

Design and implementation of a java based virtual laboratory for data communication simulation

Obinna Okoyeigbo, Edevbie Agboje, Evioghene Omuabor,
Uyi Aiyudubie Samson, Abidemi Orimogunje

Department of Electrical and Information Engineering, Covenant University, Nigeria

Article Info

Article history:

Received May 25, 2019

Revised Feb 6, 2020

Accepted May 8, 2020

Keywords:

Data communications

Java-based

Simulation

Virtual laboratory

ABSTRACT

Students in this modern age find engineering courses taught in the university very abstract and difficult, and cannot relate theoretical calculations to real life scenarios. They consequently lose interest in their coursework and perform poorly in their grades. Simulation of classroom concepts with simulation software like MATLAB, were developed to facilitate learning experience. This paper involves the development of a virtual laboratory simulation package for teaching data communication concepts such as coding schemes, modulation and filtering. Unlike other simulation packages, no prior knowledge of computer programming is required for students to grasp these concepts.

Copyright © 2020 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Obinna Okoyeigbo,
Department of Electrical and Information Engineering,
Covenant University,
Ota, Nigeria.
Email: obinna.okoyeigbo@covenantuniversity.edu.ng

1. INTRODUCTION

The Engineering field of study is not an abstract one. Laboratories are used in universities for experimenting and testing the theoretical concepts taught in the classrooms and to also give students better understanding of these theoretical concepts. However, most universities in developing countries like Nigeria are unable to afford standard laboratories [1]. Hence, a virtual laboratory serves as a solution.

The aim of this project is to design and implement a Java-based virtual laboratory that would simulate communication through a channel and explain the effects of noise and coding in the communication process to make learning of the principles of data communication more practical. Virtual laboratories are software simulation packages which simulate laboratory processes [2, 3]. These packages when distributed to students, or made available over the internet can equip students with better laboratory experiences, and also save costs for the university [4].

The concepts of communication such as coding, modulation, filtering is not easy to understand, especially when the students find these concepts abstract, and do not know their importance in real life communication. This paper provides a solution to this problem by modelling a communication system laboratory, with oscilloscopes and spectrum analyzers which would simulate signal transmission in their time and frequency domains as they pass through different stages in the communication process. Thus, allowing users who don't have access to real laboratories to understand the various communication processes virtually.

In recent years, engineers have seen the importance of simulation software in designing prototypes and also in educating students and have built various simulation packages to that effect. MATLAB, developed by MathWorks with its signal processing toolbox can simulate a wide range of processes in communication [5, 6] as filtering, modulation, coding and also provides a platform to develop graphical user

interfaces (GUIs). However, this package requires the knowledge of programming with MATLAB to utilize these functions [7].

The Java digital signal processing (J-DSP) software, developed by Arizona State University, is an online communication simulation package developed in Java, and accessible from anywhere in the world over the internet [8, 9]. This package is however not available for students without access to the internet. ScopeFIR developed by Iowegian International, is a software package that designs finite impulse response filters [10]. This software package can design lowpass, highpass, bandpass and bandstop filters, designing the filter coefficients, based on the user specifications [11, 12]. The software also has built-in window functions. The limitation to this software, is that it does not cover other schemes in communication such as modulation and coding. This paper proposes a communication systems virtual laboratory software that would be accessible to students offline in their schools and also for installation in their individual PCs. This virtual laboratory software does not require prior programming knowledge, with the user-friendly graphical user interface provided.

2. RESEARCH METHOD

The communication process follows the framework of the Master Simulator given in Figure 1. The master simulator is a model that describes the process of communication from the transmitter to the receiver.

