Sudan university for science and technology

College of Graduate Students Engineering

Using Matlab To Voice Analysis

إستخدام الماتلاب لتحليل الصوت

A project submitted in partial fulfillment the requirement of the degree of M.Sc in Electronic Engineering

Prepared by:

Safa Omer Mohammed

Supervised by:

Dr.Abd alrsol Jbar Alzibedi

2009

بسم الله الرحمن الرحيم

آية

قال تعالى:

اقْرَأْ بِاسْمِ رَبِّكَ الَّذِي خَلَقَ (١) خَلَقَ الْإِنسَانَ مِنْ عَلَقٍ (٢)
اقْرَأْ وَرَبُّكَ الْأَكْرَمُ (٣) الَّذِي عَلَّمَ بِالْقَلَمِ (٤) عَلَّمَ الْإِنسَانَ مَا لَمْ يَعْلَمْ (٥)

(صدق الله العظيم)

(سورة العلق الآية (١-٥))

Dedication

I dedicate this work

To:

One who gave me life

To:

My mothers and father who brought me up.

To:

My teachers, taught me a lot

To:

My husband and every one surround me with love.

To:

My friends.

Thanks

Acknowledgment

Thanks before and after to Allah for his giving me strength and power to complete this project.

I would like to express my great thanks to those who helped me to complete this thesis successfully:

I would like to thank my supervisor Dr.Abd alrsol Jbar Alzibedi for his guidance and supervision.

Thanks is also due to Eng. Alzahraa Abdelrahman for her help to complete this project.

Abstract

voice recognition is the process of inserting voice into computer and recording the voice files of many of the same sound through certain analysis and pre-known to recognize inserted voice.

There are several algorithms and mass diagrams concern with recognizing voices had been used by Matlab program which is currently being used in all fields of engineering, Matlab is a rich package of functions of digital signal processing and other mathematical functions.

This project aims at the analysis and voice recognition by MP3 and enter the voice records to matlab program and then analysis by the algorithm multiple attributes and try to identification with voice recorders after that taken a result of analysis to enter the circuit design because make operation control in open the sucker.

Which has been operating the sucker through the circuit design which connect with computer based on abort .

Finally which has been successfully compared to the sound input to the computer through the program designed for this research.

الخلاصة

تمييز الأصوات هو عملية إدخال الأصوات إلى الحاسوب وتسجيل ملفات عديدة لنفس الصوت ليقوم الحاسوب من خلال تحليلات معينة معروفة سلفا بتمييز الصوت الداخل.

هناك عدة خوارزميات ومخططات كتلية تعني بتمييز الأصوات قد تم استخدامها بواسطة برنامج الماتلاب الذي يستخدم حاليا في جميع حقول الهندسة، وهو من الحزم الغنية بدوال معالجة الإشارات الرقمية والدول الرياضية الأخرى.

يهدف هذا المشروع إلى تحليل وتمييز الأصوات من خلال أخذ معمارية محددة ومن ثم تحليلها بواسطة خوارزمية متعددة الصفات .

لقد قمنا في هذا البحث بتسجيل الصوت عبر الـ MP3 وإدخال الصوت المسجل إلى برنامج الماتلاب ليحلل الصوت المدخل.

أخذت نتيجة التحليل للدائرة المصممة لتقوم بالتحكم في تشغيل الـ Sucker.

قد تم بنجاح مقارنة الصوت المدخل إلى الحاسوب بواسطة البرنامج المصمم لهذا البحث.

Table of contents

Chapter Two
Types and Parameters of voice

Chapter Three
Matlab Program

Chapter Four
The hardware Design

Chapter One

1. Introduction:

1.1 Back ground :

1.1.1 Theoretical Back ground of Voice :

Voice have been used for over a century and are the most widely used form of biometric identification. Voice identification is commonly employed in forensic science to support criminal investigations, and in biometric systems .

Voice analysis is the study of speech sound for properties than other contain also study in cloud mostly medical of the voice analysis , proper of analysis and parameterization system.

1.1.2 Over view of Biometrics:

Biometrics to the automatic identification of a living person based on physiological of behavioral characteristics .

The method of biometric identification is preferred over traditional methods involving passwords and PIN numbers for various reasons:

- The person to be identified is required to be physically present at the point of identification .
- The identification based on biometric techniques obviates the need to remember a password or carry a token or a smartcard .
- What the rapid increase in use of PINs and passwords occurring as a result of the information technology revolution, it is necessary to restrict access to sensitive/ personal data.

Various types of biometric systems are being used for real-time identification; the most popular are based on face recognition and fingerprint matching. Furthermore , there are other biometric system that utilize iris and retinal scan, speech, face, and hand geometry.

1.1.3 Examples for Biometrics :

- Physical Biometrics that includes fingerprint, face recognition, voice recognition, hand geometry and iris patterns. Behavioral biometrics and it includes handwriting, signature, speech and Gait.
- Chemical/Biological Biometrics that includes perspiration, and skin composition (spectroscopy).

1.1.4 Physical Biometrics (voice) and the recognition process:

Voice (or vocalization) is the sound produced by humans and other vertebrates using the lugs and the vocal folds in the larynx, or voice box. Voice is not always produced as speech.

Voice is generated by airflow from the lunge as the vocal folds are brought close together. When air is pushed past the vocal folds with sufficient pressure, the vocal folds vibrate. If the vocal folds in the larynx did not vibrate normally, speech could only be produced as a whisper . Your voice is as unique as your fingerprint. It helps define your personality , mood and health.

Disorders of the voice involve problems with pitch, loudness, and quality. Pitch is the highness or lowness of a sound based on the frequency of the sound waves. Loudness is the perceived volume (or amplitude) of the sound , while quality refers to the character or distinctive attributes of a sound. Many people who have normal speaking skills have great difficulty communicating when their vocal apparatus fails.

Voice recognitions is the process of taking the spoken word as an input to a computer program. This process important to virtual reality simulation while allowing the user's hands to remain free. This article examine how voice recognition is accomplished, and list the academic disciplines that are central to the understanding and advancement of voice recognition technology.

Voice recognition is the technology by which sounds, words or phrases spoken by humans are converted into electrical signals, and these signals are transformed into coding patterns to which meaning has been assigned. While the concept could more generally be called " sound recognition", we focus here on the human voice because we most often and most naturally use our voices to communicate our ideas to other in our immediate surroundings. In the context of a virtual environment, the user would presumably gain the greatest feeling of immersion, or being part of the simulation, if they could use their most common form of communication, the voice .The difficulty in using voice as an input to a computer simulation lies in the fundamental differences between human speech and the more traditional forms of computer input. While computer programs are commonly designed to produce a precise and well – defined response upon receiving the proper (and equally precise) input, the human voice and spoken words are anything but precise. Each human voice is different, and identical words can have different meanings if spoken with different inflections or in different contexts. Several approaches have been tried, with varying degrees of success, to overcome these difficulties.

The most common approaches to voice recognition can be divided into three classes: "Template Matching", "Statistically Approaches" and "Artificial Intelligences Systems". Template matching is the simplest technique and has the highest accuracy when used properly, but it also suffers from the most limitations. As with any approach to voice recognition, the first step is for the user to enter the voice and then stored in ,memory in the system recognition data base. To determine the "meaning" of this voice input, the computer attempts to match the input with a digitized voice sample, or template, which has a known meaning.

This technique is a close analogy to the traditional command inputs from a keyboard. The program contains the input template, and attempts to match this template with the actual input using a simple conditional statement and if it found the recognition process is finished.

Since each person's voice is different, the program cannot possibly contain a template for each potential user if the number is huge so it used in small system because it need huge memory to store them more than the used hardware capacity .

The last approach is artificial intelligence systems; a neural network is a collection of layers of "neurons" simulating the human brain structure; Each neuron takes input from each neuron in the previous layer (or from the outside world, if it is in the first layer). Then, it adds this input up, and passes it to the next layer. Each connection between layers, however, has a certain weight. Every time the neural network processes some input, it adjusts these weights to make the output closer to a given desired value for the output. After several repetitions of this (each repetition is an iteration), the network can produce the correct output given a loose approximation of the input. We call this process training. Originally, a large neural network will be used for an entire utterance. Later, however, a modification for the structure to use a collection of networks each will be responsible for one section of a word (a frame). This is the over view and it will be discuss briefly later.

Another way to differentiate between voice recognition systems is by determining if they can handle only discrete words, connected words, or continuous speech. Most voice recognition systems are discrete word systems, and these are easiest to implement. For this type of system, the speaker must pause between words. This is fine for situations where the user is required to give only one word responses or commands, but is very unnatural for multiple word inputs. In a connected word voice recognition system, the user is allowed to speak in multiple word phrases, but he or she must still be careful to articulate each word and not slur the end of one word into the beginning of the next word. Totally natural, continuous, speech includes a great deal of "c-o articulation", where adjacent words run together without pauses or any other apparent division between words.

The template matching method of voice recognition is founded in the general principles of digital electronics and basic computer programming .

1.1.5 Applications of speaker Recognition:

Any task that involves interfacing with computer can potentially use a speech recognition model. The following applications are the most common right now

- **Dictation**

Dictation is the most common use the for a speech recognition systems today. This includes transcriptions, legal and business dictation, as well as general word processing. In some cases special vocabularies are used to increase the accuracy of the system.

- **Command and control**

Speech recognition systems that are designed to perform functions and actions on the system are defined as command and control systems. Utterances like "Open Netscape" will do just that.

- **Telephony**

Some PBX/voice mail systems allow callers to speak commands instead of pressing buttons to send specific tones.

1.2 Problem Statement :

.The password can be forgetting form the user and it easy to steal or cracked. Which has been consideration disadvantage in system.

Project to build tools to extract and parse speech attributes a speech signal, developed as part of the ASAT project , Automatic Speech attribute transcription

1.3 Proposed Solutions:

Using one of the biometric recognition " voice recognition" in system.

This method of identification is preferred over traditional methods involving password for various reasons:

The person to be identified is required to be physically present at the point – of – identification.

Identification based on biometric techniques obviates the need to remember a password.

1.4 Objective :

The primary aim of this research is to implement a series of techniques for voice image enhancement features extraction and is to develop AMA T LAB program that can be used to read the speech of the user from a - microphone, this experiments using real voice images are used to assess the performance of the implemented techniques.

1.5 Methodology:

Voice is a difficult signal to analysis because it varies so much time and is complex in nature this evidenced by the challenge scientists have been having in creating voice analysis software over decades.

In this research used certain algorithm in Matlab to make voice recognition process and signals analysis.

These techniques are then used to extract using from(as ample set of voice images, by using the extracted feature data, preliminary experiments on the stylistics of voice can then conducted.

There are two techniques for performing vocal frequency an Bio waves sound which contain tow main types of analysis health" Bin" or korg style analysis and Fourier trans form analysis.

1.6 Approach:

Voice recognition system for number of users that by entering their voices and make signal processing to extract.

The main goal that our voice system would be able to recognize the speakers' voices as the password by most as possible of the time, depending on each speaker voice fingerprint.

1.7 Expected Results:

The expected result of this research is a comparison between three different recognition classification algorithms and make six parameters which specialized to our voice recorders and take result of program to simulation of the circle and run the sucker.

1.8 Research outline:

This research is directed to have six chapters deal with the merging of speaker recognition system.

Chapter one is started by introduction of biometric especially the voice, then it mainlined the problem statement and proposed solution and it clarify the approach and expected results.

This dissertation is organized in to five main topics, with each chapter focusing on specific topic. Each chapter builds on the work discussed in earlier chapter . Chapter 2 describes the type and parameterization of voice analysis provides discussion in chapter 3 and mat lap programming chapter 2 and 3 are structured voice analysis processing in MATLAB typing each chapter contains literature review.

Chapter 4 present how to make Design of the crinite which suitable to voice analysis processing in MA TLAB.

Chapter 5 discussed the inter face to computer by the circuit

Chapter 6 contain conclusion and recommendation .

Chapter Two

Types and Parameters of voice

2.1 Introduction:

The signal is a group of frequencies that can be divided into two kinds, limited signal and unlimited signal with a bandwidth. The signal in general can be subdivided into image and audio signals.

- **Image signal:**

Modern digital technology has made it possible to manipulate multidimensional signals with systems that range from simple digital circuits to advanced parallel computers. The goal of this manipulation can be divided into three categories.

- Image processing image in –> image out.
- Image analysis image in - > measurements out.
- Image understanding image -> high- level description out.

To focus on the fundamental concepts of image processing, space does not permit to make more than a few introductory from remarks about image analysis. Image understanding requires an approach that differs fundamentally from the theme of Image processing and image analysis.

- **Audio signal:**

It is divided into voice signal and speech signal, which is more complicated than the voice.

2-2 Voice Frequency:

A voice frequency (VF) or voice band is one of the frequencies, within part of the audio range that is used for the transmission of speech.

In telephony, the usable voice frequency band ranges from approximately 300 Hz to 3400Hz. It is for this reason that the band of the electromagnetic spectrum between 300 and 300 Hz is also referred to as " Voice frequency" (despite the fact that this is electromagnetic energy, not acoustic energy).

The voice speech of a typical adult male will have a fundamental frequency of from 85 to 155 Hz, and that of a typical adult female form 165 to 255 Hz thus, the fundamental frequency of most speech falls below the bottom of the "voice frequency' band as defined above. [5]

2-3 Audio Format:

An audio format is a medium for storing sound and music. The term is applied to both the physical recording media and the recording formats of the audio content – in computer science it is often limited to the audio file format, but its wider use usually refers to the physical method used to store the data. Sound is recorded and distributed using a variety of audio formats, some of which store additional information. Here have used to wave signal. figure (2.1) explain Spectrum of Woman Voice.

Audio format different from women to man and machine and …ect.

Which different explain figures below :

2.4 Some examples of different audio format:

2.4.1 Woman audio format:

figure (2-1) explain Spectrum of Woman Voice to relation ship between time and amplitude.

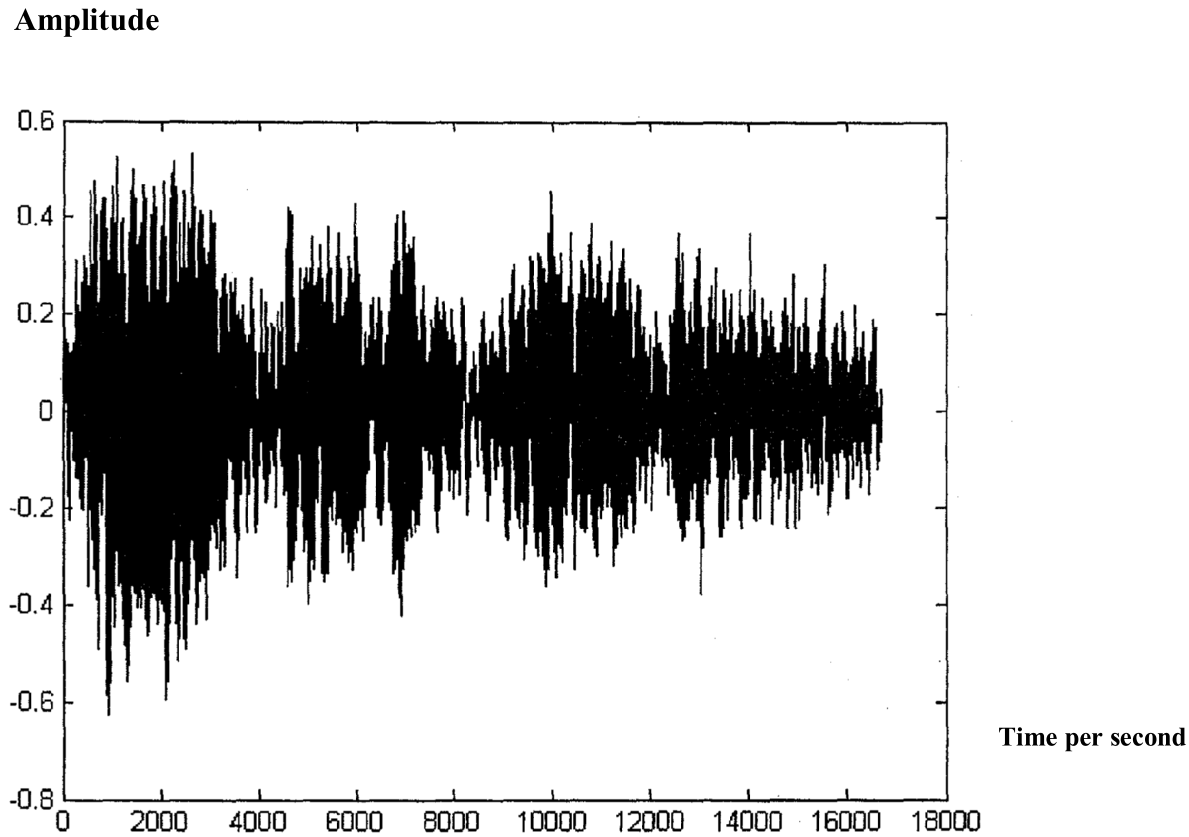


Figure (2-1) Spectrum of Woman Voice

2.4.2 Man audio format:

figure (2-2) explain Spectrum of man voice to relation ship between time and amplitude.

