

Haoqing Geng

ME515 final project report

Simulation of plucked string vibration with digital waveguide model

1. Introduction

Model based sound synthesis has emerged since 70s in the last century (Hiller and Ruiz 1971a). With increasing computational power, musicians began to realize that computers might enable new ways of interaction between musicians and virtual musical instruments.

In theory, computer is able to generate any kind of sound, but it's very meaningful to model existing traditional musical instrument, as stated by Julius (Julius O. Smith, 1996). "We simply don't know very many ways to generate deeply communicative sounds from scratch".

Various computational modeling approaches have been proposed to model musical instruments. Matti Karjalainen introduced the historical evolution of physical Plucked-String model in his paper(Matti Karjalainen, 1998). He started by explained the Karplus-Strong algorithm, which is a simple computation technique that is not directly related to physics, but could synthesize acoustic sound. Then he mentioned the digital waveguide theory, introduced by Julius, making more physical related abstractions for not only string, but also more general acoustic systems.

Many kinds of musical instruments have been physically modeled, including string instrument, wind instrument, brass instrument, voice, etc. Julien Bensa introduced how physical models of piano string can be simulated from the perspective finite difference schemes and digital waveguides theory(Julien Bensa, 2003). Guitar also received lots of interest in research. Matti Karjalainen introduced new principles of modeling to make string instrument achieve higher quality in real time by estimating model parameters and excitation signal from acoustic instrument(Matti Karjalainen, 1993). Digital waveguide modeling is also used for wind instrument simulation. Remi Mignot presents the application of state-space-representations for the acoustic model of Webster-Lokshin(Rémi Mignot, 2010).

Except for musical instrument itself, acoustic space response is also a very interesting topic in this area. Acoustic reverberation has long been a problem for theater designers. Since the echoes of sound keep bouncing in the room and create more echoes constantly, it's impossible to fully compute all echoes. In the paper "50 years of artificial reverberation", different numeric approaches of efficient reverberation algorithms are introduced(Vesa Välimäki, 2012)

2. Digital waveguide model of string vibration

Digital waveguide model is a computational method to simulate physical media through which acoustic waves propagate. It constitutes a major part in modern physical modeling audio synthesizers. Yamaha used this model to develop a lot of successful synthesizers.

2.1 Physical assumptions

1. The mass of the string per unit length is constant. The string is perfectly elastic and does not offer resistance to bending.
2. The tension caused by stretching the string is so large that gravitational effects can be neglected.
3. Every particle on the string moves strictly vertically.

2.2 Digitization of basic string vibration model

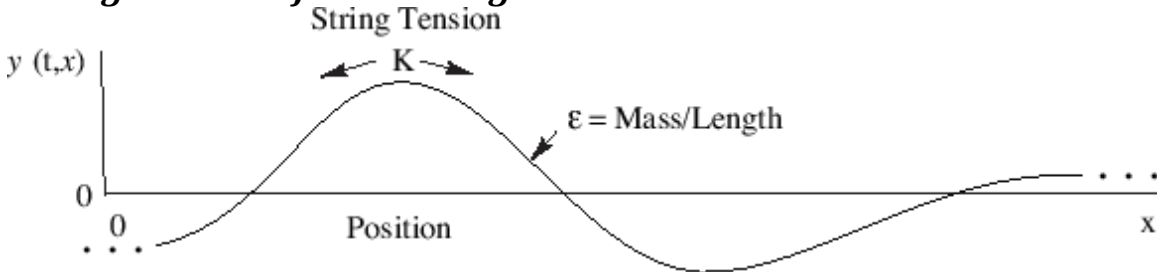


Fig 1: string model

$$Ky'' = \epsilon \ddot{y}$$

K	\triangleq	string tension	y	\triangleq	$y(t, x)$
ϵ	\triangleq	linear mass density	\dot{y}	\triangleq	$\frac{\partial}{\partial t} y(t, x)$
y	\triangleq	string displacement	y'	\triangleq	$\frac{\partial}{\partial x} y(t, x)$

Above is the PDE that describes the vibration of an ideal string. Since the string is continuous in space and vibrates continuously in time. To model the vibration in computer, it's necessary to digitalize the system in both space and time. In space, we cut the string into finite number of unit masses, as shown in the following figure:

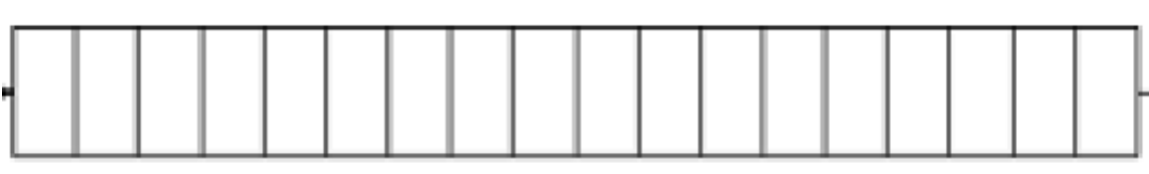


Fig 2: digitalized string with units of mass

In time, we snap shot 44100 samples for 1 second, which is the sample standard for CD quality audio.

After digitization, the wave equation $y(t,x)$ is turned into $y(nT,mX)$, T and X denotes unit time and unit mass.



Fig 3: Plucked string as initial condition

2.3 Two ways of solving the problem

As shown in figure 3, the problem is: given the boundary condition of fixed ends, and the initial condition of people plucking a string at a certain point in string to a certain height, what is the time response of the string?

There are two approaches to this question.

First, for each mass X , write an ODE equation of motion based on Newton's second law and string's tension, density. Obviously, m ODE systems will have to be solved, and linearization needs to be done to every equation. The amount of calculation is huge.

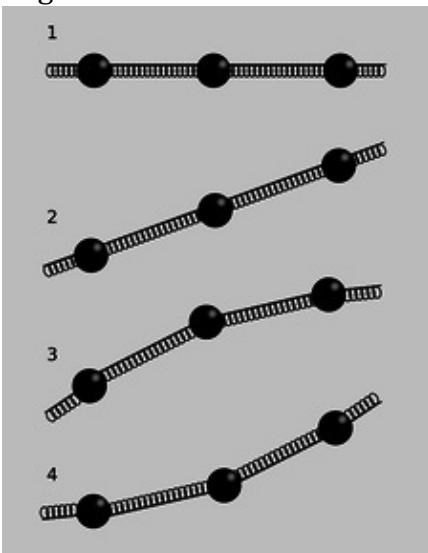


Fig 4: first approach, one ODE for each mass

Second, the wave solution of string vibration. Instead of thinking in terms of the individual vibration of each mass, the wave solution considers the vibration as the addition of two waves, which originates from the plucking point, going right and left in the speed of c (the speed wave travels in string). And the wave will be bounced back at each end of string, as shown in the following figure:

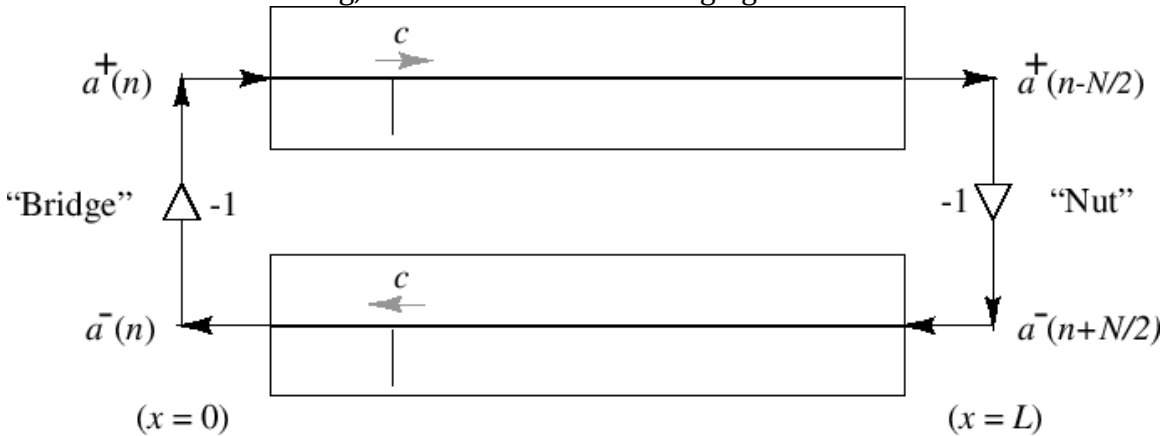


Fig 5: Two waves going opposite directions and bounce back at ends of spring

Below are the equation describing the digitalized wave solution of string vibration:

$$y(t, x) = y_r(t - x/c) + y_l(t + x/c).$$

$$y_r(nT - mX/c) + y_l(nT + mX/c)$$

$$y_r(nT - mT) + y_l(nT + mT)$$

Analytically, this kind of wave solution reduces the PDE into 2 ODEs, making it easier to solve. In digital calculation, it consists of just simple delay lines and addition, making this a very efficient algorithm to model string plucking.

