Karplus Strong Algorithm

Ayush Kumar Lall - 2020122001
Bharadwaj S. - 2020122003

25th November 2020

1 Introduction

Karplus–Strong algorithm is a method of physical modelling synthesis that loops a short waveform through a filtered delay line to simulate the sound of a hammered or plucked string. It played a revolutionary role in electronic music industries and till today it is been extensively used.
It gave us the opportunity to experiment with the sound and produce the sounds that are difficult to produce with the help of the musical instruments.
The basic functionality of this algorithm is that we are using a noise burst as our signal which is given to the input. This signal is then passed through a feedback which is having a delay component this helps in making a delay of the signal of sample length. Now in order to filter this signal we will be using the moving average filter.
The algorithm was defined as follows:

- The pluck is modeled with a short excitation input, originally a noise.
- The excitation is sent to output and to a feedback loop through a delay line of length L.
- The output of the delay line is sent to a moving average low pass filter.
- The output of the low pass filter is mixed with the excitation to the output.

So lets first see how we are analysing the input signal. For that we will be using the following block diagram:-

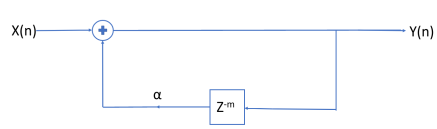

Figure 1: Block diagram without filter

So the difference equation of the corresponding setup will be

$$y(n) = x(n) + \alpha y(n - m)$$

Where α is the gain value and m is the delay value. This help to make the signal that we are giving in the input , which is in our case a white noise, as periodic by providing it a delay line .
Now lets try to put a low pass filter in the feedback loop which is a moving average filter in our case. This block diagram will let to certain differential

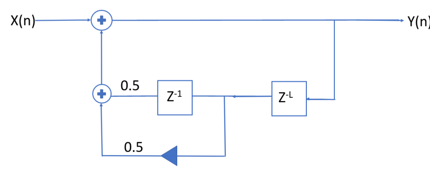

Figure 2: Block diagram with filter

equation which can be given as:-

$$y(n) = x(n) + \frac{y[n - L] + y[n - L - 1]}{2}$$

So we can see that $\frac{y[n-L]+y[n-L-1]}{2}$ is basically a 2 point moving average low pass filter. Now if we lets take the z transform of this equation which will result into :-

$$\frac{Y(z)}{X(z)} = \frac{1}{1 - 0.5z^{-L} - 0.5z^{-1(L+1)}}$$

So basically this is the required transfer function of the karplus strong algorithm.

1.1 String realization

Now lets try to analyse each step that is taking place. So in order to produce a string like sound with the help of karplus algorithm we will start our journey with the help of a white noise burst which is given as below:-

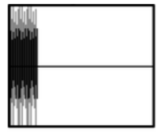

Figure 3: Input signal

So we can see that for a certain duration we are considering the noise and after that we have done zero padding which is important to make the signal periodic and to increase the no. of samples.This noise is then pass through the linear delay line which stretches the signal and then it is pass through the low-pass filter which help in procuring the required sound. The final output will a vibrating string sound. The time domain signal of this sound is given as below:-

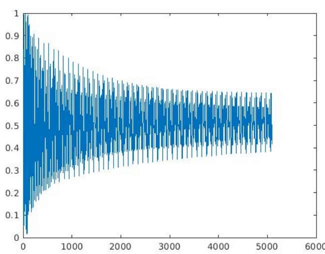

Figure 4: String sound

Now lets try to use the above concepts to produce a more technical guitar sound which involves chords. For that we have to introduce delay to each stings corresponding to the frets position of those string on the guitar.The length of delay line in samples which is tied to the desired fundamental frequency is defined as:-

$$L = \frac{F_s}{F_o}$$

Where F_s is the sampling frequency and F_o is the fundamental frequency. We have tried to produce the sound of G major chord which corresponds to 6^{th} string 3^{rd} fret, 5^{th} string 2^{nd} fret and 1^{st} string 3^{rd} fret. Now considering these string and the fret value we will be incorporating the respective delays and in guitar tuning we consider that each fret along a guitar's neck allows the player to play a half tone higher, or a note whose first harmonic is $2^{1/12}$ higher. This result in a production of a note which is basically a combination of all these harmonics. The resultant frequency domain and time domain signals are as given below:-

