VOICE MODIFICATION USING DIGITAL TECHNOLOGY

By

Nik Najiha Wahida Bt. Nik Mohd Naaim

FINAL YEAR PROJECT DISSERTATION

Submitted to the Electrical & Electronics Engineering Programme
in Partial Fulfillment of the Requirements
for the Degree
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Universiti Teknologi Petronas
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

VOICE MODIFICATION USING DIGITAL TECHNOLOGY

by

NIK NAJIHA WAHIDA BT. NIK MOHD NAAIM

A project dissertation submitted to the

Electrical & Electronics Engineering Programme

Universiti Teknologi PETRONAS

in partial fulfillment of the requirement for the

Bachelor of Engineering (Hons)

(Electrical & Electronics Engineering)

Approved:

_____

Professor P.A. Venkatachalam

Project Supervisor

Prof. Dr. P.A. Venkatachalam
Professor
Electrical & Electronic Engineering
Academic Block No. 22
Universiti Teknologi PETRONAS
Bandar Seri Iskandar
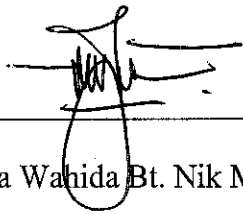31750 Tronoh, Perak Darul Ridzuan, MALAYSIA.

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

April, 2004

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

Nik Najiha Wahida Bt. Nik Mohd Naaim

ABSTRACT

This report describes the voice modification using digital technology using MATLAB software and an additional hardware prototype designed. The voice modification objective that a source voice signal is mapped into another target voice signal. Other objective of this voice modification is to create a new voice signal from a given source signal. These modifications are carried by altering the voice waveform features.

By enabling the modification, it is expected that user is able to mimic other person's voice. In addition, it can be used as a reference and guidance for voice conversion.

For the modification algorithm, an optimization technique for the coding is applied to suit the objective of the project [7].

The project's major activities are to develop programs using MATLAB, that enable speech signal recording, analysis, synthesis, modification, and conversion. The recording is just a simple procedure, where it can be recorded using any computer, with a microphone. Analysis of speech signal is an essential step where the speech waveform features are calculated. Here, we model a vocal tract, which resemble as a filter for the excitation signal input. This filter is designed based on source speech waveform features. The speech waveform parameters used in this analysis are excitation source model, vocal tract model and control model which consist of gain and pitch parameter.

Synthesis procedure is a step to produce a synthetic voice. This synthetic voice is just the display of speech signal designed in the analysis part based on the source signal.

The modification and conversion parts of the software are performed for the voice modification. The modification part enables the user to change the speech waveform parameters calculated and thus creates a new voice. The conversion part is actually performs a mapping of source speech signal to target speech signal.

In this work, hardware also is designed as an additional part to demonstrate the application. The prototype designed performs the conversion process by modulating the frequency of input signal.

Tests were carried on both software and hardware. The software conversion design can convert any source voice signal to target voice signal. But the output will contains some noises.

The hardware model can also modify any input voice signal to another form [14]. But modification is very much limited to seven types of output voices. Here also, noises are noticed.

Since the source codes of the voice conversion software are huge, it is given in Volume 2, under title Source Code of Voice Modification Software.

ACKNOWLEDGEMENTS

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

This chapter is to gives reader an overview of modification of voice or voice synthesis. It generally can be achieved using digital technology. Problems statement and objectives are described in detail in later section.

1.1 Background of Work

It is difficult to advance the field of speech analysis and synthesis without an understanding of its characteristics, features and properties of speech data. Here, briefly described the theory behind to facilitate the understanding.

Speech is generated when air is pushed from the lungs is passing through the vocal cord and along the vocal tract. The basic vibrations occur at the vocal cords, but the sound is altered by the disposition of the vocal tract—e.g. by the position of the tongue or by the shape of the mouth. The vocal tract can be considered as a filter. As the shape of the vocal tract changes relatively slow, the transfer function of the filter need to be change relatively infrequently (e.g. every 20ms or so). If the vocal tract can be modeled as a filter, then the vibrations at the vocal cords correspond to the excitation signal applied to the filter.

Speech can be classified into 3 main types [16]:

- Voiced Sound

  This is produced when the vocal cords vibrate, open and close, thus interrupting the flow of air from the lungs to the vocal tract and producing quasi-periodic pulses of air as excitation occurs. The rate of the opening and closing gives the pitch of the sound. This pitch can be adjusted by varying the shape and tension in

vocal cords and the pressure of the air behind them. Voiced sound show a high degree of periodicity at the pitch period, which is typically between 2ms and 20ms

- Unvoiced Sound

This results when the excitation is a noise-like-turbulence produced by forcing air at high velocities through a constriction in the vocal tract.

- Plosive Sound

This is caused, when the complete closure is made in the vocal tract and air pressure is built up behind this closure and released suddenly.

Combination effort of vocal tract and vocal cords though can be made to create a vast array of sounds; the speech signal is relatively predictable over time. This predictability can be used in the design of voice-coding system where if we could design a system at the both end of transmitter and receiver to make reasonably accurate prediction of how the speech signal will change over time, then we could transfer some amount of data only once.

Speech synthesis, analysis and recognition are the fundamental engineering fields for speech investigation, development and application. Applications of these fields include telephone systems, coding, data compression, voice mail, workstations, personal computers and networks. This project will focus more on the speech synthesis since one of its applications is in voice conversion field.

The objective in voice conversion is to synthesize a voice with desired characteristics, for example creating a voice that sound like Mickey Mouse. Generally, it can be accomplished by converting the voice of one speaker to sound like that of another speaker by transforming or converting the parameter of one speaker's speech to those of another speaker's speech.

The brief description of the process is in figure 1.1.
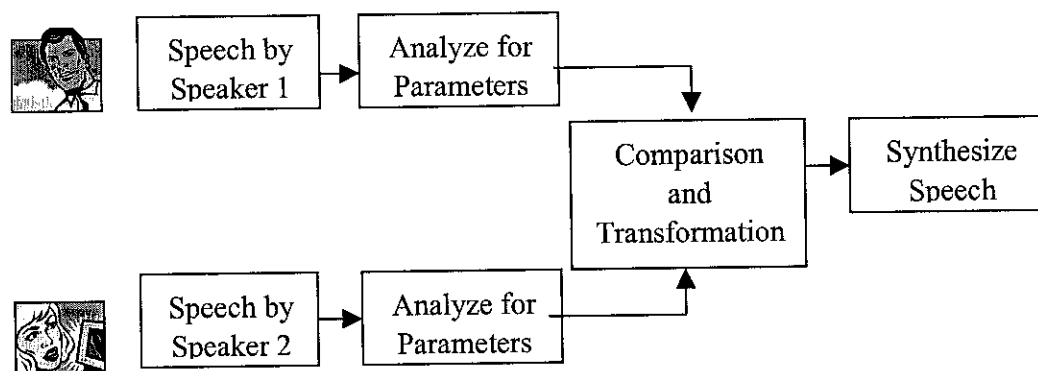
Figure 1-1 Block Diagram of Voice Conversion System

Referring to the definition of voice modifier, it is a device that enables voice to be altered, changed or modified to another totally different voice by transforming and converting the parameter of one speech's signal to sound like that of another speech's signal. The word "modifier" itself stands for a type of tool whether hardware or software that produces altered output.

1.2 Problem Statement

Voice of a person is a very rigid attribute. Compared to image or appearance of a person, voice cannot be changed. For the whole of our life we only have one voice! It will be grateful if we are gifted with a wonderful voice, but if not you cannot do anything. Different people will produce different tones of voice. A person itself can produce thousands of tones. Nevertheless, regardless of how many tones you use, you will not able to hide your voice identity. For normal person, there is no simple way that he or she could mimic other person's voice like our Prime Minister Dr.Mahathir Mohammed or any actor such as John Wayne. This exceptional to person who was gifted the ability to create other voices taking for example Mel Blanc who was known as the man of thousand voices because he created the voice of

Bugs Bunny, Daffey Duck, Tweety Birds, Sylvester the Cat, Porky Pig and many others.

Nevertheless, there is no reference model to the Voice Modifier, which can be used to analyze it overall performance. Modeling here refers to the process of developing and designing at arriving to a suitable technical description of the modifier.

1.3 Objectives and Scope of Study

Objectives and goal of voice modification project are as follows:

- To design a device which able to convert a specific voice such as that of female voice to male voice and vice versa.

- To create a new vocal characteristic much like a composer may create a new musical composition.

Thus, in constraint to this requirement, the scope of the study is focused on the following in highest priority order:

- Software

   To design an interactive software modeling that enables the transformation and conversion of voice signal component other than frequency.

- Hardware

   To design a hardware system, which can convert one type of voice to another. This is just an additional feature of the project.

Therefore, by the end of this project a working modeling application that able to mimic one speaker's sound to one or more different voices is produced.

CHAPTER 2

THEORY & LITERATURE REVIEW

This chapter describes theory behind in implementing voice modification. First, we discuss on the speech production process. Here we understand on how a speech signal is produced. Next the ADC/DAC is described, followed by discussion on analog and digital filters and lastly the speech characterization process.

2.1 Speech Production Process

The speech production involves three processes: source generation, articulation and radiation. The human vocal organs consist of lungs, trachea, glottis, larynx, pharynx, nasal cavity and oral cavity. Combinations of these organs form a connected tube or also known as vocal tract as in Figure 2.1.

(a)   (b)
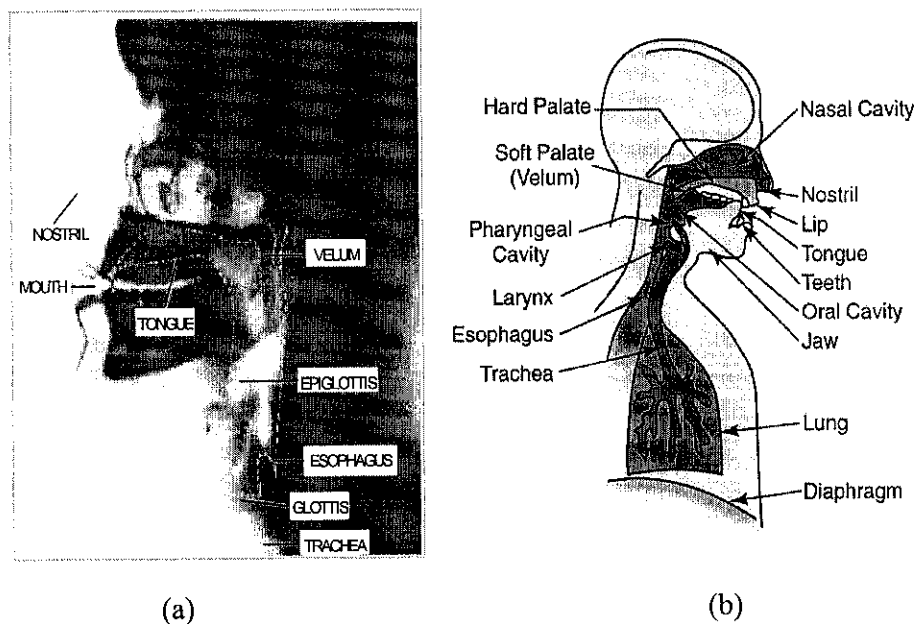
Figure 2-1 (a) Mid-sagittal X-ray of the human vocal apparatus. (b) Diagram of the Human Vocal Mechanism.

Referring to figure 2.1(a), the vocal tract, outlined by the dotted lines, begins at the opening of the vocal cord, or glottis, and ends at the lips. In the average male, the total length of the vocal tract is about 17 cm. The cross-sectional area of the vocal tract, determined by the positions of the tongue, lips, jaw and velum varies from zero (complete closure) to about 20 cm. Nasal cavity consists of nasal tract, which begins at the velum and ends at nostrils. When the velum is lowered, the nasal tract is acoustically coupled to the vocal tract to produce the nasal sounds of speech. Meanwhile, the oral cavity produces oral sound of speech. The oral tract begin at the velum and ends at mouth.

The mechanism of sound produced by the speaker begins when the abdominal muscles forces the diaphragm up (figure 2.1(b)). Air then is pushed up and out from the lungs, with the airflow passing through the trachea and glottis into the larynx. The *glottis*, or the air gap between the left and the right cord, which is usually open during breathing, becomes narrower. Interaction between the airflow and the vocal cord will open and close the gap in accordance and periodically interrupt the airflow through the glottis. The flow also called the glottal source or the source of speech.

The mechanism for vocal vibration produces when the vocal cords are strongly strained and the pressure of the air rising from the lungs is high, the open and close period (which is the vocal cords vibration period) become short and the pitch of the sound source become high. Conversely, low air pressure produces lower pitch sound. This vocal cord period also known as fundamental period and its reciprocal called the fundamental frequency [1].
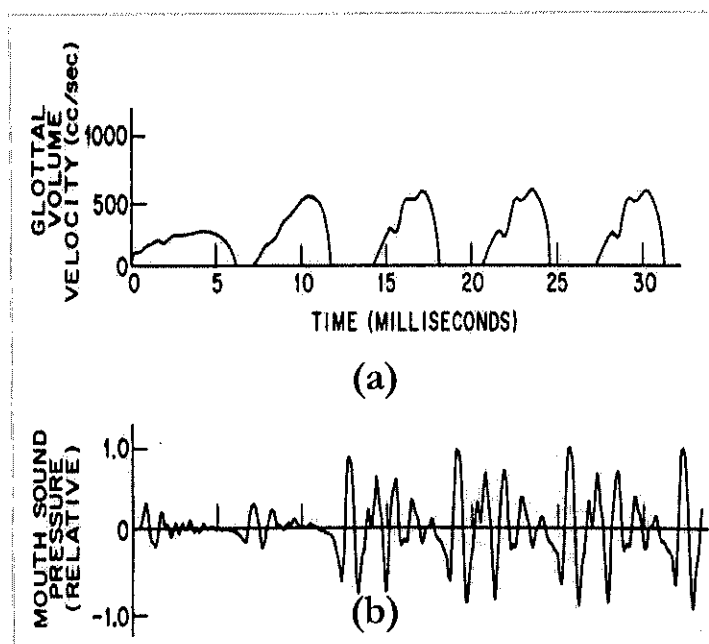
Figure 2-2 (a) Glottal volume velocity and (b) resulting sound pressure at the start of a voiced sound.

As mentioned earlier, speech signal is a quasi-periodic waveform. Figure 2.2(a) shows the glottal waveform also known as volume velocity waveform, which illustrates the quasi-periodic pulse of air. The quasi signal, gradually build-up and taking approximately 15m second (as example) to reach steady state. Figure 2.2(b) is the acoustic or the sound pressure waveform. From this example, we can see more or less the characteristic and behavior of speech signal.

2.2 Analog to Digital Conversion (ADC) and Digital to Analog Conversion (DAC)

Speech signals are *continuous*. Therefore, there is a need for the speech signal to be converted from analog to digital, for 2 reasons:

- Digital techniques facilitate highly sophisticated signal processing which cannot be realized by analog techniques.

- Digital processing is far more reliable and can be accomplished by using a compact circuit or software.

Digitized signals allow digital computers to interact. Digital information is different from its continuous counterpart in two important respects: it is *sampled*, and it is *quantized*. Digital recording converts the analog wave into a stream of numbers and records the numbers instead of the wave. The conversion is done by an ADC. To play back the output signal, the stream of numbers is converted back to an analog wave by a DAC. The analog wave produced by the DAC is amplified and fed to the speakers to produce the sound.

2.2.1 Digitization

Digitization or analog to digital conversion, consist of sampling, quantizing and coding processes. This conversion requires use of a *Codec (Encoder/Decoder)*. Sampling is the process for depicting a continuously varying signal as a periodic sequence of values. Quantization involves approximately representing a waveform value by one of a finite set of values. Coding concerns assigning an actual number to each value, which using binary number representation.

Figure 2.3 shows the complete digitization process.
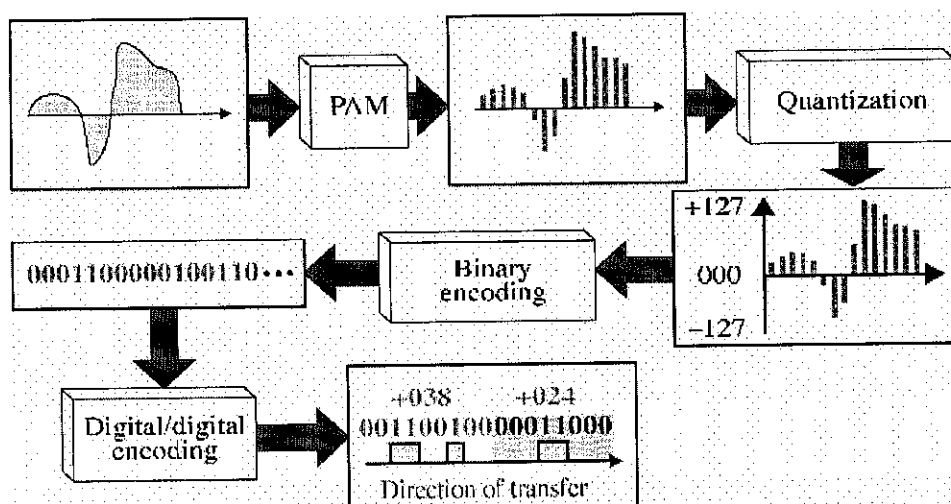
Figure 2-3 Complete Digitization Process

Referring to figure 2.3, the digitization techniques involve of several processes, which is Pulse Amplitude Modulation(PAM), Pulse Code Modulation (PCM) and binary encoding.

2.2.2 Sampling

According to Nyquist Interval Theorem -

"If a signal *f(t)* is sampled at regular intervals of time and at a rate higher than twice the highest significant signal then the samples contain all the information of the original signal".

Conversion to digital format requires the analog data to be sampled in a fixed interval. Referring to Nyquist Interval Theorem, the sampling rate of the signal has to be equal or greater than 2 times of the highest frequency.

Figure 2.4 illustrates the sampling interval of the analog signal.
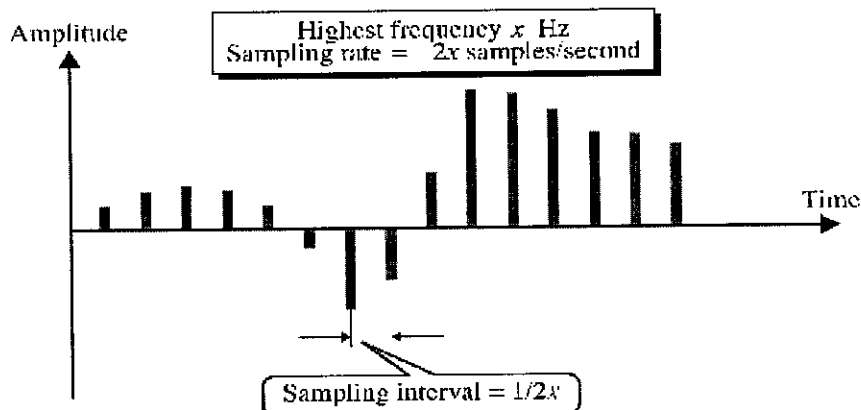
Figure 2-4 Sampling Interval of Analog Signal

Pulses with amplitude proportional with the sample signal value are generated. Sampling the signal with too large sampling interval, T, original signal cannot be reproduced from the sampled sequence; conversely, if T is too small useless sample will be reproduced.

2.2.3 Quantization and Coding

In the quantization stage, the sampled signal amplitudes then are approximated by an n-bit integer. Quantization characteristics depend on both the number of levels and on the quantization step size, δ

Let us consider the example below

Human voice is between 300-3400 Hz. In order to avoid aliasing, instead of taking 3400Hz as the maximum frequency, we take 4000Hz as the maximum.

From the above theorem, the sampling rate would be 2 x 4000Hz = 8000Hz or equal to 8000 sampling/second (1 sample/125μsec).

If we take 1 sample = 4 bits, each sample approximately being quantized in 16 levels. But one thing to remember, because the quantized value are only approximations, it is impossible to recover the original signal exactly.

Then, taking 8 bit per sample allows 256 quantization levels, received signal is comparable with analog transmission. Resulting speed for 8000 samples per second will be 64 K bits/sec.

PAM generates pulses with amplitudes proportional to the sampled value at certain sampling rate. Then the amplitudes of PAM pulses are approximated by an n-bit integer through PCM process. Figure 2.5 and 2.6 illustrate both the PAM and PCM.
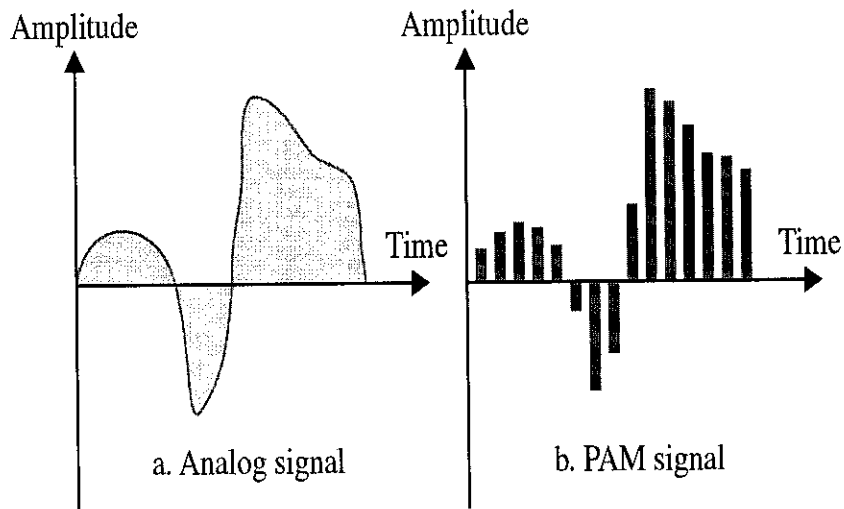
Figure 2-5 Pulse Amplitude Modulation
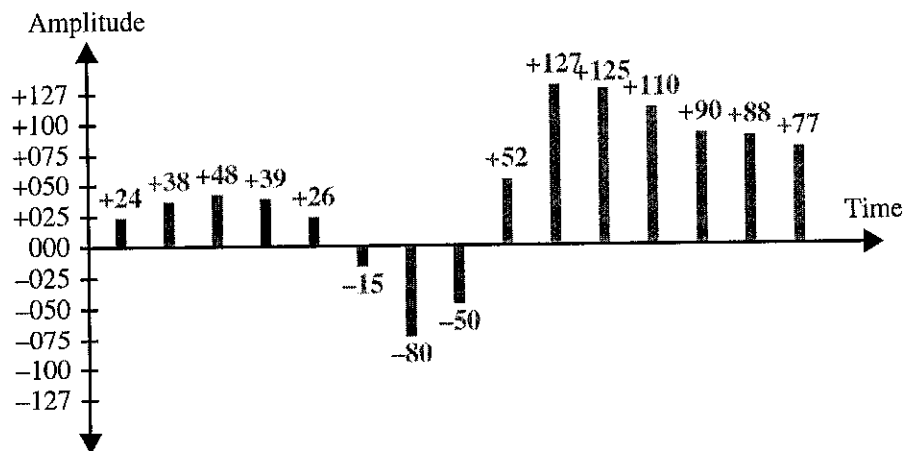
Figure 2-6 Pulse Code Modulation

Then, the quantized amplitude signal encoded into binary format at certain fixed sampling rate. Figure 2.7 is the continuance of figures 2.5 and 2.6, which illustrates the encoding process and direction of transfer.

