Program to do a Fourier analysis of an ideal guitar string (only a model) PH 398 EMI

Tobias Schmitter

12th December 2000

1 Introduction

The motivation for this project was to look at the physical system of an ideal guitar string from a theoretical point of view. At the center of the project is the analysis of the harmonic content of a picked string and how to put this theory in a little program illustrating the physical impact of the theory in a simple way and making it visual and hearable. First we look at the mathematical background and then at the physical system. At the end you find a documentation of the program itself. In the Appendix you find some plots made with this program.

2 A little bit of theory

2.1 Fourier analysis [1][2]

The French mathematician Joseph Fourier (1768-1830) showed, based on Taylor's Theorem (Taylor; 1685-1731), that every well-behaved periodic function $f(x)$ can be represented by an infinite series of the following form:

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left[a_n \cdot \cos\left(\frac{2\pi}{L} nx\right) + b_n \cdot \sin\left(\frac{2\pi}{L} nx\right) \right]$$

where L is the period of $f(x)$. The so called *Fourier coefficients* a_n and b_n are defined the following way:

$$a_n := \frac{2}{L}\int_{-\frac{L}{2}}^{\frac{L}{2}} f(k) \cdot \cos\left(\frac{2\pi}{L}nk\right) dk$$

and

$$b_n := \frac{2}{L}\int_{-\frac{L}{2}}^{\frac{L}{2}} f(k) \cdot \sin\left(\frac{2\pi}{L}nk\right) dk$$

This formalism is know as *Fourier analysis* (or *harmonic analysis*)[1]. Why is this important for our task to analyse a picked ideal guitar string? As the name *harmonic analysis* implies, the Fourier analysis gives us exactly the harmonic content of the periodic function. As we will see, this is a very powerful and amazing physical property of the mathematical formalism.

2.2 The wave equation

To apply the formalism of the Fourier analysis we first look at our physical system. An ideal string (without damping and a linear potential) obeys the 2^{nd} order partial differential equation:

$$\frac{\partial^2 u(x;t)}{\partial t^2} - c^2 \frac{\partial^2 u(x;t)}{\partial x^2} = 0$$

(with c=moving-velocity of a disturbtion on the string)
This equation is also called the *wave equation.* There is a nice way of looking at the solutions of this partial differential equation. The wave equation is a so called hyperbolic equation. This means it has two sets of characteristic curves. Normally a differential equation expressed in the so called characteristic or natural coordinates is a little bit simpler. There is a recipe on how to get the characteristic curves. For the wave equation we just get [3]:

$$x + c \cdot t = \eta$$

$$x - c \cdot t = \xi$$

In these new coordinates the wave eqation is just:

$$-4c^2 \cdot \frac{\partial^2 v(\eta;\xi)}{\partial\eta\partial\xi} = 0$$

[1] For non-periodic functions there are similar equations by replacing the infinite sum by an integral. This is then called the *Fourier transform.*

Where $v(\eta;\xi)$ is a solution of the wave equation in natural coordinates. So the wave equation may immediately be integrated:

$$\frac{\partial^2 v(\eta;\xi)}{\partial\eta\partial\xi} = 0 \Rightarrow \frac{\partial v(\eta;\xi)}{\partial\xi)} = h'(\xi) \Rightarrow v(\eta;\xi) = g(\eta) + h(\xi)$$

Returning to the original coordinates $(x;t)$ we obtain the general solution for the wave equation:

$$u(x;t) = g(x + c \cdot t) + h(x - c \cdot t) \quad [3]$$

Physically, this solution describes two independent waves, propagating left and right, each without change of form and moving at speed c.
Why is this important for the Fourier analysis of a string with fixed endpoints? We look at a string with fixed endpoints and we pick this string somewhere. At $(t = 0)$ it looks like:

