**FEATURE EXTRACTION FOR SPEECH RECOGNITION**

**ABSTRACT**

With the advancement of digital signal processing hardware and software, automatic speech recognition (ASR) has advanced significantly. But even with all these advancements, machines still fall short of human performance in terms of accuracy and speed, particularly when it comes to speaker-independent speech recognition. Therefore, speaker independent speech is the subject of a sizable percentage of speech recognition research nowadays recognizing difficulty. Due to the complexity of its applications and the limits of current methods of speech synthesis. In this paper, we briefly go over the speech recognition method known as signal modelling. It is then a summary of the fundamental techniques used in signal modelling. Additional frequently used temporal and we go into great length about spectral analysis feature extraction strategies.

**Keywords:** Automatic Speech Recognition (ASR), Digital Signal Processing (DSP), Signal Modelling, Temporal Vector, Feature Extraction.

**CHAPTER 1**

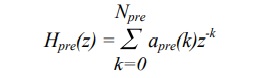
**INTRODUCTION**

Signal modelling and pattern matching are the two essential tasks that a speech recognition system does. Speech signal conversion into a set of parameters is referred to as signal modelling. Finding a parameter set from memory that closely resembles the parameter set retrieved from the input voice signal is known as pattern matching. To obtain the perceptually meaningful parameters i.e. parameters which are analogous to those used by human auditory system. To obtain the invariant parameters i.e. parameters which are robust to variations in channel, speaker and transducer. To obtain parameters that capture spectral dynamics, or changes of spectrum with time.

Four fundamental processes are involved in signal modelling: feature extraction, parametric transformation, statistical modelling, and spectral shaping. The act of transforming a spoken signal from a sound pressure wave to a digital stream and accentuating key frequency components is known as spectral shaping. The process of extracting different aspects from a speech signal includes power, pitch, and vocal tract configuration. By using a process of differentiation and concatenation, parameter transformation transforms these traits into signal parameters. Conversion of parameters in signal observation vectors is a necessary step in statistical modelling. This report focuses on feature extraction analysis approaches. Basic spectral shaping procedures are briefly covered. The spectral analysis methods for feature extraction are covered in detail.

**Spectral shaping**

Spectral shaping involves two basic operations: digitization, i.e., conversion of analog speech signal from sound pressure wave to digital signal; and digital filtering i.e., emphasizing important frequency components in the signal. This process is shown in Fig. The main purpose of digitization process is to produce a sampled data representation of speech signal with as high signal-to-noise ratio (SNR) as possible. Once signal conversion is complete, the last step of digital post filtering is most often executed using a Finite Impulse Response (FIR) filter given as,



Normally, a one coefficient digital filter known as pre-emphasis filter, is used,



A typical range of values for apre is [-1.0,-0.4]. The preemphasis filter boosts the signal spectrum approximately 20 dB per decade. Advantages of preemphasis filter. The voiced sections of speech signal naturally have a negative spectral slope (attenuation of approximately 20 dB per decade due to physiology of speech production system). The preemphasis filter serves to offset this natural slope before spectral analysis, thereby improving the efficiency of the analysis. The hearing is more sensitive above the 1-kHz region of the spectrum. The preemphasis filter amplifies this area of the spectrum. This assists the spectral analysis algorithm in modelling the perceptually important aspects of speech spectrum.

**Feature Extraction**

In speaker independent speech recognition, a premium is placed on extracting features that are somewhat invariant to changes in the speaker. So feature extraction involves analysis of speech signal. Broadly the feature extraction techniques are classified as temporal analysis and spectral analysis technique. In temporal analysis the speech waveform itself is used for analysis. In spectral analysis spectral representation of speech signal is used for analysis.

**Critical Band Filter Bank Analysis**

Critical Band Filter Bank Analysis It is one of the most fundamental concepts in speech processing. It can be regarded as crude model of the initial stages of transduction in human auditory system. Motivation for filter bank representation. According to "place theory" the position of maximum displacement along the basilar membrane for stimuli such as pure tones is proportional to the logarithm of the frequency of the tone.

**Cepstral Analysis**

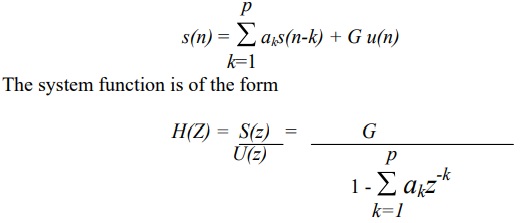
Cepstral Analysis This analysis technique is very useful as it provides methodology for separating the excitation from the vocal tract shape. In the linear acoustic model of speech production, the composite speech spectrum, consist of excitation signal filtered by a time-varying linear filter representing the vocal tract shape as shown.

**Mel Cepstrum Analysis**

Mel Cepstrum Analysis This analysis technique uses cepstrum with a nonlinear frequency axis following mel scale. For obtaining mel cepstrum the speech waveform s(n) is first windowed with analysis window w(n) and then its DFT S(k) is computed. The magnitude of S(k) is then weighted by a series of mel filter frequency responses whose center frequencies and bandwidth roughly match those of auditory critical band filters.

**Linear Predictive Coding (LPC)**

Linear Predictive Coding (LPC) Analysis, the basic idea behind the linear predictive coding (LPC) analysis is that a speech sample can be approximated as linear combination of past speech samples. By minimizing the sum of the squared differences (over a finite interval) between the actual speech samples and the linearly predicted ones, a unique set of predictor coefficients is determined. Speech is modeled as the output of linear, time-varying system excited by either quasi-periodic pulses (during voiced speech), or random noise (during unvoiced speech). The linear prediction method provides a robust, reliable, and accurate method for estimating the parameters that characterize the linear time-varying system representing vocal tract. Most recognition systems assume all pole model known as auto regressive (AR) model for speech production. The difference equation describing relation between speech samples s(n) and excitation u(n) for AR model is as follows,



**Perceptually Based Linear Predictive Analysis (PLP)**

Perceptually Based Linear Predictive Analysis (PLP) PLP analysis models perceptually motivated auditory spectrum by a low order all pole function, using the autocorrelation LP technique. Basic concept of PLP method is shown in block diagram of PLP. It involves two major steps: obtaining auditory spectrum, approximating the auditory spectrum by an all pole model. Auditory spectrum is derived from the speech waveform by critical-band filtering, equal loudness curve pre-emphasis, and intensity loudness root compression. Eighteen critical band filter outputs with their center frequencies equally spaced in bark domain, are defined.

