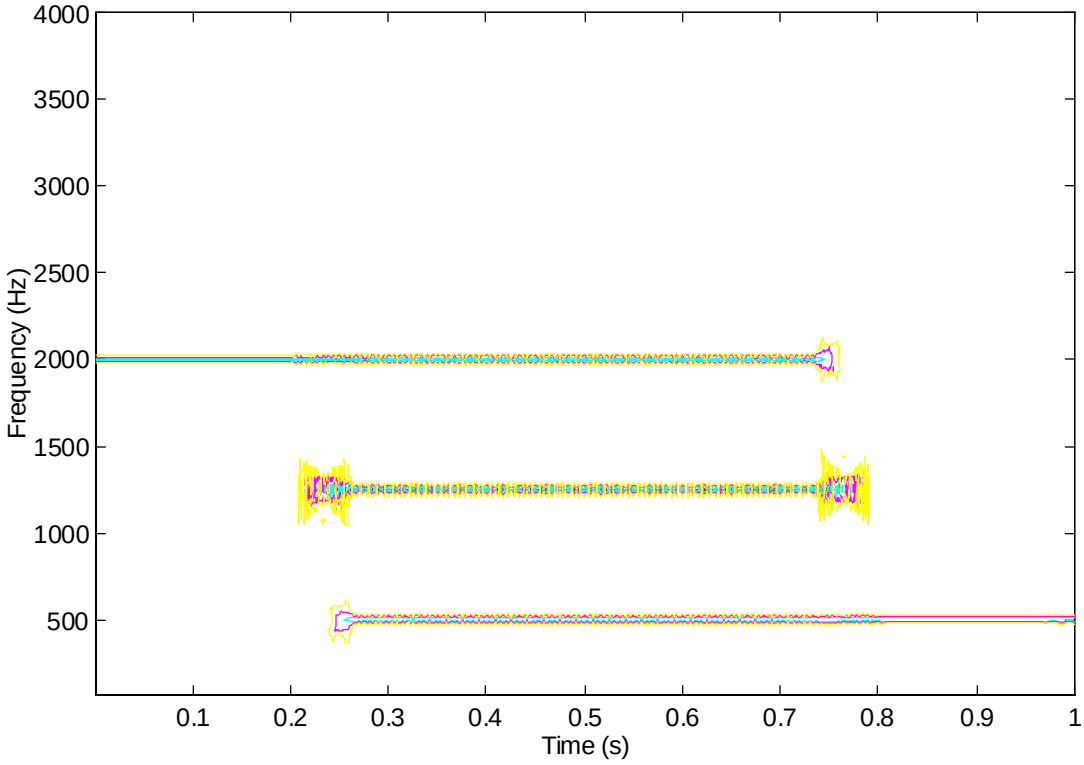

Figure 8 – Wigner distribution of the sum of two sinusoids – a 2000Hz sinusoid present from 0.0 seconds to 0.75 seconds and a 500Hz sinusoid present from 0.25 seconds to 1.0 second

The Wigner distribution has some notable disadvantages. First, the non-local nature of the Wigner distribution can produce results which are not intuitive or desirable. If a signal consists of two short sinusoids with nothing between them, the time point halfway between the sinusoids will produce an output. Also, when two sinusoids of different frequencies are present at a given time side effects known as cross products occur, which produce energy at a frequency halfway between the two sinusoids. Figure 8 illustrates how cross products of the Wigner distribution of a signal can produce confusing results. The Wigner distribution was computed using an algorithm by R. van der Heiden [3]. The distribution shows both sinusoids from Figure 3 whose

frequencies are constant with respect to time and with correct starting and ending times. However, a cross product is produced which gives the impression of a third frequency component. In fact the energy level of the cross product is greater than the energy levels in both sinusoidal components. The non-local nature of the Wigner distribution and the cross products make the distribution difficult to work with.

2.4 Choi-Williams Distribution

The Choi-Williams distribution is virtually identical to the Wigner distribution but attempts to reduce the effect of disadvantages in the Wigner distribution [2]. The Choi-Williams distribution differs in that an exponential window is applied to the folded time signal to create a more localized distribution. This window eliminates the first disadvantage of the Wigner distribution noted above. In addition, the width of the window in the time domain can be altered to produce the best results for a particular signal. The Choi-Williams distribution still produces undesirable cross products, but they are much lower in amplitude than the cross products of the Wigner distribution.

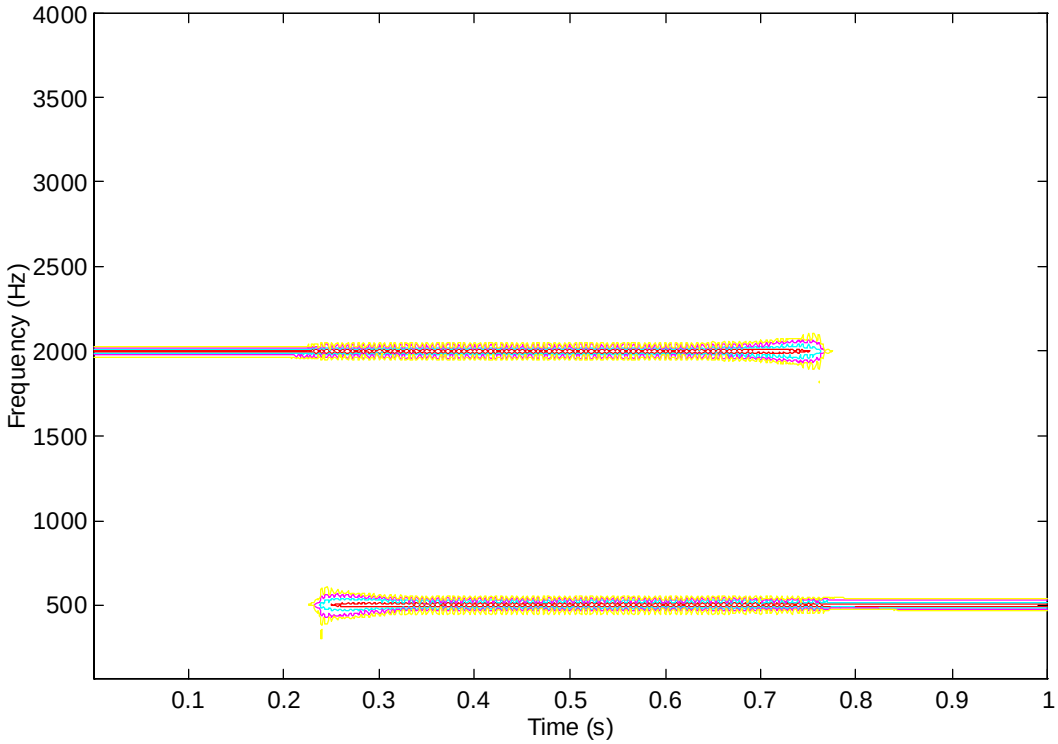

Figure 9 – Choi-Williams Distribution of the sum of two sinusoids – a 2000Hz sinusoid present from 0.0 seconds to 0.75 seconds and a 500Hz sinusoid present from 0.25 seconds to 1.0 second

As noted for the Wigner distribution, the number of time-frequency points in each distribution is greater than the number of time points in the original signal. In the case of the spectrogram in Figure 7, the total data size of the distribution was equivalent to that of the original signal. The Fourier transform of a real signal produces a symmetrical spectrum where only half of the data points contain unique information. Since the overlapping of time-frequency blocks doubles the number of time points analyzed, and only half of the frequency points produced by the Fourier transform are used, the total number of points after transformation remains constant. In the case of the Wigner distribution (Figure 8) and the Choi-Williams distribution (Figure 9), the original signal contained 4000 samples and was analyzed for 64 frequencies per time-frequency block.

Since both distributions create a time frequency block for each original time sample the data size for these distributions was 64 times the size of the original signal.

An additional disadvantage to both the Wigner and Choi-Williams distributions is the computation time of each. When generating the time-frequency plots for this paper, the total computation time was measured and the results are listed in Table 1. The times listed for each distribution are for computing the distribution of a second long audio signal sampled at 8000Hz using 16-bit quantization. Assuming there is a direct relationship between the number of samples analyzed and the total computation time, the time required to process a second long audio signal sampled at 44.1kHz would be about 5.5 times greater than the values listed in Table 1. Also, these values were obtained by calculating each distribution on a Pentium-class computer with a processor running at 166MHz. Therefore at the present time it is not feasible to consider either the Wigner or Choi-Williams distributions for real-time processing of audio signals. This characteristics along with the data size of the distribution are the most prohibitive factors preventing the Choi-Williams distribution and other similar time-frequency distributions from becoming widely used in audio processing.

Table 1 – Computation time for various time-frequency distributions

Distribution	Computation Time (seconds)	Ratio of computation time vs. computation time of spectrogram
Spectrogram	0.28	1:1
Wigner	39.16	140:1
Choi-Williams	74.98	268:1

2.5 Wavelets

Since the Fourier transform is used in all of the previous time-frequency distributions, the signals being analyzed are necessarily decomposed using an orthogonal basis of sinusoids. Wavelet

analysis differs from all previously mentioned distributions since signals other than sinusoids are used to model a signal [4]. The basis signal, or wavelet, used to decompose an audio signal does not produce information about "frequency" in the traditional sense, but rather a distribution of time and scale is created. A change in scale represents stretching or compressing the wavelet by a factor of two. It is therefore possible to reconstruct any signal using one wavelet as the basis and placing as many wavelets as are needed at different times with different amplitudes and scales.

There are several advantages to wavelets over other time-frequency distributions. First, the computation time is on the order of an FFT. This is a desirable property, especially for real-time systems. Second, wavelets provide a good means for data compression. The output of a wavelet decomposition is essentially a list of coefficients containing the time, scale, and amplitude of each wavelet. A simple form of compression is to discard all wavelets with small amplitudes. The larger amplitude wavelets model the majority of the features of a signal, so the removal of small amplitude wavelets has little effect on the signal but reduces the number of coefficients needed to reconstruct the signal. Third, the wavelets are localized in time. All of the other time-frequency distributions decompose the signal into sinusoids which last for all time, even though each time block being analyzed may be relatively short. A wavelet has a finite length of time which is better suited to short, localized signals.

The one disadvantage to using wavelets on audio signals is that audio contains significant sinusoidal content. Pitches generated by most musical instruments are periodic and are thus the sum of sinusoids whose frequency and amplitude continuously change with time. Since a

wavelet does not precisely report information on the frequency of a signal it may not be the best tool for analyzing audio signals.

When analyzing wideband signals such as audio signals wavelets make use of filter banks, which typically separate the audio signal into octaves before wavelets are applied. Working with octaves instead of linearly-spaced sinusoid frequencies is a good match for audio signals, and will play an important role in the adaptive time-frequency distribution presented in Chapter 4. The next section discusses filter banks in greater detail.

2.6 Filter Banks

Filter banks are used to break up a wideband signal into multiple bands [4]. As an example, MPEG audio compression uses filter banks to break up a signal into bands comparable to the critical bands of the human auditory system. As noted in the previous section wavelets typically use filter banks to break up a signal into octaves. Figure 10 illustrates a perfect-reconstruction filter bank which both decomposes the signal into octaves and reconstructs the signal from the decomposition. Starting on the left side of Figure 10, $x(n)$ represents the signal to be decomposed into octaves. $\mathbf{H_0}$ and $\mathbf{H_1}$ are complimentary lowpass and highpass filters respectively, each with a corner frequency of one-fourth the sampling rate of the signal. The output of the highpass filter is the frequency content contained in the highest octave of the signal, while the output of the lowpass filter is all the remaining frequency content.

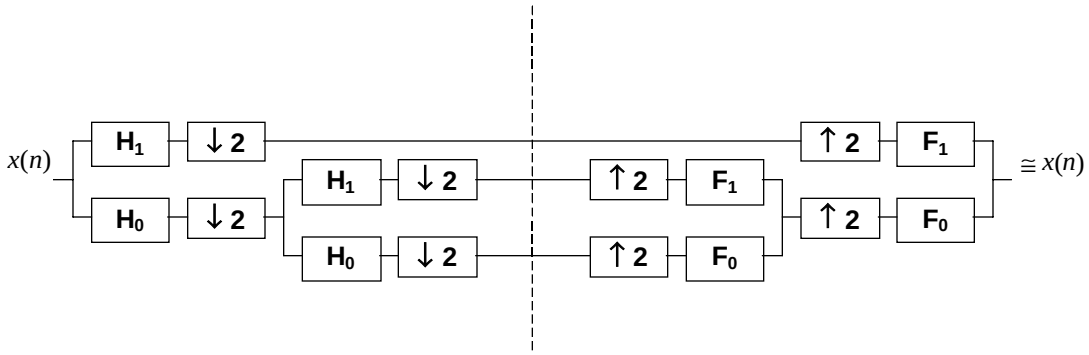

Figure 10 – Perfect Reconstruction filter bank with analysis and synthesis filters

The block following each of the filters is a downsampling operator. Without downsampling, the result of the filtering operation would be two signals with half the frequency content of the original while still maintaining the length of the original signal. Since the total number of samples has doubled while the amount of information has remained unchanged, downsampling can be applied to remove samples without removing information. Downsampling by a factor of two removes every other sample from a signal and effectively divides the sampling rate for the signal in half.

For the lowpass-filtered signal downsampling has no adverse effect. The minimum sampling rate for any band-limited signal is at least twice the maximum frequency of the signal. Since the lowpass filter forces the maximum frequency of the signal to be divided in half, reducing the sampling rate by a factor of two will not remove any signal content. In the case of the highpass-filtered signal the total bandwidth of the signal content is identical to that of the lowpass-filtered signal, but the maximum frequency is identical to that of the original signal. Downsampling will cause all frequency content above the new maximum frequency to be mirrored below the maximum frequency; this is known as aliasing. For example, if a chirp signal rising from 2kHz to 4kHz, originally sampled at 8kHz, is downsampled, the result would be a chirp signal with

frequency falling from 2kHz to DC. Note that information has not been lost – just inverted in frequency.

After filtering and downsampling the total number of samples is identical to that of the original signal while the frequency content of the signal is divided into two bands. Also, the highpass-filtered signal contains one octave of information. The complimentary highpass and lowpass filters, known as the filter bank, can be reapplied without alteration to the lowpass-filtered signal since both the frequency content and the sampling rate have been reduced by a factor of two. The filtering and downsampling operations can continue to be applied until the desired number of octaves has been obtained. Figure 10 shows a system where a two-channel filter bank has been applied twice to produce three bands of information, two of which are exact octaves. When applying filter banks to an audio system which ranges from 20Hz to 20kHz, ten octaves of information need to be separated. Therefore at least nine applications of a two-channel filter bank are needed to separate the signal into octaves.

In addition to simply dividing a signal into octaves, the proper choice of filters can lead to a condition known as perfect reconstruction. This means that the filter bank has decomposed the signal in a manner which can be reversed to recreate the original signal. The filters used to decompose the signal are known as analysis filters, and the filters used to reconstruct the signal are known as synthesis filters. The reconstruction process is essentially a mirror image to the decomposition process, where the octaves are each upsampled before being filtered by the synthesis filters. Figure 10 includes both analysis and synthesis filters and illustrates how the decomposed signal is reconstructed.

The perfect reconstruction filter bank is very useful for systems performing such tasks as audio compression or signal separation. The vertical dotted line in Figure 10 illustrates where signal processing functions for audio compression or separation would be placed. For compression, certain elements of the signal which are deemed inaudible can be removed to decrease the amount of data needed to represent the signal. The reconstruction operation simply takes the components which remain and restores the signal. Ideally the resulting signal should sound nearly identical to the original signal. The ability to identify and remove components of a signal is not inherently built into the spectrogram, Wigner, or Choi-Williams distributions since they are all dependent on the Fourier transform for signal decomposition. A perfect reconstruction filter bank is used in the adaptive time-frequency distribution presented in Chapter 4.

As shown in Figure 10, the synthesis filter bank reconstructs the original signal from the outputs of the analysis filter bank. The upsampling operator doubles the length of the signal by inserting a zero after each sample. The filters $\mathbf{F_0}$ and $\mathbf{F_1}$ are lowpass and highpass filters respectively. Both $\mathbf{F_0}$ and $\mathbf{F_1}$ are related to $\mathbf{H_0}$ and $\mathbf{H_1}$ and the relationship used for this research is discussed in Section 4.1.2. The signal filtered by $\mathbf{F_1}$ is inverted in frequency to return the signal content to its original frequency location. The signal filtered by $\mathbf{F_0}$ is not affected in frequency. The operation is repeated until all bands have been filtered. Any difference between the original signal and the reconstructed signal is a function of the stopband attenuation level in each of the analysis and synthesis filters.

Chapter 3 - Adaptive Non-Orthogonal Signal Decomposition

3.1 Introduction

The previous chapter discussed various time-frequency distributions based on either orthogonal sinusoids or wavelets. In the case of sinusoids there is virtually no adaptation possible. For wavelets the adaptive properties are in terms of scale, not frequency, and the scale of a wavelet can only be adapted by a factor of two. The transform presented in this chapter is based on sinusoids of any frequency and these frequencies can be changed based on signal content. The remainder of this chapter will discuss the computation, properties, and applications of this transform.

3.2 Computation

The theory behind the non-orthogonal signal decomposition is presented by Dologlou, Bakamidis, and Carayannis [5]. This paper teaches that any signal can be decomposed using virtually any group of non-orthogonal sinusoids.

The computation of this transform is relatively simple. First, a number of frequencies N between 0 and the half the sampling rate f_s are selected. Using the sampling rate of the signal to be analyzed these frequencies are all converted to numbers between 0 and π through the relation $\theta_n = \frac{2\pi \cdot f_n}{f_s}$, $n = 1, 2, \ldots, N$, where f_n is the frequency being converted, and θ_n is the new frequency. A matrix $\boldsymbol{A}$ is then created of size $2N \cdot 2N$ which contains the sine and cosine vectors for each of the frequencies θ_1 through θ_N. Equation 1 shows the construction of the matrix.

$$\mathbf{A} = \begin{bmatrix} \sin(n \cdot \theta_1) \\ \cos(n \cdot \theta_1) \\ \sin(n \cdot \theta_2) \\ \cos(n \cdot \theta_2) \\ \dots \end{bmatrix}^T \quad n = 0, 1, 2, \dots, 2N - 1 \qquad \text{(Eq. 1)}$$

The first two columns of the matrix contain the first 2*N* time samples for the sine and cosine of the first frequency θ_1. Each column thereafter alternates between the sine and cosine of each of the remaining *N*-1 frequencies. The last step is to simply take the inverse of the matrix $\boldsymbol{B} = \boldsymbol{A}^{-1}$. The resulting matrix $\boldsymbol{B}$ will decompose a signal *x* of length 2*N* into a spectrum of the frequencies chosen through the relation $y = \boldsymbol{B} \cdot x$.