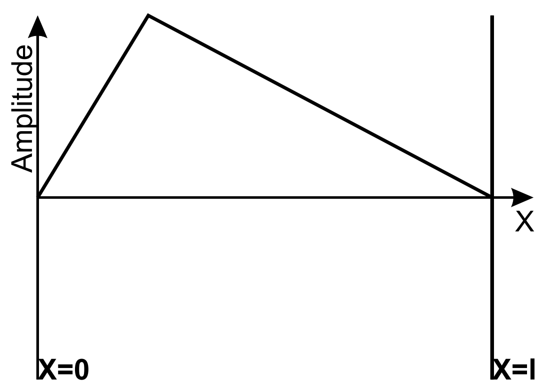


Figure 1: Ideal string picked at some point

As we have seen, the solution of the wave equation is just a superposition of left and right propagating waves. The solution has to satisfy the boundary conditions $u(x;t) \mid_{x=0} = 0$ and $u(x;t) \mid_{x=l} = 0$ for all time t. This is only possible, if g and h add just to zero at $x = 0$ and $x = l$ for all time. So if we know our initial condition (g and h) (how the string looks like) at a time $t = 0$ (plus energy conservation), we can expand this to the whole x-axis. At our initial condition we get:

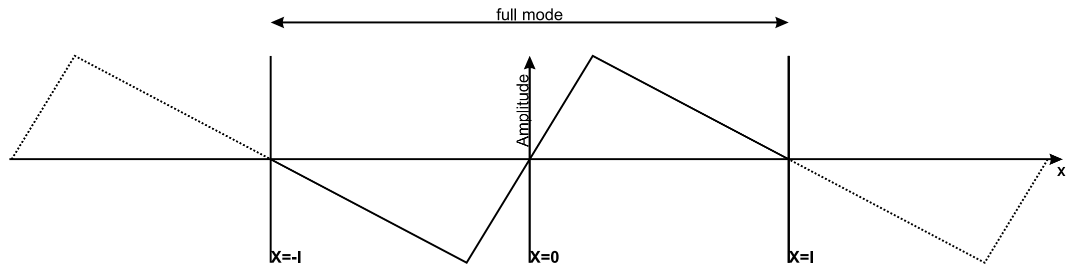


Figure 2: expansion along the x-axis

2.3 Fourier analysis of an ideal guitar string

So our physical system has the periodicity of $2l$ and we can apply the Fourier analysis to get the harmonic content of a picked string. The waveform can precisely be replicated by the following Fourier series expansion[1]:

$$u(x; t = 0) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left[a_n \cdot \cos\left(\frac{\pi n x}{l}\right) + b_n \cdot \sin\left(\frac{\pi n x}{l}\right) \right]$$

And the coefficients are determined by [1]:

$$a_0 = \frac{1}{l} \int_{-l}^{+l} u(x; t = 0) dx$$

$$a_n = \frac{1}{l} \int_{-l}^{+l} u(x; t = 0) \cos\left(\frac{\pi n x}{l}\right) dx$$

$$b_n = \frac{1}{l} \int_{-l}^{+l} u(x; t = 0) \sin\left(\frac{\pi n x}{l}\right) dx$$

So the picked guitar string is just a linear superposition of sine and cosine waves. For our guitar string the a_0 (the offset) is just zero. To fit the boundary condition $u(0, t) = 0$ and $u(l, t) = 0$ for all times t we can easily convince ourselves that all $a_n = 0$ (because the cosine is an even function and our solution is an odd function). For a more detailed discussion see [1]. So this gives us the harmonic content of the vibrating string. And the Fourier coefficients are the amplitudes of these harmonics.

3 The program: guitar

3.1 About the program

You can 'pick' a virtual ideal guitar string by clicking with the left mouse button on the graph. My program duplicates the virtual string for the calculations (not displayed) to get a full mode (**figure 2**). Then it analyses the harmonic content by determining the Fourier coefficients a_n and b_n with a very simple integration algorithm[2]. You can change the number of coefficients calculated (1 to 50).The calculated a_n are not exactly zero. The reason for that is the very simple numerical integration method. The program can then show the amplitudes of the odd and even harmonics in a linear or logarithmic graph. It can replicate the 'picked' string using the harmonics and their amplitude. Here you see how good the replication is or which coefficients vanish. And if you like, you can hear what it sounds like. For the playing you can choose a frequency (by choosing a guitar string or manual input). There are some more extra features like clipping or damping (see user interface). O.K. it doesn't sound like a guitar, but you can hear the different harmonics.

