Programming in Engineering - Guitar String Simulation and Overtone Analysis

Judith Zissoldt (s1332171) and Daniël Cox (s1228579)

October 14, 2016

Contents

1 Introduction **3**

2 Overtones **3**

3 Simulation **4**

3.1 Mass Spring model 4
3.2 Euler method 4
3.3 Implementation 5

4 Results & Discussion **6**

4.1 Output visualisation 6
4.2 Violation of Energy Conservation 6
4.3 Frequency Spectra for Picking Positions 10

5 Conclusion **11**

1 Introduction

The guitar instrument comes around in many forms, colors and materials. The oldest image of a stringed instrument, which shows features of a guitar, is around 3,300 years old [1]. So the guitar has been around for a very long time and in a lot of different cultures. There are several factors like shape, materials or strings, that play a role in the making of a tone. However for this report we will only focus on the guitar string. In the following such a guitar string will be modeled with Matlab. This model will enable the research-team to imitate and test the behavior of a real guitar string. This paper will try to answer the question as *"To what extend does the picking position on a string influence the tone?"*.

2 Overtones

An overtone is an acoustical frequency that differs from the fundamental frequency [2]. The frequency of the fundamental is described by $F_1 = v/2L$, where v is the speed of sound and L the the length of the string [3]. The fundamental's wavelength is calculated by the formula $\lambda_1 = 2L$ [3]. For each following overtone the wavelength is described by $\lambda_n = \lambda_1/n$ [3].

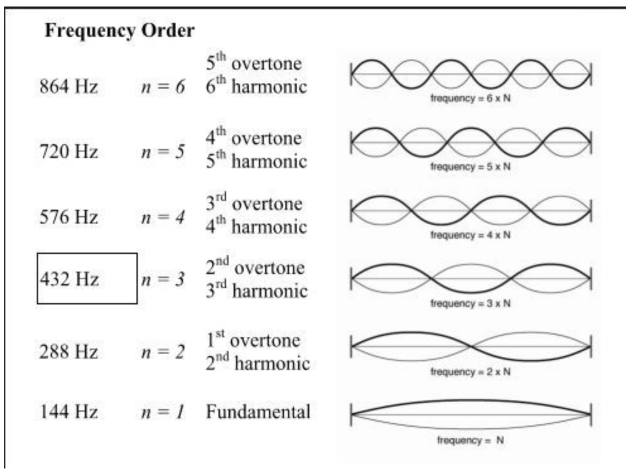


Figure 1: Harmonics and Overtones with base frequency of 144Hz

3 Simulation

3.1 Mass Spring model

In order to describe the physics of a guitar string, we'll model it as n point masses, or *nodes*, connected with springs with a given rest length l_0.

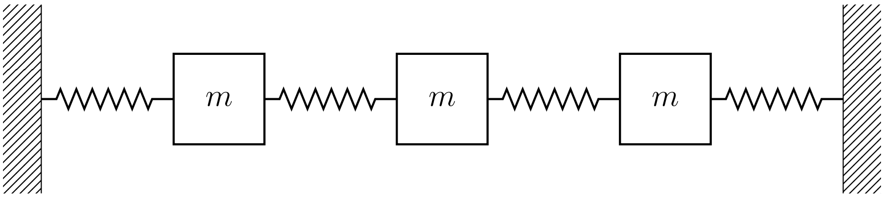


Figure 2: Mass-spring model of a guitar string

The forces involved are described by Newton's Second Law $F = m\ddot{x}$ and Hooke's Law $F = -kx$. Adjusting Hooke's Law for two dimensions and a nonzero rest length l_0, we find that:

$$F_x = -kx\left(1 - \frac{l_0}{\sqrt{x^2+y^2}}\right), \quad F_y = -ky\left(1 - \frac{l_0}{\sqrt{x^2+y^2}}\right) \tag{1}$$