The coefficients representing the spectrum actually correspond to the sine and cosine for each frequency. Equations 2 and 3 show the calculation of the magnitude and phase respectively for each pair of coefficients.

$$Magnitude(\theta_n) = \sqrt{y(2n-1)^2 + y(2n)^2} \qquad \text{(Eq. 2)}$$

$$Phase(\theta_n) = \tan^{-1}\left(\frac{y(2n-1)}{y(2n)}\right) \qquad \text{(Eq. 3)}$$

The non-orthogonal method of signal decomposition can be compared to the Fourier transform, but both methods are fundamentally different. First, the Fourier transform can process both real- and complex-valued input signals, while the non-orthogonal transform only works for real-valued input signals. Second, the 2*N* complex coefficients of the Fourier transform represent magnitude and phase for both positive and negative frequency, while the non-orthogonal transform only represents magnitude and phase for positive frequencies. Third, the Fourier transform forces one

frequency to be zero and one frequency to equal one-half the sampling frequency; the non-orthogonal transform cannot be computed if either of these frequencies is used.

3.3 Selection of Frequencies

As stated in the previous section the frequencies of zero and one-half the sampling frequency cannot be used in the non-orthogonal transform. This creates somewhat of a dilemma in selecting a group of frequencies which adequately covers the spectrum without using the two endpoints of the spectrum. Since there are no rules associated with the selection of frequencies, linear spacing in not a requirement. Both a logarithmic spacing or a random spacing of frequencies may work as well as a linear spacing.

In light of this new freedom the author attempted to make the most of this transform in terms of audio signals. Since the fundamental frequencies for the semitones of a musical scale are logarithmically spaced a transform matrix was attempted which contained every fundamental frequency in the audible spectrum of 20Hz to 20kHz, a total of 120 frequencies. This created a matrix of 240 elements square, which failed upon attempted inversion.

Since the frequencies were logarithmically spaced there were as many frequencies between 20Hz and 40Hz as there were between 10kHz and 20kHz. Also any frequency in the lowest octave would require between 1100 and 2200 samples at a sampling rate of 44.1kHz to make a complete cycle. With only 240 samples of each signal, there were 12 semitone frequencies which had between 10% and 20% of their cycle represented in the matrix. It is the author's belief that there was too much similarity between too many frequencies for the matrix to be invertible. The frequencies for the lower seven octaves had to be removed before the matrix became invertible,

which led to the use of filter banks in the adaptive time-frequency distribution presented in Chapter 4.

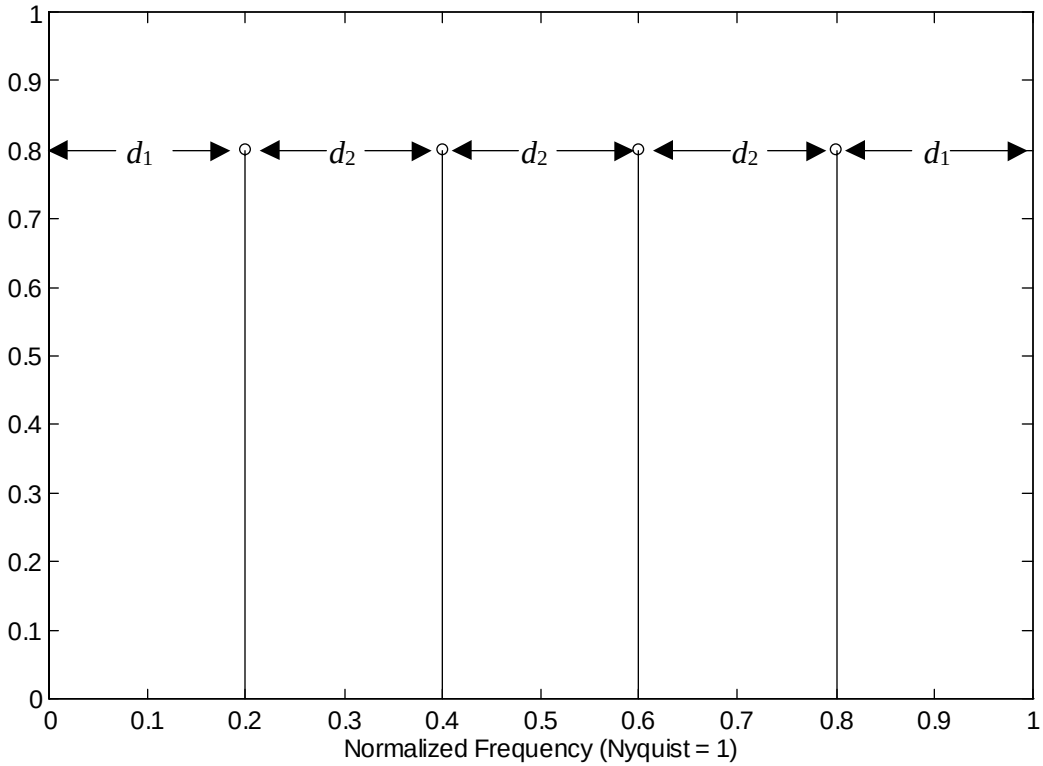

Figure 11 – Example of frequency spacing for non-adaptive signal transform

Since a random selection of frequencies does not seem a wise choice for a reliable signal decomposition method, the choice is left between logarithmic spacing and linear spacing. Figure 11 illustrates what parameters were used in choosing a linear spacing of frequencies for analysis. The parameter d_1 is the frequency difference between the lowest frequency bin and DC; it is also the frequency difference between the highest frequency bin and the Nyquist frequency. The parameter d_2 is the frequency difference between each pair of neighboring frequency bins. This method of frequency spacing is considered linear since every pair of neighboring bins has identical frequency spacing. Given the constraints above, a relationship between d_1 and d_2 exists and is shown in Equation 4,

$$d_1 = \frac{f_s - 2 \cdot (N-1) \cdot d_2}{4} \qquad \text{(Eq. 4)}$$

where N is the number of frequencies used for analysis and f_s is the sampling frequency. For the example in Figure 11, N is equal to 4 and d_1 and d_2 have identical values of $\frac{f_s}{10}$.

One test which proved useful in determining the better choice of frequency spacing was measuring the energy leakage caused by analyzing a sine wave whose frequency did not match that of the transform. Sine waves of many frequencies between 0 and half the sampling rate of 8kHz were analyzed with a sixteen-frequency transform and the results are shown in Figure 12.

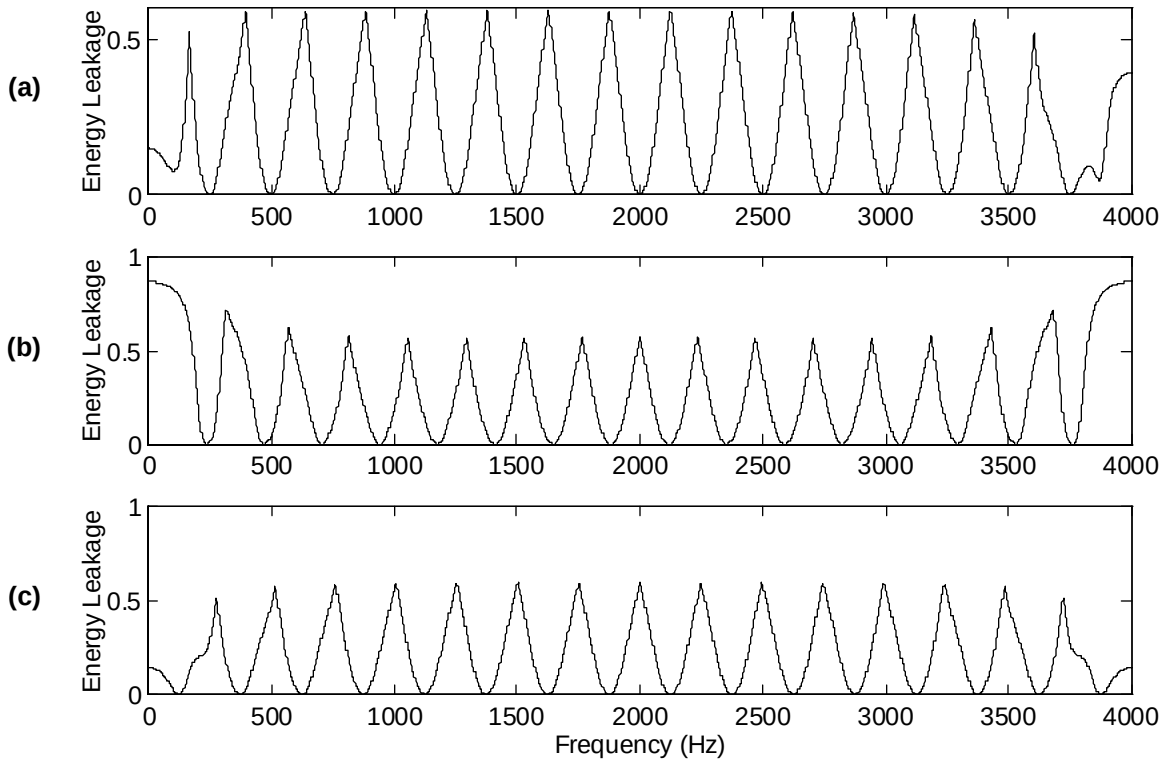

Figure 12 - Energy Leakage for entire frequency range using (a) Fourier transform, and using the non-orthogonal transform with (b) $d_2 = d_1$ (c) $d_2 = 2 \cdot d_1$

It is important to note that the energy leakage plotted in Figure 12 has been normalized with respect to the total energy calculated from the spectrum as well as the total energy calculated from the time signal. The motivation for this is discussed in Section 3.4, but it simply means the maximum energy leakage for a given signal is theoretically unity. A value of 0.5 means half of the total energy of the signal is distributed to frequency bins other than the peak frequency bin.

As a basis for comparison, Figure 12a shows the amount of energy leakage for all frequencies when using the 32-point Fourier transform for analysis. For most frequencies the Fourier transform behaves in a predictable manner. When a frequency associated with an existing bin is analyzed (such as 2000Hz), there is no energy leakage; one frequency bin fully describes the signal present. The further a sine wave frequency is from a bin, the greater the amount of energy leakage. In addition, there are irregularities near the lowest and highest frequencies. This is due to the fact that only sixteen frequencies are being analyzed and the total energy of the signal is perceived to be less by the Fourier transform. This problem is discussed in greater detail in Section 3.4.

For Figure 12b, the non-orthogonal decomposition method was used with a linear spacing of frequencies as illustrated in Figure 11. The parameters d_1 and d_2 were set equal to each other. Using Equation 4 and N equal to 16 frequencies, parameters d_1 and d_2 had values of 235.3 Hz. This selection produced equal spacing between adjacent frequency bins as well as between the extreme frequency bins and the frequency limits of DC and $\frac{f_s}{2}$. The energy leakage for this selection of frequencies is shown in Figure 12b. The amount of energy leakage between the extreme frequency bins and the frequency limits is larger than that of any other range. Under

these conditions, it would be difficult to identify a sinusoid if most of the energy is dispersed throughout the spectrum.

Since the properties for the spacing of frequencies chosen previously were undesirable, the calculations were repeated for various spacing of frequencies. The new spacing of frequencies would have to exhibit properties more closely matched to the Fourier transform to be useful. The parameter d_2 was varied and the behavior of the transform was calculated for frequencies based on the new values of d_1 and d_2. As d_2 increased, the energy leakage at the ends of the spectrum decreased significantly while the energy leakage for the rest of the spectrum increased slightly. The frequency set used to produce Figure 12c was obtained when $d_2 = 2 \cdot d_1$. This group of frequencies was selected because the amount of energy leakage between frequencies was most uniform across the spectrum, similar to that of the Fourier transform. In addition, this group of frequencies exhibits desirable properties which will be seen later in the chapter.

Other conditions could be used to determine the best spacing of frequencies. A logarithmic spacing of frequencies was tested in a similar manner to the linear spacing, but the results were much worse for signals of frequencies near the maximum or minimum frequency. Also, a different criteria for selecting the precise linear spacing of frequencies other than the most uniform energy leakage could be used. However, due to the adaptability of the frequencies to be discussed in Section 3.6, finding a perfect spacing of frequencies is not essential.

3.4 Limitations

One limitation of the non-orthogonal transform is that the total energy calculated for a signal changes depending on the frequencies used for signal decomposition. In theory, there is a fixed

amount of energy present in a signal. An ideal transform would accurately represent that energy. Any changes to the computation of the transform or the signal which did not affect the energy of the signal should have no effect on the energy in the transform. For the non-orthogonal transform, this is not the case. If two different groups of frequencies are used to analyze the same signal, the total energy as computed by the transform will not remain constant.

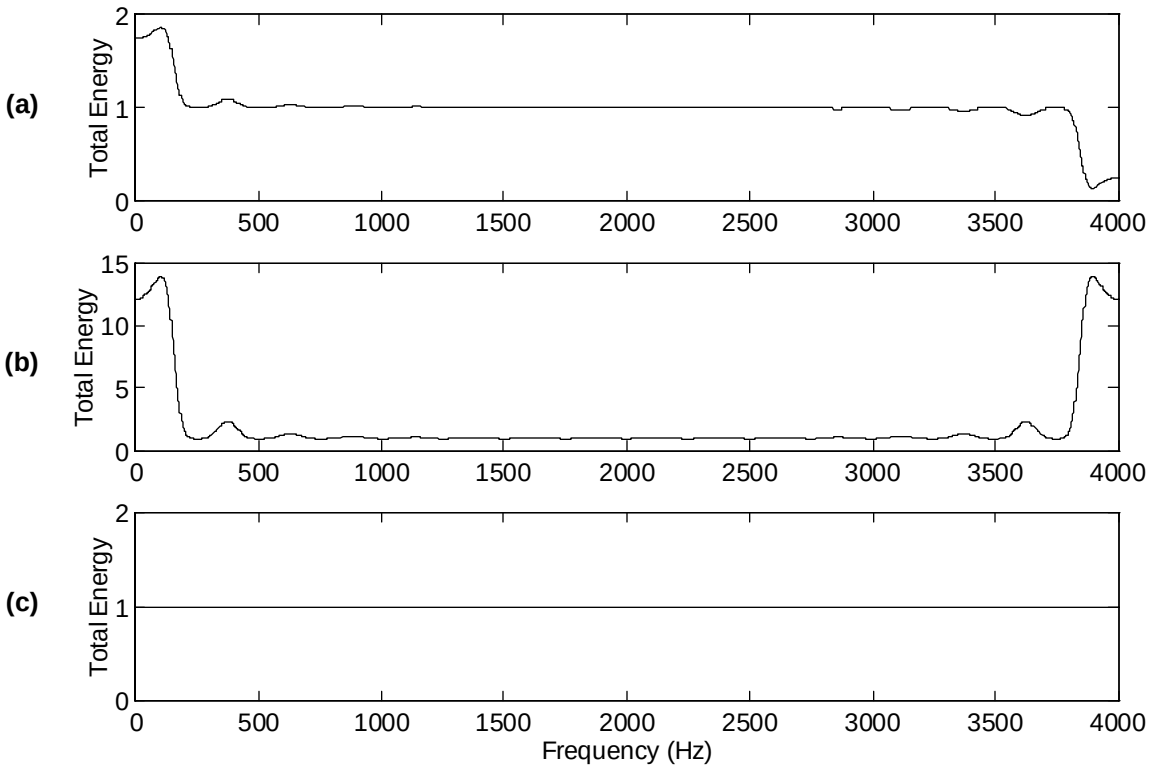

Figure 13 - Total Energy for entire frequency range using (a) Fourier transform, and using the non-orthogonal transform with (b) $d_2 = d_1$ (c) $d_2 = 2 \cdot d_1$

On the other hand, the non-orthogonal transform is capable of accurately representing a sinusoid whose frequency is distant from all the transform frequencies. For example, if a sine wave of frequency 100Hz is analyzed by the non-orthogonal transform using frequencies between 1000Hz and 10kHz, the transform will produce coefficients which will accurately reconstruct the signal upon application of the inverse transform. However, the energy contained in each of the frequency bins is quite large compared to the energy of the original signal.

Figure 13a shows the total energy computed for signals of many frequencies using the Fourier transform. Although the amplitude level of the time domain signals was constant for all frequencies, the total energy computed by summing the squared time samples was not exactly constant for all frequencies. The plot in Figure 13a actually displays the ratio of the total energy computed on the spectrum of a signal versus the energy computed on the time-domain signal. In theory, the ratio should always be one but that is not the case, even with the Fourier transform. For most frequencies, however, the ratio is either one or very close to one. Figures 13b and 13c show the total energy computed at all frequencies using the non-orthogonal transform with the same frequency spacing used for Figures 12b and 12c, respectively. Figure 13b reveals that the total energy for low and high frequencies rises well above one; this irregularity indicates why the total energy leakage in Figure 12b was unusually large for low and high frequencies. More notably, however, is the behavior of the group of frequencies used in Figure 13c. The ratio of the energy present in the spectrum to the energy present in the signal is one for all frequencies. This behavior is the most desirable of the three transforms and is not attainable by even the Fourier transform implemented as an FFT.

3.5 Performance Comparison with Fourier Transform

When comparing the spectrum produced by both the Fourier transform and the non-orthogonal transform, several differences become apparent. First, the non-orthogonal transform can produce significant energy leakage when analyzing a signal below the lowest frequency bin or above the highest frequency bin. This does not pose a problem for the Fourier transform since frequency bins exist at the absolute maximum and minimum frequencies. Second, while an impulse signal produces a flat spectrum for the Fourier transform, the non-orthogonal transform does not

produce a flat spectrum for most groups of frequencies. Figure 14a shows the spectrum of an impulse signal produced by the Fourier transform. Figures 14b and 14c show the spectrum of an impulse signal produced by the non-orthogonal transform for the frequency spacing used in Figures 12b and 12c, respectively.

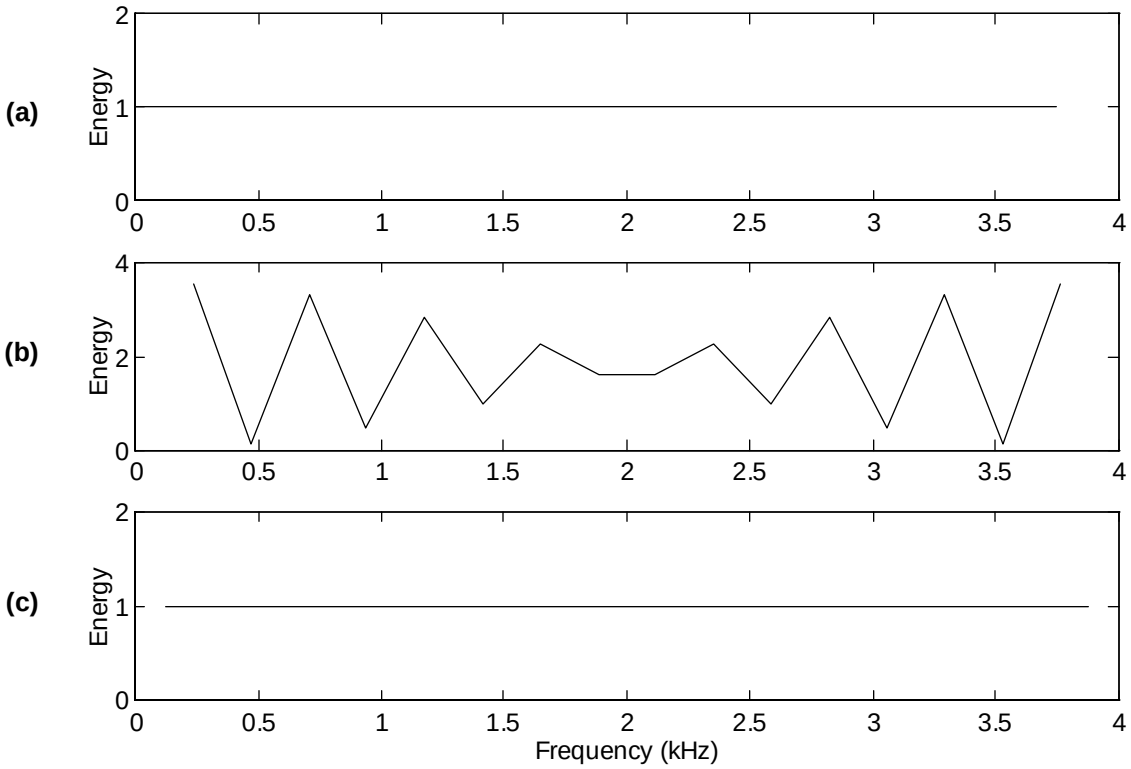

Figure 14 - Spectrum of an impulse signal produced by (a) Fourier transform, and produced by the non-orthogonal transform with (b) $d_2 = d_1$ (c) $d_2 = 2 \cdot d_1$

Consistent with theory, the Fourier transform of the impulse signal generates a flat spectrum, but the non-orthogonal transform of the impulse signal in Figure 14b is far from flat. It would be difficult to discriminate between the spectrum in Figure 14b and the spectrum of a signal with multiple sinusoids. For this reason, the non-orthogonal transform of an impulse signal was computed for many groups of frequencies. The result was that nearly every grouping of frequencies produced a non-flat spectrum for an impulse signal. The only exception was the

grouping of frequencies used to generate both Figures 12c and 13c; a flat spectrum identical to that of the Fourier transform was produced. With the results for this grouping of frequencies being as good or better than the Fourier transform, it was an obvious choice for use in the adaptive time-frequency distribution presented in Chapter 4.

