Implementing Multiple Methods of Time-Scale Modification in MATLAB

Allison Crim

Department of Electrical Engineering
Johns Hopkins University
Baltimore, Maryland 21218, USA

acrim3@jh.edu

Abstract **– Time-Scale Modification is an audio effect commonly used in audio signal processing. Time-scale modification is used to slow down and speed up audio signals without changing the pitch of the signal. This paper will explore various methods for implementing time-scale modification and compare their benefits and drawbacks. The goal of the project is to separate an audio signal, process it using the aforementioned methods, and put the audio back together.**

Index Terms – overlap-add, phase-vocoder, harmonic-percussive separation

I. Introduction

Time-scale modification (TSM) is a way to change the speed of a signal without changing the pitch of that signal. The task for this project was to understand and implement multiple methods of time-scale modification and compare their effectiveness. The three methods of TSM being implemented in this project are Overlap-Add, Phase Vocoder, and Harmonic Percussive Separation.

Exploring the various methods of TSM is important because audio signals are very complex and need to be broken down to be processed correctly. For example, in any given song there could be multiple harmonic signals such as violin and vocals plus percussive signals such as drums or castanets. These individual components of the song need to be treated differently because each signal has particular features that are most important to retaining their sound. For the harmonic signals, it is essential to keep the pitch the same before and after the signal is processed. For the percussive signals, the timing of the beats must be maintained. These conflicting needs require multiple methods of time scale modification. The following sections of the paper will cover how these methods were implemented in matlab and their pros and cons. The paper will also discuss harmonic-percussive separation and how to apply different time-scale modification methods to different portions of a song.

II. Overlap-Add Based TSM

The overlap-add method of TSM is computed in the time domain using the following steps. It is important to define that the parameter that the signal is stretched or compressed by is α which is the ratio of H_s (sysnthesis hopsize) over H_a (analysis hopsize).

To begin, the input signal is divided up into chunks of length N separated by H_a.

$$x_m(r) = \begin{cases} x(r + mH_a, & \text{if } r \in [-\frac{N}{2} : \frac{N}{2} - 1] \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

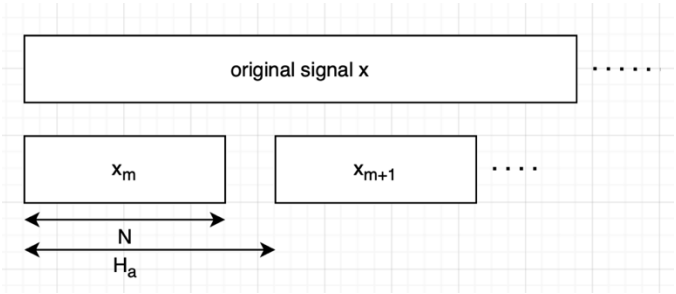

Figure 1: Break Original Signal into Analysis Frames

Next, each analysis frame is multiplied by a Hann window as defined in the Dreidger paper [2]. This Hann window removes gaps in the reconstructed signal that come from the overlap-add. The equation for the window is shown below.

$$w(r) = \begin{cases} 0.5\left(1 - \cos\left(\frac{2\pi\left(r + \frac{N}{2}\right)}{N - 1}\right)\right), if\ r \in \left[-\frac{N}{2} : \frac{N}{2}\right] \\ 0, \quad otherwise \end{cases} \quad (2)$$

The resulting Hann window is shown below.

