University of Edinburgh

College of Humanities and Social Sciences

School of Arts, Culture and Environment

MSc in Acoustic and Music Technology

Final Project:
Piano Sounds Synthesis with an emphasis on the modeling of the hammer and the piano wire.

Teng Wei Jian

(s1160217)

Table of Contents

1. Introduction 4

2. Acoustics of the Piano 5

2.1 Background and Overview 5

2.2 The Hammer Excitation 7

2.3 The Stiff Piano String 9

2.4 The soundboard and the bridge 12

3. Modelling and Synthesis Techniques 14

3.1 Review of various digital sound synthesis methods 14

- Abstract Algorithms 14
- Processed Samples 15
- Spectral Models 16
- Physical Models 16

3.2 The Finite Difference Method 17

3.2 The Digital Waveguide Modelling 18

4. Hammer Model Design 19

4.1 Modelling the Hammer Excitation 19

4.2 Discussions 22

5. String Model Design 24

5.1 The travelling wave solution and the digital waveguide 24

5.2 The digital waveguide string model........25

5.3 Filter Design: The loss filter........27

5.4 Filter Design: The dispersion filter........29

5.5 Filter Design: The tuning filter........31

5.6 Other design considerations........32

5.7 Discussions........33

6. Conclusion and Future Work........35

Future Work........35

7. References........37

8. Appendix I – Wave Equation Coefficients........39

9. Appendix II – Matlab Implementation Code........40

1. Introduction

The study of the physics involved in the sound production of musical instruments has always had an important role in the field of acoustics. Researchers are constantly examining and investigating musical instruments to have a better understanding of how specific sounds are produced and also how sounds behave in general. A separate but related field of study is the investigation of how computers can be used in musical applications. The idea of digital sound synthesis is more or less a cross between these two studies.

The aim of this project is to gain a deeper insight into the sound production of the piano and from there, consider ways in which the sound of a piano can be digitally synthesized. The piano is a very complex instrument made up of many different parts, most of which have a direct influence on the tone produced. As a result, the tone of the piano is very rich and has very characteristic tonal qualities. How successful and realistic the sound output of the model is depends on how well the various parts of the piano can be modeled.

The piano sound synthesis will be implemented using physics-based modeling techniques. As such, an in depth discussion on the physics and acoustics of the piano will be presented first in Chapter 2. Chapter 3 will then provide an overview of the various sound synthesis methods available. Chapters 4 and 5 will each be dedicated to discussing the details of the implemented model. This will include the basics of the synthesis methods used, details of the implementation as well as a discussion on the results obtained. Chapter 4 will be focused on the model of the piano hammer while Chapter 5 will discuss the piano string model.

2. Acoustics of the Piano

2.1 Background and Overview

The piano is from a family of instruments classified as struck string instruments. Its ancestry includes the hammered dulcimer, the clavichord and the harpsichord. In fact, the basic design of the modern day piano was developed by an Italian instrument maker, Bartolomeo Cristofori, who had an extensive knowledge of the harpsichord and other stringed keyboard instruments. The piano is known for its flexibility, ubiquity and also its ability to produce complex melodic and harmonic structures. As a result, since its increased popularity in the 19th century the piano has remained as one of the most popular instruments in the world having crucial roles in almost all western musical genres.

The modern day piano can be divided into two types, the upright piano and the grand piano. In the grand piano, the frame, the piano strings as well as the soundboard are all laid out horizontally while in the upright they are vertical. Despite their differences, the mechanism involved in the sound production for the two are almost similar with the only difference being that the upright piano uses springs to restore the hammers to their resting positions while the grand piano uses gravity. The focus here will be on the grand piano and any future usage of the term “piano” will refer to the grand piano.

In the modern day piano, a cast iron frame is fastened onto the case (usually made of wood). Also attached to the case is the soundboard, a large piece of wood designed to amplify the vibrations of the strings. The bridge refers to two thin wooden bars on the soundboard (there are two because of the way the strings are laid out) meant to transmit the string vibrations to the soundboard. The piano strings are then stretched across the cast iron frame, crossing the bridge, and held in place on the tuning pins by agraffes on the near end of the string (closer to the keyboard) while the other end is attached to hitch pins.

In addition to that, there is the keyboard, the keyboard action and the hammer. The action refers to a very complex mechanical structure that translates the energy from the keys to the hammers that hit the strings. The action contains many mechanical parts and is very sensitive to the player's touch on the keys.

The sound production of the piano can be generalised into 3 stages:

1. The first is the excitation by the hammer. This refers to the process from the point the player presses the key up to the point where the hammer hits the strings. It involves the transfer of energy from the player to the hammer by the action.
2. After being excited by the hammer, the string will start to vibrate. The vibration of the string is determined by many factors including the physical properties of the string and also how the strings are terminated.
3. The vibration of the string is then transmitted to the soundboard via the bridge. The soundboard is designed in such a way that it will be forced to vibrate at the same frequency as the string. The sound of the piano that is heard is actually the vibration of the soundboard since the vibration of the string alone is too minute to produce any audible sounds.

Each stage of the sound production process will be discussed in greater detail in the following subsections. Displayed in Figure 2.1 are the basic sound production mechanisms in a piano, as mentioned above.

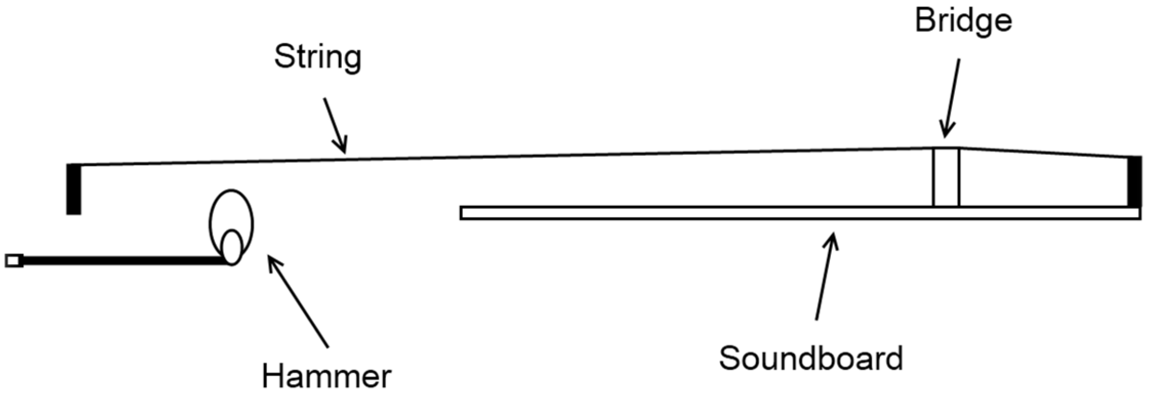

Figure 2.1 Structure of the sound production mechanism in a piano.

2.2 The Hammer Excitation

The intricacies of the design of the action can be seen from the fact that it has the capability to translate the motion of the key to the hammer, ensuring that the hammer can travel a distance of almost five times longer than the distance travelled by the key all in the same amount of time (Askenfelt, 1990). A good action design must also be able to allow for fast repetition of notes. The specifics of the piano action and the hammer have been the subject of much research. While some sources, especially pianists, insists that variations in the touch of playing can have significant influence in the tonal characteristics of the sound produced, physicists generally agree that the final velocity of the hammer is the only aspect that is controlled by the pianist. This is because it is believed that there is no longer any contact between the key and the hammer when the hammer actually strikes the string, and as such, the pianist only have indirect control over hammer. More recently, it is found that the hammer actually exhibits hysteretic behaviour (Fletcher & Rossing, 1998). This means that the hammer behaves differently during compression and relaxation; and so the current state of the hammer is actually dependent on its previous states. However, this does not suggest that there is a direct correlation between pianist and the hammer or that the pianist can have any more influence on the sound produced other than the final velocity of the hammer.

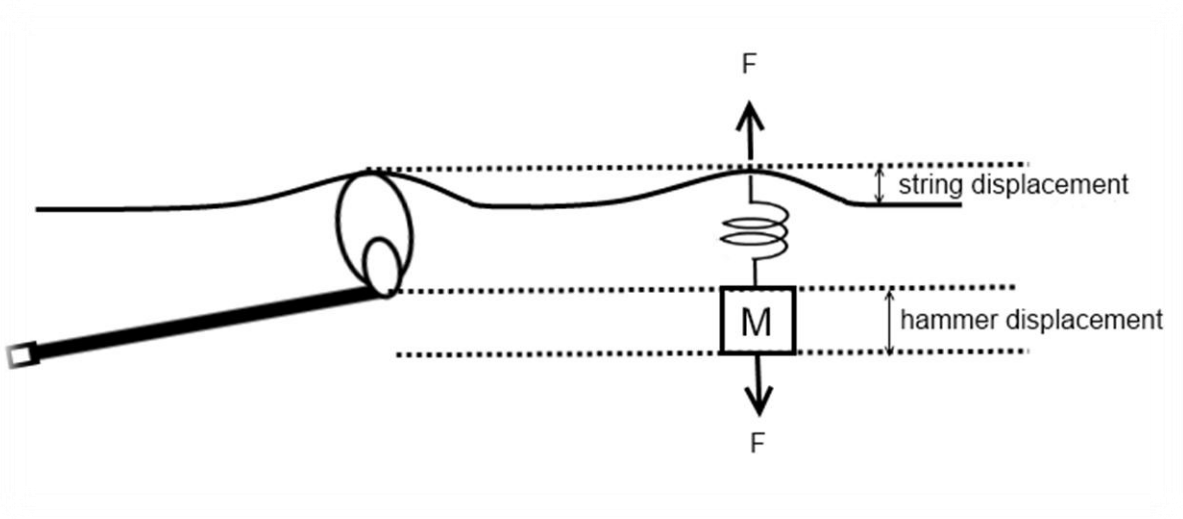

Figure 2.2 Depiction of the hammer as a mass-spring system.

