**COMPARATIVE STUDY OF LINEAR PRECODING TECHNIQUES**

**ABSTRACT**

This paper considers, The Linear Pre-coding technique of Block Diagonalization (BD) will be implementing on 5G MIMO environment. Along with the other linear pre-coding techniques such as, Maximum Ratio Transmission (MRT), Zero Forcing (ZF) and Minimum Mean Square Error (MMSE) will also be implemented and made comparison between later techniques with proposing BD technique and the parameters are verified through simulation. We have made comparison among many linear precoding techniques in 5G MIMO even though Maximum Ratio Transmission (MRT) performed better, Block Diagonalization (BD) is easier to implement.

**Keywords:** Linear pre-coding Algorithm, Block Diagonalization (BD), Maximum Ratio Transmission (MRT), Zero Forcing (ZF) Minimum Mean Square Error (MMSE), 5G mobile communication technology, MIMO.

**CHAPTER 1**

**INTRODUCTION**

**Linear Precoding**

Linear Precoding (LP) is applied to a system to serve multiple users at a time. Linear precoding makes the Beam-Forming really optimal. LP makes users to use the resources at the same time by using certain techniques to maintain orthogonality among users thereby, reducing interference between users. It is more robust in making signal less susceptible to interference which is a major concern at 5G and MIMO environments. Based on the type of use linear precoding techniques are classified as several types, they are, Zero Forcing (ZF), Minimum Mean Square Error (MMSE), Maximum Ratio Combining or Transmission (MRT) and Block Diagonalization (BD) etc.

**Zero Forcing**

Zero Forcing (ZF) is one of the earliest linear precoding techniques. ZF is once a widely used precoding technique, emergence of new LP techniques made it confined to a limited use. ZF also a beam forming technique which reduces the interference among users greatly by minimizing the collision of signals in any environment. As the name suggests, Zero Forcing (ZF) forces the other users who are in proximity to zero by reducing the interference among the users. ZF produces a vector called null vector which nullifies or diminishes the interference among users. The null vector is a pseudo inverse vector of the user on which the precoding is applied. ZF mainly depends on channel state information (CSI) which gives the status of the channel we are working with. Not only ZF but every other LP techniques depends and processes on the basis of CSI. The transmitted bit always pre-coded such that, received signal vector is always or most times the product of transmitted signal vector and the noise of the channel,

x = Hs + n

Where, s = transmitted signals which is given by, s =

n = Noise signals which is given by, n =

In order to generate the transmission weight vector from the channel matrix, a linear transformation is required, that is,

Where, G =

The transmitted signal is then transmitted as,

The pre-coding of the kth user in the cell as,

**Minimum Mean Square Error**

Minimum Mean Square Error (MMSE) technique of linear precoding is the advanced or extended or modified version. MMSE uses the channel state information (CSI) to find the precoding vector for the user on which the MMSE is applying. MMSE also uses the same procedure that ZF uses to produce the precoding vector. The equation is also the same for both the techniques, but, it uses ‘BI’ where BI is the coefficient of the precoding vector.

**CHAPTER 2**

**LITERATURE SURVEY**

**[1] W. C. Jakes, Jr., Mobile Microwave Communication. New York Wiley, 1974:**

A multiplicity of autonomous terminals simultaneously transmits data streams to a compact array of antennas. The array uses imperfect channel-state information derived from transmitted pilots to extract the individual data streams. The power radiated by the terminals can be made inversely proportional to the square-root of the number of base station antennas with no reduction in performance. In contrast if perfect channel-state information were available the power could be made inversely proportional to the number of antennas. Lower capacity bounds for maximum-ratio combining (MRC), zero-forcing (ZF) and minimum mean-square error (MMSE) detection are derived. A MRC receiver normally performs worse than ZF and MMSE. However as power levels are reduced, the cross-talk introduced by the inferior maximum-ratio receiver eventually falls below the noise level and this simple receiver becomes a viable option. The tradeoff between the energy efficiency (as measured in bits/J) and spectral efficiency (as measured in bits/channel use/terminal) is quantified. It is shown that the use of moderately large antenna arrays can improve the spectral and energy efficiency with orders of magnitude compared to a single-antenna system.

**Summary:** With the use of moderately large antenna arrays the efficiencies of energy and spectrum has improved with many orders than a one-antenna system.