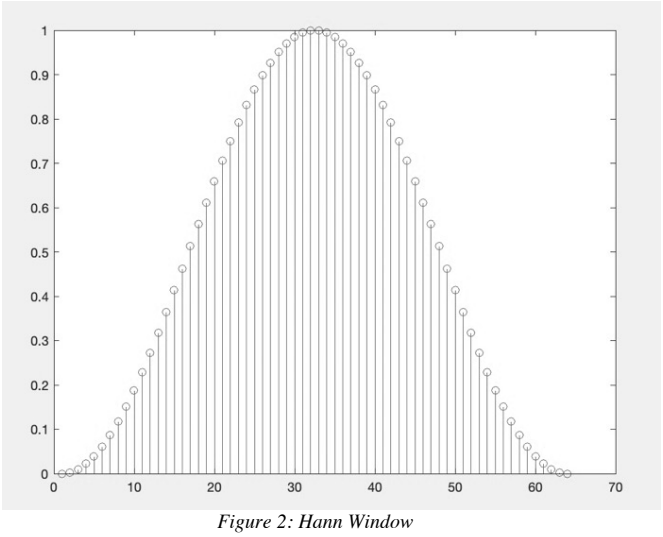

Figure 2: Hann Window

The next step is to relocate the analysis frames on the time axis with a spacing of H_s. Using the knowledge of the following property of Hann windows makes generating the synthesis frame significantly easier. To use the property, the synthesis hopsize must be chosen to be N/2.

$$\sum_{n\in\mathbb{Z}} w\left(r - n\frac{N}{2}\right) = 1 \qquad (3)$$

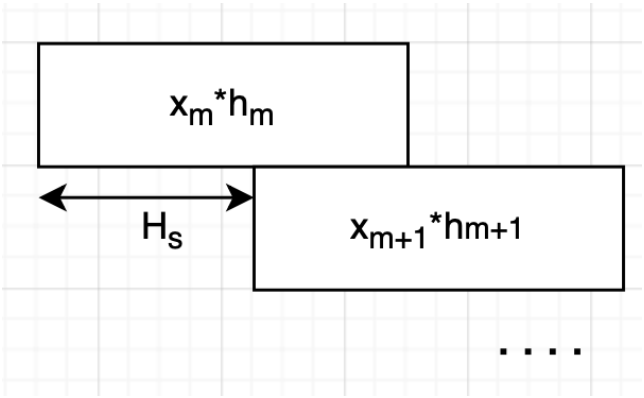

Figure 3: Windowing with Synthesis Hopsize H_s

The final step of OLA is to add all of the frames together.

II. Phase-Vocoder Based TSM

The next TSM method that was studied is called phase-vocoder. This method is computed in the frequency domain and is used to make phase estimates for each analysis frame. These estimations are used to avoid phase mismatch that is present in the OLA method.

The first step of the phase-vocoder method is the same as the overlap-add method. The input signal needs to be divided into analysis frames spaced out by H_a.

In order to transfer the signal into the frequency domain, the Fourier transform is taken. Once the signal is in the frequency domain, the phase vocoder process begins. The phase and magnitude of the signal is calculated. Next, the instantaneous frequency is calculated using the equation below. The symbol Ψ indicates a shift in the phase to be between -π to π.

$$F^{IF} = \omega \frac{\Psi(\Phi_2 - (\Phi_1 + \omega\Delta t)}{\Delta t} \qquad (4)$$

To calculate the modified phase shown in the oval in the image below, the instantaneous frequency is used.

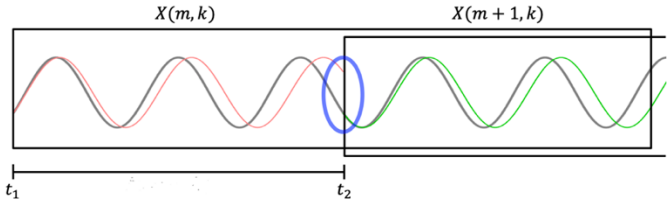

Figure 4: Modified Phase in Oval (Dreidger)

The equation below is used to calculate the modified phase.

$$\phi_{Mod}(m+1) = \phi_{Mod}(m) + F^{IF} * \frac{Hs}{Ha} \qquad (5)$$

The equation below is then used to find the modified signal in the frequency domain.

$$X^{Mod}(m) = |X(m)| * \exp(2\pi i * \phi^{Mod}(m)) \qquad (6)$$

The final steps of the phase-vocoder method are to apply a Hann window and take the inverse fft of each analysis frame. Finally, to reconstruct the signal, the analysis frames are added together in the same way as in overlap-add.

