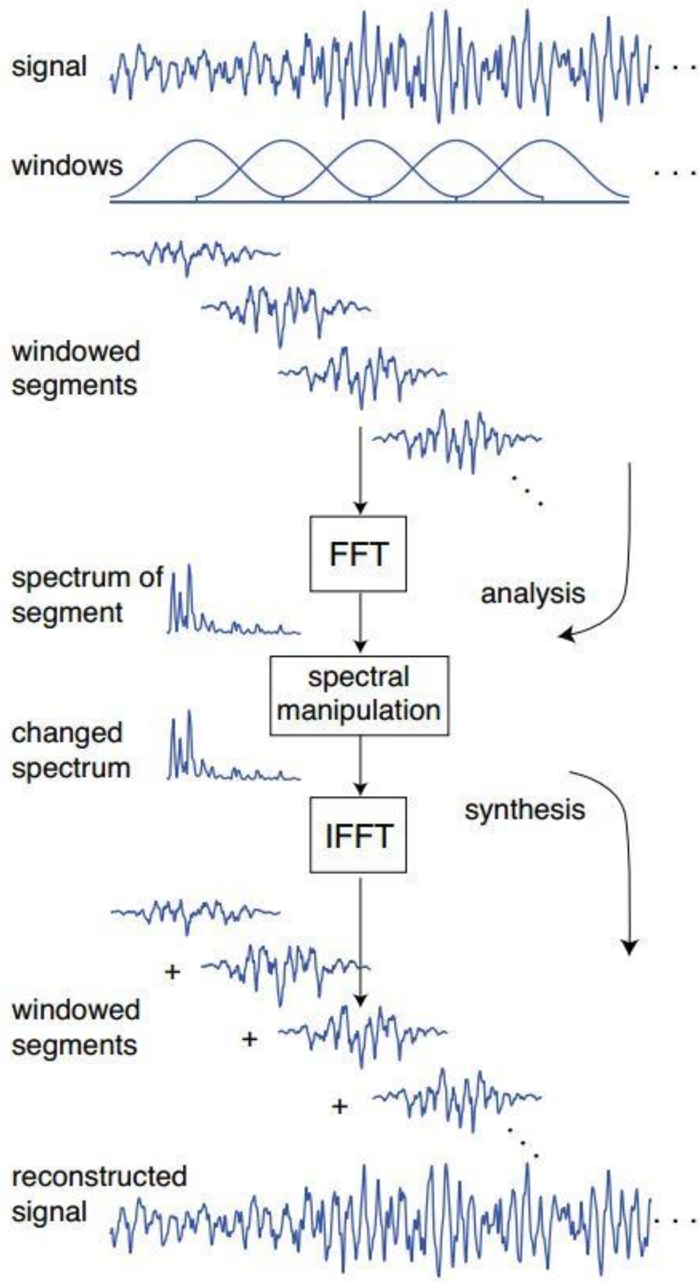
PHASE VOCODER REPORT

Spring 2017 Final Project

Jun D. Ouyang
jdouyang@ucdavis.edu

Background

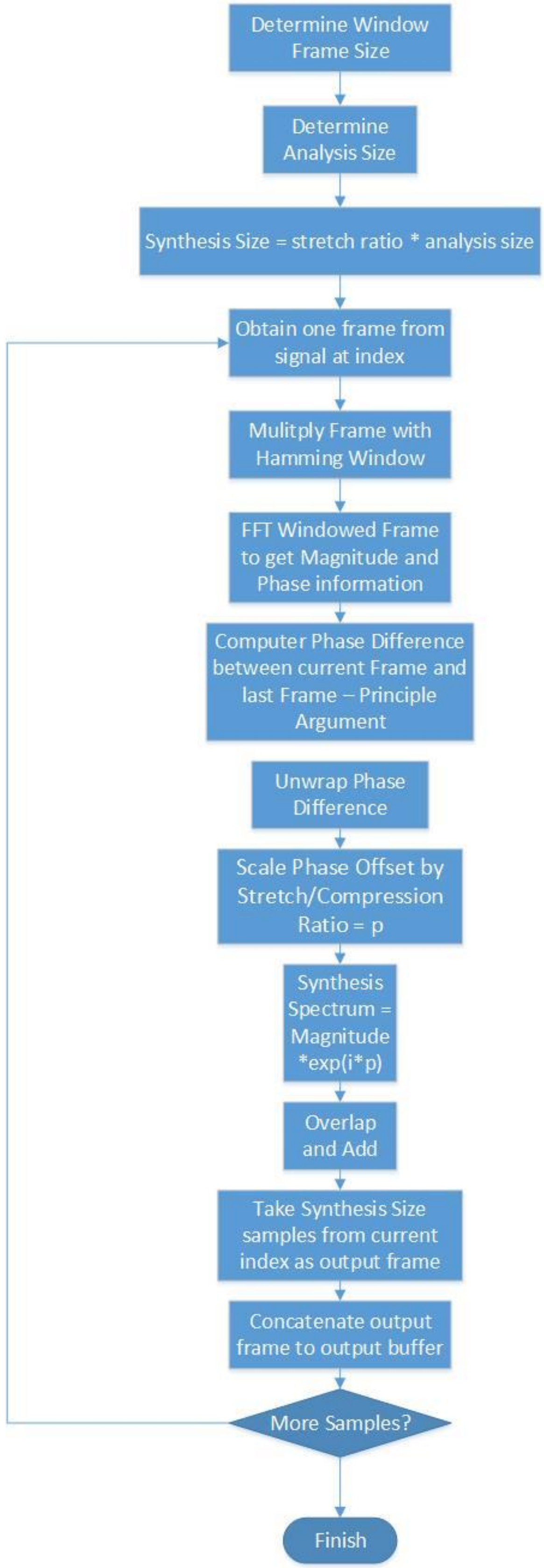
The phase vocoder utilizes phase information to reproduce ('synthesis') sound that is an approximation of the original. It does this by performing Short Time Fourier Transform (STFT), process the signal in frequency domain, perform the IDT-FT and then reconstruct the signal by concatenation. The concatenation allows overlapping to smooth the output signal and also permits different output size for time-stretching or compression effects. This process is depicted below:



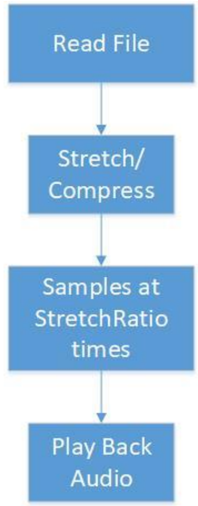
The Procedure

The cropping process is simply truncate a piece of signal in time. I will omit such discussion due to its simplicity.