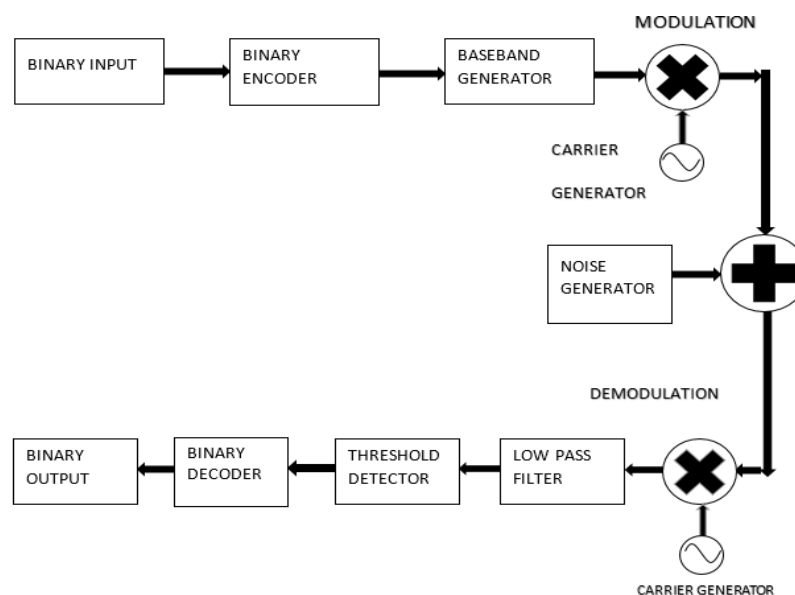


Figure 1. The master simulator [13]

2.1. Stages of the communication process

2.1.1. Binary input

At the stage, the binary digits to be sent to the receiver are keyed into the communication system by the user or sender of the message. The user is prompted for the message bits to be transmitted in this stage.

2.1.2. The binary encoder

The coding scheme used in this project is the Hamming Code Scheme. The extended hamming code scheme i.e. Hamming (8,4), which can correct one-bit error and detect two-bit errors, is an improvement of Hamming (7,4) which can only detect and correct one-bit error. The code word transmitted in the Hamming Code Scheme is made up of four message or "input" bits and three check bits for the Hamming (7,4) and four check bits for the Hamming (8,4).

The hamming distance is the total number of positions in which two code words differ from each other. The smallest hamming distance of the Hamming (7,4) is 3, while the smallest hamming distance of the Hamming (8,4) is four. This minimum hamming distance is a measure of the error correcting power of the coding scheme. The Generator Matrix (G-MATRIX) is a matrix, such that when it is vector multiplied by

the message bits, it gives the information signal with the check-bits appended to it. It is obtained by setting an identity matrix (of order k by k , where k is the number of message bits) to the left side of the matrix, and appending the check bits for the combination of each row, at the right side of the matrix. It is given by:

$$G = [I_k | A] \quad (1)$$

For Hamming (n,4), the G-matrix is given by;

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (2)$$

The check bits are given by;

$$\begin{aligned} A_1 &= K_1 + K_2 + K_4 \\ A_2 &= K_1 + K_3 + K_4 \\ A_3 &= K_2 + K_3 + K_4 \end{aligned} \quad (3)$$

where “+” signifies modulo-2 addition, A_n represents the check bit, and K_n represents the bit in the generator matrix with respect to each row.

The “input” bits or message bits are modulo-2 vector multiplied with the G-matrix. The result of this multiplication is the input bits with the check bits appended to it. For example, if bits sequence of ‘1010’ are entered by the user, the code word becomes;

$$Codeword = [1 \ 0 \ 1 \ 0] * \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} = 1010101 \quad (4)$$

Following the even parity scheme, one extra parity bit is appended to the code word. If the number of “1”s obtained in the result of the vector multiplication above is an even number, a parity bit of “0” is appended. If it is an odd number, a parity bit of “1” is appended to ensure that the number of bits transmitted is an even number. For the above example, the transmitted code word is 10101010. See Figure 2 for the flow chart.