One last difference between the two transforms is computational time. While the FFT is known and utilized for its computational speed, the computational time for the non-orthogonal transform is on the order of a Discrete Fourier Transform. One method which can decrease the computational time for the non-orthogonal transform is to force the frequencies of the transform to create an orthogonal matrix. Although it may seem to strive for orthogonality with a non-orthogonal transform, the group of frequencies used for Figures 12c, 13c, and 14c forms an orthogonal transformation matrix. This is likely the only set of frequencies which will produce an orthogonal matrix. As the concept of adaptability is introduced in the following section, it is important to remember that the properties of the non-orthogonal transform change with any change in the frequency set. Being able to predict and incorporate these changes will enhance the usefulness of the transform.

3.6 Adaptability

If the properties of this non-orthogonal transform were all that the transform had to offer it would not be obvious that the transform was useful. The most powerful aspect of the transform is the ability to adapt to the signal being analyzed. It is not an inherent ability of the transform, but it is easily incorporated.

The benefit of having a transform which adapts to the signal is that energy leakage is reduced or eliminated entirely. If a signal is made up of only a handful of sinusoids it will produce a relatively narrowband spectrum no matter which transform is used, but there will likely be energy in frequency bins where no signal exists. If the transform can exactly match the frequency and phase of the sinusoids present in a signal, there will be no energy leakage since all components of the signal have been accurately defined. This occurs in spite of the irregularity of the energy values produced by the transform alone. Once the frequency of a sinusoid is matched the ratio of leakage energy to total signal energy decreases significantly.

The adaptability of the transform is valuable for analyzing audio signals. The tones of most instruments are sinusoidal in nature, so using sinusoids for decomposition is a logical choice. Adapting to the sinusoidal components of an audio signal helps to localize the components, making them easier to identify and modify. It is important to note that the Fourier transform is an additive transform where FFT(a) + FFT(b) = FFT(a+b). If the FFT was capable of accurately identifying specific components in a signal, then signal separation would be as simple as identifying the components belonging to one signal and removing them from the spectrum. This is the goal of a signal separation system discussed later in this paper.

3.6.1 Adaptation Algorithm

Dologlou *et al* [5] have included in their paper an algorithm for adapting to the frequency of a sine wave. Although this algorithm serves as the basis for the adaptation capabilities of the adaptive time-frequency distribution, several enhancements have been added to improve the accuracy, speed, and usefulness of the results. The final adaptation algorithm will be outlined in Section 4.2. The steps for adaptation are shown in Figure 15.

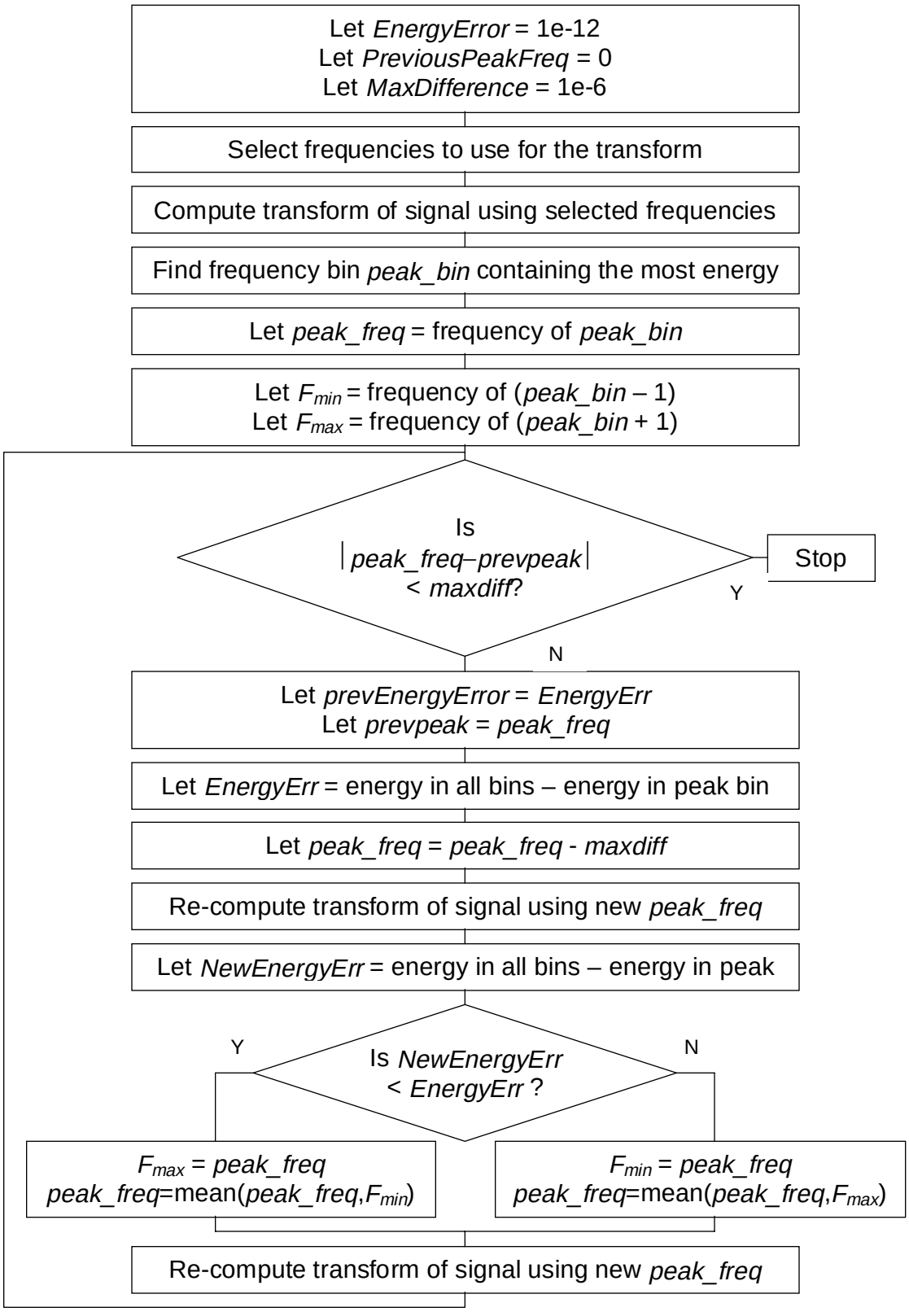

Figure 15 – Flowchart for adaptation algorithm

The adaptation begins by moving the peak frequency slightly in one direction. If the energy leakage decreases as a result, then make a larger move in that same direction; otherwise move in the opposite direction. The algorithm fails to recognize the situation where an energy increase results for a move in either direction. There are several other weaknesses of this algorithm which will be discussed in the next chapter. This algorithm works quite well in doing what it is intended to do – accurately identify the frequency of individual sine waves. Figure 16 shows two different signal spectra before and after adaptation.

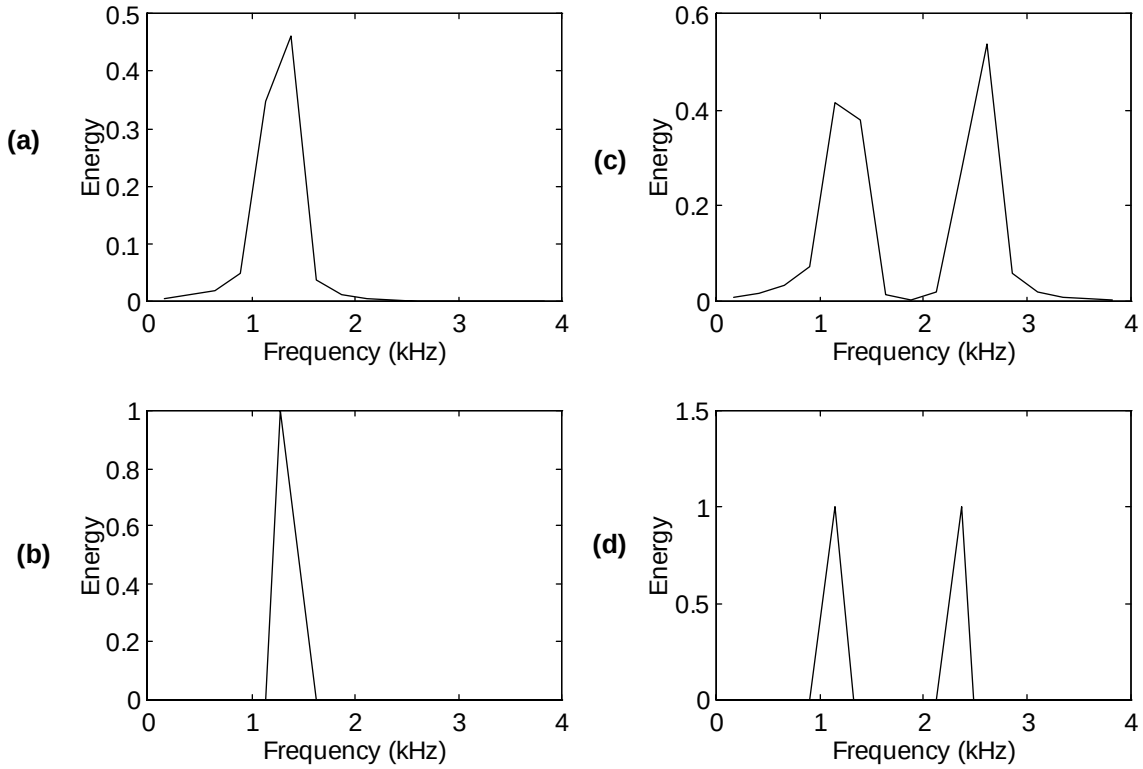

Figure 16 - Examples of adaptation using the non-orthogonal transform. Sine wave of frequency 1277Hz (a) before and (b) after adaptation. Sine waves of frequencies 1277Hz and 2503Hz (c) before and (d) after adaptation.

Figure 16a shows the spectrum of a sinusoid with frequency 1277Hz as computed by the non-orthogonal transform without adaptation. Figure 16b shows the spectrum of the same signal after adaptation to the frequency of the sinusoid. Notice that there is no energy leakage after

adaptation and the signal is fully defined by one bin. Figure 16c shows a similar example with two sinusoids of frequencies 1277Hz and 2503Hz, and Figure 16d shows the spectrum after adaptation. Again, all energy is contained within the frequency bins which exactly match the components of the signal. The ability to adapt to a signal will prove to be an invaluable asset to the adaptive time-frequency distribution presented in the following chapter.

Chapter 4 - Adaptive Time-Frequency Distribution

This chapter will present the new adaptive time-frequency distribution. Computation for the distribution includes perfect-reconstruction analysis and synthesis filter banks, non-orthogonal signal decomposition, and adaptation to the signal in every time-frequency block. The algorithms developed for this research were written for MATLAB.

4.1 Computation of Time-Frequency Distribution

The adaptive time-frequency distribution is based on the computation of the non-adaptive time-frequency distribution. If a signal is adequately described by the non-adaptive time-frequency distribution, then there is no need to adapt to the signal. This section will describe how the time-frequency distribution is computed.

4.1.1 Input signal

Before any computation can occur, the signal to be transformed must be selected. The input signal can be any audio signal sampled at rates of 44.1kHz or 48kHz with sixteen-bit quantization. Signals with eight-bit quantization could also be used, although the performance of the algorithm will decrease due to the reduced signal-to-noise ratio of the signal. Either one channel or two channels of input are allowed. The analysis of two channels is necessary for the blind signal separation system described in the next chapter.

The input signal will be broken up into time-frequency blocks. Each time-frequency block is one octave in width. The size of each time-frequency block is determined by the specific octave being analyzed and the number of frequencies used to analyze the content in the octave. To

properly analyze an octave of information, the number of time samples in an octave must be an integer multiple of the number of time samples in one time-frequency block. The input signal is padded with zeros to guarantee that each octave of information has exactly the number of samples needed for analysis. For the implementation used in this research, the length of the input signal must be an integer multiple of 8192, which corresponds to 186ms for a signal sampled at 44.1kHz. If a signal needs to be padded with zeros, half of the zeros are placed before the first sample and half are placed after the last sample.

4.1.2 Perfect-Reconstruction Filter Bank

Once it is selected and has the appropriate length, the input signal is separated into octaves using a perfect-reconstruction filter bank. The analysis filters $\mathbf{H_0}$ and $\mathbf{H_1}$ and synthesis filters $\mathbf{F_0}$ and $\mathbf{F_1}$ were designed using research from Michel Rossi, Jin-Yun Zhang, and Willem Steenaart [6]. Using this research, a 64-tap lowpass analysis filter $\mathbf{H_0}$ was designed with near-perfect reconstruction characteristics. The coefficients from $\mathbf{H_0}$ were used to design the remaining three filters. The relationship between the filter coefficients is illustrated in Table 2. The coefficients of the filters are included in the Appendix.

Table 2 – Relationship between filter coefficients (Adapted from Strang *et al*, 1996)

Filter	Sample Coefficients	Description
$\mathbf{H_0}$	a, b, c, d	
$\mathbf{H_1}$	d, –c, b, –a	Alternating flip
$\mathbf{F_0}$	d, c, b, a	Order flip
$\mathbf{F_1}$	–a, b, –c, d	Alternating signs

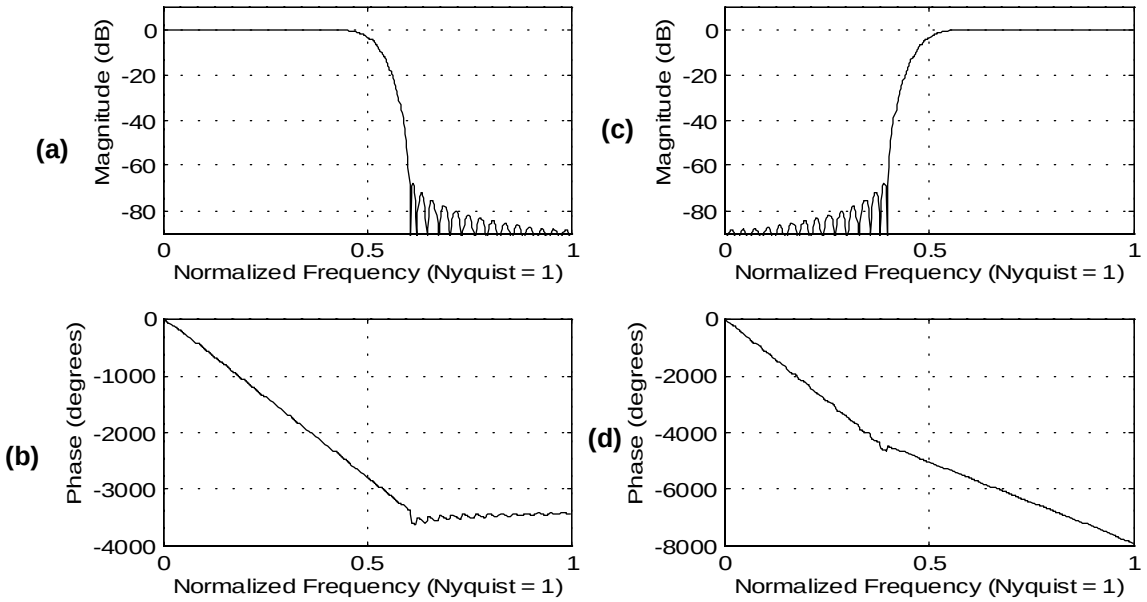

Figure 17 - Analysis filter bank frequency response. Lowpass filter H_0 (a) magnitude and (b) unwrapped phase response. Highpass filter H_1 (c) magnitude and (d) unwrapped phase response.

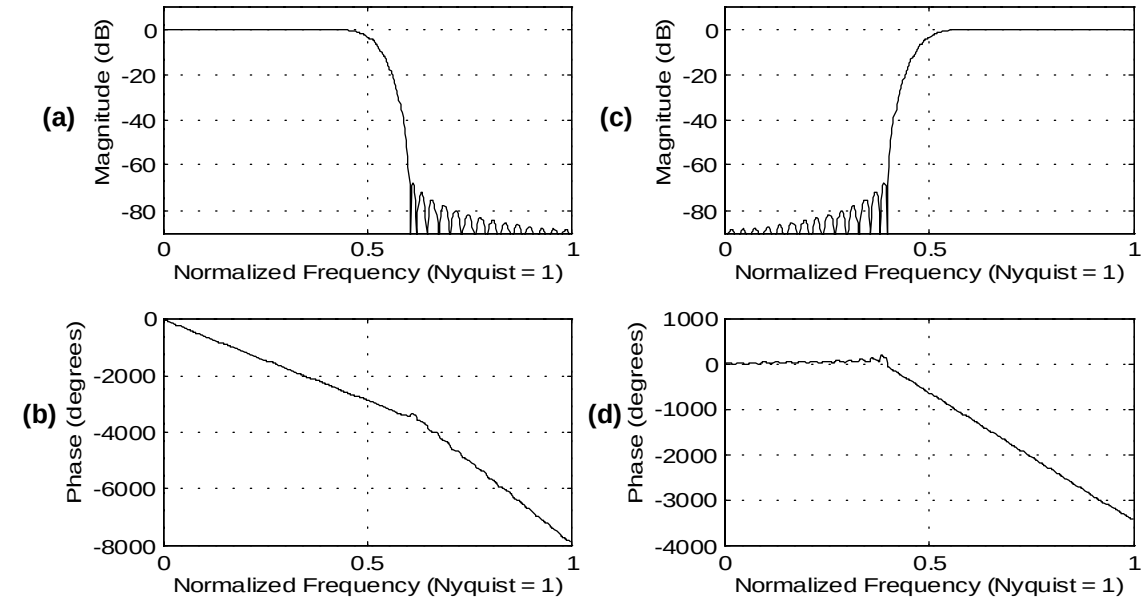

Figure 18 - Synthesis filter bank frequency response. Lowpass filter F_0 (a) magnitude and (b) unwrapped phase response. Highpass filter F_1 (c) magnitude and (d) unwrapped phase response.