**Temporal Analysis**

Temporal Analysis involves processing of the waveform of speech signal directly. It involves less computation compared to spectral analysis but limited to simple speech parameters, e.g. power and periodicity.

**Fundamental Frequency Estimation**

Fundamental Frequency Estimation Fundamental Frequency (f0) or pitch is defined as the frequency at which the vocal cords vibrate during a voiced sound. Fundamental frequency has long been difficult parameter to reliably estimate from the speech signal. Previously it was neglected for number of reasons, including large computational burden required for accurate estimation, the concern that unreliable estimation would be a barrier to achieving high performance, and difficulty in characterizing complex interactions between f0 and supra-segmental phenomenon. It is useful in speech recognition of tonal languages (e.g. Chinese) and languages that have some tonal components (e.g. Japanese). Fundamental frequency is often processed on logarithmic scale, rather than a linear scale to match the resolution of human auditory system.

**CHAPTER 2**

**LITERATURE SURVEY**

**[1] J. W. Picone, Proc. Of the IEEE, vol. 81, no.9, pp. 1215-1247, Sep. 1993.**

The principal target of talk affirmation zone is to make techniques and structures for talk commitment to machine. Talk is the basic techniques for correspondence between individuals. For reasons going from inventive enthusiasm about the segments for mechanical affirmation of human talk abilities to longing to robotize fundamental errands which require human machine associations and research in modified talk affirmation by machines has pulled in a ton of thought for quite a while. In light of genuine advances in authentic exhibiting of talk, customized talk affirmation structures today find expansive application in assignments that require human machine interface, for instance, modified call taking care of in telephone frameworks, and request based information systems that give invigorated travel information, stock esteem references, atmosphere reports, Data section, voice correspondence, access to information: travel, keeping cash, Commands, Avoinics, Automobile passage, talk elucidation, Handicapped people (amaze people) general store, railroad reservations**.**

**Summary:** Studied about signal modelling techniques in extracting features from any dataset.

**[2] L. R. Rabiner and R. W. Schafer, Englewood Cliffs, New Jersey: Prentice-Hall, 1978.** Spectral subtraction is used in this research as a method to remove noise from noisy speech signals in the frequency domain. This method consists of computing the spectrum of the noisy speech using the Fast Fourier Transform (FFT) and subtracting the average magnitude of the noise spectrum from the noisy speech spectrum. We applied spectral subtraction to the speech signal “Real graph”. A digital audio recorder system embedded in a personal computer was used to sample the speech signal “Real graph” to which we digitally added vacuum cleaner noise. The noise removal algorithm was implemented using Matlab software by storing the noisy speech data into Hanning time-widowed half-overlapped data buffers, computing the corresponding spectrums using the FFT, removing the noise from the noisy speech, and reconstructing the speech back into the time domain using the inverse Fast Fourier Transform (IFFT). The performance of the algorithm was evaluated by calculating the Speech to Noise Ratio (SNR). Frame averaging was introduced as an optional technique that could improve the SNR.

**Summary:** Studied about various digital signal processing techniques in removing noise from any digital signal.

**[3] D.O. Shaughnessy, India: University Press,2001:**

Today the wireless communications industry is heavily dependent upon advanced speech coding techniques, while the integration of personal computers and voice technology is poised for growth. In this revised and updated second edition, a timely overview of the science of speech processing helps you keep pace with these rapidly developing advances. Students of electrical engineering, along with computer scientists, systems engineers, linguists, audiologists, and psychologists, will find in this one concise volume an interdisciplinary introduction to speech communication. This reference book addresses how humans generate and interpret speech and how machines simulate human speech performance and code speech for efficient transmission. With a skillful blending of the basic principles and technical detail underlying speech communication, this broad-based book offers you essential insights into the field.

**Summary:** Studied about speech and voice processing techniques in Digital Signal Processing.

**[4] B. Gold and L. R. Rabiner,"Parallel processing techniques for estimating pitch periods of speech in the time domain," J. Acoust. Soc. America, vol.46, pt. 2, no. 2, pp 442-448, Aug. 1969:**

The paper presents detection of the vocal disorder suffered due to the reaction of antibiotics during the course of medical treatment by extracting pitch information of the speech. Extraction of pitch of the speech is an important task due to the presence of background noise. Primarily start and end points of speech is detected and thereafter pitch boundaries are recognized using autocorrelation technique. This paper emphasized on accurate end point analysis for detection of the vocal disorder suffered due to the reaction of antibiotics during the course of treatment by extracting pitch information in the phonetics of Indian regional Marathi language numerical.

**Summary:** Studied about the parallel processing techniques in digital signal processing.

**[5] H. Hermansky, B. A. Hanson, and H. Wakita, "Perceptually based linear predictive analysis of speech," Proc. IEEE Int. Conf. on Acoustic, speech, and Signal Processing," pp. 509-512, Aug.1985:**

Several significant advances have been made in continuous speech recognition over the last few years. In this chapter, we will discuss some of the current techniques in feature extraction and modeling for large vocabulary continuous speech recognition.

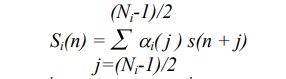
**SUMMARY:** Studied about Perceptually based linear predictive analysis of speech in acoustic, speech and signal processing.

**CHAPTER 3**

**EXISTING METHOD**

Critical Band Filter Bank Analysis It is one of the most fundamental concepts in speech processing. It can be regarded as crude model of the initial stages of transduction in human auditory system. Motivation for filter bank representation. According to "place theory" the position of maximum displacement along the basilar membrane for stimuli such as pure tones is proportional to the logarithm of the frequency of the tone. The experiments in human perception have shown that frequencies of a complex sound within a certain bandwidth of some nominal frequency cannot be individually identified unless one of the components of this sound falls outside the bandwidth. This bandwidth is known as critical bandwidth. Combination of these two theories gave rise to the critical band filter bank analysis technique.