This program helps to understand the formalism of a harmonic analysis[3]. It can visualize the presence of higher harmonics depending on the location where you pick a string. You can easily see and hear the difference between a string picked in the middle and picked near the end. You can hear the "richness" of harmonics and even if you cannot separate them clearly from the original tone you can hear them. And by switching higher harmonics on and off you can easily convince yourself, that your ear is a logarithmic device. Although the amplitudes for higher harmonics are often very small compared to the first harmonic you still hear the difference. This program gives you an idea that a tone (e.g. an A_4 on a guitar or on a piano) is determined by the first harmonic, but the "color" of a instrument depends on the presence of higher harmonics[4].

The whole program is written for LabWindows/CVI (version 5.0) and the

[2] just by adding up bars

[3] This formalism has a lot of other application e.g. part of the data reduction in the mp3 standard. You have to store only some amplitudes to describe a waveform pretty well instead of a complete digitized signal like on a CD where you get 44100 (times two for stereo) data points every second.

[4] This is true for ideal instruments, but for real instruments there are other effects that produce not only harmonics, but all kind of different frequencies.

code is mainly C (plus some special syntax for the user interface provided by LabWindows/CVI). Part of the program code is based on the program 'Fourier' written by Steve Errede and Jack Boparai. The basic routine to write a wave file was written by Steve Clayton.

3.2 The user interface

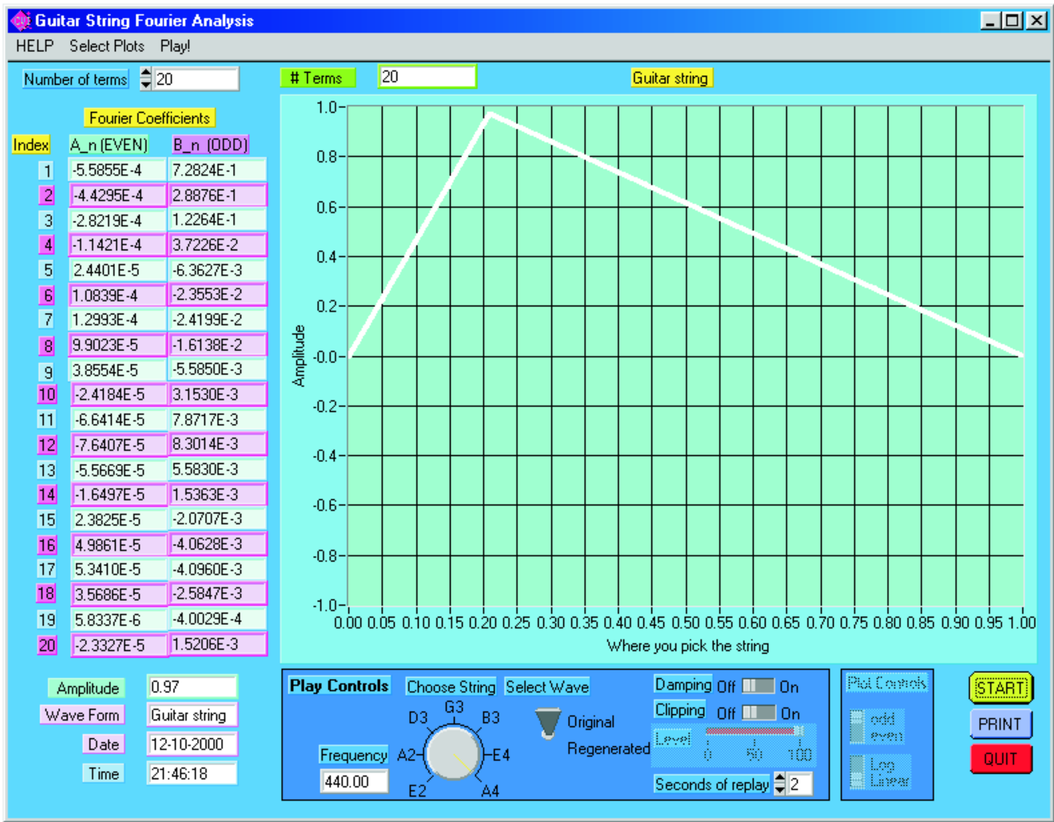


Figure 3: the user interface

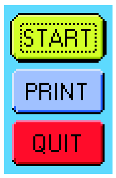
picking the virtual string

First you have to pick the virtual string by simply clicking somewhere on the graph. The graph updates immediately.

