Audio Effects Portfolio

Kyle Daruwalla & Jacques St. Louis

May 12, 2016

1 Chorus Effect

1.1 Description

This mono effect combines a track with one or more delayed copies. There is a separate gain control for the dry copy and each of the delayed copies. The delay block accepts fractional delays from a local oscillator (usually at a frequency of 0.1 Hz) where each delayed copy has an LO at a different phase. As a result, the listener will perceive multiple voices or instruments playing together (like a chorus).

1.2 Applications

The chorus effect is an easy way to add depth to a track. In particular, it is an artificial method to simulate double-tracking, a common technique used by artists to create a "fuller" sound.

1.3 Principles of Operation

The chorus effect is composed of several variable delay lines controlled by an LFO. Typically, the original, dry signal is added to the delayed, wet signals. The effect can come in two variants - standard chorus or multi-voice chorus. The standard chorus contains only a single delay line. The multi-voice chorus can contain several delay lines, where each delay line's LFO differs in phase. Figure 1 contains a block diagram for a multi-voice chorus effect with two delay lines.

1.4 Implementation

We implemented a standard chorus (chorus.m) and multi-voice chorus (chorus_multi.m) with four delay lines in MATLAB. The standard chorus had a single delay line with an LFO at 0.08 Hz and peak delay of 30 ms. The multi-voice chorus featured four LFOs that were all operating at 0.08 Hz and peak delay of 30 ms. However, the first LFO was in-phase with the original signal, the second was 45° out of phase, the third was 90° out of phase, and the fourth

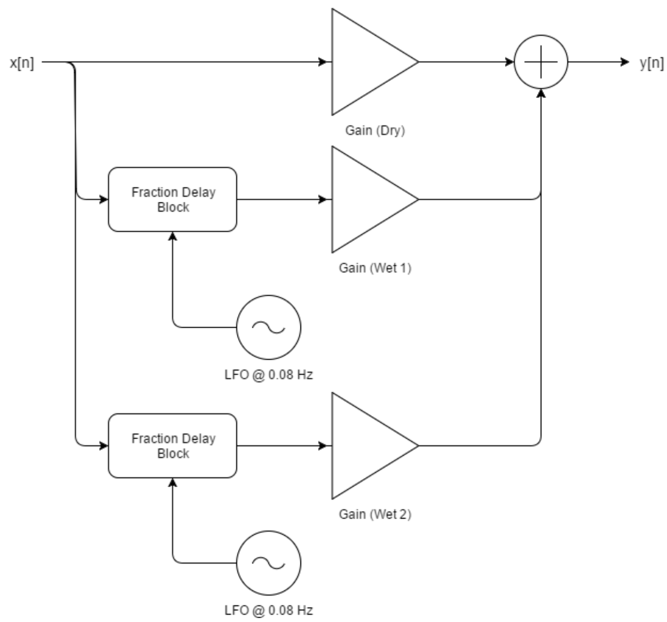

Figure 1: Block Diagram of Chorus Effect

was 135° out of phase. The first and second LFOs were hard panned to the left channel, and the third and fourth LFOs were hard panned to the right channel. The end result is a richer, fuller sound.

1.5 Demo and Discussion

The original audio is an electric guitar track that is mono and panned center. We found the best results were obtained for an LFO frequency of 0.08 Hz and a peak delay of 30 ms. These settings can be heard for both the chorus and the multi-voice chorus effects.

We also examined how the effect changes when parameters are varied. Since the LFO frequency is less than 0.1 Hz, changing the frequency is not as noticeable as an effect like flanger. However, there is a subtle change in the level of depth or richness added by the effect to the track. We created output tracks for LFO frequencies of 0.01 Hz, 0.02 Hz, 0.03 Hz, 0.04 Hz, 0.05 Hz, 0.06 Hz, 0.07 Hz, 0.08 Hz, 0.09 Hz, 0.1 Hz.

Varying the peak delay has a more pronounced effect than varying the LFO frequency. In particular, at a peak delay of 10 ms, the synthetic quality of the flanger effect can be heard. As we vary the peak delay up to 20 ms, the synthetic quality is faintly audible; it is complete gone at 30 ms.

1.6 Further Exploration

To see this effect used in a real world application, check out this YouTube video that uses a chorus pedal on a guitar.

2 Flange Effect

2.1 Description

The flange effect is very popular among musicians, as it adds interest to a track in a periodic way. Flanging sounds as if an original instrument's sound is changed from a natural to artificial and back, or as if the audio has been combined with a jet sound.

2.2 Applications

The flange effect is often used to add a periodic, synthetic sound to a track. This adds an interesting sound variation to most any musical track.

2.3 Principles of Operation

The flanger is quite simple in concept: it consists of a variable delay which is controlled by a low frequency oscillator (LFO). The delay output is then summed with a non-delayed version and output to the destination.