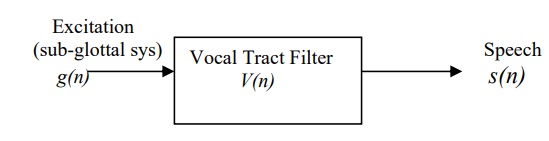
Critical bank filter bank is simply bank of linear phase FIR bandpass filters that are arranged linearly along the Bark (or mel) scale. The bandwidths are chosen to be equal to a critical bandwidth for corresponding center frequency. Bark i.e. critical Table 1 shows the critical filter banks based on Bark scale and mel scale. Each filter in digital filter bank is usually implemented as a linear phase filter so that the group delay for all filters is equal and the output signal from the filters are synchronized in time. The filter equations for linear phase filter implementation can be summarized as follows,



Where i(j) denotes j th coefficient for i th critical band filter. The output of this analysis is a vector of power values for each frame of data. These are usually combined with other parameters, such as total power, to form a signal measurement vector. The Filter bank attempts to decompose the signal into discrete set of spectral samples that contain information similar to what is presented to higher levels of processing in auditory system. Because the analysis technique is largely based on linear processing, it is generally robust to ambient noise.

**Cepstral Analysis**

Cepstral Analysis This analysis technique is very useful as it provides methodology for separating the excitation from the vocal tract shape. In the linear acoustic model of speech production, the composite speech spectrum, consist of excitation signal filtered by a time-varying linear filter representing the vocal tract shape as shown in



Hence in log domain the excitation and the vocal tract shape are superimposed, and can be separated. Cepstrum is computed by taking inverse discrete Fourier transform (IDFT) of logarithm (n) is defined as cepstrum. In speech recognition cepstral analysis is used for formant tracking and pitch (f0) detection. The samples of s(n) in its first 3ms describe v(n) and can be separated from the excitation the later is viewed as voiced if (n) exhibits sharp periodic pulses. Then the interval between these pulses is considered as pitch period. If no such structure is visible in (n), the speech is consider of magnitude of discrete Fourier transform finite length input signal as shown.

**Disadvantages:**

* Computationally complex.
* Takes more time for implementation

**CHAPTER 4**

**PROPOSED METHOD**

Linear Predictive Coding (LPC) Analysis the basic idea behind the linear predictive coding (LPC) analysis is that a speech sample can be approximated as linear combination of past speech samples. By minimizing the sum of the squared differences (over a finite interval) between the actual speech samples and the linearly predicted ones, a unique set of predictor coefficients is determined. Speech is modeled as the output of linear, time-varying system excited by either quasi-periodic pulses (during voiced speech), or random noise (during unvoiced speech). The linear prediction method provides a robust, reliable, and accurate method for estimating the parameters that characterize the linear time-varying system representing vocal tract. Most recognition systems assume all pole model known as auto regressive (AR) model for speech production. The difference equation describing relation between speech samples s(n) and excitation u(n) for AR model is as follows.

The basic approach is to find set of predictor coefficients that will minimize the mean squared error over a short segment of speech waveform. The resulting parameters are then assumed to be the parameters of the system function, H(z), in the model for speech production. The short-time average prediction error is defined.

There are three basic ways to solve above set of equations

1. Lattice method

2. Covariance method

3. Autocorrelation method

In speech recognition the autocorrelation method is almost exclusively used because of its computational efficiency and inherent stability. The autocorrelation method always produces a prediction filter whose zero lies inside the circle in z-plane.

For voiced regions of speech all pole model of LPC provides a good approximation to the vocal tract spectral envelope. During unvoiced and nasalized regions of speech the LPC model is less effective than voiced region. The computation involved in LPC processing is considerably less than cepstrum analysis. Thus the importance of method lies in ability to provide accurate estimates of speech parameters, and in its relative speed. A very important LPC parameter set which is derived directly from LPC coefficients is LPC cepstral coefficients cm.

Where G is the gain term in LPC model. This method is efficient, as it does not require explicit cepstral computation. Hence combines decorrelating property of cepstrum with computational efficiency of LPC analysis.

Linear prediction is a mathematical operation where future values of a discrete-time signal are estimated as a linear function of previous samples. In digital signal processing, linear prediction is often called linear predictive coding (LPC) and can thus be viewed as a subset of filter theory. LPC analyzes the speech signal by estimating the formants, removing their effects from the speech signal, and estimating the intensity and frequency of the remaining buzz. In speech coding applications, the LPC parameters are extracted frame-wise from the speech signal, typically at the rate of 50 frames/sec. For telephone speech sampled at 8 kHz, typically a 10'th order LPC analysis is performed. The LPC parameters are quantized prior to their trans- mission.

**CHAPTER 5**

**ADVANTAGES AND APPLICATIONS**

**Advantages:**

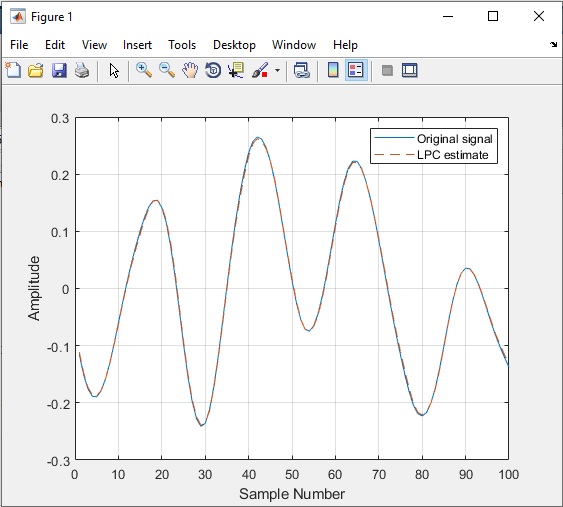
* The Linear Predictive Coding is better at extracting features from any dataset.
* The Linear Predictive Coding needs less number of iterations or steps to get features.
* The Linear Predictive Coding is easier to implement.
* The Linear Predictive Coding is a better choice when there is a cost constraint.

