# MUSICAL SOUND PROCESSING

**Oleh**

**Nurulwizana Binti Kasri**

**Disertasi ini dikemukakan kepada**

**UNIVERSITI SAINS MALAYSIA**

**Sebagai memenuhi sebahagian daripada syarat keperluan**

**Untuk ijazah dengan kepujian**

**SARJANA MUDA KEJURUTERAAN (KEJURUTERAAN ELEKTRONIK)**

**Pusat Pengajian Kejuruteraan**
**Elektrik dan Elektronik**
**Universiti Sains Malaysia** **Mei 2006**

# ABSTRACT

Digital signal processors are used in a large variety of today's domestic and industrial applications. This research emphasizes on the principle and techniques on audio processing field. This research focus on the basic issues involved in the design and implementation of digital audio effects with digital signal processor. The scope of this project was carried out into 3 main parts. Part one involves creating the basic sound effects which is echo, reverberation and flanging. It created using MATLAB and SIMULINK. Part two was studying the architecture of TMS320C6416 DSP Processor and the last part was implementation using TMS320C6416 DSP Processor Board. As results these three sound effects can be heard and comparison between echo and reverberation can be performed. Usually these three basic effects manipulated by sound engineer to add special audio effects in music audio and other applications. The Echo effect is very simple and is not suggested to be used in place Reverb effect such as hall, stage and other place especially enclosed buildings. Usually these places have to be designed such that the reverberation time is adequate for the type of sound that will be produced. These sound effects can improve the quality of performance in audio and movie applications.

## ABSTRAK

Pada masa kini pemprosessan isyarat digit telah digunakan dengan meluasnya dalam kegunaan pelbagai industri. Dalam kajian ini saya lebih menitikberatkan prinsip dan teknik dalam pemprosessan isyarat audio. Kajian ini mengfokus kepada penglibatan dalam merekabentuk dan melaksanakan audio digital dan menggunakan pemprosessan isyarat digit. Skop kajian ini boleh dibahagikan kepada tiga bahagian utama. Bahagian pertama melibatkan rekabentuk kesan bunyi iaitu gema, penggemaan dan bebibir. Pada mulanya rekabentuk dilakukan di perisian Matlab. Bahagian kedua melibatkan pengenalan dan kajian kepada papan pemproses TMS320C6416. Dan bahagian yang terakhir adalah perlaksanaan menggunakan papan pemproses TMS320C6416. Penilaian dapat dilakukan dengan mendengar ketiga-tiga kesan bunyi dan perbandingan dilakukan. Pada kebiasaanya kesan bunyi ini digunakan oleh jurutera bunyi untuk menghasilkan kesan muzik yang lebih baik dan berkualiti. Selain itu kesan bunyi ini banyak digunakan dalam bidang muzik dan filem.Dalam kehidupan seharian juga kesan penggemaan digunakan contohnya di dalam sesebuah dewan jika kesan bunyi gema tiada maka bunyi yang dihasilkan dalam dewan tidak dapat didengar dengan sebaiknya.

# CONTENTS

## LIST OF FIGURES AND TABLES

**Page**

# ACKNOWLEDGMENT

First and foremost, I would like to thank the merciful Allah who given me so much in this passing years. Most of all, I would like to thank Prof. Farid Ghani for his guidance through completing this task, for his patience and wisdom. Also thank you to Mr. Suardi who kind and dedicated helps in learning the Code Composer Studio software in a short period of time.

To the all technicians especially, En. Amir Hamzah as a Digital Signal Processing Lab. technician for his co-operate along this project held. To my entire friend that share the bitter and the sweet through the year.

My deepest gratitude goes to my respective family for their continuous support and encouragement. Thank for your support. This project could not have been possible without you all co-operation.

# CHAPTER 1
# INTRODUCTION

## 1.1 Background

Sound is part of everyday life. From requisite hearing ability to entertainment in motion pictures and music, sounds have provided the impetus for audio engineers to increase quality of life or level of entertainment through their study and manipulation. In this project, I will implement a digital sound effects processor specifically focusing on basic effects such as echo, reverberation and flanging. These three effects due to their widespread use in electronic music, as well as their abilities to be used as foundations for other more complex effects. Digital sound effects processors are used mainly by musicians to create special effects such as chorus, flanging, reverberation, pitch shifting and distortion. [1]

The different vowel sounds in speech are produced primarily by changing the shape of the mouth cavity, which changes the resonances and hence the filtering characteristics of the vocal tract. The tone control circuit in an ordinary car radio is a filter, as are the bass, midrange, and treble boosts in a stereo preamplifier. Graphic equalizers, reverberations, echo devices, phase shifters, and speaker crossover networks are further examples of useful filters in audio. These effects are normally created using analogue or digital techniques. In recent years there has been a movement away from analog signal processing towards digital signal processing of music. This movement has allowed for very precise and easily reproduced effects.

By designing DSP schemes (filter) we can produce the different sound effects such as echo, reverberation and flanging. They can be done by Time – Domain operations and Frequency Domain operations.
All these tasks will be carried out through the use of the TMS320C6416 DSP Processor, as well as simulated using MATLAB.