2.1.3. Baseband signal

The baseband is the frequency range of a signal to be transmitted prior to when it is modulated at transmission. The baseband signal is generated from the result of the encoding process. The encoded bits are formed into a continuous square wave signal with two amplitude levels of 1 and -1. This signal is then sampled $x/8$ times per bit, where x is the total number of samples in the signal in a period of 1 second. From the formula;

$$Frequency \ Resolution = \frac{1}{T} = \frac{F_s}{N}, \quad (5)$$

where F_s is the sampling frequency, N is the number of samples in one full period, and T is the period, ($T=1$). Hence, if x is 2048, the signal is sampled at 256 per bit and the sampling frequency is 2048Hz. The non-return-to-zero transforms zeros ‘0s’ in the sampled bits to amplitude levels of ‘-1’. The sampled bits become the baseband signal after the NRZ transformation is performed.

2.1.4. Modulation

Modulation is imposing the inherent characteristics of a baseband signal into a carrier signal. In this way, the signal is prepared to be transmitted over a channel that is most likely to impair the signal [14]. The modulation scheme employed is the binary phase shift keying (BPSK). BPSK offers strong resistance to noise across a channel [15].

Binary phase shift keying (BPSK) involves varying the phase of the carrier signal to provide information on the baseband signal. Non-return-to-zero (NRZ) encoding is first applied to the encoded message bits, converting the zeros to negative one (-1) and leaving the ones as 1. This signal is multiplied with the carrier signal. In BPSK, the signal is shifted by 180 degrees for every bit transition [16]. The carrier signal is given by $Asin(2\pi ft)$, BPSK is achieved by multiplication of the carrier signal and the baseband signal such that,

$$\text{Modulated Signal} = x(n) * Asin(2\pi ft) \quad (6)$$

$$\text{Modulated Signal} = Asin(2\pi ft), \text{ if } x(n) = 1 \quad (7)$$

$$\text{Modulated Signal} = -Asin(2\pi ft) = Asin(2\pi ft + \pi), \text{ if } x(n) = -1 \quad (8)$$

2.1.5. Channel

The channel simulated in this project is an (AWGN) additive white gaussian noise channel, which is characterized by having a noise value present at every frequency in the frequency spectrum. It can be simulated by a method in Java syntax called `Random.nextGaussian()`, which would generate a sequence of numbers with a mean of 0 and standard deviation of 1 [17]. The process of transmission over the channel is described using a flowchart in Figure 3.

