Timothy Sjoquist

David Chen

EE443

Electric Guitar Effects Board

1. **Motivation**

Music has always been a cultural aspect of society in America with large amounts of people wanting to listen to it as well as play it for themselves. One such instrument that I have always been fascinated with is the electric guitar and the various effects related to them. However, most of these effects can only be used if an individual buys an expensive product which only applies one particular effect. Buying many of these effect boxes get quite expensive and which makes it difficult for the average consumer to have a large catalog of effects to utilize. Many expert guitar players are willing to buy such expensive high quality boxes as they use them extensively for performances, but the majority of guitar players do not want to pay large amounts of money for each individual effect and are not necessarily needing the highest quality effects. For this reason for our capstone project we have decided to program the C6713 DSK board to be an effects board primarily for an electric guitar. We want to create a quality product which does not implement a single effect but has numerous effects in it. We will also be able to have more than one effect being used at a single time so the guitar player can simulate using many different boxes at the same time. Lastly we would want to keep the price relatively cheap to market the board to a large consumer base. As long as the quality is present, our product would be profitable as our board would implement many effects compared to a single effect. This product would be aimed to more intermediate\advanced guitar players who do not want to pay large sums of money for effect boxes. In essence, the motivation for our project is to create a quality effects board and be able to sell it for a lower price allowing us to make a profit.

1. **Problem Formulation**

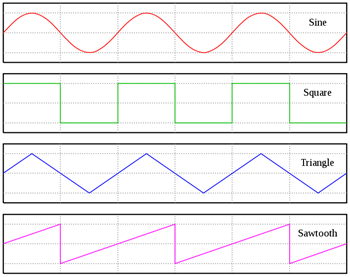
The primary uses of guitar special effects are to produce all the significant sounds that give a guitar performance its own flavor.  Our goal is to providing a powerful, comprehensive guitar effect generator while simplifying the user controls. We intend for a GUI design that allows the user to turn on all different effects and control their gains and tones with sliders.  Our effects should simulate the ones produced in industries and we may figure out a more convenient way to control the effects than placing a multiple-knob stompboxes at floor level. We would also like to test our design on a real guitar to show our generator's capacity for a real performance.

1. **Proposed Tasks and Data Collection**

As previously stated, we have decided to program an effects board for which would be used by an electric guitar. We will plug the guitar into the board as real time input and have the output of the board go to an amplifier allowing us to hear the signal. Using the board, we will be able to manipulate the input guitar signal various ways to add particular effects to the clean guitar signal and hear them out of the amplifier. We will program the board to be controlled via the PC using RTDX to change what effect is being used and in what magnitude. The effects that we have chosen to implement are distortion, chorus, flanger, phaser, tremolo, vibrato, as well as an equalizer.

Before going deep into discussing effect, we need to know that many effects requires a carrier signal to modulate one of the original signal’s characteristic(be it amplitude or signal delay length); this can largely be achieved by an LFO (Low Frequency Oscillator). Our LFO is designed to carry either a square, triangular, or a sine waveform; its depth and its frequency can be changed in real time using RTDX to reflect on the different modulations we like to bring onto our music.

Figure 1: LFO waveforms. Our effects board provide the first three waveforms and depth/frequency control.



The first effect, distortion, is when clipping is present in the signal. Slight distortion is when the sine wave’s tops and bottoms are clipped. Mid-level distortion is when the curves of the sine wave start to become more like a square wave. Extreme distortion is when the sine wave changes form to approach the shape of a square wave. We accomplished this task by creating a max and min level for the waveform. If the values of the wave reach above or below this level, the value is changed to be the max or min value. In our implementation we control the amount of distortion with a slider which, as its increased, lowers the maximum level and conversely raises the minimum level. The maximum and minimum level are equal to each other except for a sign reversal. As the distortion level is raised, more of the input waveform gets leveled off making the original wave approach the look of a square wave. An example of this can be seen in Figure 1. In the first line no parts of the signal are being removed while in the next line some clipping is present with the peaks of the signal being chopped off. The last line the threshold level has been lowered further, making the waveform appear much more like a square wave.