To time stretch/compress the sound samples, a used the following procedure



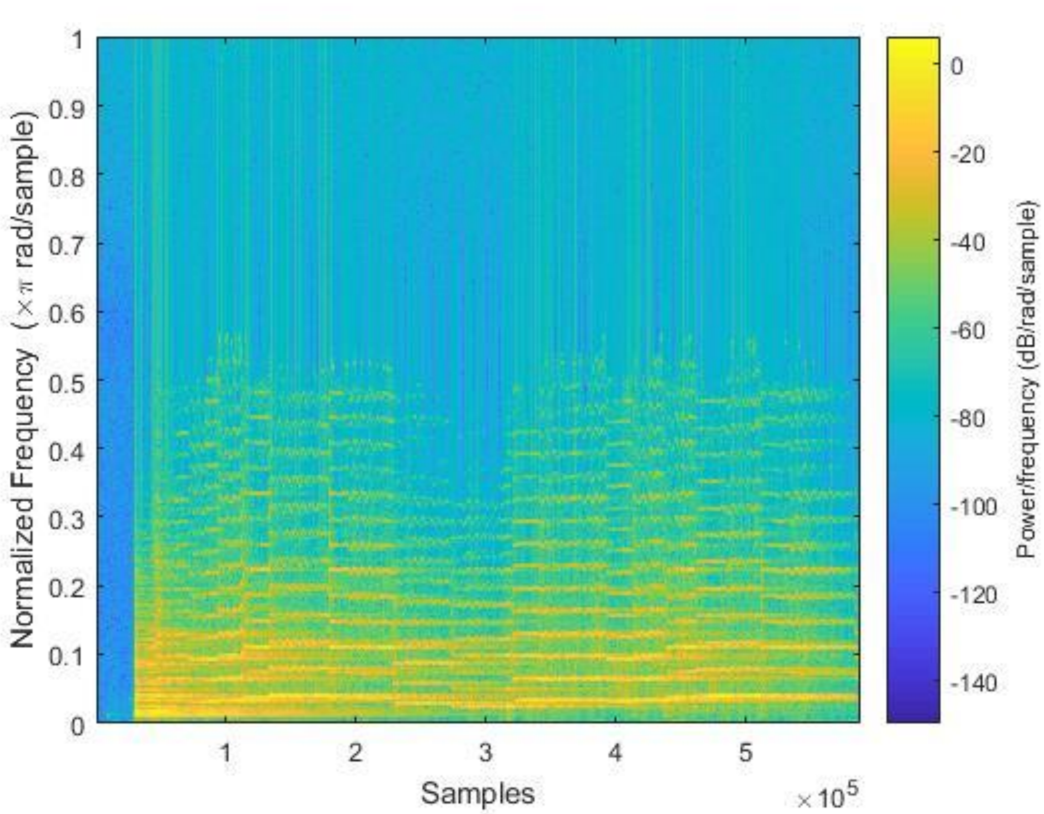
Naming such procedure into a function 'stretch', the frequency shift/pitch shifting can be readily applied using the following flow



Challenges

The main challenge of the stretching and compression lies in the phase estimation.

The spectrogram of the Violin signal is shown below



I observed that there are high frequency components near the beginning and the second half of the signal. Since the phases are directly related to the number of frequency components estimation by the following equation

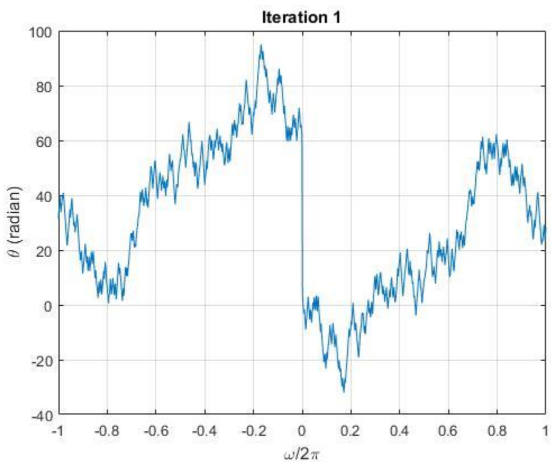
$$f_n = \frac{(\theta_1 - \theta_2) + 2\pi n}{2\pi \Delta t}$$

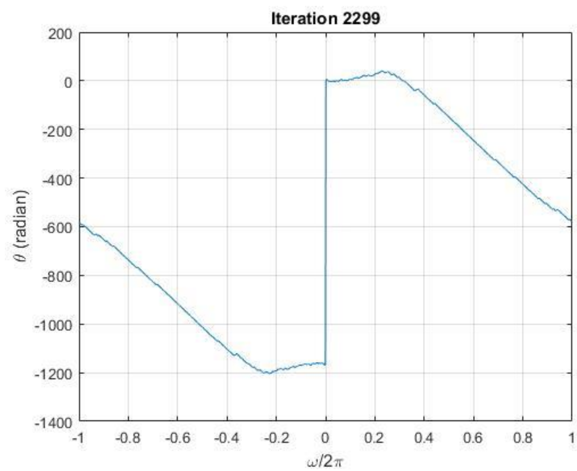
I will expect the phase variation is more violent in the beginning and near the end as well.