**[2] Rusek F, Persson D, Lau B K, et al. Scaling up MIMO: Opportunities and challenges with very large arrays [J]. Signal Processing Magazine, IEEE, 2013, 30(1): 40-60:**

Massive multiple-input multiple-output (MIMO) techniques have the potential to bring tremendous improvements in spectral efficiency to future communication systems. Counterintuitively, the practical issues of having uncertain channel knowledge, high propagation losses, and implementing optimal non-linear precoding are solved more or less automatically by enlarging system dimensions. However, the computational precoding complexity grows with the system dimensions. For example, the close-to-optimal and relatively “antenna-efficient” regularized zero-forcing (RZF) precoding is very complicated to implement in practice, since it requires fast inversions of large matrices in every coherence period. Motivated by the high performance of RZF, we propose to replace the matrix inversion and multiplication by a truncated polynomial expansion (TPE), thereby obtaining the new TPE precoding scheme which is more suitable for real-time hardware implementation and significantly reduces the delay to the first transmitted symbol. The degree of the matrix polynomial can be adapted to the available hardware resources and enables smooth transition between simple maximum ratio transmission and more advanced RZF. By deriving new random matrix results, we obtain a deterministic expression for the asymptotic signal-to-interference-and-noise ratio (SINR) achieved by TPE precoding in massive MIMO systems. Furthermore, we provide a closed-form expression for the polynomial coefficients that maximizes this SINR. To maintain a fixed per-user rate loss as compared to RZF, the polynomial degree does not need to scale with the system, but it should be increased with the quality of the channel knowledge and the signal-to-noise ratio.

**Summary:** For maintaining a fixed per user rate loss as compared to RZF, the degree of polynomial need not to vary, but it should be increased with the quality of the channel knowledge and SNR.

**[3] Zarei S, Gerstacker W, Schober R. A low-complexity linear precoding and power allocation scheme for downlink massive MIMO systems[C]//Signals, Systems and Computers, 2013 Asilomar Conference on.IEEE, 2013: 285-290:**

In wireless communication fading of channels is the serious cause of the received degraded signals. The effect of fading can be minimized by using various time and space domain techniques. However, space domain techniques are preferred over the others due to its advantages. In this paper, comparison of the wireless MIMO system under Almouti's and maximum ratio combining schemes is presented. Basic idea in these schemes is to transmit and receive more than one copy of the original signals. Using two transmitter antennas and one receiver antenna, the scheme provides the nearly same diversity order as the maximal-ratio receiver combining (MRRC) with one transmitter antenna, and two receiver antennas. Results for one transmitter and four receivers under MRRC is also presented and compared. Finally, results are presented while varying the average transmitted power.

**Summary:** Comparison of the Maximum ratio combining and Almouti’s schemes for transmitting and receiving more than one copy of original signals.

**[4] Zarei S, Gerstacker W, Muller R R, et al. Low-complexity linear precoding for downlink large-scale MIMO systems[C]//Personal Indoor and Mobile Radio Communications (PIMRC), 2013 IEEE 24th International Symposium on. IEEE, 2013: 1119-1124:**

The performance of linear precoding schemes in downlink Massive MIMO systems is dealt with in this paper. Linear precoding schemes are incorporated with zero forcing (ZF) and maximum ratio transmission (MRT), truncated polynomial expansion (TPE), regularized zero force (RZF) in Downlink massive MIMO systems. Massive MIMO downlink output is evaluated with linear precoding included. This paper expresses the performance of achievable sum rate linear precoding with variable signal-to-noise (SNR) ratio and achievable sum rate and several transmitter-receiver antennas, such as imperfect CSI, less complex processing and inter-user interference. The transmitter has complete state information on the channel. The information narrate how a signal propagates to the receiver from the transmitter and reflects, for example, the cumulative effect of distance scattering, fading, and power decay. They show that the performance analysis of two linear precoding techniques, i.e., Maximum Ratio Transmission (MRT) and Zero Forcing (ZF) for downlink mMIMO output network over a perfect chain. The results show the improved ZF precoding achievable sum rate compared to the MRT precoding schemes and also compared the average achievable rate RZF and TPE.