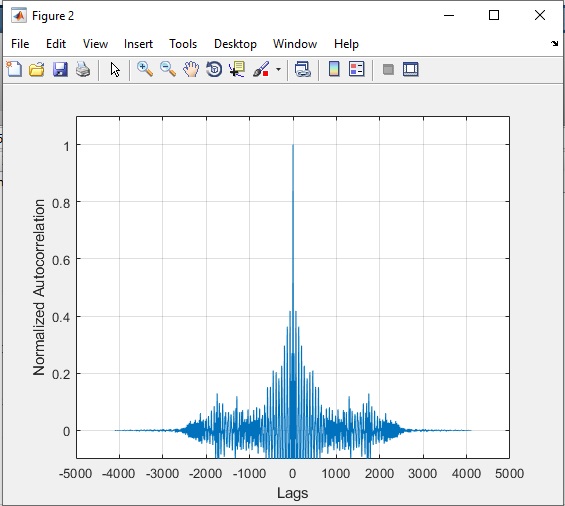
**Applications:**

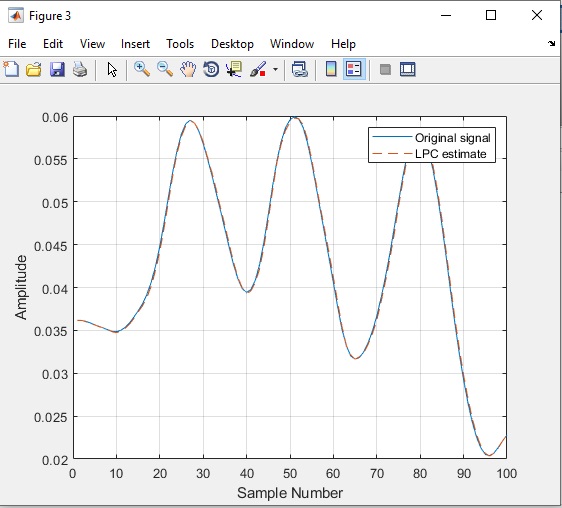
* Digital Signal Processing
* Medical Signal Processing
* Machine Learning
* Deep Learning
* Artificial Intelligence Environments

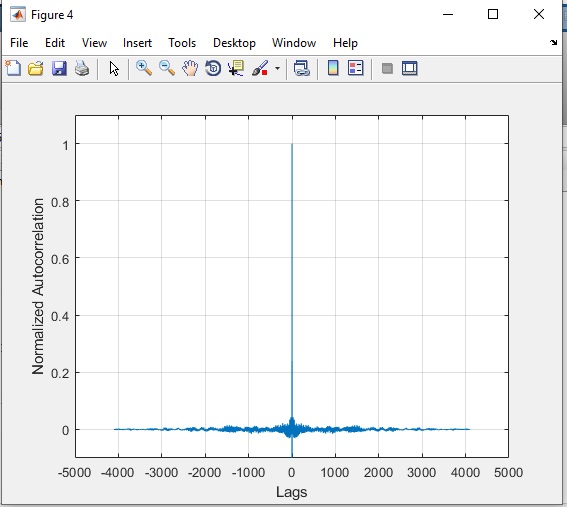
**CHAPTER 6**

**RESULTS**

****

****

****

****

**CHAPTER 7**

**CONCLUSION**

Here, in this paper we implemented the basic operations in speech recognition system have been discussed briefly. Different temporal and spectral analysis techniques for feature extraction have been studied in detail and following conclusions are drawn LP derived cepstral coefficients have decorrelating property of cepstrum and computational ease of LPC analysis. We can finally conclude that the LPC yielded better results than any other existing techniques.

**REFERENCES**

[1] J. W. Picone, "Signal modelling technique in speech recognition," Proc. Of the IEEE, vol. 81, no.9, pp. 1215-1247, Sep. 1993.

[2] L. R. Rabiner and R. W. Schafer, Digital Processing of Speech Signals. Englewood Cliffs, New Jersey: Prentice-Hall, 1978.

[3] D.O. Shaughnessy, Speech Communication: Human and Machine. India: University Press,2001.

[4] B. Gold and L. R. Rabiner,"Parallel processing techniques for estimating pitch periods of speech in the time domain," J. Acoust. Soc. America, vol.46, pt. 2, no. 2, pp 442-448, Aug. 1969.

[5] H. Hermansky, B. A. Hanson, and H. Wakita, "Perceptually based linear predictive analysis of speech," Proc. IEEE Int. Conf. on Acoustic, speech, and Signal Processing," pp. 509-512, Aug.1985.

[6] L. R. Rabiner and B. H. Juang, Fundamentals of Speech Recognition, Englewood Cliffs, New Jersey: Prentice- Hall, 1978.

[7] H. Hermansky, B. A. Hanson, and H. Wakita, "Perceptually based processing in automatic speech recognition," Proc. IEEE Int. Conf. on Acoustic, speech, and Signal Processing," pp. 1971-1974, Apr.1986.

[8] L. Roderer, The Physics and Psychophysics of Music: An Introduction, New York, Springer Verlag, 1995.

**BIBLIOGRAPHY**

**Introduction To Matlab**

What Is MATLAB?

The name MATLAB stands for Matrix Laboratory. The software is built up around vectors and matrices. This makes the software particularly useful for linear algebra but MATLAB is also a great tool for solving algebraic and differential equations and for numerical integration. MATLAB has powerful graphic tools and can produce nice pictures in both 2D and 3D. It is also a programming language, and is one of the easiest programming languages for writing mathematical programs. These factors make MATLAB an excellent tool for teaching and research.

MATLAB was written originally to provide easy access to matrix software developed by the LINPACK (linear system package) and EISPACK (Eigen system package) projects. It integrates computation, visualization, and programming environment. Furthermore, MATLAB is a modern programming language environment: it has sophisticated data structures, contains built-in editing and debugging tools, and supports object-oriented programming. MATLAB has many advantages compared to conventional computer languages (e.g., C, FORTRAN) for solving technical problems.

MATLAB abilities a family of add-on software program utility software application software program software utility software-unique solutions called toolboxes. Very essential to maximum customers of MATLAB, toolboxes assist you to studies and observe specialized technology. Toolboxes are entire collections of MATLAB abilities (M-files) that increase the MATLAB surroundings to remedy precise schooling of problems. Areas in which toolboxes are to be had embody signal processing, manipulate systems, neural networks, fuzzy correct judgment, wavelets, simulation, and hundreds of others.

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 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.