Number of terms

You can change the number of the odd and even calculated and played Fourier amplitudes with this control.

main control

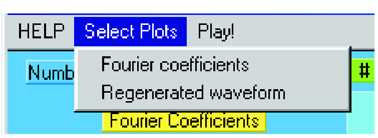


- *START*: starts the Fourier analysis of the actual graph. You always have to start the analysis before you can plot any graphs or play the regenerated wave form. When the calculations are complete, the table on the left will be updated.
- *PRINT*: prints the whole window to the standard printer
- *QUIT*: quits the program. IMPORTANT: If the program is playing a wave file or doing some calculations, the program will quit after these processes are done!

Help menu

Here you find a short help to the most important functions of the program.

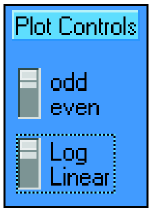
Select plot menu



- *Fourier coefficients*: displays the absolute value of the Fourier coefficients in a graph (see Appendix A **figure 5 & 6**). In this mode the Plot Controls (see: Plot Controls) are active.

- *Regenerated waveform*: Displays for 4 seconds all calculated harmonics in one graph (see Appendix A **figure 8**). After that the Fourier series is calculated and is displayed for each single term. So you can see the developing process of the series (see Appendix A **figure 7**).

Plot controls



These controls are only active when Fourier coefficients are plotted (see *Select plot* menu).

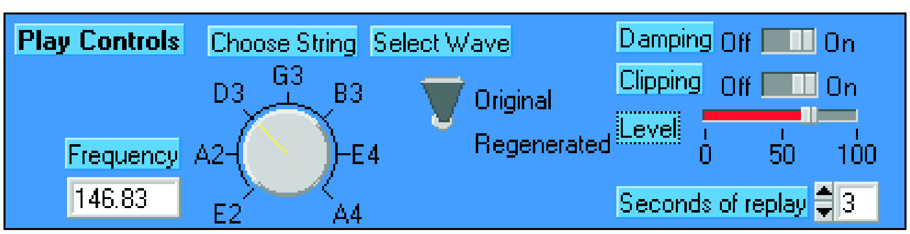
- *odd-even*: switches between the plots of the even (a_n) and the odd (b_n) Fourier coefficients.

- *Log-Linear*: switches between a logarithmic and a linear scale of the y-axis (see Appendix A **figure 5 & 6**). Our ear is a logarithmic device, so the logarithmic plot gives you an idea which harmonics you can hear and which not.

Play! menu

This menu item plays a wave file referring to the Play controls (see: Play controls)

Play controls



- *Frequency*: You can type in a frequency for the first harmonic of the played waveform (limited to 20000Hz). This textbox is also updated when you choose a string with the string selector (see: Choose string)
- *Choose string*: selector to choose the frequency of the played waveform by selecting the corresponding string (on a normal guitar). The A4 selection is not a string on a guitar, but the standard frequency of 440 Hz. (This is the definition of our tone system)
- *Select wave*: switches between the original waveform and the regenerated waveform (the rebuild Fourier series with the number of terms you have chosen).
- *Damping*: damps the amplitude of the played wave file (amplitude decays exponentially).
- *Clipping*: switch to activate the clipping feature. These feature cuts off the wave file at a certain amplitude (see clipping level) and is then amplified to the original amplitude. IMPORTANT: The clipping does not effect the Fourier analysis!
- *Clipping level*: adjusts the clipping level in percent of the maximum amplitude. This is visualized by a thin red line in the graph (see Appendix A **figure 9**). All absolute values of the amplitude bigger than this level are cut off. (This control is only active, if *Clipping* is ON!)
- *Seconds of replay*: determines how long the waveform is repeated and played through the sound card. IMPORTANT: While a wave file is played through the sound card, no other commands are executed! A longer replay will require longer calculations! Depending on the CPU, this could be several seconds!