2.4 Implementation Notes

To implement this effect, we used MATLAB (see Flange.m). Starting with the basic audio file implementation from class to handle importing and playing the audio files. We built upon this and added both a new line in the step function for adding the two signals to be produce together with varying gains. A block diagram can be seen in Figure 2.

2.5 Demo and Discussion

Flanging can be used in a variety of ways to produce a wide range of sound effects from the same track.We adjusted the maximum delay using a 4ms delay and an 8ms delay using a 0.4Hz LFO. We then tried used a 1Hz LFO with a 4ms and an 8ms delay.

The flanging sound is very noticeable, but not extremely deep. Many flangers have feedback in their delay line, creating resonances in the track which can increase the effect's presence in the track.

We then implemented a stereo flanger, using a second LFO with a $\pi/2$ phase shift. The results of our adjustments include maximum delay using a 4ms delay and an 8ms delay using a 0.4Hz LFO. We then tried used a 1Hz LFO with a 4ms and an 8ms delay.

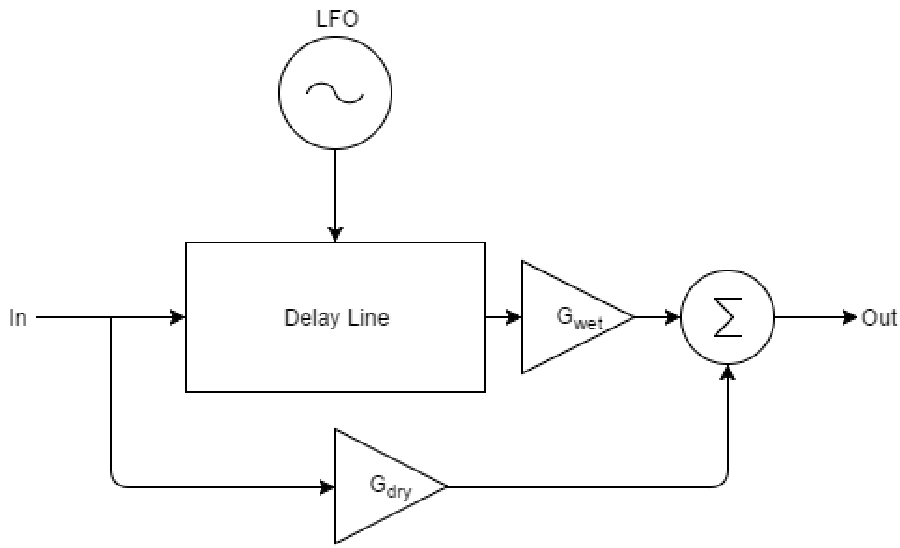

Figure 2: Flange Block Diagram

2.6 Further Exploration

Something to investigate is how this effect would change if it also had a chorus aspects, placing the flange effect at different rates on different chorus channels. This would be an interesting demonstration of two effects from this chapter. A cool way to further investigate this effect is to watch this neat YouTube video about the flange effect here.

3 Ring Modulation

3.1 Description

This mono effect modulates a track with carrier signal to create a wavering robotic sound.

3.2 Applications

Ring modulation can be used to added a robotic sound to a track. Often, it is used on speech or guitar tracks.

3.3 Principles of Operation

Ring modulation is done by modulating the input signal with a carrier signal whose frequency is in the range of the audio. The carrier signal is described by

$$m[n] = 1 - \alpha + \alpha \cos(2\pi f_c n)$$

where α is the depth of the carrier signal and f_c is the carrier frequency. Using $\alpha = 1$, suppressed carrier amplitude modulation is achieved. Decreasing α will add more of the DC term back into the carrier signal. The carrier frequency typically ranges between 10 Hz and 1 kHz. The output is then given by

$$y[n] = x[n]m[n]$$

A block diagram can be seen in Figure 3.

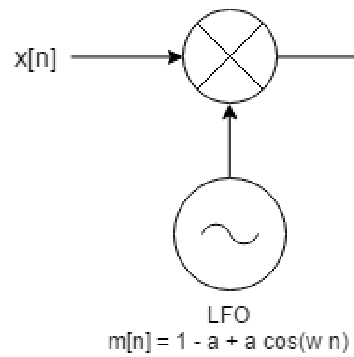

Figure 3: Block Diagram of Chorus Effect

3.4 Implementation

We used MATLAB to implement the ring modulation effect (see ring_modulation.m). An LFO object was used to generate the signal $\alpha \cos(2\pi f_c n)$. This was then used to create the signal $m[n]$. We multiplied the generated carrier signal with the input $x[n]$ to produce the output. Each channel of a stereo track was modulated individually.

3.5 Demo and Discussion

The effect is most obvious when comparing these original vocals to the modulated output (LFO at 250 Hz and $\alpha = 0.5$).

However, a more pleasing use of this effect is achieved when it is applied to this original bass track. To study the effect, the different values of α were tried: 0.1, 0.5, 0.8, 1.0.
Comparing the $\alpha = 0.1$ and $\alpha = 1.0$ tracks, the effect of the depth parameter can be heard. With a larger α, the rich spectrum of the audio is lost, and several notes sound flat.