**Summary:** Comparison of linear pre-coding techniques such as, MRT and ZF.

**[5] K. Zu R. C. de Lamare and M. Haardt "Generalized design of Low-Complexity block diagonalization type precoding algorithms for multiuser MIMO systems" IEEE Transactions on Communications vol. 61 no. 10 pp. 4232-4242 October 2013:**

Block diagonalization (BD) based precoding techniques are well-known linear transmit strategies for multiuser MIMO (MU-MIMO) systems. By employing BD-type precoding algorithms at the transmit side, the MU-MIMO broadcast channel is decomposed into multiple independent parallel single user MIMO (SU-MIMO) channels and achieves the maximum diversity order at high data rates. The main computational complexity of BD-type precoding algorithms comes from two singular value decomposition (SVD) operations, which depend on the number of users and the dimensions of each user's channel matrix. In this work, low-complexity precoding algorithms are proposed to reduce the computational complexity and improve the performance of BD-type precoding algorithms. We devise a strategy based on a common channel inversion technique, QR decompositions, and lattice reductions to decouple the MU-MIMO channel into equivalent SU-MIMO channels. Analytical and simulation results show that the proposed precoding algorithms can achieve a comparable sum-rate performance as BD-type precoding algorithms, substantial bit error rate (BER) performance gains, and a simplified receiver structure, while requiring a much lower complexity.

**SUMMARY:** Block diagonalization (BD) precoding technique was implemented on Multi-user MIMO systems for making a low-complexity precoding technique at MU-MIMO environments.

**CHAPTER 3**

**EXISTING METHOD**

We studied the performance of several main Linear Precoding algorithms in large-scale MIMO systems. For the convenience of expression, in a single cell system, we use indicates the downlink channel matrix from base station to user. Based on the theoretical analysis, we studied the performance of the main Linear Precoding algorithms, and made performance simulation under the actual scene conditions, and compared with the theoretical results.





In the single cell large-scale MIMO transmitter block diagram shown in Fig. 1, the base station precodes the signal and sends the signal vector to the user. S represents the original signal, and X represents the information vector sent by the sender to the user after precoding.

**A. Maximum Ratio Emission Precoding Algorithm**

In downlink, the MRT precoding technology is actually the matched filter precoding algorithm. The performance of MRT precoding in large-scale MIMO system depends on the channel transmission environment to a great extent, and the ideal environment is that the channels from the base station to different user terminals are as independent as possible. The expression of MRT precoding for the kth user in the cell is formula 3。MRT precoding is a very simple precoding technology, which can maximize the SNR of each user, but does not consider the interference between users.



**B. Zero Forcing Precoding Algorithm**

Zero forcing linear precoding scheme was originally proposed by Freescale Semiconductor Company. Different from the MRT precoding technology, ZERO FORCING precoding can completely remove the interference among users. It requires that all the signals received by users in the system do not contain the interference generated by other users, that is, make the precoding vector wk of user K in the channel matrix of other users in the zero space of, that is, the interference items of other users in the signals received by user K.



**C. Minimum Mean Square Error Precoding Algorithm**

Zero forcing precoding scheme reduces the interference of other users to zero, and does not consider the impact of noise on the system. When the noise causes the channel matrix to become ill conditioned or near ill conditioned, and the coefficient will be close to zero, the received signal component will be weakened, and the received signal to interference plus noise ratio (SINR) will be reduced. Meanwhile, the system will be affected. The difference is that in 2005, Christian B. peel, Bertrand M. hochwald and A. Lee swindlehurst proposed the minimum mean square error precoding scheme. It aims to maximize the SINR of the receiver, comprehensively considers the performance of the whole system, and makes MMSE precoding scheme perform better in the environment of low SNR by introducing parameters. The precoding matrix can be expressed as follows.



Next, the performance of several representative Linear Precoding algorithms described above is simulated. Assuming that the channel is Rayleigh fading, the channel matrix is made up of elements randomly generated by CN (0,1) with independent and same distribution, and normalized. The total transmit power of each base station is fixed, and the power is evenly distributed. In the scenario, it is assumed that the number of transmit antennas is m = 20, the number of receive users is k = 20, and the number of receive antennas for each user is 1. Here, we use Matlab software to write simulation program for simulation.