References

[1] Errede, Steve, Fourier Analysis I, 2000

[2] Teubner-Taschenbuch der Mathematik (Bronstein), 1996

[3] Goldbart, Paul, Phycs 498 MMA, 2000

[4] specifications of a wave file, http://www.intersrv.com/ dcross/wavio.html

A Graphs produced with this program

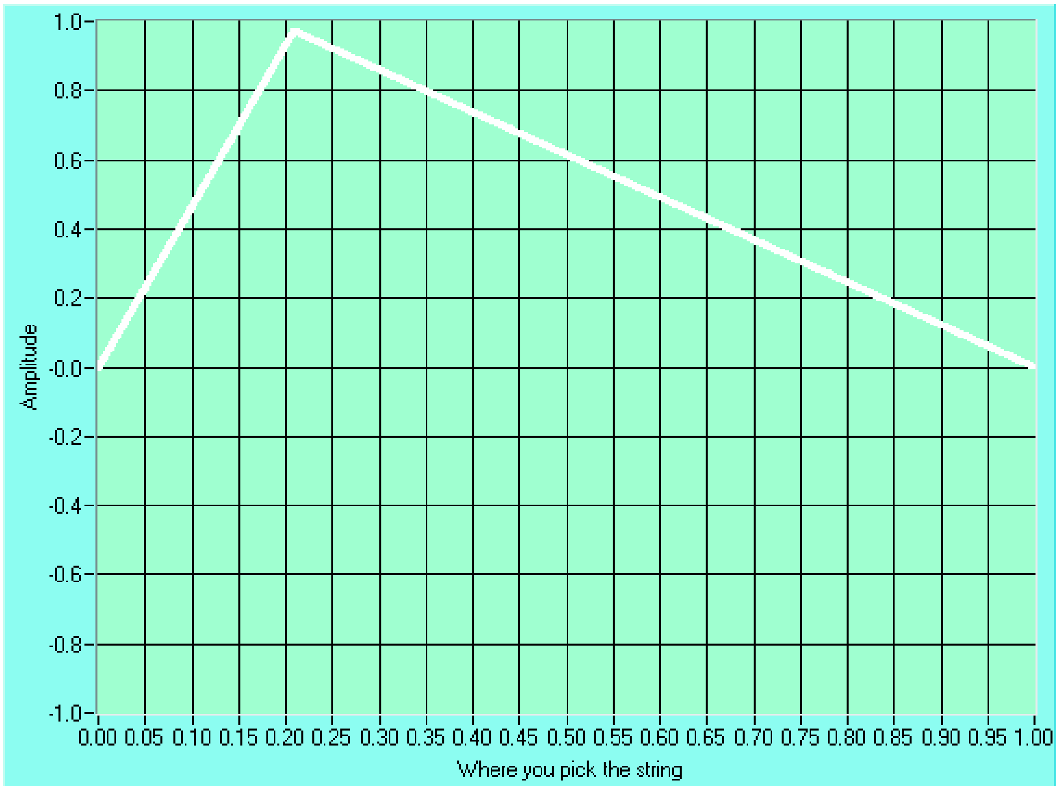