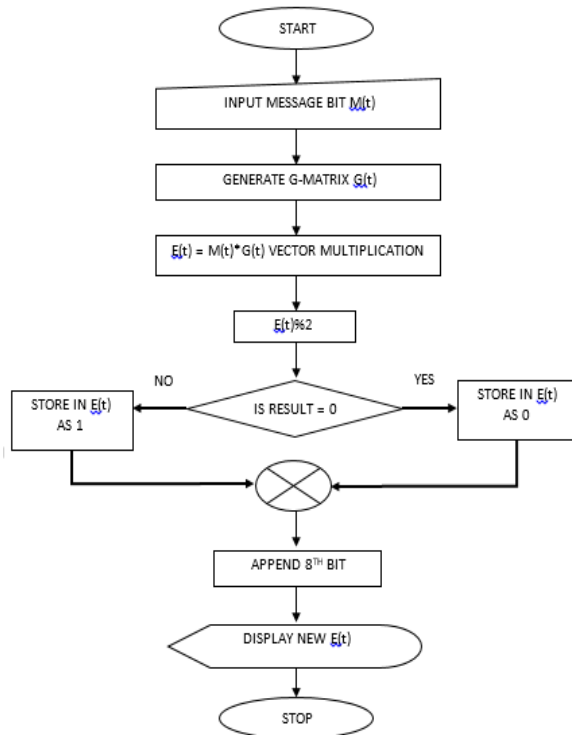


Figure 2. Flowchart showing encoding process

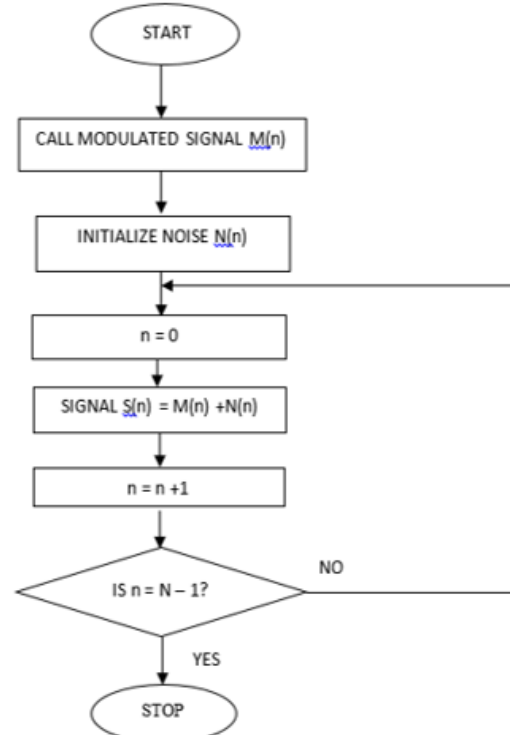


Figure 3. Flowchart showing transmission over the channel

2.1.6. Demodulation

Demodulation is a process that seeks to obtain the baseband signal from the modulated signal. The BPSK demodulator is obtained by generating a carrier signal which is at the same phase and of the same frequency as the carrier signal for modulation at the transmitter and multiplying it with the received signal. In practical, a phase locked loop (PLL) [18] or a costas loop [19] is used to obtain the frequency and phase of the carrier from the received signal but however, designing a PLL is beyond the scope of this project. Hence, a carrier signal is generated, which is at the same phase and known frequency of the carrier signal used for modulation, and used to multiply the modulated signal [20].

2.1.7. Filtering

The purpose of filters in a communication system is to remove unwanted frequency components of a signal, implemented to remove noise or extract certain components of the signal. The filter used in this project is a finite impulse response (FIR), low pass filter. The demodulated signal is passed through the filter which would allow frequencies lower than the cut-off frequency and attenuate signals higher than the cut-off frequency, with the assurance of stability in its operation. Filtering in the time domain is a process of convolution of the filter coefficients and the demodulated signal [21]. If the length or number of taps of a filter is given by $2M + 1$, the impulse response of a finite impulse response low pass filter whose terms are truncated to the length of the filter, is given by;

$$h(n) = \begin{cases} \frac{\Omega_c}{\pi} & n = 0 \\ \frac{\sin(\Omega_c n)}{n\pi} & \text{for } n \neq 0 \\ & M \leq n \leq M \end{cases} \quad (9)$$

The filter coefficients b_n is obtained by shifting the impulse response of the filter by M samples:

$$b_n = h(n - M) \quad (10)$$

The filtered signal is finally given by:

$$y(n) = \sum_{i=0}^{2M} b_i x(n - i) \quad (11)$$

which is the convolution of the filter coefficients and the input signal.

Due to the truncation of the filter coefficients during FIR filter designing, Gibbs oscillations are formed on the pass band of the filter [22]. A window function is multiplied with the filter coefficients to produce a filter with fewer oscillations. The window functions used in this paper are;

- The rectangular window [23]

$$w(n) = 1, \quad -M \leq n \leq M \quad (12)$$

- The bartlett window [23]

$$w(n) = 1 - \frac{|n|}{M}, \quad -M \leq n \leq M \quad (13)$$

- The hanning window [23]

$$w(n) = 0.5 \left(1 + \cos \frac{\pi n}{M} \right), \quad -M \leq n \leq M \quad (14)$$

- The hamming window [23]

$$w(n) = 0.54 - 0.46 * \cos \frac{2\pi n}{2M}, \quad 0 \leq n \leq 2M \quad (15)$$

2.1.8. Threshold detection

The filtered signal is sampled at the midpoint for each set of bits used to represent an input. For example, if the signal was sampled at 2048 bits, the number of bits used to represent an input bit is $2048/8=256$ bits. Hence, each set of 256 bits is sampled at midpoints, which could be 127, in this case. If the sampled amplitude is greater than 0, it is converted to '1', and if it is less than 0, i.e. near '-1', it is converted to '0'.

