Digital music synthesis using DSP

Rahul Bhat (124074002), Sandeep Bhagwat (123074011),
Gaurang Naik (123079009), Shrikant Venkataramani (123079042)
DSP Application Assignment, Group No. 4
Department of Electrical Engineering, Indian Institute of Technology Bombay

***Abstract*—The Piano or the Keyboard is one of the simplest and most widely used musical instruments. Similar to other musical instruments like the Guitar, the piano music synthesis is based on the principle of a resonator excited by a periodic excitation from a source. This project aims to model the piano using simple digital signal processing techniques. The report first deals with the internal working of the piano. The first step in the direction of modeling is a simple physical model which aims at directly modeling the output of the resonator. Another physical modeling technique is to design a linear shift invariant filter which would produce an equivalent output signal. Two such techniques viz. The Source Filter model involving a cascade realization and The parallel realization are discussed.**

I. Introduction

Sound waves are nothing but pressure waves which travel across a medium and reach our ears. Most of the waveforms encountered in electrical engineering applications are transverse in nature, i.e. displacement of the particles in the medium is in a direction perpendicular to the propagation of the waves. Sound waves, on the other hand, are longitudinal in nature, i.e. displacement of the particles in the medium is along the direction of propagation of waves.

Music is a form of sound that is aesthetically pleasing. As the popular online encyclopedia Wikipedia puts it, Music is an art form whose medium is sound and silence. Music is produced by means of a musical instrument. Several musical instruments such as a guitar, table, violin, piano etc. are known to many. While these instruments are used to produce sound in different environments, the underlying principle by which they produce music i.e. sound is similar in case of many of these instruments.

Music can be distinguished from other sounds [1] owing to some of its distinct properties. They are pitch, dynamics, tone color and duration. Pitch of the sound is its relative lowness or highness when the music reaches our ears. The pitch of the sound is closely related to the frequency of waveform producing it. Dynamics implies the loudness or softness of the music that reaches our ears. Tone color makes us distinguish between different instruments playing music at the same tone and same dynamic level. Duration, as the name suggests, is simply the time interval for which the tone lasts. Music, or human speech can be characterized and studied in the time or frequency domain. Like all other branches of electrical engineering, frequency domain analysis provides several advantages over time domain analysis and hence, is widely used in music analysis and synthesis.

Formants [2] are often used for characterizing speech in frequency domain. They are the distinguishable frequency components of music or any other form of sound. Any distinguishable information between the vowels can be represented by the frequency components of the vowel sounds. Formants are simply the amplitude peaks in the frequency spectrum of the sound. The formant with lowest frequency is termed as f1. The formant with second lowest frequency is termed as f2, and so on. Vowels may have multiple formants which may go above 4. However for all practical purposes, the first two formants, i.e. f1, and f2 are sufficient to distinguish a vowel sound from another. The formant frequencies for various vowels are summarized in Table 1.

TABLE I
VOWEL FORMANT CENTERS

Vowel	Formant f1 (Hz)	Formant f2 (Hz)
u	320	800
o	500	1000
a	1000	1450
e	500	2300
i	320	2500

In this paper, we synthesise music using various methods and then implement the Graphical User Interface (GUI) of a Piano using MATLAB. The outline of rest of the paper is as follows. Section II gives a brief idea about the physical structure of a piano. Section III explains two methods for modeling digital music synthesis. Section IV provides the algorithm used for implementing the models described in section III. In section V, we introduce GUI programming in MATLAB and show the implementation of the piano in the GUI. Section VI shows the various results obtained and the authors give concluding remarks in section VII.

II. Physical structure of a piano

The Piano[3] is one of the most popular music instrument and known to almost everyone. The prime component of a piano is a string, which on vibration produces the aesthetic sound. It is hence safe to say that the strings lie at the heart of a Piano and its mechanism. Figure 1 shows the simplified diagram of the physical structure of a piano key. In order to produce music, the pianist presses a key, which causes the hammer to strike the string. The strings transform part of the kinetic energy of the hammer into vibrational energy. These vibrations are passed onto the soundboard via the bridge. The soundboard then produces the desired sound.