+024	00011000	−015	10001111	+125	01111101
+038	00100110	−080	11010000	+110	01101110
+048	00110000	−050	10110010	+090	01011010
+039	00100111	+052	00110110	+088	01011000
+026	00011010	+127	01111111	+077	01001101

Sign bit
+ is 0 − is 1

(a)

+024   +038   −048

0 0 0 1 1 0 0 0   0 0 1 0 0 1 1 0   0 0 1 1 0 0 0 0   ● ● ●

Direction of transfer
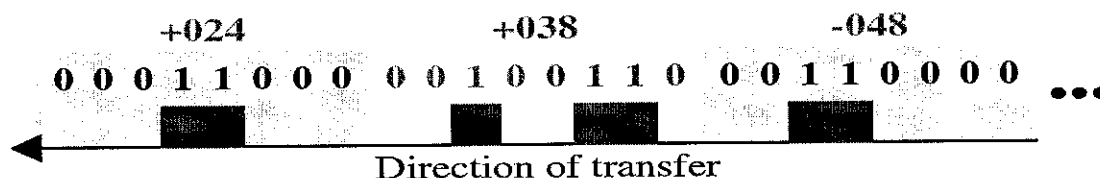
(b)

Figure 2-7 Encoding Process

In figure 2.8, illustrates the digitization process. The conversion is broken into two stages to allow the effects of sampling to be separated from the effect of quantization. The first stage is the sample-and-hold(S/H), where the only information retained is the instantaneous value of the signal when the periodic sampling takes place. In the second stage, the ADC converts the voltage to the nearest integer number. This result in each sample in the digitized signal having an error of up to $+/-\frac{t}{2}$ LSB, as shown in figure 2.8(d). As a result, quantization can usually be modeled as simply adding noise to the signal [12].
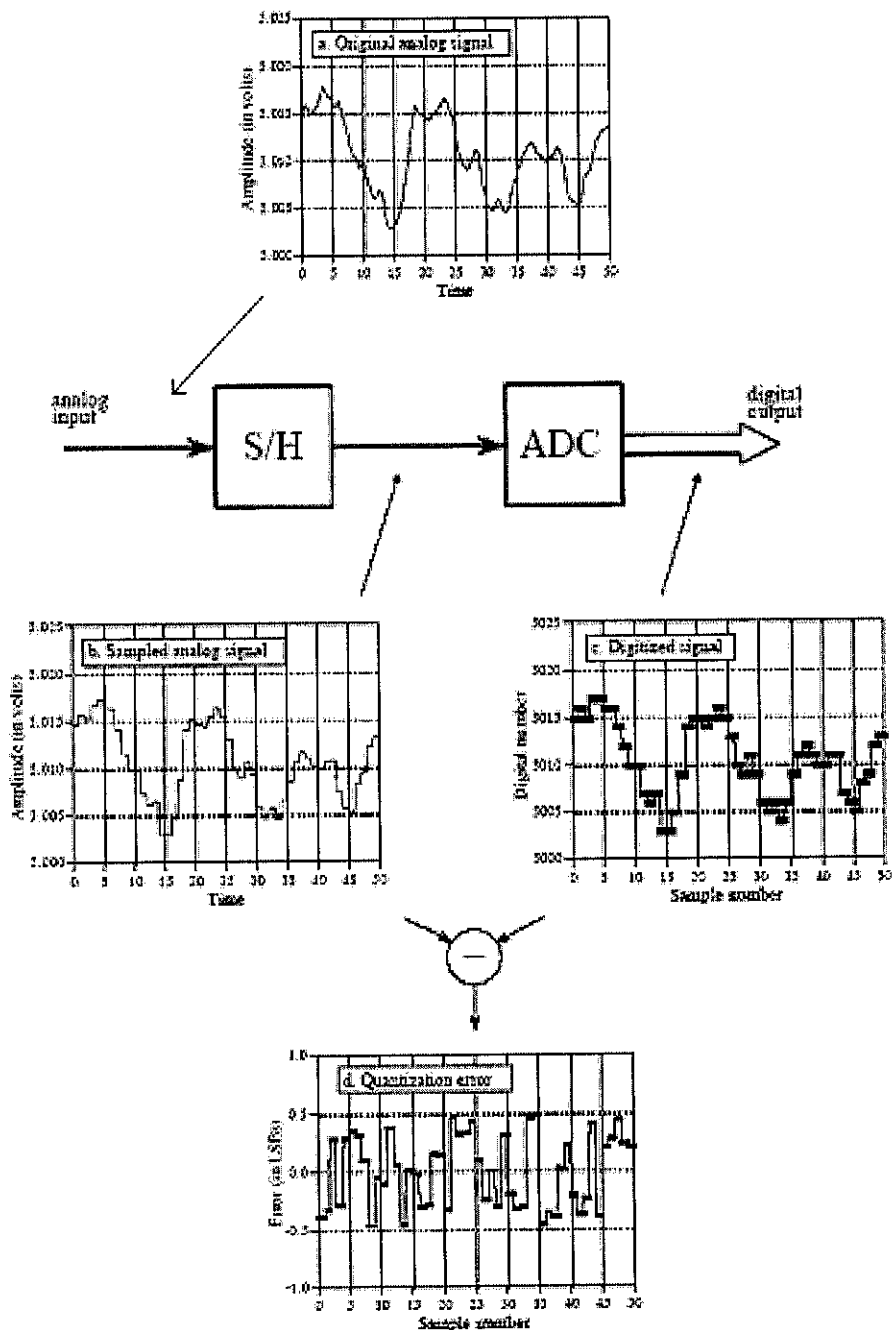
Figure 2-8 Digitization processes.

In theory, the simplest method for digital-to-analog conversion is to pull the samples from memory and convert them into an *impulse train*. This is illustrated in figure 2.9(a), with the corresponding frequency spectrum in figure 2.9(b). As just described, the original analog signal can be perfectly reconstructed by passing this impulse train through a low-pass filter, with the cutoff frequency equal to one-half of the sampling rate. In other words, the original signal and the impulse train have identical frequency spectra below the Nyquist frequency (one-half the sampling rate). At higher frequencies, the impulse train contains a duplication of this information, while the original analog signal contains nothing (assuming aliasing did not occur) [12 and 13].

While this method is mathematically pure, it is difficult to generate the required narrow pulses in electronics. To get around this, nearly all DACs operate by holding the last value until another sample is received. This is called a **zeroth-order hold**, the DAC equivalent of the sample-and-hold used during ADC. A first-order hold is straight lines between the points; a second-order hold uses parabolas, etc. The zeroth-order hold produces the staircase appearance shown in figure 2.9(c) [12].

In figure 2.9(d), the light line shows the frequency spectrum of the impulse train (the "correct" spectrum), while the dark line shows the sinc. The frequency spectrum of the zeroth order hold signal is equal to the product of these two curves.

The analog filter used to convert the zeroth-order hold signal, figure 2.9(c), into the reconstructed signal, figure 2.9(f), needs to do two things:

- remove all frequencies above one-half of the sampling rate,

- boost the frequencies by the reciprocal of the zeroth-order hold's effect, i.e., *1/sinc(x)*.

Figure 2.9(e) shows the ideal frequency response of this analog filter [12].

Time Domain

Frequency Domain

a. Impulse train

b. Spectrum of impulse train

c. Zeroth-order hold

d. Spectrum multiplied by sinc

"correct" spectrum

sinc

e. Ideal reconstruction filter

f. Reconstructed analog signal

g. Reconstructed spectrum
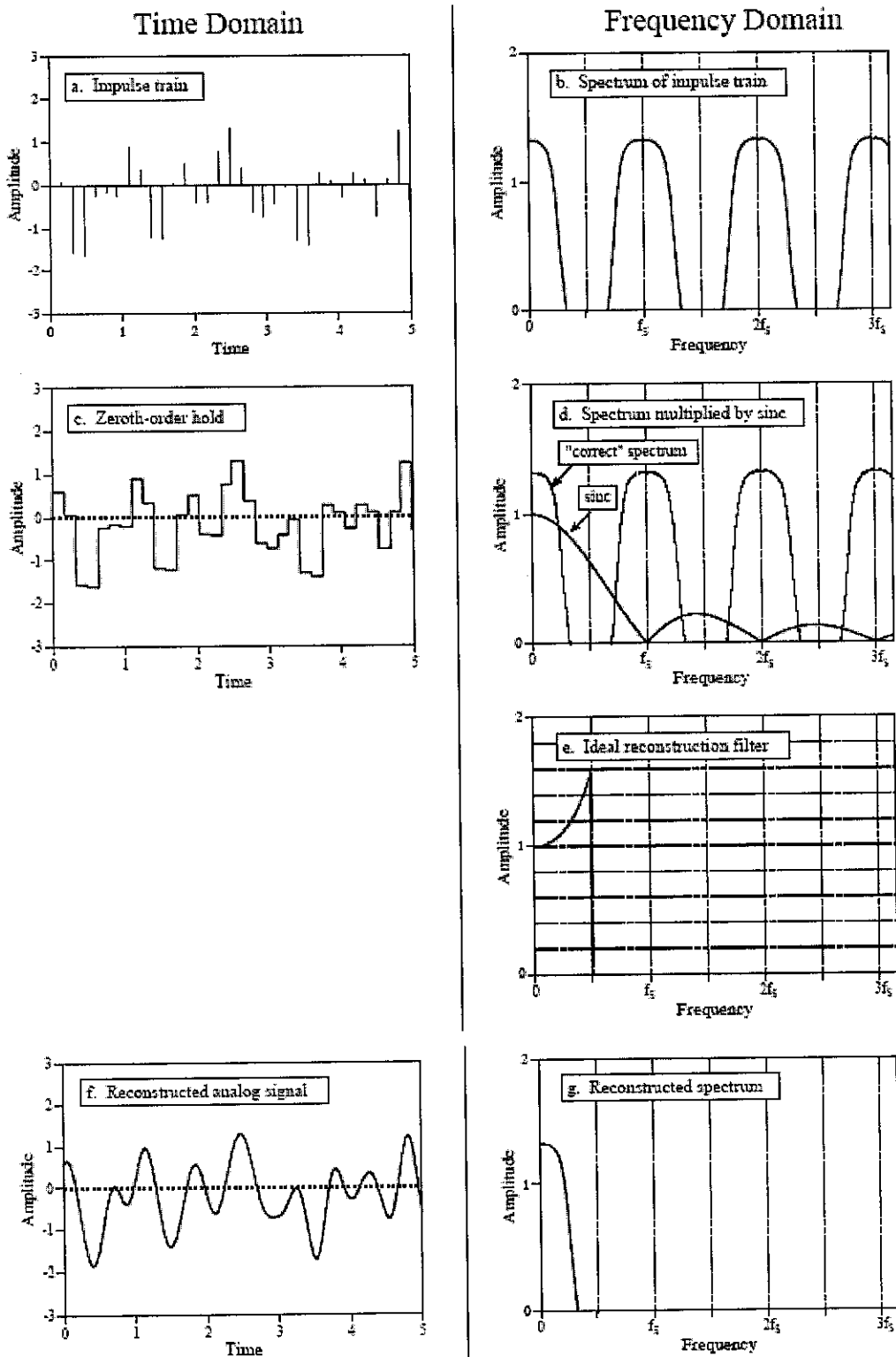
Figure 2-9 DAC process.

2.3 Analog Filtering

Figure 2.10 describes the complete hardware application of signal processing. Analog signal is processed by the *anti-alias filter* before encountering ADC and filtered once again through *reconstruction filter* after the signal is reconstructed by DAC.

Anti-alias filter serve as a device to remove any frequency component above one-half of the sampling rate might be aliasing during the sampling. Reconstruction filter meanwhile also serve to eliminate any frequency above the Nyquist's sampling rate, and correction of zero-order hold.
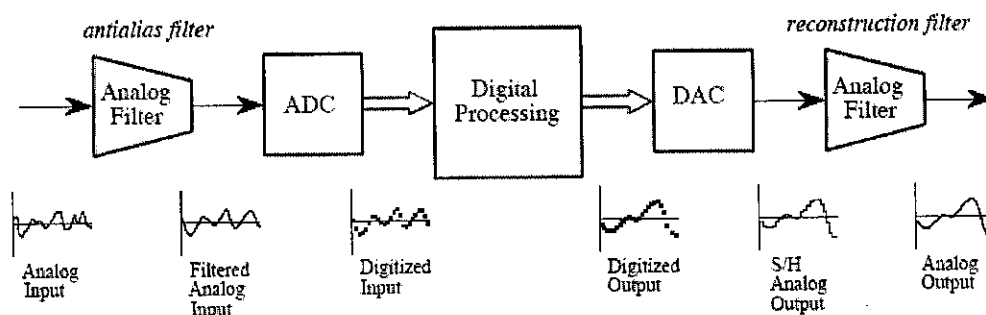
Figure 2-10 Block Diagram of Digital Signal Processing System

Understanding of analog filter is essential in understanding the digital signal characteristics and digital filter designed. There are 3 common type of analog filter used: **Butterworth, Chebyshev and Bessel.** Each of the filters serves different application and thus produces different performance. Nevertheless, these filters are the same in term of number of poles and zeros. Number of poles and zeros describes what the filter does. The more poles in a filter, the more electronics it require and the better it performs. Figure 2.11 shows the common building block for analog filter.

$$R = \frac{k_1}{C f_c}$$

$$R_f = R_1 k_2$$
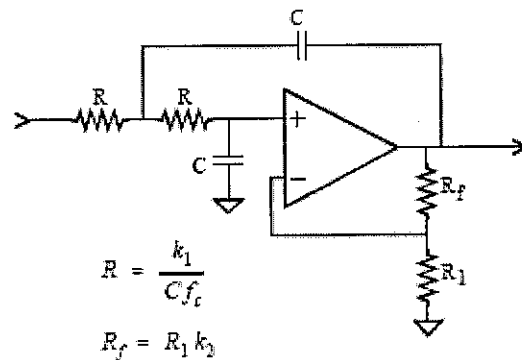
Figure 2-11 Building Block for active filter design.

The circuit shows a 2 poles low pass filter. Higher order filter (more poles) can be achieved by cascading stages. High pass filter can be achieved by swapping the R and C.

2.3.1 Selecting Filter

The performances of each analog filter is characterized by the cut-off frequency sharpness and roll off, passband ripple and step response.

First, the cut-off frequency sharpness is the efficiency of a filter to block certain range of frequency and pass another frequency. Let take a low pass filter for example, it is designed to block all frequencies above cut-off frequency and pass all frequency below. The better the roll-off of the frequencies, the better a filter is. Figure 2.12(a) shows the filter with 1Hz cut-off frequency. In decreasing order, Chebyshev is the best, followed by Butterworth and the lastly is the Bessel [12].

(a)   (b)
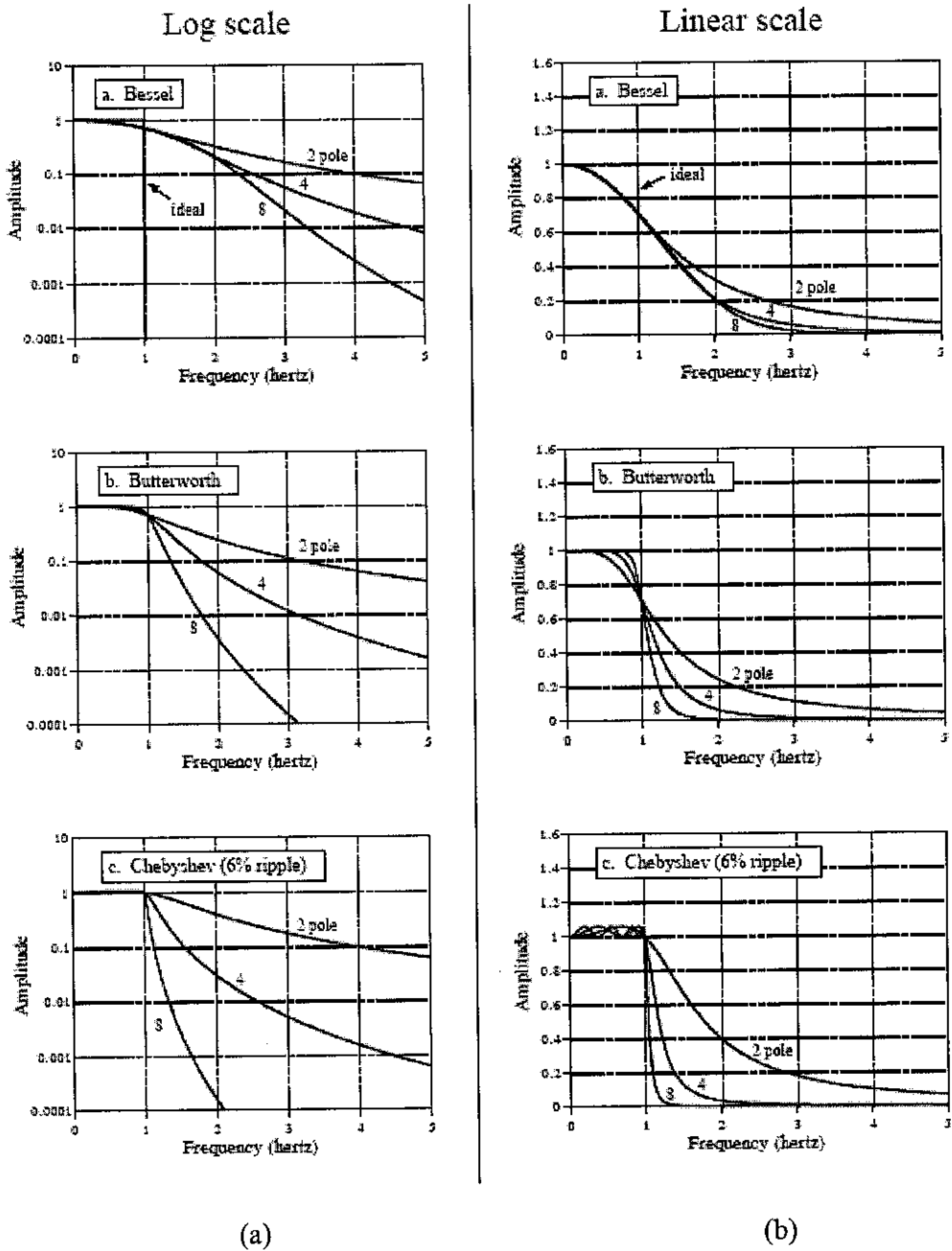
Figure 2-12   (a) Frequency response of three filters on a logarithmetic scale. (b) Frequency response of the three filters on a linear scale.

In term of the pass-band attenuation, Chebyshev filter shows wavy variations in the amplitude of the passed frequencies. In fact, the Chebyshev filter obtains its excellent roll-off by *allowing* this passband ripple. When more passband ripple is allowed in a filter, a faster roll-off can be achieved.

In comparison, the Butterworth filter is optimized to provide the sharpest rolloff possible *without* allowing ripple in the passband. It is commonly called the *maximally flat filter*, and is identical to a Chebyshev designed for zero passband ripple. The Bessel filter has no ripple in the passband, but the rolloff is far worse than the Butterworth

The last parameter to evaluate is the **step response**, how the filter responds when the input rapidly changes from one value to another. The Butterworth and Chebyshev filters **overshoot** and show **ringing** (oscillations that slowly decreasing in amplitude). In comparison, the Bessel filter has neither of these nasty problems.
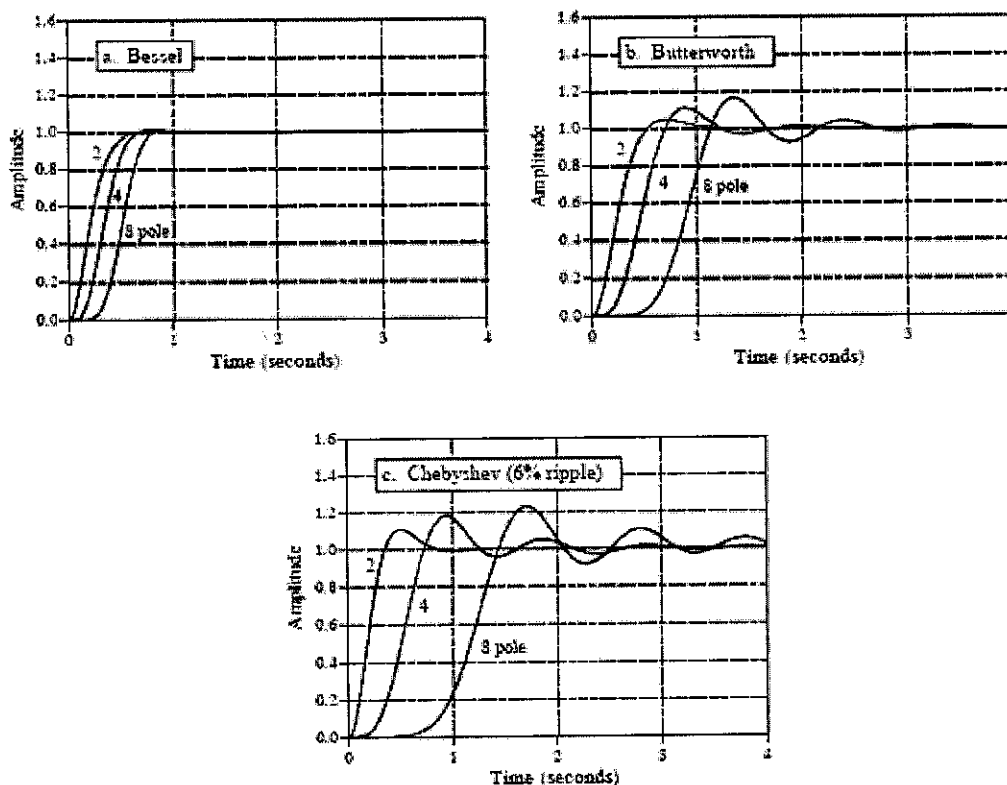
Figure 2-13 Step Response of the 3 filters.

In figure 2.13, the times shown on the horizontal axis correspond to 1Hz cut-off frequency. The Bessel is the optimum filter when overshoot and ringing must minimize [12].

Thus, from the characteristics in above discussion, shows how each filter optimizes a particular parameter at the expense of everything else. In order to choose the best filter to be used, for audio signal processing, which will be used for this project, frequency domain encoding is an excellent choice. When a person hears speech or music, the perceived sound depends on the frequencies present, and not on the particular *shape* of the waveform. This can be shown by passing an audio signal through a circuit that changes the phase of the various sinusoids, but retains their frequency and amplitude. The resulting signal *looks* completely different on an

oscilloscope, but *sounds* identical. The pertinent information has been left intact, even though the waveform has been significantly altered. Since aliasing misplaces and overlaps frequency components, it directly destroys information encoded in the frequency domain. Consequently, digitization of these signals usually involves an antialias filter with a sharp cutoff, such as a Chebyshev, Elliptic, or Butterworth. The bad step response performance of these filters can be ignored since the encoded information isn't affected by this the distortion.