III. Results

The goal of the initial project was to compare OLA and phase-vocoder TSM when they are processing harmonic and percussive signals. The following results capture the effectiveness of each method and points out drawbacks. Because these results are captured in a paper rather than an audio/visual medium like a presentation, the input signals chosen were the simplest signals available to show the behavior.

A. Harmonic Results

The first portion of the analysis is focused on the impact of processing a harmonic signal with both the OLA and phase-vocoder method. The Dreidger paper states that "While OLA is unsuited for modifying audio signals with harmonic content, it delivers high quality results for purely percussive signals…"[2]. The image below is a zoomed in plot of a sine wave at 0.5x and 2x speed. The processed signals clearly did not maintain their periodic structure which is the common complaint for OLA. It is vital for harmonic signals to maintain their periodic structure to sound right.

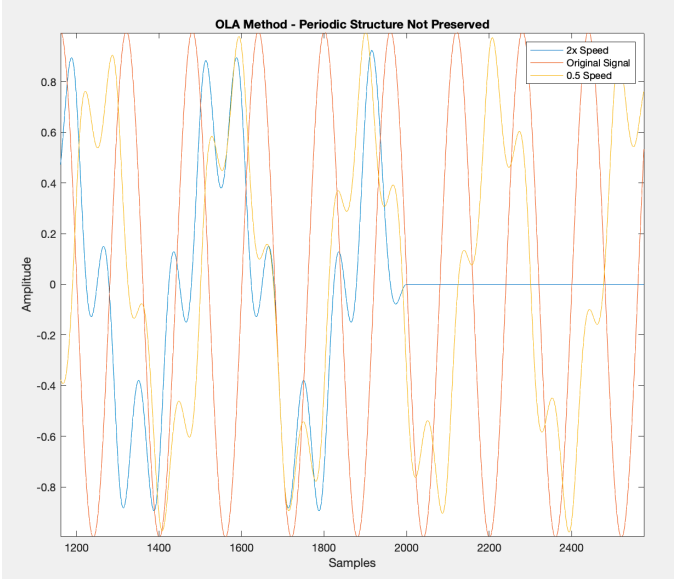

Figure 5: Harmonic Signal with OLA Implementation

Next, the harmonic signal was processed using the phase-vocoder method. The same sine wave was used, and the image below was produced. The phase-vocoder method maintains the

periodic structure and sounds significantly better than the OLA version, as expected. It is clear that there is an issue with the amplitude of the signal which is the cause of some minor reverberation.

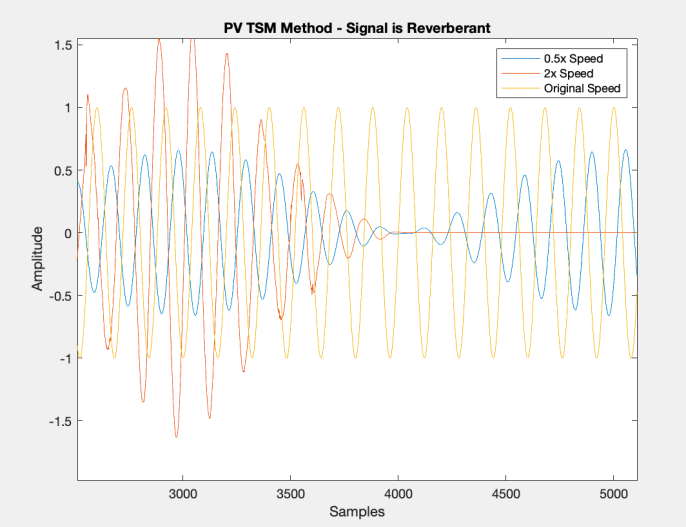

Figure 6: Harmonic Signal with OLA Implementation

B. Harmonic Results

While overlap-add is not great at processing harmonic signals, it is clearly superior at processing percussive signals. Because overlap-add takes place in the time domain, it manages to maintain the crisp beats of a drum signal as shown in the graph below.