Figures 17 and 18 show the frequency responses of the analysis and synthesis filters respectively. These filters are applied to the input signal $x(n)$ as illustrated in Figure 19. The analysis filter bank is applied nine times to generate ten separate octaves of information contained in $y_1(n)$

through $y_{10}(n)$. The downsampling operator reflects the upper half of all frequency content below the new Nyquist frequency while the remaining frequency content is left unaltered. The highpass analysis filter $\mathbf{H_1}$ removes most but not all of the lower frequency content before downsampling. For this reason, signal content from adjacent octaves will be present in each octave.

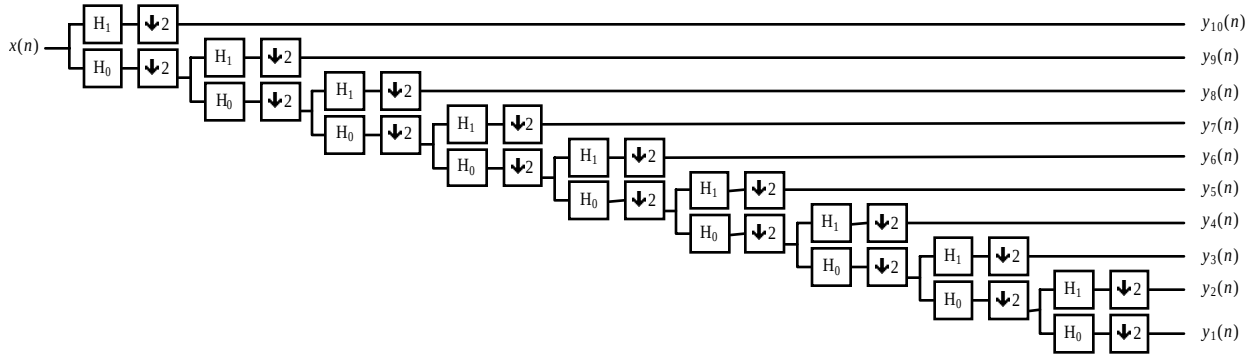

Figure 19 – Analysis filter bank implementation

If the intended use of the time-frequency distribution is for analysis only, then perfect-reconstruction filters are not needed. Filters with narrower transition bands could be used in place of the perfect-reconstruction filters in order to reduce energy leakage from adjacent octaves.

The algorithm which applies the analysis filter bank to the input signal also compensates for the delay introduced by each filter. While the delay compensation allows for more accurate time-frequency plots, it also affects the accuracy of the perfect reconstruction operation. The delay compensation is implemented by removing from the beginning of the signals $y_1(n)$ through $y_{10}(n)$ the number of samples corresponding to the delay and adding the same number of zeros to the end of each signal. The removal of samples affects only the beginning of the reconstructed signal. As a result, the time-frequency blocks from all octaves are correctly aligned.

4.1.3 Non-Orthogonal Signal Decomposition

Once the signal is broken up into octaves, it can reliably be decomposed by the non-orthogonal signal decomposition transform. The transform will be used independently on each of the ten filtered signals $y_1(n)$ through $y_{10}(n)$. A time-frequency distribution will be generated by analyzing each signal in short time segments and calculating the spectral content for each time segment. In principle, the method of calculation is identical to calculating the spectrogram of a signal without overlapping.

With the non-orthogonal transform, a set of frequencies need to be selected for analysis. For the sake of simplicity, the number of frequencies and their values could be made identical for each band, but different sets of frequencies in each band may provide better time-frequency resolution. Table 3 shows the size of a time-frequency block for each octave using either four, eight, or sixteen frequencies per time-frequency block.

Table 3 Size of Time-Frequency Blocks

Octave	4 Frequencies	8 Frequencies	16 Frequencies
10	0.363ms	0.726ms	1.45ms
9	0.726ms	1.45ms	2.9ms
8	1.45ms	2.9ms	5.8ms
7	2.9ms	5.8ms	11.6ms
6	5.8ms	11.6ms	23.2ms
5	11.6ms	23.2ms	46.4ms
4	23.2ms	46.4ms	92.9ms
3	46.4ms	92.9ms	186ms
2	92.9ms	186ms	372ms
1	186ms	372ms	743ms

As a basis for comparison, the time block representing a 1024-point FFT of a 44.1kHz-sampled signal is 23.2ms. As Table 3 shows, for sixteen frequencies per octave, the non-orthogonal transform has equal or better time resolution than a 1024-point FFT for the upper five octaves

but worse time resolution in the lower five octaves. While the size of the time blocks differ between the FFT and the non-orthogonal transform, the number of time-frequency points remains constant in each octave. For instance, a 1024-point FFT has 256 time-frequency points in the highest octave with a time width of 23.2ms, which equates to 11.03 time-frequency points per millisecond. In the same octave, the non-orthogonal transform has sixteen time-frequency point in a period of 1.45ms, which also equates to 11.03 time-frequency points per millisecond. Therefore the new time-frequency distribution does not create better time-frequency resolution, it just allows for a more flexible distribution of the time-frequency points.

Originally, each band was selected to have sixteen frequencies per octave. This number was chosen because the highest five octaves had time resolution equal to or better than the 1024-point FFT while the lowest five octaves had better frequency resolution than the 1024-point FFT. An advantage to having the same number of frequencies for every octave is that the same matrix can be used to analyze every single time-frequency block. This is true for the same reason that the analysis and synthesis filter banks can be reused. Since the non-orthogonal transform simply analyzes frequencies between 0 and π, it can be applied to every octave.

A number of audio signals were analyzed using sixteen frequencies per octave to evaluate the distribution of time-frequency points. These signals included sounds generated by piano, acoustic guitar, electric guitar, bass guitar, and didgeridoo. In addition, an impulse signal and a chirp signal were analyzed. The time-frequency points for the highest seven octaves appeared to match the signal content well. For the lowest three octaves the time resolution was very poor and the extra frequency resolution was poorly matched to the signal. The time-frequency

distribution was recomputed for the same audio signals using both eight frequencies and four frequencies in the lowest three octaves. The arrangement which gave the best tradeoff between time resolution and frequency resolution was as follows: the highest seven octaves were set at sixteen frequencies, the third octave was set at eight frequencies, and the lowest two octaves were set at four frequencies.

The selection of frequency values for the non-orthogonal transform was discussed in detail in Section 3.3. The set of frequencies used for the adaptive time-frequency distribution were derived from Equation 4 using $d_2 = 2 \cdot d_1$. The resulting relationships are $d_1 = \frac{f_s}{4N}$ and $d_2 = \frac{f_s}{2N}$. All audio signals analyzed for this research had a sampling frequency f_s of 44.1kHz. Therefore the frequency spacing is a function of only the number of frequencies analyzed in each octave. Specifically, N is 4 for the lowest two octaves, 8 for the third octave, and 16 for the remaining seven octaves. The transformation matrix $\boldsymbol{B}$ and its inverse $\boldsymbol{A}$ are formed for each of the three values of N.

There are two aspects of the adaptive time-frequency distribution which affect the relationship between a time-frequency point and the actual frequency it represents. First, each analysis matrix is constructed from frequencies between 0 and π. These boundary frequencies correspond to the upper and lower frequency limits imposed by the filter bank. Second, the downsampling operator applied to each filtered signal in the filter bank inverts the frequency content in each octave. Therefore the upper frequency boundary π corresponds to the lower frequency boundary of the octave and vice versa. Equation 5 describes the conversion between

the digital frequency of the non-orthogonal transform and the actual frequency of a time-frequency point

$$f_{act} = \frac{f_s \cdot 2^{octave-10}}{2} \cdot \left(1 - \frac{f_{dig}}{2\pi}\right) \qquad \text{(Eq. 5)}$$

where f_{dig} is the digital frequency, f_{act} is the actual frequency, and *octave* is an integer between one and ten with one representing the lowest octave.

Once the transformation matrix $\boldsymbol{B}$ is computed for all possible frequency selections, the appropriate matrix is applied to each octave-filtered signal. For example, the time-frequency distribution of the highest octave $y_{10}(n)$ is computed by first resizing y_{10} from a vector of length L to a matrix y_{10a} with 32 rows and $\frac{L}{32}$ columns. Each column of matrix y_{10a} consists of a time block of 32 samples. The time-frequency distribution z_{10} is the product of matrix $\boldsymbol{B}$ with matrix y_{10a}. Matrix z_{10} is equal in size to matrix y_{10a} and contains the coefficients representing magnitude and phase information for each frequency. For magnitude information only, the matrix z_{10a} is formed which has half as many rows as matrix z_{10} and each element is the sum of the square of each pair of coefficients.

The time-frequency distribution is computed for each octave using the correct number of frequencies per octave. A time-frequency plot is generated by combining all of the octave time-frequency distributions z_{1a} through z_{10a} into one matrix. This matrix must be formed in order to allow the MATLAB program to create one plot containing all time-frequency points. Since z_{10a} contains the greatest number of time-frequency points, all other distribution are forced to be equal in size to z_{10a}. For example, distribution z_{10a} has four times the number of frequencies of z_{1a}

and 128 times the number of time-frequency blocks of z_{1a}. To make the two distributions equal in size, each element of z_{1a} is replaced with a matrix of four rows and 128 columns where each element in the new matrix is identical to the element being replaced. The time-frequency plots of several different signals are included in the next chapter.

One issue involving the use of the non-orthogonal transform for a time-frequency distribution was the use of windowing or overlapping. Due to the adverse effects on the signal and corresponding distribution, windowing and overlapping were not applied to the distribution. Since the adaptive distribution can adapt to the signal, the problems caused by the effects of discontinuity at time block edges are lessened. In addition, overlapping would not be useful in a perfect reconstruction system since the additional time-frequency blocks could not be used for reconstructing the signal.

4.2 Improved Adaptation Algorithm

Figure 20 contains the flowchart for the improved adaptation algorithm. The adaptive time-frequency distribution uses all the information from the time-frequency distribution discussed thus far to determine if any adaptation is needed. One disadvantage of adapting to the signal is that the frequencies used to decompose each time-frequency block must be recorded. Unlike the non-adaptive case, where one frequency set is used for all the time-frequency blocks in an octave, the adaptive distribution could potentially have a different set of frequencies for each and every time-frequency block. Since there are two coefficients associated with each frequency, the data size of the distribution plus the adapted frequencies is 50% greater than the data size of the original signal.

Several improvements were made to the adaptation algorithm listed in Section 3.6.1. First, the total energy contained in each time-frequency block is calculated. All blocks with energy below a certain threshold are ignored during the adaptation process. The threshold is set at 40dB below the time-frequency point in any octave with the largest amount of energy. Second, several frequencies can be adapted to in one time-frequency block if it is determined that there is more than one frequency bin with significant energy. This is tested for by sorting the time-frequency points in a block in order of energy level. Only the time-frequency points within 10dB of the peak energy point in the block are adapted. One additional constraint on the number of frequencies adapted to is that no more than one-quarter of the frequencies in a time-frequency block can be adapted. These improvements were made to reduce the computational time of adapting to the signal and to improve the results generated by the algorithm. The threshold values of 40dB and 10dB were selected after analyzing the audio signals discussed in the previous section. The only purpose of the threshold values is to reduce computation time by ignoring time-frequency points with little significant energy. Without the threshold values, the algorithm would adapt to one of every four time-frequency points.

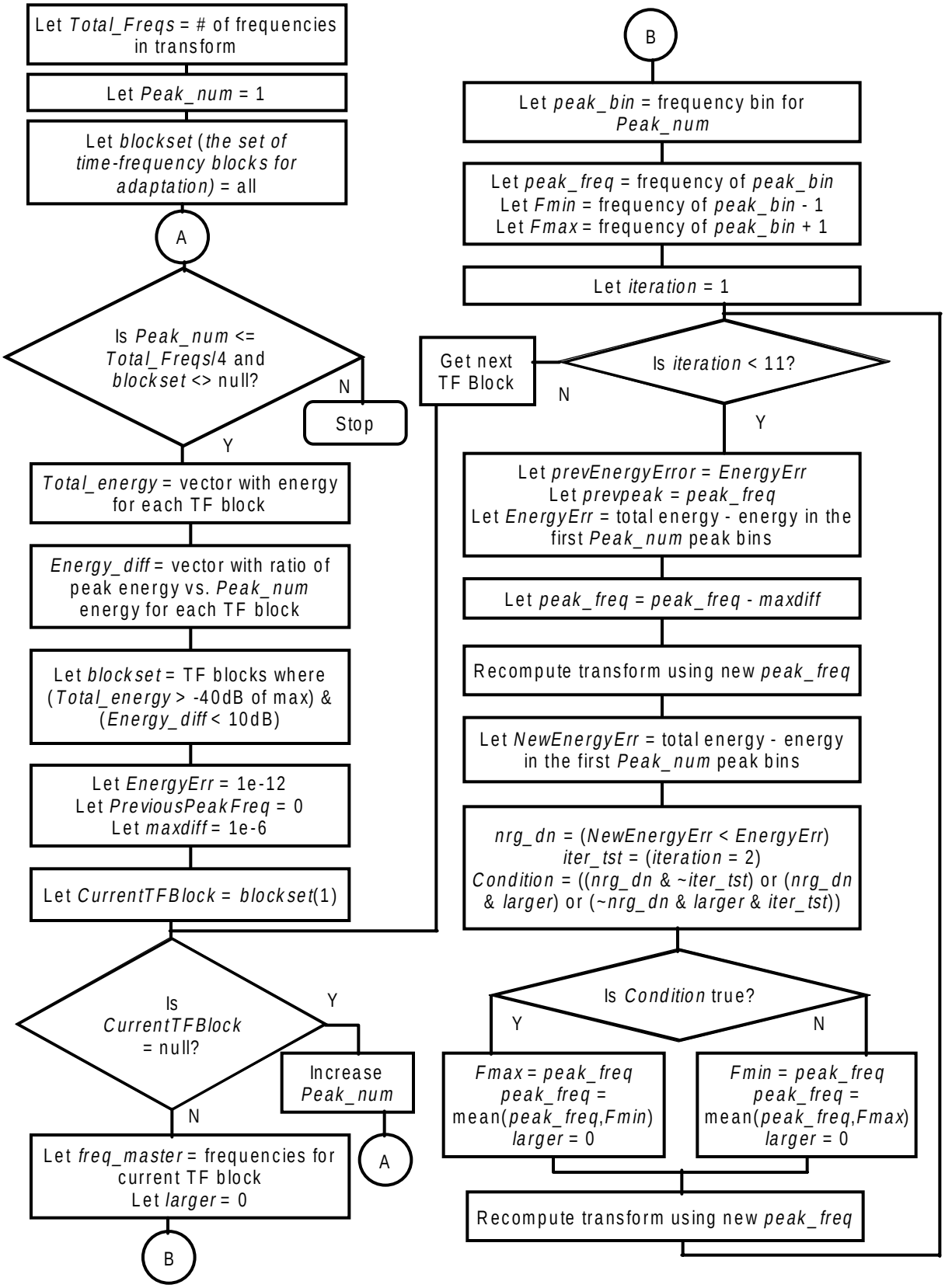

Figure 20 – Flowchart for Improved Adaptation Algorithm

A third improvement made to the algorithm is a condition which forces any time-frequency point to stay within certain boundaries during adaptation. During testing of the algorithm, certain signals were analyzed and the peak frequency would not converge on a nearby frequency. Instead it would continuously move to a higher or lower frequency until it was virtually on top of an existing frequency. This caused the distribution to produce worse results than before any adaptation occurred. The solution for this problem was to monitor the second iteration of an adaptation. For example, if a peak frequency moved lower after one iteration, it would be halfway between the original peak frequency and the neighboring frequency bin. If the frequency needed to be decreased again, the algorithm forced the frequency to be increased. Even if the peak frequency continues to move lower in the remaining iterations, the decrease in step size for each iteration guarantees that the peak frequency will always be closer to the original peak frequency than any other frequency bin.

One change made to existing code in the original adaptation algorithm was the criteria for determining when to cease adaptation. The original algorithm compared the difference between the last adapted frequency and the current frequency. If that difference was less than a threshold value, adaptation was complete for the particular frequency. The new algorithm counts the number of iterations performed on a particular frequency and stops adaptation after exactly ten iterations. Since the difference between frequencies before adaptation is known, and the step size for a frequency decreases by a factor of two after each iteration, the number of iterations directly corresponds to the step size. With the analysis frequencies chosen for this research, the frequency spacing before adaptation is $\pi/16$ and after ten iterations the maximum frequency

difference is $\frac{\pi}{16384}$. For a frequency of 1000Hz, which lies in the sixth octave, the frequency spacing is 40Hz and ten iterations provides a maximum frequency difference of 0.04Hz.

One disadvantage to this method is that the algorithm does not know when it has exactly matched the frequency of a signal. For example, if a sine wave with a frequency halfway between two frequency bins is analyzed, only one iteration is needed to accurately adapt to the signal. However, the algorithm is designed to provide a maximum error in determining the frequency of a signal and does not know when the signal has been precisely matched. If only one frequency was present in a time-frequency block, the energy leakage could be monitored and the adaptation algorithm could be exited when the energy leakage is zero. If there are multiple frequencies in a time-frequency block, this method would not work.

Chapter 5 - Distributions of Audio Signals

In this chapter, several time-frequency distributions created for this research will be included. The properties of the distribution will become apparent and the results of adapting to a signal will be observed.

5.1 Impulse Signal

Many electrical systems are tested and identified through the use of an impulse signal. A basic system can be completely defined by its impulse response. For system identification applications, an impulse signal is sent through the system input and the measured output is the impulse response of the system. Figure 21 shows the time-frequency distribution of an impulse signal.