Figure 4: The graph of a "picked" ideal guitar string

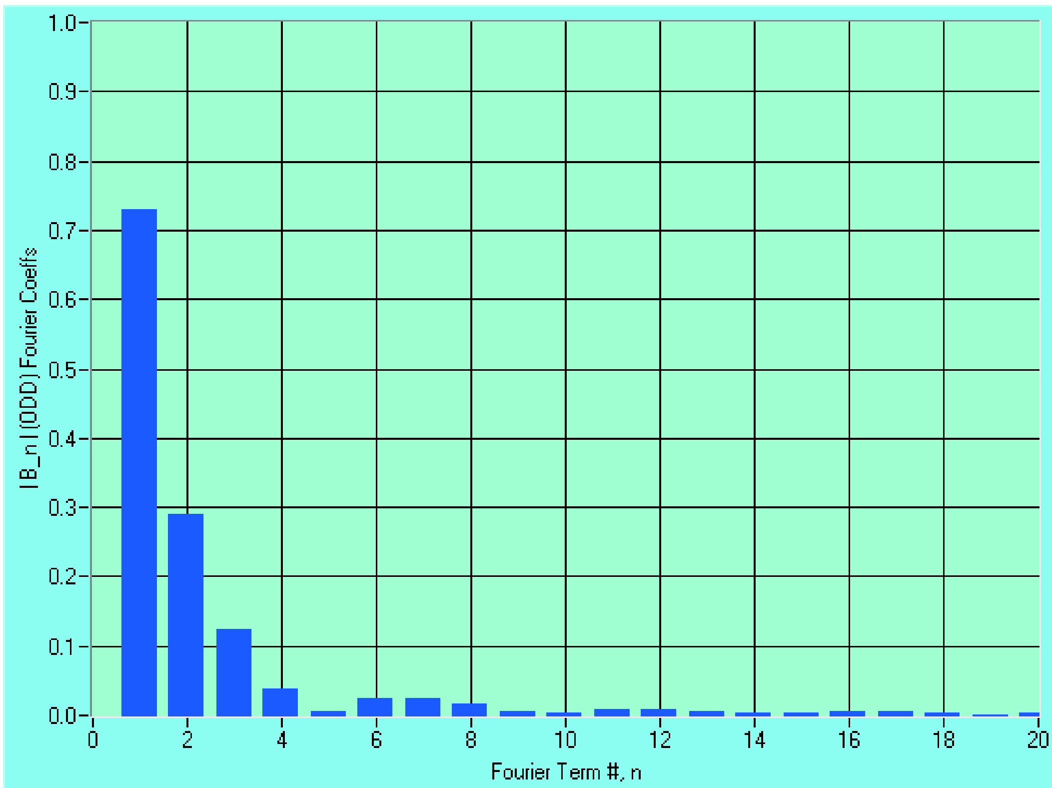


Figure 5: The linear plot of the odd Fourier coefficients

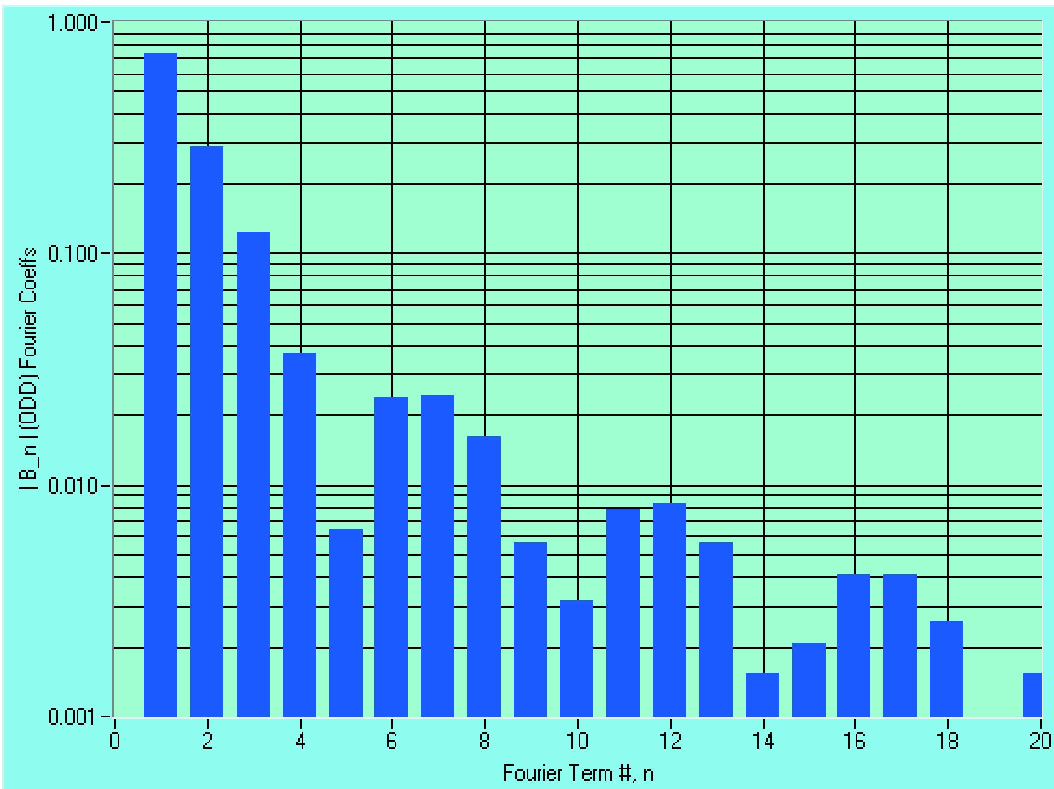


Figure 6: The logarithmic plot of the odd Fourier coefficients. The harmonics that have a node near the maximum of the picked string have lower amplitudes or vanish