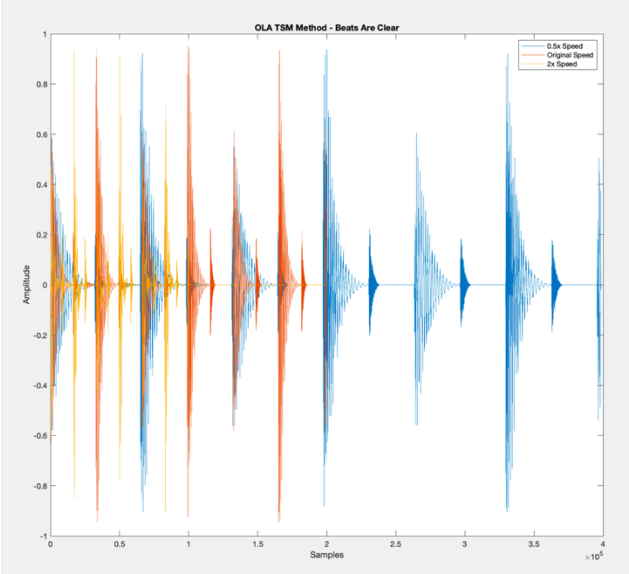

Figure 7: Harmonic Signal with OLA Implementation

Conversely, the phase-vocoder method of TSM is not as effective at processing percussive signals. According to Dreidger, "The loss of vertical phase coherence affects the time localization of events such as transients.". This means that the beats of the signal will not sound as crisp leading to a subpar result. The plot below illustrates the lack of precision of phase-vocoder TSM.

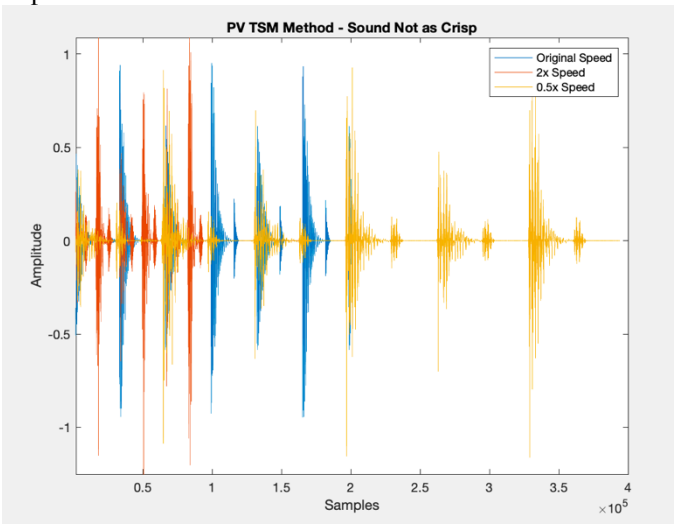

Figure 8: Harmonic Signal with OLA Implementation

IV. Harmonic Percussive Separation

After discovering the pros and cons of each TSM method it is important to take this a step further and determine the effectiveness of separating a signal into independent instruments, then applying the TSM methods. This can be directly applied in music synchronization or in creating remixes of songs where the different parts need to match up. This portion of the project will explore how to implement harmonic-percussive separation.

The image below does a good job of illustrating the concept of harmonic-percussive separation that will be investigated in this paper.

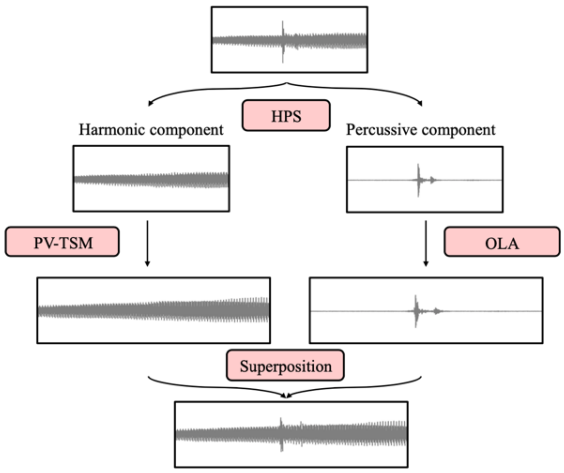

Figure 9: Windowing with Synthesis Hopsize H_s

The image below is a spectrogram of the original signal which contains both harmonic and percussive signals. The horizontal lines are the harmonic component and the vertical lines are the percussive component. The objective of harmonic-percussive separation is to end up with two spectrograms – one with primarily horizontal lines and one with primarily vertical lines.