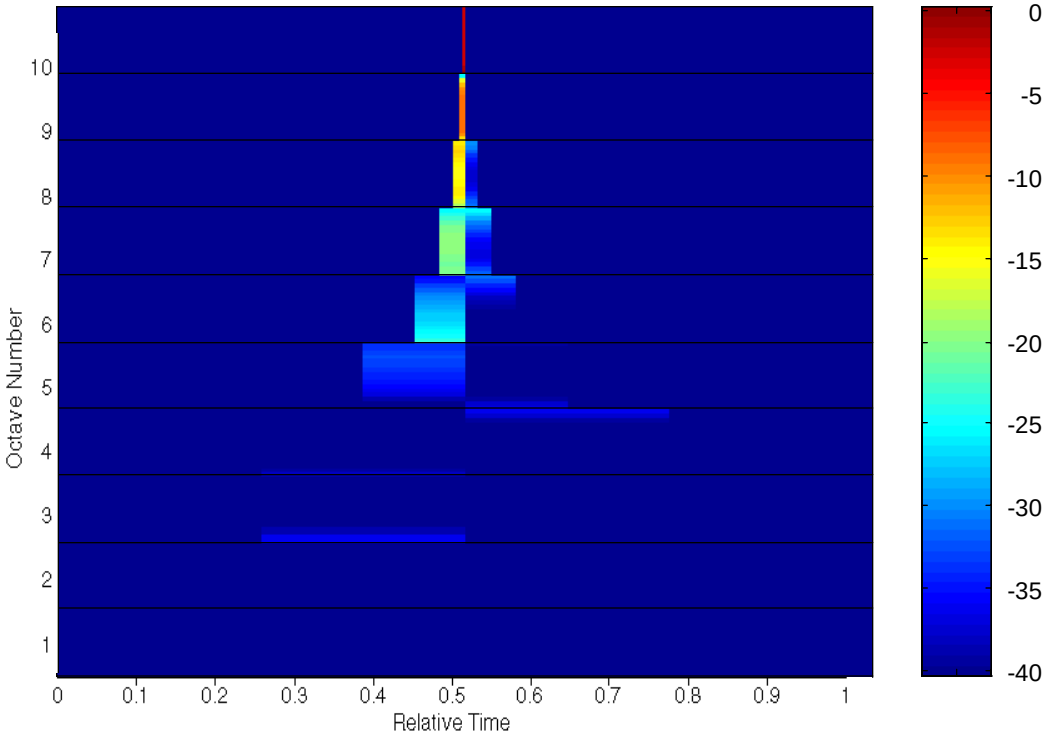

Figure 21 – Time-frequency distribution of impulse signal without adaptation (energy is grayscale)

The energy in the distribution is plotted in terms of decibels with the time-frequency point containing the most energy set to 0dB. The amount of energy present at each time-frequency point is represented as grayscale shading where black represents 0dB and white represent –40dB.

The Fourier transform of an impulse signal produces equal energy at all frequencies. For this distribution, each of the ten octaves is analyzing a filtered impulse signal. The time-frequency distribution shows a wideband spectrum for the time blocks which contain energy. The impulse signal appears to be wider in time for lower octaves. This is a consequence of the width of each time-frequency block at lower octaves. For each of the highest six octaves, the impulse signal creates energy in only one or two time-frequency blocks. Also, there is little energy in the lowest four octaves.

5.2 Chirp Signal

The time-frequency distribution of a chirp signal before adaptation is shown in Figure 22. The use of octaves in the time-frequency distribution creates a distribution unlike the spectrogram of a chirp signal. The signal appears to have a frequency which changes exponentially rather than linearly with respect to time. This property would appear to be a disadvantage in analyzing signals. However, the chirp signal is not like most musical signals since it moves quickly through the lower octaves while most of the signal energy is contained in the higher octaves.

For each of the times when the chirp signal transitions from one octave to the next, the distribution shows energy in both adjacent octave simultaneously. This is due to the wide transition bands of the analysis filters. When the chirp signal is filtered, any signal energy which falls within the transition band of the analysis filter remains as part of the signal to be analyzed.

If the perfect reconstruction of the signal is not desired, analysis filters with narrower transition bands could be used to alleviate this problem.

Although the chirp signal consists of only one frequency at any given time, the distribution shows energy leakage spread to many other frequency bins. The advantage to using an adaptive time-frequency distribution is that the frequencies used to analyze the chirp signal can be altered in a way which best matches the signal. The distribution of the chirp signal after adaptation is shown in Figure 23. The algorithm calculates the amount of energy leakage before and after adaptation, and for the chirp signal the energy leakage is reduced by over 95% after adaptation. The higher octaves of the distribution approach the ideal time-frequency distribution for a chirp signal. The method used to plot the distribution only plots time versus frequency bin, not actual frequency. If the distribution was formed using the frequency for each bin, the distributions in the higher octaves would be perfectly straight lines. Whether the adaptive time-frequency distribution is used for analysis only or for applications such as signal separation and audio compression, the advantages of adaptation are clear. The distribution approaches the ideal time-frequency distribution and localized the components present in a signal.

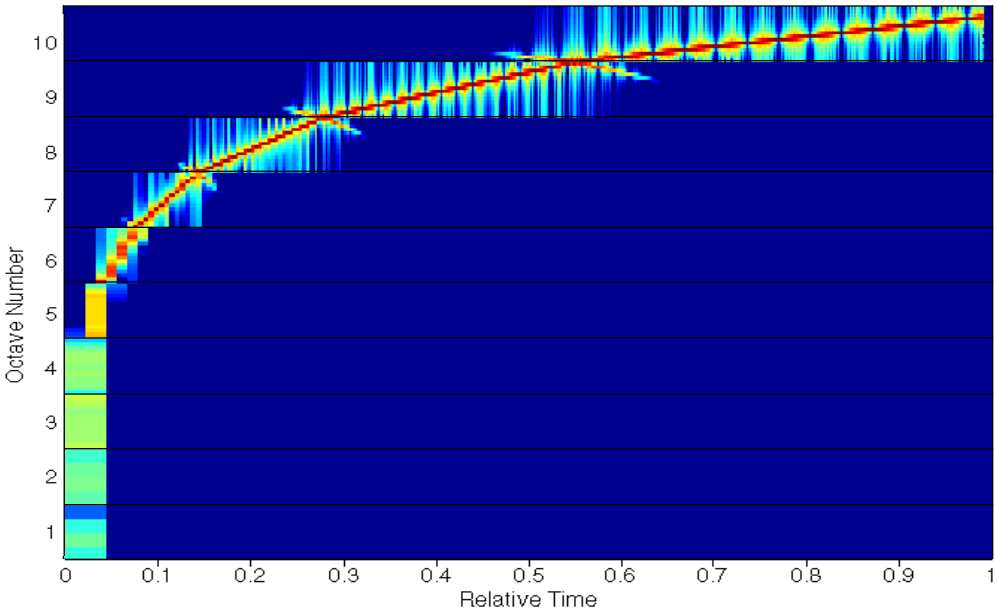

Figure 22 – Distribution of chirp signal before adaptation

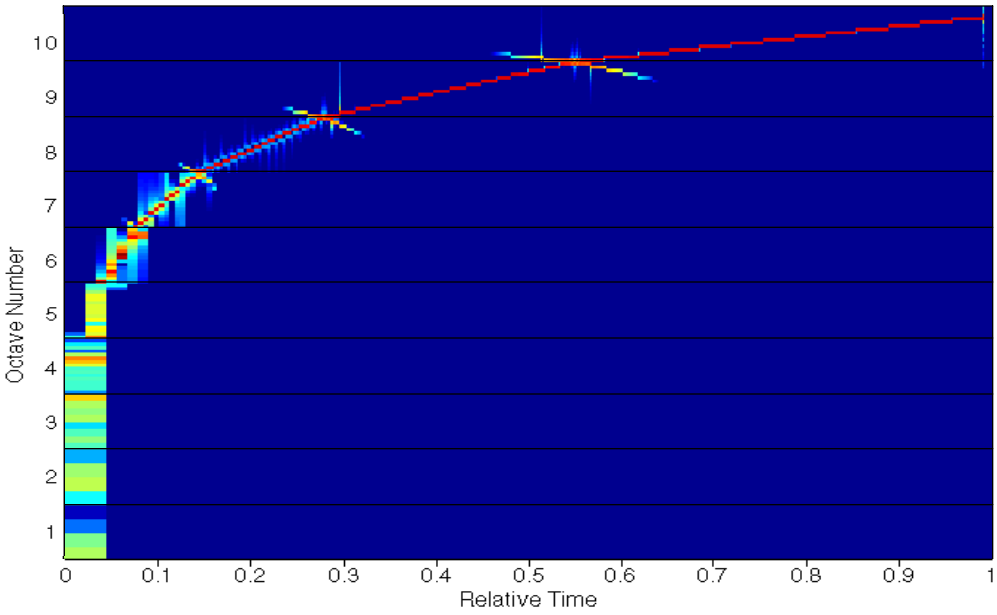

Figure 23 – Distribution of chirp signal after adaptation

5.3 Piano

Figures 24 and 25 show the time-frequency distributions of a piano excerpt before and after adaptation, respectively. Many components of the signal which were difficult to discern before adaptation appear to be more sinusoidal after adaptation. In addition, there is virtually no frequency content in the highest three octaves of the signal. Any other time-frequency distribution with a linear distribution of frequencies would produce a plot where the highest three octaves would take up nearly 90% of the time-frequency plane. This does not mean that the frequency content above 2.5kHz is unimportant for audio, but rather the frequency information usually difficult to measure with other distributions is more clearly illustrated in the adaptive time-frequency distribution.

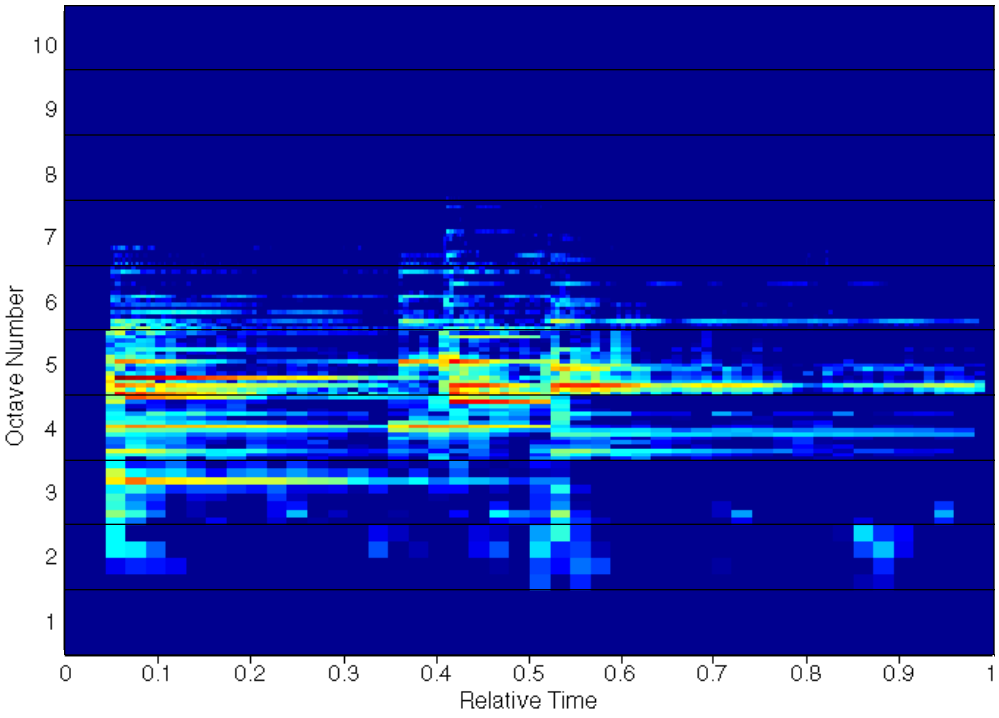

Figure 24 – Distribution of piano recording before adaptation

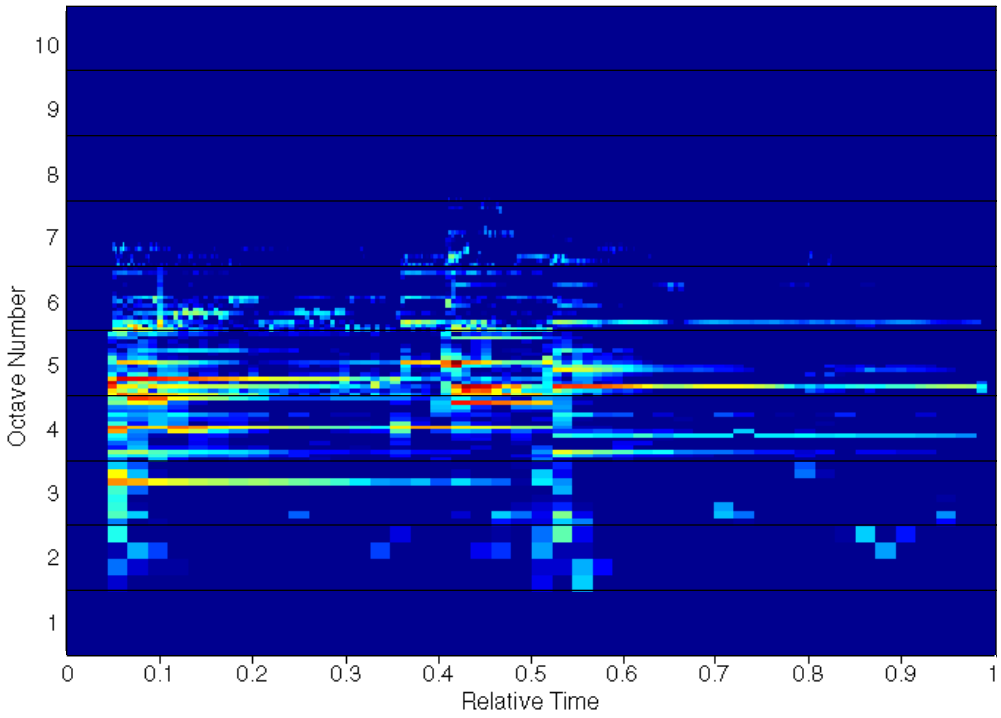

Figure 25 – Distribution of piano recording after adaptation

5.4 Electric Guitar

The distribution of an electric guitar excerpt before and after adaptation is shown in Figures 26 and 27 respectively. The frequency content of the signal is highly sinusoidal and is a good match for the transform. The first portion of the signal contains a string bend with vibrato which can be more clearly seen from the adapted distribution. If the plot was generated in terms of the exact frequencies used in each time-frequency block, the bend and vibrato would appear as smoother curves rather than jerky lines. Although the signal content was described fairly well by the distribution before adaptation, significant energy leakage is reduced after adaptation.

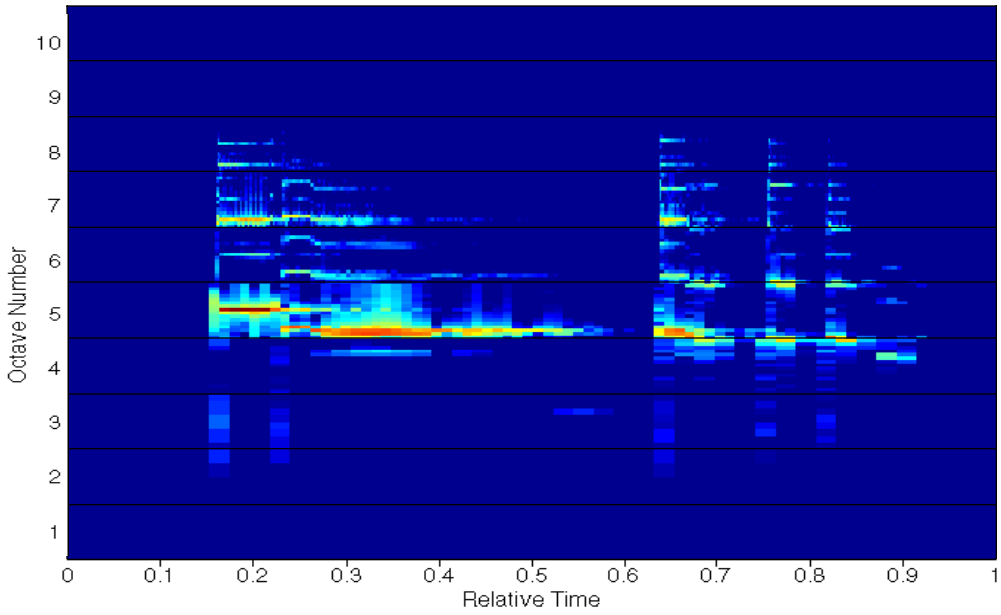

Figure 26 – Distribution of electric guitar recording before adaptation

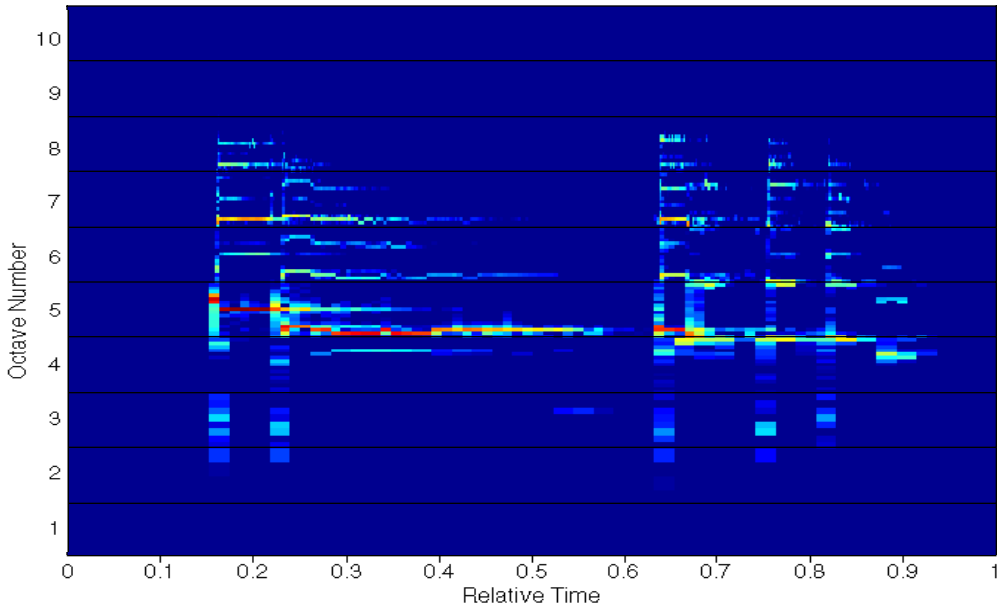

Figure 27 – Distribution of electric guitar recording after adaptation

Chapter 6 - Blind Signal Separation

6.1 Introduction

The general field of study which relates to separating signals without information on the signals or the signal mixtures is known as blind signal separation. The structure of a blind signal separation problem consists of a number of unknown signals which have been combined in an unknown manner. The only assumption about the signals is that they are statistically independent of each other. A successful blind signal separator would input a number of observations of the combined source signals and output each of the source signals. For example, if five people were speaking in the same room at the same time, each person is considered a source. In addition, if a number of microphones were present in the room, each microphone would pick up a combination of the sources present at the microphone's location, which is considered an observation of the sources. Most current research deals with the case when the number of source observations is greater than or equal to the number of sources present. In terms of audio recordings, if each instrument is considered a source and each channel of the recording is considered an observation, then it is very common to have a recording with more sources than observations.

There is an interest in being able to remove specific sound sources from audio recordings. As an example, a common source of entertainment at clubs today is Karaoke, where a recording of a popular song is played without the lead vocals and a person sings the missing part. Similarly, there are many young musicians who are inspired to play an instrument because of a musical role model. These same musicians often learn their instrument by attempting to play along with their favorite recordings. This process would be made easier if the desired instrument was separated

from the remaining instruments in the recording, leaving the desired instrument by itself and the remaining instruments intact. The musician could then listen to the instrument alone to capture all of the nuances of the performance and then play along with the remaining instruments when practicing the part. The list of applications for separating sound sources from audio recordings is extensive.

One application of a perfect signal separation system for audio recordings is not obvious but worth mentioning. If a system could accurately separate sources from a two-channel recording, then a lossless audio compression algorithm has been developed. For instance, a recording engineer wishing to preserve the original source tracks used on a recording usually has to save a master recording which keeps each track stored separately. With the aid of a perfect separation system, the engineer could instead create a two-channel recording where each source is simply panned to a unique location in the stereo field. Each source could then be extracted one at a time to recreate the master recording with no loss in audio quality. The compression ratio is the number of original source tracks over two tracks. In essence, the same media which is used to store the commercial version of an album could be used to store each of the source tracks for every song on an entire album.