Where x and y are the x- and y-distance between nodes. Neglecting friction and torque, the following differential equations can be constructed for the ith node:

$$\begin{aligned} \ddot{x}_i &= -\frac{kx_{i-1}}{m}\left(1 - \frac{l_0}{\sqrt{x_{i-1}^2+y_{i-1}^2}}\right) - \frac{kx_{i+1}}{m}\left(1 - \frac{l_0}{\sqrt{x_{i+1}^2+y_{i+1}^2}}\right) \\ \ddot{y}_i &= -\frac{ky_{i-1}}{m}\left(1 - \frac{l_0}{\sqrt{x_{i-1}^2+y_{i-1}^2}}\right) - \frac{ky_{i+1}}{m}\left(1 - \frac{l_0}{\sqrt{x_{i+1}^2+y_{i+1}^2}}\right) \end{aligned} \tag{2}$$

3.2 Euler method

In order to solve the aforementioned differential equation, we're using the so called *Euler method*. Though it is a rather inefficient method in terms of computational costs versus computational error, it is a very easy to understand method. In short, it can be described with the following equations, where v_n and x_n are the velocity and position at the nth time step respectively:

$$\delta v = \frac{F}{m}\delta t, \quad v_{n+1} = v_n + \delta v \tag{3}$$

$$\delta x = v\,\delta t, \quad x_{n+1} = x_n + \delta x \tag{4}$$

As δt is in principle just a discrete form of a the differential dt. Generally, the smaller it is, the more accurate the simulation, but the larger the computational costs. In section 4.2 the influence of δt on the deviation of the total energy of the system will be tested.

3.3 Implementation

The main function of the guitar string simulation is called `guitarstring.m`. It has either zero or one input argument. This argument is a structure containing at least one of the desired parameters. Any missing parameters will be set to the standard values. If no argument is provided, all parameters will be set to the standard values. (See figure 4)

The function `writeguitar` writes the settings structure, which contains one or more parameters, to the file `guitarsetting.txt`. (The user will be prompted for a file location.) The function `openguitar.m` can open such a file and return a corresponding structure. This allows the user to save and read the desired parameters.

In `guitarstring.m`, after the variables are initialized the constants are calculated, the nodes are evenly distributed over the string, the total energy of first time step is calculated and the string is given a 'kick'. (One node gets an initial velocity in the y-direction.) The behavior of a moving guitar string is modeled in a loop. The number of iterations of the loop can be set with the parameter `steps`. During each iteration the forces are calculated (equation 1) and the velocities and positions are updated to the new time step (equation 3 and 4).

As a visual aid, a small script (`guitarstring_animate.m`) that shows an animation of the simulated guitar string is included in the project files.

To test our simulation, several different aspects concerning energy conservation were investigated:
1. Normalized Standard Deviation of Total Energy depending on the size of time steps: `energyconservation_dt.m`
2. Normalized Standard Deviation of Total Energy depending on the number of nodes: `energyconservation_n.m`
3. Normalized Standard Deviation of Total Energy depending on the node that is picked: `energyconservation_p.m`
4. Normalized Standard Deviation of Total Energy depending on the starting speed of the first node: `energyconservation_vy0.m`

Furthermore, the frequency spectra of picking positions were investigated with: `overtone_picking.m`.

Apart from the provided zip file, the whole project can also be downloaded from: `https://github.com/juzis123/GuitarString`.

4 Results & Discussion

4.1 Output visualisation

Figure 3 shows the x and y coordinates of the nodes of a simulation run, at time step 15000. This is essentially a drawing of the string at time step 15000 and can even be used for animations!

4.2 Violation of Energy Conservation

In a real isolated physical system, the total energy of the system is always preserved. However, as this is an imperfect simulation of such a system, this is not entirely true. The variation in total energy could be used as a measure of the accuracy of the simulation. In the following subsections the influence of various parameters on the Normalized Standard Deviation of the Total Energy $\sigma_{n,TE}$ (or $\sigma_{E_{tot}}/\langle E_{tot}\rangle$ in the plot labels) was tested.