In order to model the excitation on the piano string, it is necessary to take a closer look on the interaction between the hammer and the string. Consider the hammer that is made of a soft material that can be compressed (most commonly felt), the hammer can be approximated as a mass-spring system, as shown in Figure 2.2 (Hall, 1992). More precisely, because of the properties of the felt tip, the hammer should be approximated as a non-linear spring with increasing stiffness as the compression increases. The formula describing such a system is given as (Bank, 2000):

$$-F_H = m_h \frac{d^2\eta}{dt^2} \tag{1}$$

Where F_H is the force of the hammer, m_h is the mass of the hammer and η is the hammer displacement. The force of the non-linear felt hammer is given by the power law:

$$F_H = K\ |\Delta y(t)|^p \tag{2}$$

Where K and p are the stiffness and stiffness exponent of the hammer felt respectively. $\Delta y(t)$ is the felt compression and is the difference between the hammer displacement $\eta(t)$ and the string displacement $y(x_0, t)$ at the point of contact on the string (x_0). The compression is given as: $\Delta y(t) = \eta(t) - y(x_0, t)$. When the hammer displacement is less than the displacement of the string, indicating that the hammer is no longer in contact with the string, the interaction between the hammer and string is assumed to be over:

$$\Delta y(t) = \begin{cases} \eta(t) - y(x_0, t) & if\ \eta(t) > y(x_0, t) \\ 0 & if\ \eta(t) < y(x_0, t) \end{cases} \tag{3}$$

The model used for the current implementation will compute the output which is the hammer force, F_H, based on the initial hammer velocity,V_{H0}. The details of the model will be discussed in later sections.

It is interesting to note that the resultant hammer force usually has multiple pulses even though the hammer technically only strikes the string once, as indicated in Figure 2.3. This is because of the interaction between the hammer and the reflected waves from the ends of the strings. This is better illustrated by the following scenario. When the hammer first strikes the string, the string will be displaced. The displacement will cause travelling waves to form, moving in both directions of the string. As the waves reach the end of the strings, they will be reflected back but the waves will become inversed. The reflected wave will then travel back towards the point of contact. As the reflected wave passes the hammer, the string will be pushed down towards the hammer, effectively increasing the force exerted by the hammer onto the string.

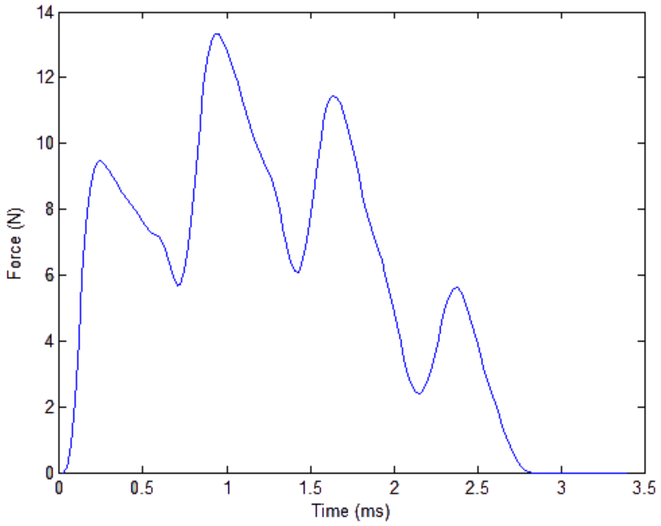

Figure 2.3Example of Hammer Force signal.

2.3 The Stiff Piano String

The piano string is without a doubt the most important part of the sound production process of the piano. In order to increase the efficiency, the piano string are stretched to 30-60% of its yield strength (Fletcher & Rossing, 1998) resulting in very high string tensions (around 700N) (Bank, 2000). Traditionally, the fundamental frequency of the tone produced by a string is inversely proportional to the length of the string. For the case of the piano, in order to make sure that the bass notes are of a reasonable length, the mass of these strings are increased. However simply having thicker wires will increase the stiffness of the strings and, therefore, its inharmonicity. The solution is to wind these wires with either one or two layers of copper wire. This effectively increases the wire's mass without increasing its stiffness. All these strict requirements make the piano string one of the most demanding applications for steel.

Essentially the piano string is a stiff string that is struck with a hammer. The equation that describes such a string is given below (Bensa et al., 2003):

$$f(t,x) = \mu \frac{\partial^2 y}{\partial t^2} - T \frac{\partial^2 y}{\partial x^2} + E\, I \frac{\partial^4 y}{\partial x^4} + b_1 \frac{\partial y}{\partial t} + b_2 \frac{\partial^3 y}{\partial x^2 \partial t} \tag{4}$$

Where:

$f(t,x) = dirving\ force\ density\ \left(\frac{N}{m}\right) at\ position\ x\ and\ time\ t$

$\mu = mass\ density\ (kg/m)$

$T = tension\ force\ along\ the\ string\ axis\ (N)$

$E = Young's\ \ Modulus\ (N/m^2)$

$I = radius\ of\ gyration\ of\ the\ string\ cross\ section\ (m)$

The force density term on the left hand side is the hammer force that excites the string. The first two terms on the right hand side of the equation describes the motion of an ideal string, and it is these two terms that determine the basics of the string vibration (e.g fundamental frequency, etc.). The piano strings are by design stiff, therefore the third term in the equation accounts for the effects of stiffness in the string. Specifically, this term describes the transverse restoring force exerted on the stiff string when it is bent. The effects of stiffness will be discussed in greater detail in the following paragraphs. The last two terms account for the losses that the wave experiences when travelling along the string. The losses cause the vibration to decay over time. In some cases the second loss term is given as $b_3\ \frac{\partial^3 y}{\partial t^3}$ (Chaigne & Askenfelt, 1994).

The stiffness of the string causes the tone produced to be slightly inharmonic. Recall the stiffness term in the wave equation described in Eq.(4). The fourth order derivative in space that makes up the stiffness term results in the wave being dispersive. Dispersion refers to the effect that the wave propagation speed is increased with increasing frequency. It means that higher frequency components will travel faster than lower frequency components. The inharmonicity is because the dispersion causes higher frequency partials to be spaced further apart than lower frequency partials while for the ideal string case, the partials are all evenly spaced.

The stretched partials are given by (Bank, 2000):

$$f_k = k\, f_0 \sqrt{1 + Bk^2} \tag{5}$$

Where f_k is the k-th partial, f_0 is the fundamental frequency and B is the inharmonicity coefficient given as:

$$B = \frac{\pi^3\, E\, d^4}{64\, l^2\, T} \tag{6}$$

Where E is the Young's modulus of the string, d is the diameter of the string, l is the length of the string and T is the tension in the string. The inharmonicity coefficient described in Eq.(6) is only accurate for normal wires. For the wound wires, the coefficient is typical lower than the values given by Eq.(6). From Eq.(6) it can be seen that the inharmonicity increases with increasing frequency except for the very low frequency strings (Bensa et al., 2003). Despite this trend, lower frequency notes are generally heard to be more inharmonic than higher frequency notes (Bank, 2000). One possible explanation to this phenomenon is that the lower frequency notes have more partials that lie within the audible range than higher frequency notes. Another possible explanation is that psychoacousticaly, the ear's perception of inharmonicity is less sensitive towards high frequencies sounds.

The vibration of the piano string is also affected by its coupling to other strings as well as to the soundboard and the bridge. In reality, three strings are used for each note, except for the lower frequency notes which only use two strings. Therefore when a key is pressed, instead of only hitting one string, the hammer actually strikes all three (or two) strings at once. Aside from increasing the overall acoustical output of the note, the coupling of multiple strings also produces two important effects that are characteristic of piano sounds. The strings for the same note are deliberately tuned to have slightly different frequencies. Therefore when they are vibrating together, beating will occur.

The second effect from the coupling of the strings is the two stage decay whereby the tone produced decays in two stages: faster at first and then slower in the second stage. One possible explanation is because of the different way the two polarization of the vibration behaves (Askenfelt, 1990). The explanation suggested is that the transmission of the vibration on the bridge is better for the vertical polarization than it is for the horizontal. As such, the vertical vibration of the string will decay faster than the horizontal since most of it is being transmitted to the soundboard through the bridge. The result is that the vertical polarization (which dominates initially) will decay faster, hence the faster decay time earlier on; but once the vertical has decayed, the horizontal is still left to decay at its slower rate. There is, of course, a coupling between the two polarizations which makes the real scenario more complicated than the case exhibited here, but the basis of the explanation remains sound.

2.4 The soundboard and the bridge

The design of the present model is concentrated on the above two parts and as such, only a brief overview of the soundboard and bridge will be given here. The bridge, being connected to all of the strings, presents a special coupling effect between the strings that is unique for each string. So when one string is vibrating, it is actually coupled to all the other strings as well through the bridge.