The signal separation system needed for this application would only be for use on two-channel recordings. This system will accept both channels as inputs and attempt to output one source at a time. In addition, it would be advantageous to use properties of psychoacoustics and mixing consoles to assist in signal separation. These properties will be discussed throughout the remainder of this chapter.

6.2 Audio Signals and Spectra

One of the limitations of this system is that the source signals will be musical in nature. They will most likely belong to the woodwind, percussion, string, brass, or voice families. Although it is impossible to mathematically classify all the potential sources as being from one group and significantly different from all other groups, there are properties of musical signals which differ from sources such as white noise or pure sinusoids. These differences must be accounted for in order to improve system performance. For example, a vibrating string or air column produces pitches with time-varying harmonics. This implies that only a limited number of sinusoids would be needed to capture a specific instrument at any given time.

6.3 Mixing Consoles

In addition to being focused on musical signals, this system pertains to the studio recording of these musical signals. Instead of assuming no prior knowledge about how musical instrument sounds appear on audio recordings, the properties of the studio and the mixing console can be used to improve the success of the separation algorithms.

In general, any instrument must be considered a monophonic source in terms of a studio. For example, if a piano is being recorded using two microphones, the output of each microphone will differ slightly based on the microphone's position with respect to the piano. However, each microphone output can be considered a filtered version of the one "true" piano sound. This is an important aspect to address since this research deals with "source separation." In order to separate the piano sound from a recording, it must be assumed that the piano is one source, regardless of how many microphones were used to record the piano.

Although there are situations where one instrument is captured using more than one microphone, it is quite common to simply use one microphone for an instrument. In that case, the mixing console receives only one signal which corresponds to a specific instrument. Since almost all commercially-available recordings are in stereo, each of these instruments is converted from a mono signal to a sound in the stereo field in one of several ways described in the sections below.

6.3.1 Mono Panned Signals

One of the most basic elements of a mixing console is the pan pot, or panoramic potentiometer. The pan pot creates a stereo signal by taking a monaural signal and placing a percentage of it in each of the stereo channels. The percentage used for each channel is dependent on the position of the pan pot. If the pan pot is assumed to have a total range of 90° of motion, with 0° as the leftmost position, the ratio of left channel level to original source level is $\cos(\theta)$ while the right channel to original source level ratio is $\sin(\theta)$, where θ is the pan pot position.

Panning is used in stereo recordings to give a sense of position to a specific instrument. The human auditory system uses interaural amplitude differences and interaural time differences to determine the location of a sound source. Panning produces the effect of interaural amplitude difference, while the procedure in the next section produces the effect of interaural time difference. A monaural recording combines all the desired sounds into one channel, which creates the sensation that all of the performers are a point source. Panning the various sources to different locations creates the sensation that each performer is standing in a unique location, which is more in tune with a real-world performance.

In terms of solving the source separation problem, panning offers an element of simplicity. Panning has no effect on the frequencies present in a signal. If a song was mixed using only panning as a stereo effect for every source, then elimination of a specific source is as simple as taking a weighted difference between the two channels. Of course, the amplitude of every other source is affected, but one source will certainly be removed. Since most modern recordings make use of other stereo effects which significantly deteriorate the success of such a method, it is hardly a solution to the general source separation problem.

6.3.2 Time Delay

As mentioned in the previous section, interaural time differences are used by the human auditory system to determine the location of a sound source. If it takes longer for a sound to reach the right ear as opposed to the left ear, then the auditory system perceives that the sound source is to the left. By electronically adding a time delay to one channel and leaving the other channel unaltered, the sound source appears to come from the direction of the unaffected channel, as described by the Haas effect.

Time delay is another stereo effect which does not alter the frequencies present in a signal. A successful separation algorithm would have to detect the time difference in order to improve source separation.

6.3.3 Phase Reversal

Phase reversal is used to create a stereo effect by simply placing the source signal unaltered in one channel and with opposite polarity in the other channel. This effect does not occur for any naturally-produced sound.

Removal of a source which has been phase-reversed in one channel is trivial in the presence of other sources which have been only panned. The simple sum of the two channels eliminates the phase-reversed source, however the amplitude of all other sources are affected. When using frequency analysis techniques such as the Fourier transform, the magnitude of the frequency response remains unchanged after phase reversal while the phase response shifts by 180°. This would have to be accounted for in the separation algorithm.

6.3.4 Reverberation

Reverberation is the natural effect of a sound source in an environment with multiple sound-reflecting surfaces. A microphone in this environment would pick up the original source material along with time-delayed reflections from the surroundings. If the reflecting surfaces are not smooth, the reflections tend to have less high frequency energy since the wavelength for higher frequencies is on the order of the surface detail.

In a recording studio, it is common for rooms to be treated in such a way that reduces or eliminates all reflections when recording a musical source. Artificial reverberation is then added to create the illusion of the source placed in a different environment. Artificial electronic reverberation is the dominant source of reverberation on commercial recordings and can be implemented by adding multiple delayed and filtered versions of the original source material to the source itself.

A decision has to be made in terms of the source separation system. If a single-channel reverberation is applied to a mono source signal, what is the desired output of the separation

system - the original source or the source with reverberation? The answer may depend upon the application, but it also is affected by another common property of artificial reverberation units - most reverberation systems accept a mono source and output two separate reverberant channels, each channel slightly different but representative of the same environment.

Two-channel reverberation creates a much more complicated separation problem if the original source signal is desired. This is often referred to as the convolutive-mixture problem rather than a linear-mixture problem. When source signals are combined using panning only, the resulting channels are linear combinations of the source signals, hence the name: linear-mixture problem. The convolutive-mixture problem occurs when the source material is filtered in some way that is not desirable upon separation. Not only do the sources need to be separated, but the filters which are present in the system need to be estimated and compensated for.

6.3.5 Other Stereo Effects

Other filtering effects process the original source signal differently for each channel. These effects are also considered as part of the convolutive-mixture problem since each source is filtered before being added to the final mix. Any convolutive-mixture problem eliminates the possibility of using simple channel addition techniques for eliminating sources.

6.3.6 Time-varying Effects

An additional factor to consider for any of the above methods is how the effect changes with time. Time-varying effects are often used on recordings to draw attention to a particular source. The most common ways to change an effect with time are to instantaneously switch from one effect to another, and to continuously vary one effect of a parameter.

An instantaneous switch between two effects is a simpler signal separation problem than the time-varying parameter condition. Before switching takes place, the effect can be considered constant with respect to time, and the same holds true for after switching. However, the switch between effects needs to be detected in order for this method to work. The time-varying parameter condition presents one of the most difficult source separation problems since the ideal separation technique would have to model the effect as well as determine one parameter which directly corresponds to the parameter being modified in the studio. A different approach is to assume that for short blocks of time the effect is constant throughout the duration of each block.

One other factor which affects the separability of sources with time-varying parameters is whether or not the effect is single-channel or dual-channel. This condition identifies a linear-mixture problem or the convolutive-mixture problem, respectively. In the single-channel condition, the source could be separated with the time-varying effect still intact. It would be more difficult to separate the original source from the time-varying effect since the effect is unknown and indistinguishable from the source in terms of separability. A dual-channel condition would give some cues as to what is similar and what is different between the channels, which could aid in determining the original source material.

6.4 Amplitude vs. Relative Amplitude

In a source separation problem, the objective is to extract the time and frequency properties of a signal in the presence of other signals. In general, we are not concerned with the overall amplitude of the signal with respect to its amplitude in the observed signals. On the other hand, the relative amplitude of a source signal in two observations is very important in defining how the

source was added to the observations. A panned signal creates relative amplitudes of the source in the observations based on the position of the pan pot. The relative amplitude of a source in two observations can be used to determine the position of the pan pot used in the mixing process, and this information can aid in separating the sources.

6.5 Digital Audio and Digital Signal Processing

The DFT will transform a signal vector of length *n* into a vector containing *n* frequencies. For a band-limited signal sampled at frequency f_S, the *n* frequencies are equally spaced between $\frac{-f_s}{2}$ and $\frac{f_s}{2}$, which produces a frequency resolution of $\frac{f_s}{n}$ between frequency bins. For example, when analyzing 1024 samples of a signal sampled 44,100 times per second, a DFT would provide frequency information at 43.1 Hz intervals. Also, when the number of samples analyzed is non-prime, a transform known as the Fast Fourier Transform, or FFT, can be used to reduce the computational complexity of the algorithm.

Although the FFT completely defines the frequency content of a group of samples, the frequency values which are used in the transform cannot be altered without changing f_S or *n*, which has several consequences. First, for a sinusoid whose frequency does not match the frequencies of the FFT, the resulting transform will show energy content spread out to nearby frequency bins. Also, since the source signals are musical instruments, the signals will often be sinusoidal in nature with multiple harmonics. However, the fundamental frequency of the notes of a musical scale are logarithmically spaced and are often grouped in octaves. The frequencies of an FFT are linearly spaced, which proves to be a poor match for audio signals. Since the hearing range of

the human auditory system is 20Hz to 20kHz, which contains ten octaves, the distribution of frequencies for an FFT is highly unbalanced. For example, a 1024 point FFT applied to a 44.1kHz-sampled signal would produce 256 frequency points in the highest octave but only one in the lowest two octaves combined. In contrast, the harmonics of a musical signal are linearly spaced, which corresponds fairly well to an FFT.

The above discussion is important because frequency analysis and the FFT are part of the family known as digital signal processing. Any type of numerical processing coupled with any discrete-time signal, whether it is audio, biomedical, radar, image, or otherwise, constitutes digital signal processing. Several DSP techniques have been applied to the source separation problem with limited success, including higher order statistics [7], beamforming [8], neural networks [9], decorrelation [10], and adaptive noise cancellation [11]. Current research using these methods will be described in detail in the following chapter. Most of these techniques assume only that the signals to be separated are statistically independent and zero mean. In addition, these techniques often require at least as many observations of the mixed sources as there are sources, and some are limited to the linear-mixture problem. There is rarely any mention of solving the problem of more sources than observations in the above techniques.

6.6 Psychoacoustics

One last issue of importance for the audio source separation problem is the human auditory system, the science of which is known as psychoacoustics. The previous discussion focuses on the issue of separating source signals from a mathematical perspective. However, the ultimate measure of whether or not several audio signals are separated is to listen to the separated signals.

If the human auditory system is incapable of detecting the presence of any other source during the playback of one separated source, then the sources are effectively separated.

The idea that psychoacoustics would play a role in the separation of signals may seem somewhat odd; if the original source signals are perfectly restored after separation, then each source would by default "sound" separated since they *are* separated. However, since there currently is no perfect separation algorithm, there will be errors such as crosstalk in the separated signals. Using psychoacoustic principles during the separation process may not make the signals any more separated mathematically, but they may "sound" more separated, which is significant for an audio source separation system. The following sections will discuss the properties of the human auditory system which may be used to aid in the audible separation of sources.

6.6.1 Masking

Masking is a phenomenon which occurs when certain sound events cannot be perceived in the presence of other sound events. There are two types of masking: frequency-domain masking and time-domain masking. Frequency-domain masking can occur when two frequency components of a sound are relatively close together in frequency but one component is larger in amplitude. The larger amplitude component would cause the smaller amplitude component to be inaudible, or masked. For a signal separation system, the sound source which contains the larger amplitude component could include both components in the estimate of the original source and there would be no audible difference.

Time-domain masking is a function of the overall amplitude and duration of a signal. A relatively loud sound source will cause a softer sound source to be inaudible if the sources occur simultaneously. If the duration of the softer source is extended both before and after the ends of the loud source, masking will still occur to a certain extent. The effect of time-domain masking on signal separation is similar to the effects of frequency-domain masking. When the louder source is the desired output of the signals separation system, the presence of other softer sources within the time mask of the loud source is inaudible.

6.6.2 Equal Loudness Curve

One of the most important aspects of the human auditory system is the difference between loudness and amplitude. Where amplitude is a physical quantity, loudness is a perceived quantity. Loudness is certainly a function of the amplitude of a source, but it is also a function of the frequency of the source material. For example, two sine waves of equal amplitude but different frequencies (1kHz and 50Hz, for instance) will be perceived as having very different loudness levels. In fact, there is a range of amplitudes at which the 50Hz tone would be inaudible but the 1kHz tone would still be perceived.

The equal loudness curve does not maintain the same shape with increasing loudness. This creates an unusual difficulty for a signal separation system. Although the two-channel mix never changes, the perception of the sources in the mix changes according to the level at which it is listened. This fact makes it difficult to create a single system which could take full advantage of the psychoacoustic properties of the human auditory system. At best, the general shape of the equal loudness curves for the typical listening levels can be used in the signal separation system.

Chapter 7 - Related Research

7.1 Adaptive Noise Cancellation

One of the classical examples of a signal separation system is the adaptive noise cancellation system [10]. Figure 28 shows an adaptive noise canceller. The purpose of the system is to separate a signal from a noise-corrupted observation of the signal. The adaptive noise canceller shown receives two channels as input, the first, $x_1(t)$, being the desired signal with noise added and the other, $x_2(t)$, being noise only. The output of the system is the difference between the signal-plus-noise channel and the noise-only channel. However, the overall level of the noise-only channel can be changed to minimize the noise in the output of the system.

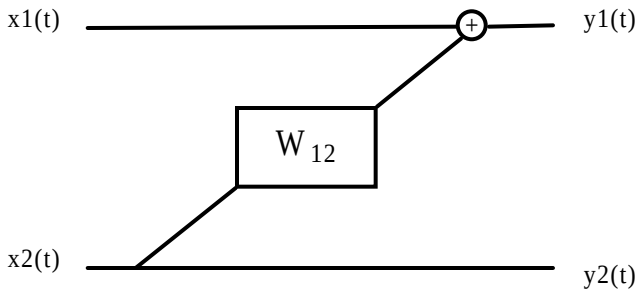

Figure 28 – Adaptive Noise Canceller (Adapted from Chan, 1996)

The adaptive noise canceller as described works best when the noise present in both channels is identical. Obtaining two channels with these properties in a practical situation is rather difficult. For example, if a lecturer is speaking in a room with a noisy air conditioner, microphones placed close to both the lecturer and the air conditioner would provide the best inputs to an adaptive noise canceller. However, there is no guarantee that the air conditioner noise received by the lecturer's microphone would exactly match the noise received by the air conditioner's microphone. Effects such as time delay and room reverberation could alter the time-domain

pattern of the air conditioner noise in the lecturer's microphone, thus degrading the performance of the adaptive noise cancellation system.

An alternative method to performing a simple difference on the time-domain signals is to create an adaptive noise canceller which works in the frequency domain [11]. Working in the frequency domain helps reduce the adverse effects of time delay and reverberation. Although a difference operation is still performed between the two input channels, the elements being subtracted represent energy in the frequency domain rather than amplitude in the time domain.

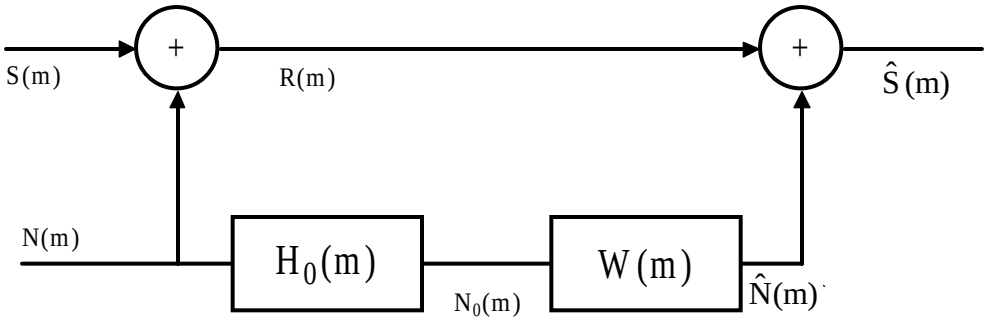

Figure 29 – Frequency domain adaptive noise canceller. (Adapted from Lindquist, 1989)

The advantage to an adaptive noise cancellation system is its relatively low computational complexity. However, it is only a successful signal separator if certain conditions are met on the input signals. For instance, the system is only meant to work in an environment where the desired signal and noise are present - in essence, two sources. In addition, one of the inputs must be an isolated source, which can be difficult to obtain in a real-world environment. Lastly, the noise present in both input channels must be highly correlated in order for the system to be effective.

It is obviously important to this research to evaluate the potential of an adaptive noise cancellation system as a solution to the signal separation problem for two-channel recordings.

Considering the conditions which are best suited for an adaptive noise canceller, a recording would have to have no more that two sources present and one of the sources would have to be isolated. From a mixing point of view, one source must be panned hard left or hard right and the other source must be panned to any other location. Effects such as time delay and reverberation would degrade the output quality of the adaptive noise cancellation system.

Overall, the adaptive noise cancellation system is a very basic method for separating signals. With respect to two-channel recordings, the adaptive noise canceller would fail for any recordings with more than two sources or where any one source could not be isolated. This situation is rarely encountered for two-channel recordings, making adaptive noise cancellation a poor fit for this research.

7.2 Decorrelation

Unlike the adaptive noise cancellation method, decorrelation is a method which uses properties of the observed signals to aid in their separation. The decorrelation method is based on calculating the correlation between the observed signals and minimizing that correlation. The correlation between two signals is simply the sum of the product of two signals. The correlation for two identical sources produces a large positive number while the correlation for two sources out of phase produces a large negative number.

Correlation does not have to be computed for two signals exactly aligned in time. A time delay between two sources in phase would create two sources out of phase. The effect of this delay on the correlation is quite extreme. For this reason, it is quite common to compute the correlation between two signals at many different time offsets. In terms of audio recordings, calculating

correlation at multiple time offsets is favorable because it responds well to effects such as time delay between channels and reverberation.