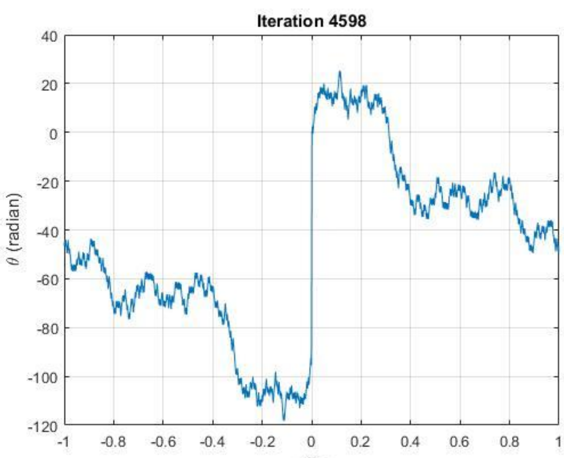
My initial approach was simply unwrapping the phase difference and scaling by the stretching ratio

```
frameAngleUnwrap = ((frameAngle - prevFrameAngle) + unwrapdata)*stretchRatio
```

The first six frame phases using the provided Violin piece are plotted and shown below







One can see that this produces quite linear phase at the output, which confirms that our phase is successfully unwrapped. However, I observe that the phase variation does not match with the spectrogram and in particular, in the middle of the piece in which it is showing a large variation in phase.

Solution

Modification is made using phase estimation introduced by http://www.cs.princeton.edu/courses/archive/spr09/cos325/Bernardini.pdf

```
    frameAngleUnwrap = frameAngle - prevFrameAngle - unwrapdata;

    frameAngleUnwrap = frameAngleUnwrap - round(frameAngleUnwrap/(2*pi))*2*pi;

    frameAngleUnwrap = (frameAngleUnwrap + unwrapdata)*stretchFactor;
```

Assume one frame has 0 phase, then the next frame in the perspective of the current frame has

$$\Delta\theta = \theta_2 - \theta_1$$

Phase shift. In the frequency domain, then, the current frame shall have

$$X\left(e^{j\Delta\theta}\right) = X\left(e^{j(\omega+\Delta\phi)}\right) = \left|X\left(e^{j(\omega+\Delta\phi)}\right)\right| \mathrm{e}^{\mathrm{j}(\omega+\Delta\phi)}$$

To isolate the phase term $\Delta\phi$, one has to subtract ω from the current phase difference to get the principle argument.

$$\Delta\phi = \Delta\theta - \omega$$

The phase increment from perspective of old frame, however, is not the true phase because of the phase error of taking many samples.

To get the true phase, one obtains the number of times k can multiply by 2π to obtain the nearest 0 phase point and subtract the principle argument by this value.

$$\phi = \Delta\phi - k * 2\pi$$

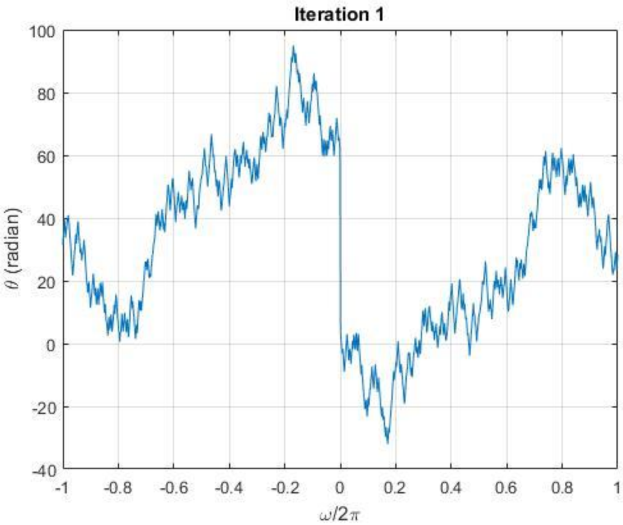
Where $k \in \mathbb{Z}^+$and gives the number such it minimizes Euclidean distance from $2\pi k$ to $\Delta\phi$.

