**Development of a New Biometric Authentication**

**Approach Based on Electrocardiogram Signals**

**ABSTRACT**

Current research shows that one of the best methods for authenticating human beings is biometrics. In this paper, the heartbeat biometric, also called Electrocardiograph (ECG), is proposed. The heartbeat biometric is chosen because the ECGs is unique. The purpose of this study is to find the best biometric features that able to identify a person, given the extractions and classification algorithms for the heartbeat biometric signal. Depending on a literature study, we proposed a new more efficient technique based on the wave modeling of the ECG signal for features extraction where this features is used as an inputs for pattern recognition classifier. This proposed methodology is tested on a real experimental ECG data, the processing of ECG signals must include signals acquisition, signal filtering & pre-processing using the most familiar and multipurpose MATLAB software. The results obtained are very accurate feature compared to features extracted from classical parameters. Therefore, the wave modeling for extracting features is the most efficient and accurate way to obtain the best results in classification. As for future work, automatic heartbeat classification is essential for real-time applications to identify person

**Keywords:** Biometrics, Electrocardiogram, Heartbeat Recognition, Waves Modeling

**CHAPTER 1**

**INTRODUCTION**

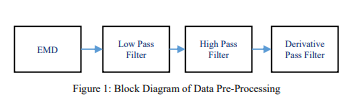
Today, life engages technology in multiple ways, thus authentication in human technologies is very important. Secure and reliable authentication is in high demand. However, traditional methods for authentication such as face recognition, voice recognition and passwords are now outdated because faces are available in social media and couldn’t differentiate between two twins, and voices can be easily recorded from calls. However, ECG signal is a universal characteristic [1]. The Electrocardiogram (ECG) is the recording of electrical activity of human heart using electrodes placed on the skin over a period of time. The shape of the waveform reveals the current state of the heart and it offers helpful information regarding the rhythm and function of the heart. There are 3 main components to an ECG: P wave, QRS complex and T wave [2]. Recently, the possibility of using this ECG signal as a biometric tool has been suggested because the composition and activity of the human heart is unique, stable, easy to collect, have a high performance and it’s socially accepted. Its validity is well supported by the fact that both the physiological and geometrical differences of the heart under different subjects reveal certain uniqueness in the signal characteristics due to existing differences in morphology among individuals.

1. DATA ACQUISITION

The ECG data used in this research is obtained from real experimental data. The total number of person is 10 subjects recorded over 210 seconds long. The subjects are 6 men aged 20-30 years and 4 women aged 20-25 years. The ECGs were recorded via a commercial ECG device (500 Hz as sampling rate and 500 dB for gain). The number of recordings for each person are obtain from ECG lead 1. In this step signal is being acquired and stored in database, which is further used by system for identification purposes.

1. DATA PRE-PROCESSING ECG

data collected usually contain noise. Due to presence of noise the feature extraction and classification becomes less accurate. To prevent misclassification, the ECG data must be processed. The first step must be to identify the noisy sources [3]. In this research, a cascaded digital filters configuration (empirical mode decomposition, low pass filter, high pass filter and derivative base filter), as shown in figure 1, is used for removal of three major noises of baseline drift, power line interference and EMG noise.



EMD is a decomposition technique that allows to represent a signal through the sum of functions derived from the latter, called Intrinsic Mode Function (IMF). The individual IMFs are obtained through a sifting operation. This is an iterative operation. The basic steps to achieve decomposition in IMF are: a. Identify local extremes. Especially, the maximum and minimum local values of the signal must be evaluated separately.

b. Evaluate the upper and lower envelope of the signal through the application of a cubic spline interpolation function of the data obtained in the previous point.

c. Compute the mean envelope, obtaining m. Then subtract the mean from the input signal.

d. Evaluate a term condition. If this is respected, then the difference between the input signal and m is the IMF and the next one is evaluated considering as a signal the difference between the input one and the IMF obtained. Otherwise, the process on the residual is repeated [4].