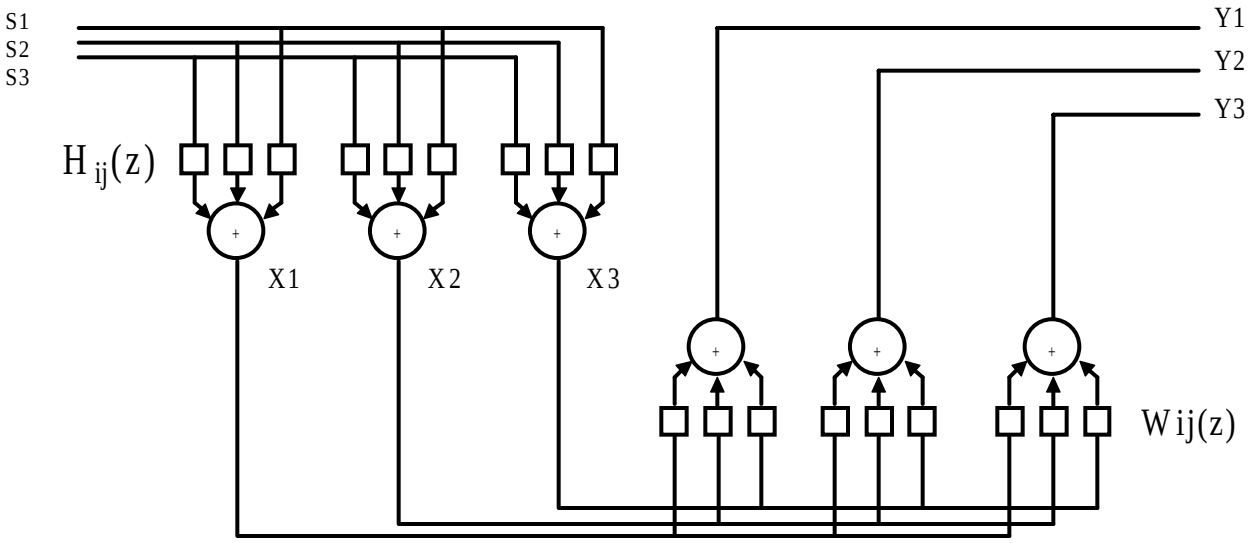

Figure 30 – Decorrelation signal separation system (Adapted from Chan, 1996)

An example of a signal separation system using decorrelation as described by Chan *et al* is shown in Figure 30 [10]. This figure shows a system with three sources (S_1, S_2, S_3) and three observations of the sources (X_1, X_2, X_3). There are nine transfer functions (H_{11} through H_{33}) which model the path between each source and each observation. For audio recordings, these transfer functions would describe any of the possible effects listed in Section 6.3 including panning, reverberation, and time delay. The separation system is represented by transfer functions W_{11} through W_{33}. Proper summation of the transfer function outputs produces the system's estimate for each of the original sources.

There are several properties of the decorrelation method which are important to this research. First, the system shown in Figure 30 is a three-input, three-output system. In general, the decorrelation method is an *n*-by-*n* system, meaning the number of sources must equal the number of observations of the sources for the system to be effective. Also, the inclusion of transfer

functions H_{11} through H_{33} allows for both linear- and convolutive-mixture problems (see Section 6.3.4).

One disadvantage of the decorrelation method is that the number of sources and the number of observations must be equal. With regard to this research, the decorrelation method would only be applicable if there were only two sources present in the two-channel recording. However, this method would perform better than the adaptive noise cancellation method described in the previous section since it is designed to work with both linear- and convolutive-mixture conditions. It should be noted that Chan *et al* [10] claim to have limited success applying the decorrelation method to mixtures with more sources than observations, especially when the mixing system is of low complexity and the number of sources is one greater than the number of observations.

Overall, the decorrelation method provides a better method for separating sources and will work with linear- and convolutive-mixture problems, but it is not a robust solution for mixtures with more sources than observations.

7.3 Beamforming

Beamforming is a technique which uses an array of sensors to measure a sound field in air [8]. The output of each sensor enters a transfer function which delays each signal by a selectable amount. The output of a beamforming system is the summation of the delayed sensor outputs. The principle behind beamforming is that a sound radiating from one point in space would reach each sensor at a different time since propagation delay is a function of distance. In order to emphasize that source, the delay times are chosen so each delay output would be in phase with

each other. If other sources are present, they would theoretically have random phase after the delays and would cancel.

Beamforming is an effective way to separate signals which originate from unique positions in space. However, a beamformer works best when there is a large number of sensors or observations of the signal. With regard to this research, we are limited to two observations of the sources, thus making beamforming a poor solution to the two-channel audio separation problem.

7.4 Higher Order Statistics and Independent Component Analysis

Two of the increasingly important fields in signal processing are Higher Order Statistics and Independent Component Analysis [7]. The two fields are separate but are used together in a blind signal separation system. Higher Order Statistics extract information from a signal which are not attainable through methods such as correlation and power spectrum, which are both second-order statistics of a signal. In general, signal separators using Higher Order Statistics will use third-order and fourth-order statistics to determine the characteristics of the signal.

Higher Order Statistics have two distinct advantages over second-order statistics such as the power spectrum. First, second-order statistics contain no phase information. Phase information provides additional cues about the characteristics of signals present in a mix, yet this information is often lost or discarded for most signal processing systems. Also, third- and fourth-order statistics are immune to Gaussian signals such as Gaussian white noise. Gaussian signal immunity is a valuable characteristic for processing audio since it is common to have noise present and it is very rare for a Gaussian signal to be placed intentionally in a recording.

Chapter 8 - Applications of Adaptive Time-Frequency Distribution

This chapter discusses the application of the adaptive time-frequency distribution to several audio signal processing problems: blind signal separation, lossy audio compression, and time stretching. The algorithms used in each case are discussed as well as the results of the operation.

8.1 Blind Signal Separation

An algorithm was developed to perform blind signal separation on a two-channel audio recording using the adaptive time-frequency distribution. To test the algorithm a two-channel mix was created using three source instruments: electric guitar, acoustic guitar, and piano. Each instrument was recorded during the performance of a song in a studio. All instruments were acoustically isolated from each other and were recorded using one microphone per instrument. The combined instrument sounds were mixed to be part of a song and were not just combinations of random, unrelated performances. For the purpose of analysis a four-second segment of the song was used for the signal separation algorithm.

The only method used to combine the instrument sounds was panning. Therefore, the signal content in each of the two channels was a linear combination of the three sources. The amount of gain between each instrument and each channel was determined by the panning angle applied and the desired level of each instrument. For the song segment selected the electric guitar was the dominant instrument and the remaining instruments were playing background parts. The order of the instruments from loudest to softest was as follows: electric guitar, acoustic guitar, and piano. In addition, the panning angle for each instrument was determined by the arbitrary

selection of channel ratios as listed in Table 4. A panning angle of 0° represents signal in the left channel only and 90° represents signal in the right channel only.

Table 4 – Panning angles applied to each instrument

Instrument	Panning Angle	Ratio of Left Channel Level to Right Channel Level
Acoustic Guitar	18.43°	3:1
Piano	45°	1:1
Electric Guitar	71.57°	1:3

Note this signal separation problem is based on a three-by-two mixture, which means there are three source instruments and two combinations of those instruments. Most of the signal separation methods discussed in the previous chapter would be incapable of separating more than two instruments from a two-channel recording.

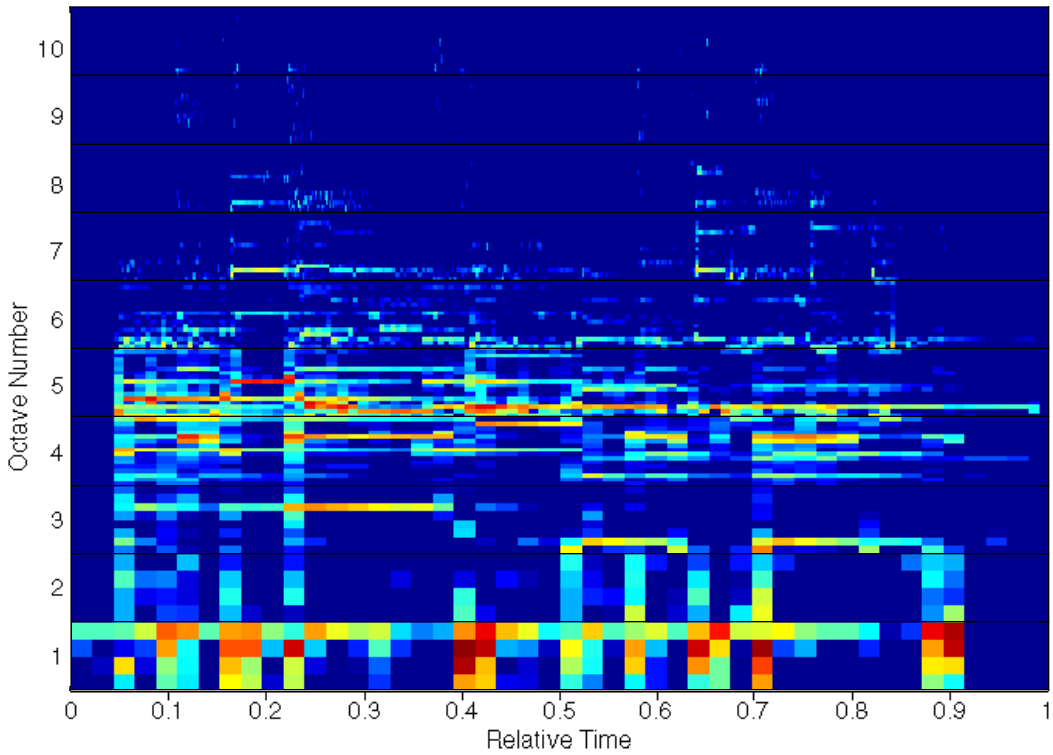

Figure 31 – Distribution of left channel of two-channel mix after adaptation

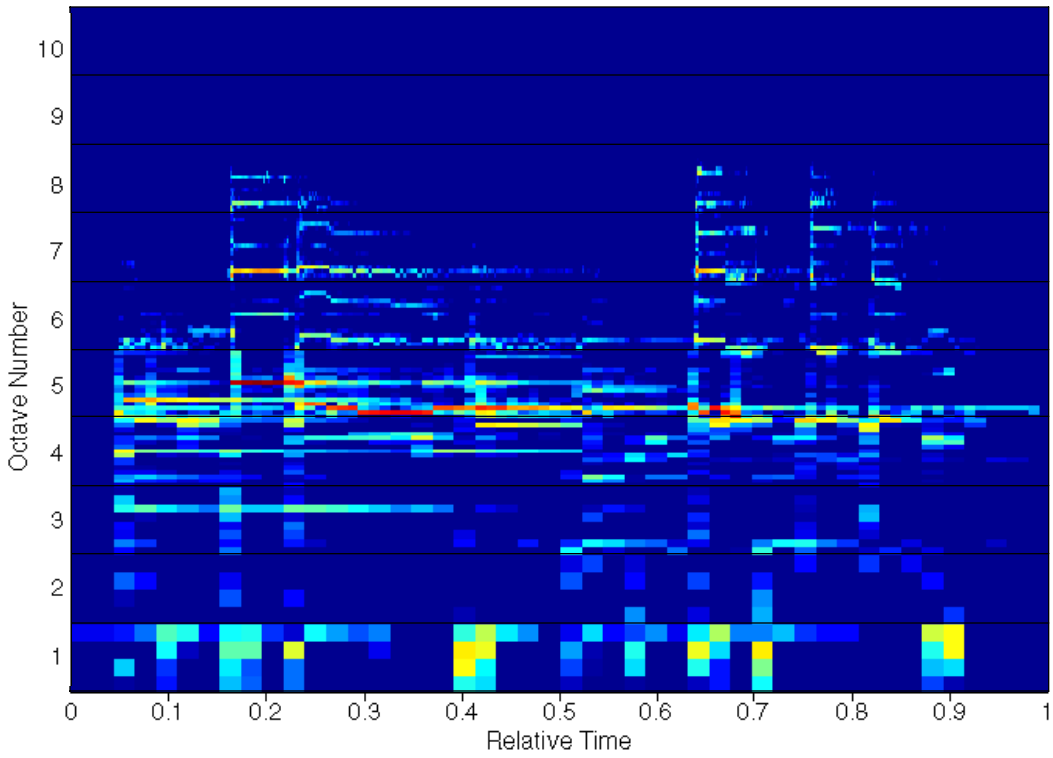

Figure 32 – Distribution of right channel of two-channel mix after adaptation

Figures 31 and 32 show the adapted time-frequency distributions of the left and right channels of the mix, respectively. The basis of the separation algorithm is to compare the energy level of corresponding time-frequency points between channels. The reasoning for this is as follows: if a time frequency point contains energy from only one source, then the energy ratio of the left channel point to the right channel point should be identical to the ratio used in the mixing process (see Table 4). If this is true then all time-frequency points associated with one source will have the same ratio between channels. These points would then be added to a new time-frequency distribution with no energy. The inverse transform of the new distribution would be computed and the result should be one source instrument isolated from all others.

Several assumptions are required to implement the signal separation algorithm. First, the number of sources must be known in advance. Second, the mixing method must be panning. Third, the position of each source in the mix must be known. All of these assumptions conflict with the idea of blind signal separation, where none of the above information is known. However, the number of sources could be determined through listening tests. Also, the position of each source could be determined by evaluating the ratios of all time-frequency points for regions of similarity. If most of the time-frequency points have ratios which fall into three general groups, then there are likely three sources and the positions can be determined by averaging the ratios within each group.

To perform a comparison of identical time-frequency points from two distributions the distributions must analyze exactly the same frequencies in every time-frequency block. This is done by adapting to one of the two channels and applying all of the frequencies used for analysis to the other channel. Not only does this guarantee that the time-frequency points are properly matched, but the computational time of the distribution for the second channel is reduced significantly.

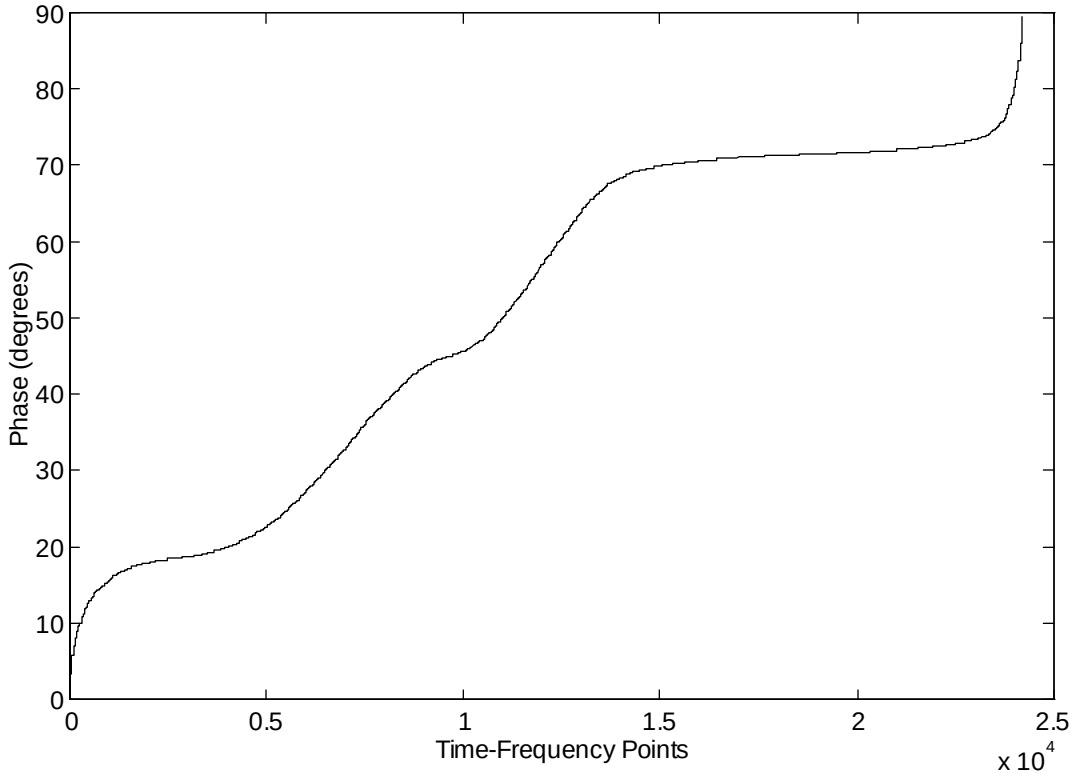

Figure 33 – Angle associated with corresponding time-frequency points

Figure 33 shows the angle for time-frequency points where most of the signal energy is contained. The angle is used in this plot rather than energy ration in order to correspond to the panning angles listed in Table 4; the inverse tangent of the ratio provides the angle. The plot shows that many of the time-frequency points with significant energy have a panning angle near 72°, which is the panning angle of the electric guitar signal. There is also a small gathering of points around the other two panning angles associated with sources. However, the lower the amplitude of the signal, the fewer the number of points which exactly match the panning angle. Since the panning angles of the time-frequency points do not exactly match the three source angles a range of angles is associated with each source.

The signal separation algorithm was applied to the mix shown in Figures 31 and 32. The results are shown in Figure 34. Figure 34c shows the separated signal and Figure 34d shows the difference between the original signal and the separated signal. When listening to the separated signal, the electric guitar sound is clearly dominant. There are also audible artifacts which somewhat distort the sound of the signal. When listening to the difference signal it is clear the artifacts are caused by the inclusion of time-frequency energy belonging to other instruments. In fact some time-frequency energy from the electric guitar signal is omitted from the separated signal. The overall effect on the separated signal is a sort of "bubbling" sound where elements of other instruments are added in very short but rapid bursts.

The problem with using only the panning angle to separate sources is caused by time-frequency points containing energy from more than one source. This method of signal separation does not attempt to split energy from one time-frequency point with more than one source. For multiple sources in one time-frequency point, the calculated panning angle may not be associated with the proper sources. This is the cause of the distortion in the separated signal.

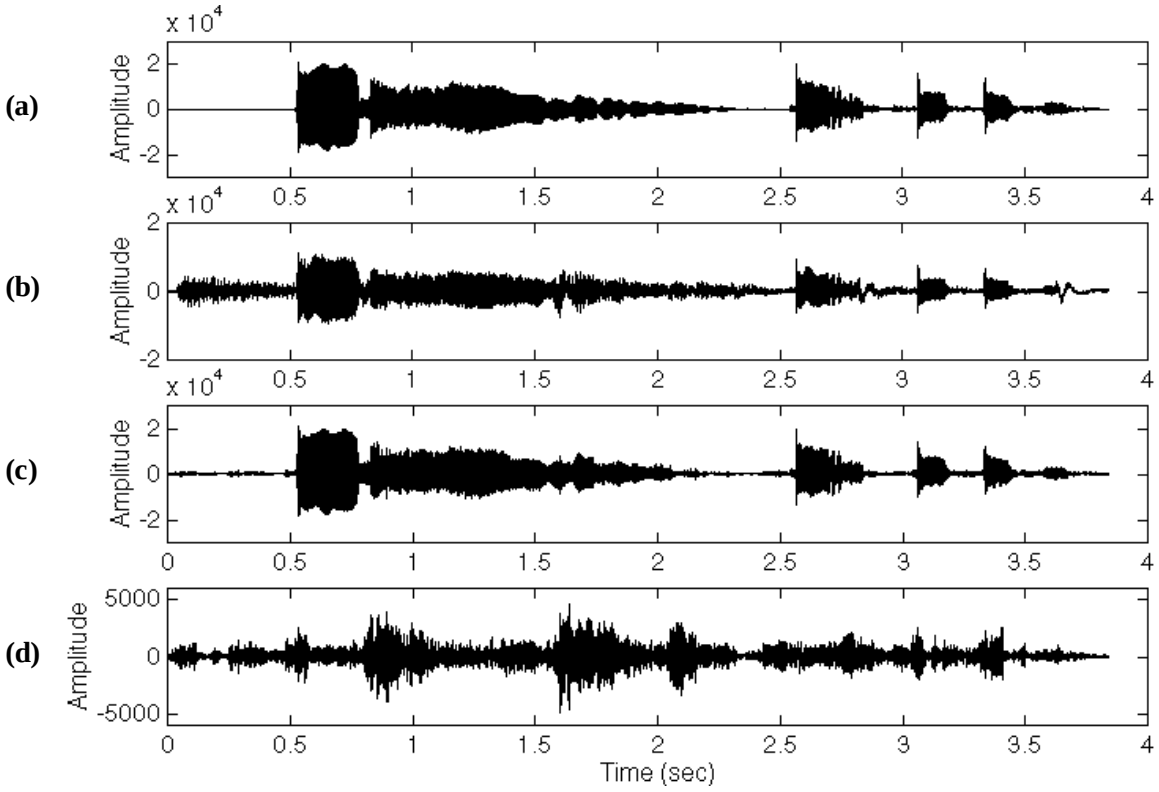

Figure 34 – Results of audio signal separation algorithm. In (a), the original electric guitar signal is shown. Figure (b) shows the right channel of the three-instrument mix. Figure (c) shows the separated guitar signal. Figure (d) shows the difference between the original signal (a) and the separated signal (c).