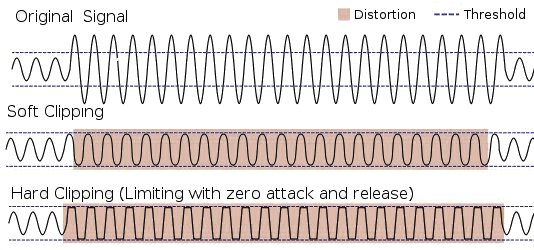


Figure 2: Clipping example (From Wikipedia source)

Chorus is the second effect that was implemented which is when the output sounds like there is more than one guitar being played at a time creating a thicker, more robust sound. Another way to think of this effect is how a string section all plays together playing the same note, but each instrument has a slightly different timbre. This was implemented by creating an array to hold the most recent 1000 samples and based on the RTDX slider control, sum the input signal with a particular delayed sample. The RTDX slider controls the amount of delay between the input sample and the delayed sample that will be combined. This delay will range from the current input sample to 1000 samples late which is equal to 62.5 milliseconds as the sampling frequency used is 16,000. Mixing the delayed sample and the current sample creates peaks and notches where parts of the signal have either negative or positive interference.

A similar effect to chorus is the flanger effect which also delays an input sample and mixes it back in with the current input signal. The difference between this and chorus is that the flanger delay time gradually changes causing a swept “comb filter” to occur as peaks and notches sweep up and down the frequency spectrum. This was implemented by creating another array to hold the 1000 most recent samples. A certain max delay amount is controlled from the RTDX slider for this effect which determines the range of delay; the range is from 0 samples to 1000 samples which is 62.5 milliseconds. Every 1000 samples, as the delayed signal is mixed in with the current signal, the delay between the two signals is either incremented or decremented by a certain amount (another slider controls this value). This changing delay will first decrease until the delayed sample is the same as the current sample. At this point the delay will switch to incrementing every 1000 samples until the delay hits the max delay amount. At this point the delay will decrease again as before. This creates more of a jet plane or spaceship sound.

The next effect is the phaser effect which is when an input signal’s phase is altered and combined with the “clean” signal. This is accomplished by splitting the signal and passing one copy of it through an all-pass filter which keeps the magnitude the same but alters the phase of different frequencies and mixing the two together. There will be an LFO continuously shifting the location of the notches and peaks so different frequencies are emphasized and de-emphasized at each pass. The rate of this LFO will be controlled by RTDX. We created the allpass filtering effect by using IIR difference equation derived from the transfer function below. The +/-θ below is the angle of the complex pole pair. By fluctuating θ we can create a sweeping effect varying the phase delays through time. While performing, the user can in real time control the depth and frequency of sweep to give the music signal color.

Figure 3: phaser configuration

(a) H(z):allpass filter transfer function. We use a single conjugate pole pair in each allpass section.

R: the radius of each pole with the conjugate pole pair.

(b) The configuration of the phaser effect

(b)

(a)

The tremolo effect is the fifth effect which is when the amplitude of the signal is quickly modulating. Using a LFO, we can create a carrier signal consisting different wave shape to modulate the amplitude of the final output. When listening to the tremolo processed signal, the listener can hear the music changing volume in a shaped pattern given by the LFO. The speed of the modulation will be controlled by RTDX.

The vibrato effect creates an output signal with the characteristic of fluctuation around the pitch of the original signal. Similar to the tremolo effect, the vibrato uses an LFO to modulate a simple property of the original signal. However it doesn’t modulate the amplitude but the delay length. The physics connecting between a varying delay length and a varying pitch is illustrated by the Doppler effect of a sound wave; using varying delays through time we fool the listener to believe our sound source is travelling closer or farther to him or her. The amount of variation would be controlled by RTDX.

Lastly we will implement an equalizer to control the tone of the music. An equalizer can either boost or diminish a range of frequencies which can result in the final doctored sound being much different than the original sound. This was implemented by using the code from one of the questions in the assignments which used assembly code to perform convolution. In the implementation three filters were used for altering the amount of particular frequencies being present. A lowpass filter with a range of 0-1500 Hz, a bandpass filter with a range of 3500-4500 Hz and a highpass filter with a range of 4500-9000 Hz were used to cover the frequency spectrum used by this project. The amount of gain for each particular filter was controlled via the RTDX GUI.