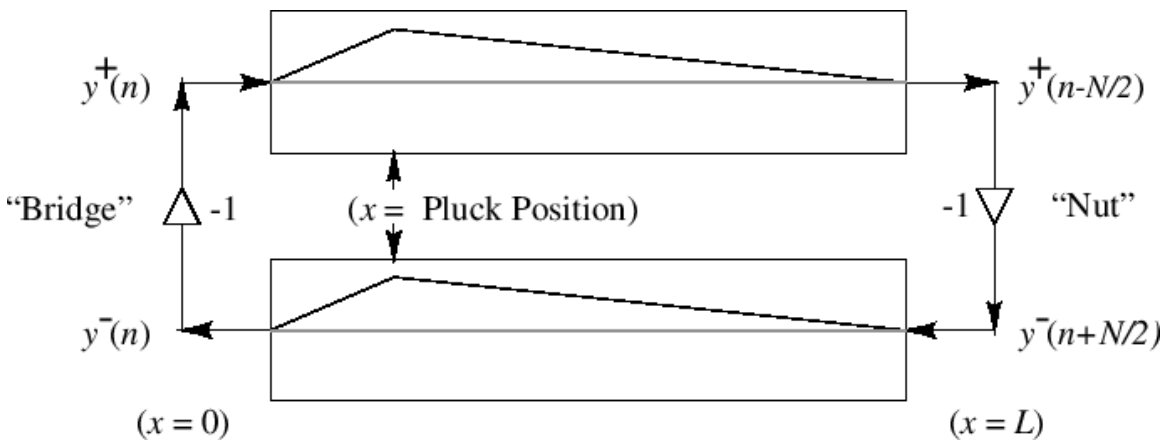


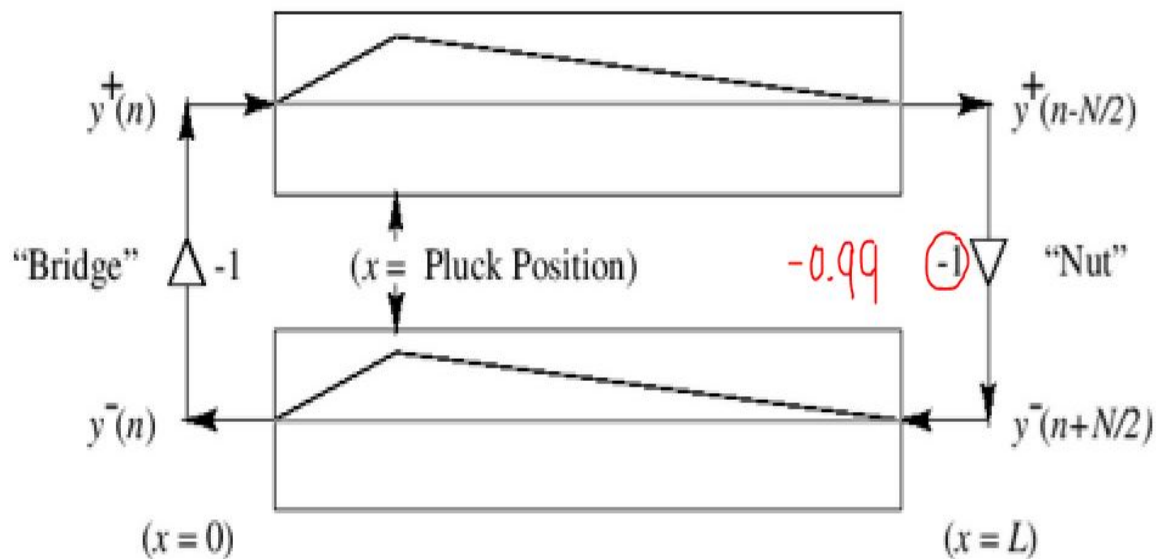
Fig 6: Initial condition of plucked string

With the above initial condition, the result can be easily done by initial data moving forward as directed in the above figure 6, and sum up the two waves at every point of string, we could get the position of every point in ideal string vibrating at any time.