2.4 Digital Filters

A filter is essentially a system or network that selectively changes the wave-shape, amplitude-frequency and/or phase frequency characteristics of a signal in a desired manner. Common filtering objectives are to improve the quality of a signal (for example, to remove or reduce noise), or to extract information from signals.

Digital filters can achieve *thousands* of times better performance than analog filters. This makes a dramatic difference in how filtering problems are approached. With analog filters, the emphasis is on handling limitations of the electronics, such as the accuracy and stability of the resistors and capacitors.

2.4.1 Type of Digital Filter

There are two ways to implement a digital filter, by convolution or recursion. In convolution method, the input signal is convolve or multiply with the digital filter's impulse response. Each output sample produces is the calculated by weighting samples in the input and adding them together. All linear filters can be made in this manner. Recursion method meanwhile, is the extension of the convolution. It uses the previously calculated values from the output, besides points from input. Recursive filters are also called **Infinite Impulse Response or IIR** filters. In comparison, filters carried out by convolution are called **Finite Impulse Response or FIR** filters. This

two type of filters can be represented by equation 2.1 and 2.2 for FIR and IIR filter respectively.

$$y(n) = \sum_{k=0}^{\infty} h(k)x(n-k) = \sum_{k=0}^{N-1} h(k)x(n-k) \qquad \text{(Eq. 2.1)}$$

Instead, the IIR filter can be express in a recursive form:

$$y(n) = \sum h(k)x(n-k) = \sum_{k=0}^{N} a_k \, x(n-k) - \sum_{k=1}^{M} b_k y(n-k) \qquad \text{(Eq. 2.2)}$$

where a_k and b_k are the coefficient of the filter.

Filter carried out using convolution can have far better performance than filter using recursion, but execute much more slowly.

Every linear filter has an impulse response, a step response and a frequency response. These parameters are described in figure 2.14. Impulse response (Figure 2.14(a)) of a filter is the output of the system when the input is an impulse. Step response (Figure 2.14(b)) then can be found by discrete integration of the impulse response. Frequency response can be found from the impulse response by using Fast Fourier Transform, FFT, and can be displayed either in linear scale (Figure 2.14(c)) or in log scale (Figure 2.14(d)) [12].
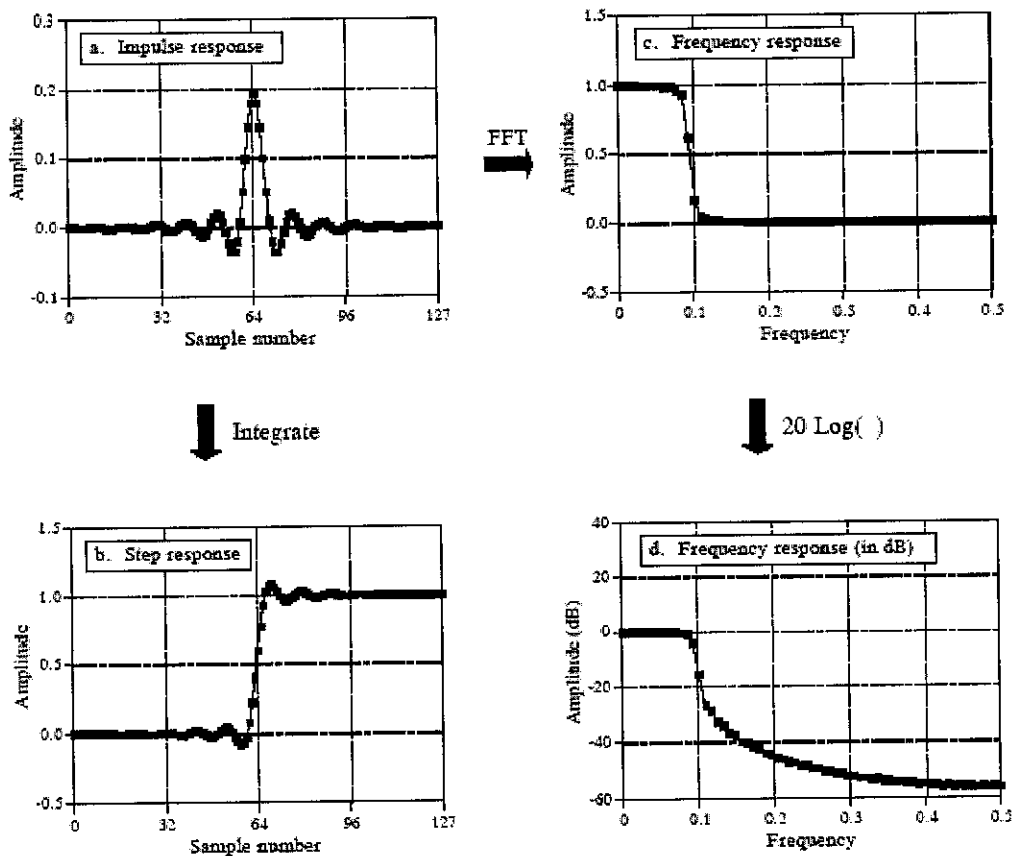
Figure 2-14 Filter parameters

2.4.2 Moving Average Filter

The moving average filter is the most common filter in DSP, mainly because it is the easiest digital filter to understand and use. In spite of its simplicity, the moving average filter is *optimal* for a common task: reducing random noise while retaining a sharp step response. This makes it the premier filter for time domain encoded signals. However, the moving average is the *worst* filter for frequency domain encoded signals, with little ability to separate one band of frequencies from another. Relatives of the moving average filter include the Gaussian, Blackman, and multiple pass moving average. These have slightly better performance in the frequency domain, at the expense of increased computation time.

As the name implies, the moving average filter operates by averaging a number of points from the input signal to produce each point in the output signal. In equation form, this is written:

$$y[i] = \frac{1}{M} \sum_{j=0}^{M-1} x[i+j]$$

(Eq. 2.3)

Not only is the moving average filter very good for many applications, it is *optimal* for a common problem, reducing random white noise while keeping the sharpest step response.

Equation below shows the frequency response of the moving average filter. It is mathematically described by the Fourier transform of the rectangular pulse,

$$H[f] = \frac{\sin(\pi f M)}{M \sin(\pi f)}$$

(Eq. 2.4)

The roll-off is very slow and the stopband attenuation is ghastly. Clearly, the moving average filter cannot separate one band of frequencies from another. Good performance in the time domain results in poor performance in the frequency domain, and vice versa. In short, the moving average is an exceptionally good *smoothing filter* (the action in the time domain), but an exceptionally bad *low-pass filter* (the action in the frequency domain) [12].

2.5   Speech Signal Characterization Parameters

2.5.1   Introduction

There are many attempts made in earlier research in characterizing speech waveform or acoustic signal for the purpose of speech analysis, synthesis, and recognition.

One successful attempt is by Miller at Bell Telephone in 1959, who tried to extract parameters related to laryngeal function by processing the acoustic or speech waveform. He verified that the glottal volume-velocity waveform (figure 2.2) obtained by inverse filtering was a true representation of the air-flow at the vocal folds. Then, further attempts were done such as by Rothenbery,1973: modified the inverse filtering method to obtain an estimate of volume-velocity of air at the mouth.

One way to modify a speech signal is to design a synthetic or acoustic replica of a specified speech signal. This method is well known as "speech synthesis". Based on the system developed by Childers[7], voice conversion is done by characterizing three speech parameters: **vocal tract, voicing source and prosodic features.**

For voice conversion, the parameter from speech waveform (obtained from speech synthesis) of one speaker (source) is mapped to the speech waveform(also obtained from speech synthesis) of another speaker(target).

2.5.2 Speech/Voice Synthesis and Conversion

2.5.2.1 Speech/Voice Synthesis

For the voicing source or known as excitation source, there are two sources, which are for voiced speech and unvoiced speech. For unvoiced sound, a stochastic codebook is used, which codes the residue using a Gaussian noise generator. For voiced sound, two models used for the excitation, LF model and Polynomial model, which specify the shape of glottal waveform. Note that only one source model can be employed for the voiced speech at a time. Figure 2.15 describes the modeling of speech synthesis. Note that pulse train generator or periodic waveform generator is modeled for voiced speech.
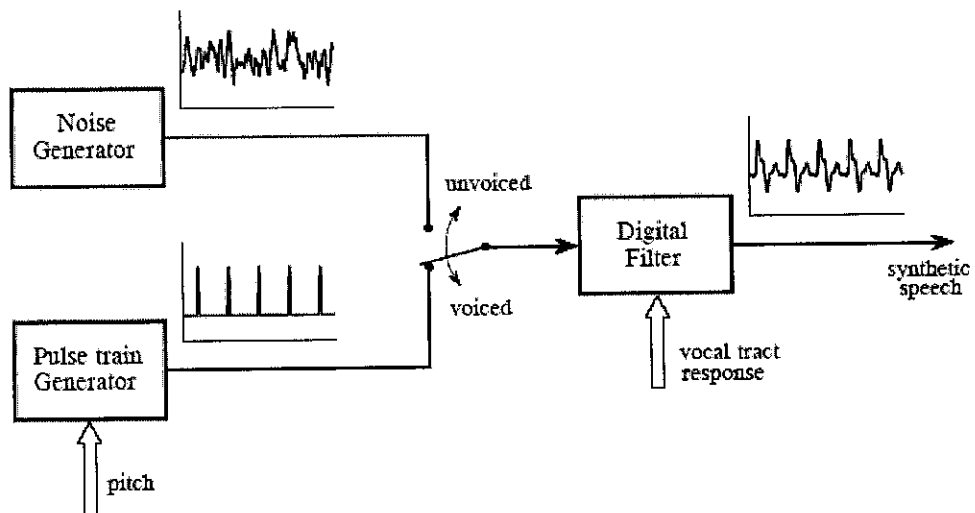
Figure 2-15 Speech Synthesis model.

The speech synthesis model in figure 2.15 describes the human speech model. Over a short period of time, speech can be modeled by three parameters:

(1) A selection of either a periodic or a noise excitation.

(2) The pitch of the periodic excitation.

(3) The coefficient of a recursive linear filter mimicking the vocal tract response.

The pitch (period) parameter $,p_p$ is the control parameter that determines the length of the glottal excitation waveform. For unvoiced speech, the pitch is fixed at 5msec.

The vocal tract response or resonant tract parameter modeled by either using formant parameter or Linear Prediction (LP) coefficient parameter. Throughout this project, LP configuration will be used. More details on the formulae used, refer to Appendix A.

The prosodic features then, are the control parameters, which assist the in mimicking the speaking style of various speakers. This control parameter includes:

- Voiced/unvoiced classification, Vc.

    Used to determine which synthesis scheme (voiced/unvoiced) is adopted for each synthesis frame.

- Gain Parameter ,g.

    Used to control the intensity of synthesized speech.

The excitation source parameter and vocal tract parameter are updated at the beginning of each pitch period, while the prosodic or control parameter controls the rate of updating.

Details of the parameters used can be obtained in Childers[7] and Rabiner[8].

2.5.2.2 Voice Conversion

Voice conversion is a process of transforming the voice of one speaker to sound like the other speaker. There are two method used for mapping the speech signal, which are **translation model** and **affine model.**

(a) Translation Model.

To alter or convert the voice of one speaker to another, the parameters of one speaker are mapped to match the parameters of another speaker. This is accomplished as follows:

A phoneme can be represented by an n-dimensional speech waveform vector. The mean of this phoneme vector over various speakers is denoted as μ. The phoneme of speaker "i" is denoted as vector s^i and given as

$$S^i = \mu + \delta$$

where the vector δ is a bias considered to be the characteristics of the speaker.

Assuming that the speech waveform features are independent and time invariant, then one can convert the speech waveform of one speaker to those of another provided the offset value between the two speakers is known, that is

$$X = Y + B$$

where X and Y are the n-dimensional speech waveform parameter vectors of target and source speaker respectively and B is the offset vector and n is the number of measured speech waveform.

The task of translation model is to estimate B, which is the difference between the two speakers. The equation above can be written as a set of m equation

$$X_1 = y_1 + b_1$$
$$X_2 = y_2 + b_2$$
$$\vdots \qquad \vdots$$
$$X_n = y_n + b_n$$

If m samples are available for the two speakers the estimate is

$$\hat{b}_i = \frac{1}{m}\sum_{k=1}^{m}(x_{ik} - y_{ik}) \quad \text{where } i=1,2.......n$$

Detail of this model can be obtained in Childers[7] and Furui[1].

(b) Affine Model

The assumption made in translation model might not be valid for some situation. More details of the model are as follows:

Relative to the previous model, the linear combination of another speaker features is

$$X = AY + B$$

where X and Y are n-dimensional speech waveform feature vectors for the target and source speaker respectively. A is an n by n matrix and B is the n-dimensional offset.

The process of mapping function between X and Y is known as the training process. Due to variation of speaking rate from one speaker to another, dynamic time-warping (DTW) is used to adjust the parameters of the source to be in accord with those the target along the time axis.

The procedures for training process are described in figure 2.16.
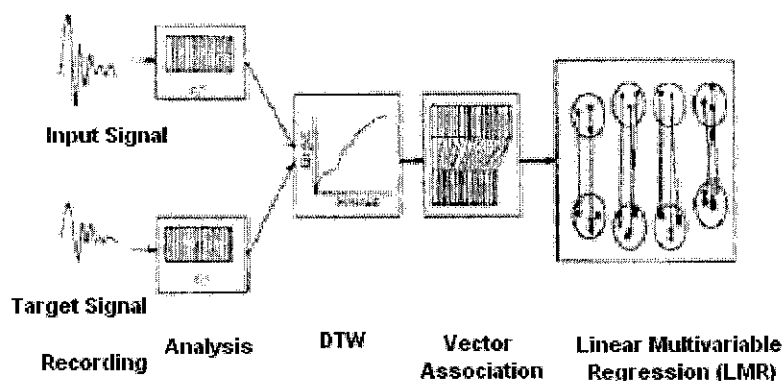
Figure 2-16 The training Process

Target and source speakers must pronounce the same sentences. Then, the speech waveform features are measured, providing a set of framed-based vectors. The source vectors are time aligned with the corresponding target vector by DTW. Linear Multiple Regression (LMR) is used to estimate the coefficient of the mapping function [7].

CHAPTER 3

METHODOLOGY/PROJECT WORK

3.1   Software- Voice Modification Application

The voice conversion system is designed using MATLAB. There are three main steps taken in performing the conversion, *analysis, modification and synthesis* (Figure 3.1).

Analysis procedure involves of determining voice classification, pitch period or pitch contour and glottal closure instant, GCIs of a source signal. This step is very essential in voice signal conversion for designing the voice synthesis model.

Modification procedure enables the voice signal parameters determined and calculated to be altered or changed. Voice conversion process from female to male is also enabled in the conversion procedure. The conversion is done using DTW process.

Synthesis procedure aims to design a synthetic voice signal. The synthetic voice can either develop a voice signal from the parameters determined and calculated in analysis part, or modified parameters in modification part.
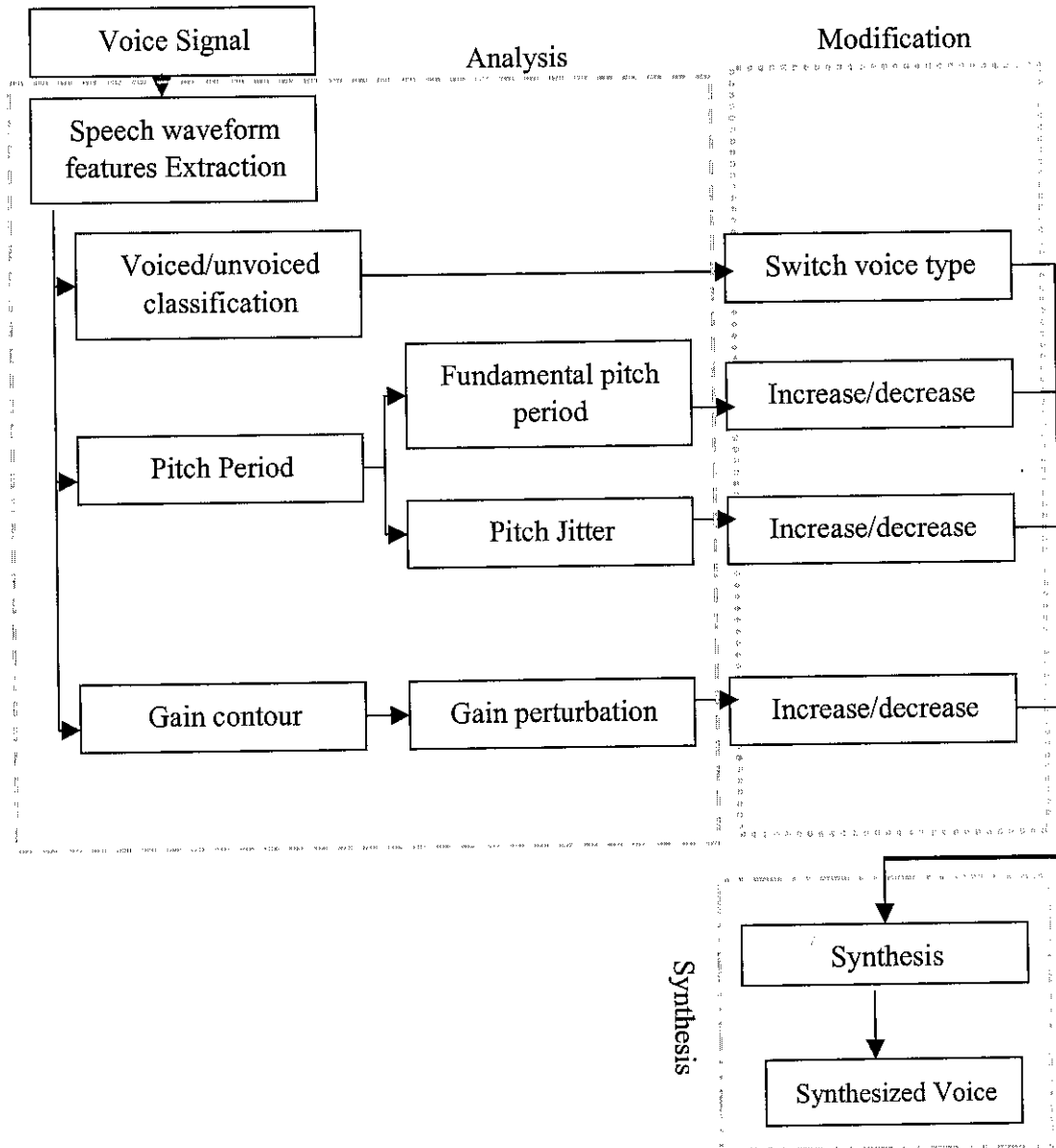
Figure 3-1 Voice Modification System

For the software designed, the calculation and each process taken for voice conversion are sourced from D.G. Childers[7]. Here, some of the coding and system have been optimized for better understanding and suited with the project.

Three easy steps are designed to perform the voice modification. First, analysis procedure has to be performed to calculate the parameters for voice synthesis. Second, the procedure can be continued by either selecting synthesis procedure or modification procedure. By selecting synthesis as the second step, original and synthesized voice will be displayed. User is able to play the sound and view the waveform and spectrogram of both the signals. Selecting modification procedure enable user to perform modification to the synthesized signal. One way is where user can manually decrease or increase the parameters values, then display the modification result by selecting the synthesis step. The other method to perform modification is by selecting voice conversion procedure. This step will perform DTW by mapping the source (input) signal to the target signal. The result of the conversion then can be displayed by selecting synthesis step. The details of how voice modification software is running are in figure 3.2.

The voice signal to be inserted (figure 3.3) is designed to be only in ASCII format, which is in ".dat" extension. Therefore, any recorded voice need to be converted to ASCII format before using the software. Error message will be displayed indicating to the user in occurrences of no signal file or other type of format, which the program cannot support. The signal amplitude then is normalized to maximum of 10000, as the standard limit. The input voice signal waveform then is plotted, to display the signal characteristics and behaviors. Signal 'chunking', LP coefficient calculation, voice type classification, glottal closure instants calculation and pitch determination are explained in later subsection. Details of the procedures taken are in Childers[7], Furui[1] and Rabiner and Schafer[8].
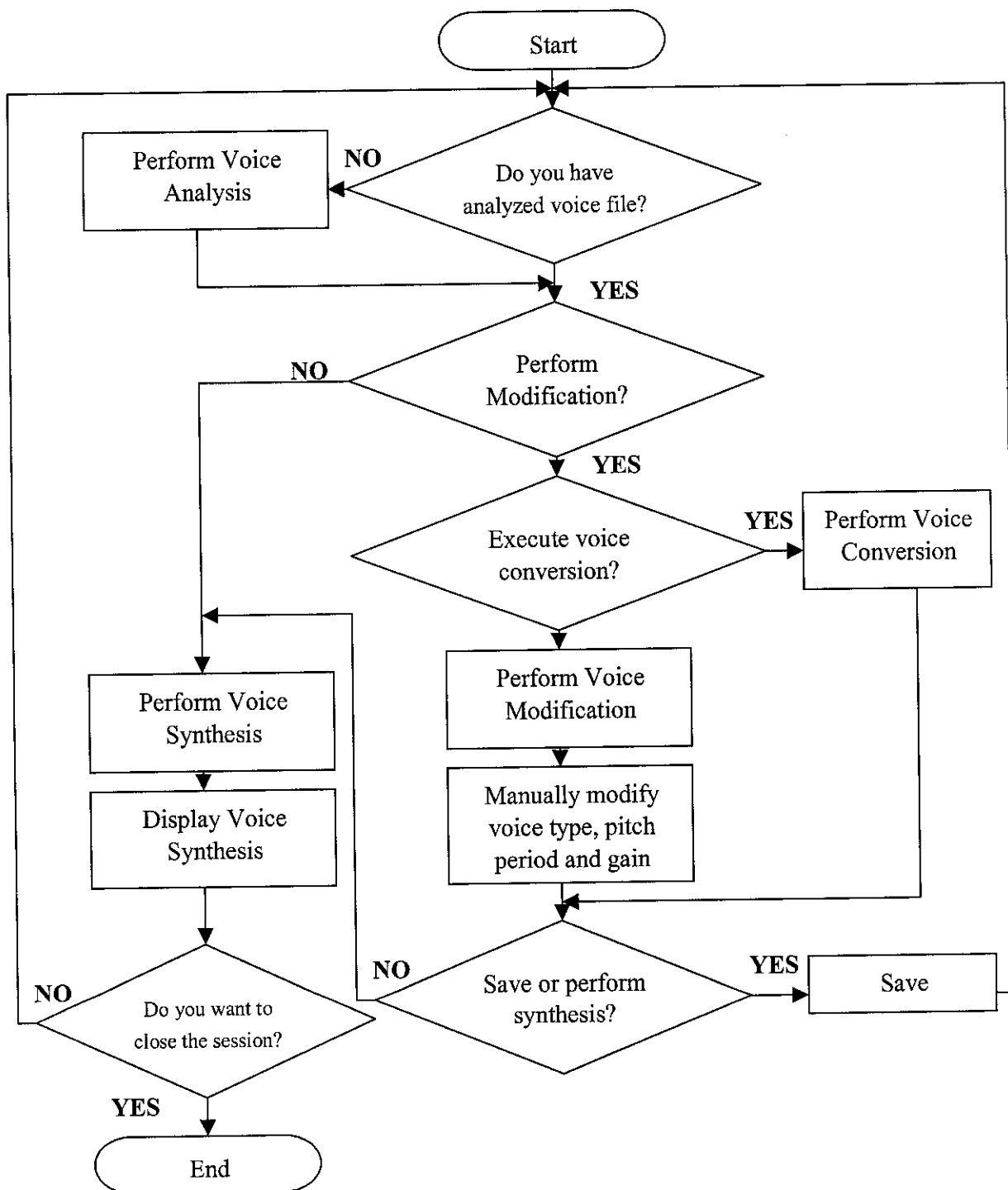
Figure 3-2 Step by Step Procedure of Voice Modification Software

3.1.1 Voice Analysis Procedures

The analysis is based on LP configuration. Procedures taken in performing the analysis are described in figure 3.3.

```
        ┌─────────────────────────────┐
        │      Load Voice Signal        │
        └─────────────────────────────┘
                      │
                      ▼
              ╱─────────────╲              NO
             ╱ Is there a signal? ╲──────────────▶  ┌──────────────────────┐
             ╲                   ╱                   │ Output error message │
              ╲─────────────╱                        └──────────────────────┘
                      │ YES
                      ▼
        ┌─────────────────────────────┐
        │  Normalized speech to maximum │
        │      amplitude: 10000         │
        └─────────────────────────────┘
                      │
                      ▼
        ┌─────────────────────────────┐
        │     Display voice signal      │
        └─────────────────────────────┘
                      │
                      ▼
        ┌─────────────────────────────┐
        │  Perform signal Segmentation  │
        └─────────────────────────────┘
                      │
                      ▼
        ┌─────────────────────────────┐
        │     Calculate LP coefficient  │
        └─────────────────────────────┘
                      │
                      ▼
        ┌─────────────────────────────┐
        │      Classify Voice types     │
        └─────────────────────────────┘
                      │
                      ▼
        ┌─────────────────────────────┐
        │     Calculate Pitch Period    │
        └─────────────────────────────┘
                      │
                      ▼
        ┌─────────────────────────────┐
        │ Calculate Glottal Closure Instants │
        └─────────────────────────────┘
                      │
                      ▼
        ┌─────────────────────────────┐
        │      Save Analyzed signal     │
        └─────────────────────────────┘
```

Figure 3-3 Analysis Procedure Flowchart.