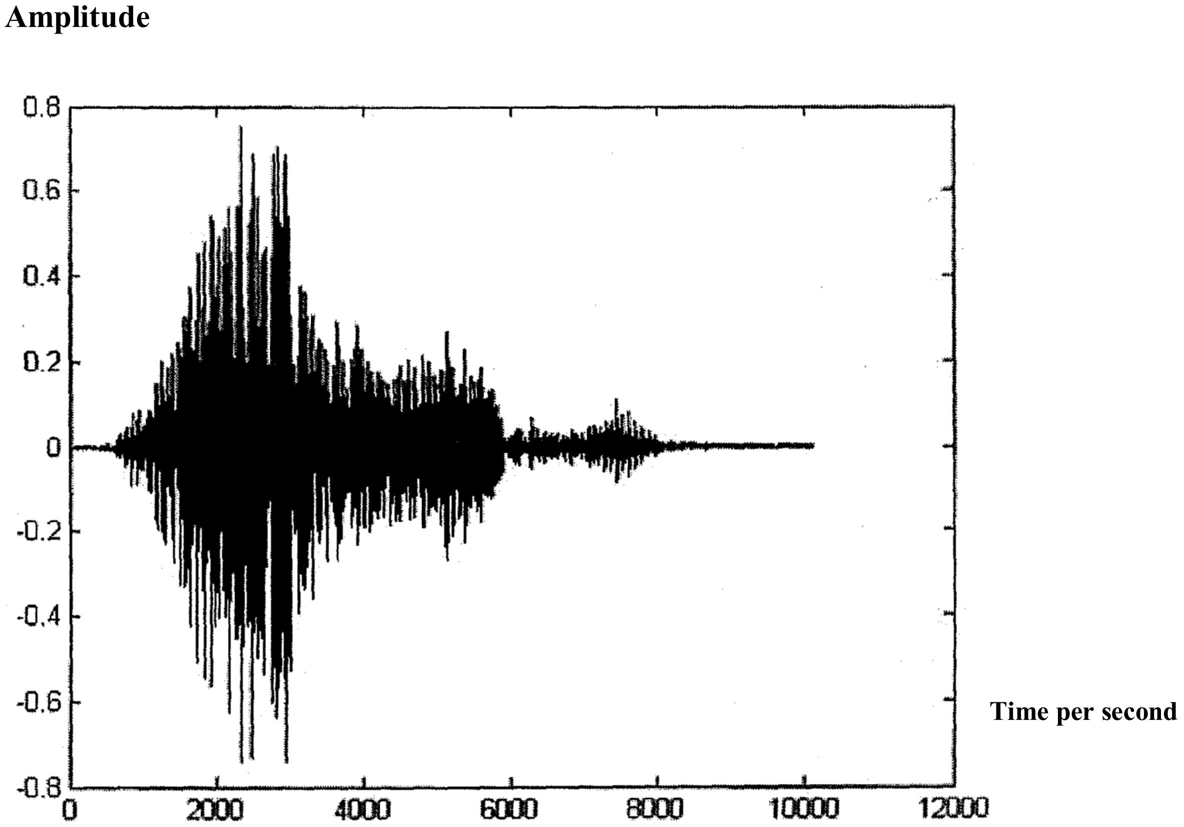


Figure (2-2) Spectrum of man voice

2.4.3 Car audio format:

figure (2-3) explain Spectrum of car voice to relation ship between time and amplitude.

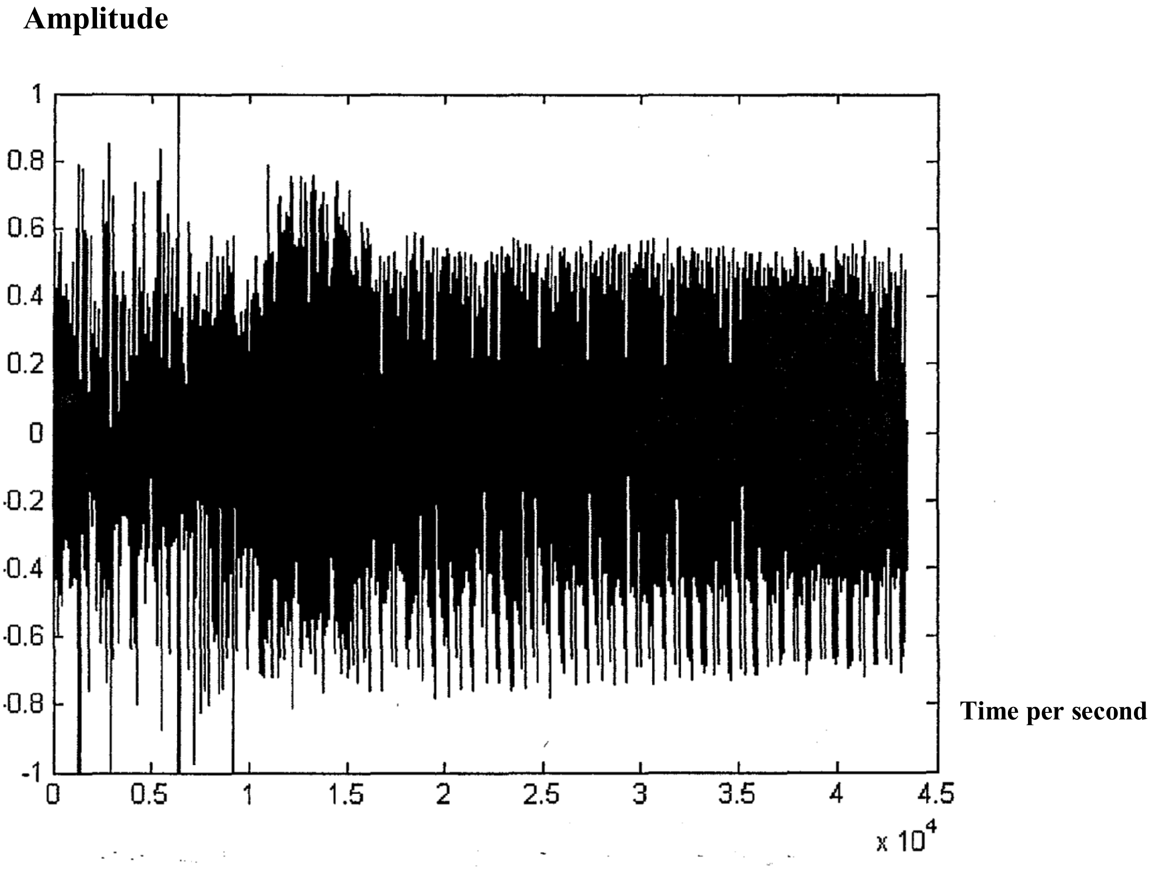


Figure (2-3) Spectrum of car voice

2.4.4 Cat audio format:

figure (2-4) explain Spectrum of cat voice to relation ship between time and amplitude.

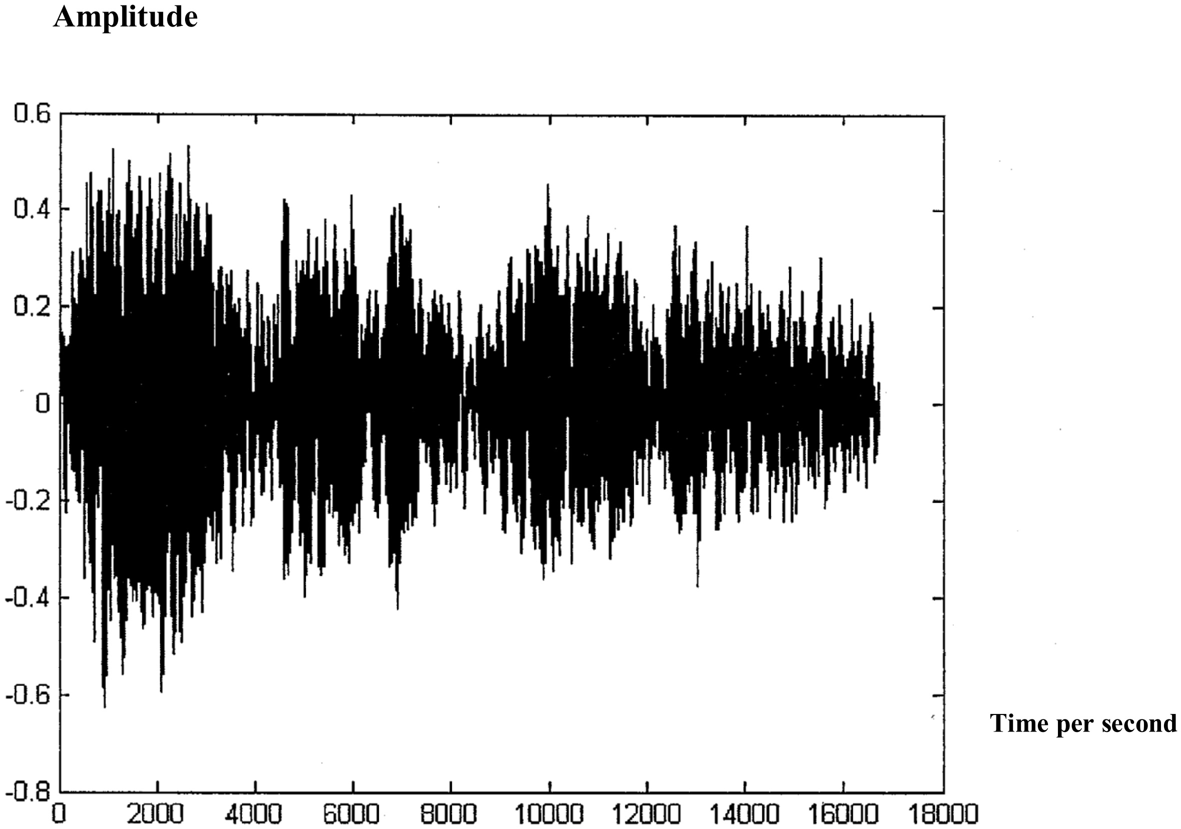


Figure (2-4) Spectrum of cat voice

2.4.5 Machine audio format:

figure (2-5) explain Spectrum of machine voice to relation ship between time and amplitude.

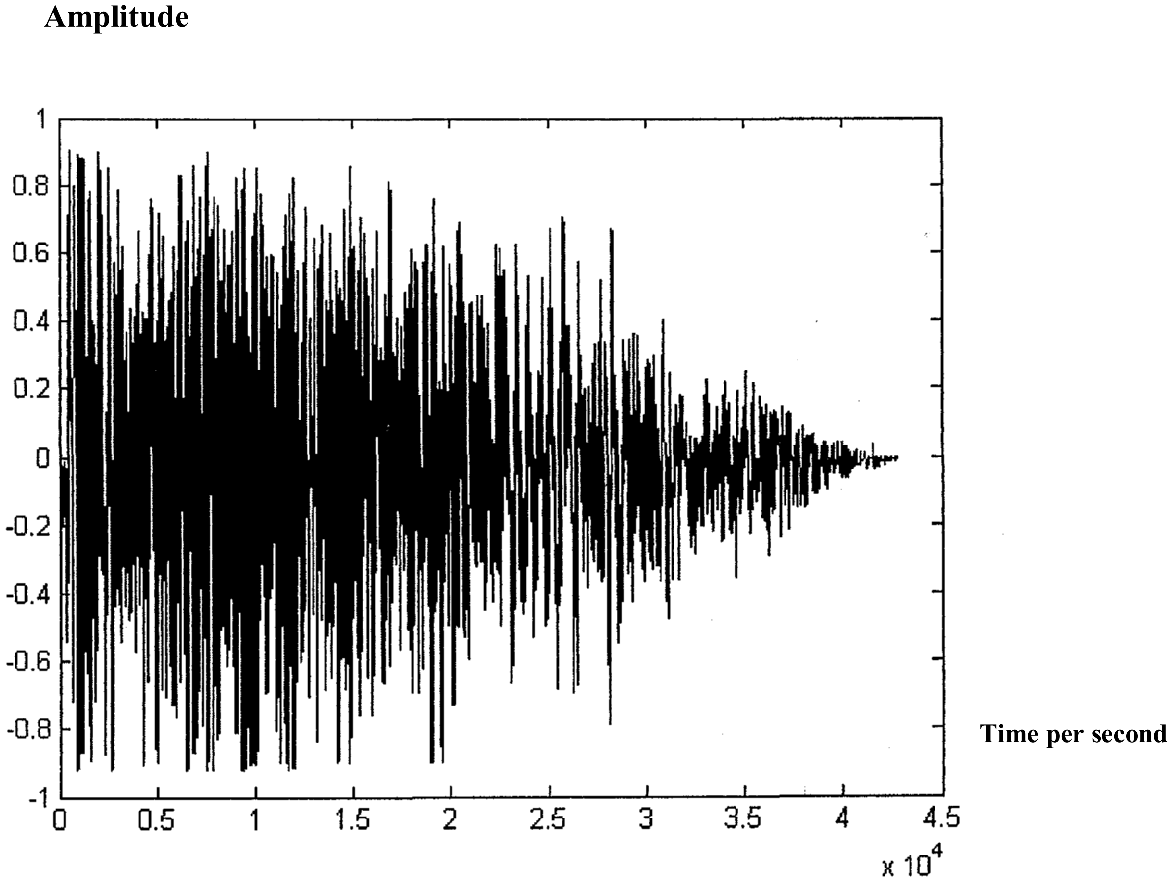


Figure (2-5) Spectrum of machine voice

2-5 Digital Signal Processing (DSP):

Digital signal processing (DSP) is concerned with the representation of the signals by a sequence of numbers or symbols and the processing of these signals.

Digital signal processing and analog signal processing are subfields of signal processing.

Since the goal of DSP is usually to measure or filter continuous real- world analog signals, the first step is usually to convert the signal from an analog to a digital form, by using an analog to digital converter.

Often, the required output signal is another analog output signal, which requires a digital to analog converter.

The signals are processed for compensation of many purposes:

- Filtering.
- Signal separation.
- Sigal restoration.
- Coding.
- Fusion of signals, and so on.

Digital signal processing will used in many field:

- Communication.
- Science.
- Control.
- Speech production and recognition.
- Medical.
- Manufacture.
- Military field, and so on.

2-6 Transformers:

The digital signal processing includes much type of transformers which mainly use to make the processing form simple. Transormers are defined as mathematical techniques used make the digital signal processing treatment easy.

One of the famous transformers is fourier transformed the signal from “time domain” to “frequency domain” to make the processing easier because the signal in frequency domain is easier, and it is equivalent to Laplace transformer that sed in analog system. Z-transformer expressed for signal.

x(n) is:

$$x(z) = \sum_{n=-oo}^{n=oo} \mathrm{x(n)z} - a^{z-n} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2.1)$$

Where Z is complex number and the frequency response is:

$$H(z)^{n} = \sum_{n=\infty}^{n=\infty} \mathrm{H(n)Z}^{-n} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2.2)$$

After the Knowledge of the Z-transormer for any input signal to specific system with the knowledge frequency response, so the Z-transormer of any output signal is:

Y = X(z)H(z)

2-7 Discrete fourier Transformer (DFT):

The discrete Fourier transform is one of the specific forms of Fourier analysis, The DFT is the type used in digital signal processing technique because the only type of signal work with computer is the discrete signal with infinite length.

If we had signal x(n) the DFT for this signal is:

$$x(f) = \sum_{n=0}^{N-1} x(n)e - j\left(\frac{2\pi}{N}\right)kn : f = 0,1 \ldots N-1 \qquad \ldots\ldots\ldots\ldots\ldots(2.3)$$

2-8 Digital Filters:

A digital filter is any electronic filter that works by performing digital mathematical operations on an intermediate form of a signal. This is in contrast to older analog filters which work entirely in the analog realm and must rely on physical networks of electronic components (such as resistors, capacitors, transistors, etc.) to achieve the desired filtering effect. Digital filters can achieve rirtually any filtering effect that can beexpressed as a mathematical function or algorithm. The two primary limitations of digital filters are their speed (the filter can't operate any faster than the computer at the heart of the filter), and their cost. However as the cost of integrated circuits has continued to drop over time, digital filters have become increasingly commonplace and are now an essential element of many every day objects such as radios, cell phones, and stereo receivers.

The digital filters use in two basics purpose:

- Signal separation**.**
- Signal restoration.

In generally, they were four types of filter:

2-9 Low- pass filter:

A low0pass filter is a filter that passes low frequency signals but attenuates (reduces the amplitude of) signals with frequencies higher than the cutoff frequency. The actual amount of attenuation for each frequency varies from filter to filter. It is sometimes called a high-cut filter, or treble cut filter when used in audio applications. The concept of a low-pass filter exists in many different forms, including electronic circuits, digital algorithms for smoothing sets of data, acoustic barriers, and so on. Low-pass filters play the same role in signal processing that moving averages do in some other fields, such as finance; both tools provide a smoother form of a signal which removes the short-term oscillations, leaving only the long-term trend.

A stiff physical barrier tends to reflect higher sound frequencies, and so acts as a low-pass filter for transmitting sound. When music is playing in another room, the low notes are easily heard, while the high notes are attenuated

2-10 High pass filter:

A high-pass filter is a filter that passes high frequencies well, but attenuates (reduces the amplitude of) frequencies lower than the cutoff frequency. It is sometimes called a low-cut filter; the terms bass-cut filter or rumble filter are also used in audio applications but speech requires a relatively low bandwidth these mean using low pass filter in audio application.

2-11 Band pass filter:

It is used for the passage of certain range of frequencies.

2.12 Band stop filter:

It is used to stop passage of certain range of frequencies while passage remains frequencies.

2-13 Voice type

Voice type is a particular kind of human singing voice perceived as having certain identifying qualities or characteristics. **Voice classification** is the process by which human voices are evaluated and are thereby designated into **voice types**. These qualities include but are not limited to: vocal range, vocal weight, vocal tessitura, vocal timbre, and vocal transition points such as breaks and lifts within the voice. Other considerations are physical characteristics, speech level, scientific testing, and vocal registration.[1] The science behind voice classification developed within European classical music and has been slow in adapting to more modern forms of singing. Voice classification is often used within opera to associate possible roles with potential voices. There are currently several different systems in use including: the German Fach system and the choral music system among many others. No system is universally applied or accepted.[2].

Voice classification is a tool for singers, composers, venues, and listeners to categorize vocal properties, and to associate possible roles with potential voices. There have been times when voice classification systems have been used too rigidly, i.e. a house assigning a singer to a specific type, and only casting him or her in roles they consider belonging to this category.