In this paper, the main performance indexes of MRT, ZF and MMSE Linear Precoding schemes are analyzed under the model of single cell MIMO. At the same time, the rate and bit error rate of MRT, ZF and MMSE precoding are compared by simulation, and MMSE is obtained The precoding technology is obviously better than the other two precoding technologies, especially in the environment of low SNR, which verifies the conclusion of theoretical derivation.

**Disadvantages:**

* The existing method results in a low Bit-error rate but, still not up to the required level.
* The existing method is inconsistent at massive MIMO 5G environments.

**CHAPTER 4**

**PROPOSED METHOD**

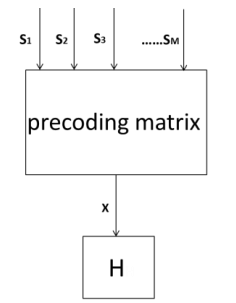
In large-scale MIMO systems, we looked at the performance of many prominent linear precoding techniques. To facilitate expression, we used a single cell system. Use denotes the user-to-base station downlink channel matrix. We examined the performance of the primary Linear Precoding methods based on a theoretical study, performed performance simulations in real-world scene conditions, and compared the results to the theoretical predictions. In the single cell large-scale MIMO transmitter block diagram shown in Fig. 1, the base station precodes the signal and sends the signal vector to the user. S represents the original signal, and X represents the information vector sent by the sender to the user after precoding.



ρ is the average transmission power of the base station. Therefore, the signal received by the kth user in the cell can be expressed as



Here, is the superposition of user interference signal and channel noise of the same pilot in other cells.



**Figure 1 Single cell large-scale MIMO transmitter block diagram**

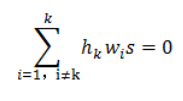
1. **Maximum Ratio Emission Precoding Algorithm**

The matched filter precoding algorithm is the MRT precoding technology used in downlink. The ideal channel transmission environment is one in which the channels from the base station to the various user terminals are as independent as feasible. The efficiency of MRT precoding in large-scale MIMO systems greatly depends on this environment. Formula 3 is the expression of MRT precoding for the kth user in the cell. MRT precoding is a very basic precoding technique that increases each user's SNR while ignoring interference from other users.



1. **Zero Forcing Precoding Algorithm**

Freescale Semiconductor Company first suggested a zero force linear precoding technique. In contrast to MRT precoding technology, ZERO FORCING precoding may entirely eliminate user interference. It is necessary for all signals received by users in the system to be free of interference created by other users, i.e., for user K's precoding vector to be in the channel matrix of other users and to be in the zero space of other users' interference items in user K's signals.

****

1) The client estimates the channel. That is, the channel among users is calculated using the pilot signal, and the estimated value of the channel matrix is obtained.

2) The client's feedback channel estimation. The precoding matrix is computed using the channel matrix that was previously determined. In the time division duplex (TDD) system, the base station must obtain the channel information state of the transmitter through the uplink feedback channel; in the TDD system, the base station directly estimates the channel information state of the downlink channel transmitter at the uplink pilot, improving the accuracy of the channel information state.

3) At the transmitter, the precoding matrix is computed. One way to define zero force precoding is as a pseudo-inverse matrix of the user channel matrix.

****

From the equations above, it is clear that precoding in multiuser MIMO can be thought of as a procedure that, to some extent, maximizes the ratio of target user gain to inter user interference + noise. MRT maximizes the signal from the intended user. MRT is a nearly ideal method in a signal-limited system when user interference is minimal relative to noise. The goal of zero force precoding is to reduce signal gain while also eliminating user disturbance. It can bring performance very close to the system capacity limit when there are a lot of users or when the noise level is high compared to the interference.

**C. Minimum Mean Square Error Precoding Algorithm**

Zero forcing precoding scheme reduces to zero the interference from other users and ignores the effect of noise on the system. The received signal component will be weaker and the received signal to interference plus noise ratio (SINR) will be lower when the noise makes the channel matrix ill-conditioned or nearly ill-conditioned and the coefficient will be close to zero.

The system will be impacted in the interim. The least mean square error precoding approach was introduced in 2005 by Christian B. Peel, Bertrand M. Hoch Wald, and A. Lee Swindle Hurst. It aims to increase the receiver's SINR, takes the performance of the entire system into account, and adds parameters to the MMSE precoding scheme to improve its performance in low-SNR environments.