**Brief History of MATLAB:**

Cleve Moler, the chairman of the computer science department at the University of New Mexico, started developing MATLAB in the late 1970s. The first MATLAB® was not a programming language; it was a simple interactive matrix calculator. There were no programs, no toolboxes, no graphics and no ODEs or FFTs. He designed it to give his student’s access to LINPACK and EISPACK without them having to learn FORTRAN. It soon spread to other universities and found a strong audience within the applied mathematics community. The mathematical basis for the first version of MATLAB was a series of research papers by J. H. Wilkinson and 18 of his colleagues, published between 1965 and 1970 and later collected in Handbook for Automatic Computation, Volume II, Linear Algebra*,* edited by Wilkinson and C. Reinsch. These papers present algorithms, implemented in Algol 60, for solving matrix linear equation and Eigen value problems.

In the 1970s and early 1980s, I was teaching Linear Algebra and Numerical Analysis at the University of New Mexico and wanted my students to have easy access to LINPACK and EISPACK without writing FORTRAN programs. By “easy access,” I meant not going through the remote batch processing and the repeated edit-compile-link-load-execute process that was ordinarily required on the campus central mainframe computer. Jack little, an engineer, was exposed to it during a visit Moler made to Stanford University in 1983. Recognizing its commercial potential, he joined with Moler and Steve Bangert. They rewrote MATLAB in C and founded Math Works in 1984 to continue its development. These rewritten libraries were known as JACKPAC. In 2000, MATLAB was rewritten to use a newer set of libraries for matrix manipulation, LAPACK. MATLAB was first adopted by researchers and practitioners in control engineering, Little's specialty, but quickly spread to many other domains. It is now also used in education, in particular the teaching of linear algebra and numerical analysis, and is popular amongst scientists involved in video processing**.**

## **EISPACK and LINPACK**:

In 1970, a group of researchers at Argonne National Laboratory proposed to the U.S. National Science Foundation (NSF) to “explore the methodology, costs, and resources required to produce, test, and disseminate high-quality mathematical software and to test, certify, disseminate, and support packages of mathematical software in certain problem areas.” The group developed EISPACK (Matrix Eigen system Package) by translating the Algol procedures for Eigen value problems in the handbook into FORTRAN and working extensively on testing and portability. The first version of EISPACK was released in 1971 and the second in 1976.

In 1975, four of us Jack Dongarra, Pete Stewart, Jim Bunch, and myself proposed to the NSF another research project that would investigate methods for the development of mathematical software. A byproduct would be the software itself, dubbed LINPACK, for Linear Equation Package. This project was also centered at Argonne. LINPACK originated in FORTRAN; it did not involve translation from Algol. The package contained 44 subroutines in each of four numeric precisions. In a sense, the LINPACK and EISPACK projects were failures. We had proposed research projects to the NSF to “explore the methodology, costs, and resources required to produce, test, and disseminate high-quality mathematical software.” We never wrote a report or paper addressing those objectives. We only produced software.

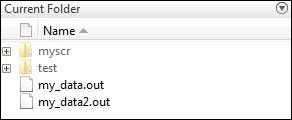
So, I studied Niklaus Wirth’s book Algorithms + Data Structures *=* Programs and learned how to parse programming languages. I wrote the first MATLAB an acronym for Matrix Laboratory in FORTRAN, with matrix as the only data type. The project was a kind of hobby, a new aspect of programming for me to learn and something for my students to use. There was never any formal outside support, and certainly no business plan. This first MATLAB was just an interactive matrix calculator. This snapshot of the start-up screen shows all the reserved words and functions. There are only 71. To add another function, you had to get the source code from me, write a FORTRAN subroutine, add your function name to the parse table, and recompile MATLAB.

**Starting MATLAB:**

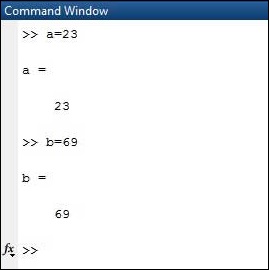
After logging into your account, you can enter MATLAB by double-clicking on the MATLAB shortcut icon (MATLAB 7.0.4) on your Windows desktop. When you start MATLAB, a special window called the MATLAB desktop appears. The desktop is a window that contains other windows. The major tools within or accessible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

* The Command Window
* The Command History
* The Workspace
* The Current Directory
* The Help Browser

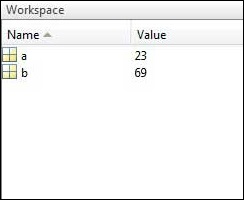
**Current Folder:** This panel allows you to access the project folders and files.



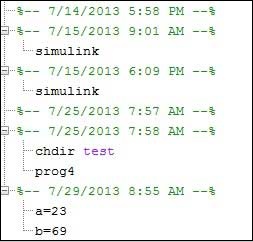
**Command Window:** This is the main area where commands can be entered at the command line. It is indicated by the command prompt (>>).



**Workspace:**  The workspace shows all the variables created and/or imported from files.



**Command History:** This panel shows or return commands that are entered at the command line.



**Help Browser:**

The critical way to get assist online is to use the MATLAB help browser, opened as a separate window every through clicking at the question mark photograph (?) on the computing tool toolbar, or through manner of typing assist browser on the spark off in the command window. The assist Browser is an internet browser blanketed into the MATLAB computing tool that shows a Hypertext Markup Language (HTML) files. The Help Browser consists of panes, the help navigator pane, used to find out information, and the show pane, used to view the information. Self-explanatory tabs apart from navigator pane are used to performs are searching out.

**MATLAB language:**

This is a high-level matrix/array language with control flow statements, functions, data structures, input/output, and object-oriented programming features. It allows both "programming in the small" to rapidly create quick and dirty throw-away programs, and "programming in the large" to create complete large and complex application programs.

**MATLAB working environment:**

This is the set of tools and facilities that you work with as the MATLAB user or programmer. It includes facilities for managing the variables in your workspace and importing and exporting data. It also includes tools for developing, managing, debugging, and profiling M-files, MATLAB's applications.

**MATLAB mathematical function library:**

This is a vast collection of computational algorithms ranging from elementary functions like sum, sine, cosine, and complex arithmetic, to more sophisticated functions like matrix inverse, matrix eigenvalues, Bessel functions, and fast Fourier transforms.