2-14 Voice types and the folds (cords) themselves

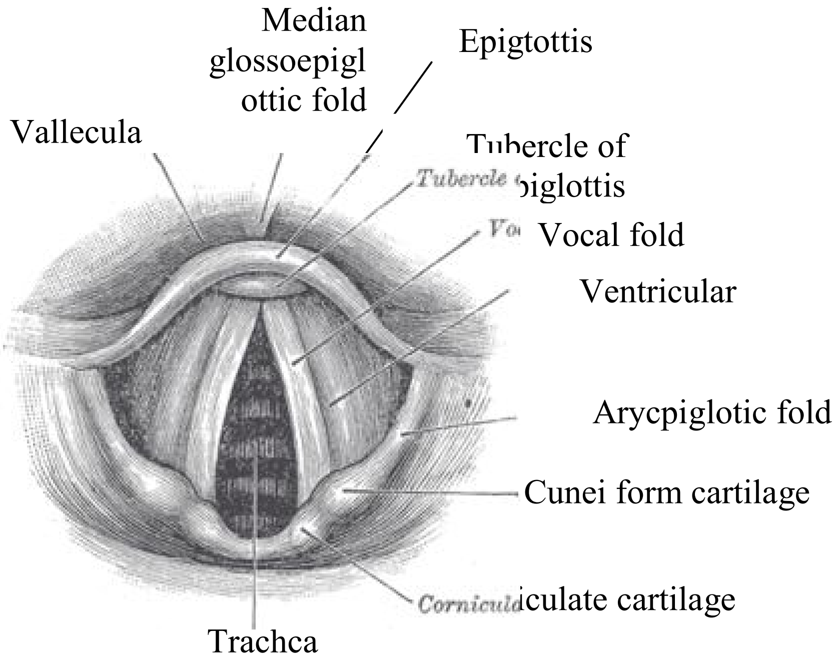


Fig (2-6) diagram of the vocal folds or cords

Men and women have different vocal folds sizes; adult male voices are usually lower-pitched and have larger folds. The male vocal folds (which would be measured vertically in the opposite diagram), are between 17 mm and 25 mm in length The female vocal folds are between 12.5 mm and 17.5 mm in length. [5]

- As seen in the illustration, the folds are located just above the trachea (the windpipe which travels from the lungs). Food and drink do not pass through the cords but instead pass through the esophagus, an unlinked tube. Both tubes are separated by the epiglottis, a "flap" that covers the opening of the trachea while swallowing.
- The folds in both sexes are within the. They are attached at the back (side nearest the spinal cord) to the arytenoid cartilages, and

at the front (side under the chin) to the thyroid cartilage. They have no outer edge as they blend into the side of the breathing tube (the illustration is out of date and does not show this well) while their inner edges or "margins" are free to vibrate (the hole). They have a three layer construction of an epithelium, vocal ligament, then muscle (vocalis muscle), which can shorten and bulge the folds. They are flat triangular bands and are pearly white in color. Above both sides of the vocal cord is the vestibular fold or false vocal cord, which has a small sac between its two folds (not illustrated).

- The difference in vocal folds size between men and women means that they have differently pitched voices. Additionally, genetics also causes variances amongst the same sex, with men and women's singing voices being categorized into types. For example, among men, there are basses, baritones and tenors, and among women, contraltos, mezzo-sopranos and sopranos. There are additional categories for operatic voices, see voice type. This is not the only source of difference between male and female voice. Men, generally speaking, have a larger vocal tract, which essentially gives the resultant voice a lower tonal quality. This is mostly independent of the vocal folds themselves.

Physiology and vocal timbre

- The sound of each individual's voice is entirely unique not only because of the actual shape and size of an individual's vocal cords but also due to the size and shape of the rest of that person's body. Humans have vocal folds which can loosen, tighten, or change their thickness, and over which breath can be transferred at varying pressures. The shape of chest and neck, the position of the tongue, and the tightness of otherwise unrelated muscles can be altered. Any one of these actions results in a change in pitch, volume, timbre, or tone of the sound produced. Sound also resonates within different parts of the body, and an individual's size and bone structure can affect the sound produced by an individual.

- Singers can also learn to project sound in certain ways so that it resonates better within their vocal tract. This is known as vocal resonation. Another major influence on vocal sound and production is the function of the larynx which people can manipulate in different ways to produce different sounds. These different kinds of laryngeal function are described as different kinds of vocal registers. The primary method for singers to accomplish this is through the use of the Singer's Formant, which has been shown to match particularly well to the most sensitive part of the ear's frequency range.

2-15 Voice parameterization:

Parameterization of an analog speech signal is the first step in the speech recognition process. Several popular signal analysis techniques have emerged as defacto standards in the literature. These algorithms are intended to produce " a perceptually meaning full" parametric representation of the speech signal (parameters that emulate some of the behavior observed in the human auditory and perceptual system). Of course, and perhaps these algorithms are also designed to maximize recognition performance.

On the other hand, modern techniques use digital processing or speech signals. So as to understand the various ways in which the basic models and the associated parameters derived from them, are used in an integrated system whose purpose is to transmit or to automatically extract information from the speech signal, it is important to made an attempt to survey the entire field of speech signal, it is important to made an attempt to survey the entire field of speech communication for application, but specifically, selected examples related to man machine communication by voice. There are several reasons for restricting the attention to man-machine communication.

Chapter Three

Matlab Program

3.1 Introduction

The name MATLAB stands for Matrix Laboratory. Matlab was written originally to provide easy access to matrix software developed by the LINPACK (linear system package) and Eispack (Eigen system package) projects.

MATLAB is a high-performance language for technical computing. It computation visualization and programming environment . Furthermore.. , MA TLAB is a modern programming language environment: it has sophisticateddata structures, contains built-in editing and debugging tools, and supports object oriented programming(OOP). These factors make MATLAB an excellent tool for teaching and research.

MATLAB has many advantages compared to conventional computer languages (e.g., C, FORTRAN) for solving technical problems MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. The software package has been commercially available since 1984 and is now considered as a standard tool at most universities and industries worldwide.

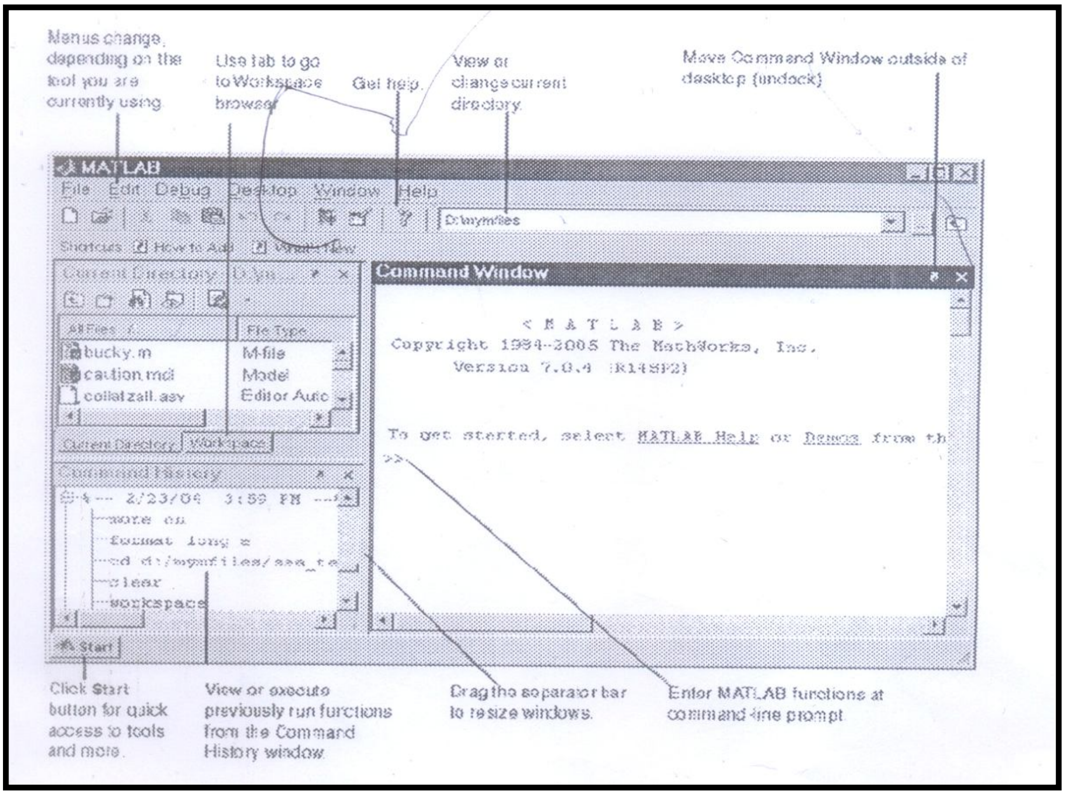
It has powerful built-in routines that enable a very wide variety of computations. It also has easy to use graphics Commands that make the visualization of results immediately available. Specific applications are collected in packages referred to as toolbox. There are toolboxes for signal processing, symbolic computation, control theory, simulation optimization and several other fields of applied science and engineering.

MATLAB is a powerful high – level programming language for scientific computer. It is very easy to learn and in solving numerically complex engineering problems. However , some basic concept of MATLB are included here for a quick review to facilitate your understand of the programs and for performing the exercises.

MATLB Consists of functions that are either built into the interpreter or available as M- files , with each containing a sequence of program statement that execute that a certain algorithm A compete new algorithm can be written as a program containing only a few of these functions and can be saved as another M- file .

MATLAB works with three types of windows on you computer screen These are the Command window, the figure window and the Editor window . The Command window has the heading Command. the Figure window and the Editor window has the heading showing figure No.1, and the Editor window has the heading Command , the showing name f an open existing M-file or Untitled if it is a new M-file under construction of an window also show the prompt >> indicating is ready to execute MATLAB Commands. Result of most printing Commands are displayed in the Command window. This window can also be used to run small programs and saved M-files. All plots generated by the plotting Commands appear in a figure window Editor new M-file or old M-files are run form the Command . Existing M-files can also be run from the Commands window by typing the name of the file.

In the remaining part of this appendix have illustrated the use of some of the most commonly used functions and review some fundamental concepts associated with MALTAB .



Menus change, depending on the tool you are currently using

Use tab to go to Workspace browser

Get help.

View or change current directory.

Move Command Window outside of desktop (undock)

```
< M A T L A B >
Copyright 1984-2005 The MathWorks, Inc.
      Version 7.0.4 (R14SP2)

To get started, select MATLAB Help or Demos from th
>>
```

```
%-- 2/23/04  3:59 PM --%
  more on
  format long e
  cd d:/mymfiles/ssa_te
  clear
  workspace
```

Click Start button for quick access to tools and more.

View or execute previously run functions from the Command History window

Drag the separator bar to resize windows.

Enter MATLAB functions at command-line prompt

MATLAB
File Edit Debug Desktop Window Help
D:\mymfiles
Shortcuts How to Add What's New
Current Directory
All Files
File Type
bucky.m M-file
caution.mdl Model
collatzall.asv Editor Auto
Current Directory Workspace
Command History
Command Window
Start

Fig (3.1) The graphical interface to the MA TLAB workspace

When MA TLAB is started for the first time, the screen looks like the one that shown in last Figure. This illustration also shows the default configuration of the MA TLAB desktop. You can customize the arrangement of tools and documents to suit your needs.

Now, we are interested ~ in doing some simple calculations. We will assume that you have sufficient understanding of your computer under which MA TLAB is being run.

You are now faced with the MATLAB desktop on your computer, which contains the prompt (>>) in the Command Window. Usually, there are 2 types of prompt:

>> for full version

EDU > for educational version

3.2 Using MA TLAB as a calculator

let's suppose you want to calculate the expression, 1 + 2 * 3. You type it at the prompt Command (») as follows,

```
> > 1+2*3
ans =           7
```

3.3 A minimum MA TLAB session

The goal of this minimum session is to learn t,he first steps:

- How to log on
- Invoke MATLAB
- Do a few simple calculations.
- How to quit MATLAB.

3.4 Starting MA TLAB

After logging into your account, you can enter MA TLAB by double-clicking on the MATLAB shortcut icon (MATLAB 7.0.4 or other newer) on your Windows desktop. When you start MA TLAB, a special window called the MATLAB desktop appears. The desktop is a window that contains other windows. The major tools within or accessible from the desktop are:

- The Command Window
- The Command History .
- The Workspace.
- The Current Directory
- The Help Browser
- The Start button

3.5 Number and Data Representation

MATLAB users conventional decimal notations to represent wit a leading minus sign for negative numbers . The approximate range of numbers that can be represented is from 10^{-305} to 10^{305} . Very large of very small numbers can be represented using exponents .Typreal examples of vahd number representations are as follows:

124.56789   123456.789E-2   1.23456789e3   -1234.56789

There should be not blank space before the exponent .The data in MATLAB are represented in the form of a rectangular matrix does not.

Require dimensioning. lts elements can be either real or complex numbers. Thus, a one dimensional discrete-time signal can be represented either as a row or a column vector. For example the row vector index MATLAB! data representation in

```
x = [35 + 4 * j-2.1-7.4*j   1.05 - 0.8 * j 0 9.2 * j];
```

denotes a complex-valued signal x of length 5. Note the use of square brackets to indicate that x is a rectangular matrix. Note also that the imaginary part of a complex number is represented using the operator * and the letter j. An alternate form of representation of the imaginary part uses the letter i instead of the letter j. The real and imaginary parts of a complex number should be2 entered without any blank spaces on cither side of the + or - sign as indicated above. The elements in the row of a matrix can also be separated by commas as indicated below:

```
x=[35+4*j,-2.1 - 7.4. j,1.05 - O.R , j,0,9.2. J];
```

The semicolon; at the end of the square brackets ensures that the data arc not printed in the command window after they have been entered. If the above data were entered without the semicolon, MATLAB would print in the Command window

```
X=

Columns 1 through 4

3 5000 + 4.0000i   -2.1000 - 7.40001            1.0500 - 0.8000i

Column 5

0 + 9.2000i
```

Alternately, if needed, the actual value of x can be printed by typing x in the Command window.

The elements of a matrix can be entered in two different ways. The rows can be typed on a single line separated with semicolons or on different lines. Fur example, the 3" 4 matrix A

```
A = [ 3 4  11

          5 6  13

          1 5  ]
```

can be entered either as

```
A = [11   3   5    7;   2    4  6  8;  9  11 13 15 ];
```

or as

```
     A=
```

The indexing of vectors and matrices in MATLAB begins with 1. For example, x(1). in the above vector x is 3.5000 + 4.0000i,x(2) is-2.1000-7.4000i, and so on. Similarly, the first element in the first row of a matrix A is given by A (1 ,1), the second clement In the first row is given by A (1.2), and so on. The index cannot be less than 1 or greater than the dimension of the vector or matrix under consideration .

The size of an array in the workspace of MATLAB can be determined by using the function size . For example . by typing size (x) we obtain the result .

```
ans =

1              5
```

The array transpose operator . Thus the transpose of X is given by the expression X ∴If X is a matrix with complex – valued elements , X' is the complex conjugate transpose of X , whereas if X is a matrix with real – elements X' is the transpose of X.

3.6 Relational Operators

The relational operations in MATLAB <, <=, >, >=, ==, and^{-}=, operations less than or equal to (≤) , greater than , grater equal (≤) , equal to, and not equal to (≠) respectively, Element-by-element comparisons between two matrices of the same size carried out using these operators with the result appearing as a matrix or the same size whose elements are set to 1 when the relation is TRUE and set to when the relation is FALSE, In the case or complex- valued matrices, the operators s <, <=, >, and > = arc applied compare only the real parts of each element of the matrices . whereas the operators = = and^{-} = are applied to compare both real and imaginary parts.

3-7 Logical Operators

The three logical operators in MATLB. & 1.and 2 perrim the logical AND, OR and NOT operations. When applied to matrices. they operate element wise, with FALSE represented by a O and TRUE represented by a1.

3.8 Control Flow:

The control flow Command of MATLAB are break . else. else if, end, error, for, return , and while. These Commands permit the conditional execution of the creating program statements. The command for is used to repeat a group of program statements a specific number of times . The command if is used to execute a group of program statement conditionally and the Command while can be used to repeat program statements an indefinite number of times. The Statements following the commands for, while, and if must be terminated with the command end. The command break used to terminate the execution of a loop . The commands else and elseif are is used with the Command if to provide inside a loop.The command error is employed to display error massage and abort functions.

3.9 Special Characters and Variables:

MATLAB use a number of special characters and words to donate certain items exclusively. These Characters and words should not be used for any other purpose . For example, if the letter i or the letter j is used as a variable, it cannot be used to represent the imaginary part of complex number. Hence, it is a good practice to restrict either the letter i or the letter j exclusively for the representation of the imaginary parts of complex numbers.

There are several permanent variables that cannot be cleared by the user and should not be used to denote any other quantities. The word ….. is used denote Thus, sin(pi/4)yields 0.707,10678118655, which is equal to 2 The veritable eps is equal to 2^{-52} and is a tolerance for determining precision of certain computations such as the rank of a matrix. It can be set to any other value by the user. NaN represents Not –a number, which is obtained when computing mathematically undefined operation such as 0/0and and ∞-∞. In represents + ∞ and rustles from operators such as dividing by zero. for example, 2/0, or form overflow, for example e ^{1000}. The variable can stores the results of the most recent operation.

The square brackets [ ] are used to enter matrices and vectors. The elements or a matrix can be separated by Spaces or commas. A semicolon , indicates end of a row in a matrix, It is also used to suppress printing .The precedence in arithmetic expressions can be indicated using the parentheses ( ) The parentheses are also employed to enclose the indices of an array and arguments of functions. The operator for transpose of an array is . However, two such symbol can be used to denote a quote. For example, 'dsp program' is a vector containing the ASCII codes of the characters inside the quotation marks. Any text following a percent symbol % denotes a command and is not treated as a program statement.

The colon symbol : has many different application in MATLAB . It is used to generate vectors. Subscript matrices and perform iterations of a block of commands.