**Design**

**Block Diagram**

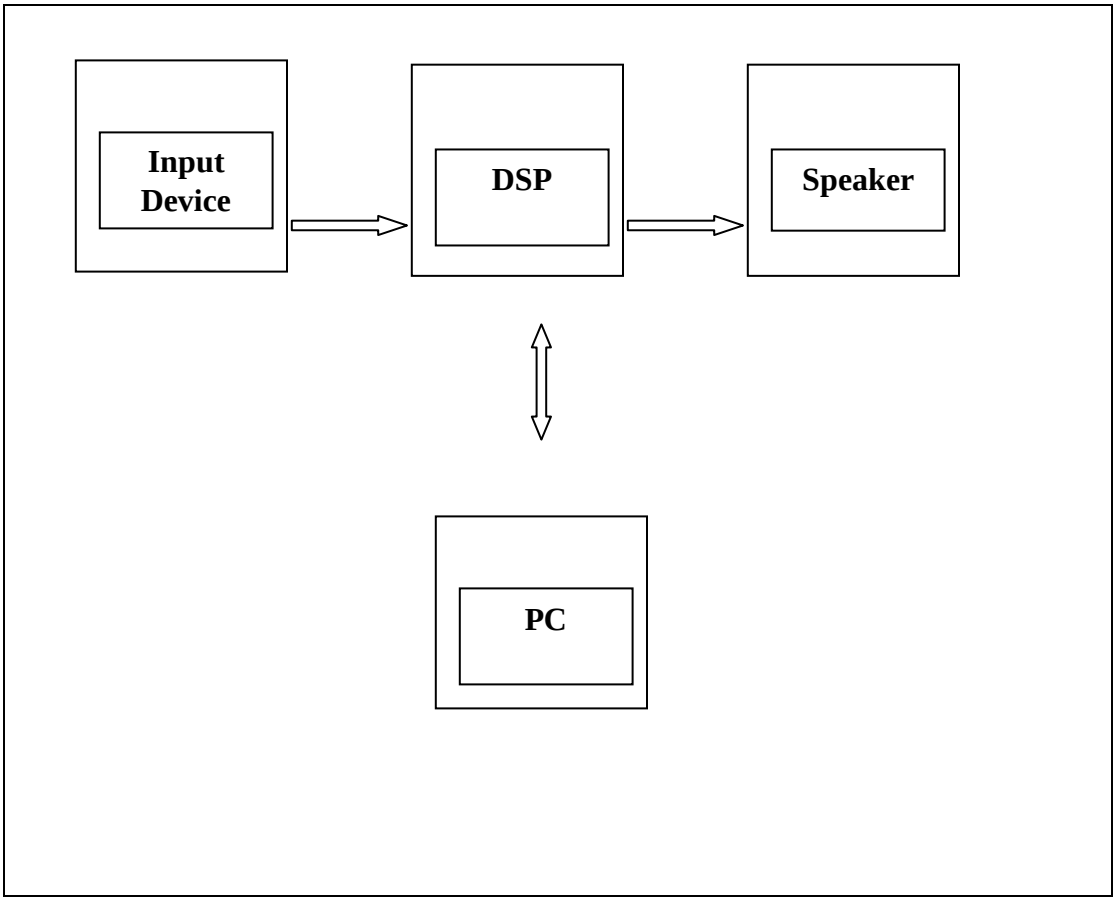



Figure 1.1 General design block diagram

## Block Descriptions

### Input Device:

This is any device such as microphone, or recorded signal that will be input into the system. The function of this block is to generate a signal that will be altered by the audio effects.

### DSP:

TI TMS320C6416 DSP Processor. This will be programmed using TI 64x assembly code and will execute all the audio effects. These audio effects will include:

- Echo
- Reverberation
- Flanging

### Speaker:

This will be used to output the signal of sound effects.

### PC:

This PC will be connected to the DSP Processor via the serial port. The PC will serve as a host for the user interface that will be created in Matlab. Through this Matlab interface, the user will be able to change settings on the audio effects. The changes will then be sent to the DSP Processor via the serial port of the computer and will write values to the DSP Processor.

## 1.2 Objectives of the Research

Investigate signal processing techniques used to process audio signals digitally. Design, build and test a digital audio effects processor to implement these techniques.
The objectives of this project are:

i. Design schemes (filters) to create sound effects using MATLAB & Simulink. The effects are echo, reverberation and flanging.
ii. Study the architecture of TMS320C6416 DSP Processor.
iii. Implemented the filters in Real Time using TMS320C6416 DSP Processor.

## 1.3 Scope of the Project

The scope of this project can be divided into three main parts. Parts one includes to study about sound effects and how to design the schemes of the effects. It involves designing the scheme (filters) to create the following audio effects. They are echo, reverberation and flanging effect. These designs consider the block diagram, transfer function and their input and output. The second part is simulation using Matlab and Simulink. The third part includes studying the architecture of TMS320C6416 DSP processor. Later the audio effects are implemented in real time using the DSP Processor. These projects concerned on the understanding with filters which will produce a better and clearer output effect.
In this project we have describe some of the filters we used , provided links to the Matlab code the response of the filters and the input and output sound files.

### 1.4 Organization of the Report

The project will be carried under the following main steps:

**a) Study the Sound Processing and designing the schemes**.

- Sound processing is one of the many applications of Digital Signal Processing. The audio effects are artificially generated using various signal processing circuits and devices and they are increasingly being performed using digital signal processing techniques. In recent years there has been a movement away from analog signal processing towards digital signal processing of music. This movement has allowed for very precise and easily reproduced effects.
- We have to determine the differential equation, transfer function and block diagram to design the schemes. In each case a definition of the effect, how the effects produced and applications of the effect will be given.

**b) Designing filter to create audio effects using MATLAB and SIMULINK**

- Matlab (short for *Matrix Laboratory*) is a language for technical computing, developed by the *The Math works, Inc*. It provides a single platform for computation, visualization, programming and software development. All problems and solutions in Matlab are expressed in notation used in linear algebra and essentially involve operations using matrices and vectors.

Will be using Matlab to solve problems in

- Circuits
- Communication systems
- **Digital signal processing**
- Control systems
- Probability and statistics

- Filters will design to create the audio effects. They are echo, reverberation and flanging effect.
- Initially MATLAB and SIMULINK will used to design the filter to produce the audio effects.

**c) Study the architecture of TMS320C6416 DSP Processor**

- The TMS320C6146 DSP Processor is available at the DSP lab of the Electrical & Electronic department. Actually there are three types/versions of the DSP processor. First for application that focuses on digital control the TMS320C2000 platform offers the most controlled optimized DSPs in the world. For application that requires low power consumption the TMS320C5000 platform offers good power efficient performance. For high performance application the TMS320C6000 platform of DSPs is considered the highest performance DSPs in the world.