2.1.9. Decoder

On receiving the bits at the receiver, the bits are checked for errors first by checking the parity bit. For an even-parity system, the number of "1"s are an odd number, it means that at least one bit is in error. The parity bit is removed and what is left of the received bit sequence is modulo-2 vector multiplied with the check (H) matrix. The H-matrix is given by:

$$H = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (16)$$

The hamming decoder utilizes the check matrix (H-matrix) to detect for the error in the received bit sequence by modulo-2 vector multiplication with the received bits to obtain a syndrome matrix (S). Assuming a received bit sequence of 10101010, after removal of the parity bit, the syndrome become,

$$S = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} * \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (17)$$

The syndrome sequence is compared with the H-matrix, and the column that corresponds with the syndrome matrix is the position with error in the received bit sequence. If the syndrome gives a sequence of zeroes, it means that there is no error in the received bits sequence, and that the error indicated is in the parity bit. The bit in the indicated error position is corrected. The received bit sequence is now ready to be decoded. The corrected received bit sequence is modulo-2 vector multiplied with the decoding matrix, denoted here as the R-matrix, to decode the signal. The original input bits and the decoded bits are compared, and if there are any differences, the number of differences is the bits in error.

$$R = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (18)$$

2.1.10. Time and frequency analysis

The signal at each stage of the communication process would be analyzed by a virtual oscilloscope that would display a two-dimensional plot of the signal as a function of time, and a virtual spectrum analyzer that displays the magnitude frequency response of the signal. The frequency domain or spectrum of a signal can be obtained by the Fourier transform of the signal. The Fourier transform of a signal gives the complete description of the signal [24, 25]. Signals are easier analyzed and synthesized in their frequency domains. For discrete time systems, the frequency spectra which is a function of continuous frequencies is obtained by the discrete time Fourier transform (DTFT). The discrete spectra are obtained by sampling the DTFT in a process known as obtaining the discrete fourier transform (DFT). The fast Fourier transform is used to obtain the DFT of a sampled signal. The Radix-2 Decimation-in-Time Cooley-Tukey algorithm for obtaining FFT of a signal was employed in this paper.

3. IMPLEMENTATION

3.1. Graphical user interface (GUI)

The GUI of the virtual laboratory shown in Figure 4 was built using Java Swing Forms, developed in NetBeans, an integrated development environment (IDE) for Java. It is user friendly, and tasks can be performed with just a click of buttons.

3.2. Performance of communication system using the virtual laboratory

The performance is obtained by plotting a graph of bit error rate (BER) on the y-axis, against the ratio of the energy per bit to noise spectral density (Eb/No) on the x-axis. Eb/No is usually measured in decibels (dB). Figure 5 gives a comparison of the simulated BPSK BER performance with and without the use of coding schemes and the theoretical BPSK BER performance.

It was observed in Figure 5 that Hamming Code scheme has better BER performance, when compared to the uncoded scheme. It shows that the communication system is less prone to errors when the Hamming Code scheme is used before transmission, compared to when it is uncoded. This confirms with the theory that there is a higher chance of receiving bits in errors when a coding scheme is not used before communication over an AWGN channel.

It was also observed that the simulated BER performance with the Hamming Code scheme was very similar to the theoretical BER performance. This is because the efficiency of the modulation scheme is enhanced with the introduction of an error checking and correction mechanism. Hence, the Hamming Code

can be used to check and correct errors in the communication system, thereby increasing the reliability and efficiency.

As shown in Figure 6, the use of window functions makes the channel more prone to errors as the Bit Error Rate is larger when window functions such as Hamming are used. This is because window functions introduce a wider transition band in the frequency response of the demodulated signal. The transition band is the band between the pass-band and the stop-band. The wider transition introduces more errors in the system.