3.10 Output Data Format:

All arithmetic operations In MATLAB are performed in double precision. However, different formats can be used to display the result of such operations in the Command window. If all results are exact integers. they are displayed as such without any decimal points. If one or more data elements are not integers, the results can be displayed with various precision. format short displays live significant decimal digits and is the default format. Format short e displays live significant decimal digits with two positive or negative decimal exponents: format long shows results in 15 significant decimal digits. while format long e adds two positive or negative decimal exponents to 15 significant decimal digits. There arc three other formats for displaying results. However, these arc not that useful in signal processing applications.

3.11 Graphics:

MATLAB includes high-level graphics capability for displaying the results of a computation. In most situations, should be concerned with two-dimensional (2-D) graphics and will use three-dimensional (3-D) graphics in some special cases. For 2-D graphics,. plotting can he dune In various forms with either linear or logarithmic scales for one or both axes of the plots. Grid lines can he added to the plots along with labels for the two axes and a title on top of the plot. Text can he placed anywhere on the graph using a mouse or specifying the starting position through commands in the program. Moreover. by including appropriate symbols in the argument of the plotting Command. specific line styles. plot symbols, and colors had displayed in the graph.

For 3-D data. plotting can also had done in various forms with either linear or logarithmic scales for one or two or all three axes of the plots. For example. line, and points can be plotted in three dimensions. Contour plots. 3-D perspective plots. surface plots. pseudo color plots. and so forth also had generated.

The M-lile in the following section illustrates the use of several graphics commands.

3.12 M-Files: Scripts and Functions:

An M-file is a sequence of MATLAB statements developed using a word processor or a text editor and saved with a name that must be in the form file name. m. The names of M flies must begin with" letter followed by at most 18 letters and/or digits (or underscores).

However certain characters. such as hyphen – and decimal .are not in the names .Also, do not use the name of existing M- files .An M- file can include references to other existing M- files .

Each statement of a new program is typed in the Editor window line by line as ASCII text file and can be edited using the text editor or the word processor of your computer program can then be saved as an M-file .

There are two type of M- files . scripts and junmbrs . A function file must contain the first line of all program statements .Arguments in a function file may be passed from anther M- file and all variables used inside the function file are local.

The script file makes use of workspace data globally . The first line of a function file contains the word function and does not make use of workspace data globally. The variables defined inside a function file are manipulated locally, and the arguments of the file may be passed . When a function is completed, all local variables are lost. Only values specifically passed out are retained.

A simple example of a function file runsu = is given below.

```
Function y  = run sum  = (x)

% Computers the mean of a vector x

L =  length (x);

Y = sun (x) /L;
```

3.13 MAT-Files

Data generated by a MATLAB program can be saved as a binary file. called MATLAB -file. for later use. For example. the noisy data generated by executing the program lowpass . m of the precious section can be saved using the command.

```
save noise – mat data
```

Later it can be retrieved using the Command

```
load noise
```

for use in another MATl.A13 session.

The data can also be saved in ASCII form. For example. after execution of the program lowpass m, we can generate a 2 x II matrix containing the noisy data and the smoothed data using the Command

```
result = [noise; z];
```

and then save the matrix result in , ASCII form using the command

```
save tempo. dat  result -ascii
```

The stored data can later be retrieved using the command

```
load tempo
```

3.14 Printing:

To develop a hardcopy version of the current Figure window , the command print can be used . There are many versions of the command . See the MATLAB Reference Guide [Mat 94]. For details . In a PC or a mae environment figure can be also be copied to the clipboard and then inserted into a word processing exhumer . This approach permits generating smaller size figure and also pasting the figure on to a text.

3.15 Diagnostics and Help Facility:

MATLAB has very good diagnostic capabilities, maxing it easier to correct any errors detected during execution. If any program statement has errors . the execution of the program will stop with a self-evident error message appertain in the command window, For example, entering the real number 1, 23';55789e3 with exponent will result in the error massage.

?77 1 23456739 e3

Missing operator. Comma, same colon

Entering the real number 1, 2345678ge3 with action in place of the decimal as 1: ~ : l' :23456789e3 will cause the error message.

?? Error using = = > colon

MATLAB provide online information for most topics through the command help. If help is typed in the command window with no arguments. a list of directories containing .MATLAB files are displayed in the window . For help on specific M- files or directories, the name of the file or the directory should be used as an argument for example typing help run sum results in

Computes the mean of a vector , x

Likewise, typing help lowpass yields

A Script M-File to Perform Lowpass Filtering

Using Three Point Averaging

Program uses the function file run sum

A list of variables in the workspace can he obtained by typing who. To obtain information about the size of the variables, use the Command whos. Other useful Commands are what, which, look for, echo, and pause.

The Command what lists all files in the current directory, whereas the Command that dryname lists the files in the directory named dryname on MATLAB search path. The Command which is used to locate functions and files on MATLAB's search path. The Command lookfor abc searches through all help entries on MATLAB's search path and looks for the string abc in the first comment line. The Command echo is useful for debugging a new program and is used to list all M-files being invoked during the execution of a program. There are several versions of this Command. See the MATLAR Rderence Guide [Mat^{94}] to determine the appropriate ones for you to use. The Command pause stops program execution temporarily at the point it is invoked; the execution can be restarted at that point by pressing any key on the keyboard. This command is particularly useful when the program is generating a large number of plots and each plot can he examined or copied individually if the Command pause is inserted after each plotting Command.

3.16 Remarks:

Even though MATLAB uses double precision arithmetic, numerical approximation used in the computations may generate errors the results . Care must be taken in such cases to interpret the results correctly. As an example, the computation of the expression 1– 0.1– 0.3– 0.2 -0.2 -0.1 -0.1 yields .5.551115123125783e – 17 in the out put format long when the result should have been ideally equal to 0 .On the other hand, a slight change in expression to 1 – (0.1+0.3+.02 +0.2 + 0.1 + 0.1) yields the correct results 0 .

Chapter Four

The hardware Design

4.1 Introduction:

Motion Control, in electric terms, means to accurately control the movement of an object based on speed, load, inertia or a combination of all these factors. There are numerous types of motion control systems, including: Sucker Brush, Branchless, Servo, Brushless Servo and more.[2]

4.2 Hardware Contains:

- Sucker.
- HD 74LS373ICs
- UlN 2803A.
- Board Panel.
- Operation Circuit Design.

4.3 Sucker:

These Suckers are also called stepping motors or step motors. The name stepper is used because this motor rotates through a fixed angular step in response to each input current pulse received by its controller.

In recent years, there has been wide-spread demand of Suckers because of the explosive growth of the computer industry. Their popularity is due to the fact that they can be controlled directly by computers, microprocessor and programmable controllers.

As we know, industrial motors are used to convert electric energy into mechanical energy but they cannot be used for precision positioning of an object or precision control of speed without using closed-loop feedback.

Suckers are ideally suited for the situations where either precise positioning or precise speed control or both are required in automation systems.

Application:

Such Sucker are used for operation control in computer peripheral, textile industry, IC fabrications and robotic etc, Application requiring incremental motion are typewriters, line printers, tape drives, floppy disk drives, numerically-controlled machine tools, process control systems and X-Y plotters. Usually, position information can be obtained simply by keeping count of the pulses sent to the motor there by eliminating the need for expensive position sensors and feed back controls. Suckers also perform countless tasks outside the computer industry. It includes commercial, military and medical applications where these Suckers perform such functions as mixing, cutting, striking, metering, blending and purging.

4.4 HD 74LS 373:

The HD 74LS 373, 8-bit register features totem-pole three-state outputs designed specially for driving highly-capacitive or relatively low impedance loads. The high impedance third state and increased high-logic level drive provide this register with the capacity of being connected directly to and driving the bus lines in a bus-organized system without need for interface or driving pull-up components. They are particularly attractive for implementing buffer registers, I/O ports, bidirectional bus drivers, and working register.

4.5: ULN 2803A:

The ULN2803A is a high-voltage, high-current Darlington transistor array. The device consist of eight npn Darlington pairs that feature high voltage outputs with common-cathode clamp diodes for switching inductive loads. The collector-current rating of each Darlington pair is 500 mA. The Darlington pairs may be connected in parallel for higher current capability.

4.6 Board Panel:

This bread board is ideally for basic design and experiment of analog or digital circuit. It can offer end-users to perform circuit development in interface or communication field with special universal connector basic plate.

4.7 Operation Circuit Design:

It is design which enables the simplified antenna by controlling for the module (two dimensions axes X for horizontal, Y for vertical) showed on the figure 5.1 below; the circuit design operates by (+5V) voltages to enable four ICs (2*HD 74LS373, 2*ULN2803A) on the board which connected by the two Suckers which have been installed in the antenna module.

All theses features are shown in figure below.

In order to develop software for a BASIS stamp, personal computer is used to write the program for the stamp, connect the PC to the stamp, and then use the PC to send the program to the stamp.

Once the stamp has been programmed, the program stays in the stamp's memory, and will run from the beginning every time on the stamp.

The programming connection between a samp and personal computer is conceptually simple.

But this simple task is complicated by a number of variables.

In the first, there are many different kids of connections on computers and some are suited to stamp programming than others.

Secondly, different modes of BASIC stamp have different connections.

4.8 Design Steps:

1/ Step one:

From D-25 connector to IC No (1) HD 74LS373

The interconnections ore following:

Table 4.1 : Connection of the microcontroller

To the IC No (1) HD 7LS 373

D-25 connector	IC No (1) HD 7 LS 373
PIN 12	PIN 8
PIN 13	PIN 7
PIN 14	PIN 4
PIN 15	PIN 3

2/ Step No two:

From D-25 connector to IC No (2) HD 4L S 373

The interconnections are following:

Table 4.2 : Connection of the microcontroller

To the IC No (2) HD 7LS373

D- 25 connector	IC No (2) HD 7 LS 373
PIN8	PIN 8
PIN 9	PIN 7
PIN 10	PIN 4
PIN 11	PIN 3

3/ Step three:

From IC No (1) HD 74LS 373 to IC No (3) ULN 2803

The interconnections are following:

Table 4.3: Connection of the IC No (1) to IC No (3) ULN 2803

IC No (1) HD 74 LS 373	IC No (3) ULN 2803
PIN 2	PIN 1
PIN 5	PIN 2
PIN 6	PIN 3
PIN 9	PIN 4

4/ Step four:

From IC No (2) HD 74LS 373 to C (4) ULN 2803

The interconnections are following:

Table: 4.4: Connection of the IC No(2) to IC No (4) ULN 2803

IC No (2) HD 74 LS 373	IC No (4) ULN 2803
PIN 2	PIN 1
PIN 5	PIN 2
PIN 6	PIN 3
PIN 9	PIN 4

5/ Step No (5)

From IC No (3) ULN 2803 Suckers No(1):

The interconnections are following:

Table: 4.5: Connection of the IC No (3) to the Suckers No (1)

IC No (3) ULN 2803	**SUCKERS (1)**
PIN 16	PIN 1 Line
Common 9	Block

6/ steps No (6)

From IC No (4) ULN 2803 to Suckers No(2)

Table 4.6 : connection of the IC No (4) to the Suckers No (2)

IC No (4) ULN 2803	**SUCKERS No (2)**
PIN 16	PIN 1 Line
Common 9	Block

4.9 Voice Analysis :

4.9.1 Steps for Voice Analysis:

In most common speaker recognition system there are some steps to follow, the speaker enters in order to change the voice to electrical signal such as microphones, sound cards .. etc. in this research microphone is used.

An important problem in voice processing is to detect the presence of a voice in a background of noise. This problem is often referred to as the end point location problem. The accurate detection of a word's start and end points means that subsequent processing of the data can be kept to a minimum.

In this step (frame blocking) the voice signal is blocked into frames of N samples, with adjacent frames being separated by M samples. The first illustrated frame consists of the first N voice samples. The second frame begins with M saples after the first frame, and overlaps it by (N-M) samples. Similarly, the third frame (or M samples after the second frame) and overlaps it by (N-2M) samples. This process continues until all the voice is accounted for within one or more frame. It is easy to see that if M < N, then adjacent frames overlap, and the resulting LPC spectral estimates will be correlated from frame to frame; if M<<N, then LPC spectral estimate from frame to frame will be quite smooth. On the other hand, if M>N, there will be no overlap between adjacent frames; in fact, some of the voice signal will be totally lost (i.e., never appear in any analysis frame), and the correlation between the resulting LPC spectral estimate of adjacent frames will contain a noisy component whose magnitude increases as M increases (i.e., as more voice is omitted from analysis). [4]

After framing, and to find out factors of LPC, any frame is multiplied by windows.

Chapter Five
Interface and
Feature of Extraction of Voice

5.1 Introduction

Voice recognition, or speech recognition in this case, is a computer technology that utilizes audio input for entering data rather than a keyboard. Speaking into a microphone, for example, produces the same result as typing words manually with a keyboard. Simply stated, voice recognition software is designed with an internal database of recognizable words or phrases. The program matches the audio signature of speech with corresponding entries in the database.

Though turning speech into text might sound easy, it is an extremely difficult task. The problem lies in the virtually infinite array of individual speech patterns and accents, compounded by the natural human tendency to run words together.

An illustration of the inherent challenges of voice recognition software appears on a T-shirt created by Apple researchers. The shirt reads, "I helped Apple wreck a nice beach." When spoken aloud, it sounds like, I helped Apple recognize speech.

Various models of voice recognition software are used for an array of applications, from personal dictation to commercial automated call routing, from aiding the disabled to sports and news event subtitling. Each model behaves differently and has its own capabilities and boundaries.

Voice recognition programs that require the user to "train" the software to recognize their particular stylized patterns of speech are called speaker dependent systems. Individuals commonly use these types of programs at home or at the office. Email, memos, letters, data and text can be input by speaking into a microphone.

Some voice recognition systems, called discrete speech systems, require the user to speak clearly and slowly and to separate words. Continuous speech systems are designed to understand a more natural mode of speaking.

Discrete speech voice recognition systems are widely used for customer service routing. The system is speaker independent, but understands only a small pool of words or phrases. The caller is given a choice to answer a question, usually with "yes" or "no." After receiving an answer, the system escalates the caller to the next level. If the caller replies with a unique answer, the automated response is usually, "Sorry, I didn't understand you; please try again," with a repeat of the question and available answers. This type of voice recognition is also referred to as grammar constrained recognition.

Continuous speech is a more sophisticated form of voice recognition software, wherein the caller can speak naturally to explain a problem or request a service. This program is designed to pick out key words or phrases and make a statistical best-guess as to what the customer wants. Speaking plainly aids voice recognition in identifying the need. This type of system has a far more intensive database than discreet speech systems and is also referred to as natural language recognition.

Automatic Speech Recognition (ASR) is a model of voice recognition designed for dictation. This software differs from previous models in that it does not strive to understand what is being said, only to identify the words spoken. Since many words in the English language sound alike, mistakes are easily made. However, major companies like Microsoft are investing in voice recognition, and Bill

Gates' own prediction has ASR understanding continuous speech by the year 2011. ASR software is often found on digital voice recorders.

Dominant players in voice recognition software have been ScanSoft and Nuance, with the former company acquiring the latter. Smaller players include Fonix Speech, Aculab and Verbio, among others, with major corporations like IBM and the aforementioned Microsoft also investing in the technology. Though many still feel it is more trouble to train software and correct mistakes than to simply use a keyboard, a time is coming when voice recognition software will likely close that gap. Augmenting keyboards with the discriminate ability to use speech will probably become commonplace.

Voice recognition software is gaining popularity as it becomes more sophisticated. It is especially useful in business where it can replace a live operator to funnel calls, disseminate information, take orders and perform other highly useful functions. However, it is also gaining favor as a desktop application, helped along by renowned software like Scan Soft's, Dragon Naturally Speaking and IBM's Via-Voice.

5.2 Feature Extracting:

Some feature extracting Techniques:

- Filter bank Cepstral Coefficients.
- Discrete Fourier Transform (DFT).
- Linear Predictive Cepstral Coefficients (LPCC).
- Perceptual linear prediction (PLP).
- Linear Predictive coding (LPC) (which is implemented among this project).

5.2.1 Digital Filter Banks

Filter Bank methods were historically the first methods introduced since they are implemented in analogue circuits. Filter Bank methods can be regarded a crude model of the initial stage of transduction in human auditory system. The output of this analysis is a vector of power/frequency pairs for each time frame of data. These are usually combined with other parameters, such as total power, to form the signal measurement vector.

5.2.2 Discrete Fourier Transform

As an initial guess, using the Discrete Fourier Transform (DFT) as a feature extractor appears to be a good idea. Unlike the time domain representation of the digital sound sample itself, the frequency magnitude does contain information about the pitch and formants. However, the spectral magnitude also holds a great deal of other information besides the pitch and formant locations. Ideally, we would like to analyze only the spectral envelope of the frequency magnitude. Thus, in order to avoid any unnecessary information, we considered three other techniques to extract pitch and formant information. Currently, both Fourier Transform and Linear Prediction techniques enjoy widespread use in various voice processing applications. [4]

5.2.3 Perceptual Linear Prediction

Perceptual Linear Prediction (PLP) was originally proposed by Hynek Hermansky as a way of warping spectra to minimize the differences between speakers while preserving the important voice information.

5.2.4 Linear predicative coding (LPC)

Linear Predictive Coding (LPC) is one of the most powerful voice analysis techniques, and one of the most useful methods for encoding good quality voice at a low bit rate. It provides extremely accurate estimates of voice parameters, and is relatively efficient for computation.

Before describing a general LPC front-end processor for speaker recognition, it is worthwhile to introduce the reasons why LPC has been so widely used. These include the following:

1- LPC provides a good model of the voice signal.

2- The way in which LPC is applied to the analysis of voice signals leads to a reasonable source-vocal tract separation.