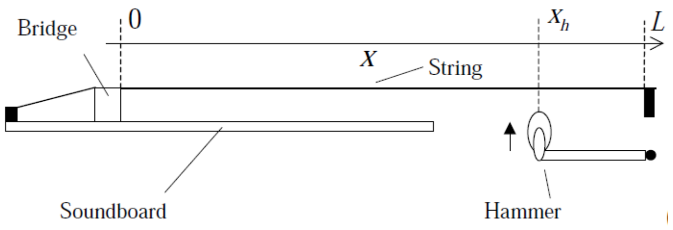

Fig. 1. A simplified diagram of the piano mechanism[4]

III. Modelling

In order to synthesise the sound digitally, it is necessary to model the piano in a manner that can be implemented with ease on a suitable platform. In this paper, we present two such models.

A. Physical Modelling

The physical model[4] of the piano consists of three parts. The first part is the hammer strike which serves as the excitation. The second part can be represented by a vibrating string which acts as the resonator as it has specific modes of vibration. It acts like a simple harmonic oscillator with a low damping factor. The third component is the soundboard of the piano. It determines the wave envelope.
Thus, the functioning of a piano can be approximated by a model of a vibrating plucked string. There are three aspects to the vibration of the string:

- Vibration: The string is constrained at both ends. When it is plucked, every point of the string is given an initial displacement. This initial displacement can be considered as resulting from the hammer strike. Then every point on the string vibrates about the mean position producing a simple harmonic motion.
- Harmonics: Since the string is fixed at two ends, it can vibrate only with certain fixed frequencies. These frequencies are determined by the tension in the string, the mass per unit length and the vibrating length of the string.
- The decay rate: After the string is released, the vibrations decay with time. This is because the initial energy imparted to the string is dissipated by different mechanisms. These are a) Stiffness of the string, b) Air resistance and c) Transfer of energy from the string to the soundboard body. Different harmonics decay at different rates.

The excited string can vibrate in two transverse and one longitudinal directions. For simplification, we consider only one direction of vibration. The vibration of the string can be modeled as a modified form of ideal wave equation in which

$$\mu\frac{\partial^2 y}{\partial t^2} = T_0\frac{\partial^2 y}{\partial x^2} - ES\kappa^2\frac{\partial^4 y}{\partial x^4} - 2R\mu\frac{\partial y}{\partial t} + d_y(x,t)$$

terms describing losses in the string are added. The equation is given as:
where,
$y(x,t)$: displacement
μ : linear density of string
T_0 : tension in the string
E : Youngs modulus of the string
S : cross section area of the string
k : radius of gyration of the string
R : frictional resistance
$d_y(x,t)$: external excitation

This equation can be solved numerically using finite difference method. But for simple linear systems, the closed form of solution is known. Computational complexity can be reduced by modeling the time domain solution rather than the wave equation itself. The string can thus be modeled as a set of second order differential equations each describing one mode of vibration. The total response of the string is a superposition of the different modal responses.
The second order differential equation describing the behavior of mode k is:

$$\frac{d^2 y_k}{dt^2} + a_{1,k}\frac{dy_k}{dt} + a_{0,k}y_k = b_{0,k}F_{y,k}(t).$$

where,
$a_{1,k}$: $2R_k$
$a_{0,k}$: $(T_0/\mu)(k\pi/L)^2 + (ESk^2/\mu)(k\pi/L)^4$
$b_{0,k}$: $2L/\mu$
$F_{y,k}(t)$: excitation force of mode k
L : length of string

The solution of the equation for an impulse input with zero initial conditions is given by:
$y_k = A_k exp(-t/\tau) sin(2\pi f_k t)$
where,
A_k : $1/(\pi L\mu f_k)$
τ_k : $1/R_k$
f_k : $f_0 k\sqrt{1+Bk^2}$
f_0 : fundamental frequency of the string
B : inharmonicity coefficient
F_0 : $(1/2L)\sqrt{T_0/\mu}$
B : $\kappa^2(ES/T_0)(\pi/L)^2$

Discretizing this modal response using impulse invariant transform and taking z-transform gives:
where,
b_k : $(A_k/f_s)Im(p_k)$
$a_{1,k}$: -$2Re(p_k)$

$$H_{\mathrm{res},k}(z) = \frac{b_k z^{-1}}{1 + a_{1,k} z^{-1} + a_{2,k} z^{-2}}$$