2.4 Simplification of damping

The ideal string with no damping will just keep vibrating in the same amplitude and perfectly periodic, but in reality, the wave is damped between every two mass. However, it's not necessary to compute the damping for every couple of masses. As a result, change the -1 factor at the end of the string into -0.99 can effectively avoid the computational cost and round off error in computer, while also achieving the the goal of decay of vibration amplitude.

$$Ky'' = \epsilon \ddot{y} + \mu \dot{y}$$



2.5 Frequency dependent damping

In reality, damping increases with frequency, below is a simple modification in the equation to consider frequency dependent damping:

$$Ky'' = \epsilon \ddot{y} + \mu \dot{y} + \mu_2 \dot{y}''.$$

To achieve this effect in the computational model, the simple -0.99 has to be replaced by a digital filter that have various gains in different frequencies, like the Bode plot we've learnt. Due to time constraint, I didn't add this feature into my matlab code.

3. Matlab code:

```
%Parameters to control for this ideal string

strLength = 50;%String length, will affect the frequency of sound
r = zeros(strLength,1)';%right travelling wave initialization
l = zeros(strLength,1)';%left travelling wave initialization
SampleRate = 44100; %CD sound quality 44100 Hz
time = 2;%The time span of one pluck sound, 2 seconds
f = zeros(time*SampleRate,strLength);%The sum of right and left
travelling sound
h = 0.4;%The height of initial condition at the plucking point of
string
x = strLength*0.6;%The point where vibration is plucked
hear = strLength*0.9;%The point where vibration is picked up in string

%initial condition of string
%Note that boundary condition is that both ends of string is fixed
%ri denotes the points from left end to the plucking point
%li denotes the points from plucking point to the right end of string
for ri = 2:x
    k = h/(x-1);%slope of plucked string
    r(ri) = k*(ri-1);
    l(ri) = r(ri);
end

for ri = x+1:strLength
    k = -h/(strLength-x);
    r(ri) = h+k*(ri-x);
    l(ri) = r(ri);
end
figure(1)
plot(r)
figure(2)
plot(l)

%Start
for t = 1:time*SampleRate%2 seconds here
    %When the wave reaches the end of string, it will be bounced back
with
    %a 1 percent loss of amplitude due to damping, this is a
simplification of
    %damping
    rlast = r(strLength);
    r = [(-1)*l(1),r(1:strLength-1)];
    l = [l(2:strLength),rlast*(-0.99)];
    for j = 1:strLength%at each time sample point, every string point
gets
        %addition of right and left wave vibration at this point
        f(t,j) = r(j)+l(j);
    end
    %The following commented code can be used to see string vibrating
in
    %real time
    % figure(3)
    % subplot(3,1,1), plot(r)
    % axis([0 100 -0.5 0.5])
    % subplot(3,1,2), plot(l)
```