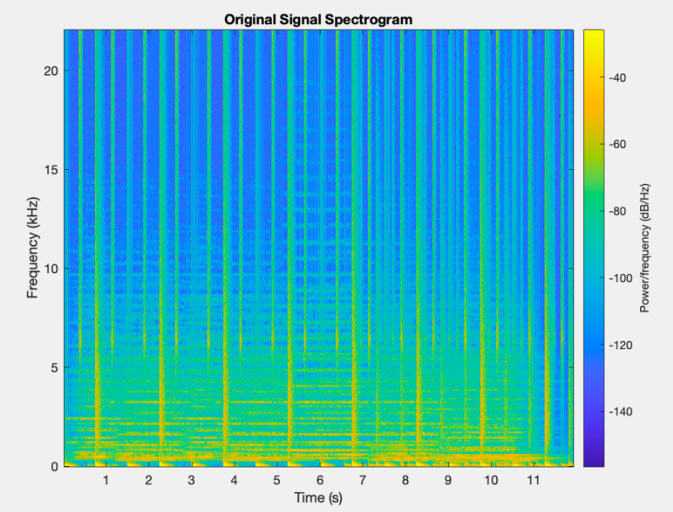

Figure 10: Windowing with Synthesis Hopsize H_s

The harmonic-percussive separation method used in the Dreidger paper was proposed by Derry Fitzgerald. To begin separating the harmonic and percussive portions of the signal, the STFT must be taken. Dreidger says "in the spectrogram Y = $|X|$, harmonic sounds form structures in the time direction, while percussive sounds yield structures in the frequency direction." [2]. The method of separating the various structures is by using a median filter in both the horizontal and vertical directions. A median filter is an image processing filter that goes through the signal pixel by pixel and replaces the pixel with the median value of all the pixels in the surrounding area.

The image shown below is the result of the median filter to enhance the harmonic signals. Notice how the horizontal lines are brighter than any of the surrounding data.

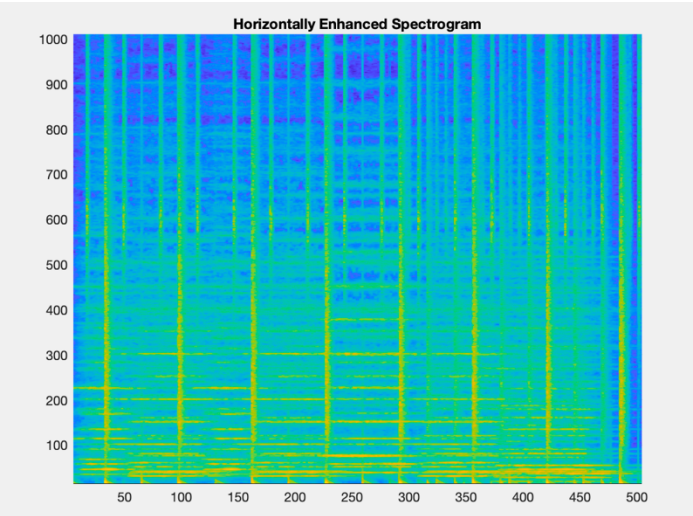

Figure 11: Windowing with Synthesis Hopsize H_s

The image shown below is the result of the median filter to enhance the percussive signal. Notice how the vertical lines are showing up a more bright yellow than any of the other surrounding pixels.