$a_{2,k}$: $(p_k)^2$
p_k : $exp(j2\pi f_k/f_s).exp(-1/\tau_k f_s)$
f_s : sampling frequency

Thus each mode is implemented by a two-pole filter. The net response is the sum of all modal filter responses. Its realization can be shown as:

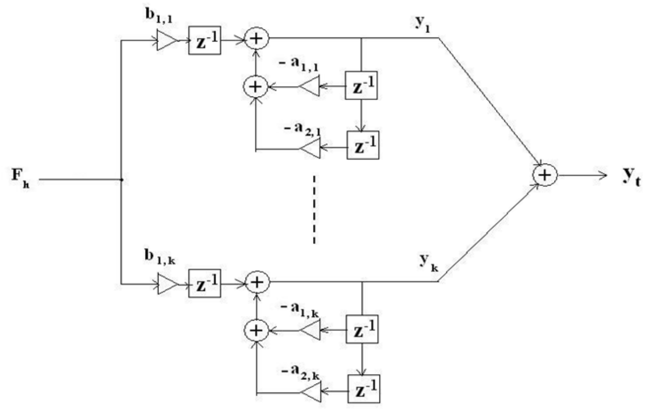

Fig. 2. Two pole filter implementation of each mode[4]

B. Source Filter Modelling

In this section, we present the human speech production system[5] and develop a model by taking parallels of each of the component in it. Figure 3 shows the human vocal system. The sub-glottal system comrpising of lungs, bronchi and

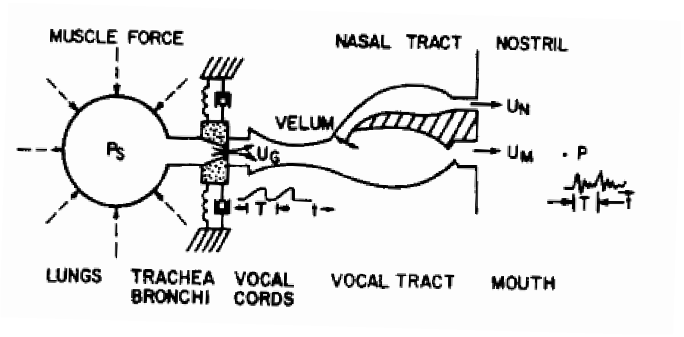

Fig. 3. Human vocal system[5]

trachea acts as the source of energy in the speech production process. Air produced by the lungs passes through the vocal tract, where the flow of air is perturbed by the constriction and thus prodcues the speech. As shown in figure 3, the vocal tract and nasal tract are tubes of non-uniform cross-sectional area. When the air passes through these tubes, its frequency spectrum is shaped according to the frequecy selectivity of these tubes. The resonant frequencies of the vocal tract tube are called as formants as discussed in section I. Different sounds are produced as a result of different shapes of the vocal tract tube. Each shape of the vocal tract tube is characterized by a set of formant frequencies.

In this paper, we propose a model in which the shape of the vocal tract is modelled by an Linear Shift-Invariant (LSI) system. The periodic burst of air produced by the lungs is modelled by an impulse train. The impulse train, when passed through the LSI sytem produces an output sequence.This sequence corresponds to the sound produced when the air passes through the vocal tract of the shape for which the LSI system is modelled.

In the human speech production system, the bursts of air produced by the lungs are always the same. Different shapes of the vocal tract tube give rise to different sounds and these correspond to different LSI systems. However, in our work, we have developed the LSI system only correspong to the sound /a/ and /i/. The prime difference between the model in our work and the human vocal system is that we pass impulse trains of different frequencies (i.e. different pitch) through the same LSI system. Increasing the frequency (pitch period) to very high values distorts the sound produced by the LSI system due to under-sampling in the frequency domain. Thus, by varying the pitch period of the impulse train, we can produce different sounds from the same LSI system. As previously mentioned, we have developed the LSI system corresponding to two sounds i.e. /a/ and /i/.

Figure 4 depicts the brief idea of our model. The output of the LSI system (corresponding to sound /a/) is the argument of the sound function provided by MATLAB. This produces the desired sound. A similar system model has also been developed for sound /i/.