As mentioned, the soundboard is responsible for resonating the vibrations of the string and to amplify these vibrations while the bridge's role is to transmit the string vibrations to the soundboard. The soundboard is made from special types of wood called tonewoods known for their ability to produce consistent tones when vibrated. Spruce is usually used, with a preference of wood with straighter and denser grains. The material selection is very precise as it has a direct influence on the quality of the sound the piano produces. To increase the soundboard's efficiency, ribs are attached to the bottom of the soundboard. Ribs are strips of softwood that is attached onto the soundboard at a ninety degree angle from the planks of the soundboard. They increase the radiation efficiency of the soundboard by increasing its stiffness and also help to strengthen the soundboard.

Since the soundboard is responsible for amplifying the sound and giving it the piano's characteristic timbre, the idea is to effectively couple the string vibrations to the soundboard as efficiently as possible over a large range of frequencies. The most efficient way to do this would be to connect the strings directly to the soundboard. However, such a coupling would result in a very short decay time although the initial sound would be very loud. The design of the soundboard and bridge is therefore a trade-off between the loudness of the sound produced and the sustaining capability of the sound.

3. Modelling and Synthesis Techniques

This section will review the different methods of sound synthesis as noted by Smith (1991) and (Bank, 2000). The methods used for the current implementation will then be discussed in greater detail.

3.1 Review of various digital sound synthesis methods

Digital sound synthesis refers to the act of recreating sounds using the computer. The first practice of digital sound synthesis is generally attributed to the experiments conducted at Bell Telephone Laboratories in 1957 (Roads, 1996). The researchers there have shown that the computer is capable of producing sounds with specific pitch, time-varying frequency as well as amplitude envelopes. Fast forward in time and today there are many different ways in which a computer can be used to synthesize sounds. These methods differ in complexity and computational costs, level of physical intuitiveness and in general, the quality of the sound and how musically useful the results are. There are many motivations for digital sound synthesis, including the desire to extend and to explore the possibilities of conventional instruments to produce different sounds or even to control the design of sound in general as a compositional tool.

The categorization of the digital sound synthesis methods, as proposed by Smith (1991), is based on the different approaches used to synthesize the sounds. They are categorized into these four groups:

- **Abstract Algorithms**

  Examples of synthesizing methods that used abstract algorithms include frequency modulation (FM), Voltage Controlled Oscillator (VCO)/Voltage Controlled Amplifier (VCA), wave shaping and also the very famous Karplus-Strong algorithm. These methods revolve around modifying sound in different ways. Although most of the methods here are capable of producing spectrally rich sounds by adjusting its control parameters, they are normally not able to reproduce sounds of actual instruments. These methods are simple and they usually have a number of control parameters (albeit small). On the other hand, due to the fact that the controls are less physically meaningful, the analysis of these methods can be complicated. Although limited in its application to reproduce sounds from real instruments, these methods are useful in creating novel sounds.

One such method worth noting is of course the Karplus-Strong algorithm. Introduced in 1983, this algorithm is able to quite accurately simulate the sound of plucked stringed instruments and also drum instruments. Although originally developed as a modified version of the wavetable synthesis, it is later found that the Karplus-Strong algorithm is actually a version of digital waveguide modelling.

- **Processed Samples**

In this method, samples from actual instruments are pre-recorded and then manipulated to produce specific sounds. The advantage is of course that the sounds produced are very accurate and realistic, since they are all recorded samples. The downside to this is that there is a very limited amount of control on the condition of playing that is not already sampled. The other disadvantage of this method is that it requires a significant amount of memory in order to store the recorded samples.

Examples of synthesis methods that use processed samples include sampling synthesis, wavetable synthesis and granular synthesis. It is interesting to note that commercially available digital pianos are all implemented using sampling synthesis where one period of the desired sound is sampled and then played back by looping the sample. Amplitude envelopes and filters are then used to control the dynamic of the sound and also to simulate the amplitude and timbre evolutions (Roads, 1996). Although this method is preferred in the commercial sense, it provides little to no insight into the way pianos produce sounds. Therefore, for the purposes of this project, other more physically meanignful methods will be used.

- **Spectral Models**

The previously mentioned methods are mostly based on the manipulation of signals in the time domain to perform the synthesis. On the contrary, the basic idea of this method is to approach the problem from the frequency domain. Due to the fact that this method mainly operates in the frequency domain, it is easier for these methods to account for psychoacoustic properties. For all its strengths, this method suffers from the fact that the spectral characteristics of real instruments are generally very complicated and therefore requires many parameters in order to accurately describe it. Another disadvantage is that the simulation of transient is difficult which makes these methods less appealing for synthesizing piano sounds. Examples of synthesizing methods that fall into this category include: additive synthesis where sinusoidal signals of different frequencies and amplitudes are summed together, subtractive synthesis which is the opposite of additive synthesis where filters are used to attenuate partials of certain sound waves in order to form new sound waves and also the phase vocoder.

- **Physical Models**

As the name implies, methods that fall into this category are based in the physics of how the sounds are produced. Unlike the previous methods which try to recreate the signal of the sound wanted, sound synthesis using physical models tries to recreate the source of the sound instead. The major advantage to using physical models is that it has the potential of producing the most accurate instrument sounds. Also, being based on the actual instrument, the control parameters are more intuitive and physically meaningful, therefore providing a more realistic means of interaction between the player and the "instrument". The major drawback of physical models is that they are generally computationally expensive.

Examples of digital synthesis methods that employ physical models include modal synthesis where the vibrational modes of the structure is modelled, finite difference methods and also digital waveguide synthesis. A combination of the finite difference and digital waveguide synthesis methods will be used to synthesize the sound of the piano in this project and will be discussed in greater detail in later sections. The reason for using physical models in the current implementation is that they provide a better insight into the way the sounds are actually produced in real pianos and while physical models are typically more expensive computationally than other methods, computers today with its improved processing powers is more than capable of performing the task within reasonable constraints.

3.2 The Finite Difference Method

The finite difference method involves solving for the discrete time solution of the differential equations that describe the sound. Strictly speaking, the finite difference method is a way of approximating the numerical solutions of differential equations. It was originally used to solve Maxwell equations but is now used to solve differential equations in various engineering fields.

The basic idea is to use difference equations to approximate the derivatives of the differential equations. The finite difference schemes operate in a space-time grid, a discretized version of the actual continuous space and time domain. The difference equation will use neighbouring points in the space-time grid to calculate an estimation of the derivatives. An example is given in Eq.(7)

$$\frac{\partial x}{\partial t} \approx \frac{x_{n+1} - x_{n-1}}{2\,\Delta t} \tag{7}$$

In Eq.(6), x's first derivative in time is estimated as the average differential between the samples taken one time step before and after the current sample. The derivatives of the differential equations can be substituted by these difference equations and then rearranged to form recursive relations. The recursive relations can then be implemented as algorithms and be used to solve the differential equations.

The advantages of using the finite difference method is that it is a solution of the actual equations describing the wave motion and as such, as long the equations are accurate, the result of the finite difference synthesis will reflect the physics involved in the real instrument. However, finite difference schemes generally have high computational costs, especially if very accurate results are required. Chaigne & Askenfelt (1994) used the finite difference method to model the string of the piano. The model used by Chaigne & Askenfelt includes the modeling of the hammer action as well as the piano string. The hammer model used for the implementation of this project is based on Chaigne & Askenfelt's model.

3.2 The Digital Waveguide Modelling

The digital waveguide synthesis was proposed by Julius O. Smith in 1992 and is based on the idea of travelling wave solutions (Smith J. O., 1992). A general solution to the wave equation is first solved to find the travelling wave solutions. The discretized version of the travelling wave is then modelled using digital waveguides. Delay lines are used to model the way the waves travel along a medium and filters are used to model the other characteristics of the medium including loss, dispersion and etc. In most cases, loss and dispersion characteristics can be modelled using linear time-invariant filters and because of the commutative property of LTI systems, they can be lumped together. As a result, the simulation of the entire system will consist mainly of delay lines with only one filter to account for loss and dispersion. It is for this reason that the computational costs of digital waveguide models are much lower than finite difference methods, at least for one-dimensional systems like strings and tubes. For two and three dimensional systems, finite difference methods tend to be more computationally efficient. The piano string in the model presented in this project will be modelled using digital waveguides.

4. Hammer Model Design

This chapter will detail the model design for the hammer. The implementation details will be discussed followed by a discussion on the result of the model.

4.1 Modelling the Hammer Excitation

The model of the hammer is responsible for simulating the force that the hammer exerts on the string based on the input, which is the initial velocity of the hammer. The design of the hammer model will be based on the finite difference model proposed by Chaigne & Askenfelt (1994). The model is based on the equations relating the force of the hammer and the displacement of the hammer given in Eq.(1) and Eq.(2).

Recall that the force of the hammer is dependent on the compression of the hammer tip which is given by the difference between the displacement of the hammer and the displacement of the string. In order to accurately model the hammer, the displacement of the string needs to be modelled as well. Therefore, a finite difference model of the string is included in the model of the hammer. Note that the finite difference string model is only used in the hammer model; the actual implementation of the piano string will use a digital waveguide model instead. The reason for using the finite difference string model is that since simulation of the hammer force is only active for a short amount of time (< 5ms, which is a few hundred samples at most for most sampling frequencies) the computational cost would not be too much of a problem. Also a finite difference string model can be easily interfaced with the finite difference implementation of the hammer. Some sources have proposed using a digital waveguide model for the modelling of the hammer (Borin et al, 1992) (Bank, 2000). The advantage is that it can be readily interfaced with a digital waveguide string model and can have a direct feedback from the string model (unlike the model presented currently in the project, where the hammer model and string model is clearly separable). However, stability can sometimes be an issue for the digital waveguide hammer model and the necessary feedback from the string model complicates the design for both the string model as well as the hammer model.