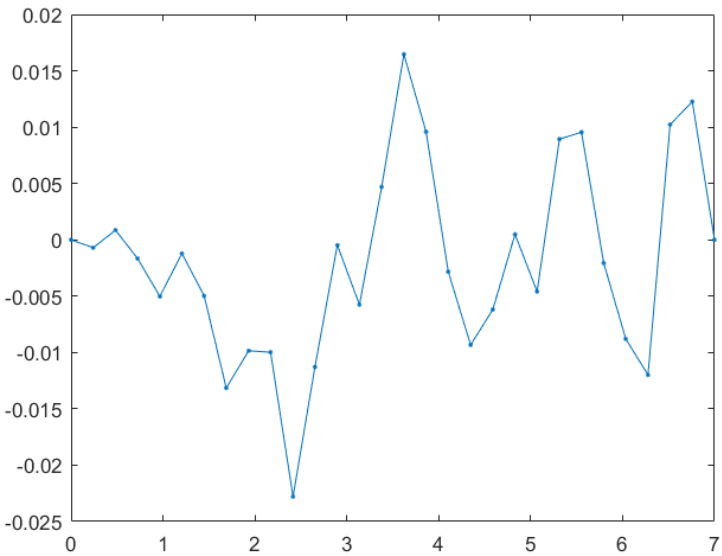


Figure 3: node coordinates at frame 15000

For the simulations several standard parameter values were used. These standard parameter values are defined in the main function `guitarstring.m`. A few parameters were adjusted for the specific tests. (These can be found in the code.) But this list of values gives a sense of what the numbers represent. It can be noted that all values are considered to be in SI-units. And of course, the parameters that were actually tested (`dt`, `n`, `p`, `vy0`) don't use these standard values in the tests where they were varied.

The results of these tests will be discussed in the conclusion (section 5).

```
    M     = 20;            % Total mass
    k     = 3;             % Spring constant
    n     = 30;            % Number of nodes
    p     = 2;             % relative picking position
    Ltot  = 7;             % length of string (when stretched)
    L0    = 4;             % Length of whole string (at rest)
    dt    = 0.005;         % Size of simulation time step
    steps = 50000;         % number of time steps
    vy0 = 0.1;             %Starting speed of first node
```

guitarstring.m

Figure 4: The standard parameter values.

4.2.1 Time step size

For the first test the size of the time steps `dt` was varied. The result is a neatly linear graph of $\sigma_{n,TE}$ versus the time step `dt` (figure 5a). This makes sense, as `dt` is only involved in the Euler integration method, which is the direct cause of the violation of energy conservation (see section 3.2).

4.2.2 Number of nodes

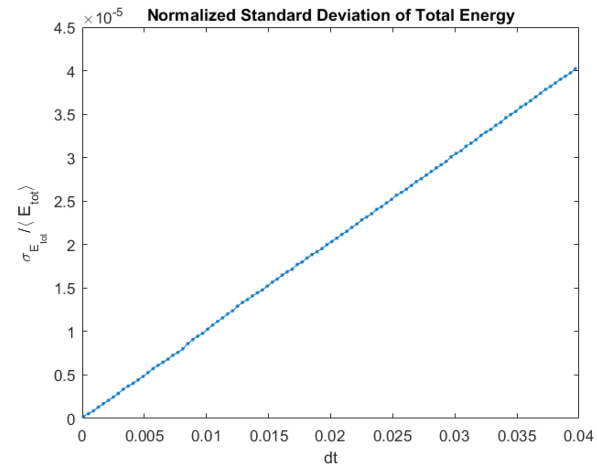
In the second test, the simulation was run for different numbers of nodes n. Though the first and last set of values seem to show a roughly square root kind of relation, the middle part shows a very interesting pattern. (Figure 5b) Between 40 to 150 nodes, there seems to be some sort of oscillating relation. As for now, it's not entirely clear what causes this pattern. One of our hypotheses is that changing the number of nodes slightly changes the speed of sound, causing the standing waves to focus the kinetic energy in certain points. The higher velocities arising from this effect would then cause the larger error. (See equation 4.)