3- LPC is an analytically tractable model. The method of LPC is mathematically precise and is simple and straightforward to implement in either hardware or software. The computations involved in LPC processing are considerably less than the required for all-digital implementation of—for example-the bank of filters model.

5.2.5 Basic Principles of Linear predictive coding:

Linear predictive coding (LPC) is defined as a digital method for encoding an analog signal in which a particular value is predicted by a linear function of the past values of the signal.

Linear prediction is a good tool for analysis of voice signals. Linear prediction models the human vocal tracts as an infinite impulse response (IIR) system that produces the voice signal. For vowel sounds and other voiced regions of speech, which have a resonant structure and high degree of similarity over time shifts that are multiple of their pitch period, this modeling produces an efficient representation of the sound.

Primarily, interested in obtaining an estimate of the spectral envelope of the frequency magnitude spectrum in order to find formant locations and thus classify speakers based on their unique voice patterns. Linear predictive coding (LPC) offers a powerful, yet simple method to provide exactly this type of information. Basically, the LPC algorithm produces a vector of coefficients that represent a smooth spectral envelope of the DFT magnitude of a temporal input signal. These coefficients are found by modeling each temporal sample as a linear combination of the previous p samples as:

$$x(n) = \sum_{k=1}^{p} a_k x(n-k) + e(n) = \hat{x}(n) + e(n) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5.1)$$

5.2.6 LPC coefficient representations:

LPC is frequently used for transmitting spectral envelope information, and as such it has to be tolerant for transmission errors. Transmission of the filter coefficients directly is undesirable, since they are very sensitive to errors. In other words, a very small error can distort the whole spectrum, or worse, a small error might make the prediction filter unstable.

There are more advanced representations such as log area ratios (LAR), line spectrum pairs (LSP) decomposition and reflection coefficients. Of these, especially LSP decomposition has gained popularity, since it ensures stability of the predictor, and spectral errors are local for small coefficient deviations.

We can represent the LPC model in terms of two key components:

1. Analysis part (i.e. encoding):

- Breaking the voice signal into segments;

- For each segment, label it as voiced/unvoiced,

- Then do pitch period estimation, and estimate the model parameters.

2. Synthesis part (i.e. decoding):

- The reverse of the encoding process.

- Each segment is decoded individually.

- The sequence of reproduced sound segments is joined together to represent the entire input voice signal.

5.2.7 Linear Predictive Coefficients

Linear Predictor analysis is widely used in voice signal analysis in time domain. The principle of a linear predictor is that, it predicts each samples of speech wave from as a linear sum of the past samples. The weights which minimize the mean - square prediction error are called the prediction square error coefficients. The equation for computing present sample is given by

$X (n) = al\, X (n-i) + a2\, X (n-2)- - - -aK\, X(n-K) + e(n)$ ………(5.2)

$X (n)$ =Current Samples

$a1$ - - K = Linear prediction coefficients

$X (n-1)$ - - $X (n-K)$ = Previous samples

$e(n)$ = Mean square error

The Linear Predictor is used to separate speaker dependent information such as vocal - tract length, pitch, formant frequencies from speech. The predictor square coefficients represent the combined information about the formant frequencies, their bandwidth and the glottal wave. There are other LPC coefficients are currently being used in speaker verification and identification purpose and some of them are given below:

1. LP coefficients: Found effective for speaker verification and identification.
2. LP Residual coefficients: This contains information about pitch.

The corruption of speech signal due to noise and the transmission channel is unavoidable; it was proven that some LPC coefficients are robust to noise.

5.2.8 Spectral Parameters:

It is spectrum coefficient that can be calculated direct from signal, or from LPC parameters, it give smoother spectrum cover from that LPC parameters give , hence the same word that told by the same user several time can be presented in settable presented way.

These parameters calculated from every frame to high light dynamical changes in frame spectrum.

5.3 Audio Signal processing in MATLAB

The quality of the ridge structures in a audio signals is an important characteristic, as the ridges carry the information of characteristic features required for minutiae extraction. Ideally, in a well-defined audio signals the ridges and valleys should alternate and flow in locally constant direction audio signals are signal with special characteristics, to perform audio signal processing in MATLAB will hopefully a better feel for connection between an algorithm described as mathematical formulas and how such algorithm might be implemented.

MATLAB is a bit different from other programming environments, processing of data stored in arrays is usually performed using iterations of the array elements (ie. Using for perform iteration over the array elements, since most image processing operations can be described as elements wise operations, linear combination , or calls to MATLAB Functions).

5.3.1 Practical:

The first step is for the user to enter the voice and then stored in ,memory in the system recognition data base. To determine the "meaning" of this voice input, the computer attempts to match the input with a digitized voice sample, or template, which has a known meaning.

The second step in the practical imported of the audio signals from it is file to MATLAB. Which used to find script and function M- files are organized into numerous directories (or folders) and sub directories (or sub folders . [4]

The third step but the file name (Zero 4) it recognition by mp3 in the code after run which result of site parameter.

This explain voice is happened analysis by certain alogri the in matlab after that give Zero crossing this mean 100%.

The voice recorded by Mp3 recognition, (Voice file. wav), the results of the program processed by the circuit (HD7445373).

All the samples of the audio signals provided at sucker ,and then converted by a HD74LS373 to digitize the audio signals.

5.3.2 Audio Signals Files:

audio signals data can be saved to files and reloaded into MATLAB using many different file formats, the normal MATLAB share and load function support voice data in wither double or 8 unit format.

And then the audio signals were read ,and converted to binary voice.

5.3.3 Audio Signals filters

Filtering is a technique for modifying or enhancing voice, for example, you can filter voice to emphasize certain features or remove other features.

5.3.4 Audio Signals Transform:

The usual mathematical representation of audio signals is a function of two spatial variables: X(t,f). The term transform refers to an alternative mathematical representation of voice.

5.3.5 Fast Fourier F.F:

The time and frequency domains are alternative ways of representing signals. The Fourier transform is the mathematical relationship between these two representations,

5.3.6 The Magnitude Spectrum:

A logarithmic transformation is applied to the magnitude spectrum of the enhanced audio signals and mapped to a log scale

5.3.7 Audio Signals Feature:

Many methods are used to get the features of the audio signals.

In this research audio signals transformed to frequency domain to extract audio signals features in which the value of magnitude spectrum.(of the absolute value) converted to one dimension vector to resolve frequency feature. The PSD (Power Spectral Density) and MS SPECTRUM (Mean square (power) Spectrum) data objects directly.

These objects can be used to contain existing spectrum data and enable us to use the plotting features. These objects also accept the data in a matrix format where each column is different spectral estimate.

Chapter Six

Conclusion and Recommendations

6.1 Conclusion:

The Matlab can be account as one of developed and strong technique that is easy to implement if it had been taken on account improve the Matlab design to make more suitable for the wanted application and that by good choice of best architecture and best transformation algorithm.

This research show good path to choose the Matlab architecture and that include choose the number of nodes and layers and transformation function and by trial and error strategy and the designed network generalization was improved to make more effective when it deal with new situation because the human voice is not invariant in time therefore the biometric template must be adapted during progressing time. The human Voice is also variable through temporal variations of the Voice, caused by a cold, hoarseness, stress, emotional different states or puberty vocal change.

The advantage of a speaker recognition system is the very cheap hardware that is needed - in most computers a soundcard and a microphone in addition to recording mechanism (usually a software program such as Microsoft sound recorder which is implemented in this project). It is very easy to use amid to implement with many applications especially for security purposes.

In addition it can been calculated that the speaker recognition is one of the few recognition areas where machines can outperform humans. Also speaker recognition is a viable technology available for implementation and can be augmented with other authentication methods to increase the level of security.

6.2 Recommendations:

- The human Voice is variable through temporal variations of the Voice, caused by a cold, hoarseness, stress, emotional different states or puberty vocal change. Therefore, for the future work these problems would be under consideration.
- Develop this programs to achieve more accurate, to be usable in practical field, such as attendants for student, security, …etc.
- More research to make suitable circuits with programs that can make relation between voice and what is the health situation (ie. To test some disease).
- More research make all the system automatically which mean run the suckers with circuit design without take voice file manually to enter matlab program and automatically make run the code (ie developing the program).

References

1. Lawerence Rabiner .Biing –Hawang Jung. “Fundamental of speech recognition” prentice hall .new Jersey 1993.
2. http/www.wikimediafoundation.org.
3. Math Works INS , “signal processing toolbox for user with MATLAB” 1955.
4. Dr. Abdalrasool Gbar Alzebaidy “Character Speaker Recognition System”, August 2008.
5. http/www.wikimediafoundation.org-,Voice classification.
6. http/secure.nch.com.au/cgi-bin/register.exe.
7. Software = record pad and svar = relpo off ss off.

٨. المرجع الأساسي لتعليم ماتلاب ٦.١

مهندس محمد عارف نعمه – مهندس مازن نعمه

دار الكتب العلمية للنشر والتوزيع ، ٢٠٠٣.

Appendices

Appendix (1)

```
%this program computes a number of statistics

%for an utterance stored in data file

x=wavread('Zero')

fprintf('digit statistics\n\n')

fprintf('mean:%f\n',mean (x))

fprintf('standerd deviation:%f\n',std(x))

fprintf('variance:%f\n',std(x)^2)

fprintf('average power:%f\n',mean(x.^2)

fprintf('average magnitude:%f\n',mean(abs(x)))

crossings = x؛

for j = 1 :length(x)-1؛

    if x(j)*x(j+1)<0

        crossings1= crossings+1؛

    end

end

fprintf('Zero crossings:%f\n',crossings)؛
```

```
this program computes a number of statistics

%for an utterance stored in data file

x=wavread(' Zero 1')

fprintf('digit statistics\n\n')

fprintf('mean:%f\n',mean (x))

fprintf('standerd deviation:%f\n',std(x))

fprintf('variance:%f\n',std(x)^2)

fprintf('average power:%f\n',mean(x.^2)

fprintf('average magnitude:%f\n',mean(abs(x)))

crossings = x؛

for j = 1 :length(x)-1؛

    if x(j)*x(j+1)<0

        crossings2= crossings+1؛

    end

end

fprintf(' Zero 1 crossings:%f\n',crossings)؛
```

```
%this program computes a number of statistics

%for an utterance stored in data file

x=wavread(' Zero 2')؛

fprintf('digit statistics\n\n')

fprintf('mean:%f\n',mean (x))

fprintf('standerd deviation:%f\n',std(x))

fprintf('variance:%f\n',std(x)^2)

fprintf('average power:%f\n',mean(x.^2)

fprintf('average magnitude:%f\n',mean(abs(x)))

crossings = x؛

for j = 1 :length(x)-1؛

    if x(j)*x(j+1)<0

        crossings3 = crossings+1؛

    end

end

fprintf(' Zero 2 crossings:%f\n',crossings)؛
```

```
%this program computes a number of statistics
%for an utterance stored in data file
x=wavread(' Zero 3')؛
fprintf('digit statistics\n\n')
fprintf('mean:%f\n',mean (x))
fprintf('standerd deviation:%f\n',std(x))
fprintf('variance:%f\n',std(x)^2)
fprintf('average power:%f\n',mean(x.^2)
fprintf('average magnitude:%f\n',mean(abs(x)))
crossings = x؛
for j = 1 :length(x)-1؛
    if x(j)*x(j+1)<0
        crossings4 = crossings+1؛
    end
end
fprintf(' Zero 3 crossings:%f\n',crossings)؛
```

Appendix (2)

Function of Matlab

Command	Description
abs	Absolute value
acker	Compute the K matrix to place the poles of A-BK, see also place
axis	Set the scale of the current plot, see also plot, figure
bode	Draw the Bode plot, see also logspace, margin, nyquist1
c2dm	Continuous system to discrete system
clf	Clear figure (use clg in Matlab 3.5)
conv	Convolution (useful for multiplying polynomials), see also deconv
ctrb	The controllability matrix, see also obsv
deconv	Deconvolution and polynomial division, see also conv
det	Find the determinant of a matrix
dimpulse	Impulse response of discrete-time linear systems, see also dstep
dlqr	Linear-quadratic requlator design for discrete-time systems, see also lqr
dlsim	Simulation of discrete-time linear systems, see also lsim
dstep	Step response of discrete-time linear systems, see also stairs
eig	Compute the eigenvalues of a matrix
eps	Matlab's numerical tolerance
feedback	Feedback connection of two systems.
figure	Create a new figure or redefine the current figure, see also subplot, axis
for	For, next loop
format	Number format (significant digits, exponents)
function	Creates function m-files
grid	Draw the grid lines on the current plot
gtext	Add a piece of text to the current plot, see also text
help	HELP!
hold	Hold the current graph, see also figure

if	Conditionally execute statements
imag	Returns the imaginary part of a complex number, see also real
impulse	Impulse response of continuous-time linear systems, see also step, lsim, dlsim
input	Prompt for user input
inv	Find the inverse of a matrix
jgrid	Generate grid lines of constant damping ratio (zeta) and settling time (sigma), see also sgrid, sigrid, zgrid
legend	Graph legend
length	Length of a vector, see also size
linspace	Returns a linearly spaced vector
lnyquist1	Produce a Nyquist plot on a logarithmic scale, see also nyquist1
log	natural logarithm, also log10: common logarithm
loglog	Plot using log-log scale, also semilogx/semilogy
logspace	Returns a logarithmically spaced vector
lqr	Linear quadratic regulator design for continuous systems, see also dlqr
lsim	Simulate a linear system, see also step, impulse, dlsim.
margin	Returns the gain margin, phase margin, and crossover frequencies, see also bode
norm	Norm of a vector
nyquist1	Draw the Nyquist plot, see also lnyquist1. Note this command was written to replace the Matlab standard command nyquist to get more accurate Nyquist plots.
obsv	The observability matrix, see also ctrb
ones	Returns a vector or matrix of ones, see also zeros
place	Compute the K matrix to place the poles of A-BK, see also acker
plot	Draw a plot, see also figure, axis, subplot.
poly	Returns the characteristic polynomial
polyadd	Add two different polynomials
polyval	Polynomial evaluation
print	Print the current plot (to a printer or postscript file)

pzmap	Pole-zero map of linear systems
rank	Find the number of linearly independent rows or columns of a matrix
real	Returns the real part of a complex number, see also imag
rlocfind	Find the value of k and the poles at the selected point
rlocus	Draw the root locus
roots	Find the roots of a polynomial
rscale	Find the scale factor for a full-state feedback system
set	Set(gca,'Xtick',xticks,'Ytick',yticks) to control the number and spacing of tick marks on the axes
series	Series interconnection of Linear time-independent systems
sgrid	Generate grid lines of constant damping ratio (zeta) and natural frequency (Wn), see also jgrid, sigrid, zgrid
sigrid	Generate grid lines of constant settling time (sigma), see also jgrid, sgrid, zgrid
size	Gives the dimension of a vector or matrix, see also length
sqrt	Square root
ss	Create state-space models or convert LTI model to state space, see also tf
ss2tf	State-space to transfer function representation, see also tf2ss
ss2zp	State-space to pole-zero representation, see also zp2ss
stairs	Stairstep plot for discreste response, see also dstep
step	Plot the step response, see also impulse, lsim, dlsim.
subplot	Divide the plot window up into pieces, see also plot, figure
text	Add a piece of text to the current plot, see also title, xlabel, ylabel, gtext
tf	Creation of transfer functions or conversion to transfer function, see also ss
tf2ss	Transfer function to state-space representation, see also ss2tf
tf2zp	Transfer function to Pole-zero representation, see also zp2tf
title	Add a title to the current plot
wbw	Returns the bandwidth frequency given the damping ratio and the rise or settling time.
xlabel/ylabel	Add a label to the horizontal/vertical axis of the current

	plot, see also title, text, gtext
zeros	Returns a vector or matrix of zeros
zgrid	Generates grid lines of constant damping ratio (zeta) and natural frequency (Wn), see also sgrid, jgrid, sigrid
zp2ss	Pole-zero to state-space representation, see also ss2zp
zp2tf	Pole-zero to transfer function representation, see also tf2zp

VOICEBOX: Speech Processing Toolbox for MATLAB:

VOICEBOX is a speech processing toolbox consists of MATLAB routines that are maintained by and mostly written by Mike Brookes, Department of Electrical & Electronic Engineering, Imperial College, Exhibition Road, London SW7 2BT, UK. Several of the routines require MATLAB V6.5 or above and require (normally slight) modification to work with ea Audio File Input/Output

Routines are available to read and, in some cases write, a variety of file formats:

Read	**Write**	**Suffix**	
readwav	writewav	.wav	These routines allow an arbitrary number of channels and can deal with linear PCM (any precision up to 32 bits), A-law PCM and Mu-law PCM. Large files can be read and written in small chunks.
readhtk	writehtk	.htk	Read and write waveform and parameter files used by Microsoft's Hidden Markov Toolkit.
readsfs		.sfs	Speech Filing system files from Mark Huckvale at UCL.
readsph		.sph	NIST Sphere format files (including TIMIT). Needs SHORTEN for compressed files.
readaif		.aif	AIFF format (Audio Interchange File Format) used by Mac users.
readcnx		cnx	Read Connex database files (from BT)
readau		au	Read AV audio files (from Sun)

Frequency Scale Conversion

From *f*	**To *f***	**Scale**	
frq2mel	mel2frq	mel	The *mel scale* is based on the human perception of sinewave pitch.
frq2erb	erb2frq	erb	The *erb* scale is based on the equivalent rectangular bandwidths of the human ear.
frq2erb	erb2frq	bark	The *bark* scale is based on critical bands and masking in the human ear.

frq2midi midi2frq midi	The *midi standard* specifies a numbering of *semitones* with middle C being 60. They can use the normal equal tempered scale or else the pythagorean scale of just intonation. They will in addition output note names in a character format.