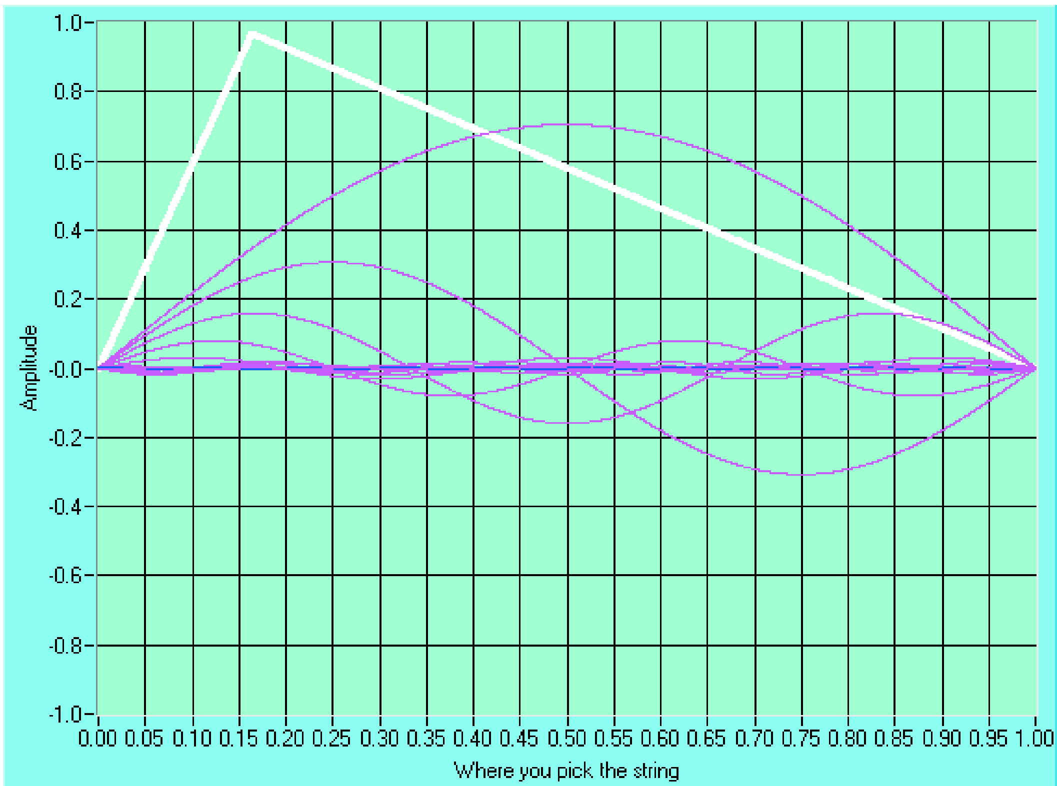


Figure 7: Plot of all calculated harmonics displayed at the same time

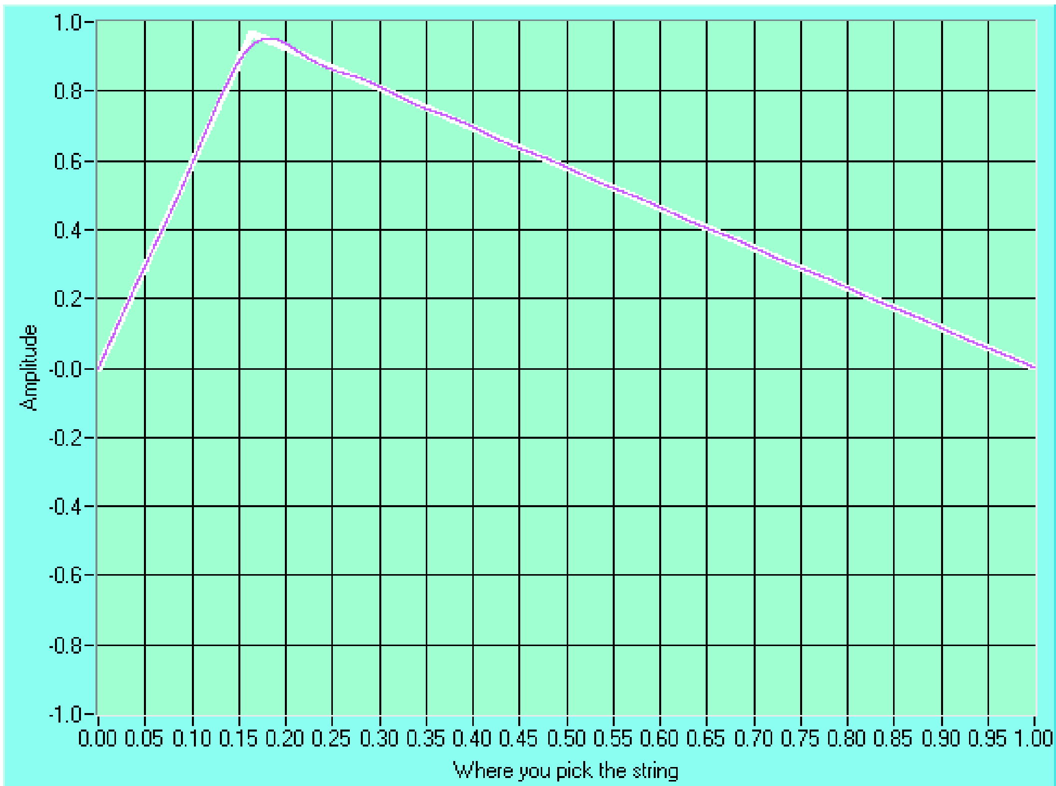


Figure 8: Plot of the regenerated (thin line) compared to the original waveform (white line). Here you can see how good the