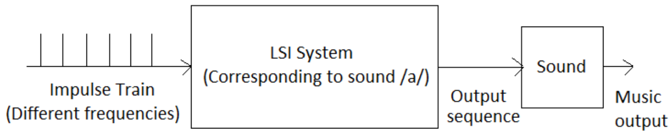

Fig. 4. Source Filtering Model

IV. MATLAB GUI Implementation

The entire implementation of the models described above has been done using MATLAB. We use the MATLAB GUI[6] development environment to develop the User Interface of the piano. The piano we have implemented is a three octave piano. Figure 5 shows the GUI of the piano.

Each key of the piano is implemented by using the pushbutton of MATLAB GUI. When we press a key, its callback function is executed. Within the callback function of each key, we specify the frequency of the sound it produces. This is done by passing an argument 'freq' to the function *Playnote* which produces the resulting sound using the *sound* function.

When we click on a particular key, the frequency of the sound is displayed in the GUI. As shown in figure 5, the key *D2* corresponds to frequency of 587.33 Hz.

The duration for which the sound lasts when the key

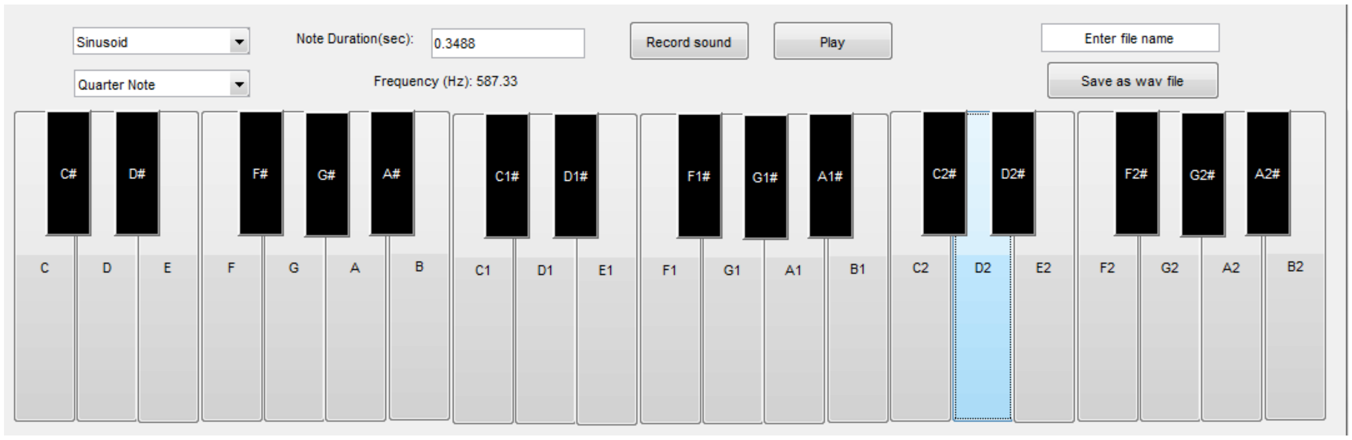

Fig. 5. Piano GUI

is pressed is called the note. We have implemented the three octave piano for the quarter note, half note and full note. Quarter note corresponds to 0.3488 seconds, while the half note and full note correspond to twice and four times the quarter note respectively. Figure 6 shows the selection of the note through the MATLAB GUI. We can also specify the note duration manually by entering the value of the note duration in the text box as shown in figure 5.

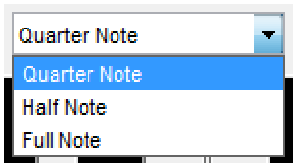

Fig. 6. Selection of Note

The sound produced by each key can be generated in different methods. Figure 7 shows the different modes of generation of sound. The sinusoidal signal is the purest signal and it contains only the fundamental frequency. The sawtooth and square signal on the other hand consist of the fundamental frequency as well as its harmonics. We can study the effect of these harmonics by using the sawtooth and rectangular modes of generation of sound.

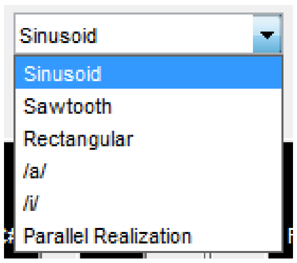

Fig. 7. Mode selection

The sampling rate used is 20kHz. Depending upon the note duration, the sampling rate of 20kHz will produce a fixed number of samples. In the first three modes, using the *sine*, *sawtooth* and *square* functions of MATLAB, we generate the corresponding samples and these samples are fed to the *sound* function. The sound function then produces the corresponding sound.