By substituting the derivative with a difference equation, Eq.(1) can be rewritten into a recurrence relation for the hammer displacement:

$$\eta(n) = 2\eta(n-1) - \eta(n-2) - \frac{F_H(n-1)\,\Delta t^2}{M_H} \tag{8}$$

Similarly the recurrence relation for the string model is given as:

$$\begin{aligned} y(i,n) = {} & a_1 y(i,n-1) + a_2 y(i,n-2) + a_3[y(i+1,n-1) + y(i-1,n-1)] \\ & + a_4[y(i+2,n-1) + y(i-2,n-1)] \\ & + a_5[y(i+1,n-2) + y(i-1,n-2) + y(i,n-3)] \\ & + [\Delta t^2 N F_H(n-1) g(i,i_0)]/M_S \end{aligned} \tag{9}$$

The coefficients $a_1$to a_5 are given in Appendix I. N is the number of string segments and $g(i,i_0)$ is a dimensionless spatial window that accounts for the width of the hammer.

Several assumptions are made regarding the initial conditions of the hammer model currently presented; although it is possible to vary these initial conditions. It is assumed that the initial velocity is given as V_{H0} and that the hammer and string are both at rest initially. This means that the hammer displacement, hammer force and string displacement are all zero at $n = 0$. With the string at rest:

$$y(i,0) = 0 \tag{10}$$

i here refers to the discretized spatial index, similar to how n refers to the discretized time index. i_0 will be used to refer to the position of the hammer strike on the string. At the first time step, $= 1$, the hammer displacement can be calculated from the initial hammer velocity:

$$\eta(1) = V_{H0}\,\Delta t \tag{11}$$

The string displacement is then needed in order to calculate the hammer force at $n = 1$. However, the complete recurrence relation for the string displacement given in Eq.(9)

cannot yet be used since it requires the information from three previous time steps in order to compute its current value. Therefore, an estimate of the current string displacement is calculated by using a Taylor series approximation. $y(i, 1)$ can be estimated as:

$$y(i, 1) = \frac{y(i + 1{,}0) \ + \ y(i - 1{,}0)}{2} \tag{12}$$

The hammer force can then be found from Eq.(2):

$$F_H(1) = K |\ \eta(1) - y(i_0, 1)\ |^p \tag{13}$$

For $n = 2$, the string displacement can be estimated by using a simplified version of the complete recurrence relation, where the terms that require values from more than 2 time steps ago is neglected:

$$y(i, 2) = y(i + 1{,}1) \ + \ y(i - 1{,}1) - y(i, 0) + \frac{\Delta t^2 \ NF_H(1) g(i, i_0)}{M_S} \tag{14}$$

The hammer displacement at $n = 2$ can then be computed using Eq.(8):

$$\eta(2) = 2\eta(1) - \eta(0) - \frac{F_H(1)\ \Delta t^2}{M_H} \tag{15}$$

The hammer force is then:

$$F_H(2) = K |\ \eta(2) - y(i_0, 2)\ |^p \tag{16}$$

After determining the string displacement for the first three time steps, the complete recurrence relation for the string as given in Eq.(9) can be used to calculate the future displacements. The process of calculating the string displacement, calculating the hammer displacement and then calculating the hammer force is repeated until the hammer is no longer in contact with the string, when $\eta(t) < y(x_0, t)$. The stability conditions of the scheme are also noted by (Chaigne & Askenfelt, 1994).

4.2 Discussions

Figure 4.1 below shows the force signals generated using the hammer model presented based on different values of initial hammer velocity. The force signal generated is in agreement with what is expected. The first is the presence of the multiple peaks as described in Chapter 2.2. Aside from that, the shape of the force signal with respect to the initial hammer velocity is also more or less as expected. Higher hammer velocity represents a harder key press, producing a force signal with sharper and higher peaks while slower initial hammer velocity produces output that have lower and flatter peaks. Some sources have found that the result of the finite difference model is comparable to that of the modified digital waveguide models (Bank, 2000) (Bank, et al., 2003).

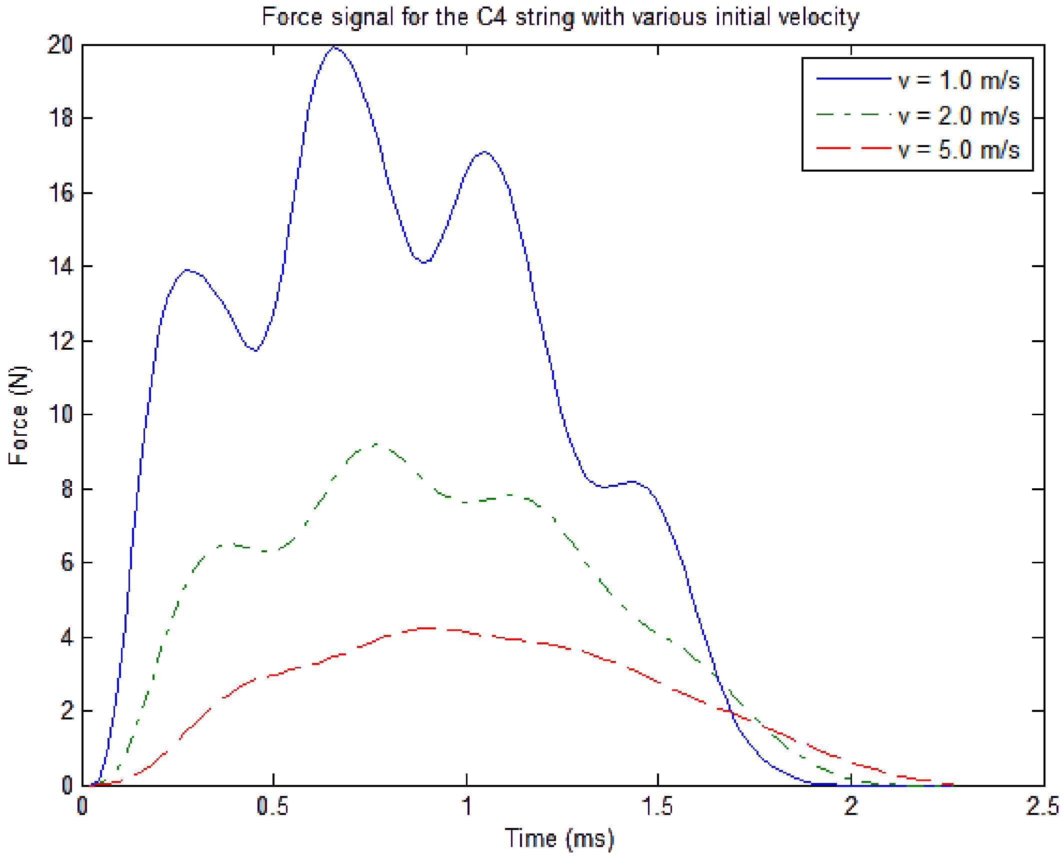

Figure 4.1 Force signal for the C4 string with various initial velocity.

This finite difference model of the hammer requires a large amount of input parameters, including both the hammer parameters as well as string parameters. Some of the required parameters include the mass of the string, the mass of the hammer, the length of the string, the striking position of the hammer just to name a few. A more complete list of parameters is given in Table 4.1. The data is obtained from Chaigne & Askenfelt's measurements in (Chaigne & Askenfelt, 1994).

	C2	C4	C7	Units
$\boldsymbol{M_S}$	35.0	3.93	0.467	g
$\boldsymbol{L}$	1.90	0.62	0.09	m
$\boldsymbol{M_H}$	4.9	2.97	2.2	g
$\boldsymbol{T}$	750	670	750	N
$\boldsymbol{p}$	2.3	2.5	3.0	
$\boldsymbol{K}$	$4.0\ 10^{8}$	$4.5\ 10^{9}$	$1.0\ 10^{12}$	
$\boldsymbol{M_H/M_S}$	0.14	0.75	4.71	
$\boldsymbol{\alpha}$	0.12	0.12	0.0625	
$\boldsymbol{f_0}$	65.4	262	2093	
$\boldsymbol{E}$	$2.0\ 10^{11}$	$2.0\ 10^{11}$	$2.0\ 10^{11}$	N/m^2
$\boldsymbol{b_1}$	0.5	0.5	0.5	s^{-1}
$\boldsymbol{b_3}$	$6.25\ 10^{-9}$	$6.25\ 10^{-9}$	$2.6\ 10^{-10}$	s
$\boldsymbol{\varepsilon}$	$7.5\ 10^{-6}$	$3.82\ 10^{-5}$	$8.67\ 10^{-4}$	

Table 4.1Values for the model parameters.

As indicated in the table, some of these values are very specific and without actual measurements, the hammer model cannot accurately simulate the force signals for all the strings. The model certainly has the capability to do so, but the problem is with the availability of these input parameters. Even taking the measurements from an actual piano is no trivial task and requires specific setups. In the string model to be presented in the following chapter, in order to simulate the sounds for notes other than the ones specified in Table 4.1, the force signal for the C4 string is used as input to be fed into the digital waveguide model. The justification is that while the parameters for different strings are undoubtedly different, the resulting force signal should be more or less of the same form. This is mostly true except at the extreme ends of the keyboard. For example in the higher frequency where shorter strings are used, for the same amount of output, or at least acomparable amount, a larger amount of force is required. Since the same force signal is used for all strings, the actual output can be quite small for the high frequencies. The output of each string is normalized so that the output is comparable.