**CHAPTER 2**

**LITERATURE SURVEY**

**[1] Ala Abdulhakim Alariki, Sayed Mahmoud Alavy, Mohammad Reza Yousufi, Mohammad Tareq Aziz and Christine Murray, A Review Study of Heartbeat Biometric Authentication, Volume 13, Number 8, August 2018**

Today, life engages technology in multiple ways, thus authentication in human technologies is very

important. Secure and reliable authentication is in high demand. However, traditional methods for

authentication such as passwords and tokens are now outdated because it is possible to steal, lose and share such authentication methods. Current research shows that one of the best methods for authenticating human beings is biometrics. In this paper, the heartbeat biometric, also called Electrocardiographic (ECG), is proposed. The heartbeat biometric is chosen because unique human ECGs cannot be falsely created or replicated. While other biometric methods, such as face recognition, can be compromised by user photographs, or fingerprints, which can be compromised by use of fake fingers, the ECG signal is based on the individualized mechanical movements of each human heart, which features contain unique physiological information. The purpose of this paper is, then, to review various relevant, recent works that study the heartbeat biometric to find the best biometrics features, given the extractions and classification algorithms for the heartbeat biometric signal. This paper concludes that the morphological (P wave) feature is recommended as the most important feature and the Neural Network (NN) classifier is the most reliable classification with the highest performance accuracy for heartbeat biometric. Therefore, to achieve highest accuracy and result for authenticating through heartbeat biometric, it is recommended to consider the mentioned feature extractions and classification.

**Summary:** Studied about Heartbeat Biometric Authentication.

**Kiran KumarPatro and P. RajeshKumar,** **Effective Feature Extraction of ECG for Biometric Application, Procedia Computer Science, Volume 115, 2017, Pages 296-306**

Biometric systems performing identity recognition based upon extracted informative data from an individual are vital for security applications. The vital characteristics of an ECG signal depend upon its Characteristic points’ P, Q, R, S and T. In this paper, an effective feature extraction method is proposed, in which for each record of ECG, the best 6-PQRST fragments are extracted according to priority basis and their positions are normalized. A total of 72 different features are calculated, finally the performance of feature set is examined and compared using ANN. The proposed algorithm is tested for MIT-BIH ECG ID database signals.

**Summary:** Studied about Effective Feature Extraction of ECG for Biometric Application

**Gaganpreet Kaur, Dr. Dheerendra Singh and Simranjeet Kaur, Electrocardiogram (ECG) as a Biometric characteristic: A Review, International Journal ofEmerging Research in Management &Technology ISSN: 2278-9359 (Volume-4, Issue-5)**

In the last decades, researchers have shown the potential of using Electrocardiogram (ECG) as a biometric trait due to its uniqueness and hidden nature. However, despite the great number of approaches found in the literature, no agreement exists on the most appropriate methodology. This paper presents a systematic review of data acquisition methods, aiming to understand the impact of some variables from the data acquisition protocol of an ECG signal in the biometric identification process. We searched for papers on the subject using Scopus, defining several keywords and restrictions, and found a total of 121 papers. Data acquisition hardware and methods vary widely throughout the literature. We reviewed the intrusiveness of acquisitions, the number of leads used, and the duration of acquisitions. Moreover, by analyzing the literature, we can conclude that the preferable solutions include: (1) the use of off-the-person acquisitions as they bring ECG biometrics closer to viable, unconstrained applications; (2) the use of a one-lead setup; and (3) short-term acquisitions as they required fewer numbers of contact points, making the data acquisition of benefit to user acceptance and allow faster acquisitions, resulting in a user-friendly biometric system. Thus, this paper reviews data acquisition methods, summarizes multiple perspectives, and highlights existing challenges and problems. In contrast, most reviews on ECG-based biometrics focus on feature extraction and classification methods.

**Summary:** Studied about Electrocardiogram (ECG) as a Biometric characteristic

**Rupert Faltermeier, Ingo R. Keck, Ana Maria Tomé and Carlos G. Puntonet, Empirical Mode Decomposition -An Introduction, July 2010, DOI: 10.1109/IJCNN.2010.5596829**

Due to external stimuli, biomedical signals are in general non-linear and non-stationary. Empirical Mode Decomposition in conjunction with a Hilbert spectral transform, together called Hilbert-Huang Transform, is ideally suited to extract essential components which are characteristic of the underlying biological or physiological processes. The method is fully adaptive and generates the basis to represent the data solely from these data and based on them. The basis functions, called Intrinsic Mode Functions (IMFs) represent a complete set of locally orthogonal basis functions whose amplitude and frequency may vary over time. The contribution reviews the technique of EMD and related algorithms and discusses illustrative applications.