The *Parallel Realization* mode plays the sound by the mechanism explained in section III (Physical Modelling).

As shown in figure 5, we have added additional functionality to record and play the sound produced by pressing multiple keys one after another. This is done by storing the frequency corresponding to the first key in a vector and appending the frequencies corresponding to subsequent keystrokes. After several keys have been pressed, we can press the *Play* button to listen to the recorded music. This is done by passing the vector formed by appending several frequencies as an argument to the *sound* function. We can also save the recorded music to .wav file by entering the desired name in the textbox as shown in figure 5. The *Save as wav file* button generates the .wav file of the desired name and saves it in the working directory.

V. Plots and Results

Speech and music signals are, in their very basic nature, time varying signals. The resonator and/or the excitation source both undergo significant variations throughout the duration under test. Thus a simple direct application of the Fourier transform for obtaining the spectra is not a feasible idea.

However, we may use a modification of the Fourier transform to obtain the frequency domain plots. This technique is the Short-time Fourier Transform. Here, instead of applying the transformation upon the signal in its entirety, we isolate a portion of the signal using a suitable window w[n]. Thereafter, we apply the Fourier transformation in a traditional fashion. The advantage of this approach is that, we are empowered to analyse the spectral characteristics of a single phone if the window spans the duration of the phone alone. As and when required, the effect of the neighbouring phones on the phone under consideration, can also be obtained by choosing a longer window to span the adjacent phones as well. Also, this method

places no restriction on the choice of the window type which is used.
What remains to be observed is the effect of the window on the spectra. This can be explained in the form of images as in figure 8. We now plot the transform domain representation of the windowed signal to observe the effect of windowing. This is shown in figure 9. The Fourier representation 3rd waveform with the Fourier transform of the window. Here, we consider the effect of only the window main lobe. Thus windowing essentially spreads out the energy over the frequency domain. Also a short window, which thus has a larger main lobe width, will give a good temporal resolution, but will have a poor frequency resolution. On the contrary, a large window will give a good frequency resolution but a poor temporal resolution. Also windows with a shorter main lobe width, like the Hamming window are thus better, when compared to rectangular windows, which have a larger main lobe width for a window of the same length.
The frequency responses for various modes have been shown below. The results have been shown for Narrow band (30ms Hamming window) and Wideband (10ms Hamming window) for each of the six modes of operation i.e. *Sinusoidal*, *Rectangular*, *Sawtooth*, */a/*, */i/* and *Parallel Realization*.

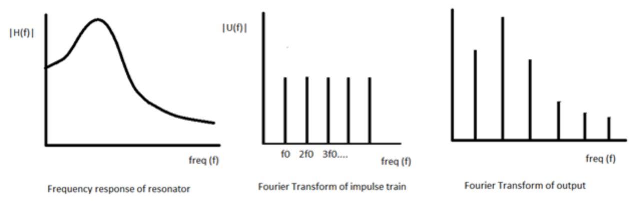
|H(f)|
freq (f)
Frequency response of resonator
|U(f)|
f0 2f0 3f0....
freq (f)
Fourier Transform of impulse train
freq (f)
Fourier Transform of output