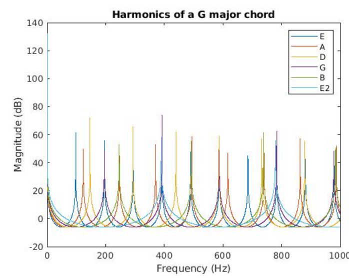

Figure 5: chord harmonics

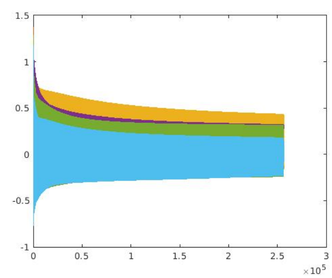

Figure 6: chord sound

2 Improvisation of Karplus Strong Synthesis

Smith recognized that the Karplus-Strong algorithm was a lossy digital waveguide in physical terms. The low pass filter represented the loss of energy on each string vibration cycle, and it was frequency-dependent, producing a higher damping at higher frequencies, just as in the case of real string. The difference between the Karplus-Strong algorithm and the previous model of plucked string (modeled with a bidirectional waveguide) was that the white noise burst had to be physically interpreted as the string was plucked with random displacement and velocity all along its length and not only in one point (as in the real world).

2.1 Digital Wave guide

The vibration of a string can be modeled simply using a digital wave-guide. This consists of two delay lines representing two travelling waves moving in opposite directions. By summing the values at a certain location along the delay lines at every time-step, we obtain a waveform. This waveform is the sound heard with the pickup point placed at that relative location. The delay elements are initialized with a shape corresponding to the initial displacement of the string. For simplicity triangular wave is assumed to represent the waveform of vibration created even though in reality the initial displacement of a plucked string will not be shaped exactly like a triangle. Simply using two delay lines in this fashion would require arbitrarily long delay lines depending on the length of the desired output. By feeding the delay lines into each other a system can be created that can run for an arbitrary amount of time using fixed size delay elements.

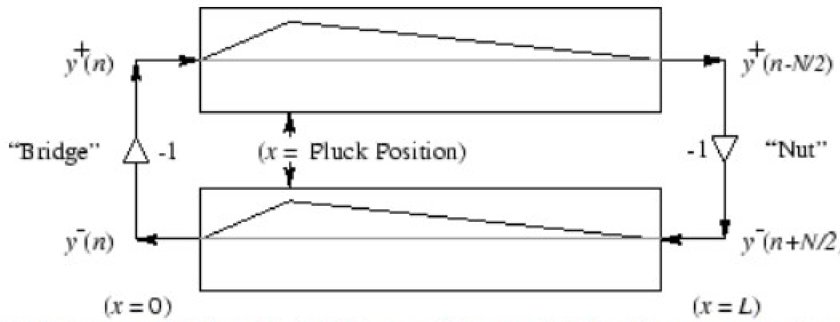

Figure 1: Digital waveguide with initial conditions of delay lines set to triangular waves.

In modeling a guitar it is important to note that the ends of the string are rigidly terminated, so the waves reflect at either end of the string. This effect can be modelled by negating each sample after it reaches the end of a delay line, before feeding it into the next delay line, as shown in Figure 1. Finally, we must add an attenuation factor. Without the attenuation factor, the model described up until now results in ideal string vibration that never decays. In the real world, due to friction and air resistance, the amplitude of the string vibrations decay over time, so it is important to model this effect in the digital wave-guide. To attenuate the output we simply add a damping factor at the ends of the delay lines so that the values are damped before being fed into the other delay line.The damping factor should be such that the decay of the output samples shouldn't be too slow or too rapid , an optimal damping factors is an ideal way to model the wave-guide.

The length of the delay lines controls the frequency of oscillation, and consequently the pitch of the output signal. This corresponds to fretting a string on a guitar. Fretting a string limits the vibration to a certain length of the string. This changes the wavelength of the travelling waves, which in turn changes the pitch of the sound. Due to the looping nature of wave-guide and the lack of additional input the output at every period is the same except attenuated slightly. Therefore the overall output will be periodic with a period depending to the length of the delay line. Therefore, if the desired frequency of the output is f and the sampling frequency is fs we set each delay line length to N/2 where N = fs/f.