**d) Real Time realization on the DSP Processor.**

- A real-time process is a task which needs to be performed within a specified time limit.
- The audio effect will implement in real time using TMS320C6416 DSP processor. Real-time implementation is very important from a cost point of view. Any speech coding algorithm can be implemented using available digital signal processor (DSP) chip technology, but the cost of that implementation will increase rapidly with the increase in the number of DSP chip used. The other important consideration in real-time implementation is the power consumption of the final product.

# CHAPTER 2
# CREATING SOUND EFFECTS

## 2.1 Introduction

**Sound effects** or **audio effects** are artificially created or enhanced sounds, or sound processes used to emphasize artistic or other content of movies, video games, music, or other media.

In motion picture and television production, a sound effect is a sound recorded and presented to make a specific storytelling or creative point without the use of dialogue or music. The term often refers to a process applied to a recording, without necessarily referring to the recording itself. In professional motion picture and television production, the segregations between dialogue, music, and sound effects recordings are quite severe, and it is important to understand that in such contexts dialogue and music recordings are never referred to as sound effects, though the processes applied to them, such as delay, echo, reverberation or flanging. [2]

## 2.2 Echo

The main concept behind the echo is delay. An echo is simply a slightly delayed repetition of a sound. In our acoustic environment echoes are heard as the result of sound waves being reflected from a surface and returned to the listener. The time difference between when the listener hears the direct sound and when the reflection (echo) is heard is referred to as the echo delay time. This delay time depends on the distances between the listener, the sound source and the reflecting surface, with greater distances resulting in longer delay times.

The popular technique for creating echo made use of the tape recorder. The principle is when sound signal was recorded on tape and immediately played back to provide a copy of the original signal, the delay by the amount of time it took tape to travel from the record head of the recorder to the playback head. When the delayed signal was mixed

with the original signal over loudspeakers, the delayed signal was heard as an echo of the original sound. Tape echo has continued to be a popular musical effect, it useful for recording and performing. [3]

### 2.3 Reverberation

Reverb is used to simulate the acoustical effect of rooms and enclosed buildings. It is the result of too many reflections of a sound that occur in room. The reverb effect is often used in audience hall. If the reverb time is too long, the room can make speech difficult to understand, too short a reverb time will make a room sound dry and lifeless.

Reverb is one of the most interesting aspects of digital signal processing effects for audio. It is a form of processing that is well-suited to digital processing, while being completely impractical with analog electronics. Because of this, digital signal processing has a profound affect on our ability to place elements of our music into different "spaces."

In the recording studio it is frequently desirable to add reverberation to recorded tracks. The recorded track to which reverberation is to be added is played back through the loud speakers, thus stimulating the chamber’s natural reverberation. [3]

**Comparison between Echo and Reverberation.**

- Echo refers to a discrete of an original sound. Reverb refers to persistence of sound in a room after the original sound has ceased.
- Echo generally implies a distinct, delayed version of a sound as you would hear with a delay meanwhile with reverb each delayed sound wave arrives in such a short period of time that we do not perceive each reflection.
- An echo the delay time is long, while reverb is short.
- Echo is a sound phenomenon that we are accustomed in our outdoor environment, whereas indoors the sound reflections are so numerous that we usually hear reverberation.

## 2.4 Flanging

When two identical signals are combined there are normally no unusual effects. However if one signal delayed slightly in time and then the signals are combined the result is a series of cancellations of certain frequencies. The canceled frequencies depending on the precise amount of time delay between two signals. In days gone by, flanging used to be created by sound engineers who put their finger onto the tape reel's flange, thus slowing it down. Two identical recordings are played back simultaneously, and one is slowed down to give the flanging effect. [3]

Flanging has become a very popular effect both for recording and for on-stage use. The flanging effect produces a pronounced whooshing sound, similar to a jet aircraft on take off. [4]

## 2.5 Designing DSP schemes

### Representation of systems

System are normally represented in the following two manners

1. Block Diagram representation
2. Mathematical models

### Block Diagram representation

In this form of representation each block is an operation on the input signal and the operation is shown on the block itself. Some typical operations include summation, delay, and multiplication.

**Mathematical model representation**

The mathematical model of a system consists of mathematical equations relating the signal interests. System can be modeled using

1. Difference equation
2. Impulse response
3. The transfer function

From transfer function we can define the type of filter.
When designing and analyzing digital signal processing algorithms, the difference equation model is commonly not the first choice. Quite often, for instance the transfer function model described below is a better tool. However, when it comes down to writing the actual computer program code to implement the algorithm, it has to be done using a difference equation model.

Digital filters are fundamental to digital audio processing.

Digital filters are implemented according to one of two basics principles, according to how they respond to an impulse:

Infinite impulse response (IIR)
Finite impulse response (FIR)

### 2.5.1 Design an echo scheme

Echo is produced by adding a time-delayed signal to the output. This produces a single echo. Multiple echoes are achieved by feeding the output of the echo unit back into its input through and attenuator. The attenuator (delay) determines the decay of the echoes, which is how quickly each echo dies out.

Theoretically, feedback can be used to produce a sound which continues forever, but practically at some point the sound level will fall below and no longer be audible. A gain is added after the delay to allow the echo to eventually decrease to zero. The gain is number between 0 and 1.The user can vary the gain of echo. The smaller value will make it die out even more quickly. The echo delay can be changed on the ranges from microseconds to several seconds. [3]

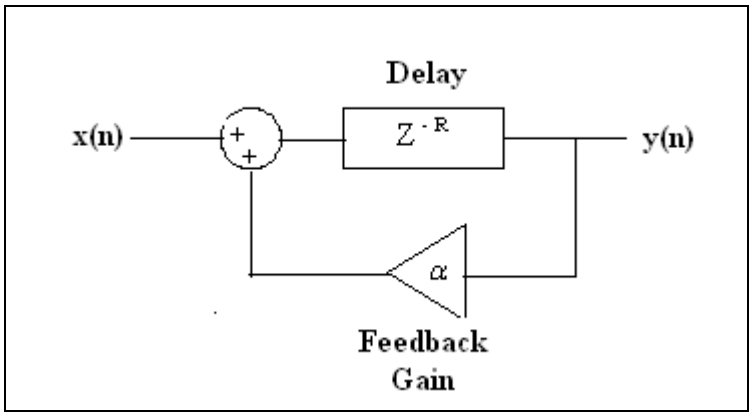



Figure 2.1: Block diagram of Echo

**Difference equation of Echo**

$$y(n) = x(n-R) + \alpha y(n-R)$$
$$y(n) + \alpha y(n-R) = x(n-R)$$

where:

$y(n-R)$ is sound sample from the past with is corresponding gain which actually creates the echo

$R$ is user adjustable delay

$\alpha$ is feedback gain <1

The corresponding transfer function

$$Y(z) - Y(z)\alpha Z^{-R} = X(z)Z^{-R}$$

$$Y(z)(1 - \alpha Z^{-R}) = X(z)Z^{-R}$$

$$\frac{Y(z)}{X(z)} = \frac{Z^{-R}}{1 - \alpha Z^{-R}}$$

**Transfer Function**

$$H(z) = \frac{Z^{-R}}{1 - \alpha Z^{-R}}$$

Impulse Response of Echo. [2]

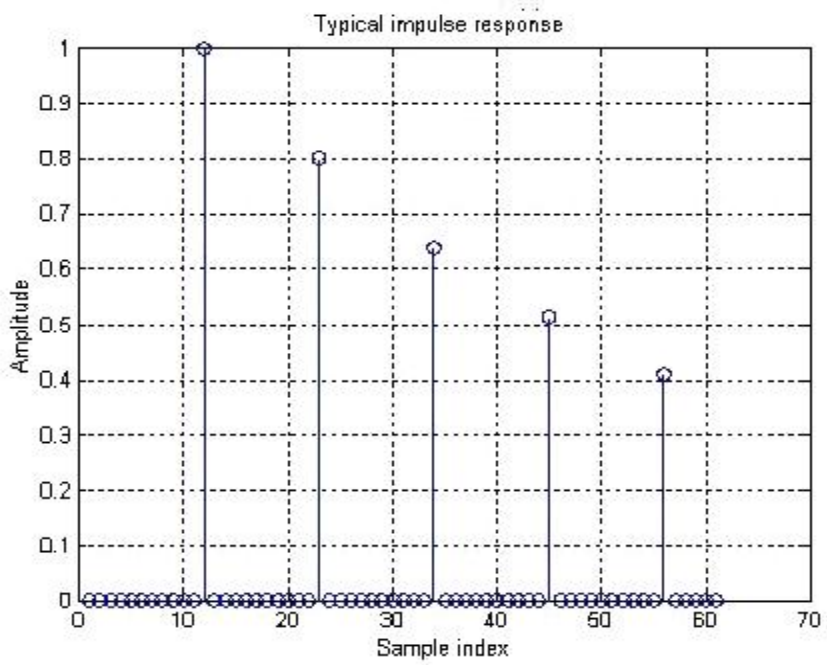



Figure 2.2: Impulse Response of Echo

### 2.5.2 Design Reverberation scheme

The reverberation was created with a feedback loop that takes the output of the delay and adds it to the input. A gain is added after the delay to allow the reverb to eventually decrease to zero. The reverb delay can be changed on the interface and ranges between 0 and 500ms. The user can also vary the gain of the reverb. A block diagram of the reverb effect is shown below.

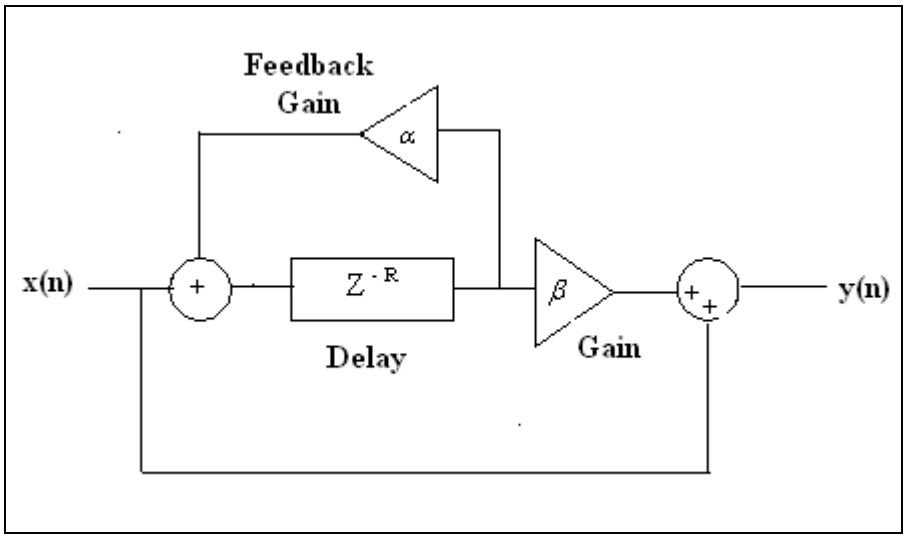



Figure 2.3: Block diagram of Reverberation

**Difference equation of Reverberation**

$$y(n) = x(n) - \alpha x(n-R) + \beta x(n-R) + \alpha y(n-R)$$

where:

$R$ is the delay

$\alpha, \beta$ is the gain which is the value 0<gain<1

$$Y(z) = X(z) - X(z)\alpha Z^{-R} + X(z)\beta Z^{-R} + Y(z)\alpha Z^{-R}$$

$$Y(z) - Y(z)\alpha Z^{-R} = X(z) - X(z)\alpha Z^{-R} + X(z)\beta Z^{-R}$$

$$Y(z)(1 - \alpha Z^{-R}) = X(z)(1 - \alpha Z^{-R} + \beta Z^{-R})$$

$$\frac{Y(z)}{X(z)} = \frac{1 - \alpha Z^{-R} + \beta Z^{-R}}{1 - \alpha Z^{-R}}$$

**Transfer Function**

More information how to get the transfer function of Reverberation is mentioned below. To get transfer function of Reverberation figure 2.3 can be divided into two parts