5. String Model Design

In this section, the model used to simulate the piano string is discussed. The concept of travelling wave solution is detailed in relation to the digital waveguide model. The digital waveguide model used to simulate the piano string will then be presented, followed by a detailed account of the filter designs. Additional considerations to make the model more realistic will also be discussed. At the end of the chapter will be a short discussion of the results of the model presented.

5.1 The travelling wave solution and the digital waveguide

The concept of travelling was solution was proposed by mathematician Jean le Rond d'Alembert. d'Alembert suggested that a lossless wave equation, for example the wave equation for an ideal string as given in Eq.(17), will have a general solution in the form of two superimposed waves travelling in opposite directions.

$$\mu \frac{\partial^2 y}{\partial t^2} = T \frac{\partial^2 y}{\partial x^2} \tag{17}$$

The travelling wave solutions will then be:

$$\begin{aligned} y^{+}(x,t) &= g\left(t - \frac{x}{c}\right) \\ y^{-}(x,t) &= h\left(t + \frac{x}{c}\right) \end{aligned} \tag{18}$$

Where $= \sqrt{\frac{T}{\mu}}$. The general solution can be found by adding the travelling waves together:

$$y(x,t) = g\left(t - \frac{x}{c}\right) + h\left(t + \frac{x}{c}\right) \tag{19}$$

When discretizing the solution, the spatial interval is set to coincide with the distance that the wave travels in one time step ($\Delta x = c\ \Delta t$). This means that in the span of one time step the wave will travel the distance of exactly one spatial step. By having this setting, the digital waveguide uses delays to simulate how the waves travel along the string, shown in Figure 5.1.

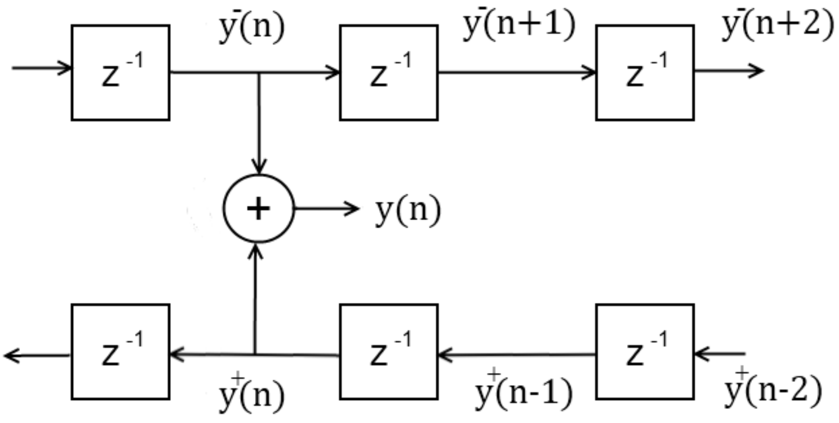

Figure 5.1Basic idea of the digital waveguide.

5.2 The digital waveguide string model

The left and right travelling waves that form the solution to the ideal string wave equation can be modelled by using two delay lines, each of length M. For the ideal string case, assuming infinitely rigid terminations on both ends, the reflection of the waves can be modelled by using a -1 gain term. The digital waveguide model for an ideal string is shown in Figure 5.2. M_i refers to the position of the hammer strike and can be calculated from $M_i = \alpha\ M$. (α is obtained from Table 4.1)

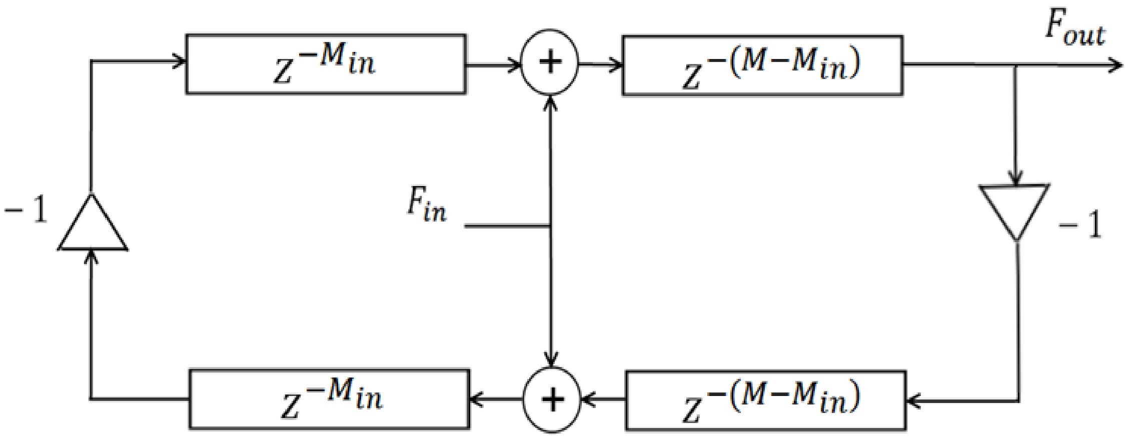

Figure 5.2 The digital waveguide model for an ideal string.

The length of the delay line is determined by the fundamental frequency of the tone that is to be modelled. The length of the entire delay line is given by:

$$N = \frac{F_S}{f_0} = 2M \tag{20}$$

The digital waveguide model of the string presented in this project actually models the velocity of the string and not the displacement. Despite this, the travelling wave solution and also the digital waveguide model still holds. One of the reasons for choosing to model the string velocity instead of displacement is that velocity allows for easier interfacing with the hammer force output from the hammer model. The input function F_{in} to the digital waveguide model can be obtained by:

$$F_{in} = \frac{F_H}{2R_0}$$

Where R_0 is known as the wave impedance of the string and is defined as $R_0 = \sqrt{\mu T}$.

To extend the digital waveguide string model presented above to a non-ideal string, the effects of loss and dispersion needs to be accounted for. As mentioned in earlier sections, these effects can be lumped to one point and can be represented by one digital filter, the reflection filter. The design of this reflection filter is what determines how realistic the model behaves. The details on the design of the filter will be discussed in following sections. The digital waveguide model for the piano string is shown in Figure 5.3 below. Also included in the reflection filter will be a tuning filter that introduces fractional delays, details of this filter will be discussed in a later section.

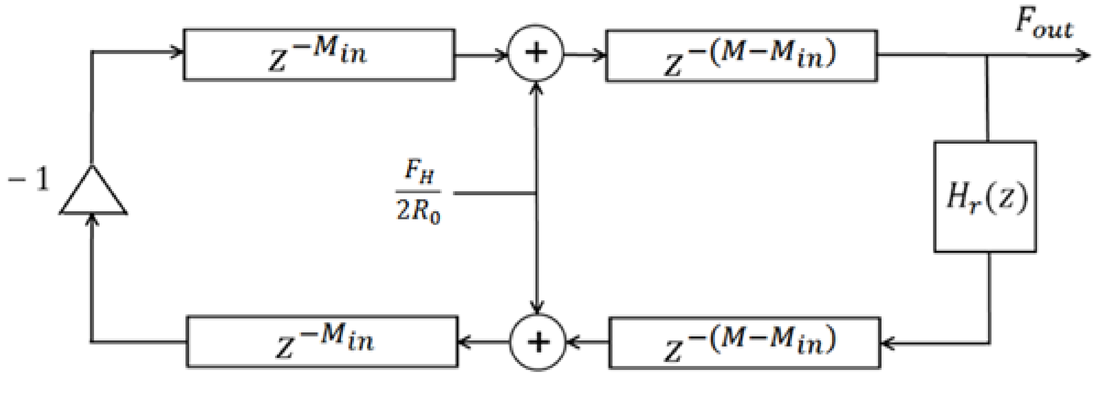

Figure 5.3 The digital waveguide model for the piano string.

5.3 Filter Design: The loss filter

The purpose of the loss filter is to account for the losses that the wave experiences as it travels along the string. As a result of losses, the tone produced will decay over time, unlike the case for an ideal string where once excited, the string will vibrate theoretically forever. The simplest method to account for such a loss term would be to use a gain term of less than 1. Such a term will ensure that the wave will decay over time and the gain term can be set according to the desired decay time. However, this is insufficient to model the behaviour of an actual piano string.

By using a test solution of the form $y(x,t) = e^{st+jkx}$, where $s = \sigma \pm j\omega$ and solving the characteristic equation for the wave equation of the piano string given in Eq.(4), it can be seen that the loss term is actually frequency dependent. Therefore, instead of having the same decay time, which is the case of using a DC gain term of < 1, different partials actually have different decay times.

Assuming that the decay times for the partials are known, the gain associated with each partial can be calculated from (Bank, 2000):

$$g_k = e^{\frac{-1}{f_0 \tau_k}} \tag{21}$$

Where g_k is the gain of the k-th partial, τ_k is the decay time of the k-th partial and f_0 is the fundamental frequency. However, the task of trying to fit a filter to the g_k coefficients specified by Eq.(21) is by no means straightforward, as the error in decay time is a non-linear function of the amplitude error. The stability of the loss filter can also become an issue. Although such a design, fitted for all the decay times of the partials would ideally give the best results, this method is generally disregarded in favor of simpler more graceful designs that give satisfactory results.