Additionally, we can vary the frequency of the LFO. Changing the frequency will change the rate of the wavering in the output audio. At higher frequencies, the wavering is occurring so fast that the output audio sounds robotic. We attempt frequencies of 10 Hz, 50 Hz, 100 Hz, 1000 Hz.

3.6 Further Exploration

To learn more about this odd effect, check out this YouTube video.

4 Frequency Bin Shifting

4.1 Description

Pitch Shifting with frequency bins is a way in which one can change the pitch of a given track. The bin application stems from taking the Fast Fourier Transform (FFT) of a given waveform. An advantage to this processing technique is its speed, though it must be noted the outcomes do not always sound the best compared to time stretching/compressing. Another plus to using frequency bins for pitch shifting is the ability to define frequencies grouped into each bin and the number of bins to shift by, allowing precise control of the shifting degree. The second part is in compression, where a certain amount of information, particularly from the low end, is thrown out.

4.2 Applications

This process is often used in certain types of music and audio production. Some examples of pitch shifting can be found music, making certain vocalists voices sound deeper than they normally are. In film, pitch shifting may be used to make an actor sound younger than they are, such as making a voice higher to sound younger. The compression technique gives an interesting sound, preserving certain frequencies while throwing out others, producing a hollow sounding audio track.

4.3 Principles of Operation

THe basic idea here is that by taking the FFT, one can break the audio waveform into levels of specific frequencies.By taking the amplitudes and phases in these bins, changing the pitch is as simple as moving a bin from one location to another, high for a higher pitch, lower for a lower pitch. The degree of shift determines how much of a change will take place, and the more numerous the bins over a given range of frequencies the better, giving the user greater control over the amount of shift as well as increased resolution in frequency representation and reconstruction.

The Compression simply upsamples the audio file and then throws out the middle section, leaving fewer frequencies spread out along the spectrum.

4.4 Implementation Notes

To implement this effect, we leveraged the FFT function, part of the DSP toolbox, in MATLAB. As the program stepped through each windowed chunk of audio, the program would take the FFT, producing an array of values, each index corresponding to a small range of frequencies. The program then circularly shifted the array, careful to shift the upper and lower halves in opposite direction to maintain positive and negative frequency bins on the appropriate sides. A visual representation is shown in figure 4

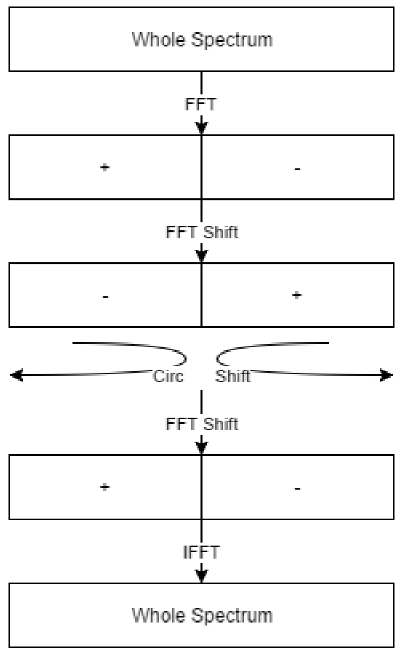

Figure 4: Bin Shift Process

In creating the spectrum compression effect, we had the stepped audio broken into bins as done before with the FFT. After we obtained the spectrum, we upsampled the spectrum array by the factor given and then cut out the middle part during the rebuilding of the array phase, ending the first section at (StreachFactor-1/4)*ArraySize and started the next array chunk at (StreachFactor-1/2)*ArraySize. A visualization of the precess is shown in figure 4

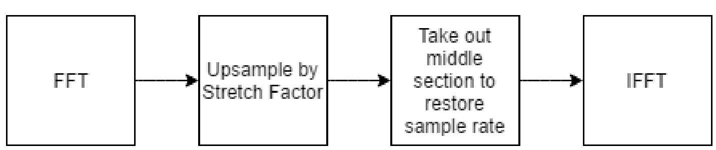

Figure 5: Compression Process

4.5 Demo and Discussion

In using frequency bin shifting, the shift factor has one of the most profound effects, other changes have significant changes as well. First we set the hop size to 10ms and the shift factor to change, first being 5 Bins up and another at 5 Bins down. We then changed the hop size to 20ms, then repeated with 5 Bins up and again at 5 Bins down.

We found that by lowering the hop size, some of the harsh roboticization was reduces and the audio sounded more fluid.

For spectrum stretching we used th esame audio file, setting the hop size to 10ms and the shift factor to change, first being a factor of 10 and another at a factor of 20. We then changed the hop size to 20ms, then repeated with a factor of 10 and again at a factor of 20.

4.6 Further Exploration

Pitch shifting is especially active in the guitarist community. For further work and discovery in the FFT bin shift and more about the FFT, check out the link here. The web page provides good figures and explanations to understand just what each step in the program is accomplishing.