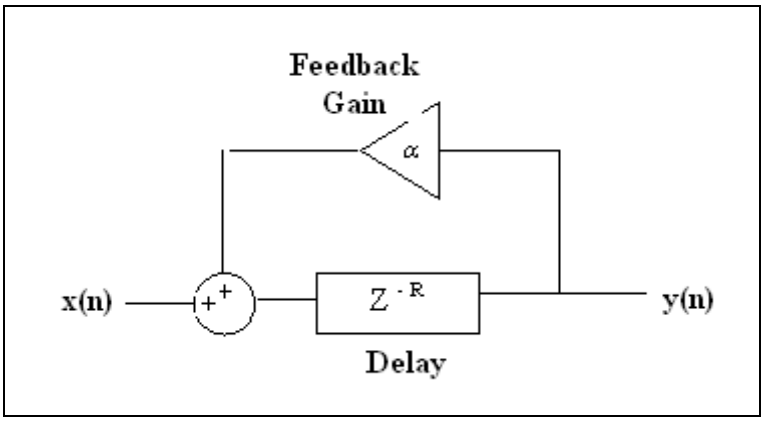



Figure 2.3.1: Block diagram to get transfer function of Reverberation (part1)

$$y(n) = x(n - R) + \alpha y(n - R)$$

$$y(n) + \alpha y(n - R) = x(n - R)$$

$$H'(z) = \frac{Z^{-R}}{1 - \alpha Z^{-R}}$$

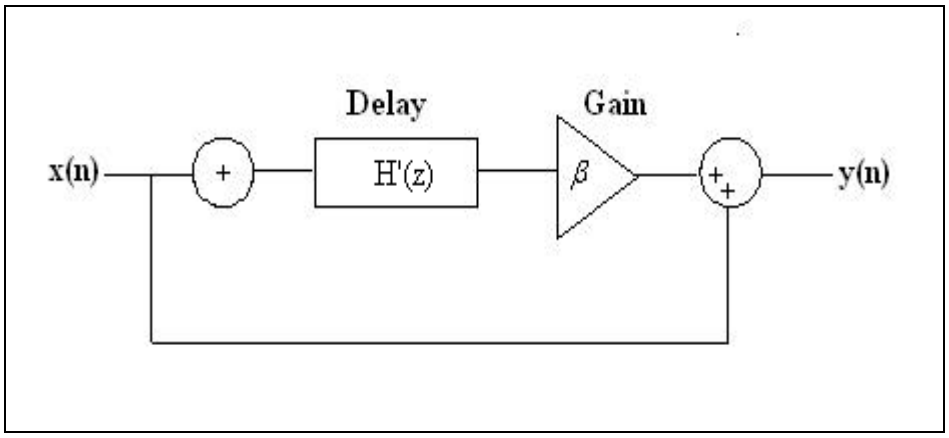



Figure 2.3.2: Block diagram to get transfer function of Reverberation (part2)

$$y(n) = x(n) + \beta x\left(n - H'(z)\right)$$

$$Y(z) = X(z) + \beta X(z)H'(z)$$

$$Y(z) = X(z)\left(1 + \beta H'(z)\right)$$

$$H(z) = 1 + H'(z) * \beta$$

$$H(z) = 1 + \frac{\beta Z^{-R}}{1 - \alpha Z^{-R}}$$

$$H(z) = \frac{1 - \alpha Z^{-R} + \beta Z^{-R}}{1 - \alpha Z^{-R}}$$

Impulse response of Reverberation. [2]

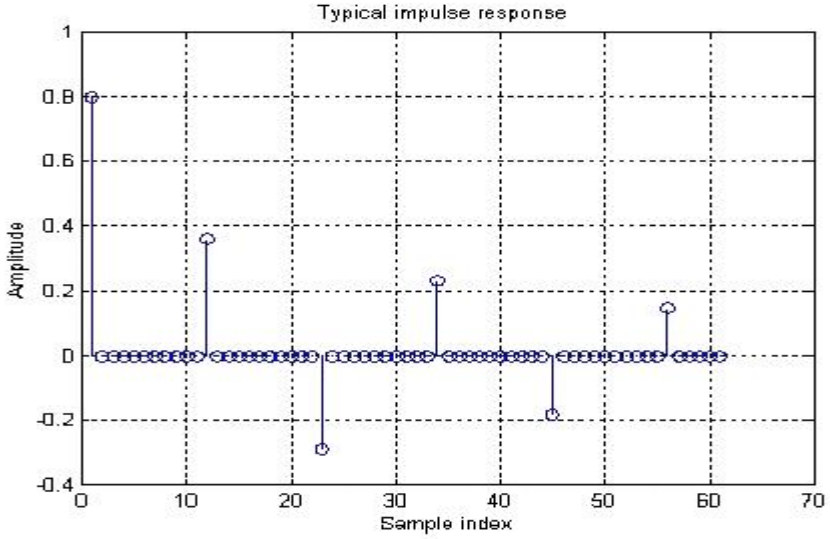



Figure 2.4: Impulse response of Reverberation

### 2.5.3 Design Flanging scheme

Typically, the delay of the echo for a flanger is varied between 0ms and 5ms at a rate of 0.5Hz. When the time delay is increasing the notches sweep from high to low, and as the delay is decreased the notches sweep low to high. The time delays required for the effect are quite short, about one millisecond. However, sweeping the comb filter through the spectrum requires that the delay be varied from about 0.1 msec. up to about 10 msec. [7]