3.1.1.1 Perform Signal Segmentation

Voice signal input consists of streams of data. Thus, it is impossible to perform calculation on the data, if it is the execution will be much slower. This will decrease the performance of the software. In order to eliminate the problem, the signal is segmented into few frames, which user determines the length. This 'observation windows' create fewer points to be calculated. In addition, the voiced and unvoiced sound signal classification will easier to be determined. The detail of each step taken in performing segmentation is as in figure 3.4.

3.1.1.2 LP Coefficient Calculation

After the input signal is segmented, LP coefficients, which consist of first order reflection coefficient, signal energy, prediction error and LP coefficient (using auto-regression concept) is calculated. The results of the calculation then used for the classification of voice signal, pitch period and GCIs. Flowchart of the process is as in Figure 3.5 and 3.6.

3.1.1.3 Classify Voice Types

The voice types of voice signal is classify into voiced and unvoiced. More details on classification can be referred to section 1.1: Background of Work. In order to 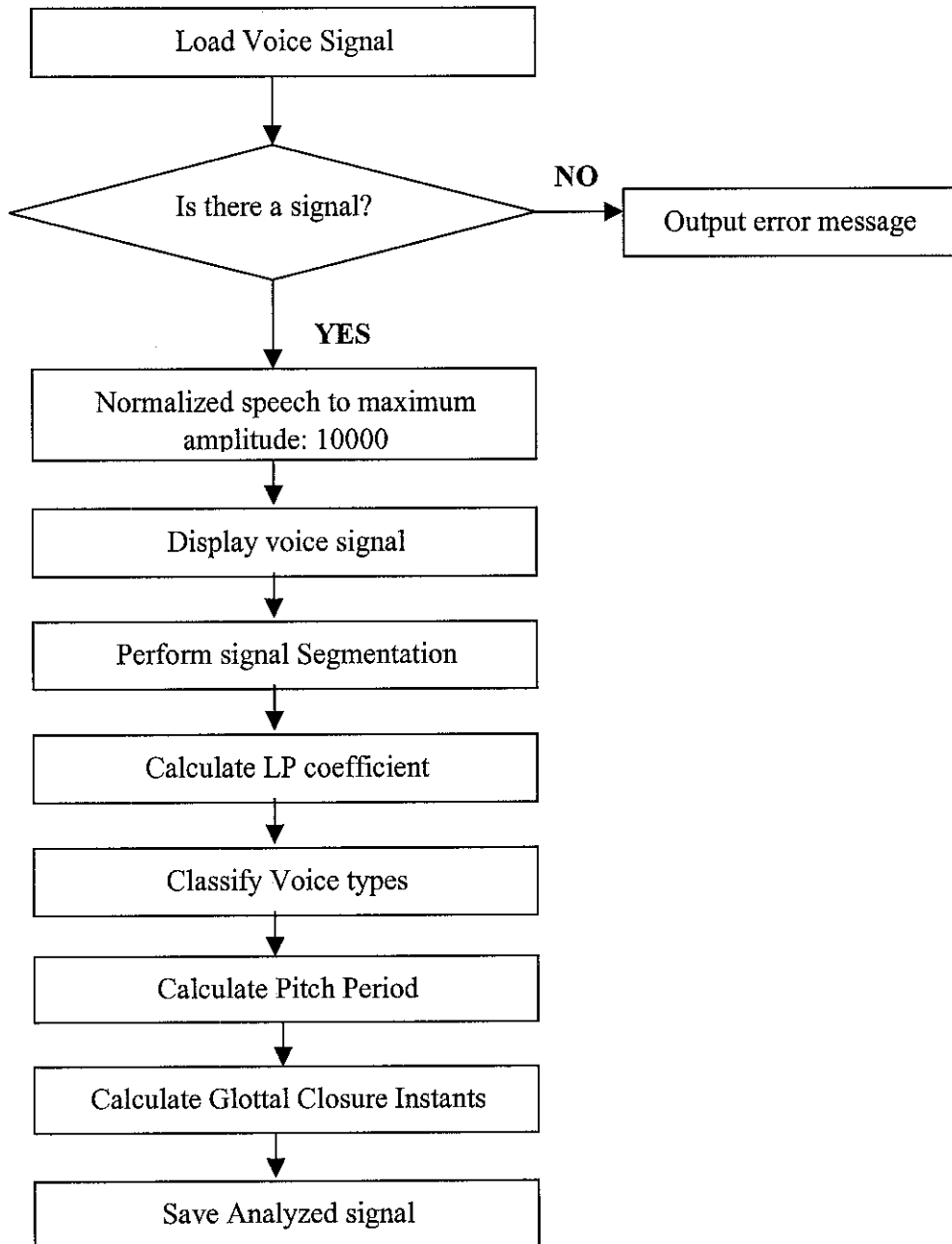model a human speech model (Figure 2.15: Human Speech Synthesis Model), the classification of voice type functions to select signal to be generated; either periodic signal or a noise excitation. The signals generated then feed into a digital filter, which is vocal tract response. The details process of the voice signal classification is as in figure 3.7.

3.1.1.4 Pitch Period Calculation

The pitch period is determined by using the result obtained in LP analysis. The pitch period is then fed into the periodic impulse generator. The pitch estimation of the voice signal is used to as the control parameter that determines the length of glottal excitation waveform. The details for the estimation are as in figure 3.8.

3.1.1.5 Calculate GCIs

This method used the results from both voiced/unvoiced classification procedure and the error prediction waveform, e(n). The GCIs estimations are obtained using the glottal excitation waveform, that is, the LP residual signal and peak-picking algorithm. The procedures involved in performing the calculation are as in figure 3.9.

Figure 3-4 Voice Signal Segmentation Flowchart

```
        ┌─────────────────┐
        │      Start       │
        └─────────────────┘
                 │
                 ▼
   ┌───────────────────────────┐
   │    Insert voice signal     │
   └───────────────────────────┘
                 │
                 ▼
   ┌───────────────────────────┐
   │ Calculate flexible LP      │
   │ coefficient for            │
   │ first frame                │
   └───────────────────────────┘
                 │
                 ▼
   ┌───────────────────────────┐
   │ Calculate flexible LP      │
   │ coefficient for            │
   │ second frame till the      │
   │ second last                │
   └───────────────────────────┘
                 │
                 ▼
   ┌───────────────────────────┐
   │ Calculate flexible LP      │
   │ coefficient for            │
   │ last frame                 │
   └───────────────────────────┘
                 │
                 ▼
   ┌───────────────────────────┐
   │ Combine the LP flexible    │
   │ coefficient                │
   └───────────────────────────┘
                 │
                 ▼
        ┌─────────────────┐
        │      End         │
        └─────────────────┘
```
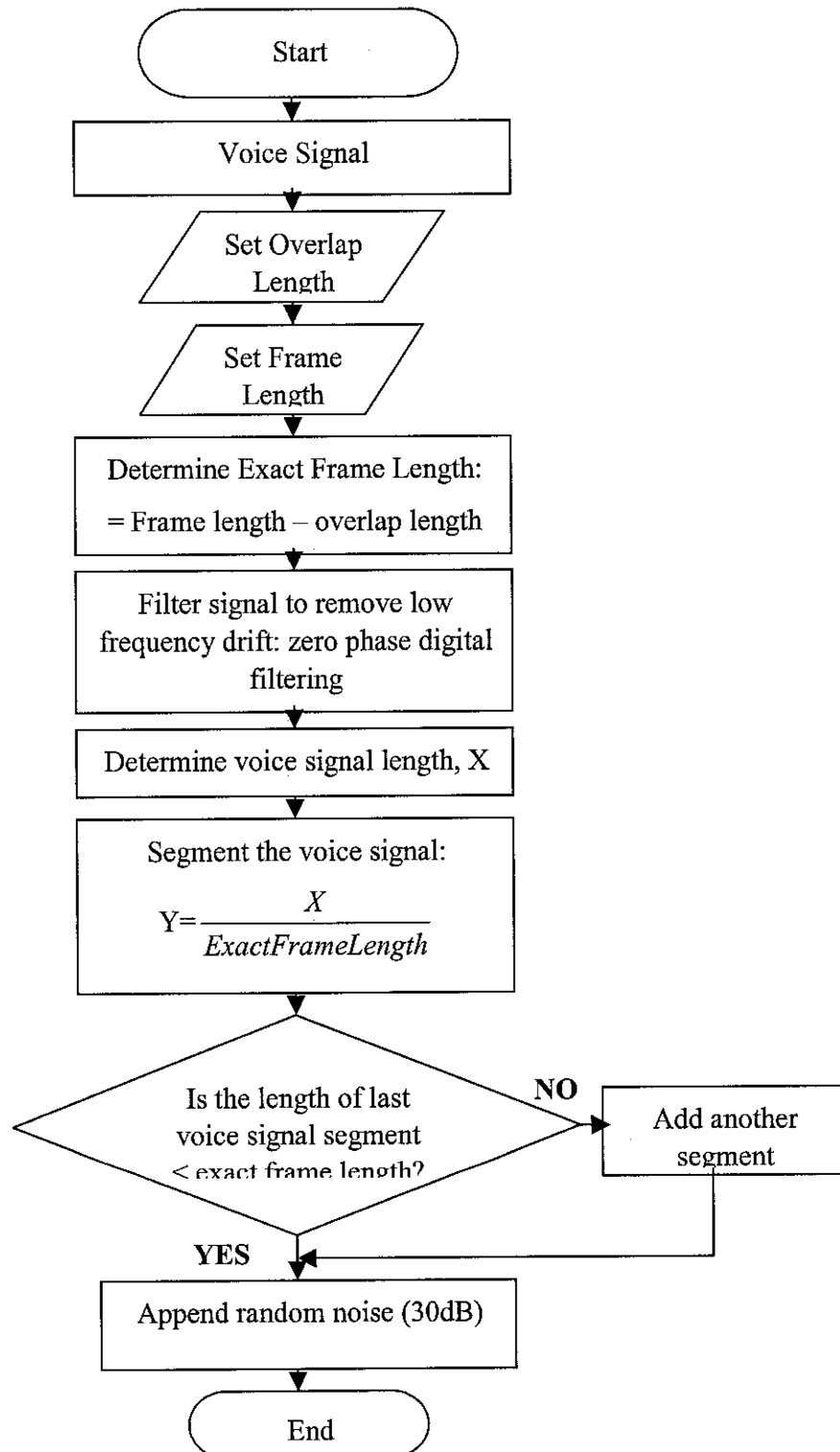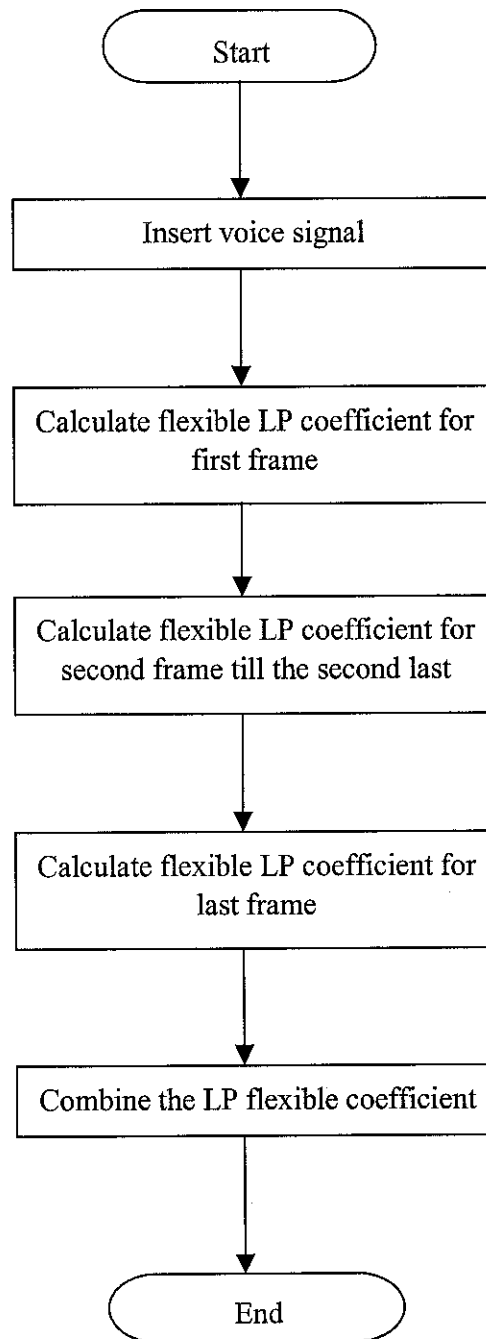
Figure 3-5 LP Calculation Flowchart
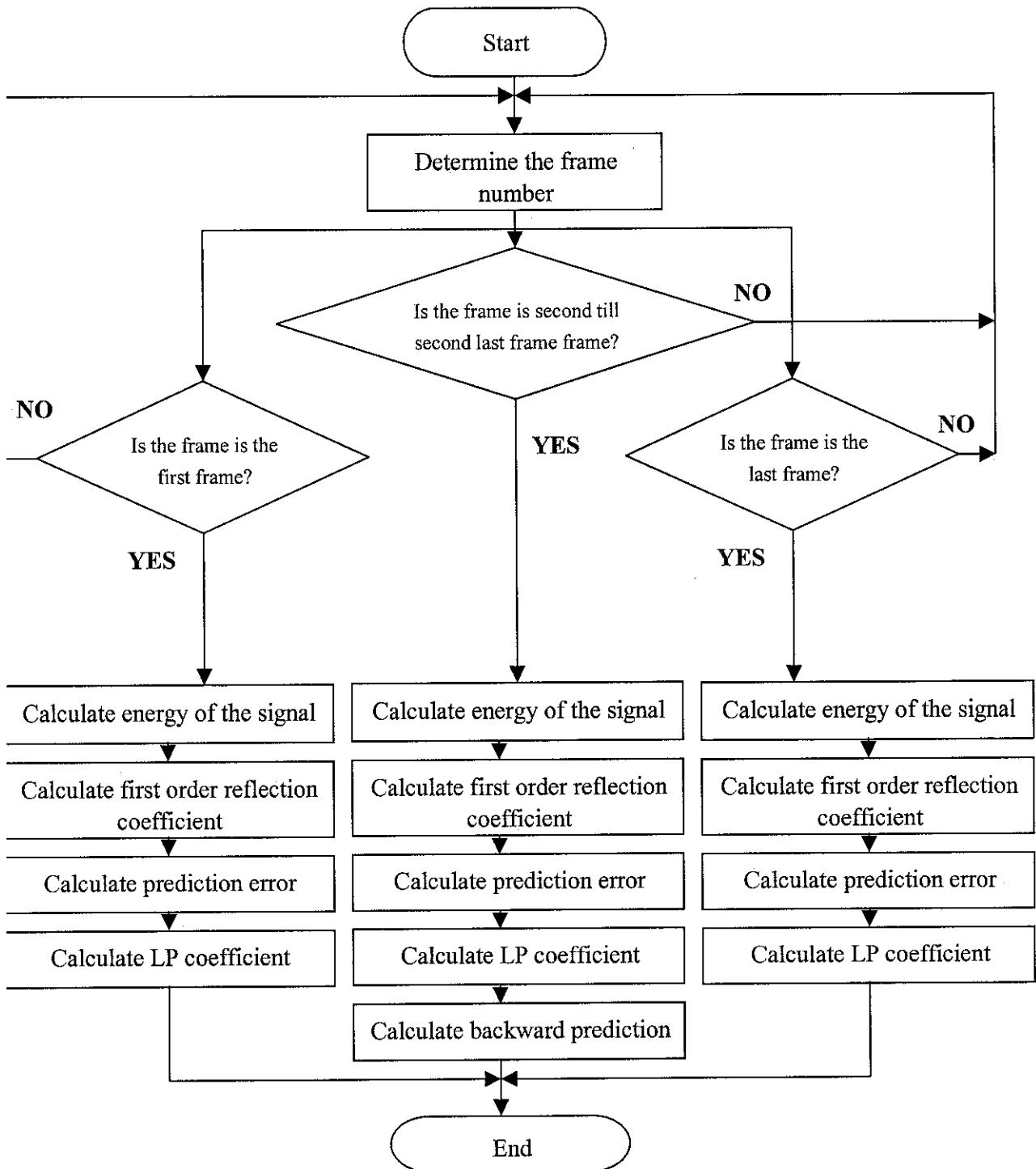
Figure 3-6 Flexible LP Coefficient Calculation Flowchart

Start

Get the energy prediction error of Y_n (the frame/segment currently calculate)

Get the first reflection coefficient of Y_n

Is the current frame/segment reflection coefficient > 0.2 && energy prediction error > 2×10^7 ?

YES → Classify frame as voiced

NO

Is the current frame/segment reflection coefficient > 0.3 && energy prediction error > 1×10^7 && previous frame is also voiced?

YES → Classify frame as voiced

NO

Classify frame as unvoiced

End

Figure 3-7 Voice type Classification Flowchart

Start

Low pass filter segment of prediction error waveform. Filtered waveform is

$$e_{LP}(n)$$

Calculate Ceptrum sequence:

$$Ce(n) = IFFT(|\, FFTe_{LP}(n)\,|)$$

Find the maximum amplitude from the cepstrum sequence, Ce(n),

in range $25 \le n \le N$

Denote sequent which maximum amplitude occurred, 'n' as M

Find the maximum amplitude from the cepstrum sequence, Ce(n),

in range $26 \le n \le M - 12$

Denote sequent which maximum amplitude occurred, 'n' as K

If Ce(K) > 0.7*Ce(M)?

YES

K is the estimated pitch

NO

M is the estimated pitch

End

where $1 \le n \le N$ and N is the frame length

IFFT is Inverse Fast Fourier Transform and FFT is Fast Fourier Transform

Figure 3-8 Pitch Period Estimation Flowchart

```
                        ┌─────────────┐
                        │    Start    │
                        └──────┬──────┘
                               ▼
        ┌──────────────────────────────────────┐
        │ In each frame/segment, find the most  │
        │ negative peak in e_LP(n) waveform.    │
        └──────────────────┬───────────────────┘
                           ▼
        ┌──────────────────────────────────────┐
        │     Build template around the peak.   │
        │  Length of the template=46 samples    │
        │  Length before the peak=15 samples    │
        │     Length after the peak=30          │
        └──────────────────┬───────────────────┘
                           ▼
        ┌──────────────────────────────────────┐
        │       Correlate the template with     │
        │          e_LP(n) waveform.            │
        └──────────────────┬───────────────────┘
                           ▼
        ┌──────────────────────────────────────┐
        │ Find the positive peaks from the result│
        │          of previous step             │
        └──────────────────┬───────────────────┘
                           ▼
        ┌──────────────────────────────────────┐
        │      Set as initial estimate for GCIs │
        └──────────────────┬───────────────────┘
                           ▼
        ┌──────────────────────────────────────┐
        │       Correlate the template with     │
        │          e_LP(n) waveform.            │
        └──────────────────┬───────────────────┘
                           ▼
                      ◇─────────◇
            Is two GCIs located within   YES   ┌──────────────────────┐
            25 samples of on another?    ────► │  Adjust the position.│
                      ◇─────────◇              └──────────┬───────────┘
                         │ NO                             │
                         ▼                                │
        ┌──────────────────────────────────────┐         │
        │              Set as GCIs              │◄────────┘
        └──────────────────┬───────────────────┘
                           ▼
                    ┌─────────────┐
                    │     End     │
                    └─────────────┘
```

Figure 3-9 GCIs Calculation Flowchart

3.1.2 Voice Modification Procedures

The modification procedure can be performed in two ways; Modification and Conversion. The detail is as in figure 3.10.
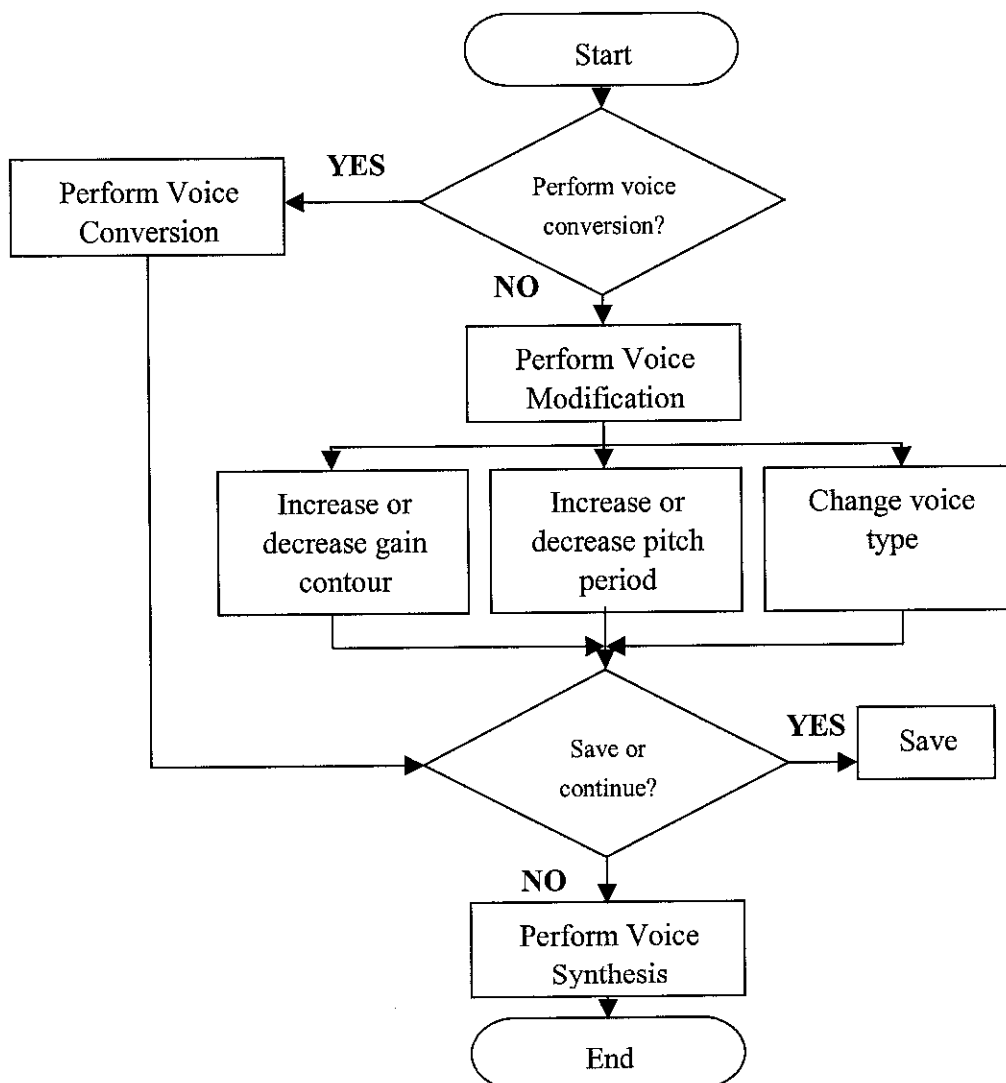
Figure 3-10 Flowchart for Modification Procedure

The modification process involved changing the gain contour, pitch period and voice type. These parameters are randomly changed and the output produces is not predictable.