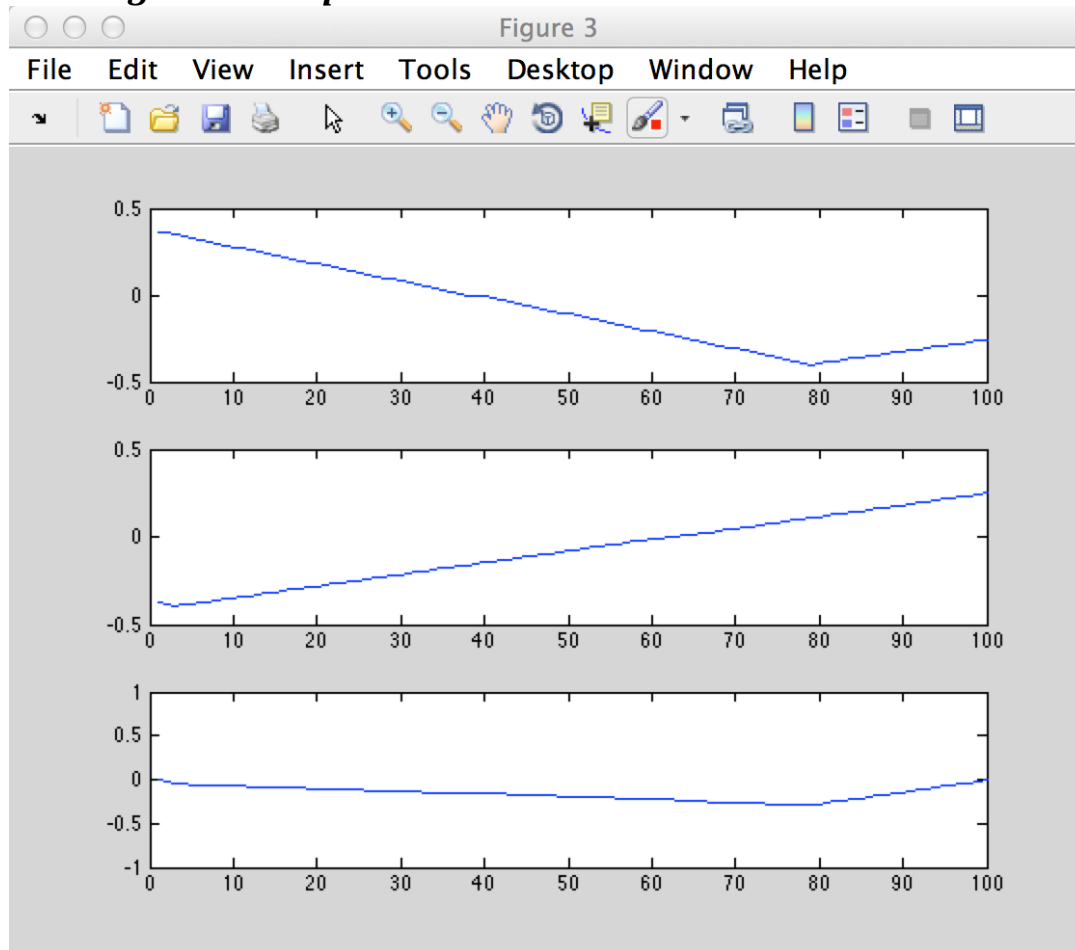
```

% axis([0 100 -0.5 0.5])
% subplot(3,1,3), plot(f(t,:))
% axis([0 100 -1 1])
% getframe;
end
figure(3)
plot(f(:,hear))%wave form of sound in two seconds
figure(4)
spectrogram(f(:,hear),128,120,128,1E3); %spectral analysis
sound(f(:,hear),44100)