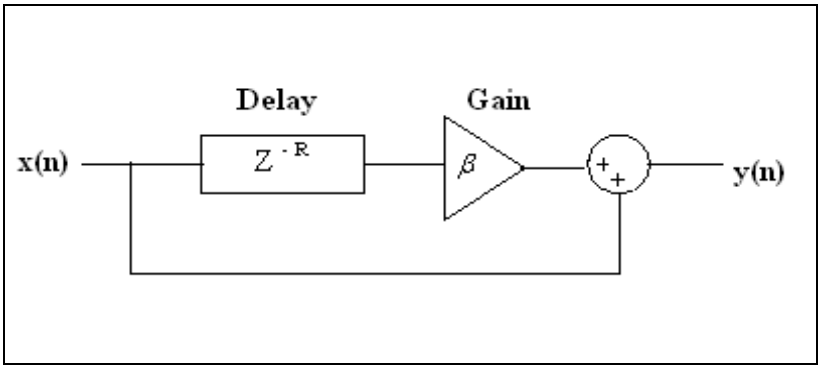



Figure 2.5: Block diagram of Flanging

**Difference equation of flanging:**

$$y(n) = x(n) + \beta x(n - R)$$

where:

$R$ is discrete delay

$\beta$ is proper scaling factor decay (gain)

$$Y(z) = X(z) + X(z)\beta Z^{-R}$$

$$\frac{Y(z)}{X(z)} = 1 + \beta Z^{-R}$$

**Transfer function**

$$H(z) = 1 + \beta Z^{-R}$$

Impulse response of Flanging.