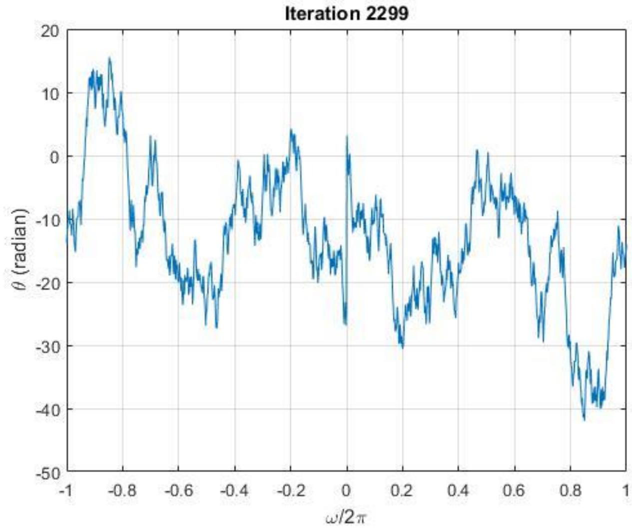
After obtaining the true phase, one restore the frequency information to frequency spectrum phase

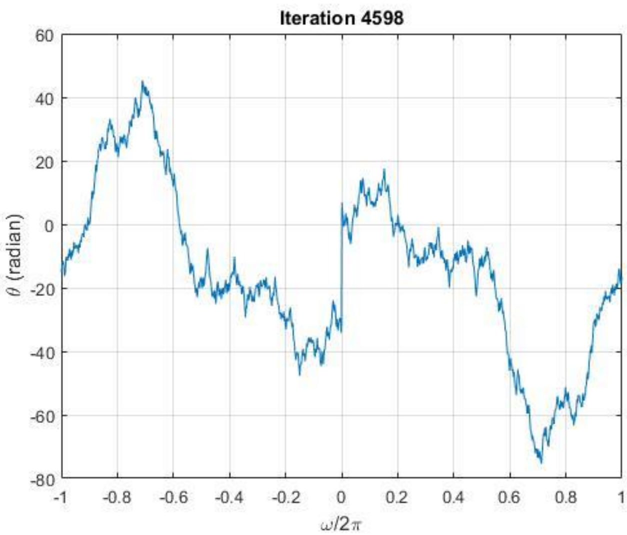
$$\theta = \phi + \omega$$

Then using $\left|X\left(e^{j(\omega+\phi)}\right)\right|\mathrm{e}^{\mathrm{j}(\omega+\phi)} = \left|X\left(e^{j\theta}\right)\right|\mathrm{e}^{\mathrm{j}\theta}$, one can take the inverse Fourier transform to obtain the time samples.