The sound synthesized by this model sounds very artificial. It does nothing to account for the timbre of the instrument, and modeling the string pluck as a triangle wave is not very accurate. In addition, it does not take into account the fact that a real string vibrates in both the horizontal and vertical planes and interacts with the other strings on the guitar. Despite this, it is important to note that it does get a lot right. The damping of the string depends on the frequency - low pitched notes have a lot of sustain whereas high frequency notes attenuate very rapidly. The digital wave-guide model simulates this frequency dependant damping effect quite well. It also does a good job creating audible harmonics present in the sound of any stringed instrument. While the basic digital wave-guide plucked string model shows good experimental results in simulating a generic plucked string, special considerations for the specific instrument being modelled must also be taken into account. Although the interaction between the strings and frets cannot be modelled.

One problem with implementing this system is that the size of the delay lines must be an integer to create a digital filter. If we wish to always use a set sampling frequency, then the delay line lengths will not always be integers. For example to generate the note A4, if our sampling frequency = 44.1 kHz the delay line length would be 44100/370 = 119.189. In this case if we set the delay line length to 119 the output would be of frequency 44100/119 = 370.588. Thus, the resulting output is slightly out of tune. This effect gets greater as the frequency is increased. Another problem with creating this system in the time domain is that it is computationally expensive.In general many computational results done in previous research shows Synthesizing a three second tone takes over one minute.

2.2 Digital Wave guide Implementation Using Digital Filtering Techniques

To simplify the implementation of the wave guide, the two delay lines can be combined into one, and the damping values at the terminations can be lumped together in the feedback loop (See Figure 8). The -1 multipliers cancel each other out, and the two delay lines can be combined leaving only a length N delay line and the damping factors. The damping factors at each delay can then be lumped together into one damping factor.

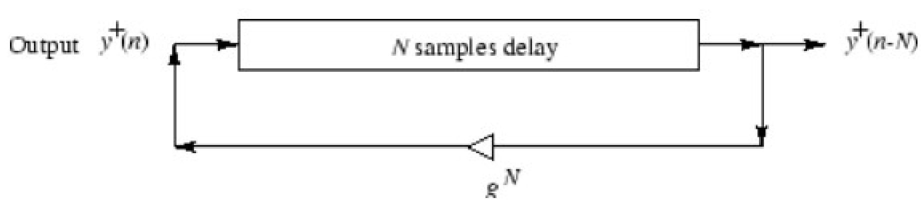

Figure 7: Simplified digital wave guide after combining delay lines and damping factors

Now the model closely resembles the model originally proposed by Karplus and Strong.However, in a real guitar not all frequencies will decay at equal rates. Therefore, for further realism the lumped damping factor is replaced by a 'loop filter' that damps each frequency differently. This loop filter always has a low pass characteristic to it. In the Karplus-Strong model this loop filter is a single zero FIR filter that averages the Nth and (N+1)th sample. This corresponds to the following difference equation:

$$Y[k] = .5^*(Y[k-N] + Y[k-N-1]) \tag{1}$$

Another difference in the Karplus-Strong model is that white noise is used as the initial conditions. The periodic nature of the filter creates a steady state output that is of the proper frequency regardless of the initial conditions. Using white noise it is very difficult to accurately reproduce the attack portion of a guitar pluck. To synthesize the attack accurately we need to use another approach other than the white noise.

To simulate the pluck position on the instrument using the simplified model, we can feed the input into an order M comb filter before feeding it into the Karplus-Strong wave-guide. The order M is a fraction of N, where N is the length of the delay line, and it determines where the string excitation is applied along the delay line.

To fix the fractional delay problem associated with a digital wave-guide, it is necessary to interpolate the value at the fractional point along the delay line. To implement the fractional delay a Lagrange filter with necessary order can be used (3rd order in this case) to give the desired sample to get the desired output frequency. This is a linear-phase filter so it doesn't distort the output. The Lagrange Interpolation filter is implemented based on the thesis provided by Standford CCRMA.