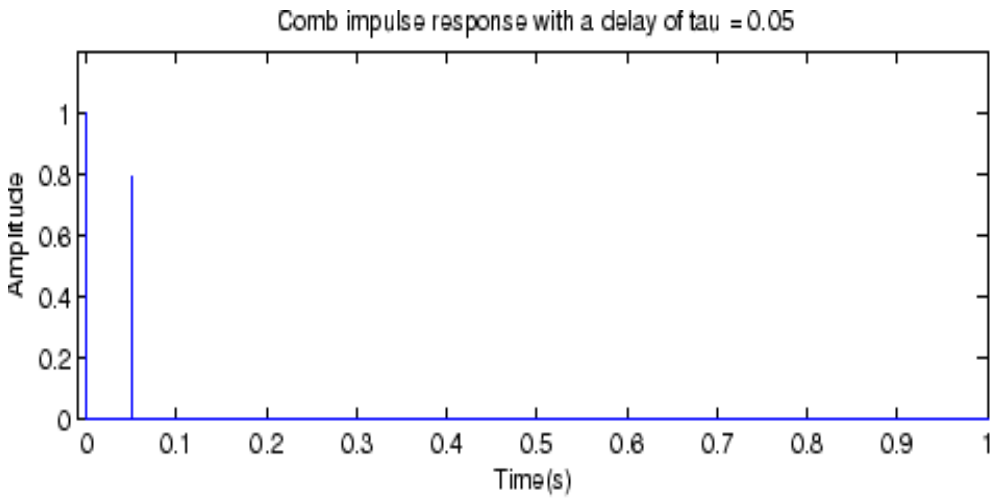



Figure 2.6: Impulse response of Flanging

# CHAPTER 3

# SIMULATION USING MATLAB & SIMULINK

## 3.1 Simulink Model of Echo

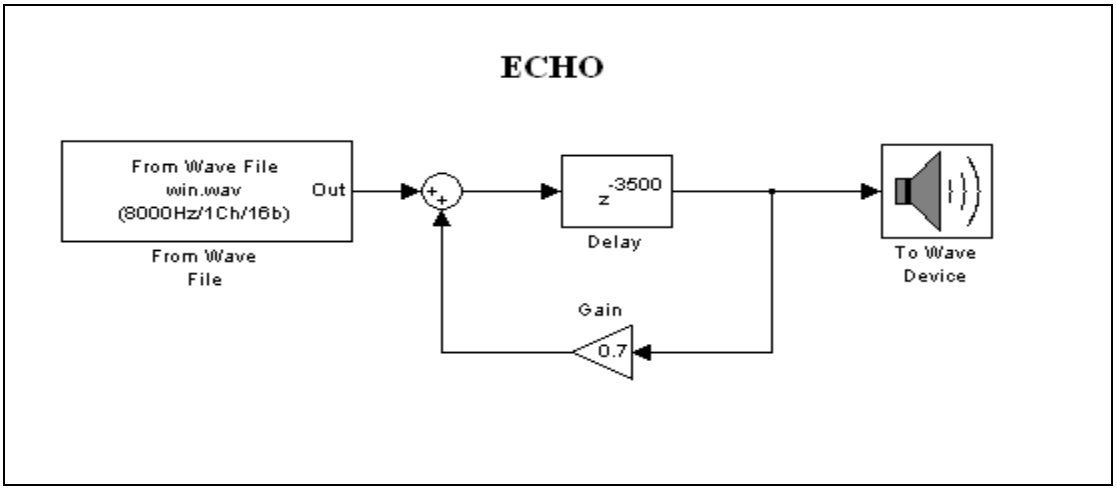



Figure 3.1**:** Simulink Model of Echo

## 3.2 Simulink Model of Reverberation

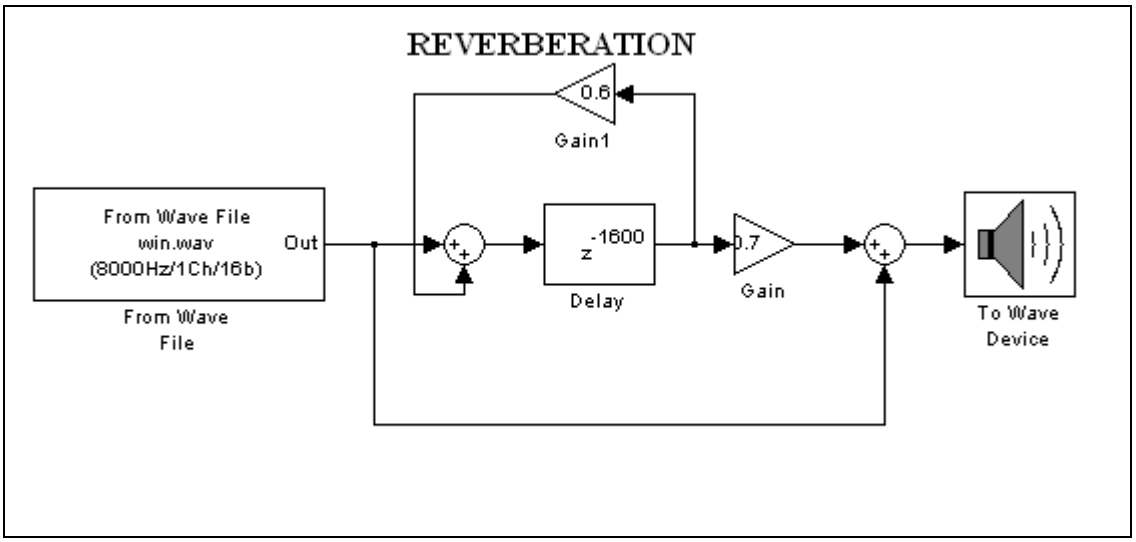



Figure 3.2: Simulink Model of Reverberation

### 3.3 Simulink Model of Flanging

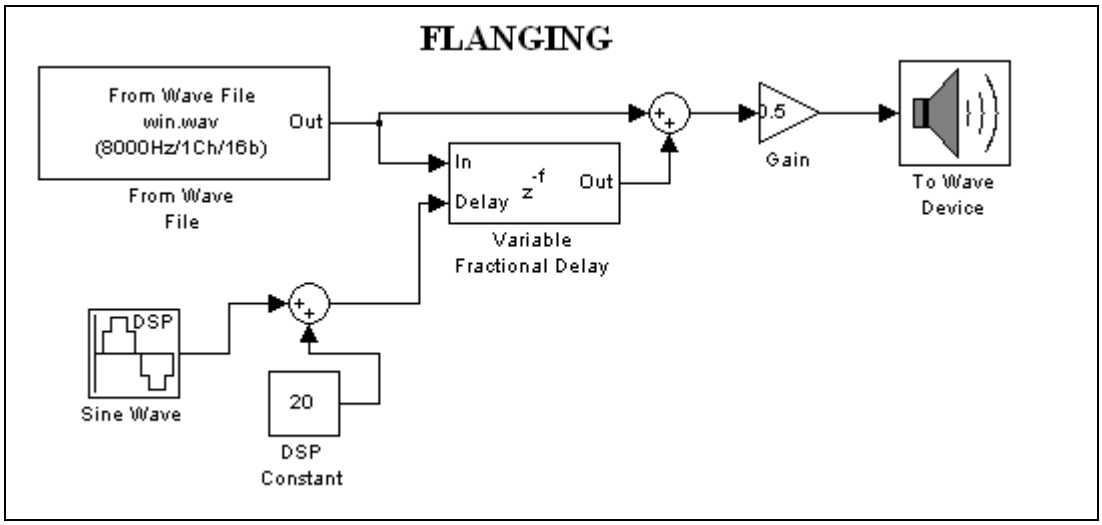


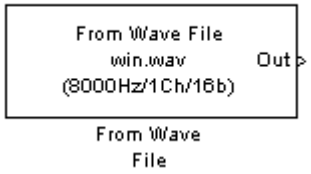

Figure 3.3**:** Simulink Model of Flanging

### 3.4 Block Description

#### 3.4.1 From Wave File

**Purpose**: Read Audio data from a Microsoft Wave (.wav) file.

**Symbol**:




**Library**: Platform-specific I/O / Windows (WIN32)

**Description**:

The From Wave File block reads audio data from a Microsoft Wave (.wav) file and generates a signal with one of the data types and amplitude ranges in the following table:

Table 3.1: Output data type and its amplitude range

| Output Data Type | Output Amplitude Range |
|---|---|
| `double` | ±1 |
| `single` | ±1 |
| `int16` | -32768 to 32767 ($-2^{15}$ to $2^{15}$ - 1) |
| `uint8` | 0 to 255 |

The audio data must be in uncompressed pulse code modulation (PCM) format. The block supports 8-, 16-, 24- and 32-bit Microsoft Wave (.wav) files. [MATLAB 7.0]

**Dialog Box:**

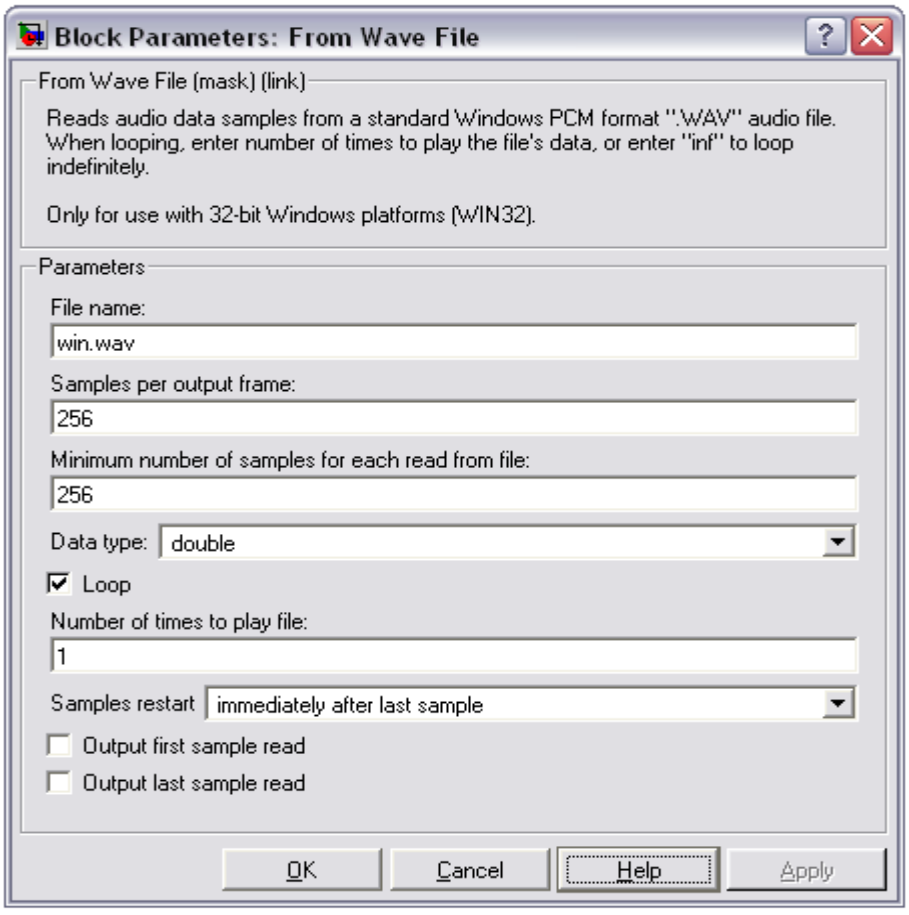



Figure 3.4: Block parameters dialog box for From Wave File.