Using this approach, the phase diagram are updated as follows



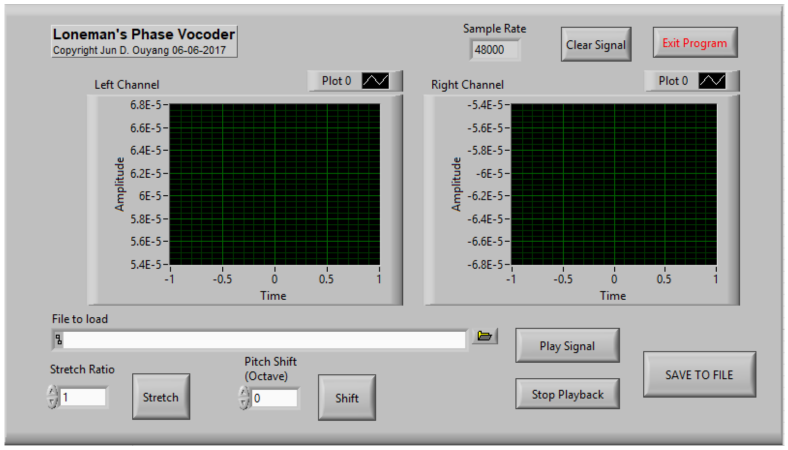




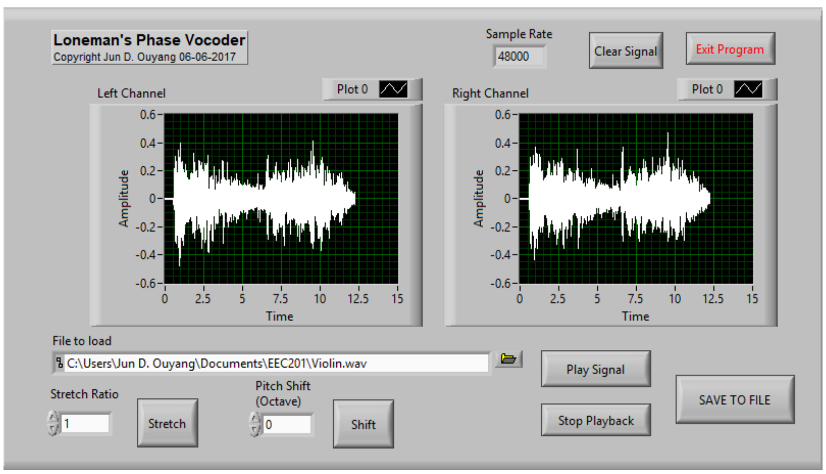
This time the phase variation agrees with the signal's spectrogram in variation of frequency. Instead of using phase difference directly, using rounding error for principle argument allowed smaller phase variations which disabled the phase wrapping in the middle of the piece.

GUI

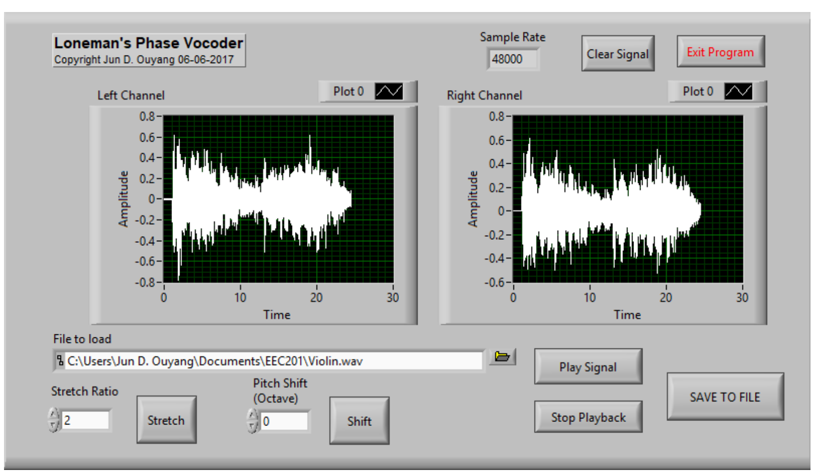
A LabView-based GUI is created to allow users to test my implementation with a click away. The panel is shown below



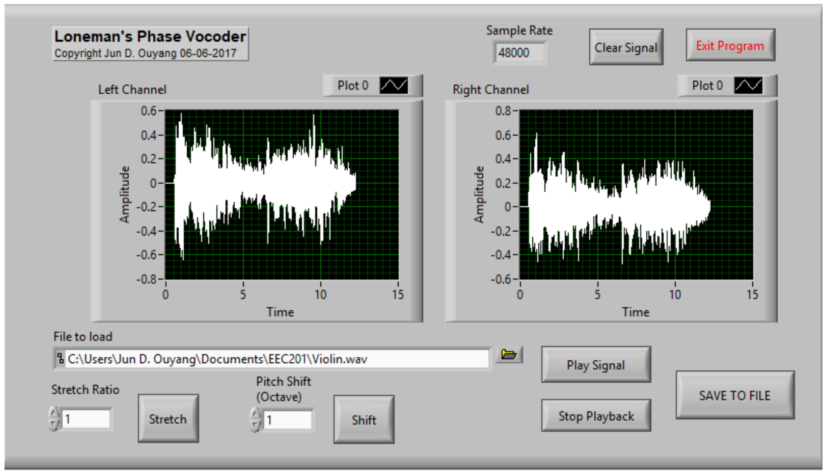
After running the program, the user can load file from their directory and apply stretching and pitch shifting effects. They can then play back their signal, apply further effects and ultimately save the modified file to disk. The GUI with Violin.wav music loaded is shown below



After applying stretching the following diagram will result



One can see the time stretching effect by observing the x-axis on the left and right channel plot. Lastly, to demonstrate the implementation of pitch shifting, the same piece after shifting 1 octave using the GUI is shown below



Conclusion

In this report, I have demonstrated the implementation of phase-vocoder using STFT and showed that it can be applied to audio signals. Using this technique, I have also built a GUI around it in LabView to facilitate simple tests of my theory.