Fourier, DCT and Hartley Transforms

Forward	**Inverse**	
rfft	irfft	Forward and inverse discrete fourier transforms on real data. Only the first half of the conjugate symmetric transform is generated. For even length data, the inverse routine is asumptotically twice as fast as the built-in MATLAB routine.
rsfft		Forward transform of real, symmetric data to give the first half only of the real, symmetric transform.
zoomfft		Calculate the discrete fourier transform at an arbitrary set of linearly spaced frequencies. Can be used to zoom into a subset of the full frequency range.
rdct	irdct	Forward and inverse discrete cosine transform on real data.
rhartley	rhartley	Hartley transform on real data (forward and inverse transforms are the same).

Random Numbers and Probability Distributions

- **Random Number Generation**
- The routine randvec generates random vectors from gaussian or lognormal mixture distributions.
- randiscr generates discrete random values with a specified probability vector
- The routine usasi generates noise with a USASI spectrum.
- The routine randfilt generates filtered gaussian noise without any startup transients.
- rnsubset selects a random subset of *k* elements from the numbers 1:*n*

- **Probability Density Functions**
- gmmlpdf calculates the log pdf of a multivariate Gaussian mixture

- lognmpdf calculates the pdf of a lognormal distribution

- **Miscellaneous**
- histndim calculates an n-dimensional histogram (and plots a 2-D one)
- gausprod calculates the product of two gaussian distributions
- maxgauss calculates the mean and variance of the maximum of a gaussian vector

Vector Distance

- The routine disteusq calculates the squared euclidean distance between all pairs of rows of two matrices.
- The routines distitar, distisar and distchar calculate the Itakura, Itakura-Saito and COSH spectral distances between sets of AR coefficients.
- The routines distitpf, distispf and distchpf calculate the Itakura, Itakura-Saito and COSH spectral distances between power spectra.

Speech Analysis

enframe	can be used to split a signal up into frames. It can optionally apply a window to each frame.
fram2wav	converts a sequence of frame-based value into a waveform
ewgrpdel	calculates the energy-weighted group delay waveform.
activlev	calculates the active level of a speech segment according to ITU-T recommendation P.56.
spgrambw	draws a monochrome spectrogram with a dB scale.
txalign	finds the best alignment (in a least squares sense) between two sets of time markers (e.g. glottal closure instants).
dypsa	estimates the glottal closure instants from the speech waveform.
fxrapt	is an implementation of the RAPT pitch tracker by David Talkin.
soundspeed	gives the speed of sound as a function of temperature

LPC Analysis of Speech

lpcauto & lpccovar	perform linear predictive coding (LPC) analysis. The routines relating to LPC are described in more detail on another page. A large number of conversion routines are included for changing

the form of the LPC coefficients (e.g. AR coefficients, reflection coefficients etc.): these are of the form lpcxx2yy where xx and yy denote the coefficient sets.

lpcrr2am calculates LPC filters for all orders up to a given maximum.

lpcbwexp performs bandwidth expansion on an LPC filter.

ccwarpf performs frequency warping in the complex cepstrum domain.

lpcifilt performs inverse filtering to estimate the glottal waveform from the speech signal and the lpc coefficients.

lpcrand can be used to generate random, stable filters for testing purposes.

Speech Synthesis

- The routines glotros and glotlf implement two common models for the waveform of airflow through the vocal folds.

Speech Enhancement

The routine estnoisem iuses a minimum-statistics algorithm to estimate the noise spectrum from a noisy speech signal that has been divided into frames.

The routine specsub implements spectral subtraction while ssubmmse implements minimum mean square estimation of spectral amplitudes or log amplitudes. Both of these use estnoisem to estimate the noise spectrum.

Speech Coding

- The routines lin2pcma, lin2pcmu, pcma2lin, and pcmu2lin convert audio waveforms to and from the 8-bit A-law and Mu-law PCM formats that are used in telecommunications: Mu-law is used in the USA and Japan while A-law is used in the rest of the world. The two formats are very similar and, for speech waveforms, give about the same perceived quality as 12-bit linear encoding. Alternate bits in the A-law format are usually inverted before transmission: the conversion routines can optionally include this. The conversions are defined by ITU standard G.711.
- The routines kmeans, kmeanlbg and kmeanhar perform vector quantisation using the k-means and k-harmonic means algorithms.
- The routine potsband calculates a bandpass filter corresponding to the standard telephone passband.

Speech Recognition

- The routine melcepst implements a mel-cepstrum front end for a recogniser. The associated bandpass filter matrix is generated by melbankm .
- The routines cep2pow and pow2cep convert state means and variances between the mel-cepstrum and power domains.
- The routine gaussmix fits a gaussian mixture distribution to a collection of observation vectors with either diagonal or full covariance matrices.
- ldatrace performs Linear Discriminant Analysis with optional constraints on the transform matrix

Signal Processing

findpeaks	finds the peaks in a signal
maxfilt	performs running maximum filter
meansqtf	calculates the output power of a rational filter with a white noise input
windows	generates window functions
zerocros	finds the zero crossings of a signal with interpolation
ditherq	adds dither and quantizes a signal
schmitt	passes a signal through a schmitt trigger having hysteresis
dlyapsq	solves the discrete lyapunov equation using an efficient square root algorithm
momfilt	generate running moments from a signal

Information Theory

huffman	calculates optimum D-ary symbol code from a probability mass vector
entropy	calculates entropy and conditional entropy for discrete and continuous distributions

Computer Vision

rot--2--	converts between the following representations of rotations: rotation matrix (ro), euler angles (eu), axis of rotation (ax), plane of rotation (pl), real quaternion vector (qr), real quaternion matrix (mr), complex quaternion vector (qc), complex quaternion matrix (mc). A detailed description is given here.
peak2dquad	find a quadratically-interpolated peak in a 2D array by fitting a biquadratic function to the array values

Printing and Display Functions

figbolden	makes the lines on a figure bold and enlarges font sizes for printing clearly
sprintsi	prints a value with the correct standard SI multiplier (e.g. 2100 prints as 2.1 k)
bitsprec	rounds values to a precision of n bits
frac2bin	converts numbers to fixed-point binary strings

Voicebox Parameters and System Interface

voicebox	contains a number of installation-dependent global parameters and is likely to need editing for each particular setup.
unixwhich	searches the WINDOWS system path for an executable (like UNIX which command)
winenvar	Obtains WINDOWS environment variables

Utility Functions

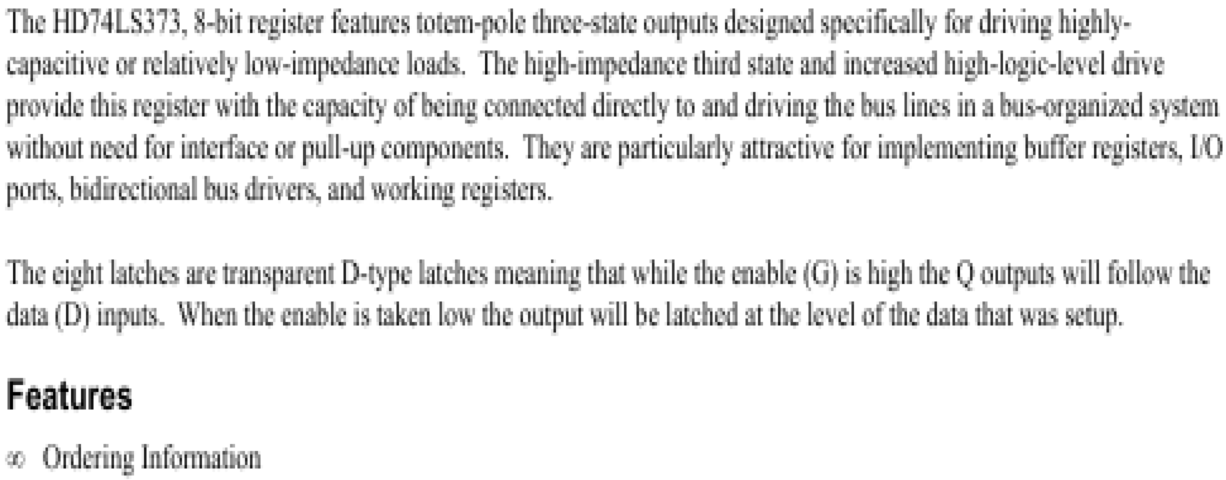
zerotrim	removes from a matrix any trailing rows and columns that are all zero.
logsum	calculates log(sum(exp(x))) without overflow problems.
dualdiag	simultaneously diagonalises two matrices: this is useful in computing LDA or IMELDA transforms.
permutes	all possible permutations of the numbers 1:n
choosenk	all possible ways of choosing k elements out of the numbers 1:n without duplications
choosrnk	all possible ways of choosing k elements out of the numbers

	1:n with duplications allowed
rotation	generates rotation matrices
skew3d	manipulates 3#3 skew symmetric matrices
atan2sc	arctangent function that returns the sin and cos of the angle
bitsprec	Rounds values to a precision of n bits
dlyapsq	Solve the discrete lyapunov equation
finishat	Estimate the finishing time of a long loop
m2htmlpwd	Create HTML documentation of matlab routines in the current directory

Appendix (3)

Data sheet of HD74LS373

The HD74LS373, 8-bit register features totem-pole three-state outputs designed specifically for driving highly- capacitive or relatively low-impedance loads. The high-impedance third state and increased high-logic-level drive provide this register with the capacity of being connected directly to and driving the bus lines in a bus-organized system without need for interface or pull-up components. They are particularly attractive for implementing buffer registers, I/O ports, bidirectional bus drivers, and working registers. The eight latches are transparent D-type latches meaning that while the enable (G) is high the Q outputs will follow the data (D) inputs. When the enable is taken low the output will be latched at the level of the data that was setup. Features • Ordering Information



The HD74LS373, 8-bit register features totem-pole three-state outputs designed specifically for driving highly-capacitive or relatively low-impedance loads. The high-impedance third state and increased high-logic-level drive provide this register with the capacity of being connected directly to and driving the bus lines in a bus-organized system without need for interface or pull-up components. They are particularly attractive for implementing buffer registers, I/O ports, bidirectional bus drivers, and working registers.

The eight latches are transparent D-type latches meaning that while the enable (G) is high the Q outputs will follow the data (D) inputs. When the enable is taken low the output will be latched at the level of the data that was setup.

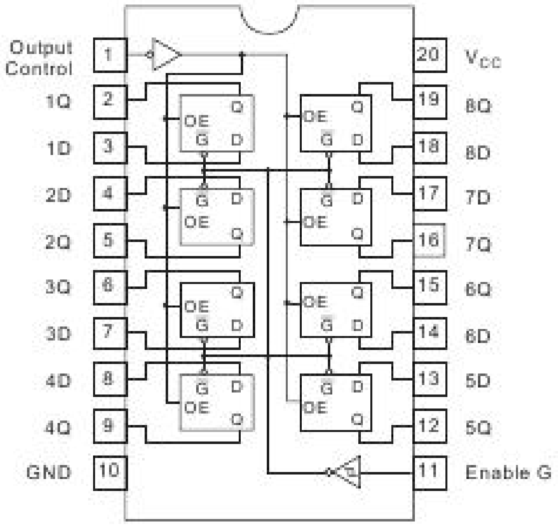
Features

∞ Ordering Information

Part Name	Package Type	Package Code (Previous Code)	Package Abbreviation	Taping Abbreviation (Quantity)
HD74LS373P	DILP-20 pin	PRDP0020AC-B (DP-20NEV)	P	—
HD74LS373FPEL	SOP-20 pin (JEITA)	PRSP0020DD-B (FP-20DAV)	FP	EL (2,000 pcs/reel)
HD74LS373RPEL	SOP-20 pin (JEDEC)	PRSP0020DC-A (FP-20DBV)	RP	EL (1,000 pcs/reel)

Note: Please consult the sales office for the above package availability.

Pin Arrangement



(Top view)

Function Table

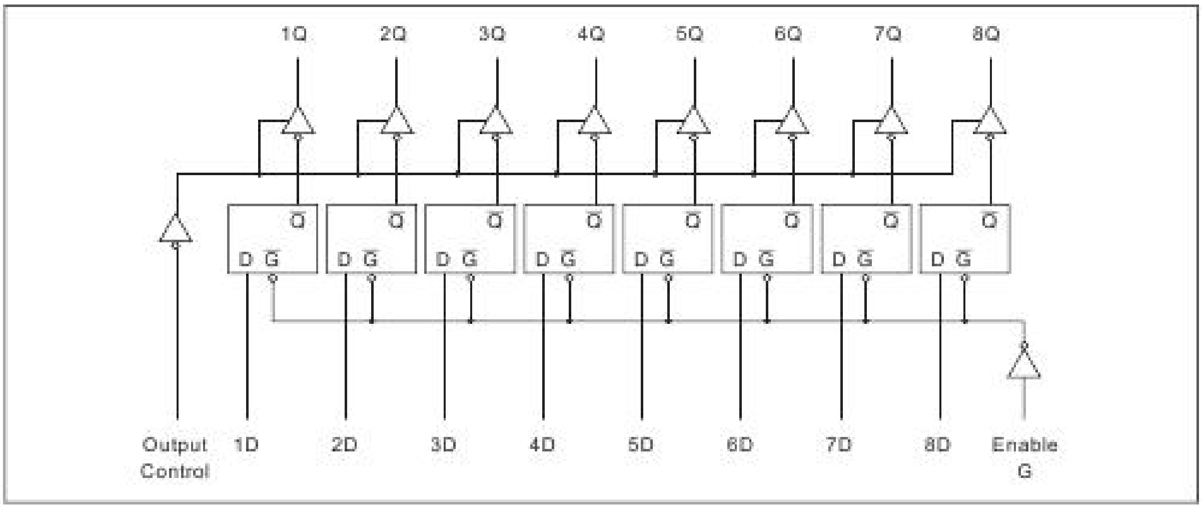
Inputs			Output
Output control	**Enable G**	**D**	**Q**
L	H	H	H
L	H	L	L
L	L	X	Q_0
H	X	X	Z

Notes: H; high level, L; low level, X; irrelevant

Q_0; level of Q before the indicated steady-state input conditions were established

Z; off (high-impedance) state of a three-state output

Block Diagram



Absolute Maximum Ratings

Item	Symbol	Ratings	Unit
Supply voltage	V_{CC}	7	V
Input voltage	V_{IN}	7	V
Power dissipation	P_T	400	mW
Storage temperature	Tstg	–65 to +150	ϒC

Note: Voltage value, unless otherwise noted, are with respect to network ground terminal.

Recommended Operating Conditions

Item		Symbol	Min	Typ	Max	Unit
Supply voltage		V_{CC}	4.75	5.00	5.25	V
Output voltage		V_{OH}	—	—	5.5	V
Output current		I_{OH}	—	—	–2.6	mA
		I_{OL}	—	—	24	mA
Operating temperature		Topr	–20	25	75	ϒC
Enable pulse width	"H" Level	t_w	15	—	—	ns
	"L" Level		15	—	—	ns
Data setup time		t_{su}	5↓	—	—	ns
Data hold time		t_h	20↓	—	—	ns

Electrical Characteristics

(Ta = –20 to +75 °C)

Item	Symbol	min.	typ.*	max.	Unit	Condition	
Input voltage	V_{IH}	2.0	—	—	V		
	V_{IL}	—	—	0.7	V	Data inputs	
		—	—	0.8		G, Output control inputs	
Output voltage	V_{OH}	2.4	—	—	V	V_{CC} = 4.75 V, V_{IH} = 2 V, V_{IL} = $V_{IL\,max}$ I_{OH} = –2.6 mA	
	V_{OL}	—	—	0.4	V	I_{OL} = 12 mA	V_{CC} = 4.75 V, V_{IH} = 2 V, V_{IL} = $V_{IL\,max}$
		—	—	0.5		I_{OL} = 24 mA	
Output current	I_{OZH}	—	—	20	∝A	V_O = 2.7 V	V_{CC} = 5.25 V, V_{IH} = 2 V
	I_{OZL}	—	—	–20		V_O = 0.4 V	
Input current	I_{IH}	—	—	20	∝A	V_{CC} = 5.25 V, V_I = 2.7 V	
	I_{IL}	—	—	–0.4	mA	V_{CC} = 5.25 V, V_I = 0.4 V	
	I_I	—	—	0.1	mA	V_{CC} = 5.25 V, V_I = 7 V	
Short-circuit output current	I_{OS}	–30	—	–130	mA	V_{CC} = 5.25 V	
Supply current	I_{CC}	—	24	40	mA	V_{CC} = 5.25 V, V_I = 4.5 V (Output control)	
Input clamp voltage	V_{IK}	—	—	–1.5	V	V_{CC} = 4.75 V, I_{IN} = –18 mA	

Note: * V_{CC} = 5 V, Ta = 25°C

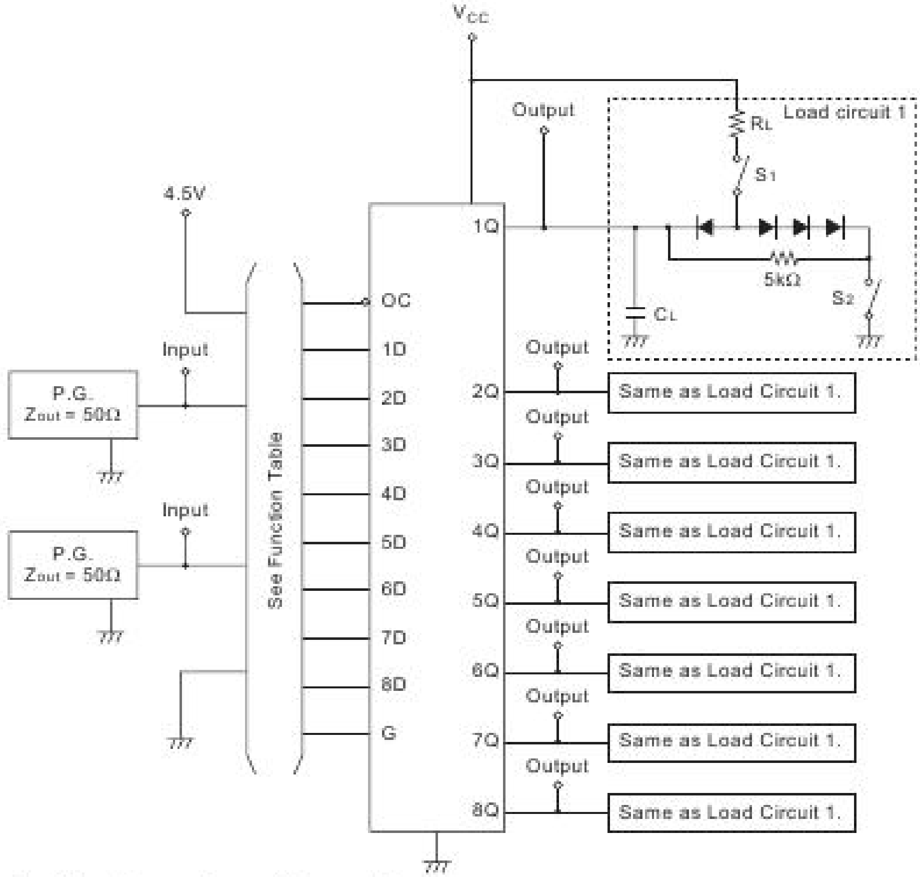
Switching Characteristics

(V_{CC} = 5 V, Ta = 25°C)