****

Where, when SINR is the best, indicates the ratio of total power to noise power in the downlink. From the formula six shows that MMSE precoding algorithm is similar to ZF precoding matrix. When the signal-to-noise ratio is large, the interference of other users is the main factor that restricts the system performance. At this time, β tends to 0, and MMSE precoding degenerates into ZF precoding; when the signal-to-noise ratio is small, it becomes larger, and MMSE precoding matrix, at the cost of allowing partial interference to be retained, tries to maximize the received SNR. Therefore, the performance of MMSE precoding technology is better than ZF precoding technology.

**D Block Diagonalization Method**

In this section, we represent a novel BD method for multi-user MIMO systems. The BD algorithm is an ex-tension of the ZF method for multi-user MIMO systems where each user has multiple antennas. Each user's linear precoder and receiver filter can be obtained by twice SVD operations.

Diagonalization, the key idea of the BD algorithm is to employ the precoding matrix to suppress the MUI completely. To eliminate all MUI, the following constraint is imposed. where kis the diagonal matrix of which the diagonal elements are non-negative singular values of ~Hk and its dimension equals to the rank of ~Hk. Vk(0)contains vectors corresponding to the zero singular values, and Vk(1)consists of the singular vectors corresponding to nonzero singular values. Thus, Vk(0)is an orthogonal basis for the null space of ~Hk. In order to maximize the achievable sum rate of the BD, the water filling algorithm can be additionally incorporated.

****

**CHAPTER 5**

**ADVANTAGES AND APPLICATIONS**

**Advantages:**

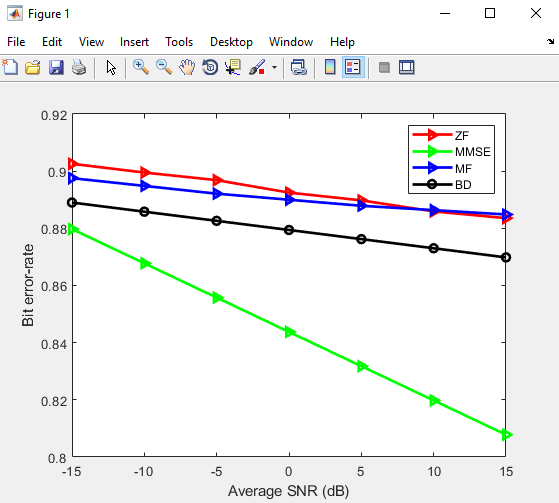
* The advantage of linear pre-coding technique of maximum ratio combining is its low bit-error rate compared to existing techniques.
* The MRT is good at low SNR environments.
* The ZF and MMSE are similar and good to compute at even higher SNR environments.
* The BD has easy computation.

**Applications:**

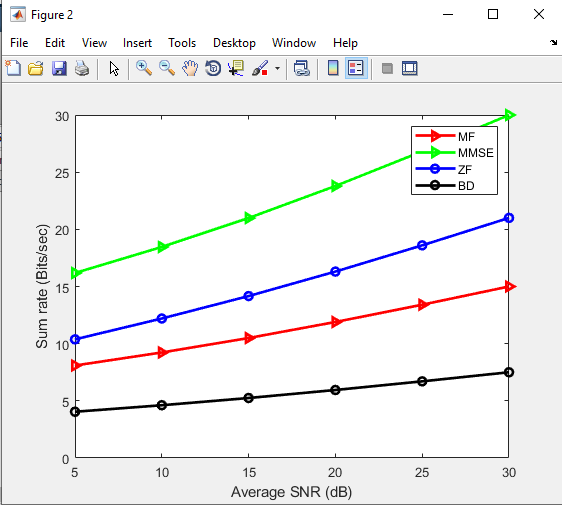
* Applications of are: MIMO environments, Beam Forming, Satellite Communication.