Lastly in the implementation one other slider was used to control the overall volume for the output signal. This gain was only applied after other effects were potentially applied to the signal so as to not obstruct the implementation of the effects. For example, if the overall gain was applied before the distortion effect, more of the signal would be flattened than would be originally desired.

When we worked on this project, we used the oscilloscope and the function generator extensively to help collect our data. When programming the effects, originally we never used any input, but used data from a table to make it easier to test and prove that our code was performing correctly. Once this was accomplished, we hooked the input to a function generator and looked at the output via a oscilloscope. The next step involved joining the code of each individual effect together so we could choose between various effects. Once all the effects were coded in the same file, we changed from controlling the input parameters with sliders in CCS to the use of RTDX and Matlab to control them. We designed a GUI in Matlab to control all the effects which uses RTDX to alter the parameters as desired. Once all of this was accomplished we hooked up the electric guitar to the board and the output to a set of speakers.

1. **Performance Analyses**

At the conclusion of this quarter we fulfilled our goal of creating a multi effect board successfully implementing distortion, chorus, flanger, phaser, tremolo, vibrato, and equalizer. We also designed a graphical user interface to control the board which is shown in Figure 2. Each toggle switch turns on the corresponding effect while the sliders modify particular aspects of each effect. The distortion slider increases the amount of distortion in the signal, while the chorus slider determines what the delay will be for the chorus effect. For the flanger effect, the slider controls what the range of the delay will be for the effect itself as the delay will change, but will remain between the maximum delay and the current sample. The three sliders for the equalizer correspond to how much gain is for each particular filter. For phaser, tremolo, and vibrato, each particular effect has a slider for depth which determines the extent the effect is applied. The frequency sliders determine the rate the effect is being applied. The type of wave choice determines what kind of wave the carrier wave is to apply the particular effect. The volume slider controls the overall volume. The start button loads the program and starts it running while the stop button ends the program. Lastly the “Click to Show Values” button prints out what the values are for the sliders.

In the project one particular problem that we kept running into was timing. Originally we used polling, but it resulted in incorrect output as it was not fast enough. We overcame this by changing to interrupts which greatly increased the speed as it is more efficient and allowed our program to work correctly. While most of the problems were dealt with in this project and the majority turned out very well, the project could be polished in particular areas if we were allowed more time. One such thing that would be a priority would be determining why a clicking sound is present for the flanger effect and the phaser, tremolo, and vibrato effect if the square wave is being used. Also timing remains a bit of an issue as cascading all of the effects together result in incorrect output. The slowest effect is the equalizer which applies three different filters, but the remaining effects work well cascaded together. Lastly, many of the effects require cautious tuning of the coefficients to prevent the signal from blowing up and destroying the musicality of the signal. Those are the three major issues that would be investigated as the rest of the project works reasonably well.