Välimäki et al. proposed a considerably simpler method that can account for the frequency dependent loss characteristics of a string by using a first order one-pole filter (Välimäki et al., 1996). The design was originally proposed for plucked string intrsuments and has shown good results for its intended purpose. This proposed filter will be used in the present model. The transfre function of such a filter is given as:

$$H_l(z) = g\frac{1 + a_1}{1 - a_1 z^{-1}} \tag{22}$$

g is the DC gain and the pole of the filter is – a_1. The gain is set according to the decay time of the first partial. The pole is then determined by adjusting a_1 while searching looking for the minimum approximation error.

Alternatively, (Bank, 2000) has suggested a method of determining the values for g and a_1. Bank noted that the decay time for the k-th partial is given by:

$$\tau_k \approx \frac{1}{b_1 + b_3\vartheta_k^2} \tag{23}$$

Where $\vartheta_k = 2\pi f_k/F_s$ and b_1, b_3 can be calculated from the parameters of the first oreder one-pole filter:

$$b_1 = f_0(1 - g) \qquad\qquad b_3 = -f_0\frac{a_1}{2(a_1 + 1)^2} \tag{24}$$

Eq.(24) can be reversed to find the filter parameters. The coefficients b_1 and b_3 here refer to the same coefficients as the ones define in the piano string wave equation defined by (Chaigne & Askenfelt, 1994). (Bank, 2000) also proposed a way to determined these two coefficients by using polynomial regression from known values of the partial decay times. g should have values within the range 0< $g < 1$ and a_1 should be negative ($a_1 < 0$).

As a rough estimate, the filter parameters can be calculated using the b_1 and b_3 values taken from (Chaigne & Askenfelt, 1994) as listed in Table 4.1. For the C4 string, g is found to be about 0.998, but the resulting a_1 has a value of 0. This is because of the very small value of b_3. If the value of b_3 is very small, then the effect of frequency dependent loss of the string is very small as well. Subsequently, if the effect of the frequency dependent loss is neglected and the normal non-frequency dependent loss is assumed to dominate, then a DC gain will be sufficient to account for the loss. This is reflected in the result of the calculation, because when b_3 is very small, then effectively the pole of the filter is 0 and the entire loss filter will only have a DC gain, g.

Without the actual decay times of the partials, a_1 cannot be accurately determined, therefore, a typical value for the a_1 is used in the model presented. The pole typically has very small and negative values, ranging from -0.001 to -0.01 (Välimäki et al., 1996). a_1 is adjusted within this range to find the most reasonable result. Note that even though the range of the pole is very small, it can have very significant effects on the decay of the output.

Aside from a simpler design, the first order one-pole filter also has another advantage in that it is by design always stable. The design of the filter requires that the pole of the filter be negative, which means that the filter is always a low-pass filter. In addition to that, the DC gain of the filter is by necessity always less than 1. This ensures the stability of the loss filter and also the digital waveguide model.

5.4 Filter Design: The dispersion filter

The role of the dispersion filter is to model the dispersive character of the string as a result of stiffness. As mentioned in earlier sections, the major effect of stiffness is that it causes the partials to be slightly inharmonic. The major task to consider when designing the dispersion filter is to try and imitate the inharmonicity of a real piano string. The phase delay that the dispersion filter introduces to the partials is given by (Bank, et al., 2003):

$$D_d(f_k) = \frac{F_s k}{f_k} - N \tag{25}$$

Where D_d denotes the phase delay of the dispersion filter and N is the length of the delay line. In the model presented, the effect of dispersion is simulated using the approach suggested by (Van Duyne & Smith, 1994). The method proposed uses first order all-pass filters to achieve this effect. The characteristic of an all-pass filter is that it introduces frequency dependent delay. However, the effect of one all-pass filter is insufficient to accurately produce the dispersion effect required. Therefore, several first order all-pass filters connected in series is used. The transfer function for one such filter is given as:

$$H_{ap}(z) = \frac{a_2 + z^{-1}}{1 + a_2 z^{-1}} \tag{26}$$

An expression for the phase response of the all-pass filter can be found by substituting $z = e^{j\omega T}$ and then performing some manipulations. The phase response will be:

$$\phi(\omega T) = \arctan\left[\frac{(a_2^2 - 1)\, sin\ \omega T}{2a_2 + (a_2^2 + 1)\, cos\ \omega T}\right] \tag{27}$$

As mentioned earlier, the effect of an all-pass filters is that it introduces frequency dependent delay, and because of that, when the all-pass filters are introduced directly into the digital waveguide model without making any changes, the position of fundamental frequency will be shifted as well. In order to make sure that the fundamental remains at the desire frequency, the delays that are introduced by the all-pass filters need to be taken into account and the length of the delay line has to be modified. This means that there are actually three unknowns that need to be determined, the number of all-pass filters to be used Q, the filter coefficient a_2 and the length of the delay line N.

The method suggested by (Van Duyne & Smith, 1994) involves starting from an arbitrary value for a_2. From there, the apparent partial numbers can be calculated. The number of filters used can then be estimated by minimizing the error between the apparent partials and the actual recorded partials. The length of the delay line can then be calculated using the equation given below:

$$N = \frac{2\pi k + Q\phi(\ 2\pi\ p(k)\ T\ )}{2\pi\ p(k)\ T} \tag{28}$$

Where $p(k)$ is the measured partial frequencies. The coefficient a_2 is set arbitrarily at the beginning of the process, but the general influence a_2 has on the other two parameters is that when a_2 is very close to zero, the partials will be more evenly stretched, but the number of filter required will increase. On the other hand if a_2 is close to -1, less filters will be needed, but the lower frequencies will experience a larger amount of stretching compared to the higher frequencies. With enough data, it is possible to optimize all three parameters.

For the purposes of this project, without the measured frequency of the partials for each note, two parameters are arbitrarily set (by trial and error to make sure that the values chosen are reasonable for a range of notes). The filter coefficient a_2 is fixed while the number of filters used is changed depending on the frequency of the note to be simulated. The length of the delay line is then calculated from the values of Q and a_2 using Eq.(28).

For simulating higher frequency notes, a smaller number of all-pass filters is used and vice versa. The reason for doing so is because as the frequency of the note increases, the length of the delay line becomes shorter and the presence of all-pass filters will shorten the delay line even more. Therefore, for high frequency notes, the number of filters used needs to be decreased so that the delay line does not become too short that the note cannot be properly simulated. As an example, 16 all-pass filters are used when simulating the C4 note, but if the same amount of filters is used for a C7 note, it will be impossible to simulate the note because the length of the delay line calculated using Eq.(28) will become negative value.

5.5 Filter Design: The tuning filter

The purpose of the tuning filter is to accurate tune the frequency of the note produced. Since the entire model is implemented digitally, the delay line length can only have integer values. However, the elements of real instruments are almost never an integer value and this has to be reflected in the model in order for it to produce an accurate tone. This is especially important for high frequencies when the length of the delay line is so short that the difference of a fraction of a unit will produce sounds that are a semitone apart.

The difference in the delay can be compensated by using a tuning filter (fractional delay filter). Jaffe & Smith suggested using a first order all-pass filter for tuning the Karplus-Strong algorithm (Jaffe & Smith, 1983), but the same filter design can be used for the piano string as well. The tuning filter is connected in series with the other filters as well and the transfer function is the same as the one used for the dispersion filter (Eq.(26)) with the only difference being the value of the coefficient. The coefficient of the filter can be found from:

$$a_2 = \frac{1-P}{1+P} \tag{29}$$

Where P is defined as the difference between exact length of the delay line and the actual length implemented. Recall that in the model, two parallel delay lines each of length M are used. M is half of the total delay line length required, $M = N/2$, and N can be determined from Eq.(28). Regardless of the N calculated from Eq.(28), M is the length that is implemented and so M is the one that must be an integer value and takes the floor value of $N/2$. Therefore, P is given as:

$$P = N - 2(floor(N/2)) \tag{30}$$

5.6 Other design considerations

While not the focus of the currently presented model, there are some additional features of the piano that needs to be taken into account. As mentioned in Chapter 2, real pianos actually use multiple strings vibrating together to produce one single note and that gives rise to a subtle beating effect and also the two stage decay. In the present model, the effect of multiple strings vibrating together is modeled by having multiple waveguide models of the string connected in parallel and then summing the individual output of the waveguide models. Each individual waveguide model is tuned slightly differently to emulate the setup of a real piano. This is done by making small variations to the tuning filter. This method is more or less an ad hoc approach to achieve a more realistic effect and is by no means an actual way of modeling the behavior of a real piano. There are some suggested methods that can more accurate model the actual behavior, including using resonators in parallel with the waveguide model (Bank, 2000). In another approach, parallel waveguide models are used, similar to the method currently being used, but a portion of the summed output is fed back into the waveguides to account for the coupling between the strings.

In addition to the coupling of the strings, the soundboard and bridge also have some significant effects on the tone produced. There are many different suggested ways of modeling the effect of the soundboard (Bank, 2000). In the strictest sense, there is the use of the finite different schemes in modeling the actual soundboard. However, solving the two dimensional model of the soundboard can become very costly. An alternative would be a digital waveguide model of the soundboard which can sometimes be challenging to design. On the other hand, there are methods that attempt to recreate the reverberation characteristics of the soundboard by using feedback delay networks.

Since the soundboard is not the focus of the model currently presented, a simpler method of estimating the effect is used. A recording of the sound produced when the body of the piano being knocked is convolved with the output of the digital waveguide model. The justification for doing that is that it provides a rough estimate of the impulse response of the piano body including the soundboard as well. It is in a way similar to the reasoning for the feedback delay network approach in that the reverberation of the piano is being modeled instead of the actual physical object causing the reverberation. The result of the convoluted output sounds warmer and less metallic than the clean output. Also, the convoluted output has a slight attacking transient which gives the sound a more realistic feel. This is probably because of the “knocking” nature of the sound used.