**MATLAB Application Program Interface (API):**

This is a library that allows you to write C and FORTRAN programs that interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

**MATLAB DESKTOP:**

MATLAB Desktop is the precept MATLAB utility window. The computing tool includes five sub home windows, the command window, the workspace browser, the modern-day-day list window, the command records window, and one or greater decide domestic windows, which is probably confirmed high-quality on the identical time due to the truth the client suggests a photo. The command window is in which the character types MATLAB instructions and expressions at the spark off (>>) and in which the output of these commands is displayed. MATLAB defines the workspace because the set of variables that the client creates in a bit consultation. The workspace browser suggests those variables and some facts about them. Double clicking on a variable within the workspace browser launches the Array Editor, which may be used to gain statistics and profits instances edit exceptional homes of the variable.

The modern-day-day-day Directory tab above the workspace tab suggests the contents of the cutting-edge list, whose path is shown inside the modern-day list window. For example, in the home windows on foot machine the path is probably as follows: C: MATLAB Work, indicating that listing “artwork” is a subdirectory of the number one list “MATLAB”; WHICH IS INSTALLED IN DRIVE C. Clicking on the arrow within the modern list window suggests a listing of these days used paths. Clicking at the button to the right of the window permits the individual to trade the present day listing. MATLAB uses a seeking out path to find out M-documents and one-of-a-type MATLAB associated documents, which can be put together in directories within the computer document tool. Any report run in MATLAB need to be dwelling in the modern-day-day listing or in a list that is on is looking for course. By default, the documents supplied with MATLAB and math works toolboxes are included inside the searching out direction. The first-rate manner to look which directories are on the searching out route. The satisfactory manner to appearance which directories are speedy the quest route, or to characteristic or regulate a searching for course, is to pick out outset path from the File menu the computing device, and then use the set course talk discipline. It is proper exercise to feature any generally used directories to the hunt route to avoid again and again having the exchange the cutting-edge-day listing.

The Command History Window contains a file of the instructions a person has entered in the command window, together with every contemporary-day and former MATLAB periods. Previously entered MATLAB instructions can be determined on and re-completed from the command statistics window thru proper clicking on a command or series of commands. This movement launches a menu from which to select numerous options similarly to executing the commands. This is useful to select out abilities options in addition to executing the instructions. This is a beneficial feature at the equal time as experimenting with numerous commands in a piece session.

**Using the MATLAB Editor to create M-Files:**

The MATLAB editorial manager is a literary substance proofreader particular for growing M-facts and a graphical MATLAB debugger. The supervisor can seem in a window through command facts technique for itself, or it is probably a right-clicking inside the PC. M-information this gadget signified through the use of the expansion .M, as in pixel up.M. The MATLAB editorial supervisor window has a few draws down menus for obligations collectively with sparing, seeing, and troubleshooting facts. Since it plays more than one easy test and furthermore affects utilization of shade to separate among exclusive variables of code, this article editorial supervisor is often supported due to reality the system of a need for composing and altering M-talents. To open the manager, type at enact opens the M-document filename. M in a supervisor window, sorted out for enhancing. As stated earlier than, the file should be inside the cutting-edge posting, or in a posting in the seeking out direction.

## **Features of MATLAB:**

Following are the basic features of MATLAB.

* It is a high-level language for numerical computation, visualization and application development.
* It also provides an interactive environment for iterative exploration, design and problem solving.
* It provides vast library of mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, numerical integration and solving ordinary differential equations.
* It provides built-in graphics for visualizing data and tools for creating custom plots.
* MATLAB's programming interface gives development tools for improving code quality maintainability and maximizing performance.
* It provides tools for building applications with custom graphical interfaces.
* It provides functions for integrating MATLAB based algorithms with external applications and languages such as C, Java, .NET and Microsoft Excel.

## **Uses of MATLAB:**

MATLAB is widely used as a computational tool in science and engineering encompassing the fields of physics, chemistry, math and all engineering streams. It is used in a range of applications including

* Signal Processing and Communications
* Video and Video Processing
* Control Systems
* Test and Measurement
* Computational Finance
* Computational Biology

**Applications of MATLAB:**

MATLAB can be used as a tool for simulating various electrical networks but the recent developments in MATLAB make it a very competitive tool for Artificial Intelligence, Robotics, Video processing, Wireless communication, Machine learning, Data analytics and whatnot. Though it’s mostly used by circuit branches and mechanical in the engineering domain to solve a basic set of problems its application is vast. It is a tool that enables computation, programming and graphically visualizing the results. The basic data element of MATLAB as the name suggests is the Matrix or an array. MATLAB toolboxes are professionally built and enable you to turn your imaginations into reality. MATLAB programming is quite similar to C programming and just requires a little brush up of your basic programming skills to start working with.

Below are a few applications of MATLAB –

* **Statistics and machine learning (ML)**

This toolbox in MATLAB can be very handy for the programmers. Statistical methods such as descriptive or inferential can be easily implemented. So is the case with machine learning. Various models can be employed to solve modern-day problems. The algorithms used can also be used for big data applications.

* **Curve fitting**

The curve fitting toolbox helps to analyze the pattern of occurrence of data. After a particular trend which can be a curve or surface is obtained, its future trends can be predicted. Further plotting, calculating integrals, derivatives, interpolation, etc. can be done.

* **Control systems**

Systems nature can be obtained. Factors such as closed-loop, open-loop, its controllability and observability, Bode plot, NY Quist plot, etc. can be obtained. Various controlling techniques such as PD, PI and PID can be visualized. Analysis can be done in the time domain or frequency domain.

* **Signal Processing**

Signals and systems and digital signal processing are taught in various engineering streams. But MATLAB provides the opportunity for proper visualization of this. Various transforms such as Laplace, Z, etc. can be done on any given signal. Theorems can be validated. Analysis can be done in the time domain or frequency domain. There are multiple built-in functions that can be used.

* **Mapping**  
  Mapping has multiple applications in various domains. For example, in Big Data, the Map Reduce tool is quite important which has multiple applications in the real world. Theft analysis or financial fraud detection, regression models, contingency analysis, predicting techniques in social media, data monitoring, etc. can be done by data mapping.
* **Deep learning**

It’s a subclass of machine learning which can be used for speech recognition, financial fraud detection, and medical video analysis. Tools such as time-series, Artificial neural network (ANN), Fuzzy logic or combination of such tools can be employed.