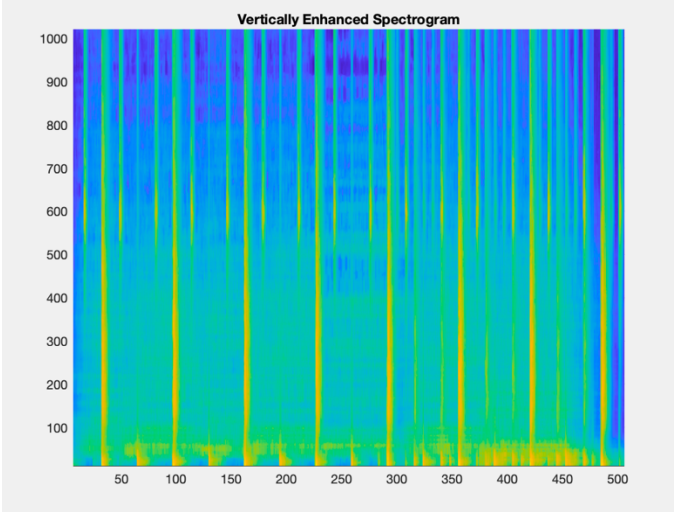

Figure 12: Windowing with Synthesis Hopsize H_s

After the horizontal and vertically enhanced spectrograms are calculated, the pixels are compared to determine whether the harmonic or percussive component is stronger. This process creates binary masks as shown below.

$$M_h(m,k) = \begin{Bmatrix} 1, if\ Y_h(m,k) > Y_h(m,k) \\ 0, otherwise \end{Bmatrix} \quad (7)$$

$$M_p(m,k) = \begin{Bmatrix} 1, if\ Y_p(m,k) > Y_h(m,k) \\ 0, otherwise \end{Bmatrix} \quad (8)$$

These masks are then applied to the original STFT signal to get the modified harmonic and percussive portions of the signal. This is applied by multiplying the two signals. To get to the final separated signals, the inverse STFT is taken.

The image below is the resulting harmonic signal found after harmonic-percussive separation. It is clear that the harmonic signal is significantly stronger than the noise. In listening to the resulting harmonic signal, there is a very quiet percussive signal that did not get filtered out.

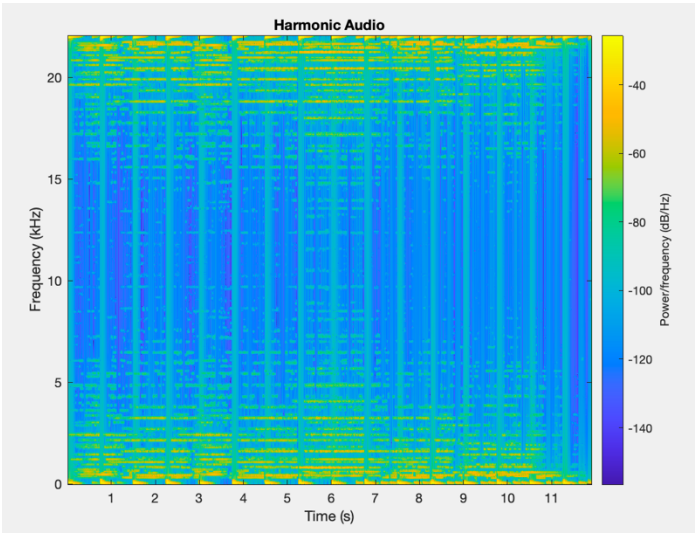

Figure 13: Windowing with Synthesis Hopsize H_s

The image below is the resulting percussive signal found after harmonic-percussive separation. While this signal appears weaker than the harmonic signal above, there are clear bands that indicate the presence of a percussive signal. The major issue with this separation is that the signal sounds slightly quieter in comparison to the harmonic signal and it is harder to pick it out in the resynthesized signal.