2.3 Loop Filter Design

To accurately model an acoustic guitar, it is necessary to create a loop filter that damps the different harmonics of the fundamental frequency in the same way a real guitar would. This accounts for the effect of the guitar body on the plucked string sound and begins to give the model a timbre similar to that of a real instrument. This can be done by the procedure presented by Karjalainen, Valimaki and Janosy to create a loop filter based on the recording of a guitar. The algorithm consists of fitting a straight line to the temporal envelopes of a number of early harmonics then using the slopes of the lines to estimate the attenuation factors for those harmonics. Figure 7 shows the temporal envelopes and figure 8 shows the slopes

One problem it has that initially our desired frequency response was a brick-wall filter. This was because we computed the attenuation factors only for the early harmonics and we used zero for the remaining attenuation factors. Therefore our frequency response was essentially an ideal low-pass filter. The result is that our loop filter had a large gain in the transition band. Moreover, an ideal low-pass filter is not even desirable because we want to retain the higher frequencies but they should decay more rapidly than the lower frequencies. Therefore, we need to compute the attenuation factors of the frequencies above the early harmonics based upon nonzero slopes that decreased linearly as the frequency increased. With a little tweaking we achieved results very similar to those of

Karjalainen, Välimäki, and Jánosy. However, while they used an iterative approach that weighted the early harmonics more heavily we simply used invfreqz to generate the numerator and denominator of the IIR filter. The resulting filter has the following transfer function:

$$H(z) = \frac{0.8995 + 0.1087z^{-1}}{1 + 0.0136z^{-1}} \tag{2}$$

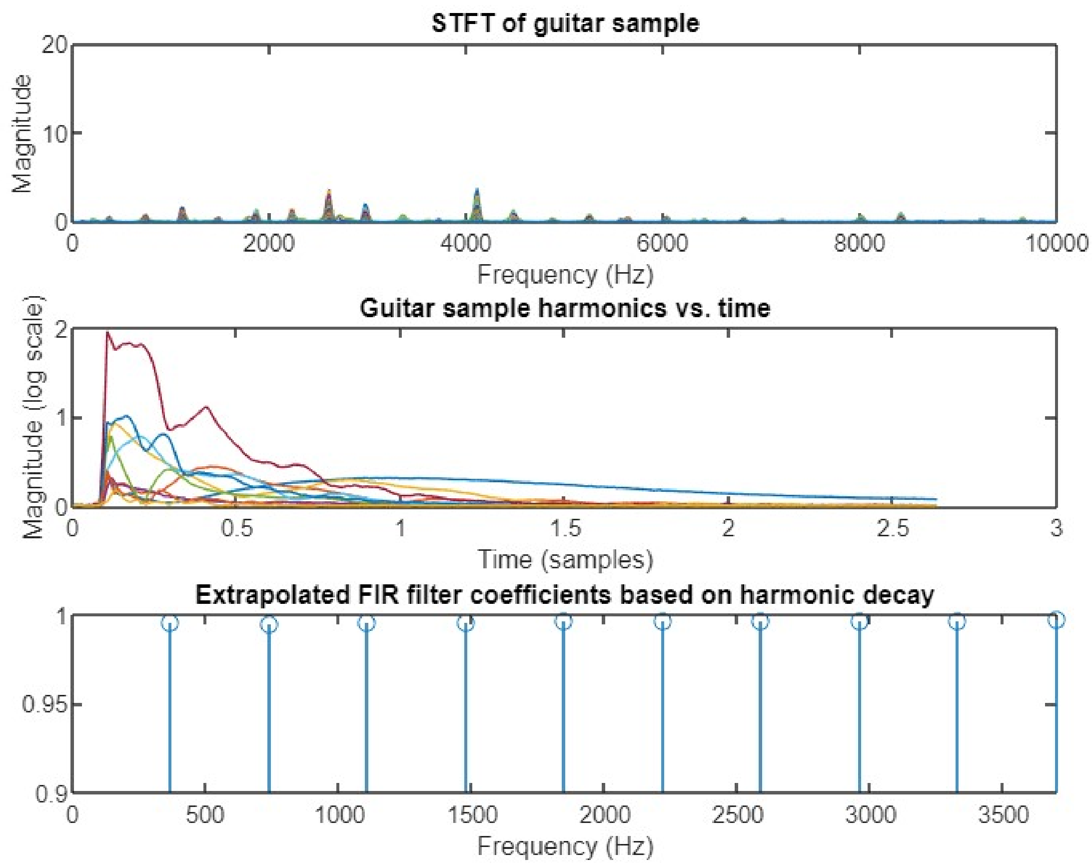

Figure 8: STFT of early harmonics of recorded guitar sound

2.4 Generating Excitation Signals via Inverse Filtering

Once the loop filter has been determined, an excitation signal that is a more accurate model of an actual guitar string pluck can be generated from a recording. This involves putting the recorded signal through the inverse wave-guide filter

$$A(z) = 1 - H(z)z^{-N} \tag{3}$$

where H(z) is the loop filter designed above.