* **Financial analysis**

An entrepreneur before starting any endeavor needs to do a proper survey and the financial analysis in order to plan the course of action. The tools needed for this are all available in MATLAB. Elements such as profitability, solvency, liquidity, and stability can be identified. Business valuation, capital budgeting, cost of capital, etc. can be evaluated.

* **Video processing**

The most common application that we observe almost every day are bar code scanners, selfie (face beauty, blurring the background, face detection), video enhancement, etc. The digital video processing also plays quite an important role in transmitting data from far off satellites and receiving and decoding it in the same way. Algorithms to support all such applications are available.

* **Text analysis**

Based on the text, sentiment analysis can be done. Google gives millions of search results for any text entered within a few milliseconds. All this is possible because of text analysis. Handwriting comparison in forensics can be done. No limit to the application and just one software which can do this all.

* **Electric vehicles designing**

Used for modeling electric vehicles and analyze their performance with a change in system inputs. Speed torque comparison, designing and simulating of a vehicle, whatnot.

* **Aerospace**

This toolbox in MATLAB is used for analyzing the navigation and to visualize flight simulator.

* **Audio toolbox**

Provides tools for audio processing, speech analysis, and acoustic measurement. It also provides algorithms for audio and speech feature extraction and audio signal transformation.

**COMMUNICATION:**

Communications System Toolbox™ offers algorithms and gear for the layout, simulation, and analysis of communications systems. These capabilities are furnished as MATLAB ® features, MATLAB System gadgets™, and Simulink ® blocks. The machine toolbox includes algorithms for source coding, channel coding, interleaving, modulation, equalization, synchronization, and channel modeling. Tools are supplied for bit blunders charge evaluation, producing eye and constellation diagrams, and visualizing channel characteristics. The machine toolbox additionally provides adaptive algorithms that allow you to version dynamic communications structures that use OFDM, OFDMA, and MIMO techniques. Algorithms support fixed-point facts arithmetic and C or HDL code era.

**Key Features**

▪ Algorithms for designing the physical layer of communications systems, which includes supply coding, channel coding, interleaving, modulation, channel fashions, MIMO, equalization, and synchronization

▪ GPU-enabled System objects for computationally intensive algorithms together with Turbo, LDPC, and Viterbi decoders

▪ Interactive visualization equipment, consisting of eye diagrams, constellations, and channel scattering capabilities

▪ Graphical tool for evaluating the simulated bit mistakes rate of a machine with analytical outcomes

▪ Channel models, consisting of AWGN, Multipath Rayleigh Fading, Rician Fading, MIMO Multipath Fading, and

LTE MIMO Multipath Fading

▪ Basic RF impairments, along with nonlinearity, section noise, thermal noise, and section and frequency offsets

▪ Algorithms available as MATLAB features, MATLAB System objects, and Simulink blocks

▪ Support for fixed-point modeling and C and HDL code technology

**System Design, Characterization, and Visualization:**

The layout and simulation of a communications gadget requires analyzing its reaction to the noise and interference inherent in real-world environments, reading its behavior the usage of graphical and quantitative manner, and determining whether the resulting overall performance meets requirements of acceptability. Communications System Toolbox implements a selection of obligations for communications machine layout and simulation. Many of the functions, System objects™, and blocks inside the device toolbox perform computations associated with a specific thing of a communications gadget, consisting of a demodulator or equalizer. Other talents are designed for visualization or evaluation.

**System Characterization**

The system toolbox offers several standard methods for quantitatively characterizing system performance:

▪ Bit error rate (BER) computations

▪ Adjacent channel power ratio (ACPR) measurements

▪ Error vector magnitude (EVM) measurements

▪ Modulation error ratio (MER) measurements

Because BER computations are fundamental to the characterization of any communications system, the system toolbox provides the following tools and capabilities for configuring BER test scenarios and accelerating BER simulations:

**BER tool**— A graphical user interface that enables you to analyze BER performance of communications systems. You can analyze performance via a simulation-based, semi analytic, or theoretical approach.

**Error Rate Test Console** — A MATLAB object that runs simulations for communications systems to measure error rate performance. It supports user-specified test points and generation of parametric performance plots and surfaces. Accelerated performance can be realized when running on a multi core computing platform.

**Multi core and GPU acceleration** — A capability provided by Parallel Computing Toolbox™ that enables you to accelerate simulation performance using multi core and GPU hardware within your computer.

**Distributed computing and cloud computing support** — Capabilities provided by Parallel Computing Toolbox and MATLAB Distributed Computing Server™ that enable you to leverage the computing power of your server farms and the Amazon EC2 Web service. Performance Visualization. The system toolbox provides the following capabilities for visualizing system performance:

**Channel visualization tool** — For visualizing the characteristics of a fading channel

**Eye diagrams and signal constellation scatter plots** — for a qualitative, visual understanding of system behavior that enables you to make initial design decisions

**Signal trajectory plots** — for a continuous picture of the signal’s trajectory between decision points

**BER plots** — for visualizing quantitative BER performance of a design candidate, parameterized by metrics such as SNR and fixed-point word size

**Analog and Digital Modulation**

Analog and digital modulation strategies encode the facts circulation into a sign this is appropriate for transmission. Communications System Toolbox presents some of modulation and corresponding demodulation abilities. These talents are available as MATLAB features and gadgets, MATLAB System Modulation sorts provided by the toolbox are:

**Source and Channel Coding**

Communications System Toolbox affords source and channel coding talents that can help you develop and compare communications architectures fast, enabling you to discover what-if eventualities and avoid the need to create coding competencies from scratch.

**Source Coding**

Source coding, also referred to as quantization or signal formatting, is a manner of processing facts a good way to lessen redundancy or prepare it for later processing. The system toolbox offers a diffusion of styles of algorithms for imposing source coding and interpreting, inclusive of:

▪ Quantizing

▪ Companding (*µ*-law and A-law)

▪ Differential pulse code modulation (DPCM)

▪ Huffman coding

▪ Arithmetic coding

**Channel Coding**

▪ orthogonal area-time block code (OSTBC) (encoder and decoder for MIMO channels)

▪ Turbo encoder and decoder examples

The gadget toolbox offers application functions for developing your personal channel coding. You can create generator polynomials and coefficients and syndrome deciphering tables, in addition to product parity-take a look at and generator matrices.