Fourier analysis works. To make it better increase the number of terms

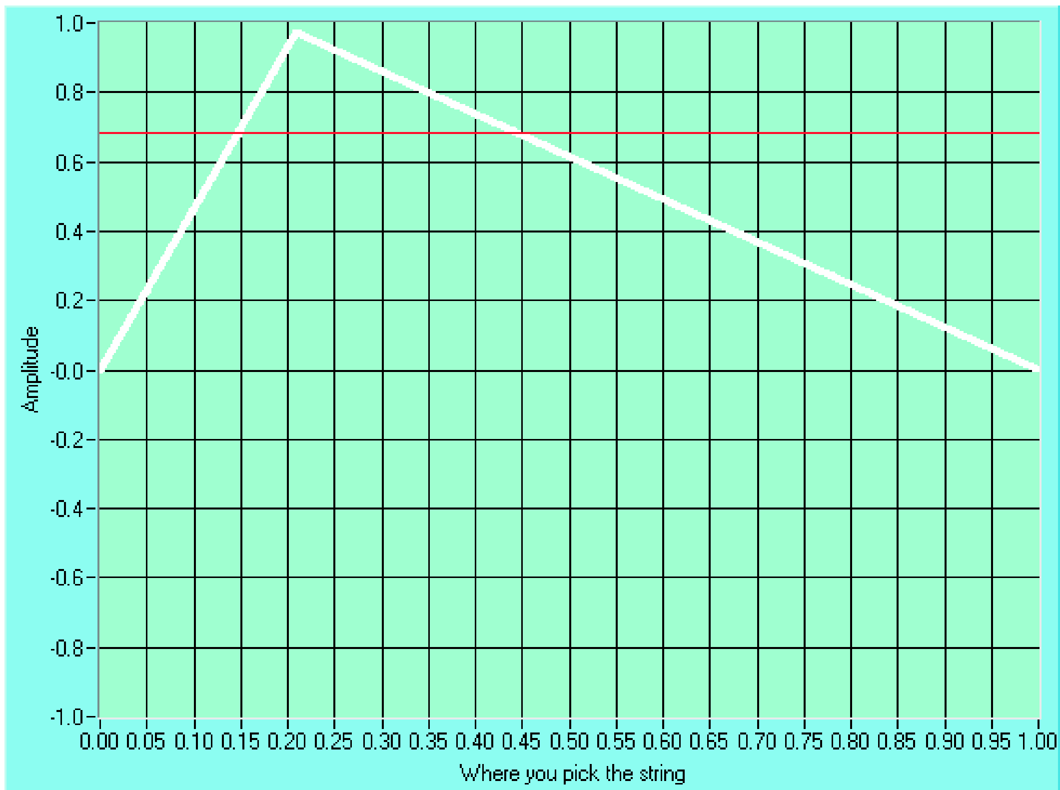


Figure 9: Plot with the clipping level. All absolute values of the amplitude above the horizontal line will be cut off. IMPORTANT: This doesn't effect the Fourier analysis!