5.7 Discussions

The general result of the model currently presented will be discussed here. Despite its simplicity, the result from the string model is actually quite believable. The model is designed to change the parameters of the filters according to the frequency of the note to be modeled. The reason for doing this is to get more believable results. The methods suggested here can theoretical give good results without having to manually change the parameters using conditional statements. However, most of the methods require a certain amount of measured data in order to compute or estimate these parameters. Since these data is not available, the filter parameters can only be set by varying typical values until a satisfactory result is obtained.

It should be noted that while convoluting the output of the digital waveguide model with the sound of the piano body being knocked does give it a more realistic feel; the clean output from the string model is quite believable as well. The clean output has the fundamental tonal color of the piano, but feels a little “metallic”, as if listening too closely to a vibrating string only.

The model presented can simulate the full range of the piano, from A0 up to C8, in the sense that the tones produced have the correct fundamental frequency, but the quality of the sound can be a bit of a problem for extreme frequencies. Between C1 and A6 the results are mostly consistent. The higher frequencies have a problem with the attack sound resulting from the convolution. For the higher frequency notes, generally the output will become “softer” as some (if not most) of the partials will be outside the audible range. On the other hand, the sound of the piano body being knocked consists of a full range of frequencies. Therefore, as the frequency of the note increases, the knocking sound will start to dominate over the actual sound of the note. This is especially true for the highest octave (C7-C8) where the knocking sound becomes very prominent. On the other extreme end, the three lowest notes (A0,Bb0 and B0) sound slightly unnatural by comparison to the others. One possible explanation might be that low frequency notes have a lot more partials that lie within the audible range than other notes. As a result, the way the partials are dispersed can have very noticeable effects on the sound produced. Therefore, if the partials produced by the model do not match that of an actual piano note, the result is that sound will seem unnatural even though the fundamental is accurate.

6. Conclusion and Future Work

This projected presented a rough model to modelling and synthesizing the sound produced by a piano. The main focus is on the modelling of the string which was implemented using a digital waveguide model. A model for the hammer that uses the finite difference method to physically model the hammer is also presented. Most of the methods currently presented have the ability to give very good results, provided that enough measured data is available. Instead of using actual measured values, the parameters for the design use typical values that are adjusted to give a desired outcome. Although by no means precise, the result from the model presented is acoustically acceptable, given the circumstances. In addition to that, the general theories behind the modelling and synthesis of a piano sound have laid out the groundwork and fundamentals for future work.

Future Work

There are still many avenues where improvements can still be made on the model of the piano. The first is to accurately design and fine tune the filter designs in the model presented. As mentioned earlier, many of the filter designs presented depend on measured data to compute the filter parameters. If efficient measurement techniques are used and accurate data obtained, it will make the filter design much more direct and also the results much better. Additionally, better design methods can also be considered as most of the methods mentioned presently are the basic ones that can approximate the actual effect. There are more involved design techniques that can more accurately model the behaviour of the piano.

Another aspect of the model than can be improved on is to incorporate the control of the dynamic level of the sound produced. This refers to the model’s ability to produce sounds that have different levels of loudness depending on different input parameters. The output of the hammer model shows some level of control in that depending on the initial velocity, the force signal produced will differ. This level of control is, however, lost on the string model, as regardless of the force signal fed into digital waveguide model, the output is more or less at the same level. This is probably because the output of the string model is actually normalized to make sure that the higher frequency notes do not become too soft. Therefore, if more complete information regarding the strings and the hammer can be obtained, the force signal for the high frequency strings can be modelled more accurately and that would eliminate the need to normalize the output just to keep it consistent.

In addition to that, an actual model for the soundboard and bridge can also be added. As mentioned in earlier chapters, the soundboard and bridge have very significant effects on the sound produced by the piano, particularly the soundboard. However, the soundboard and the bridge are not modelled in the model presented. It would be interesting to consider a more involved method for physically modelling the soundboard just to have an even better understanding of how the soundboard operates.

Another interesting direction would be to implement the piano model in real time. A real time implementation would probably bring to light some issues that are not present in an offline implementation. A real time model would also be more robust and flexible; and in general more useful.

7. References

Askenfelt, A. (1990). *Five Lectures on the Acoustics of the Piano.* Royal Swedish Academy of Music. Retrieved from http://www.speech.kth.se/music/5_lectures/contents.html

Bank, B. (2000). *Physics-Based Sound Synthesis of the Piano.* Hungary: Budapest University of Technology and Economics.

Bank, B., Avanzini, F., Borin, G., Poli, G. D., Fontana, F., & Rocchesso, D. (2003). Physically Informed Signal Processing Methods for Piano Sound Synthesis: A Research Overview. *Journal on Applied Signal Processing*, 941-952.

Bensa, J., Bilbao, S., Kronland-Martinet, R., & Smith, a. J. (2003). The simulation of piano string vibration: From physical models to finite difference schemes and digital waveguides. *Journal of the Acoustical Society of America, 114*(2), 1095-1107.

Borin, G., De Poli, G., & Sarti, A. (1992). Sound Synthesis by Dynamic System Interaction. (D. Baggi, Ed.) *Readings in Computer-Generated Music*, 139-160.

Chaigne, A., & Askenfelt, A. (1994). Numerical Simulations of Piano Strings. I. A Physical Model for a Struck String Using Finite Difference Methods. *Journal of the Acoustical Society of America, 95*(2), 1112-1118.

Chaigne, A., & Askenfelt, A. (1994). Numerical Simulations of Piano Strings. II. Comparisons with Measurements and Systematic Exploration of some Hammer-String Parameters. *Journal of the Acoustical Society of America, 95*(3), 1631-1640.

Fletcher, N. H., & Rossing, T. D. (1998). *The Physics of Musical Instruments* (2nd ed.). New York: Springer.

Hall, D. E. (1992). Piano string excitation. VI: Nonlinear modeling. *Journal of the Acoustical Society of America, 92*(1), 95-105.

Jaffe, D. A., & Smith, J. O. (1983). Extensions of the Karplus-Strong Plucked-String Algorithm. *Computer Music Journal, 7*(2), 56-69.

Jukka Rauhala, V. V. (2006). Dispersion Modeling in Waveguide Piano Synthesis Using Tunable Allpass Filters. *Proceedings of the 9th International Conference on Digital Audio Effects*, 71-76.

Roads, C. (1996). *The Computer Music Tutorial.* Cambridge, Massachusetts: MIT Press.

Saitis, C. (2008). *Physical modelling of the piano: An investigation into the effect of string stiffness on the hammer-string interaction.* Belfast, Northern Ireland, UK: Sonic Arts Research Centre, Queen's University Belfast.

Smith, J. O. (1991). Viewpoints on the History of Digital Synthesis. *Proceedings of the International Computer Music Conference*, 1-10.

Smith, J. O. (1992). Physical Modeling using Digital Waveguides. *Computer Music Journal: Physical Modeling of Musical Instruments, Part I, 16*(4), 74-91.

Välimäki, V., Huopaniemi, J., Karjalainen, M., & Jánosy, Z. (1996). Physical Modeling of Plucked String Instruments with Application to Real-Time Sound Synthesis. *Journal of the Audio Engineering Society, 44*(5), 331-353.

Van Duyne, S., & Smith, J. O. (1994). A Simplified Approach to Modeling Dispersion Caused by Stiffness in Strings and Plates. *Proceedings of the International Computer Music Conference*, 407-410.

8. Appendix I – Wave Equation Coefficients

$$D = 1 + b_1\Delta t + \frac{2b_3}{\Delta t}$$

$$r = \frac{c\Delta x}{\Delta t}$$

$$a_1 = \frac{\left[2 - 2r^2 + \frac{b_3}{\Delta t} - 6\varepsilon N^2 r^2\right]}{D}$$

$$a_2 = \frac{\left[-1 + b_1\Delta t + \frac{2b_3}{\Delta t}\right]}{D}$$

$$a_3 = \frac{[r^2(1 + 4\varepsilon N^2)]}{D}$$

$$a_4 = \frac{\left[\frac{b_3}{\Delta t} - \varepsilon N^2 r^2\right]}{D}$$

$$a_5 = \frac{\left[-\frac{b_3}{\Delta t}\right]}{D}$$

9. Appendix II – Matlab Implementation Code

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%   Final Project: Piano Sound Synthesis
%   By: Teng Wei Jian
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% NOTES:
% 1) This is a function that synthesizes the sound of a piano. It takes the
%    frequency of the note to be modeled as an input and returns the signal
%    of the synthesized note as an output. The function will also play the
%    sound once.
% 2) The code is divided into two parts, the first part will model the
%    force signal of the piano hammer and the second part will model the
%    piano string. Extra notes regarding each part will be included at the
%    beginning of each section.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function output=piano(f0)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PART I: PIANO HAMMER MODEL
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 1) As detailed in the report, the model is defaulted to simulate the
%    force of the hammer strike for a C4 string with the initial velocity
%    of 4 m/s.
% 2) The parameters of the hammer can be changed according to the measured
%    values (if available). Otherwise the same force signal will be used to
%    simulate ALL the notes.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Fs=44100;               % Sampling frequency
N=65;                   % Number of spatial grid points
L=0.62;                 % Length of the piano wire
Ms=3.93/1000;           % Mass of the piano wire
Mh=2.97/1000;           % Mass of the hammer
K=4.5*10^9;             % Hammer stiffness coefficient
T=670;                  % Tension in the piano wire
p=2.5;                  % Stiffness non-linear component
alpha=0.12;             % Relative striking position
b1=0.5;                 % Damping coefficient
b3=6.25*10^-9;          % Damping coefficient
epsilon=3.82*10^-5;     % String stiffness parameter

v=4;                    % Initial hammer velocity

R0=sqrt(T*Ms/L);        % Wave impedance of the piano wire
i0=round(alpha*N);      % Striking position of the hammer
c=sqrt(T/(Ms/L));       % Wave speed
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Defining the coefficients of the wave equation %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
D=1+b1/Fs+2*b3*Fs;
r=c*N/(Fs*L);
a1=( 2 - 2*r^2 + b3*Fs - 6*epsilon*N^2*r^2 )/D;
a2=( -1 + b1/Fs + 2*b3*Fs )/D;
a3=( r^2*( 1 + 4*epsilon*N^2 ) )/D;
a4=( b3*Fs - epsilon*N^2*r^2 )/D;
a5=( -b3*Fs )/D;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Initializing some variables %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
input_length=150;
y=zeros(N,input_length);    % Displacement of the string
yh=zeros(1,input_length);   % Displacement of the hammer
F=zeros(1,input_length);    % Force signal output