Fig. 8. Effect of window on spectra

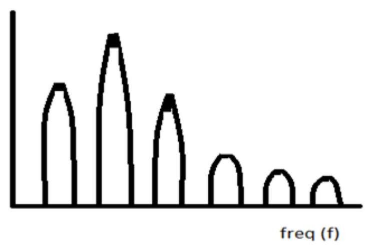

Fig. 9. Fourier domain representation of windowed signal

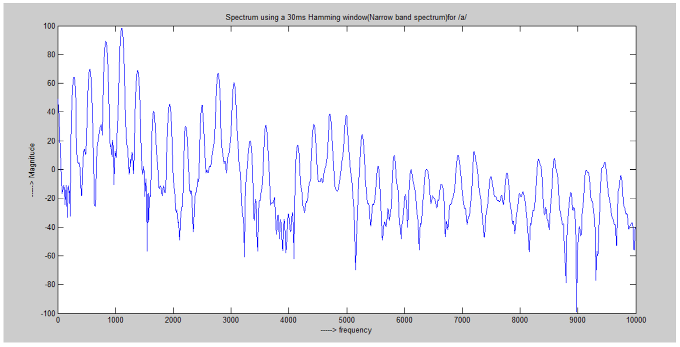

Fig. 10. Spectrum using a 30ms Hamming window(Narrow band spectrum) for /a/

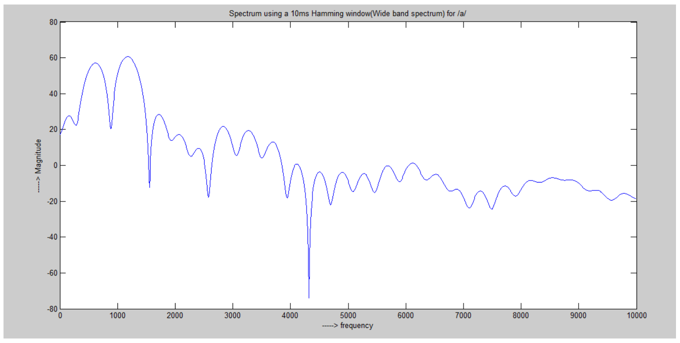

Fig. 11. Spectrum using a 10ms Hamming window(Wide band spectrum) for /a/

Figures 10 and 11 show the spectrum for the music produced by /a/ mode using a 30ms and 10ms Hamming window respectively.

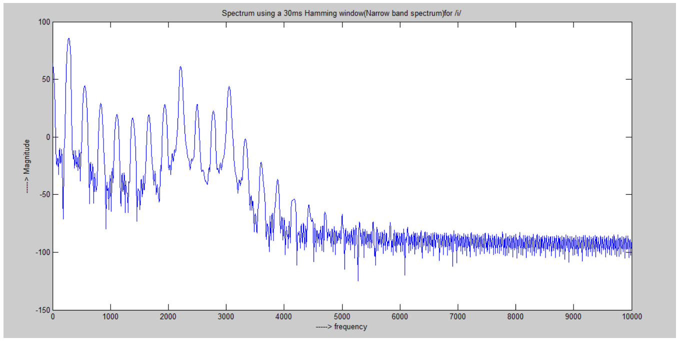

Fig. 12. Spectrum using a 30ms Hamming window(Narrow band spectrum) for /i/

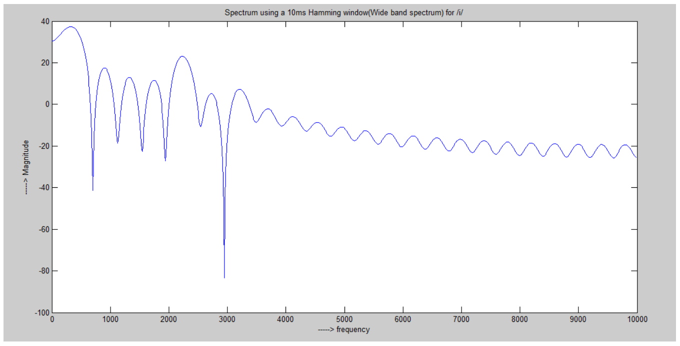

Fig. 13. Spectrum using a 10ms Hamming window(Wide band spectrum) for /i/

Figures 12 and 13 show the spectrum for the music produced by /i/ mode using a 30ms and 10ms Hamming window respectively.

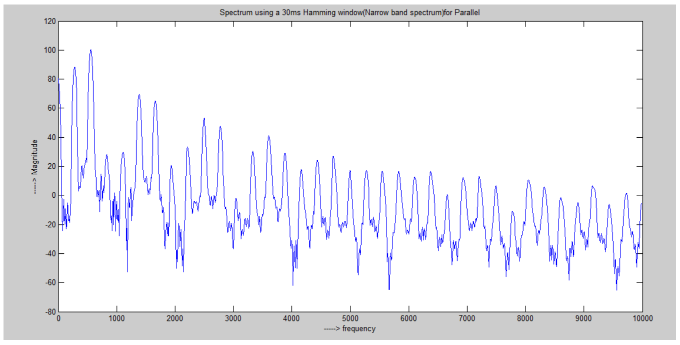

Fig. 14. Spectrum using a 30ms Hamming window(Narrow band spectrum) for Parallel Realization