```

4. Results and discussion:

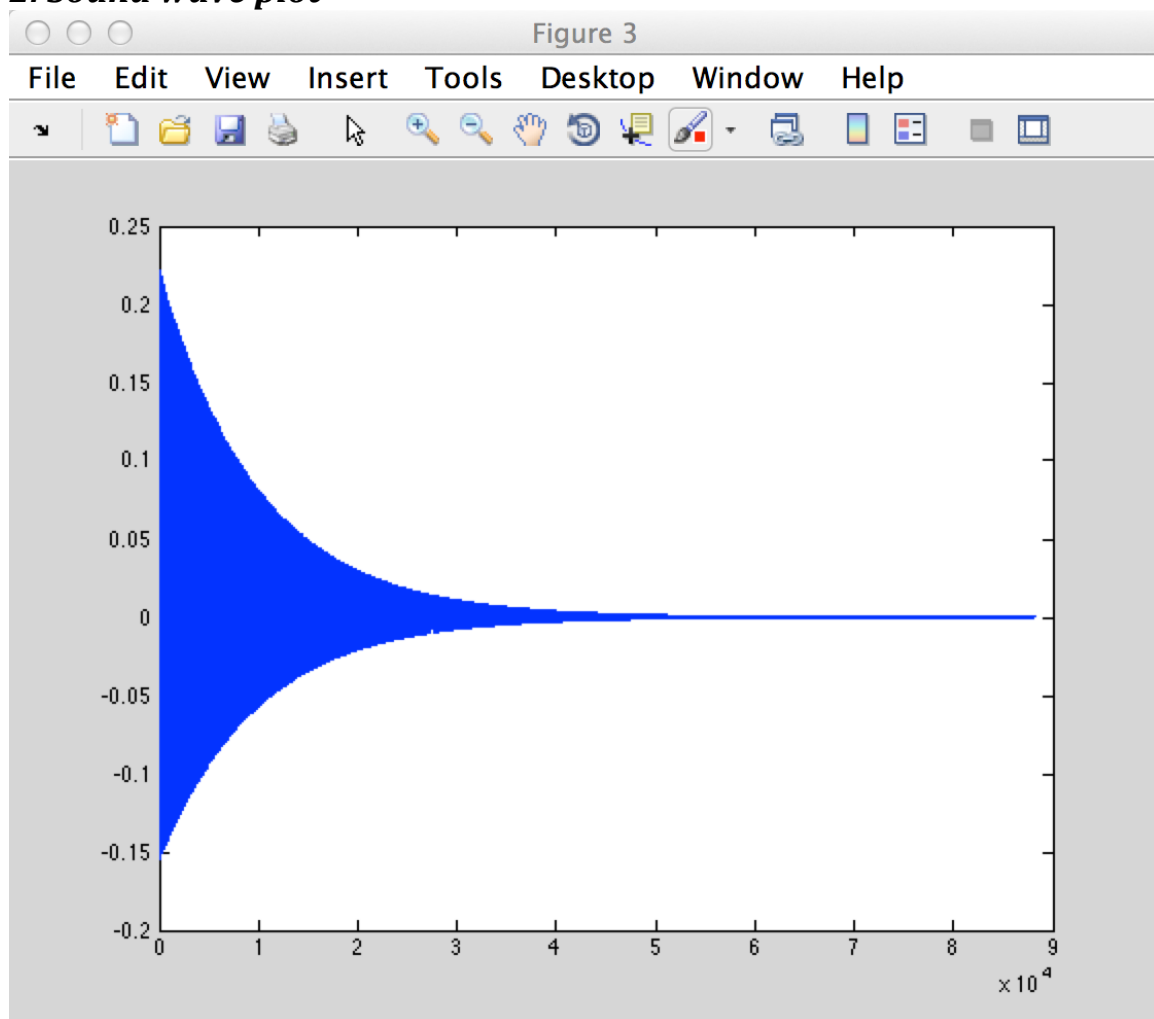
1. String vibration plot video



In the above figure, the first plot shows the right going wave, the second shows the left going wave, and the third one is the string vibration.

Compared the time framed video from matlab with the high speed camera video from youtube video, this simulation in matlab works very well.