The gain parameter is the average value of voice signal energy for each pitch period. It functions to control the energy transition. This parameter is related to the intensity (loudness) of voice signal. As the loudness increases, gain increases.

The GCIs vector, then is sorted in a sequence. The distance between the GCIs is the length of glottal pulse or the pitch period. The GCIs determine the timing for the generation of the glottal pulses as well as the time at which the vocal tract parameters are to be updated. Changing the length of GCIs sequence alters the pitch period (fundamental frequency of synthesized voice signal).

The voice type can be change by altering the classification and the range for voice and unvoiced signal. More detail on the voice type classification, refer to figure 3.7.

The voice conversion is done by performing the DTW process. The result obtained is more predictable, which the user already specify whether from male to female or vice versa. Detail of this process is as in section 2.5.2.2: Voice Conversion.

3.1.3   Voice Synthesis

The voice synthesis procedure is designed as in figure 3.11.
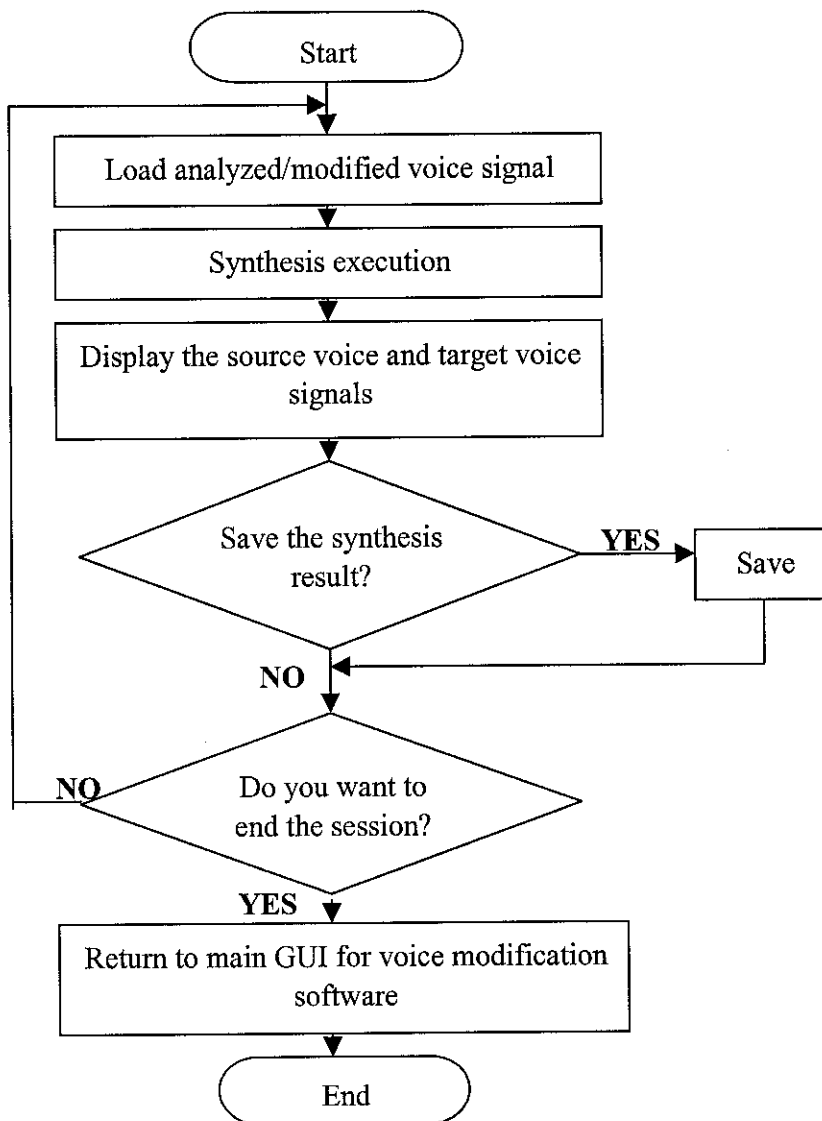
Figure 3-11   Synthesis Flowchart

The detail explanation of the synthesis execution in producing mimic voice is as in figure 3.12.
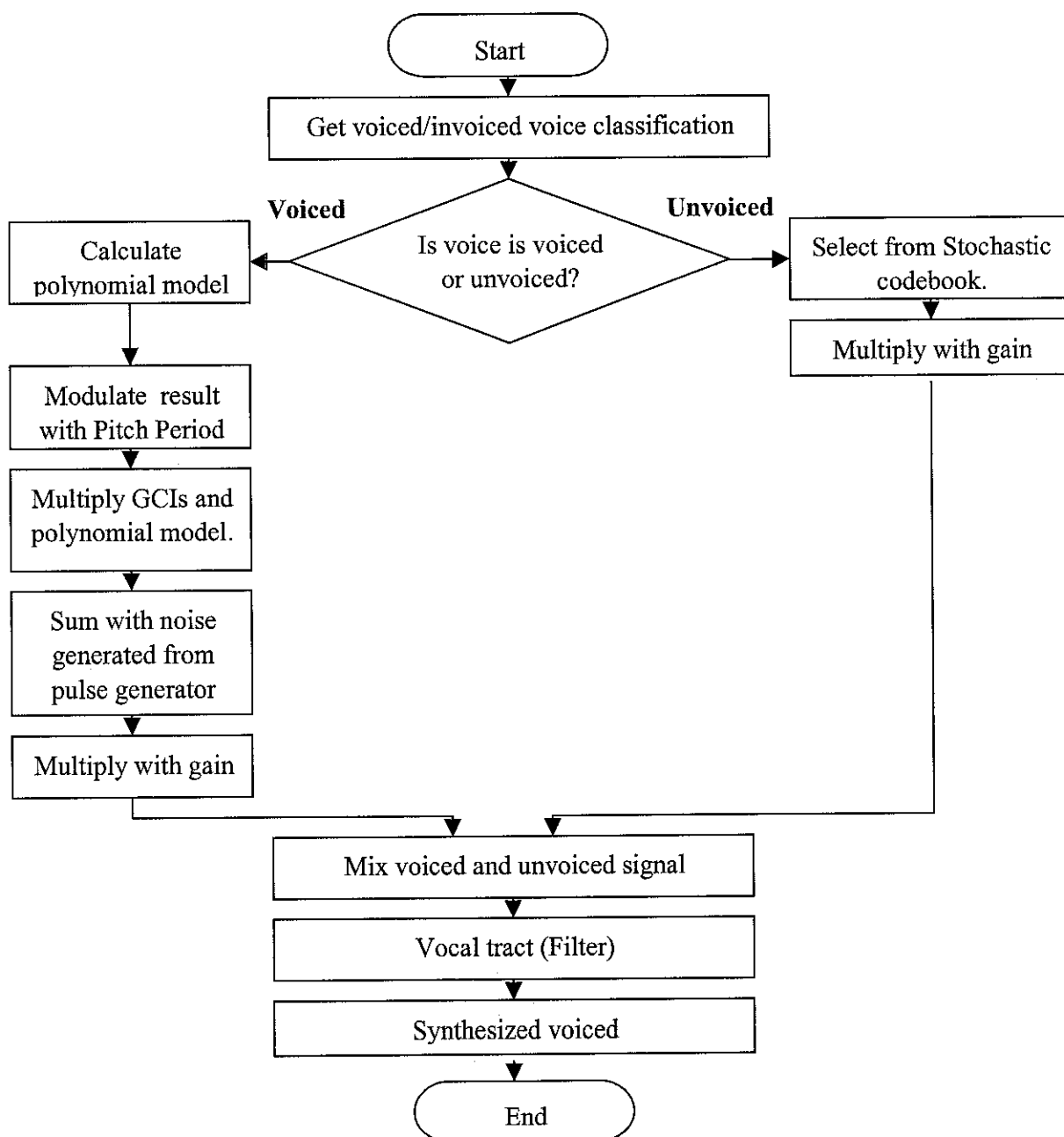
Figure 3-12 Synthesis execution Flowchart

The classification obtained from the analysis step is essential for the excitation
waveform selection, either using Polynomial model to produce a periodic waveform

or stochastic codebook to produce a noise. This procedure is referring to figure 2.15: Speech Synthesis Model. Pitch period is asserted to determine the timing for software to generate glottal pulses for each voiced signal classified. Both type of signals then multiply with gain to increase the loudness (intensity) of the signal. Vocal tract or the filter designed then filter the signal produces to specific bandwidth. Thus, finally producing a synthesized voice signal.

3.2 Hardware-Voice Changer

In order to design a solid-design for the voice changer, I had done few researches on the Internet and RS Catalogue for suitable component to be used for speech signal conversion.

Digital Signal Processing chip, which enable design of digital filter and selection of signal component such as pitch and intensity to be modified is only available in Surface Mount Technology (SMT) or comes with a bulgy development kit which is very expensive. For small, individual project purposes using SMT chip is very difficult and impossible to implement. The reasons are:

- SMT devices require a special soldering machine designed specifically for this purpose. This machine is usually available in semiconductor manufacturing industries.

- SMT devices also require a special Printed Circuit Board(PCB), to solder on.

Meanwhile, for Dual Inline Package(DIP) chip, it can be used universally either to individual user or company. In addition the PCB is easily available and cheap in price.

Using the Voice Modulator Chip from Holtek, HT8950A (16 pin DIP) as the modulation component, the frequency of input voice signal is altered.
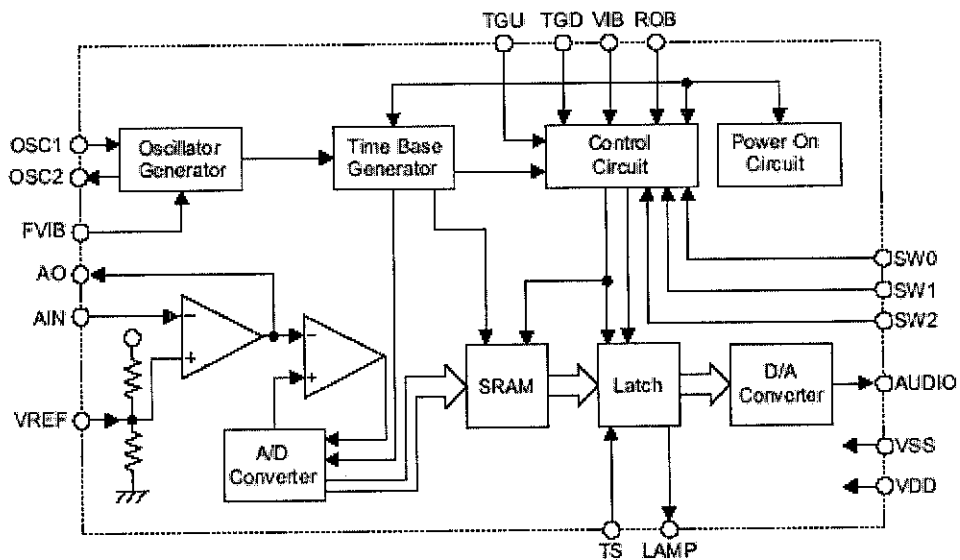
Figure 3-13 Block Diagram of HT8950 Voice Modulator Chip.

Inside the chip, there already builds-in oscillator, ADC/DAC, analog filter, pre-amplifier circuitry, and a modulator to alter the frequency of input signal in real-time. The block diagram of the modulation process is as in figure 3.13. Input signal is asserted into AIN pin to the pre-amplifier circuitry, then to an analog filter. Triggering the switches SW0,SW1,SW2 or TGU,TGD,VIB and ROB perform the modulation process. The two type of switches provide an alternative switching to user, either to use a toggle switches (function in binary) or micro-switches respectively. Table 1 gives the details on the steps in shifting the frequency.

Table 1 Switching steps for shifting the frequency.

Input			Step Mode	Speed Ratio
SW2	SW1	SW0		
1	1	1	Controlled by TGU and TDG	Notes
1	1	0	UP3	2
1	0	1	UP2	8/5
1	0	0	UP1	4/3
0	1	1	NORMAL	1
0	1	0	DN1	8/9
0	0	1	DN2	4/5
0	0	0	DN3	2/3
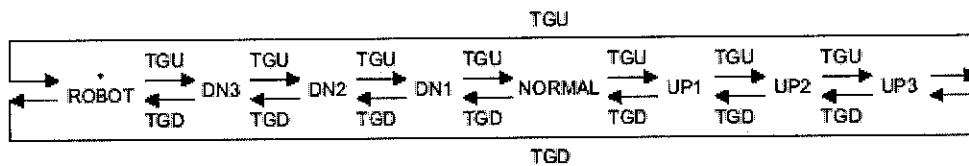
Figure 3-14 Frequency shifting using switches TGU and TGD.

Using the chip, it provides 7 steps of frequency shifting from UP3 till DN3 as described in table 1. Using toggle switches, SW0 , SW1, SW2 user has to trigger on or off the switches to get '1' or '0'. Meanwhile using the micro-switches is far easier where the user just needed to push the TGU or TGD switch as shown in figure 3.14.

The circuitry used to design the voice changer is as follows:
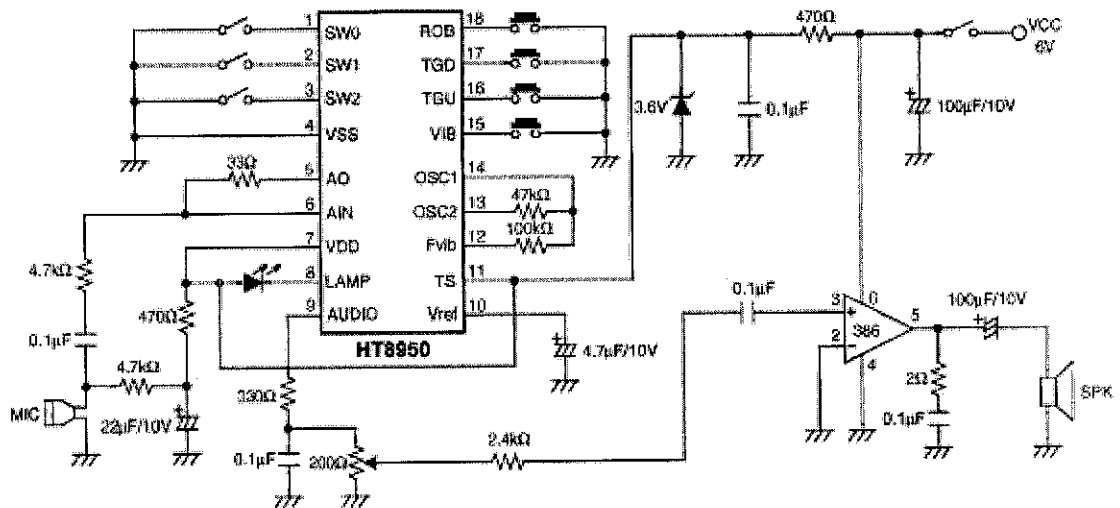
Figure 3-15 Voice Changer Circuitry.

For more details on the Holtek chip and other component used, please refer to Appendix D and E.

CHAPTER 4

RESULT AND DISCUSSION

4.1 Software-Voice Modification Application

The source codes to perform the conversion are attached in Appendix C.

On the execution of the software, user will be prompt a Graphical User Interface
(GUI) as in figure 4.1.
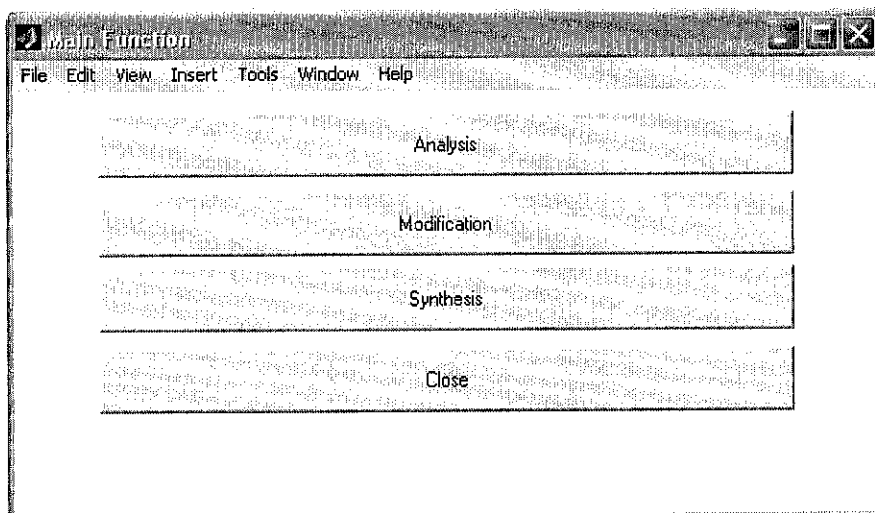
Figure 4-1 Main GUI Interface for Voice Modification Software

Triggering one of the procedure; analysis, modification and synthesis, will perform
the calculation and steps as described in section 3.1: Software: Voice Modification
Application. Close button is to close the overall modification process.

The software is able to perform the modification and conversion process. Since the
voice modification program is designed to only support voice signal in '.dat'

extension, a program to convert recorded voice (usually in '.wav') to ASCII format is also designed. User will need to enable *'wav2ascii.m'* in 'ASCII_conversion' folder.

There are two ways for user to view modification and conversion results, either by enabling synthesis procedure or by executing the *'speech_display.m'* program. Synthesis procedure display voice signal saved in ".mat" extension and *'speech_display.m'* display voice signal saved in ".dat" extension. Enabling the synthesis procedure, in the case of modified signal, an original and a synthesize voice signal are displayed. In the case of converted voice signal either from male to female and vice versa, an original, a synthesis and a target voice signal are displayed. Thus enabling user to see the differences between the original, synthetic and target signal. Meanwhile, executing the *'speech_display.m'* program, display four spaces ready for resulted modified or converted voice to be loaded.

The waveform from voice samples saying "Ayang is online" from a male and female are as in figure 4.2. We can observe from the signals, the waveform is gradually built up the then decreases with time for each word spoken. From the two signals, there are differences of in term of intensity (amplitude of the signal) and pitch (variation of frequency and intensity). In the software, analysis process is performed to take advantage of these differences, estimating the values of each to develop mimic sound in synthesis process.

This document will focus on the result obtained from the voice modification and conversion process. Later sections discuss both modification and conversion procedures.
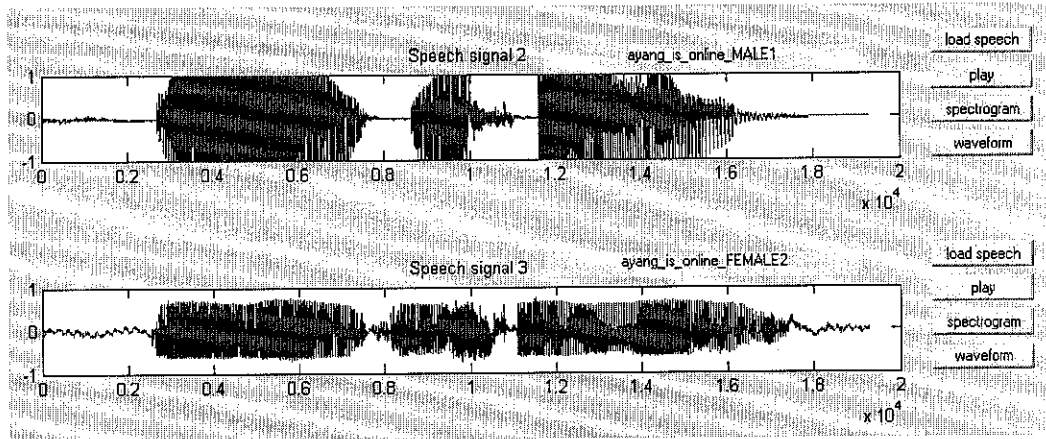
Figure 4-2 Original signals of male and female voice saying "Ayang Is Online".