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Initializing the values for the first few time steps of the simulation %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
y(:,1)=0;
yh(2)=v/Fs;

y(1,2)=0;
y(N,2)=0;
y(2:N-1,2)=(y(3:N,1)+y(1:N-2,1))/2;
F(2)=K*abs(yh(2)-y(i0,2))^p;

y(1,3)=0;
y(N,3)=0;
y(2:N-1,3)=y(3:N,2)+y(1:N-2,2)-y(2:N-1,1);
y(i0,3)=y(i0+1,2)+y(i0-1,2)-y(i0,1)+((1/Fs)^2*N*F(2))/Ms;
yh(3)=2*yh(2)-yh(1)-((1/Fs)^2*F(2))/Mh;
F(3)=K*abs(yh(3)-y(i0,3))^p;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Loop through the remaining time steps, implementing the finite difference
% hammer and string model.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for n=4:input_length

    y(1,n)=0;
    y(N,n)=0;
    y(2,n)=     a1*y(2,n-1)+a2*y(2,n-2)+...
                a3*(y(3,n-1)+y(1,n-1))+...
                a4*(y(4,n-1)-y(2,n-1))+...
                a5*(y(3,n-2)+y(1,n-2)+y(2,n-3));
    y(N-1,n)=   a1*y(N-1,n-1)+a2*y(N-1,n-2)+...
                a3*(y(N,n-1)+y(N-2,n-1))+...
                a4*(y(N-3,n-1)-y(N-1,n-1))+...
                a5*(y(N,n-2)+y(N-2,n-2)+y(N-1,n-3));

    y(3:N-2,n)= a1*y(3:N-2,n-1)+a2*y(3:N-2,n-2)+...
                a3*(y(4:N-1,n-1)+y(2:N-3,n-1))+...
                a4*(y(5:N,n-1)+y(1:N-4,n-1))+...
                a5*(y(4:N-1,n-2)+y(2:N-3,n-2)+y(3:N-2,n-3));

    y(i0,n)=    a1*y(i0,n-1)+a2*y(i0,n-2)+...
                a3*(y(i0+1,n-1)+y(i0-1,n-1))+...
                a4*(y(i0+2,n-1)+y(i0-2,n-1))+...
                a5*(y(i0+1,n-2)+y(i0-1,n-2)+y(i0,n-3))+...
                ((1/Fs)^2*N*F(n-1))/Ms;

    yh(n)=2*yh(n-1)-yh(n-2)-((1/Fs)^2*F(n-1))/Mh;

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Check for when the hammer is no longer in contact with the string %
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    if (yh(n)-y(i0,n))>0
        F(n)=K*abs(yh(n)-y(i0,n))^p;
    else
        F(n)=0;
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Changes the force signal into a veolcity to be fed into the digital
% waveguide model.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
v=F/(2*R0);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% PART I: PIANO STRING MODEL
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 1) For easier reference the variables used will be listed here:
%
%    al        Loss Filter coefficient
%    gl        Loss Filter gain
%    ad        Dispersion Filter coefficient
%    ap_num    Number of allpass filters used in the Dispersion Filter
%    offtune   Variation in the Tuning Filter to make sure the three
%              waveguides have different frequency
%    N         Length of the entire delay line of the waveguide model
%    M         Length of the two parallel delay lines
%    P         Difference between the exact delay line length reqruied and
%              the actual length implemented
%    C         The Tuning Filter coefficient
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%
% Initialize the output %
%%%%%%%%%%%%%%%%%%%%%%%%%
output_length=100000;
output=zeros(1,output_length);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Convolves the input signal with the recorded response of the piano boday
% being knocked.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ir=wavread('IR.wav');
v_new=conv(v,ir);
v_in=[v_new' zeros(1,length(output)-length(v_new))];
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Define/Calculate some of the parameters that will be used.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% The parameters of the filter are changed according to frequency to give a
% more consistent and normalized output.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if f0>3000
    gl=-0.997;
    ap_num=0;
    offtune=0.01;
elseif f0>1900
    gl=-0.997;
    ap_num=2;
    offtune=0.005;
elseif f0>1800
    gl=-0.997;
    ap_num=3;
    offtune=0.005;
elseif f0>1500
    gl=-0.995;
    ap_num=4;
    offtune=0.01;
elseif f0>980
    gl=-0.995;
    ap_num=6;
    offtune=0.02;
elseif f0>750
    gl=-0.993;
    ap_num=8;
    offtune=0.03;
elseif f0>390
    gl=-0.99;
    ap_num=12;
    offtune=0.04;
elseif f0>261.626;
    gl=-0.985;
    ap_num=14;
    offtune=0.06;
elseif f0>200;
    gl=-0.98;
    ap_num=16;
    offtune=0.09;
elseif f0>150;
    gl=-0.975;
    ap_num=18;
    offtune=0.13;
elseif f0>120;
    gl=-0.968;
    ap_num=20;
    offtune=0.18;
else
    gl=-0.96;
    ap_num=20;
    offtune=0.25;
end

al=-0.001;
ad=-0.30;
N_exact=((2*pi+ap_num*atan(((ad^2-1)*sin(2*pi*f0/Fs))/...
        (2*ad+(ad^2+1)*cos(2*pi*f0/Fs))))/(2*pi*f0/Fs));
M=floor(N_exact/2);
P=N_exact-2*M;
C=(1-P)/(1+P);
i0=round(alpha*M);



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Defines the transfer function for the delays and filters used:
%
%   DL1   The delay line representing the segment of the string from the
%         agraffe to the point of contact with the hammer
%   DL2   The delay line representing the segment of the string from the
%         point of contact with the hammer to the bridge
%   Hl    The Loss Filter
%   Hd    One of the the allpass filters that make up the Dipersion Filter
%   Hfd   Hfd1, Hfd2 and Hfd2 are the Tuning Filters used to tune the
%         fundamental frequency. The 3 filters are each used in one of the
%         3 parallelly connected digital waveguide models.
%
%   H     H1, H2 and H3 are the Reflections Filters for each of the 3
%         digital waveguide models. They are made up of the Loss Filter,
%         the Dipersion Filter and the Tuning Filter.
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
z=tf('z',1/Fs);
DL1=(z)^-(M-i0);
DL2=(z)^-(i0);
Hl=gl*(1+al)/(1+al*z^-1);
Hd=(ad+z^-1)/(1+ad*z^-1);
Hfd1=(C+z^-1)/(1+C*z^-1);
Hfd2=(C*(1+offtune)+z^-1)/(1+C*(1+offtune)*z^-1);
Hfd3=(C*(1-offtune)+z^-1)/(1+C*(1-offtune)*z^-1);

if C*(1+offtune)>=1                 % Makes sure the coefficient of the
    Hfd2=(1+z^-1)/(1+1*z^-1);       % Tuning Filter does not excede 1
end

H1=Hl*Hd^ap_num*Hfd1;
H2=Hl*Hd^ap_num*Hfd2;
H3=Hl*Hd^ap_num*Hfd3;



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% The filters are then combined according to the digital waveguide model of
% the piano string.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
DW1=DL1/(1+H1*DL1*DL1*DL2*DL2)+DL2*DL2*DL1*(-1)/(1+H1*DL1*DL1*DL2*DL2);
DW2=DL1/(1+H2*DL1*DL1*DL2*DL2)+DL2*DL2*DL1*(-1)/(1+H2*DL1*DL1*DL2*DL2);
DW3=DL1/(1+H3*DL1*DL1*DL2*DL2)+DL2*DL2*DL1*(-1)/(1+H3*DL1*DL1*DL2*DL2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% The Digital Waveguide filter is then used on the input velocity.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[b a]=tfdata(DW1,'v');
output1=filter(b,a,v_in);
[b a]=tfdata(DW2,'v');
output2=filter(b,a,v_in);
[b a]=tfdata(DW3,'v');
output3=filter(b,a,v_in);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Output of the three digital waveguides re summed together. The sum is
% then normalized. The final output is then played.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
output=output1+output2+output3;
output=output/max(abs(output))*(1 - 1/32768);
soundsc(output,Fs)
```