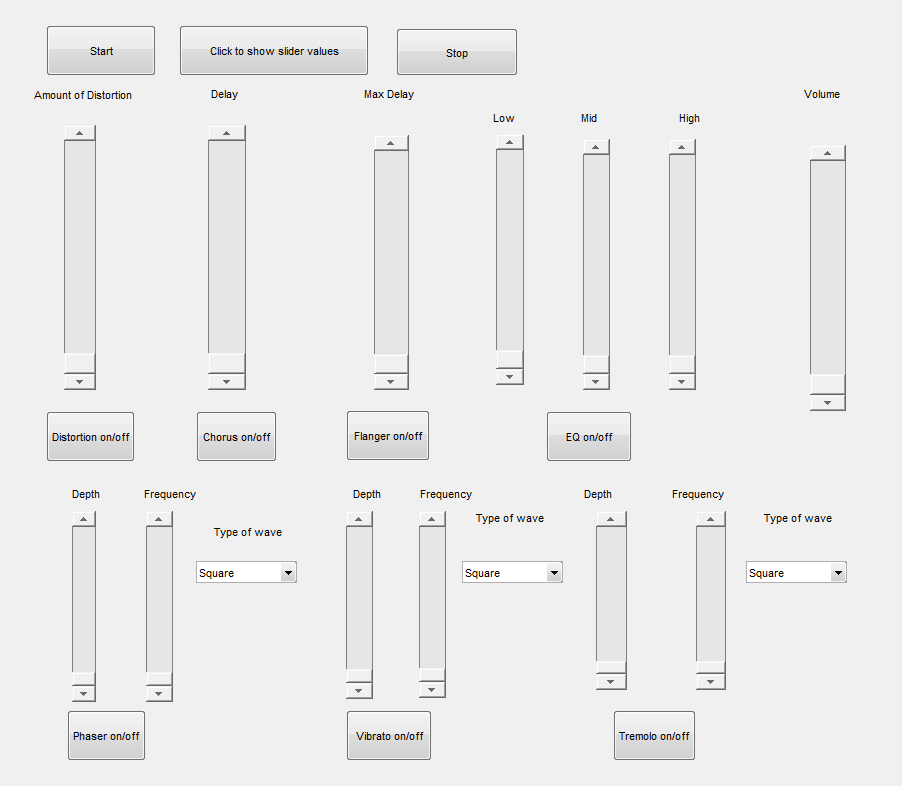


Figure 4: GUI design

1. **Conclusion**

In conclusion, for our capstone project we designed an electric guitar effects board which implements several effects instead of a single one. This is so to give beginning or intermediate players a device to modify their guitar sound with various effects without needing to spend large sums of money to do so. We successfully implemented distortion, chorus, flanger, phaser, vibrato, tremolo, and an equalizer which could be applied to any input, including an electric guitar. We stumbled upon a few problems and had to overcome them to allow the program to work correctly especially needing to change from polling to interrupts to have the program run fast enough for the input. Overall this was a very interesting project as it helped to connect the theory learned in class to real world products. It was rewarding to be able to see the progress of it and ultimately see a working project from our labors

1. **References**

**Works Cited**

"Effects." *Teach Me Audio | Effects*. Teachmeaudio.com. Web. 07 May 2012. <http://www.teachmeaudio.com/production/mixing/5-equipment/30-effects>.

"Effects Unit." *Wikipedia.com*. Wikipedia. Web. 7 May 2012. <http://en.wikipedia.org/wiki/Effects\_unit>.

Hunter, Dave. "Effects Explained: Modulation II-Vibrato, Tremolo, Octave Divider, Ring Modulator." *Gibson.com*. Gibson, 1 July 2008. Web. 7 May 2012. <http://www.gibson.com/en-us/Lifestyle/ProductSpotlight/GearAndInstruments/vibrato-tremolo-octave-divider/>.

Hunter, Dave. "Effects Explained: Modulation-Phasing, Flanging, and Chorus." *Gibson.com*. Gibson, 25 June 2008. Web. 7 May 2012. <http://www2.gibson.com/News-Lifestyle/Features/en-us/effects-explained-modulation.aspx>.

Nave, R. "Electric Guitar Effects." *Hyperphysics.edu*. Hyperphysics.edu. Web. 7 May 2012. <http://hyperphysics.phy-astr.gsu.edu/hbase/music/eguiteff.html#c3>.

"Phaser (effect)." *Wikipedia*. Wikimedia Foundation, 29 May 2012. Web. 06 June 2012. <http://en.wikipedia.org/wiki/Phaser\_%28effect%29>.

"Allpass Phaser Architecture." *Allpass Phaser Architecture*. Web. 06 June 2012. <https://ccrma.stanford.edu/~jos/pasp05/Allpass\_Phaser\_Architecture.html>.

Clarke, Adrian. *Digital Implementation of Vibrato*. Rep. Faculty of Architecture, Design and Planning, The University of Sydney. Web. <http://ses.library.usyd.edu.au/bitstream/2123/8302/2/Adrian%20Clarke\_Report1\_final%20submission.pdf>.