**Summary:** Studied about Empirical Mode Decomposition(EMD).

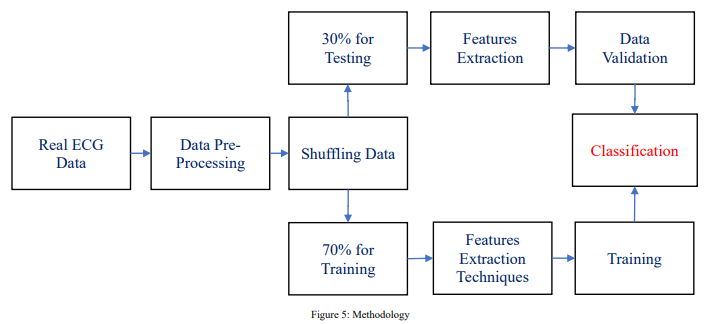
**Shweta H. Jambukia, Vipul K. Dabhi and Harshadkumar B. Prajapati, Classification of ECG signals using Machine Learning Techniques a Survey, 978-1-4673-6911-415,2015**

Electrocardiogram (ECG) signals are the impulses generated by the heart which are used to analyze the proper functioning of heart. Our work deals with the efficient analysis of Electrocardiogram (ECG) signals imported from MIT-BIH database into MATLAB platform, generation of the imported ECG signal, pre-processing the generated signal to remove the noises mainly the baseline wandering and power line interference from which features are extracted. For adequate study of the ECG signal Daubechies and Haar wavelet techniques are compared. Proper decomposition of the signal is achieved using Db4 and Db5 Daubechies wavelets as their scaling functions are analogous to ECG signal. PAN TOMPKINSONS algorithm is considered in our study as it serves best for precise identification of most prominent features namely QRS complexes, RR interval's as they constitute the major data required for clinical analysis and research. After feature extraction ECG signals are trained using machine learning techniques for detecting the presence of Arrhythmia using different classifiers adopting Weka software.

**SUMMARY:** Studied about Classification of ECG signals using Machine Learning Techniques.

**CHAPTER 3**

**EXISTING METHOD**



After demonizing the signal, the data is then divided into train and test sets as shown in figure 5: i. In training phase: 70% of the filtered data is used as the training samples for classification. ii. In testing phase: 30% of the filtered data is use as testing data for validation

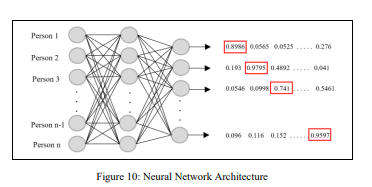
1. FEATURE EXTRACTION The feature extraction stage is the key to the success in the heartbeat classification using in the ECG signal. The feature can be extracted in various form directly from the ECG signal’s morphology in the time domain and/or in the frequency domain or from the cardiac rhythm.
2. 1) Peak detection: The third step is detection of peaks locations and boundaries of three types of waves, P wave, QRS complex, and T wave which are used in heartbeat signals [1] . Generally, in ECG, the major characteristic wave is R-peak because it has the higher amplitude. After finding R-peak location, other components P, Q, S, and T are detected by taking R-peak location as reference and tracing from R peak relative position

2) Segmentation: After identifying the points, ECG waveform is segmented into individual heartbeat, as shown in figure 8, and decomposed as a form of sine wave

3) Wave Modeling: The purpose of the feature extraction is to have identical model from same user and different model from different users. This is one of the main step of biometric heart beat because a vector features will be composed and extracted from heartbeat signals as input. We use “cftool” in Matlab to choose the best model of curve fitting and then we generate the code. (1) shows the general model of sum of sine method with degree 5 that is used for modeling.

B. BIOMETRIC TEMPLATE IN DATABASE In this step, the vector features of each individual is stored in database. Then the stored data is compared with the entered data from testing phase to identify person for decision making [1].

IV.CLASSIFICATION Classification is used to classify entered data in different set of classes that would be easy to compare with stored data. There are different methods for classification [1] . In this study, the biometric identification is done by feeding the input data of ECG features to an Artificial Neural Network (ANN) [6] . At the primary stage the neural network has to be trained with the ECG data of different persons. Then afterwards the neural network generated from the training is used for biometric identification of the persons. Figure 10 shows the architecture of neural network that is composed by features as an input, hidden layers and output that identify the person.