8.2 Lossy Audio Compression

Another application of the adaptive time-frequency distribution is the lossy compression of audio signals. A simple form of compression would involve adapting to a signal and storing only the time-frequency points which contain the majority of the signal energy. Upon reconstruction the time-frequency blocks are formed using the stored time-frequency points and setting the remaining points to zero. The adaptive time-frequency distribution is a good tool for audio compression since the signal is broken into bands before being analyzed. Other compression methods such as MPEG use a similar technique for audio compression.

The lossy compression of audio signals was performed on the same sound sources used in the signal separation example. Several steps are used when computing the compression algorithm. After the adaptive time-frequency distribution is computed for the signal to be compressed, the time-frequency points within each time-frequency block are sorted in order of energy level. For time-frequency blocks with sixteen frequencies, only the three points with the most energy in the block are stored. In the case of eight frequencies per block, only two frequencies are kept; for four frequencies, only one is kept. In terms of the number of time-frequency points kept, the compression ratio is better than 5:1. However, if a time-frequency point was adapted, the associated frequency value must also be stored, which reduces the compression ratio to about 4:1. The results of the compression sounded similar to the separated signals from the previous section, but with less distortion.

8.3 Pitch Shifting and Time Scaling

Pitch shifting and time scaling algorithms can take advantage of the flexibility of the adaptive time-frequency distribution. A signal which has been accurately localized by the distribution can be pitch shifted by simply increasing the frequency values for the time-frequency points with significant energy. An allpass filter would also be needed in order to compensate for phase distortions caused by the shifting of the time-frequency points. Time scaling can be accomplished by adding or removing frequency bins without adjusting any of the existing time-frequency points. Upon application of the inverse transform the signal will have either more or fewer time samples derived from the proper spectrum.

A time scaling algorithm was developed for use on the audio signal processed in the previous sections of this chapter. The purpose of the time scaling algorithm is to double the length of the signal without changing the pitch. First, the adaptive time-frequency distribution is computed for the signal. Second, each time-frequency block is padded with extra time-frequency points containing no energy in order to double the total number of points. Referring to Figure 11, the frequencies for the new frequency bins were chosen by using the same value of d_2 as the current block but the value of d_1 for the lowest frequency was cut in half. This essentially causes the new points to be interleaved with the existing points. Third, perform the inverse of the distribution using the new frequency points. The new signal is double the length of the original signal. The sound quality of this algorithm was fair and not quite comparable to other currently existing methods.

Chapter 9 - Further Research

This chapter provides details for the improvements needed for the adaptive time-frequency distribution algorithm and the areas for additional research.

9.1 Object-Oriented Model

One of the most valuable additions to the adaptive time-frequency distribution algorithm would be the application of an object-oriented model. The following two subsections discuss the added benefits and flexibility an object-oriented system would provide.

9.1.1 Variable-sized Time-Frequency Blocks

The ability to adjust the size of each time-frequency block is as important as being able to adapt to the frequencies within a block. In the same way that the adaptation of frequencies improves the frequency resolution of the distribution; the adaptation of the block size improves resolution in the time domain.

The first example of how the distribution would improve with adaptive time blocks is illustrated with an impulse signal. The impulse signal is narrow in the time domain in the same way that a sine wave is narrow in the frequency domain. One of the goals of any signal analysis system is to localize information present in the signal, so the ability to accurately localize a short time-domain signal is very important. In the current adaptive distribution algorithm, if an impulse signal occurs in the middle of a time block, all the frequencies of the transform attempt to represent the impulse signal. The result is a time-frequency block with wideband energy. If the algorithm was improved to include the ability to adapt the block size, the normal-sized block could be divided

into three sections: all the samples leading up to the impulse, the impulse plus a minimum of neighboring samples, and all the samples following the impulse. Instead of having one block containing wideband energy, two of three smaller blocks would have no energy and a very small block would have wide band energy, perhaps using as few as two frequencies for analysis. The information obtained in the latter case indicates signal energy for a very short amount of time. Frequency information was exchanged for time information, and since a wideband spectrum provides little frequency information, the tradeoff is advantageous.

A second example of improving the information provided by the distribution relates to the start time of a given signal. If the first two-thirds of a time block contain silence and the last third contains the start of a sinusoid, the frequencies of the transform attempt to model the entire event. If the original time block was divided into a block containing only silence and a block containing the sinusoid, each block will accurately describe the signal.

A third use for adaptive block size is for audio compression. If a sinusoid of constant frequency and duration of one second is analyzed, the result will be many time-frequency blocks representing small pieces of the sinusoid. If all of the blocks containing the sinusoid energy were replaced with one large block, the information conveyed would be the same. In terms of audio compression, the one-second sinusoid could be analyzed with a time block of length one second. Since only one of the many frequency bins will have any energy, the remaining bins can be ignored and only two coefficients and one frequency value would need to be stored to reconstruct the sinusoid. There is no more compact way to represent a sinusoid than with values for frequency, phase, and magnitude.

9.1.2 Compact frequency data storage

Along with the benefits of adaptive block size, an object-oriented model also provides the benefit of reducing the storage requirements for maintaining all time-frequency points which have been adapted. The model would be implemented by creating a time-frequency block object. Each object could have any number of frequencies and any frequency values. The current algorithm stores all the time-frequency coefficients in one matrix and the corresponding frequency for each coefficient in another matrix. Using the object-oriented model, a linked list of objects would be used to hold one octave of information. The object would contain all the coefficients for the time-frequency block but only the frequency values which had been adapted. Since the current algorithm limits the number of adaptations per time-frequency block to one-fourth the total number of frequencies, the absolute maximum number of adapted frequencies to be stored is one-fourth that of the original algorithm.

9.2 Information Sharing

The current algorithm adapts to the frequency content of one block based only on the contents of that block. For the audio signals analyzed in this research, there were many instances where several frequencies in different octaves were all the product of the same source. It would be advantageous to use information from other time-frequency blocks to determine the content of the current block. For example, if a sinusoidal signal was detected in several previous blocks, chances are good that the same sinusoid has continued into the current block. Instead of blindly selecting the peak frequency bin and adapting ten times, the algorithm could start with the frequencies from the previous block. Also, the algorithm could search for harmonic components in other octaves to estimate the signal content in the current block.

9.3 Alternate Analysis Window

The current algorithm divides the input into frequency bands by filtering and downsampling. One reason for downsampling is to force all the frequency content to be evenly distributed between 0 and π. Without downsampling, the frequency content is between $\frac{\pi}{2}$ and π. If the Fourier transform is used to analyze a signal with frequency content limited to half of the total bandwidth, then only half of the frequency bins would contain useful information. With the non-orthogonal transform, the frequencies do not necessarily need to be evenly distributed between 0 and π. If the filtered signal was not downsampled but was analyzed with a non-orthogonal transform with shifted frequencies, the time-frequency resolution would be doubled and there would be no aliasing.

A problem occurs when the frequencies of the non-orthogonal transform are not evenly distributed. The transform will only produce useful results when the set of frequencies extends beyond both the lowest and highest frequencies of the band to be analyzed. Although there may be unforeseen disadvantages, the potential benefits of a shifted frequency set are worth consideration.

9.4 Improving Adaptation

Several improvements to the adaptation algorithm could be considered. First, instead of using the average of the current frequency and an adjacent frequency for selecting the new peak frequency, the energy leakage value could be used to obtain a better estimate of the exact frequency of the signal. For example, if the energy leakage decreases by a significant amount after one iteration, a formula could be used to calculate the best estimate for the step size.

Second, if the energy leakage after an iteration is below an absolute threshold, it could be assumed that the frequency chosen is well-matched to the signal. Third, instead of calculating the energy leakage for a very small move in only one direction, the transform could be computed for a very small move in both directions and a move could be made in the direction of the least amount of energy leakage.

9.5 Adaptation Speed

One of the disadvantages of using the adaptation capabilities of the time-frequency distribution is the long computation time. Using the improved adaptation algorithm, the computation time for the adaptive time-frequency distribution ranges from one to five minutes for every one second of audio. The largest bottleneck in the adaptation algorithm is the need to perform twenty matrix inverses for each frequency adapted. For each iteration, one inverse is computed when a frequency is moved slightly in one direction and one inverse is computed when the frequency makes a definite move. The unique aspect of the matrix inverses is for each set of twenty matrices, all but two of the columns of the matrices to be inverted remain constant. The two columns which change correspond to the one frequency that was changed. If an algorithm could be implemented which takes advantage of this redundancy, the computation speed of the adaptation would be greatly increased.

9.6 AM and FM Component Tracking

If the intended purpose of the adaptive time-frequency distribution is for analysis only, additional processing could be used to obtain accurate information on the amplitude- and frequency-modulation components of a sinusoid. This would be potentially very useful for audio signals since the timbre of an instrument is mostly defined by the AM and FM components of the

fundamental frequency and associated harmonics. Research performed by one of the authors of the non-orthogonal signal decomposition method provides details on using their decomposition method for AM and FM component tracking. [12] When only one sinusoidal component is being analyzed, an analysis block containing only two frequencies could be used. This provides excellent time-frequency resolution since only four time samples are needed to perform the analysis. In addition, frequency resolution is not lost since the frequencies are adaptive and only one frequency component exists. The short time block also minimizes the chance that the signal frequency would change within the block.

To apply this method to the adaptive time-frequency distribution, a specific sinusoidal component would have to be located in the distribution. After determining the time duration of the component, a bandpass filter could be applied to the original signal to obtain the best time-frequency resolution. The non-orthogonal transform with only two frequencies would then be used on the filtered signal. This could be repeated for all sinusoidal components which are isolated in the time-frequency distribution. In other words, no other signal components could exist at the same time and in the same frequency region.

Chapter 10 – Conclusion

The overall results of this research are very positive. A time-frequency distribution is developed which not only analyzes the signal content, but can be used as a processing block for solving signal processing problems. The time-frequency points of the distribution can be distributed in a way which best suits the signal being analyzed. In addition, the frequencies used in the analysis of the signal can be adapted to the signal. This results in a distribution with more energy contained in fewer time-frequency points – the components of the signal become more localized.

The adaptive time-frequency distribution has been optimized for processing audio signals. The signal is broken up into octaves to provide a more logarithmic distribution of frequencies. Also, the signal is decomposed using sinusoids rather than wavelets or other bases. The size of the distribution is on the order of the size of the signal being analyzed and the computation time without adaptation is on the order of a Discrete Fourier Transform. The filter banks used to break the signal into octaves can be made up of perfect reconstruction filters if the distribution is used as a processing block in a system. If the distribution is used for analysis only then filters with narrower transition bands and better frequency response can be selected.

There are several disadvantages of using the adaptive time-frequency distribution for signal analysis and processing. First, the speed of the adaptation algorithm is slow and unusable in a real-time system. Second, for systems where perfect reconstruction is needed, the frequency response of the analysis filters allows signal content from adjacent octaves to affect the results of

the distribution. Third, the adaptation of frequencies is not well suited to signals which are wideband. Adaptation to wideband signals may provide poor or misleading results.

The adaptive time-frequency distribution was used as the main processing block to solve several audio signal processing problems and the results were promising. The signal separation, lossy audio compression, and time stretching algorithms were simplistic yet effective. The signal separation system was successful when there were more source instruments than observations of the sources. The separation algorithm was only tested for a simple mixture of signals; more complex mixtures would be more difficult to process. The separation system produced signals which were dominated by the desired source signal but audible artifacts existed. One of the biggest problems for the separation system is being able to separate energy within one time-frequency point caused by more than one source.

Groundwork has been laid for a powerful signal processing tool, and suggestions for further research are highlighted in the previous chapter.

Appendix – Coefficients for Analysis and Synthesis Filters

H0		H1		F0		F1	
-0.000036	0.199040	-0.000087	0.480526	-0.000087	0.480526	0.000036	-0.199040
0.000044	-0.153081	0.000106	-0.369571	-0.000106	0.369571	0.000044	-0.153081
0.000005	-0.195913	-0.000002	0.081150	-0.000002	0.081150	-0.000005	0.195913
-0.000127	-0.030047	0.000053	0.012446	-0.000053	-0.012446	-0.000127	-0.030047
-0.000060	0.029613	-0.000145	0.071492	-0.000145	0.071492	0.000060	-0.029613
0.000012	-0.037822	0.000028	-0.091310	-0.000028	0.091310	0.000012	-0.037822
-0.000040	-0.052328	0.000017	0.021675	0.000017	0.021675	0.000040	0.052328
0.000129	0.002472	-0.000053	-0.001024	0.000053	0.001024	0.000129	0.002472
0.000165	0.015250	0.000399	0.036817	0.000399	0.036817	-0.000165	-0.015250
-0.000105	-0.015762	-0.000253	-0.038053	0.000253	0.038053	-0.000105	-0.015762
0.000704	-0.016672	-0.000292	0.006906	-0.000292	0.006906	-0.000704	0.016672
0.002003	0.007092	-0.000830	-0.002937	0.000830	0.002937	0.002003	0.007092
0.000904	0.007599	0.002182	0.018346	0.002182	0.018346	-0.000904	-0.007599
0.000049	-0.006118	0.000119	-0.014769	-0.000119	0.014769	0.000049	-0.006118
0.004381	-0.003853	-0.001815	0.001596	-0.001815	0.001596	-0.004381	0.003853
0.007417	0.005094	-0.003072	-0.002110	0.003072	0.002110	0.007417	0.005094
0.002110	0.003072	0.005094	0.007417	0.005094	0.007417	-0.002110	-0.003072
0.001596	-0.001815	0.003853	-0.004381	-0.003853	0.004381	0.001596	-0.001815
0.014769	-0.000119	-0.006118	0.000049	-0.006118	0.000049	-0.014769	0.000119
0.018346	0.002182	-0.007599	-0.000904	0.007599	0.000904	0.018346	0.002182
0.002937	0.000830	0.007092	0.002003	0.007092	0.002003	-0.002937	-0.000830
0.006906	-0.000292	0.016672	-0.000704	-0.016672	0.000704	0.006906	-0.000292
0.038053	0.000253	-0.015762	-0.000105	-0.015762	-0.000105	-0.038053	-0.000253
0.036817	0.000399	-0.015250	-0.000165	0.015250	0.000165	0.036817	0.000399
0.001024	0.000053	0.002472	0.000129	0.002472	0.000129	-0.001024	-0.000053
0.021675	0.000017	0.052328	0.000040	-0.052328	-0.000040	0.021675	0.000017
0.091310	-0.000028	-0.037822	0.000012	-0.037822	0.000012	-0.091310	0.000028
0.071492	-0.000145	-0.029613	0.000060	0.029613	-0.000060	0.071492	-0.000145
-0.012446	-0.000053	-0.030047	-0.000127	-0.030047	-0.000127	0.012446	0.000053
0.081150	-0.000002	0.195913	-0.000005	-0.195913	0.000005	0.081150	-0.000002
0.369571	-0.000106	-0.153081	0.000044	-0.153081	0.000044	-0.369571	0.000106
0.480526	-0.000087	-0.199040	0.000036	0.199040	-0.000036	0.480526	-0.000087

References

[1] M. Skolnik, *Introduction to Radar Systems* (McGraw-Hill Book Co., 1980).
[2] L. Cohen, *Time-Frequency Analysis* (Prentice Hall, Englewood Cliffs, NJ, 1995).
[3] R. van der Heiden, "FEL-TNO report 92-B200", App. A., (1992).
[4] G. Strang, T. Nguyen, *Wavelets and Filter Banks* (Wellesley – Cambridge Press, Wellesley, MA, 1996).
[5] I. Dologlou, S. Bakamidis, G. Carayannis, "Signal decomposition in terms of non-orthogonal sinusoidal bases," *Signal Processing*, vol. 51, pp. 79-91 (1996).
[6] M. Rossi, J. Zhang, W. Steenaart, "Iterative Constrained Least Squares Design of Near Perfect Reconstruction Pseudo QMF Banks", in *Proc. CCECE'96*, Calgary (1996).
[7] B. Laheld, J. Cardoso, "Adaptive source separation with uniform performance", in *EUSIPCO-94*, Edinburgh (1994).
[8] B. Van Veen, K. Buckley, "Beamforming: A Versatile Approach to Spatial Filtering", *IEEE ASSP*, pp. 4-24 (1988).
[9] Y. Deville, "A unified stability analysis of the Herault-Jutten source separation neural network", *Signal Processing*, vol. 51, pp. 229-233 (1996).
[10] D. Chan, S. Godsill, P. Rayner, "Multi-channel Multi-tap Signal Separation By Output Decorrelation", *CUED/F-INFENG/TR 250* (1996).
[11] C. Lindquist, *Adaptive & Digital Signal Processing* (Steward & Sons, Miami, FL, 1989).
[12] S. Bakamidis, "A New Frequency Domain Method for Amplitude and Frequency Demodulation," *ICSPAT*, Boston, MA (1996).

1 M. Skolnik, *Introduction to Radar Systems* (McGraw-Hill Book Co., 1980).
2 L. Cohen, *Time-Frequency Analysis* (Prentice Hall, Englewood Cliffs, NJ, 1995).
3 R. van der Heiden, "FEL-TNO report 92-B200", App. A., (1992).
4 G. Strang, T. Nguyen, *Wavelets and Filter Banks* (Wellesley – Cambridge Press, Wellesley, MA, 1996).
5 I. Dologlou, S. Bakamidis, G. Carayannis, "Signal decomposition in terms of non-orthogonal sinusoidal bases," *Signal Processing*, vol. 51, pp. 79-91 (1996).
6 M. Rossi, J. Zhang, W. Steenaart, "Iterative Constrained Least Squares Design of Near Perfect Reconstruction Pseudo QMF Banks", in *Proc. CCECE'96*, Calgary (1996).
7 B. Laheld, J. Cardoso, "Adaptive source separation with uniform performance", in *EUSIPCO-94*, Edinburgh (1994).
8 B. Van Veen, K. Buckley, "Beamforming: A Versatile Approach to Spatial Filtering", *IEEE ASSP*, pp. 4-24 (1988).
9 Y. Deville, "A unified stability analysis of the Herault-Jutten source separation neural network", *Signal Processing*, vol. 51, pp. 229-233 (1996).
10 D. Chan, S. Godsill, P. Rayner, "Multi-channel Multi-tap Signal Separation By Output Decorrelation", *CUED/F-INFENG/TR 250* (1996).
11 C. Lindquist, *Adaptive & Digital Signal Processing* (Steward & Sons, Miami, FL, 1989).
12 S. Bakamidis, "A New Frequency Domain Method for Amplitude and Frequency Demodulation," *ICSPAT*, Boston, MA (1996).