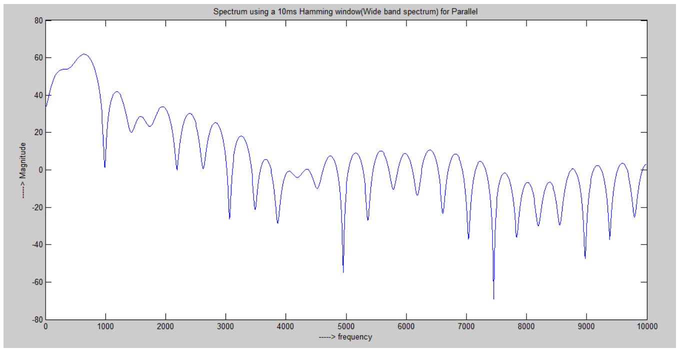

Fig. 15. Spectrum using a 10ms Hamming window(Wide band spectrum) for Parallel Realization

Figures 14 and 15 show the spectrum for the music produced by Parallel Realization mode using a 30ms and 10ms Hamming window respectively.

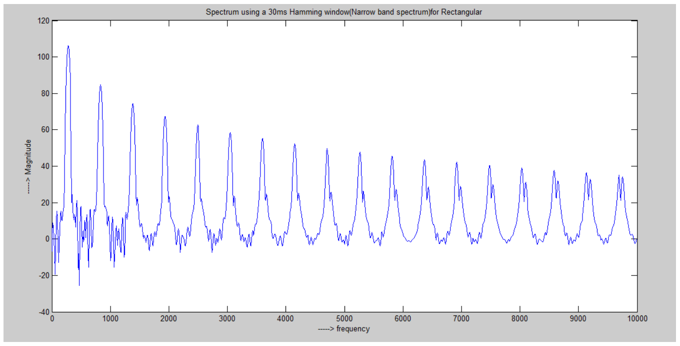

Fig. 16. Spectrum using a 30ms Hamming window(Narrow band spectrum) for Rectangular

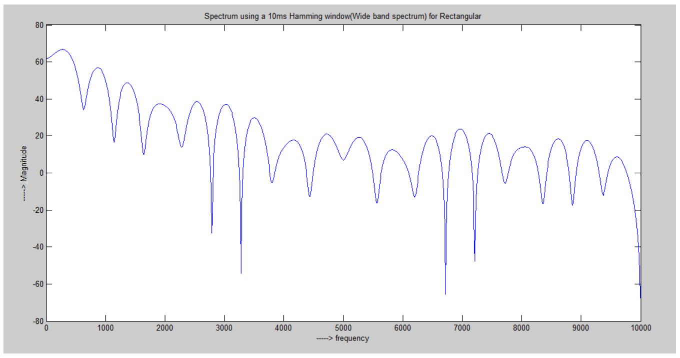

Fig. 17. Spectrum using a 10ms Hamming window(Wide band spectrum) for Rectangular

Figures 16 and 17 show the spectrum for the music produced by Rectangular mode using a 30ms and 10ms Hamming window respectively.