Item	Symbol	Input	Output	min.	typ.	max.	Unit	Condition
Propagation delay time	t_{PLH}	D	Q	—	12	18	ns	C_L = 45 pF, R_L = 667 ∣
	t_{PHL}			—	12	18		
	t_{PLH}	G	Q	—	20	30		
	t_{PHL}			—	18	30		
Output enable time	t_{ZH}	OC	Q	—	15	28		
	t_{ZL}			—	25	36		
Output disable time	t_{HZ}	OC	Q	—	12	20		C_L = 5 pF, R_L = 667 ∣
	t_{LZ}			—	15	25		

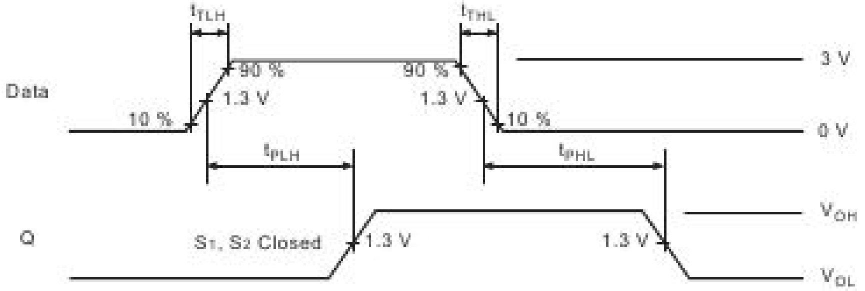
Testing Method

Test Circuit



Notes: 1. C_L includes probe and jig capacitance.
2. All diodes are 1S2074(H).

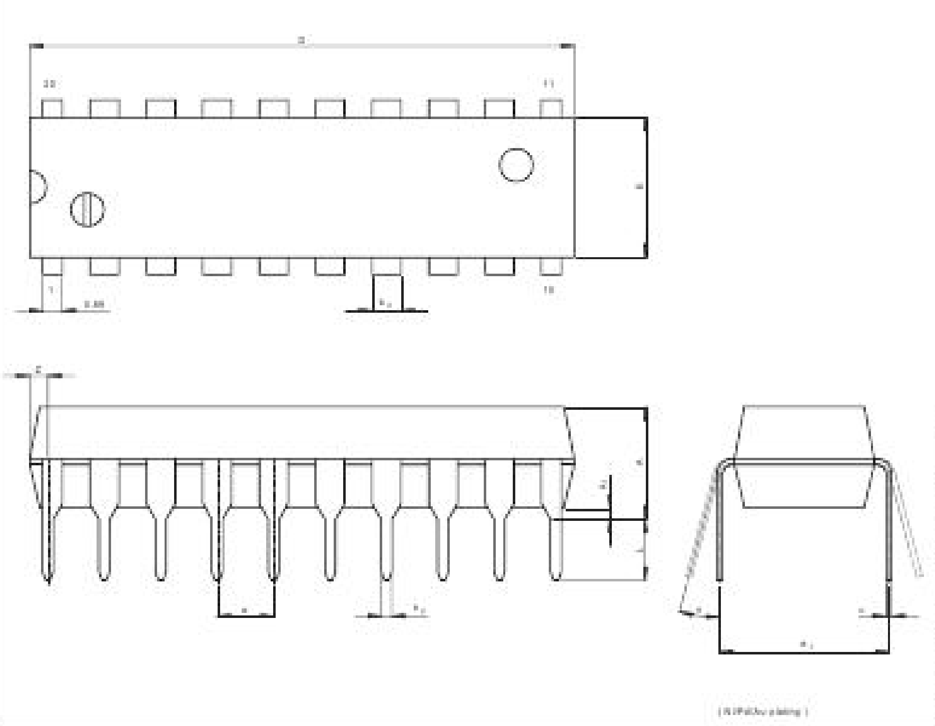
Waveforms 1



Note: Input pulse; t_{TLH} " 15 ns, t_{THL} " 6 ns, PRR = 1 MHz, duty cycle 50%

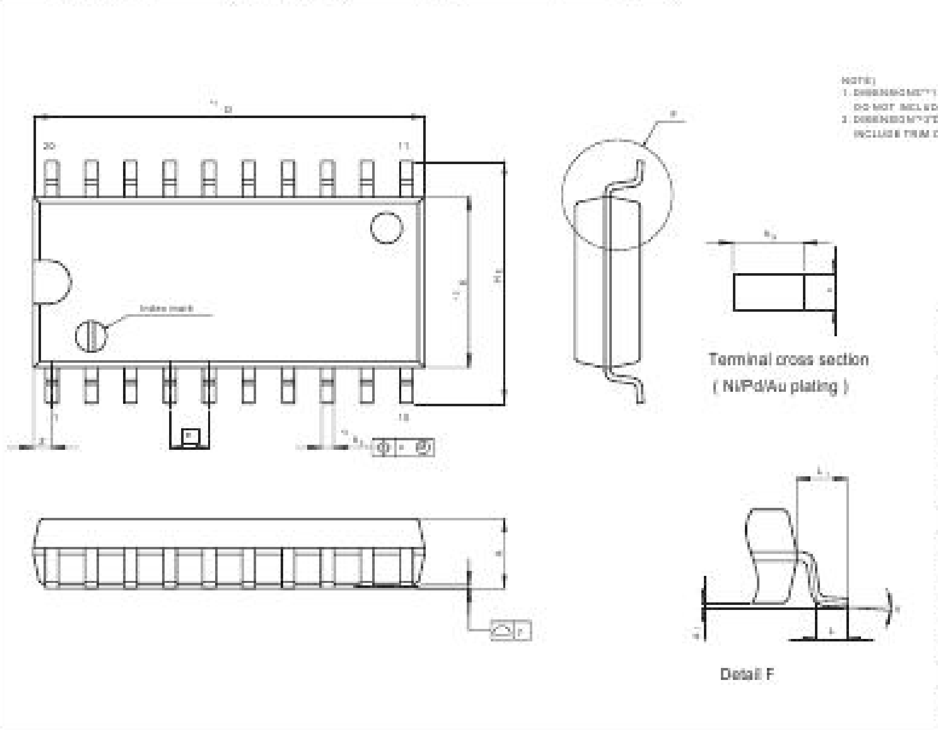
Package Dimensions

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-DIP20-6.3x24.5-2.54	PRDP0020AC-B	DP-20NEV	1.26g



Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
e_1	—	7.62	—
D	—	24.50	25.40
E	—	6.30	7.00
A	—	—	5.08
A_1	0.51	—	—
b_p	0.40	0.48	0.56
b_3	—	1.30	—
c	0.19	0.25	0.31
θ	0°	—	15°
e	2.29	2.54	2.79
Z	—	—	1.27
L	2.54	—	—

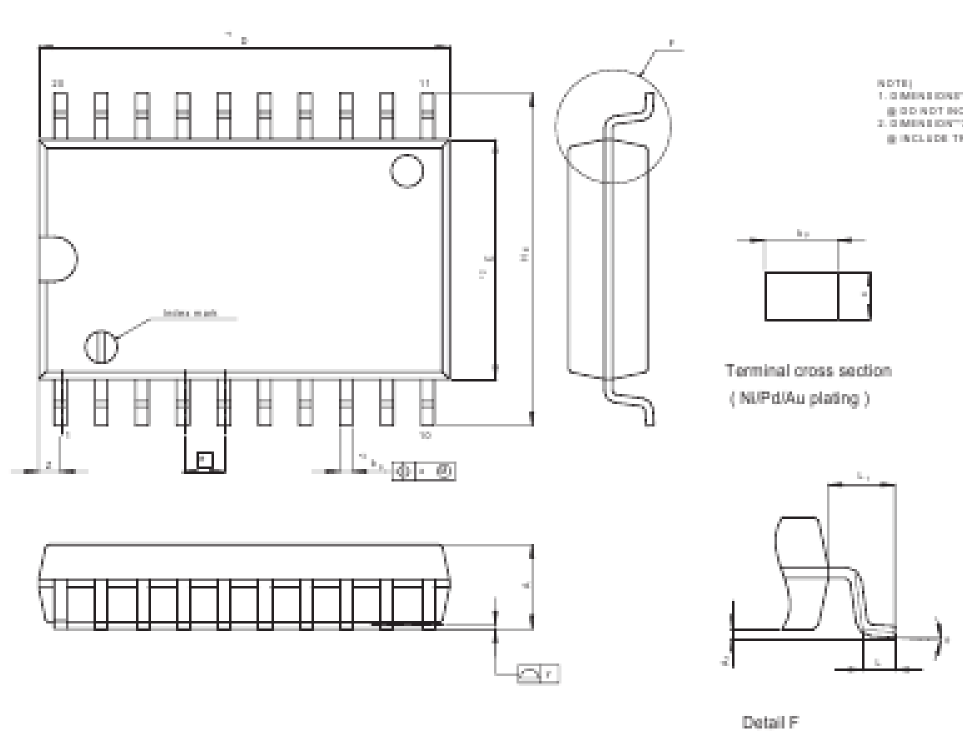
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-SOP20-5.5x12.6-1.27	PRSP0020DD-B	FP-20DAV	0.31g



NOTE)
1. DIMENSIONS"*1 (Nom)"AND"*2"
DO NOT INCLUDE MOLD FLASH.
2. DIMENSION"*3"DOES NOT
INCLUDE TRIM OFFSET.

Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	12.60	13.0
E	—	5.50	—
A_2	—	—	—
A_1	0.00	0.10	0.20
A	—	—	2.20
b_p	0.34	0.40	0.46
b_1	—	—	—
c	0.15	0.20	0.25
c_1	—	—	—
θ	0°	—	8°
H_E	7.50	7.80	8.00
e	—	1.27	—
x	—	—	0.12
y	—	—	0.15
Z	—	—	0.80
L	0.50	0.70	0.90
L_1	—	1.15	—

JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-SOP20-7.5x12.6-1.27	PRSP0020DC-A	FP-20DAV	0.31g



NOTE)
1. DIMENSIONS"*1 (Nom)"AND"*2"
DO NOT INCLUDE MOLD FLASH.
2. DIMENSION"*3"DOES NOT
INCLUDE TRIM OFFSET.

Terminal cross section
(Ni/Pd/Au plating)

Detail F

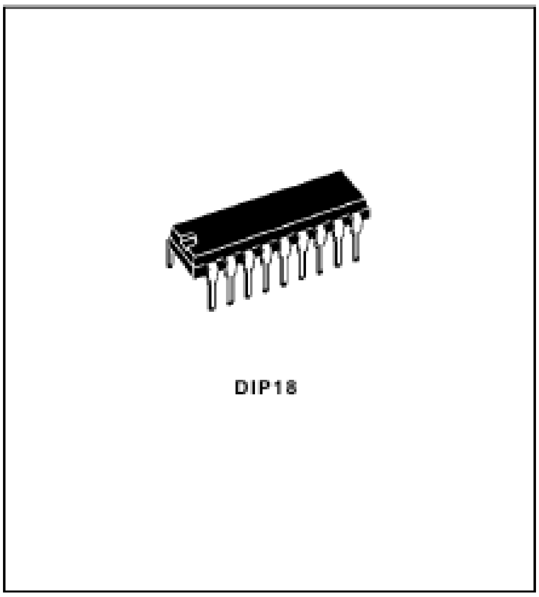
Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	—	12.60	13.0
E	—	7.50	—
A_2	—	—	—
A_1	0.10	0.20	0.30
A	—	—	2.65
b_p	0.34	0.40	0.46
b_1	—	—	—
c	0.20	0.25	0.30
c_1	—	—	—
θ	0°	—	8°
H_E	10.00	10.40	10.65
$\boxed{e}$	—	1.27	—
x	—	—	0.12
y	—	—	0.15
Z	—	—	0.935
L	0.40	0.70	1.27
L_1	—	1.40	—

Appendix (4)

Data sheet of ULN2801A

ULN2801A
ULN2804A - ULN2805A
ULN2802A - ULN2803A
September 1997
EIGHT DARLINGTON ARRAYS

- EIGHT DARLINGTONS WITH COMMON EMITTERS
- OUTPUT CURRENT TO 500 mA
- OUTPUT VOLTAGE TO 50 V
- INTEGRAL SUPPRESSION DIODES
- VERSIONS FOR ALL POPULAR LOGIC FAMILIES
- OUTPUT CAN BE PARALLELED
- INPUTS PINNED OPPOSITE OUTPUTS TO SIMPLIFY BOARD LAYOUT



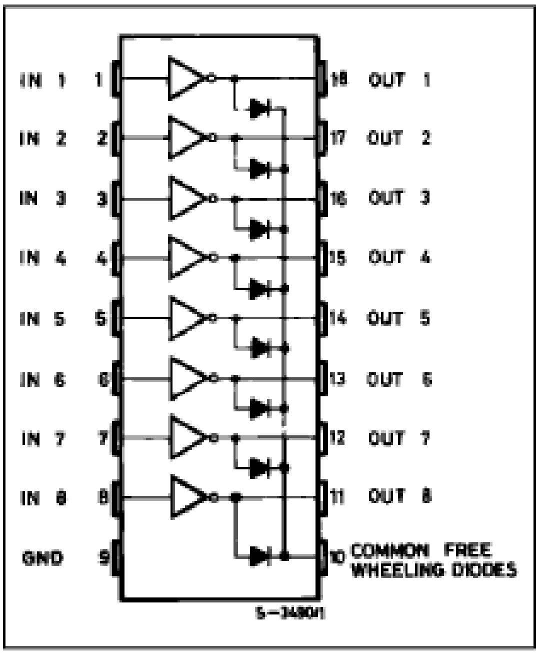
DESCRIPTION

The ULN2801A-ULN2805A each contain eight darlington transistors with common emitters and integral suppression diodes for inductive loads. Each darlington features a peak load current rating of 600mA (500mA continuous) and can withstand at least 50V in the off state. Outputs may be paralleled for higher current capability.

Five versions are available to simplify interfacing to standard logic families : the ULN2801A is designed for general purpose applications with a current limit resistor ; the ULN2802A has a 10.5k| input resistor and zener for 14-25V PMOS ; the ULN2803A has a 2.7k| input resistor for 5V TTL and CMOS ; the ULN2804A has a 10.5k| input resistor for 6-15V CMOS and the ULN2805A is designed to sink a minimum of 350mA for standard and Schottky TTL where higher output current is required.

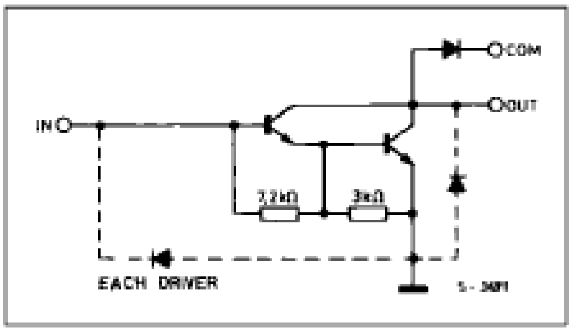
All types are supplied in a 18-lead plastic DIP with a copper lead from and feature the convenient input-opposite-output pinout to simplify board layout.

PIN CONNECTION (top view)

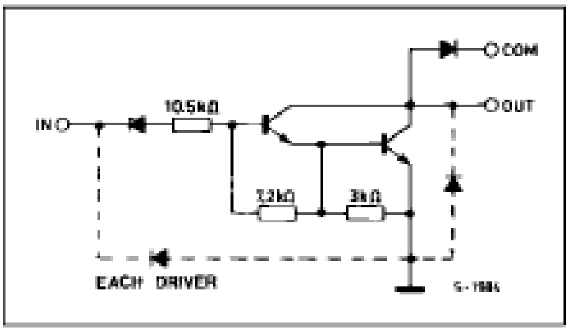


SCHEMATIC DIAGRAM AND ORDER CODES

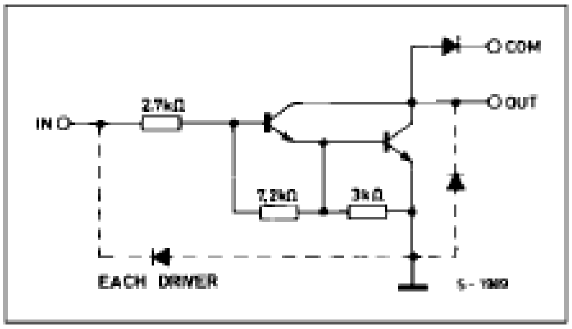
For ULN2801A (each driver for PMOS-CMOS)



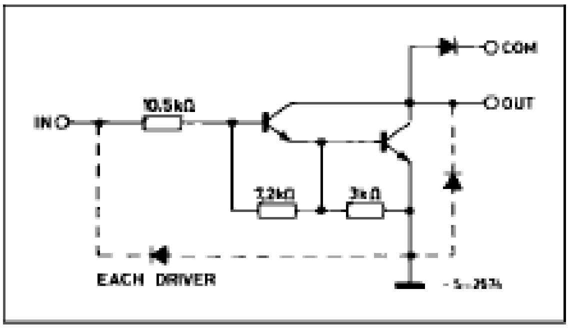
For ULN2802A (each driver for 14-15 V PMOS)



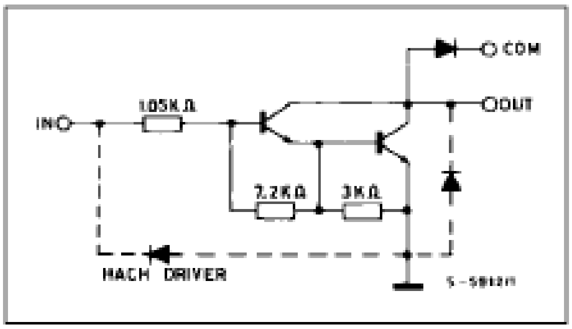
For ULN2803A (each driver for 5 V, TTL/CMOS)



For ULN2804A (each driver for 6-15 V CMOS/PMOS



For ULN2805A (each driver for high out TTL)



TEST CIRCUITS

Figure 1a.