4.1.1 Voice Modification

Enabling the 'Modification' pushbutton in the main voice modification GUI will prompt user another GUI as in figure 4.3. Details on the steps performed are as described in figure 3.2: Step by Step Procedure in Using Voice Modification Software. Once voice signal is loaded in analysis procedure, user is not required to load a new analyzed file to continue to modification procedure, since the data is already in memory. 'Load speech' pushbutton is enabled then, to perform modification on existing analyzed voice signal. Voice modification is a process where the parameters obtained in analysis procedure previously is randomly altered to produce a new voice as output signal. Detail usage of this procedure can be referred to figure 4.3. The output to the modification cannot be predicted, since the voice type, pitch period and gain of the signal is randomly changed. More predicted modification of voice signal is performed in 'Voice Conversion' stage, where the source voice signal is transformed into target voice signal to produce synthesized signal.
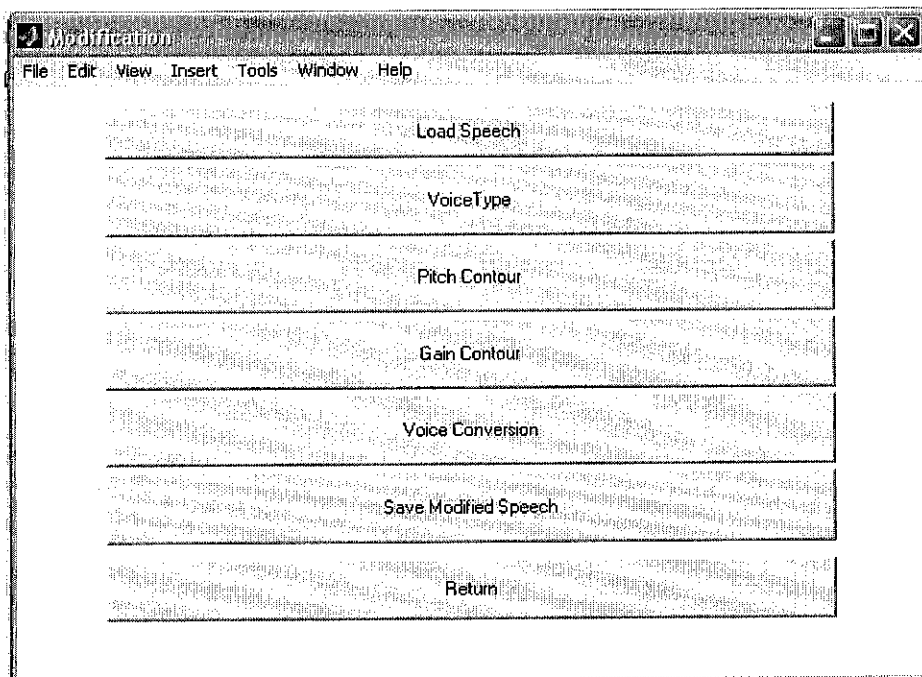
Figure 4-3 Modification GUI

Performing voice type, pitch period and gain modification resulted in changing voice characteristics. Figure 4.4 and 4.5 described an example result of random modification done to male and female voice respectively. The modification is done randomly to voice parameters. The modified male voice (figure 4.4), when played the voice output produces is more harsh and rough, as if producing another type of male voice. The modification meets the objective in term of producing a new voice, which can be observed by the difference in the synthesized output signal waveform. Meanwhile the female voice (figure 4.5) when played, also produces more harsh and rough voice output.
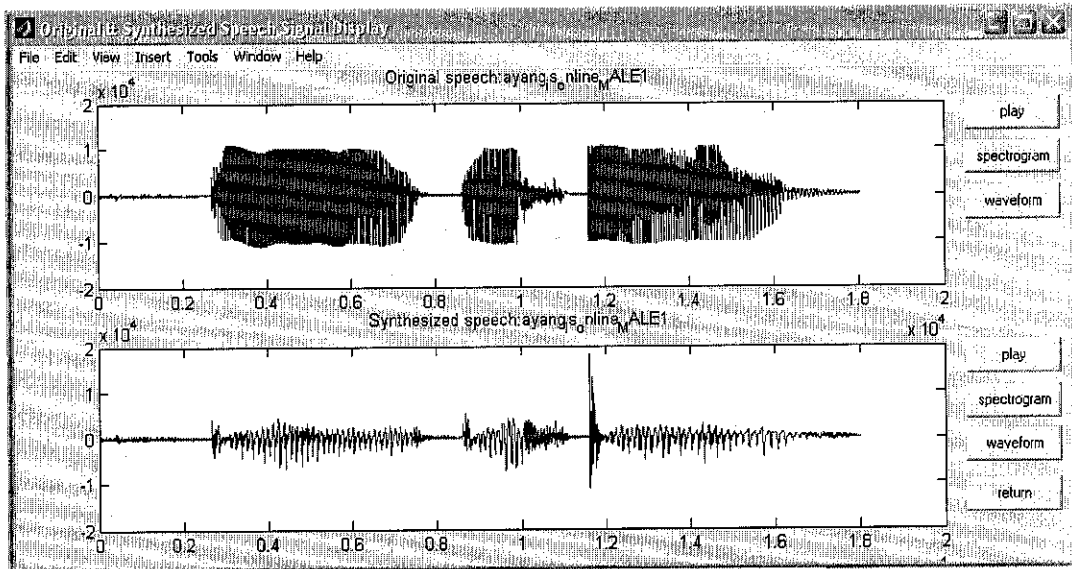
Figure 4-4 Modification in pitch and gain contour of male voice saying "Ayang Is Online".
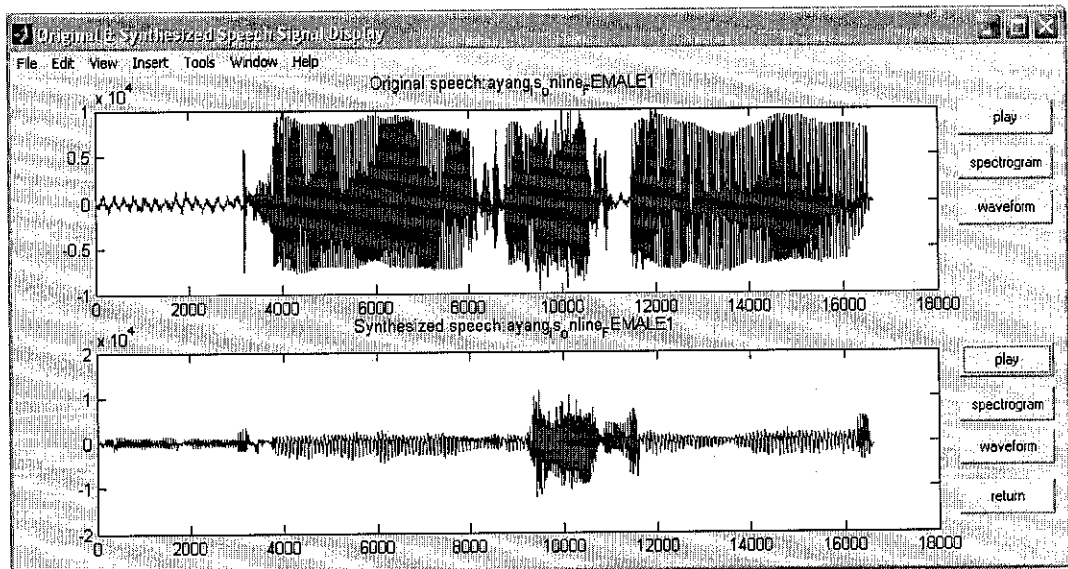
Figure 4-5 Modification in pitch and gain contour of female voice saying "Ayang Is Online".

Note that, from the synthesized signal for both types of voices, the amplitudes are much lesser than the original. The voice intensity (perceived as loudness of a sound by listener) maybe degrades, but this does not affect the quality. The play, spectrogram and waveform pushbuttons enable user to play the voice sound, view the spectrogram waveform and redisplay the waveform respectively.

More samples of the result obtained can be referred in the Appendix B.

4.1.2 Voice Conversion

The conversion process is done using DTW method. Details of the steps and algorithms are as in section 3.1.2: Voice Modification and in figures 3.11 and 3.12. The conversion from male to female voice signal is as in figure 4.6. As we can see in the output (synthesized signal), the converted voice waveform produced is different from both source and target voice. Thus the conversion is a success. Playing the output sound produces a sound almost similar to female voice but a bit rough. The conversion from female to male conversely is as in figure 4.7. The voice produces when played is quite harsh almost similar to male but yet soft.

More sample of the result obtained can be referred in the Appendix B.
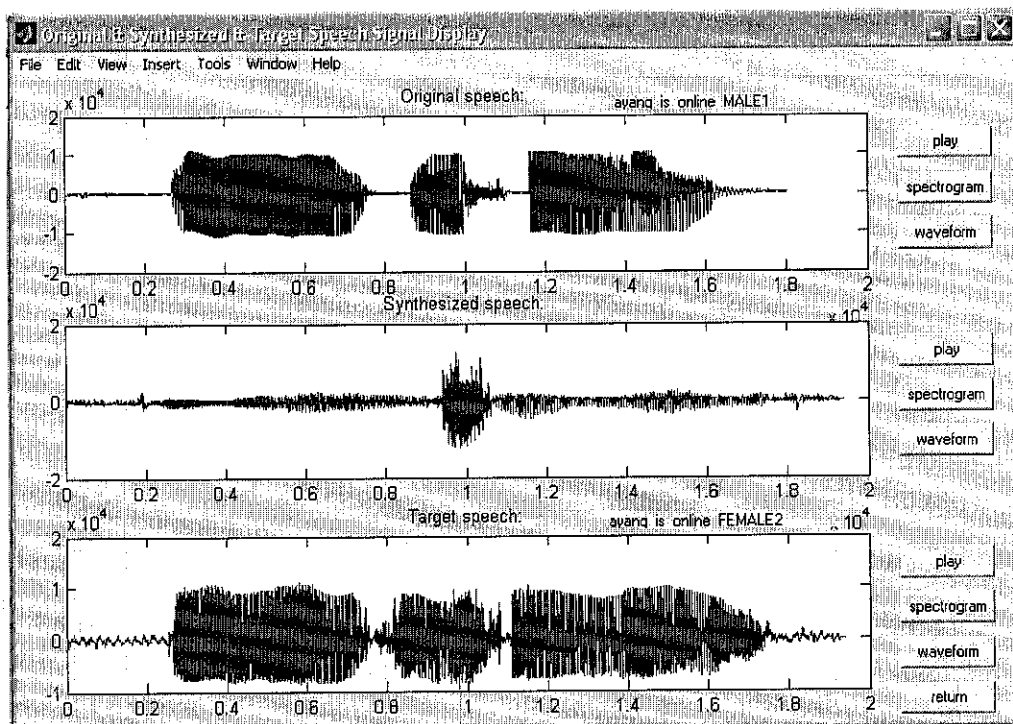
Figure 4-6 Voice Conversion from male to female saying "Ayang Is Online".
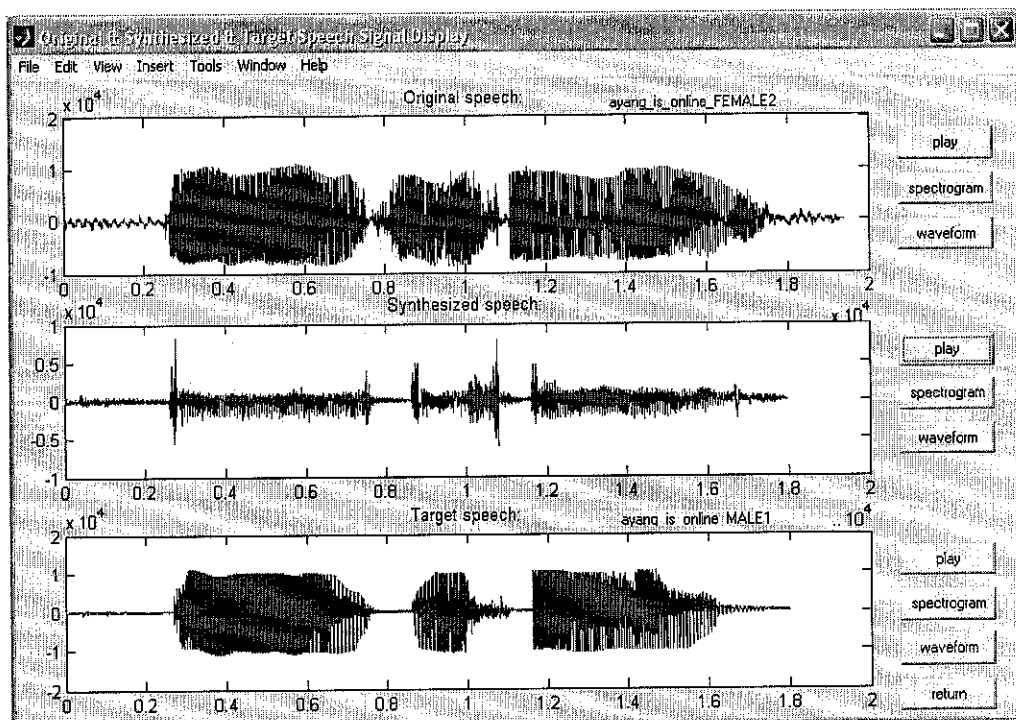
Figure 4-7 Conversion from female to male saying "Ayang Is Online".

4.2 Hardware-Voice Changer

The voice changer kit designed is as in figure 4.8:
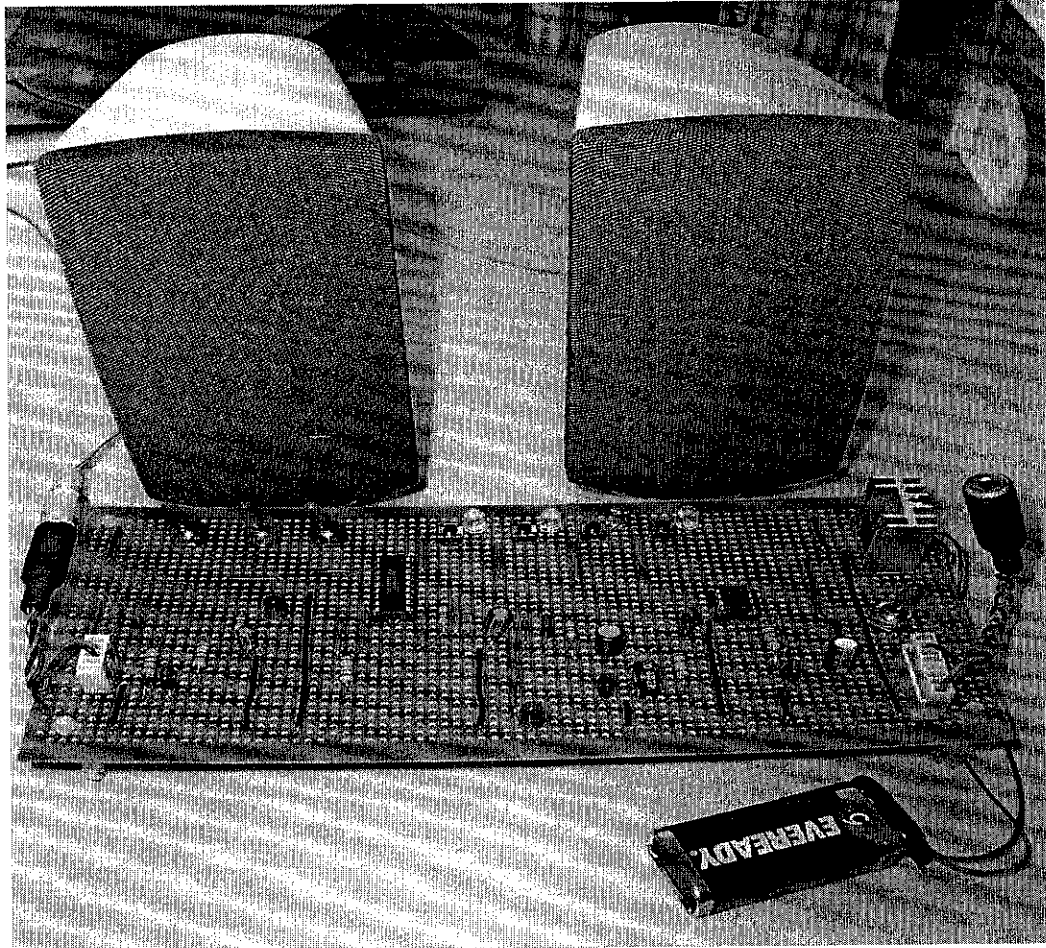
Figure 4-8 Voice Changer Kit

The voice changer kit can change your voice to various different tones. This kit lets your voice be changed to add a vibrato, like those used to disguise secret witnesses. It also allows you to change the tone of your voice upward or downward, women sound like men and vice versa. The kit also includes a Robotic voice as well.

Circuit Description:

The circuit operates around a voice modulator Holtek 16 pin dip package integrated circuit part number HT8950. This IC provides two special effects, vibrato and robot.

The vibrato effect is generated by alternating the frequency of an input signal up and down at a rate of 8Hz. The robot function on the other hand converts an input voice into a robot voice. Both effects can be triggered, either robot or vibe. For the output frequency level shifting, the chip provides 7 steps which can be selected from the two groups of pins namely, SW0, SW1, SW2 and TGU, TGD. Configuration of the switches provides 7 different tones each. The other IC in the circuit is a LM386 AF power output 1W with an output to a 8 ohm speaker. The microphone is soldered directly to the PC board. The ICs are static sensitive so caution should be used when removing them off the static dissipative foam block. Static can be removed from your body by touching a ground or metal cabinet with your finger for a full second. A static wrist strip can also be used. A 9V battery used to power up the circuit.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

Based on the result obtained in section 4, concluded that voice modification applies in software and hardware is able to be prosecuted. The applications are able to perform modification from a source voice signal to another voice signal.

For the software part, voice modification program is designed using MATLAB. The program is designed to support only modification of voice signal saved in ".dat" extension. The program is divided into three major steps to perform the voice modification that is analysis, modification and synthesis. The analysis step is performed based on Linear Predictive (LP) method to calculate the voice waveform features such as pitch, intensity and glottal closure. These features are then used to perform synthesis step, which to mimic the source voice signal. The modification step provides a platform for modification of features calculated previously. In addition, performing conversion of male voice signal to female voice signal and vice versa based on Dynamic Time Wrapping (DTW) method.

For the hardware part, voice modification kit is designed. The kit is designed using HOLTEK voice modulator chip. The kit provides conversion of any input voice signal to seven type of output voice signal. The conversion is done by modulating the frequency of the input voice signal.

Nevertheless, the voice modification applications: software and hardware; both has limitations. The applications are prone to noise, in addition producing output voice signal which contains noises. Moreover, noise itself is generated in the software part, to produce an unvoiced signal.

As for recommendation for future expansion and continuation of this project, a better filter is needed to be introduced both in software and hardware to filter all the noises.

In this project, there are two separate parts to perform the voice modification, which is software and hardware. These two parts apply two different concepts which is signal modulation and signal processing; but with the same objective, which to produce a modified voice signal. Therefore, designing a compact hardware application, which using digital signal processing chip to perform voice modification will be able to improve the modification analysis and performances. The voice signal then will not only be limited to seven types only for more voice features is able to be modified. Moreover, digital processing is more reliable than analog processing (signal modulation), thus able to produce a better output voice signal.

Another recommendation is to enhance the voice modification simulation so that it able to modify a voice signals to another type of voice such as child voice. In addition, able to perform the simulation in real-time without the need to record voice signals, and play two voice signals at the same time (chorus).

REFERENCES

1. Sadaoki Furui, "Digital Speech Processing, Synthesis, and Recognition", Marcel Dekker,Inc, New York and Bessel, 1989

2. Converssy Technologies, "Distributed Voice Processing" White Paper, 2001

3. Bryan L. Pellom and John H.L Hansen "Experimental Study of Speaker Verification Sensitivity to Computer Voice Altered Imposter" Robust Speech Laboratory, Duke University,2000.

4. Lajos Hanzo,F.Clare A. Somerville,Jason P. Wodard. "Voice Compression and Communication- Principal Application for Fixed and Wireless Channel.", IEEE Series in Digital & Mobile Communication-Wiley Intersciece, 2001.

5. N.Jayant, "Signal Compression-Coding of Speech, Audio, Text, Image and Video." , World Scientific, 1997.

6. Farzad Nekoogar & Gene Moriorty, "Digital Control using Digital Signal Processing.", Prentice Hall, 1999.

7. D.G. Childers. "Speech Processing and Synthesis Toolboxes", John Wiley & Sons, Inc, 2000.

8. L.R. Rabiner, R.W. Schafer. "Digital Processing of Speech Signals.", Prentice Hall, 1978.

9. W.B.Kleijn, K.K. Paliwal editors, "Speech Coding and Synthesis." Elsevier Science B.V., 1995.

10. James H. McChellan, C. Sidney Burrus, Alan V. Appenheim, Thomas W. Paks, Ronald W. Schafer, Hans W. Schuessler. "Computer-Based Exercises for Signal Processing using Matlab 5.0.", Prentice Hall, 1998.

11. RS South Asia Catalogue, April 2002

12. Steven W. Smith , "The Scientist and Engineer's Guide to Digital Signal Processing", Ph.D. California Technical Publishing, 1997

13. Stallings, W., "Data & Computer Communication", Sixth Edition, Prentice Hall, 2000

14. Holtek Semiconductor Inc., HT8950 Datasheet,2003.

15. National Semiconductor, LM386 Low Voltage Power Amplifier, August 2003.

16. www.mobile.esc.soton.ac.uk/speech_codecs

17. www.sciencedirect.com

18. http://users.pandora.be/oldmicrophones/microphone_info.htm

19. http://en.wikipedia.org/wiki/Microphone

20. http://electronics.howstuffworks.com/amplifier1.htm

APPENDIX A
FORMULAE USED IN LINEAR PREDICTIVE COEFFICIENT
CALCULATION

Flexible LP Coefficient

Energy

Energy of the voice signal can be obtained by multiplying the signal with its conjugate.

$$Energy = \frac{1}{N} \sum_{n=1}^{N} S(n)S(n)$$

First Order Reflection Coefficient

The first order reflection coefficient is

$$r_1 = \frac{Rss(1)}{Rss(0)}$$

where

$$Rss(0) = Energy$$
$$Rss(1) = \frac{1}{N} \sum_{n=1}^{N-1} S(n)S(n+1)$$

where N is the number of samples in the analysis frame and s(n) is the speech sample.

Linear Predictive Error

The linear predictive error is

$$E_T^2 = \sum_{n=0}^{N-1} e^2(n) = \sum_{n=0}^{N-1} (s(n) - \hat{s}(n))^2 = \sum_{n=0}^{N-1} (s(n) + \sum_{k=1}^{p} \hat{a}s(n-k))^2$$

where $\hat{s}$ is the estimate of true signal values and $\hat{a}$ is the negative of first order reflection coefficient. Meanwhile p is the number of first order reflection coefficient.

LP Coefficient

This calculation is performed using Auto-regression technique. This coefficient is used to design the LP model which is the design of the vocal tract filter.

$$s(n) = -\sum_{k=1}^{p} \hat{a}S(n-k) + e(n) = \hat{s}(n) + e(n)$$

z-transform to the equation yield

$$S(z) = -\left[\sum_{k=1}^{p} \hat{a}z^{-1}\right] S(z) + E(z) = \hat{s}(z) + E(z)$$

$$S(z) = \frac{E(z)}{1 + \left[\sum_{k=1}^{p} \hat{a}_k z^{-1}\right]} = \frac{E(z)}{A(z)}$$

where S(z) is the filter equation implemented.

The Polynomial Model

The polynomial model described here are excerpted from Childer[8]. The model is

$$p(t) = c_0 + c_1\tau + c_2\tau^2 + c_3\tau^3 + c_4\tau^4 + c_5\tau^5 + c_6\tau^6$$

where t is the independent variable and $\tau = \dfrac{1}{T}$. T is the period of the pitch period.