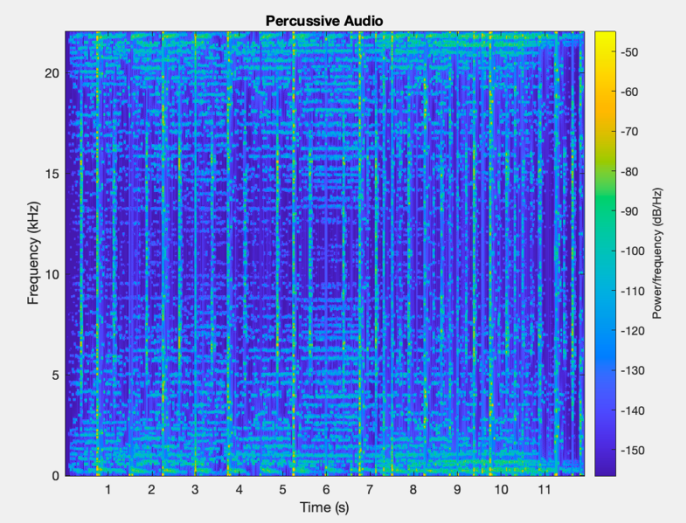

Figure 14: Windowing with Synthesis Hopsize H_s

Following the separation of the main signal, overlap-add was applied to the percussive signal and phase-vocoder was applied to the harmonic signal. The process of doing TSM separately on each signal adds a complication in that the two signals most likely will not be the same size. In this case, the solution was to take the smaller of the two signals and append zeros on the end so they are the same length. In the case of remixing a real song, the artist would choose where in the song they want to start and end their stretched/compressed signal which would require special software to make it easier.

When the resynthesized signal is played with both signals at 2x speed, there are distinct violin and drum sounds as expected. Because the phase vocoder introduced some reverberation, that is audible in the resynthesized signal. Although the signal does not sound perfect, it accomplishes the goal set in the beginning of the paper which was to separate, process, and put an audio signal back together using harmonic-percussive separation.

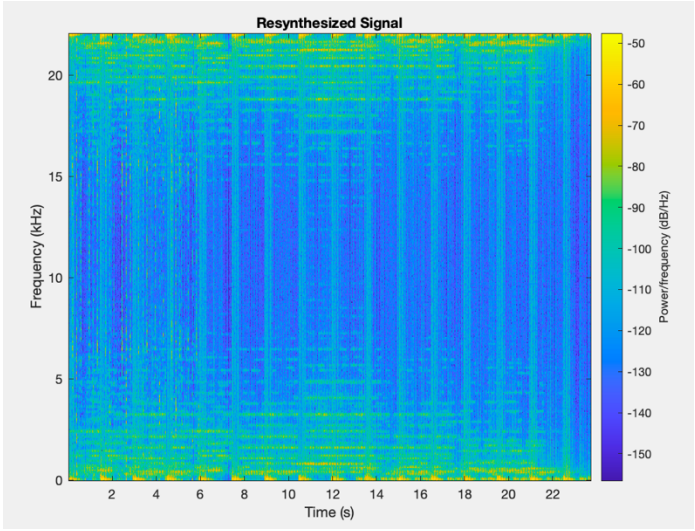

Figure 15: Windowing with Synthesis Hopsize H_s

V. Conclusion

Although the process of music mixing on the outside seems simple because of current software, to understand why it works is quite complicated. Time-scale modification is used heavily in this area and knowing which method to use for each type of signal is necessary for getting the best sounding output possible. After evaluating the performance of OLA and PV-TSM, it was determined that OLA works best for percussive signals and PV-TSM works best for harmonic signals. With this knowledge, harmonic-separation was implemented. Taking a more complex song and performing time-scale modification was a great introduction to audio signal processing.

References

[1] A. Crim, “An Exploration of Time-Scale Modification Methods”

[2] Driedger, J., M. Müller, “A review of time-scale modification of music signals,” *Applied Sciences*, vol. 6, no. 2, p. 57, 2016.

[3] Driedger, J., M. Muller, and S. Ewert. "Improving Time-Scale Modification of Music Signals Using Harmonic-Percussive Separation." *IEEE Signal Processing Letters*. Vol. 21. Issue 1. pp. 105-109, 2014.

[4] “Time-frequency masking for harmonic-percussive source separation,” *Time-Frequency Masking for Harmonic-Percussive Source Separation - MATLAB & Simulink*. [Online]. Available: https://www.mathworks.com/help/audio/ug/time-frequency-masking-for-harmonic-percussive-source-separation.html. [Accessed: 06-May-2022].