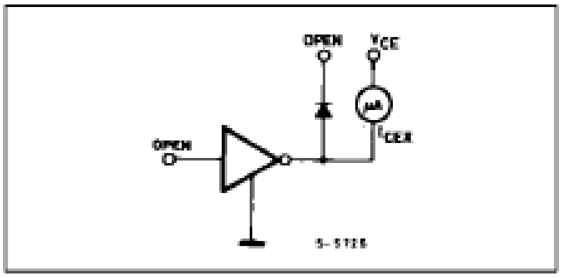


Figure 1b.

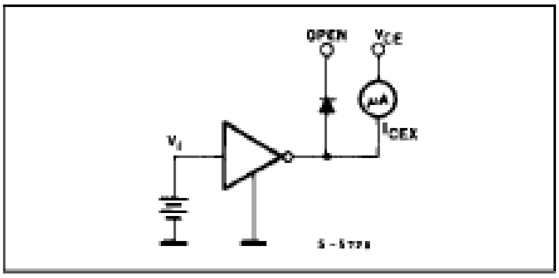


Figure 2.

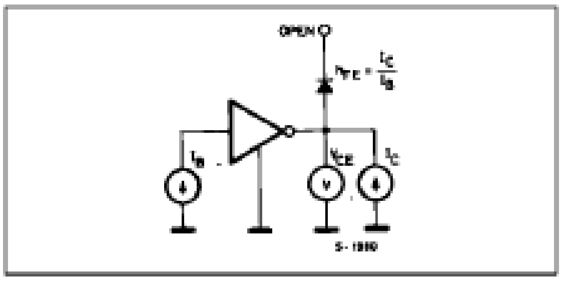


Figure 3.

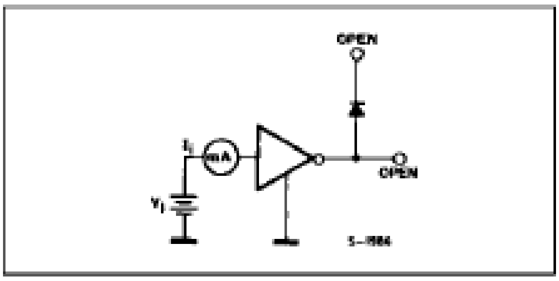


Figure 4.

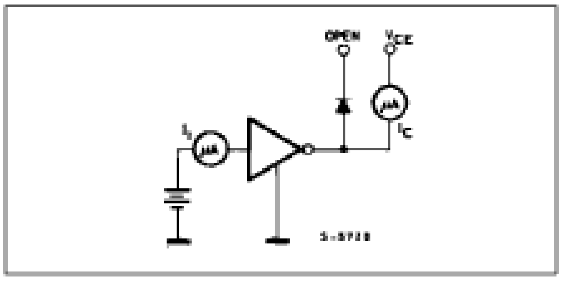


Figure 5.

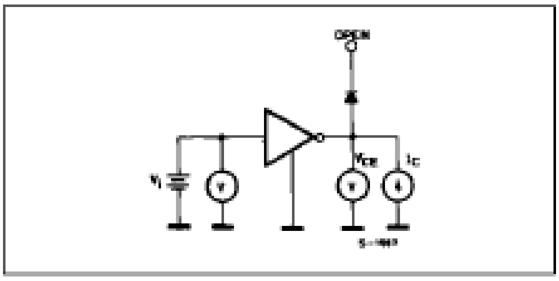


Figure 6.

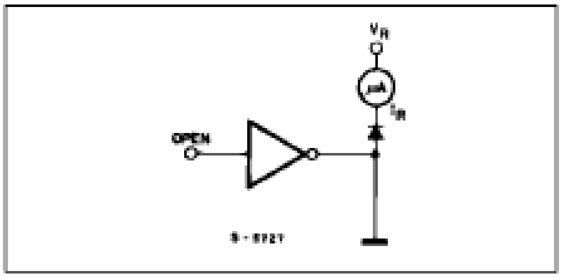
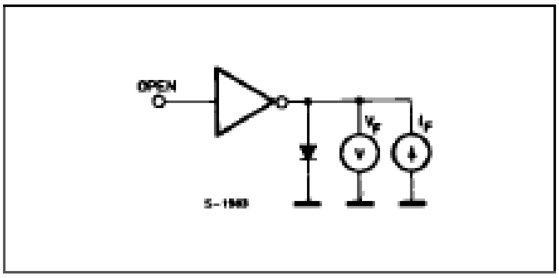


Figure 7.



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_o	Output Voltage	50	V

V_i	Input Voltage
	for ULN2802A, UL2803A, ULN2804A
	for ULN2805A
		
		30
		15
			V

I_C	Continuous Collector Current	500	mA
I_B	Continuous Base Current	25	mA
P_{tot}	Power Dissipation (one Darlington pair) (total package)	 1.0 2.25	W
T_{amb}	Operating Ambient Temperature Range	– 20 to 85	ϒC
T_{stg}	Storage Temperature Range	– 55 to 150	ϒC
Tj	Junction Temperature Range	– 20 to 150	ϒC

THERMAL DATA

Symbol	Parameter		Value	Unit
$R_{th\ j-amb}$	Thermal Resistance Junction-ambient	Max.	55	ϒC/W

ELECTRICAL CHARACTERISTICS (T_{amb} = 25°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit	Fig.
I_{CEX}	Output Leakage Current	V_{CE} = 50V T_{amb} = 70ϒC, V_{CE} = 50V T_{amb} = 70ϒC for ULN2802A V_{CE} = 50V, V_i = 6V for ULN2804A V_{CE} = 50V, V_i = 1V			50 100 500 500	∝A ∝A ∝A ∝A	1a 1a 1b 1b
$V_{CE(sat)}$	Collector-emitter Saturation Voltage	I_C = 100mA, I_B = 250∝A I_C = 200mA, I_B = 350∝A I_C = 350mA, I_B = 500∝A		0.9 1.1 1.3	1.1 1.3 1.6	V V V	2

$I_{i(on)}$	Input Current	for ULN2802A V_i = 17V
		for ULN2803A V_i = 3.85V
		for ULN2804A V_i = 5V
		V_i = 12V
		for ULN2805A V_i = 3V
				0.82
				0.93
				0.35
				1
				1.5
					1.25
					1.35
					0.5
					1.45
					2.4
						mA
						mA
						mA
						mA
						mA
							3

$I_{i(off)}$	Input Current	T_{amb} = 70ϒC, I_C = 500∝A	50	65		∝A	4
$V_{i(on)}$	Input Voltage	V_{CE} = 2 V for ULN2802A I_C = 300mA for ULN2803A I_C = 200mA I_C = 250mA I_C = 300mA for ULN2804A I_C = 125mA I_C = 200mA I_C = 275mA I_C = 350mA for ULN2805A I_C = 350mA			 13 2.4 2.7 3 5 6 7 8 2.4	 V V V V V V V V V	5
h_{FE}	DC Forward Current Gain	for ULN2801A V_{CE} = 2V, I_C = 350mA	 1000			 –	 2
C_i	Input Capacitance			15	25	pF	–
t_{PLH}	Turn-on Delay Time	0.5 V_i to 0.5 V_o		0.25	1	∝s	–
t_{PHL}	Turn-off Delay Time	0.5 V_i to 0.5 V_o		0.25	1	∝s	–
I_R	Clamp Diode Leakage Current	V_R = 50V T_{amb} = 70ϒC, V_R = 50V			50 100	∝A ∝A	6 6
V_F	Clamp Diode Forward Voltage	I_F = 350mA		1.7	2	V	7

Figure 8 : Collector Current as a Function of Saturation Voltage.

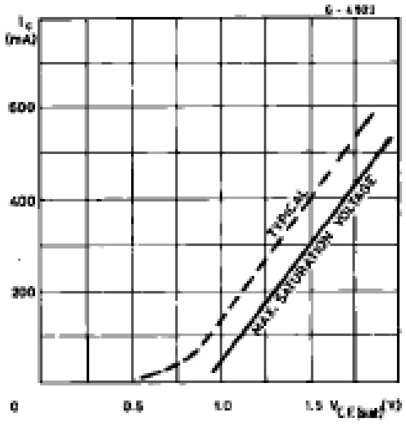


Figure 9 : Collector Current as a Function of Input Current.

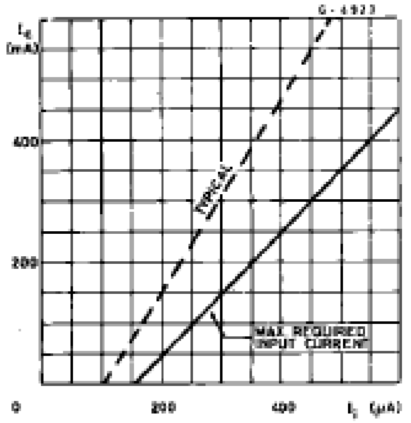


Figure 10 : Allowable Average Power Dissipation as a Function of Ambient Temperature.

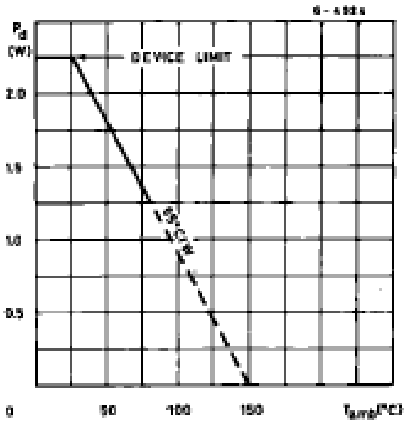


Figure 11 : Peak Collector Current as a Function of Duty Cycle.

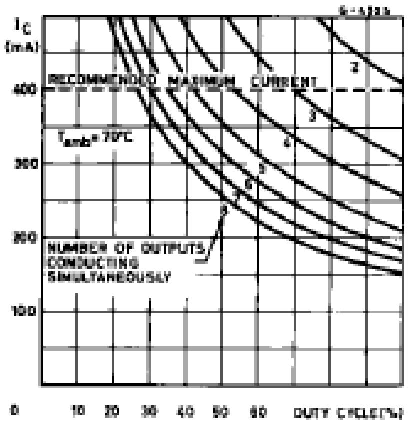


Figure 12 : Peak Collector Current as a Function of Duty.

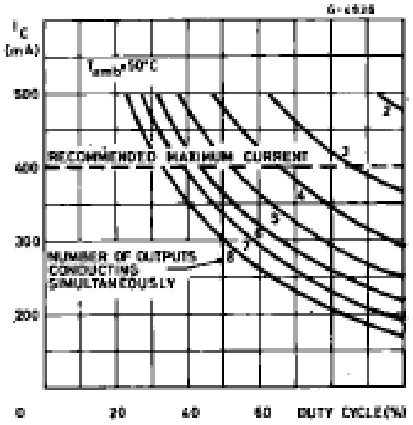


Figure 13 : Input Current as a Function of Input Voltage (for ULN2802A).

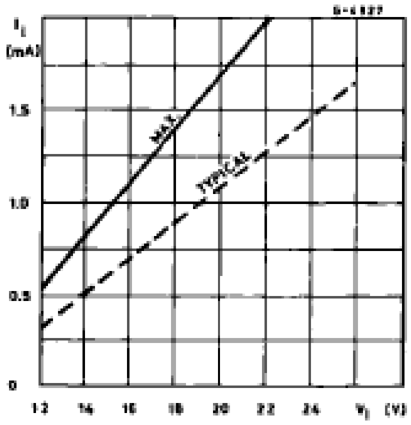


Figure 14 : Input Current as a Function of Input Voltage (for ULN2804A)

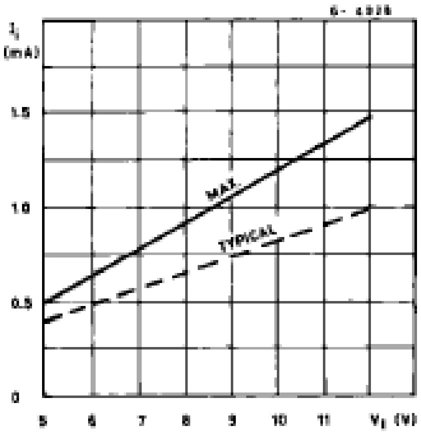


Figure 15 : Input Current as a Function of Input Voltage (for ULN2803A)

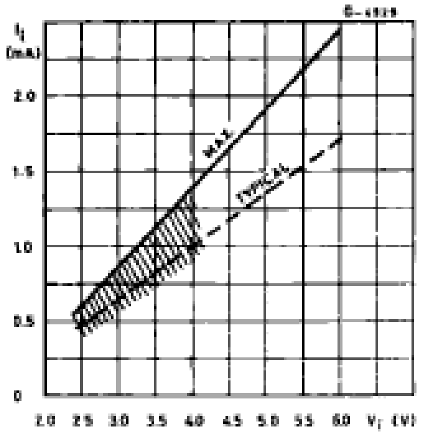
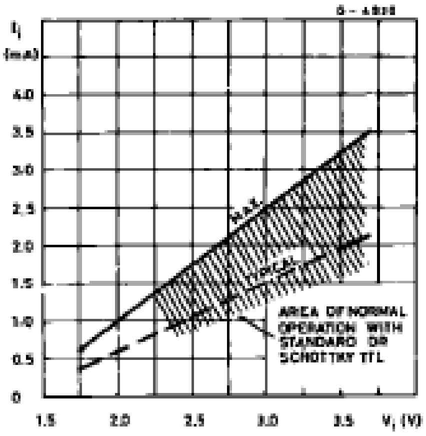
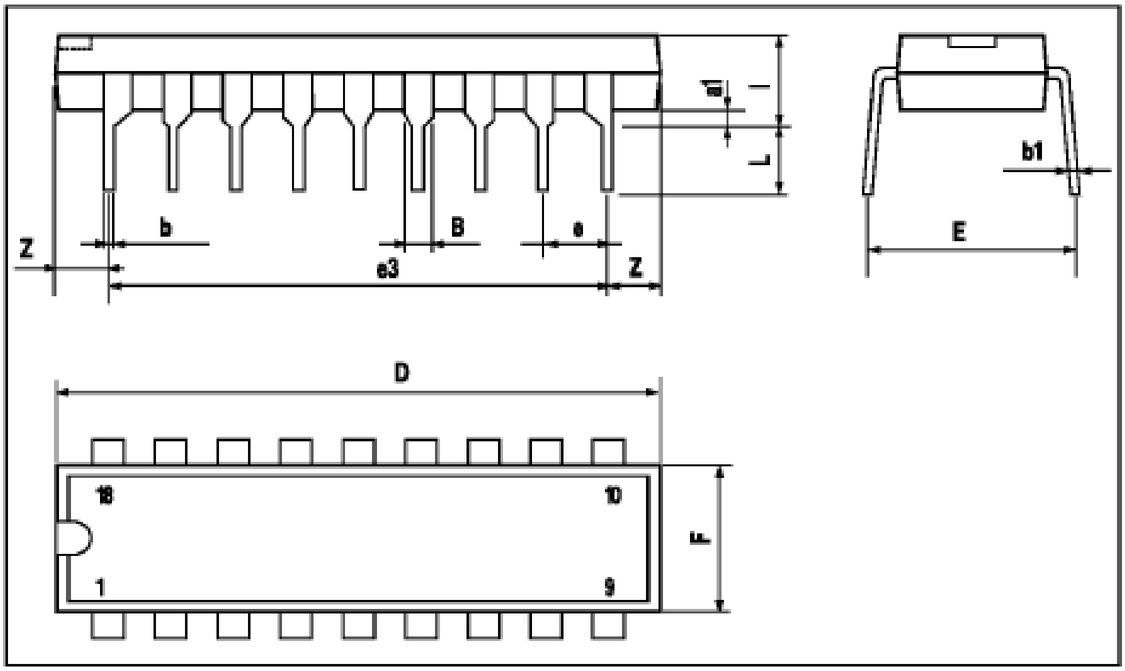


Figure 16 : Input Current as a Function of Input Voltage (for ULN2805A)



DIP18 PACKAGE MECHANICAL DATA

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
a1	0.254			0.010		
B	1.39		1.65	0.055		0.065
b		0.46			0.018	
b1		0.25			0.010	
D			23.24			0.915
E		8.5			0.335	
e		2.54			0.100	
e3		20.32			0.800	
F			7.1			0.280
I			3.93			0.155
L		3.3			0.130	
Z		1.27	1.59		0.050	0.063


a1
I
L
b1
Z
b
B
e
e3
Z
E
D
18
10
1
9
F