4.2.3 Picking Position

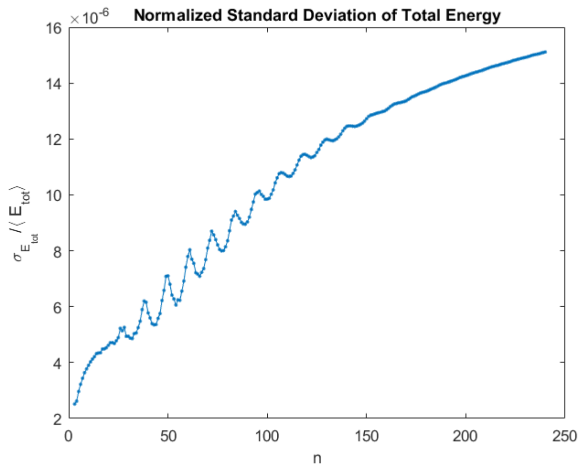
In the third test, the simulation was run for different picking positions. This also results in quite an interesting pattern (figure 5c). Our hypotheses for this pattern is that the higher/lower errors are again caused by higher/lower velocities. In this case, the velocities would be, as we will further study in section 4.3, caused by inducing certain overtones.

4.2.4 Initial speed of First Node

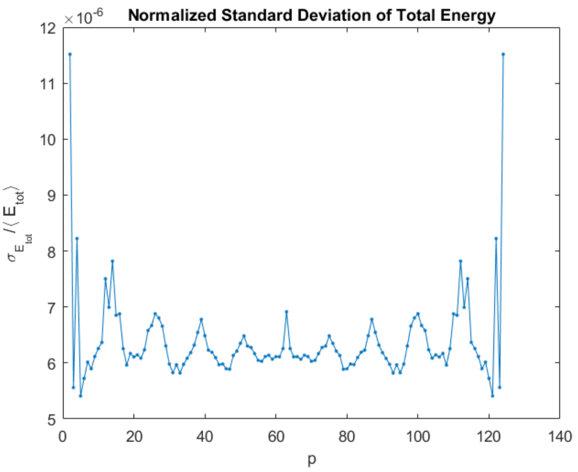
In the fourth test, the simulation was run for different initial velocities v_{y0} for the node that gets the first 'kick'. The figure 5d shows that the normalized standard deviation increases as the initial speed of the first node increases. However this increase does not seem to be entirely steady/regular. In figure 5d it can be seen that the second half of tested values seem to show a rather peculiar relation. We have no hypotheses as to what might cause this behavior.