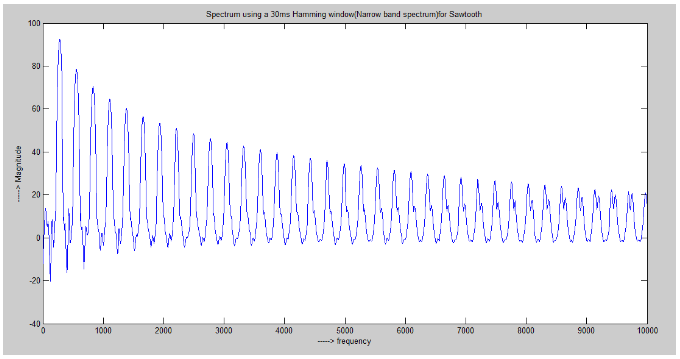

Fig. 18. Spectrum using a 30ms Hamming window(Narrow band spectrum) for Sawtooth

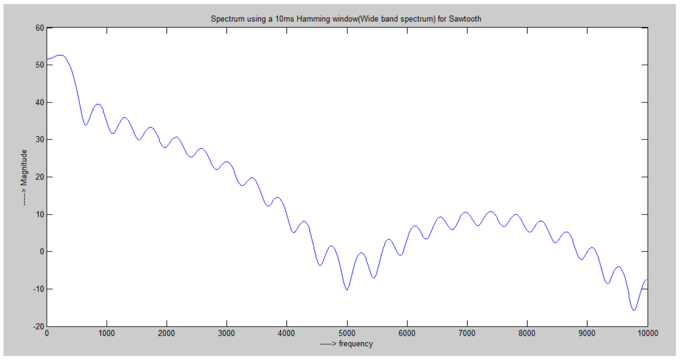

Fig. 19. Spectrum using a 10ms Hamming window(Wide band spectrum) for Sawtooth

Figures 18 and 19 show the spectrum for the music produced by Sawtooth mode using a 30ms and 10ms Hamming window respectively.

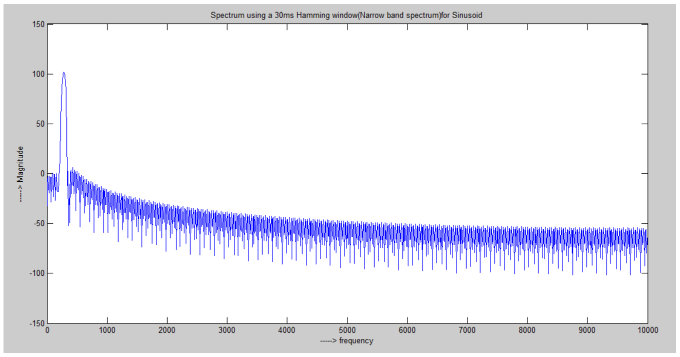

Fig. 20. Spectrum using a 30ms Hamming window(Narrow band spectrum) for Sinusoid

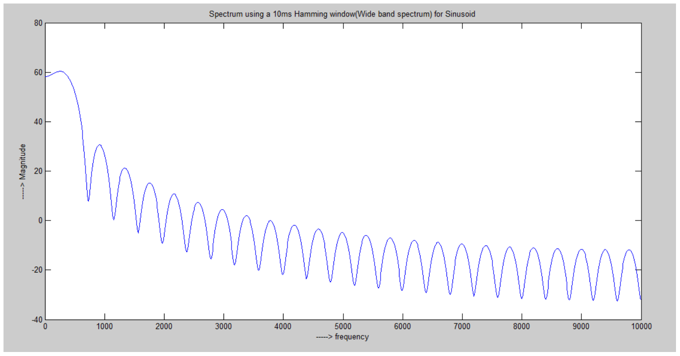

Fig. 21. Spectrum using a 10ms Hamming window(Wide band spectrum) for Sinusoid

Figures 20 and 21 show the spectrum for the music produced by Sawtooth mode using a 30ms and 10ms Hamming window respectively.

VI. Conclusion

In this project, we have implemented the GUI of a three octave piano. We incorporated several modes of production of music including physical modelling and source filter modelling. In order to study the effect of harmonics of the formants, we implemented the rectangular and sawtooth waveforms for generation of music. Provision to change the duration of the note was provided. Additional functionality such as recording and saving of music generated was also implemented in the GUI.

Acknowledgment

The authors would like to thank Prof. Vikram M. Gadre, Instructor of the course Digital Signal Processing and its Applications for giving an opportunity to study and implement one of the possible real-life implementation of DSP.

References

[1] *Elements of Music/Properties of Sound*, Available: http://historyofmusic.tripod.com/id6.html, (24 October 2012)
[2] *Formant*, Available: http://en.wikipedia.org/wiki/Formant, (24 October 2012)
[3] C. Saitis, *Physical modelling of the piano: An investigation into the effect of string stiffness on the hammer-string interaction*, Dissertation, Sonic Arts Research Centre, September 2008.
[4] B. Bank and S. Zambon and F. Fontana, *A Modal-Based Real-Time Piano Synthesizer*, IEEE TRANS. ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, VOL. 18, NO. 4, PP. 809821, MAY 2010.
[5] L. Rabiner and R. Schafer, *Digital Processing of Speech Singals*, Ninth Edition, Pearson, 2012.
[6] S. Chapman, *MATLAB Programming for Engineers*, Third Edition, CENGAGE Learning, 2003.