APPENDIX B

SAMPLE RESULTS FROM VOICE MODIFICACATION

SOFTWARE

This sample results is taken for voice signal saying "We were away a year ago". From the waveform we can observe the differences between the original and synthesized signal. These results are taken from both random modification and conversion from male to female and vice versa.
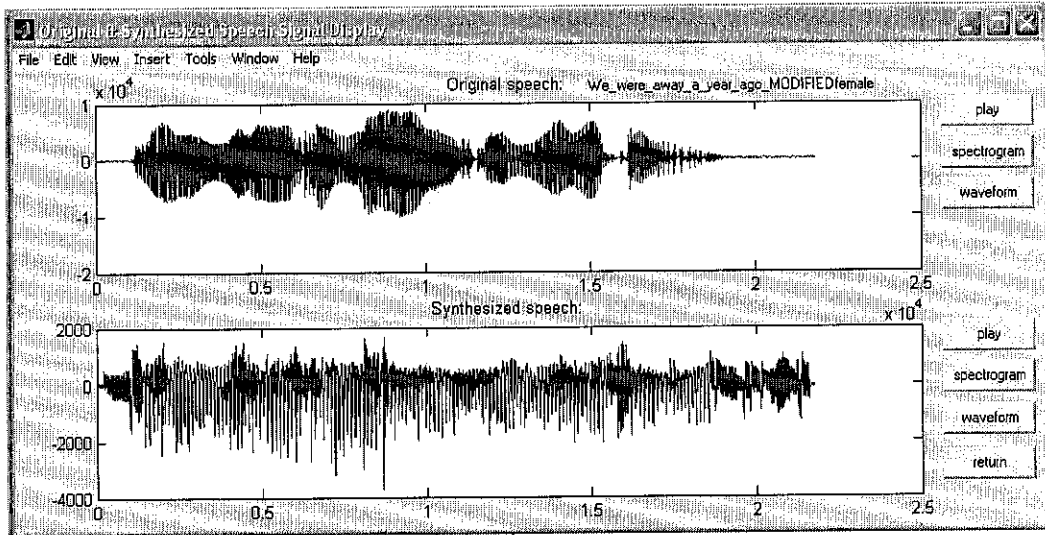
Figure A. Randomly modified female voice saying "we were away a year ago".
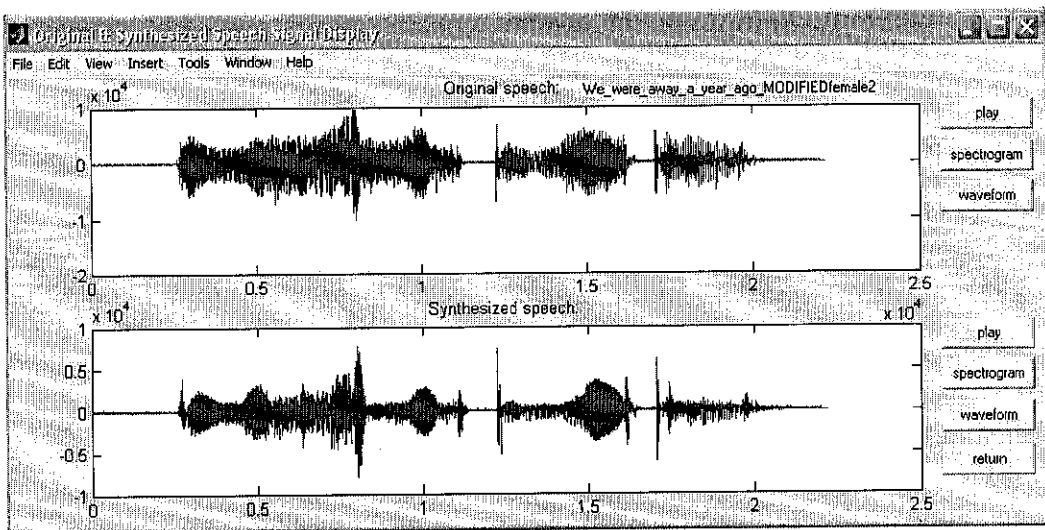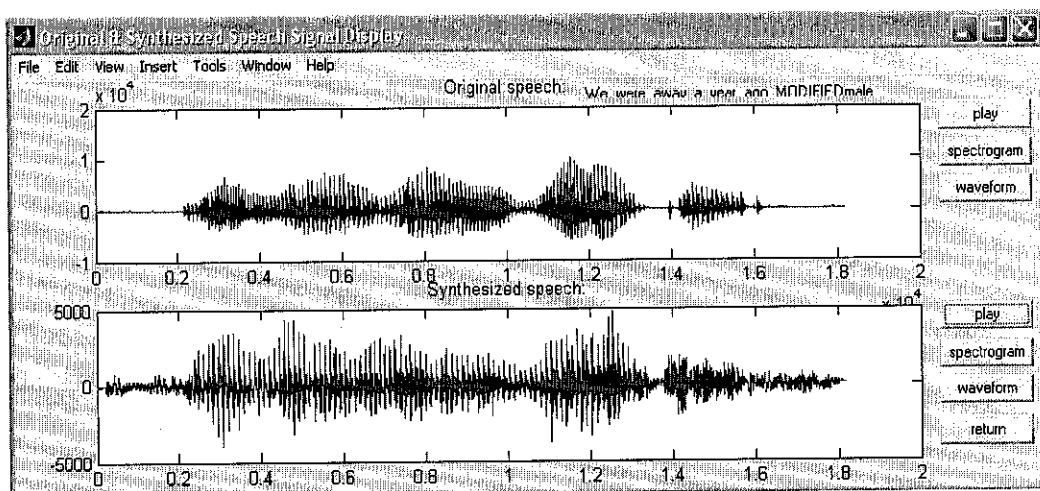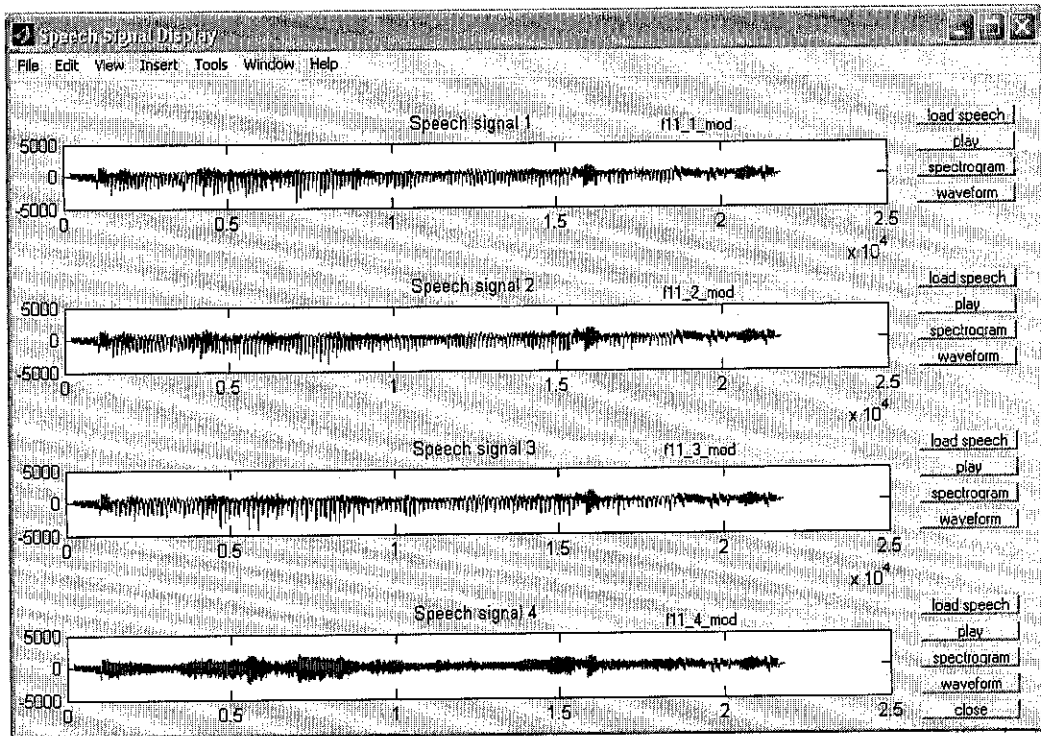

Figure B. Randomly modified female voice saying "we were away a year ago".
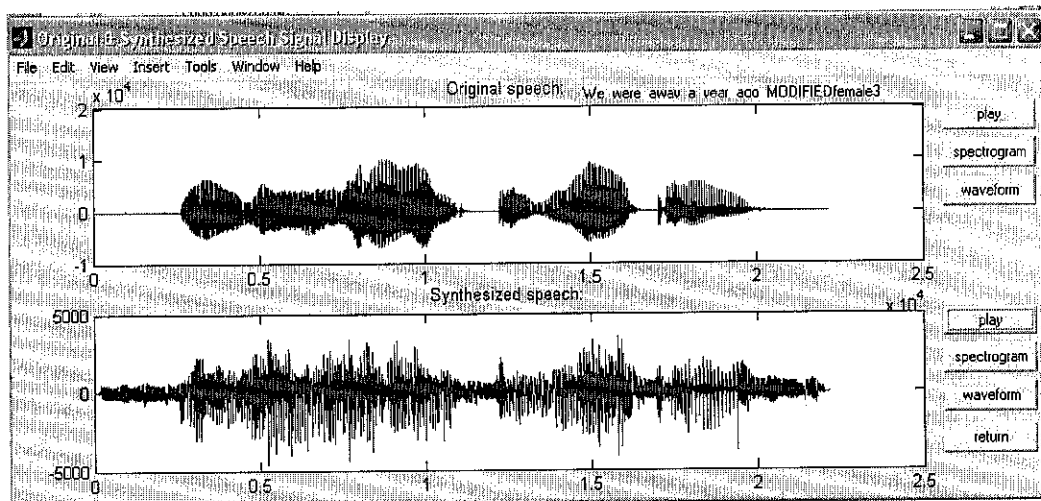
Figure C. Randomly modified female voice saying "we were away a year ago".

Figure D. Randomly modified male voice saying "we were away a year ago".

Figure E. Comparison of synthesized voice signal waveform of a woman saying "we were away a year ago."

Note that in figure E, show the sample of 4 different sampling synthesized female voice signal saying "we were away a year ago." On the fourth waveform (speech signal 4) then, we observe the difference in term of intensity and pitch. When played, the voice produces is harsh and different from original.

APPENDIX C

SOURCE CODES FOR VOICE MODIFICATION SOFTWARE

The source code is available in Volume II: Source Code of Voice Modification Software.

APPENDIX D
HOLTEK VOICE MODULATOR CHIP, HT8950

HT8950
Voice Modulator

Faxback Doc. # 9007547

tures

single power supply: 2.4~4.0V
)n chip SRAM
Robot function
/ibrato function
8KHz sampling rate
' steps of level shifting

- 8 bit A/D and D/A converters
- LED indicator with voice level
- Selection of push button or electronic mode
- Few external components required
- 16/18 pin DIP package

olications

Toys
Mixers
Recorders

- Audio system
- Speech system
- Telephone system

neral Description

HT8950 is a single chip CMOS LSI of a e modulator. It provides 7 steps to shift the uency of an input voice, yielding a dramati-change in the output.

HT8950 provides two special effects: Vi-to and Robot. The Vibrato effect is generated alternating the frequency of an input signal and down at a rate of 8Hz. The Robot func-., on the other hand, converts an input voice a Robot voice. Both effects are selected ending on ROB and VIB pin be triggered.

the output frequency level shifting, the 3950 provides 7 steps and they are selected ending on two groups of pins, namely SW0,

SW1 and SW2 for electronic direct selection and ROB, TGD, TGU and VIB for push button selection.
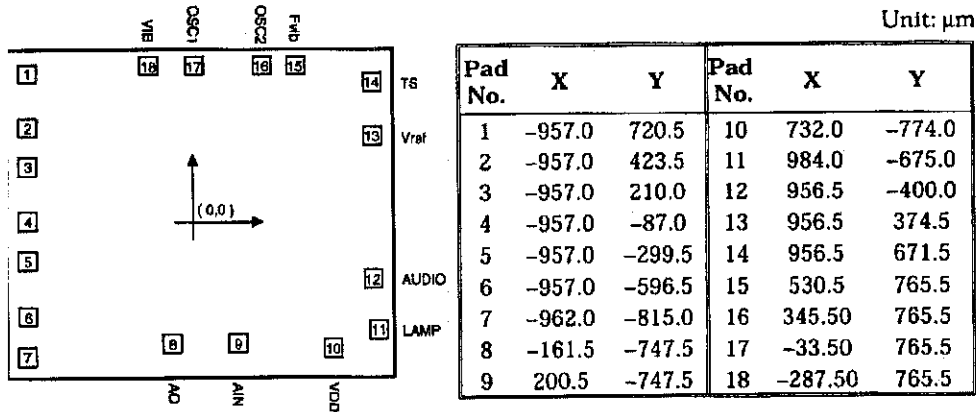
The HT8950 includes a built-in microphone amplifier with an internal bias, an 8 bit A/D converter, a built-in SRAM as well as a current output type of 8 bit D/A converter. The 8 bit A/D and D/A converters with a sampling rate of 8KHz ensure an output voice both of high quality and with a high S/N ratio. The LSI provides an LED indicator flashing with the volume of input voices. It is offered in a 16 pin or 18 pin DIP package.

Assignment

HT8950A – 16 DIP

SC1	1	16	OSC2
VIB	2	15	Fvib
TGU	3	14	TS
TGD	4	13	Vref
ROB	5	12	AUDIO
VSS	6	11	LAMP
NC	7	10	VDD
AO	8	9	AIN

HT8950 – 18 DIP

SW0	1	18	ROB
SW1	2	17	TGD
SW2	3	16	TGU
VSS	4	15	VIB
AO	5	14	OSC1
AIN	6	13	OSC2
VDD	7	12	Fvib
LAMP	8	11	TS
AUDIO	9	10	Vref

ck Diagram

Coordinates

Unit: μm

Pad No.	X	Y	Pad No.	X	Y
1	−957.0	720.5	10	732.0	−774.0
2	−957.0	423.5	11	984.0	−675.0
3	−957.0	210.0	12	956.5	−400.0
4	−957.0	−87.0	13	956.5	374.5
5	−957.0	−299.5	14	956.5	671.5
6	−957.0	−596.5	15	530.5	765.5
7	−962.0	−815.0	16	345.50	765.5
8	−161.5	−747.5	17	−33.50	765.5
9	200.5	−747.5	18	−287.50	765.5

Chip size: 2350×2080 (μm)^2

e IC substrate should be connected to VDD in the PCB layout artwork.

Description (18 Pin Versoin)

ı No.	Pin Name	I/O	Internal Connection	Description
ı	SW0	I	Pull-High	Function setting pin (for electronic setting)
?	SW1	I	Pull-High	Function setting pin (for electronic setting)
3	SW2	I	Pull-High	Function setting pin (for electronic setting)
4	VSS	I	—	Negative power supply (GND)
5	AO	O	—	Output of an internal amplifier
6	AIN	I	—	Input of an internal amplifier (inverting)
7	VDD	O	—	Positive power supply
8	LAMP	O	NMOS Open Drain	Lamp output (brightness changes with voice volume)
9	AUDIO	O	PMOS Open Drain	Audio output
10	Vref	I	—	Reference voltage of an internal amplifier
11	TS	I	—	For IC test only
12	Fvib	O	—	Vibrato frequency control
13	OSC2	O	—	Oscillator output
14	OSC1	I	—	Oscillator input
15	VIB	I	Pull-High	Switch to the Vibrato mode (toggle)

No.	Pin Name	I/O	Internal Connection	Description
6	TGU	I	Pull-High	Switch to the Upward step mode
7	TGD	I	Pull-High	Switch to the Downward step mode (internal pull-high)
8	ROB	I	Pull-High	Switch to the Robot mode (internal pull-high)

olute Maximum Ratings

ly Voltage ....–0.3V to 6V   Storage Temperature.................–50°C to 125°C

t Voltage................ V$_{SS}$–0.3V to V$_{DD}$+0.3V   Operating Temperature..............–20°C to 70°C

trical Characteristics

(Ta=25°C)

nbol	Parameter	Test Condition		Min.	Typ.	Max.	Unit
		V$_{DD}$	Condition				
D	Operating Voltage	—	—	2.4	3.0	4.0	V
P	Operating Current	3V	No load, F$_{OSC}$=640KHz	—	2.0	10	mA
N	Input Signal	3V	—	—	—	580	mV
MP	Lamp Sink Current	3V	V$_{OL}$=1.3V	5.0	9.5	—	mA
r	OPA Gain Value	3V	Open loop	—	2000	—	
	Audio Output Voltage	3V	At silence	—	–1.0	—	mA
H	"H" Input Voltage	—	—	0.7V$_{DD}$	—	—	V
L	"L" Input Voltage	—	—	—	—	0.3V$_{DD}$	V
SC	System Frequency	3V	R$_{OSC}$=47KΩ	—	512	—	KHz

ctional Description

HT8950 is a single chip LSI designed for modulation. It provides 7 steps to shift the uency of an input voice signal up and down. LSI in addition is equipped with 2 special ts, Vibrato and Robot.

HT8950 includes a built-in amplifier, 8 bit converter and current output type of 8 bit converter in addition to a built-in SRAM. brightness of an LED indicator changes the volume of an input voice signal.

Power on initial

The HT8950 enters the Robot state right after power is initially switched on.

Robot state

The system goes into the Robot state after the ROB pin is triggered or power is turned on. In this mode, an input voice can be converted into a robot voice.

ato

)utput voice will be generated with a vibrato
:t when the VIB pin is triggered, regardless
hat state the system is in. The vibrato effect
)ggle activated. That is to say, when a voice
)ut is playing with a vibrato effect, this ef-
can be eliminated by retriggering the VIB
The rate of vibrato effect can be changed by
isting the resistance of an external resistor
veen the OSC2 and Fvil pins.

:e modulation

: HT8950 provides an 8 bit A/D and a D/A
verters with a sampling rate of 8KHz, ensur-
a voice output of high quality and with a
ı S/N ratio. The LSI includes 7 steps to shift
frequency of an input signal. The voice
iulation is selected and decided by the
0~SW2 inputs.

Input			Step Mode	Speed Ratio
V2	SW1	SW0		
1	1	1	Controlled by TGU and TDG	Notes
1	1	0	UP3	2
1	0	1	UP2	8/5
1	0	0	UP1	4/3
)	1	1	NORMAL	1
0	1	0	DN1	8/9
0	0	1	DN2	4/5
0	0	0	DN3	2/3

tes: The TGU switch elevates one step mode,
d the TGD switch, on the other hand, falls
ɛ step mode step by step as shown:
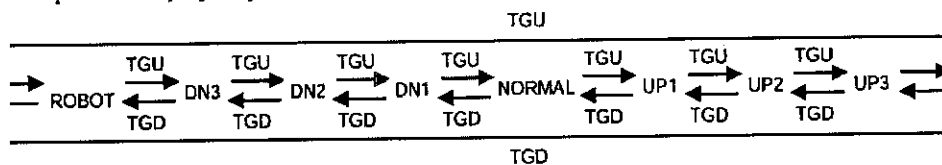
- The system changes to the Robot state after
  the ROB pin is triggered, regardless of what
  state the system is in.

- A voice output is accompanied with a vibrato
  effect after the VIB pin is triggered, regard-
  less of what state the system is in.

LED Indicator

The HT8950 provides a LAMP pin to drive an
external LED. The brightness of LED changes
with the volume of an input voice signal.

TGU

Application Circuits

APPENDIX E
LM386-POWER AMPLIFIER.

N *National Semiconductor*

LM386
Low Voltage Audio Power Amplifier

General Description

The LM386 is a power amplifier designed for use in low voltage consumer applications. The gain is internally set to 20 to keep external part count low, but the addition of an external resistor and capacitor between pins 1 and 8 will increase the gain to any value from 20 to 200.

The inputs are ground referenced while the output automatically biases to one-half the supply voltage. The quiescent power drain is only 24 milliwatts when operating from a 6 volt supply, making the LM386 ideal for battery operation.

Features

- Battery operation
- Minimum external parts
- Wide supply voltage range: 4V–12V or 5V–18V
- Low quiescent current drain: 4mA
- Voltage gains from 20 to 200
- Ground referenced input
- Self-centering output quiescent voltage
- Low distortion: 0.2% (A_V = 20, V_S = 6V, R_L = 8Ω, P_O = 125mW, f = 1kHz)
- Available in 8 pin MSOP package

Applications

- AM-FM radio amplifiers
- Portable tape player amplifiers
- Intercoms
- TV sound systems
- Line drivers
- Ultrasonic drivers
- Small servo drivers
- Power converters

Equivalent Schematic and Connection Diagrams

DS006976-1

DS006976-2

Top View
Order Number LM386M-1,
LM386MM-1, LM386N-1,
LM386N-3 or LM386N-4
See NS Package Number
M08A, MUA08A or N08E

Small Outline,
Molded Mini Small Outline,
and Dual-In-Line Packages

Absolute Maximum Ratings (Note 2)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

Supply Voltage	
(LM386N-1, -3, LM386M-1)	15V
Supply Voltage (LM386N-4)	22V
Package Dissipation (Note 3)	
(LM386N)	1.25W
(LM386M)	0.73W
(LM386MM-1)	0.595W
Input Voltage	±0.4V
Storage Temperature	−65°C to +150°C
Operating Temperature	0°C to +70°C
Junction Temperature	+150°C
Soldering Information	

Dual-In-Line Package	
Soldering (10 sec)	+260°C
Small Outline Package	
(SOIC and MSOP)	
Vapor Phase (60 sec)	+215°C
Infrared (15 sec)	+220°C

See AN-450 "Surface Mounting Methods and Their Effect on Product Reliability" for other methods of soldering surface mount devices.

Thermal Resistance	
θ_{JC} (DIP)	37°C/W
θ_{JA} (DIP)	107°C/W
θ_{JC} (SO Package)	35°C/W
θ_{JA} (SO Package)	172°C/W
θ_{JA} (MSOP)	210°C/W
θ_{JC} (MSOP)	56°C/W

Electrical Characteristics (Notes 1, 2)

$T_A = 25°C$

Parameter	Conditions	Min	Typ	Max	Units
Operating Supply Voltage (V_S)					
LM386N-1, -3, LM386M-1, LM386MM-1		4		12	V
LM386N-4		5		18	V
Quiescent Current (I_Q)	$V_S = 6V$, $V_{IN} = 0$		4	8	mA
Output Power (P_{OUT})					
LM386N-1, LM386M-1, LM386MM-1	$V_S = 6V$, $R_L = 8\Omega$, THD = 10%	250	325		mW
LM386N-3	$V_S = 9V$, $R_L = 8\Omega$, THD = 10%	500	700		mW
LM386N-4	$V_S = 16V$, $R_L = 32\Omega$, THD = 10%	700	1000		mW
Voltage Gain (A_V)	$V_S = 6V$, f = 1 kHz		26		dB
	10 µF from Pin 1 to 8		46		dB
Bandwidth (BW)	$V_S = 6V$, Pins 1 and 8 Open		300		kHz
Total Harmonic Distortion (THD)	$V_S = 6V$, $R_L = 8\Omega$, P_{OUT} = 125 mW		0.2		%
	f = 1 kHz, Pins 1 and 8 Open				
Power Supply Rejection Ratio (PSRR)	$V_S = 6V$, f = 1 kHz, C_{BYPASS} = 10 µF		50		dB
	Pins 1 and 8 Open, Referred to Output				
Input Resistance (R_{IN})			50		kΩ
Input Bias Current (I_{BIAS})	$V_S = 6V$, Pins 2 and 3 Open		250		nA

Note 1: All voltages are measured with respect to the ground pin, unless otherwise specified.