2. Sound wave plot

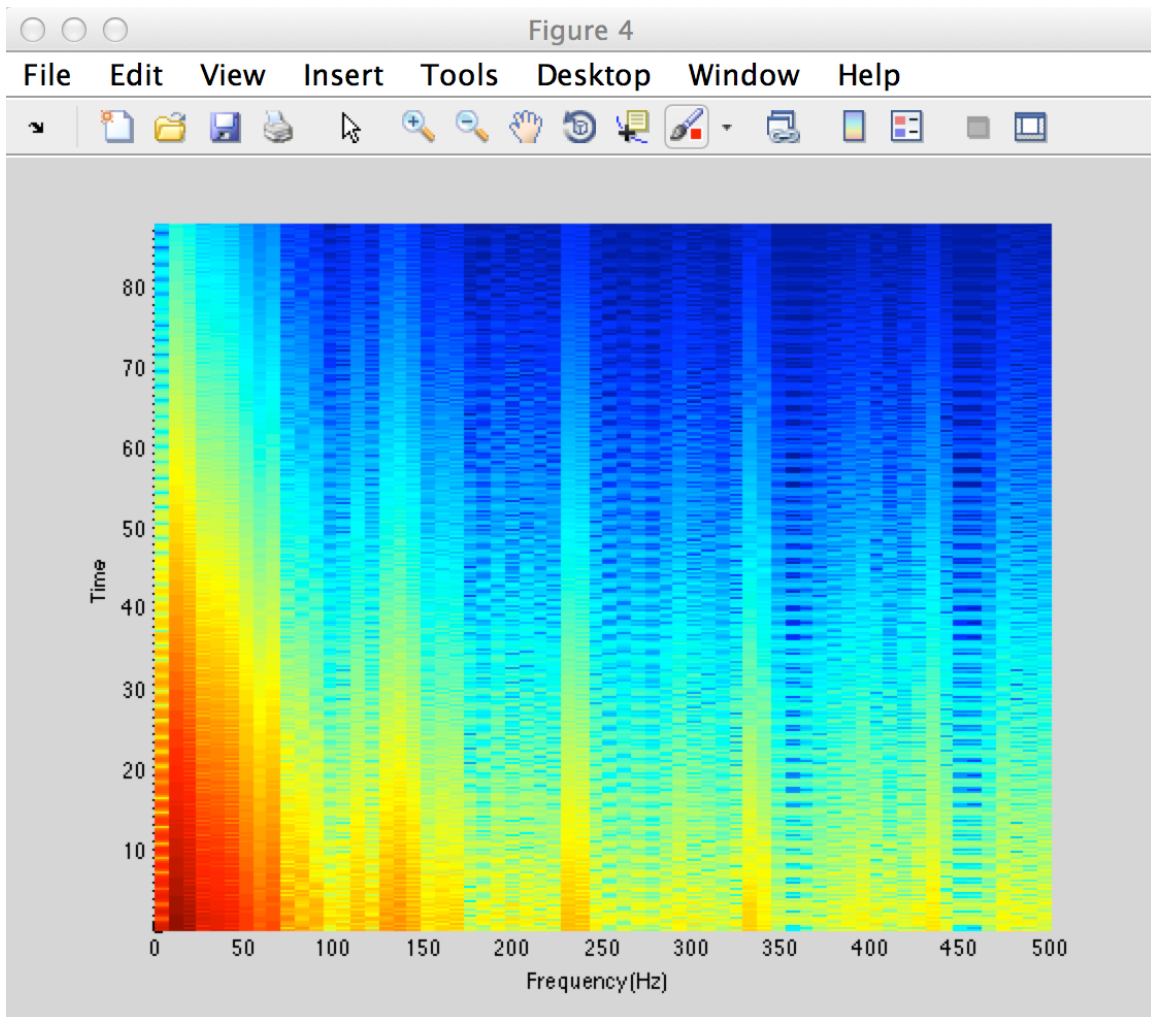


This plot is the vibration of a certain mass in the string, which produces the output sound wave to be played. Compare this sound to a real guitar sound, it sounds more dry, probably due to the following reasons:

1. The damping is not frequency dependent like in real guitar.
2. The initial condition is slightly different, when a man uses his finger to pluck the string, the initial condition of a string is not strictly a straight line with a slope, but rather a bent line.

3. The real guitar has a wood body to provide reverberations, which produces sound summed with its echoes.
4. In real guitar, every point on string is affecting the output sound in 3 dimensional space, but in this code, only one point of vibration is picked up as output sound.

3. Spectral analysis



This is a spectral analysis of the output sound, it can be seen that as time goes on, the higher frequencies decayed into almost zero. It fits well with the analytic solution in the textbook book.

5. Conclusion

It can be seen that digital waveguide theory provides a very efficient computational model to simulate string vibration from the approach of acoustic waves propagating, more digital filter designs can be made to simulate frequency dependent damping.

6. Reference

1. Hiller, L., and P. Ruiz. 1971a. "Synthesizing Musical Sounds by Solving the Wave Equation for Vibrating Objects: Part 1." *Journal of the Audio Engineering Society* 19(6):462-470.
2. Smith, J.O. 1996. "Physical Modeling Synthesis Update." *Computer Music Journal* 20(2):44-56.
3. Matti Karjalainen, Vesa Välimäki and Tero Tolonen 1998. "Plucked-String Models: From the Karplus-Strong Algorithm to Digital Waveguides and beyond." *Computer Music Journal* 22(3):17-32.
4. Julien Bensa, Stefan Bilbao, Richard Kronland-Martinet, and Julius O. Smith III. "The simulation of piano string vibration: From physical models to finite difference schemes and digital waveguides" 2003 Acoustical Society of America.
5. M. Karjalainen, V. Valimaki, Z. J'anosy . "Towards high-quality sound synthesis of the guitar and string instruments" International Computer Music Conference, September 10-15, 1993, Tokyo, Japan
6. Rémi Mignot, Thomas Hélie, and Denis Matignon. "Digital Waveguide Modeling for Wind Instruments: Building a State-Space Representation Based on the Webster-Lokshin Model". *IEEE transactions on audio, speech and language processing*, VOL. 18, NO. 4, MAY 2010
7. Vesa Välimäki, Julian D. Parker, Lauri Savioja, Julius O. Smith Jonathan S. Abel, "Fifty years of artificial reverberation". *IEEE transactions on audio, speech and language processing*, VOL. 20, NO. 5, JULY 2012