**File name**

Contains the path and name of the file to read. For this model the file name is win which is saved in C: \MATLAB7\work folder

**Samples per output frame**

Shows the number of samples in each output frame.

**Minimum number of samples for each read from file**

Shows the number of consecutive samples to acquire from the file with each file access.

**Data type**

Verify the output data type: double, single, unit8 or int16. The data type setting determines the output's amplitude range. Double is selected for this model.

**Loop**

The check box is selected to play the file more than once.

**Sample restart**

To repeat the audio file immediately, immediately after last sampled is selected. At the beginning of next frame is selected to place the first sample of the audio file in the first position of the next output frame.

**Output first sample read.**

To determine whether the frame output from the output port contains the first sample of the audio file.

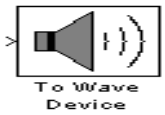
**Output last sample read.**

To determine whether the frame output from the output per contains the last sample of the audio file.

### 3.4.2 To Wave Device

**Purpose**: Send audio data to a standard audio device in real-time

**Symbol**:




**Library**: Platform-specific I/O / Windows (WIN32)

**Description**:

The To Wave Device block sends audio data to a standard Windows audio device in real time. It is compatible with most popular Windows hardware, including Sound Blaster cards. The amplitude of the input must be in a valid range that depends on the input data type (see the following table). Amplitudes outside the valid range are clipped to the nearest allowable value.

Table 3.2: Input data type and its amplitude range

| Input Data Type | Valid Input Amplitude Range |
|---|---|
| double | ±1 |
| single | ±1 |
| int16 | -32768 to 32767 ($-2^{15}$ to $2^{15}$ - 1) |
| uint8 | 0 to 255 |

The data is sent to the hardware in uncompressed pulse code modulation (PCM) format, and should typically be sampled at one of the standard Windows audio device rates: 8000, 11025, 22050, or 44100 Hz. Some hardware might support other rates in addition to these.

**Dialog Box:**

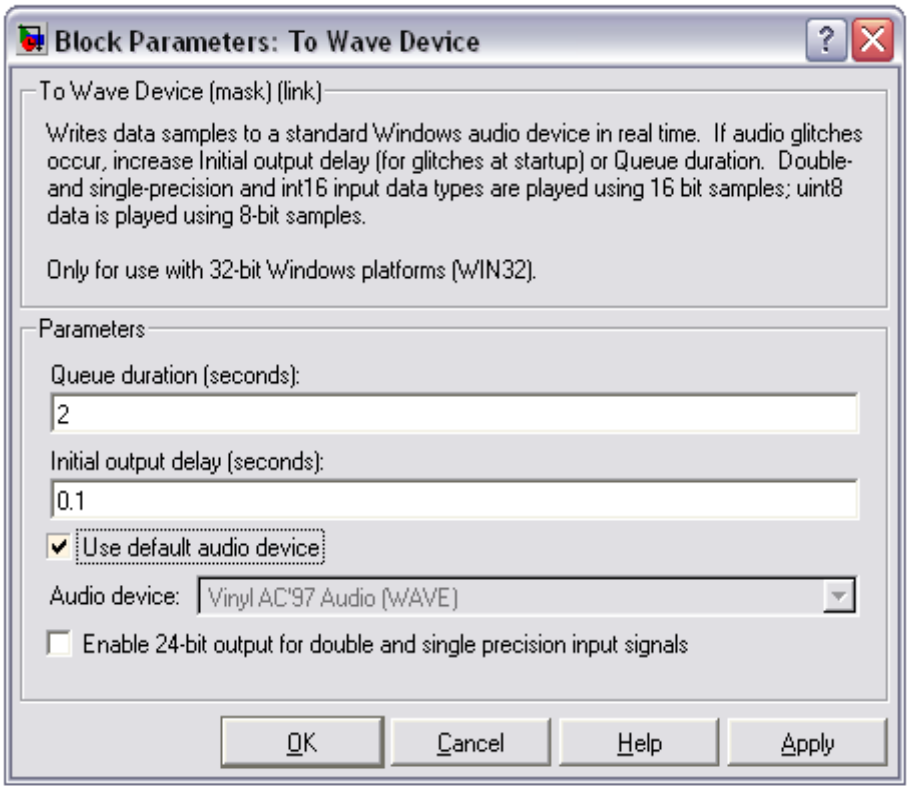



Figure 3.5: Block parameter dialog for To Wave Device

**Queue duration (seconds)**

The length of signal (in seconds) to buffer to the hardware at the start of the simulation.

**Initial output delay (seconds)**

The amount of time by which to delay the initial output to the audio device. A value of 0 specifies the smallest possible initial delay, a single frame.

**Use default audio device**

Directs audio output to the system's default audio device when selected. Clear to enable the Audio device parameter and select a device.

**Audio device**

The name of the audio device to receive the audio output (lists the names of the installed audio device drivers). Select Use default audio device when the system has only a single audio card installed.

**Enable 24-bit output for double and single precision input signals**

To select the output 24-bit data when inputs are double- or single-precision. Otherwise, the block outputs 16-bit data for double- and single-precision inputs.

### 3.4.3 Gain

**Purpose:** Multiply the input by a constant

**Symbol:**


**Library:** Math operation

**Description:**

The Gain block multiplies the input by a constant value (gain). The input and the gain can each be a scalar, vector, or matrix.