The system toolbox additionally presents block and convolutional interleaving and deinters leaving functions to reduce facts errors as a result of burst mistakes in a conversation machine:

**Block,** including General block interleaver, algebraic interleaver, helical scan interleaver, matrix interleaver, and random interleaver.

**Convolutional,** including General multiplexed interleaver, convolutional interleaver, and helical interleaver

**Channel Modeling and RF Impairments**

Channel Modeling

Communications System Toolbox provides algorithms and tools for modeling noise, fading, interference, and different distortions which might be commonly found in communications channels. The system toolbox supports the subsequent styles of channels:

▪ Additive white Gaussian noise (AWGN)

▪ Multiple-enter multiple-output (MIMO) fading

▪ Single-enter single-output (SISO), Rayleigh, and Rician fading

▪ Binary symmetric

A MATLAB channel object provides a concise, configurable implementation of channel models, enabling you to

specify parameters such as:

▪ Path delays

▪ Average path gains

▪ Maximum Doppler shifts

▪ K-Factor for Rician fading channels

▪ Doppler spectrum parameters

For MIMO systems, the MATLAB MIMO channel object expands these parameters to also include:

▪ Number of transmit antennas (up to 8)

▪ Number of receive antennas (up to 8)

▪ Transmit correlation matrix

▪ Receive correlation matrix

To combat the effects noise and channel corruption, the system toolbox provides block and convolutional coding and decoding techniques to implement error detection and correction. For simple error detection with no inherent correction, a cyclic redundancy check capability is also available. Channel coding capabilities provided by the system toolbox include:

▪ BCH encoder and decoder

▪ Reed-Solomon encoder and decoder

▪ LDPC encoder and decoder

▪ Convolutional encoder and Viterbi decoder

****

**RF Impairments**

To model the effects of a non-ideal RF front end, you can introduce the following impairments into your communications system, enabling you to explore and characterize performance with real-world effects:

▪ Memory less nonlinearity

▪ Phase and frequency offset

▪ Phase noise

▪ Thermal noise

You can include more complex RF impairments and RF circuit models in your design using SimRF™.

****

**Equalization and Synchronization**

Communications System Toolbox lets you discover equalization and synchronization strategies. These techniques are usually adaptive in nature and tough to design and symbolize. The machine toolbox affords algorithms and tools that will let you swiftly select the proper approach on your communications machine. Equalization To compare one-of-a-kind techniques to equalization, the device toolbox offers you with adaptive algorithms which include:

▪ LMS

▪ Normalized LMS

▪ Variable step LMS

▪ Signed LMS

▪ MLSE (Viterbi)

▪ RLS

▪ CMA

These adaptive equalizers are available as nonlinear decision feedback equalizer (DFE) implementations and as

Linear (symbol or fractionally spaced) equalizer implementations.

**Synchronization**

The device toolbox provides algorithms for each service segment synchronization and timing phase synchronization. For timing section synchronization, the machine toolbox presents a MATLAB Timing Phase Synchronizer object that offers the following implementation techniques:

▪ Early-late gate timing method

▪ Gardner’s method

▪ Fourth-order nonlinearity method

**Stream Processing in MATLAB and Simulink**

Most verbal exchange structures cope with streaming and frame-primarily based statistics using a aggregate of temporal processing and simultaneous multi frequency and multichannel processing. This form of streaming multidimensional processing can be visible in superior communication architectures consisting of OFDM and MIMO. Communications System Toolbox enables the simulation of advanced communications structures via helping move processing and frame-based simulation in MATLAB and Simulink. In MATLAB, circulate processing is enabled by way of System items™, which use MATLAB objects to symbolize time-based and facts-driven algorithms, sources, and sinks. System objects implicitly manipulate many information of flow processing, including information indexing, buffering, and management of set of rules state. You can mix System gadgets with fashionable MATLAB functions and operators. Most System items have a corresponding Simulink block with the identical abilities. Simulink handles circulation processing implicitly with the aid of coping with the float of information thru the blocks that make up a Simulink model. Simulink is an interactive graphical environment for modeling and simulating dynamic systems that uses hierarchical diagrams to symbolize a machine version. It includes a library of widespread-reason, predefined blocks to represent algorithms, resources, sinks, and device hierarchy.

**Implementing a Communications System**

Fixed-Point Modeling Many communications systems use hardware that requires a fixed-point representation of your design.

Communications System Toolbox supports fixed-point modeling in all relevant blocks and System objects™ with tools that help you configure fixed-point attributes.

Fixed-point support in the system toolbox includes:

▪ Word sizes from 1 to 128 bits

▪ Arbitrary binary-point placement

▪ Overflow handling methods (wrap or saturation)

▪ Rounding methods: ceiling, convergent, floor, nearest, round, simplest, and zero

Fixed-Point Tool in Simulink Fixed Point™ facilitates the conversion of floating-point data types to fixed point. For configuration of fixed-point properties, the tool tracks overflows and maxima and minima.

**Code Generation**

Once you've got advanced your set of rules or communications device, you can robotically generate C code from it for verification, rapid prototyping, and implementation. Most System gadgets, functions, and blocks in Communications System Toolbox can generate ANSI/ISO C code the use of MATLAB Coder™, Simulink Coder™, or Embedded Coder™. A subset of System gadgets and Simulink blocks also can generate HDL code. To leverage present highbrow belongings, you can choose optimizations for specific processor architectures and integrate legacy C code with the generated code.

You can also generate C code for both floating-point and fixed-point data types.

DSP Proto typing DSPs are used in communication system implementation for verification, rapid prototyping, or final hardware implementation. Using the processor-in-the-loop (PIL) simulation capability found in Embedded Coder, you can verify generated source code and compiled code by running your algorithm’s implementation code on a target processor. FPGA Prototyping

FPGAs are used in communication systems for implementing high-speed signal processing algorithms. Using the FPGA-in-the-loop (FIL) capability found in HDL Verifier™, you can test RTL code in real hardware for any existing HDL code, either manually written or automatically generated HDL code.