Note 2: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. Operating Ratings indicate conditions for which the device is functional, but do not guarantee specific performance limits. Electrical Characteristics state DC and AC electrical specifications under particular test conditions which guarantee specific performance limits. This assumes that the device is within the Operating Ratings. Specifications are not guaranteed for parameters where no limit is given, however, the typical value is a good indication of device performance.

Note 3: For operation in ambient temperatures above 25°C, the device must be derated based on a 150°C maximum junction temperature and 1) a thermal resistance of 107°C/W junction to ambient for the dual-in-line package and 2) a thermal resistance of 170°C/W for the small outline package.

plication Hints

CONTROL

ake the LM386 a more versatile amplifier, two pins (1) are provided for gain control. With pins 1 and 8 open 35 kΩ resistor sets the gain at 20 (26 dB). If a capacitor from pin 1 to 8, bypassing the 1.35 kΩ resistor, the will go up to 200 (46 dB). If a resistor is placed in series he capacitor, the gain can be set to any value from 20 0. Gain control can also be done by capacitively cou- a resistor (or FET) from pin 1 to ground.

onal external components can be placed in parallel he internal feedback resistors to tailor the gain and fre- cy response for individual applications. For example, an compensate poor speaker bass response by fre- cy shaping the feedback path. This is done with a series om pin 1 to 5 (paralleling the internal 15 kΩ resistor). dB effective bass boost: R ≃ 15 kΩ, the lowest value iod stable operation is R = 10 kΩ if pin 8 is open. If pins I 8 are bypassed then R as low as 2 kΩ can be used. estriction is because the amplifier is only compensated osed-loop gains greater than 9.

INPUT BIASING

The schematic shows that both inputs are biased to ground with a 50 kΩ resistor. The base current of the input transistors is about 250 nA, so the inputs are at about 12.5 mV when left open. If the dc source resistance driving the LM386 is higher than 250 kΩ it will contribute very little additional offset (about 2.5 mV at the input, 50 mV at the output). If the dc source resistance is less than 10 kΩ, then shorting the unused input to ground will keep the offset low (about 2.5 mV at the input, 50 mV at the output). For dc source resistances between these values we can eliminate excess offset by putting a resistor from the unused input to ground, equal in value to the dc source resistance. Of course all offset problems are eliminated if the input is capacitively coupled.

When using the LM386 with higher gains (bypassing the 1.35 kΩ resistor between pins 1 and 8) it is necessary to bypass the unused input, preventing degradation of gain and possible instabilities. This is done with a 0.1 µF capacitor or a short to ground depending on the dc source resistance on the driven input.

Typical Performance Characteristics

Quiescent Supply Current vs Supply Voltage

DS006976-5
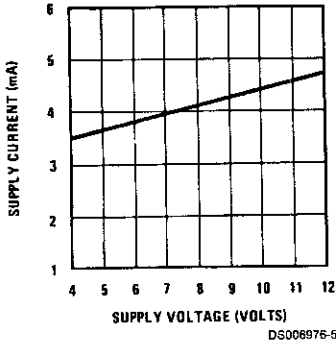
Power Supply Rejection Ratio (Referred to the Output) vs Frequency

DS006976-12
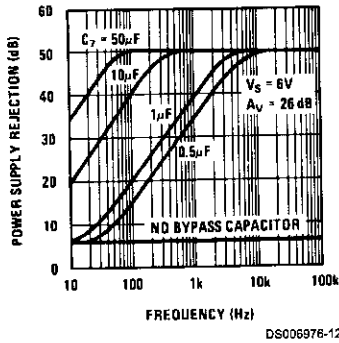
Peak-to-Peak Output Voltage Swing vs Supply Voltage

DS006976-13
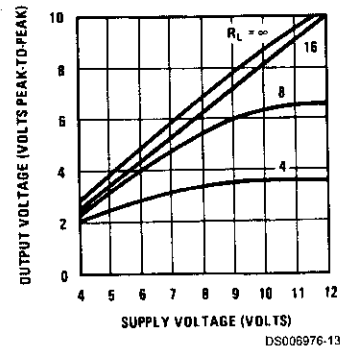
Voltage Gain vs Frequency

DS006976-14
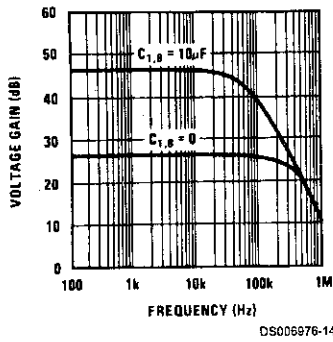
Distortion vs Frequency

DS006976-15
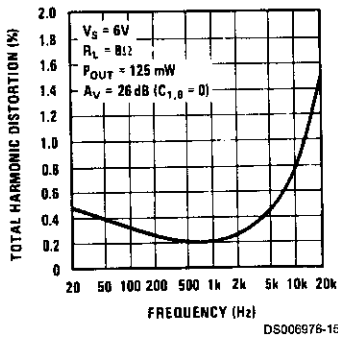
Distortion vs Output Power

DS006976-16
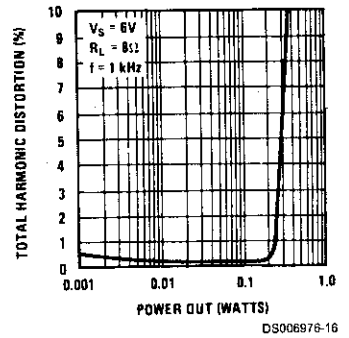
Device Dissipation vs Output Power—4Ω Load

DS006976-17
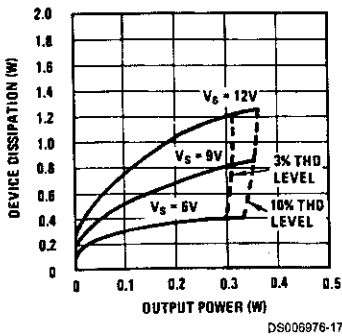
Device Dissipation vs Output Power—8Ω Load

DS006976-18
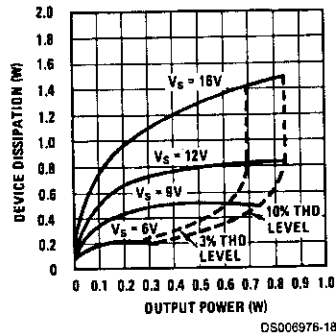
Device Dissipation vs Output Power—16Ω Load

DS006976-19
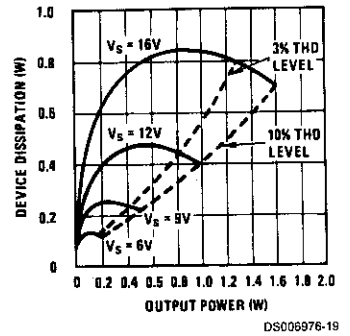
ıical Applications

Amplifier with Gain = 20 Minimum Parts
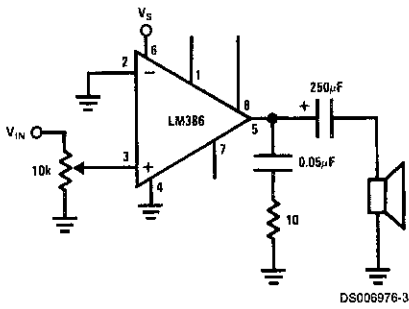
DS006976-3

Amplifier with Gain = 200
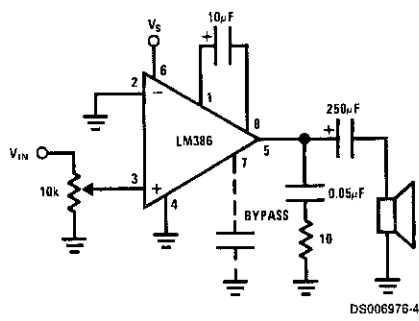
DS006976-4

Amplifier with Gain = 50
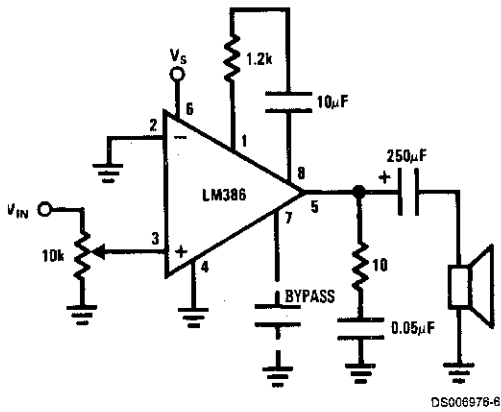
DS006976-6

Low Distortion Power Wienbridge Oscillator

f = 1 kHz
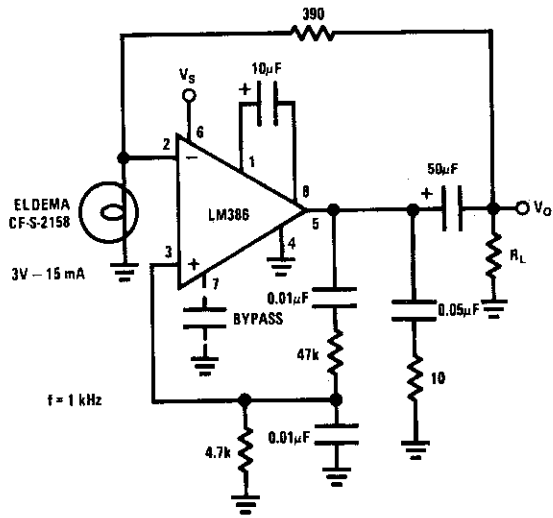
DS006976-7

Amplifier with Bass Boost
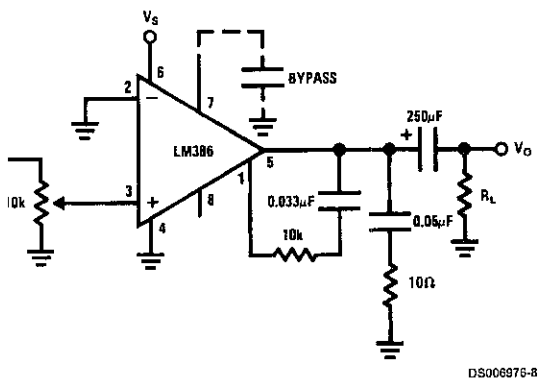
DS006976-8

Square Wave Oscillator

f = 1 kHz
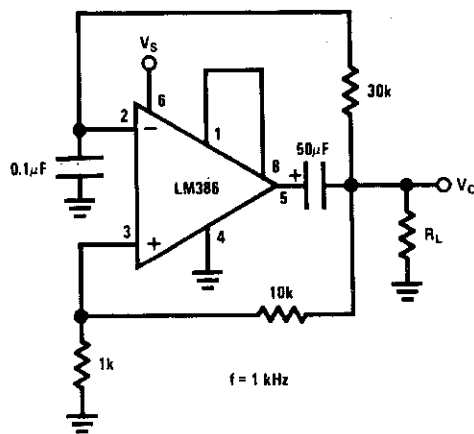
DS006976-9

Typical Applications (Continued)

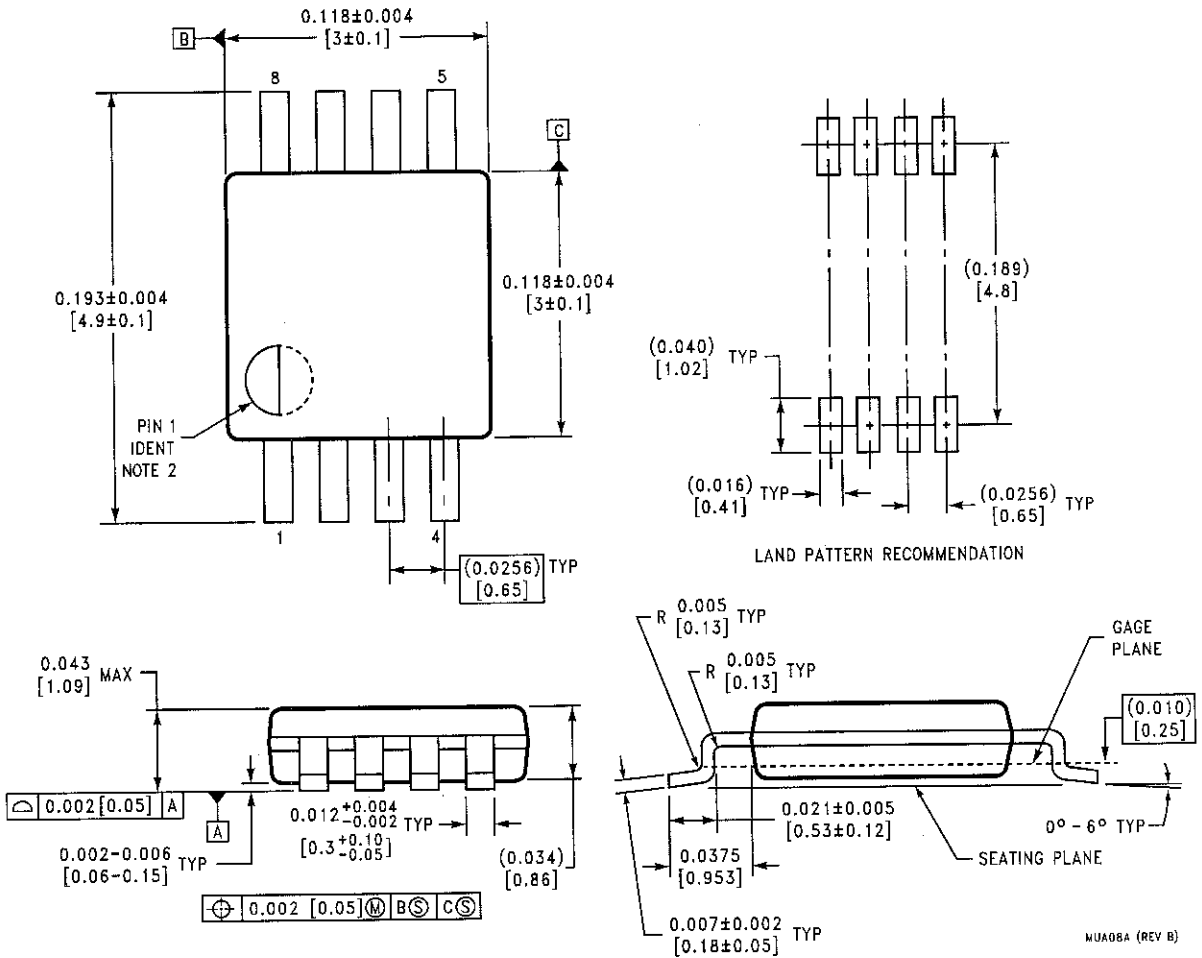## Frequency Response with Bass Boost
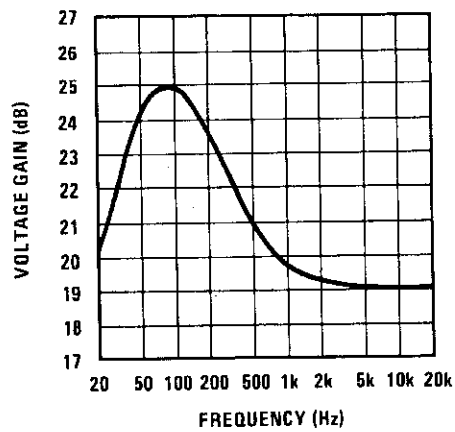
DS006976-10

AM Radio Power Amplifier
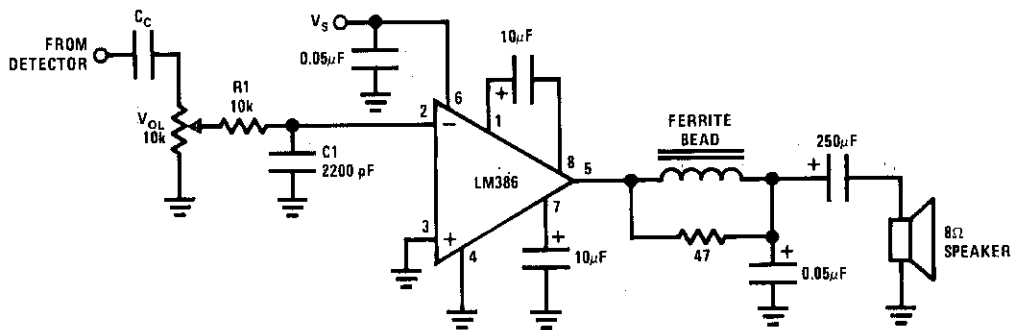
DS006976-11

Note 4: Twist Supply lead and supply ground very tightly.

Note 5: Twist speaker lead and ground very tightly.

Note 6: Ferrite bead in Ferroxcube K5-001-001/3B with 3 turns of wire.

Note 7: R1C1 band limits input signals.

Note 8: All components must be spaced very closely to IC.

ıSical Dimensions inches (millimeters) unless otherwise noted
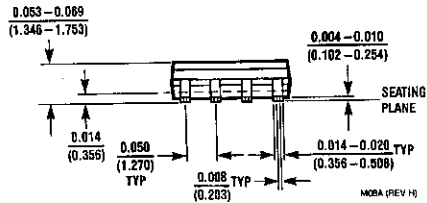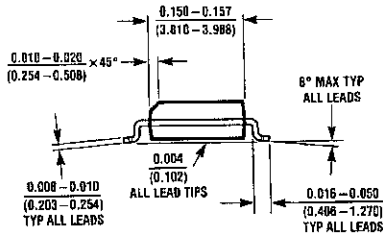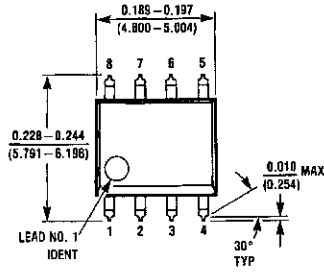
SO Package (M)
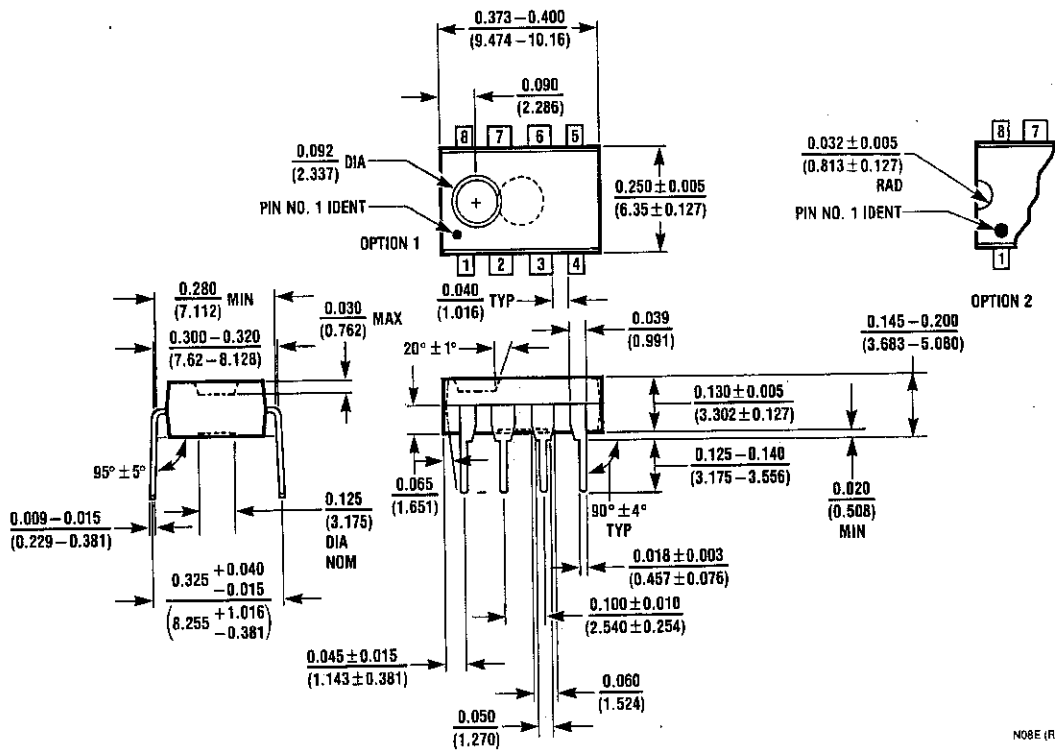**Order Number LM386M-1**
NS Package Number M08A

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)

0.118±0.004
[3±0.1]

B

8   5

C

0.193±0.004
[4.9±0.1]

0.118±0.004
[3±0.1]

PIN 1
IDENT
NOTE 2

1   4

(0.0256) TYP
[0.65]

LAND PATTERN RECOMMENDATION

(0.189)
[4.8]

(0.040) TYP
[1.02]

(0.016) TYP
[0.41]

(0.0256) TYP
[0.65]

0.043 MAX
[1.09]

0.002 [0.05] A

A

0.002-0.006 TYP
[0.06-0.15]

0.012 +0.004 -0.002 TYP
[0.3 +0.10 -0.05]

(0.034)
[0.86]

⊕ 0.002 [0.05] M B Ⓢ C Ⓢ

R 0.005 TYP
[0.13]

R 0.005 TYP
[0.13]

GAGE
PLANE

(0.010)
[0.25]

0.021±0.005
[0.53±0.12]

0° – 6° TYP

0.0375
[0.953]

SEATING PLANE

0.007±0.002 TYP
[0.18±0.05]

MUA08A (REV B)

8-Lead (0.118" Wide) Molded Mini Small Outline Package
Order Number LM386MM-1
NS Package Number MUA08A

ysical Dimensions inches (millimeters) unless otherwise noted (Continued)

0.373 – 0.400
(9.474 – 10.16)

0.090
(2.286)

0.092
(2.337) DIA

PIN NO. 1 IDENT

OPTION 1

0.250 ± 0.005
(6.35 ± 0.127)

0.032 ± 0.005
(0.813 ± 0.127)
RAD

PIN NO. 1 IDENT

OPTION 2

0.280
(7.112) MIN

0.030
(0.762) MAX

0.300 – 0.320
(7.62 – 8.128)

20° ± 1°

0.040
(1.016) TYP

0.039
(0.991)

0.145 – 0.200
(3.683 – 5.080)

0.130 ± 0.005
(3.302 ± 0.127)

95° ± 5°

0.009 – 0.015
(0.229 – 0.381)

0.125
(3.175)
DIA
NOM

0.065
(1.651)

0.125 – 0.140
(3.175 – 3.556)

90° ± 4°
TYP

0.020
(0.508)
MIN

0.018 ± 0.003
(0.457 ± 0.076)

0.325 +0.040 −0.015

(8.255 +1.016 −0.381)

0.100 ± 0.010
(2.540 ± 0.254)

0.045 ± 0.015
(1.143 ± 0.381)

0.060
(1.524)

0.050
(1.270)

N08E (REV F)

Dual-In-Line Package (N)
Order Number LM386N-1, LM386N-3 or LM386N-4
NS Package Number N08E