**CHAPTER 6**

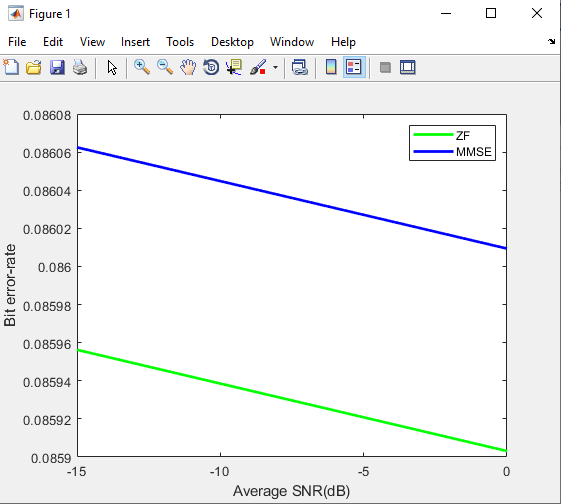
**RESULTS**

****

**Fig 1 Results of several linear precoding algorithms**

****

**Fig 2 Sum rate of several linear precoding algorithms**

****

**Figure 3 Bit error rate of several linear precoding algorithms**

**CHAPTER 7**

**CONCLUSION**

Here, in this paper we implemented four of the linear pre-coding techniques and made a comparison between them. The implementation have shown that, MRT has low bit error-rate and high sum-rate among the four implemented techniques. Hence, we can finally conclude that MRT is better among the four, but, ZF and MMSE are better at high SNR environments and BD is robust and easier to implement. Finally, we can conclude that, the Block Diagonalization (BD) precoding technique is easier to implement among all the other linear precoding techniques.

**REFERENCES**

[1] H. Ju and R. Zhang, “Throughput maximization in wireless powered communication networks,” IEEE Trans. Wireless Commun., vol. 13, no. 1, pp. 418-428, 2014.

[2] X. Kang, C. K. Ho, and S. Sun, “Full-duplex wireless-powered communication network with energy causality,” IEEE Transactions on Wireless Communications, vol. 14, no. 10, pp. 5539-5551, 2015.

[3] D. Xu and Q. Li, “Optimization of wireless information and power transfer in multiuser ofdm systems,” AEU-International Journal of Electronics and Communications, vol. 90, pp. 171-174, 2018.

[4] H. Lee, K.-J. Lee, H. Kim, B. Clerckx, and I. Lee, “Resource allocation techniques for wireless powered communication networks with energy storage constraint,” IEEE Transactions on Wireless Communications, vol. 15, no. 4, pp. 2619-2628, 2016.

[5] D. Xu and Q. Li, “Resource allocation for secure communications in cooperative cognitive wireless powered communication networks,” IEEE Systems Journal, 2018, doi:10.1109/JSYST.2018.2883491.

[6] Y.-C. Liang, K.-C. Chen, G. Y. Li, and P. Mahonen, “Cognitive radio networking and communications: An overview,” IEEE Trans. Veh. Technol., vol. 60, no. 7, pp. 3386-3407, 2011.

**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.

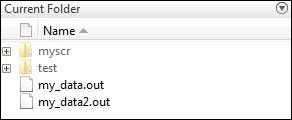
So, I studied Niklaus Wirth’s book Algorithms + Data Structures *=* Programs and learned how to parse programming languages. I wrote the first MATLAB an acronym for Matrix Laboratory in FORTRAN, with matrix as the only data type. The project was a kind of hobby, a new aspect of programming for me to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

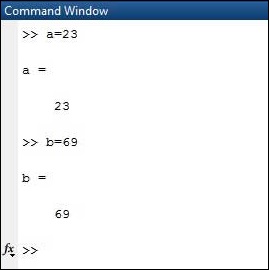
After logging into your account, you can enter MATLAB by double-clicking on the MATLAB shortcut icon (MATLAB 7.0.4) on your Windows desktop. When you start MATLAB, a special window called the MATLAB desktop appears. The desktop is a window that contains other windows. The major tools within or accessible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

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

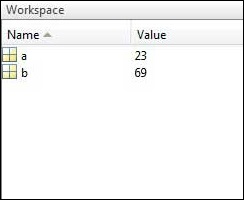
**Current Folder:** This panel allows you to access the project folders and files.



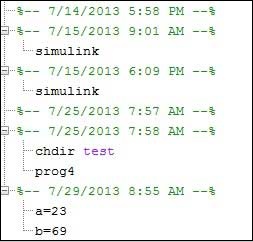
**Command Window:** This is the main area where commands can be entered at the command line. It is indicated by the command prompt (>>).



**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.