The excitation has a noticeable effect on the attack within the first 500ms (22500 samples). It looks very similar to the attack in the recorded waveform. The difference between the two signals becomes apparent after the attack dies away. The original signal attenuates more rapidly and is missing the very low frequency components of the original sound. Despite this, it should be almost identical to the original note.

Overall, using the excitation signals generated using this technique offers a large improvement in sound quality over white noise or some other artificial excitation signal as it is based on the actual excitation that is applied to a guitar string. It takes into account the effect that the guitar body has on the string excitation, resulting in an even more accurate model. This process can be used to simulate different plucking techniques as well, resulting in a an even more versatile model. The pick signals sound very harsh when compared to the others.

2.5 Bringing it all together

Below is the block diagram of the final filter we have designed to synthesize an acoustic guitar:

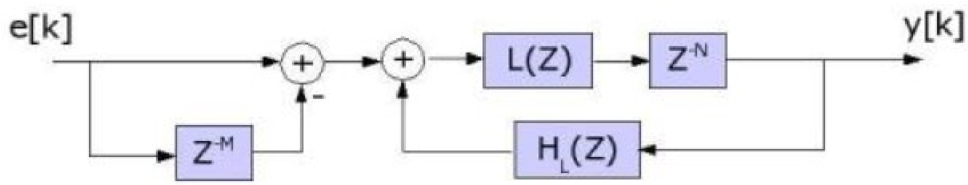

Figure 13: Block diagram of the final filter.

Figure 9: Caption

One can see the length N delay line from the original Karplus-Strong digital wave-guide model. The Lagrange interpolation filter (L(Z)) feeds into the delay line for proper tuning. It also has an improved loop filter (HL(Z)) based on recordings from an actual guitar. A comb filter has been placed at the input (the left-hand portion of the block diagram) to simulate the effect of plucking position on guitar. The input to the system is an excitation signal (e[k]) obtained through inverse filtering of a guitar recording.

3 Conclusion

So we saw how we can get generate guitar string sound using the Karplus Strong algorithm and also we saw some improvements that we can do in order to make the sound clearer and more closer to that of a real life musical instrument strings. We tried to do some creativity and build a GUI model which is using the basic karplus algorithm to generate guitar string sound and the chord sound, which is as given below. During this journey we learned about a very innovative and quite engaging way to generate electronic music and also how we implement filters to process signal.

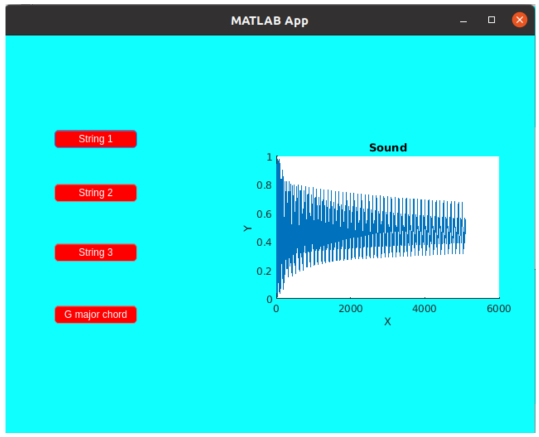

Figure 10: Our GUI using matlab APP

4 Reference

1. Smith, Julius O. Digital Wave-guide Modeling of Musical Instruments, Center for Computer Research in Music and Acoustics (CCRMA), Stanford University, $2003 - 12$ -10.

2. K. Karplus and A. Strong, "Digital synthesis of plucked string and drum timbres," Computer Music Journal, vol. 7 , no. 2, pp. $43 - 55, 1983$.

3. D.A. Jaffe and J.O. Smith, "Extensions of the Karplus-Strong plucked string algorithm," Computer Music Journal, vol. 7 , no. 2 , pp. $56 - 69, 1983$.

4. C.Roads, ed., The Music Machine, Cambridge, MA: MIT Press, 1989.

5. M. Karjalainen, V. Välimäki, and Z. Jánosy, "Towards high-quality sound synthesis of the guitar and string instruments," in Proceedings of the 1993 International Computer Music Conference, Tokyo, pp. 56-63, Computer