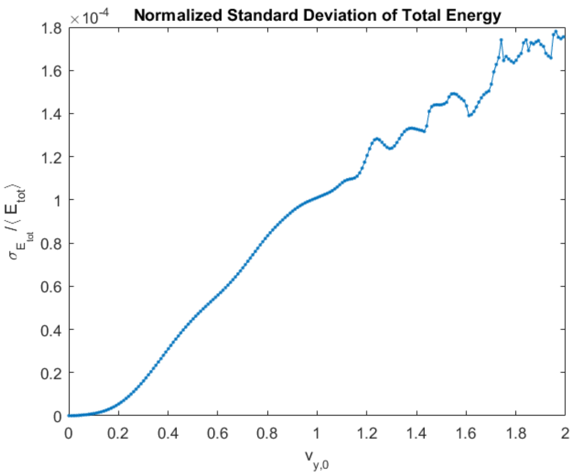
(a) Size of time step: `dt`



(b) Number of Nodes: `n`



(c) Picking Position on string: `p`



(d) Initial Velocity of first node: `vy0`

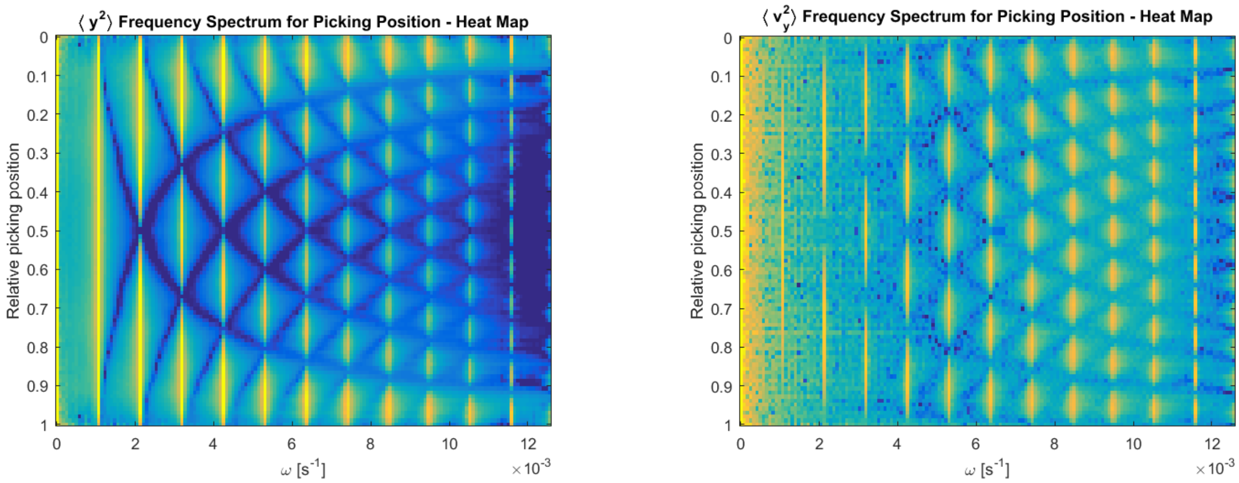
Figure 5: The Normalized Standard Deviation of the Total Energy as a function of several parameters.

4.3 Frequency Spectra for Picking Positions

In this subsection the frequency spectrum of the guitar nodes is studied in order to learn what the influence of the picking position might be. The simulation was run for 100 different picking positions. For all these simulations, samples were constructed from the average y-position squared $\langle y^2 \rangle$ as well as the average y-velocity squared $\langle v_y^2 \rangle$. Collecting samples from the average y-position $\langle y \rangle$ and average v-velocity $\langle v_y \rangle$ might make more sense, but if all nodes weigh equally in this average, the even overtones would cancel themselves out. Hence, $\langle y^2 \rangle$ and $\langle v_y^2 \rangle$ were chosen.

The frequency spectrum for each set of samples was then calculated with Matlab's built-in FFT algorithm. All of the aforementioned frequency spectra were combined to form a frequency heat map. These heat maps are shown in figure 6.

It can be immediately observed that the peak values correspond to the overtones of a string, including the evenly spaced frequencies (see figure 1). Furthermore, the picking positions inducing a certain overtone are exactly the positions where this overtone would vibrate; the shape of the overtone induction is the shape of the overtone itself!



(a) *(b)*

Figure 6: Frequency Spectra Heat Maps for Picking Position. The color mapping is logarithmic; yellow corresponds with a high value, blue with a low value.

5 Conclusion

The research question of this report was as *"To what extend does the picking position influence the tone?"*. Our results seem to suggest that the strength of the overtones of the end signal corresponds to the overtone 'presence' at the picking position. In other words, the shape of the overtone induction corresponds to the shape of the overtone itself.

Furthermore, testing the Normalized Standard Deviation of the Total Energy showed how different parameters influence the accuracy of the model to differing degrees and in different ways. This can be seen in the tests with the number of nodes. It was observed that for a certain range the influence of n changed. As explained in the result section the relation or cause of some of the occurrences or patterns is not clear yet. Some hypotheses were made, but further research is needed explore these effects.

References

[1] Michael Kasha, *"A New Look at The History of the Classic Guitar"*, Guitar Review 1968

[2] Overtones
http://www.dictionary.com/browse/overtone

[3] Wave Overtones
https://en.wikibooks.org/wiki/Physics_Study_Guide/Wave_overtones