**CHAPTER 4**

**PROPOSED METHOD**

**CHAPTER 5**

**ADVANTAGES AND APPLICATIONS**

**Advantages:**

* High accuracy
* Continuous authentication
* Health monitoring
* Cost-effective
* Non-invasive

**Applications:**

1. Applied in DSP applications.

2. ECG Peak Detection.

3. Bio-Medical Signal Processing.

4. Image Processing.

**CHAPTER 6**

**RESULTS**

**CHAPTER 7**

**CONCLUSION**

ECG being none mimic able can more accurately identify a person and can offer more robust and effective human identification system. In order to provide more accuracy in identification and verification process of individual, this paper provides an overview of major steps in ECG signal analysis of de-noising ECG, characteristic points identification, feature extraction and effective feature extraction finally classification. The review recognized different methods of extracting features of the heartbeat signals and compared based on the accuracy result. A good feature extraction methodology can accurately work for biometric applications.

**REFERENCES**

[1] Ala Abdulhakim Alariki, Sayed Mahmoud Alavy, Mohammad Reza Yousufi, Mohammad Tareq Aziz and Christine Murray, A Review Study of Heartbeat Biometric Authentication, Volume 13, Number 8, August 2018

[2] Kiran KumarPatro and P. RajeshKumar, Effective Feature Extraction of ECG for Biometric Application, Procedia Computer Science, Volume 115, 2017, Pages 296-306

[3] Gaganpreet Kaur, Dr. Dheerendra Singh and Simranjeet Kaur, Electrocardiogram (ECG) as a Biometric Characteristic: A Review, International Journal of Emerging Research in Management &Technology ISSN: 2278-9359 (Volume-4, Issue-5)

[4] Rupert Faltermeier, Ingo R. Keck, Ana Maria Tomé and Carlos G. Puntonet, Empirical Mode Decomposition - An Introduction, July 2010, DOI: 10.1109/IJCNN.2010.5596829

[5] Shweta H. Jambukia, Vipul K. Dabhi and Harshadkumar B. Prajapati, Classification of ECG signals using Machine Learning Techniques a Survey, 978-1-4673- 6911-415,2015

[6] Alaa Daher, Ghaleb Hoblos, Yahya Chetouani, Mohamad Khalil, modified fuzzy c-means combined with Neural Network based Fault Diagnosis Approach for a distillation column, IEEE International Multidisciplinary Conference on Engineering Technology (IMCET), 2016

**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.

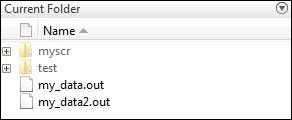
So, I studied Niklaus Wirth’s book Algorithms + Data Structures *=* Programs and learned how to parse programming languages. I wrote the first MATLAB an acronym for Matrix Laboratory in FORTRAN, with matrix as the only data type. The project was a kind of hobby, a new aspect of programming for me to 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:**

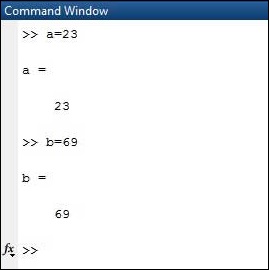
After logging into your account, you can enter MATLAB by double-clicking on the MATLAB shortcut icon (MATLAB 7.0.4) on your Windows desktop. When you start MATLAB, a special window called the MATLAB desktop appears. The desktop is a window that contains other windows. The major tools within or accessible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

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

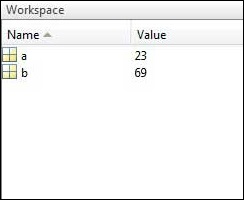
**Current Folder:** This panel allows you to access the project folders and files.



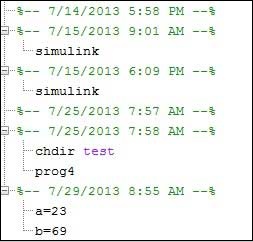
**Command Window:** This is the main area where commands can be entered at the command line. It is indicated by the command prompt (>>).



**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.