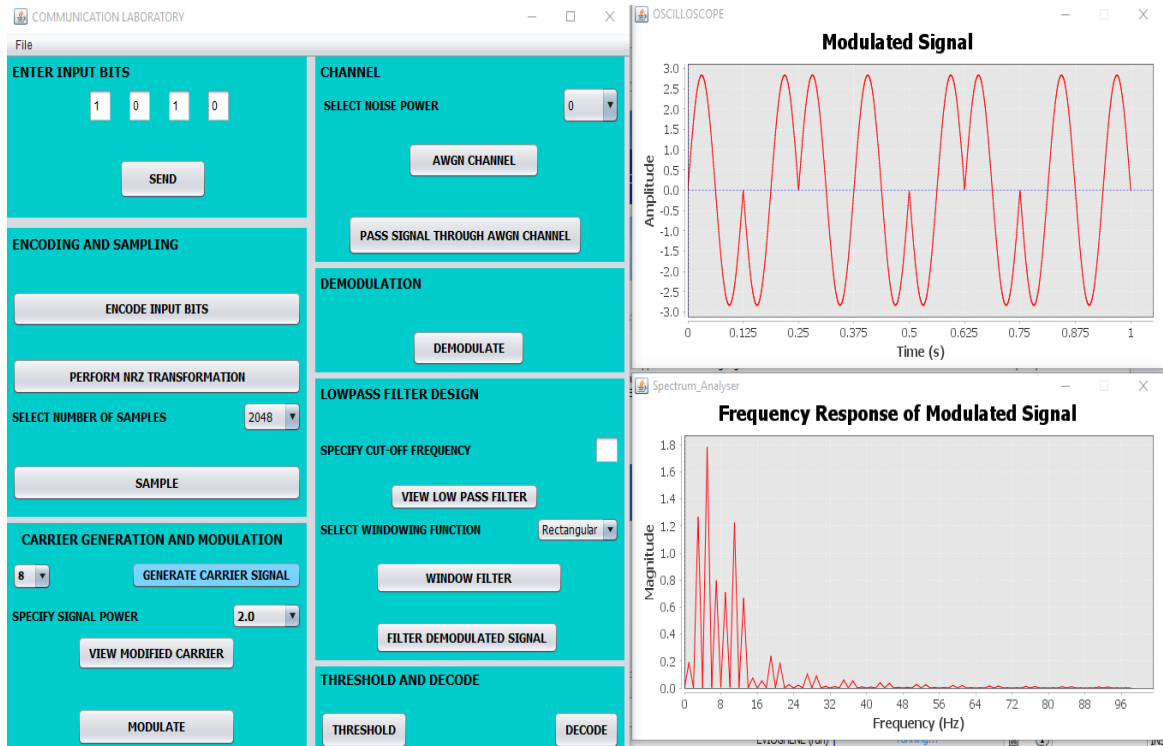


Figure 4. Graphical user interface of simulation package

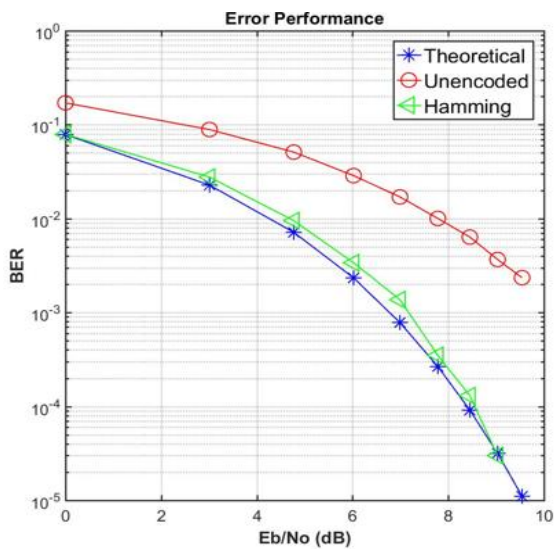


Figure 5. BER comparison between simulated (hamming and unencoded) and theoretical BPSK

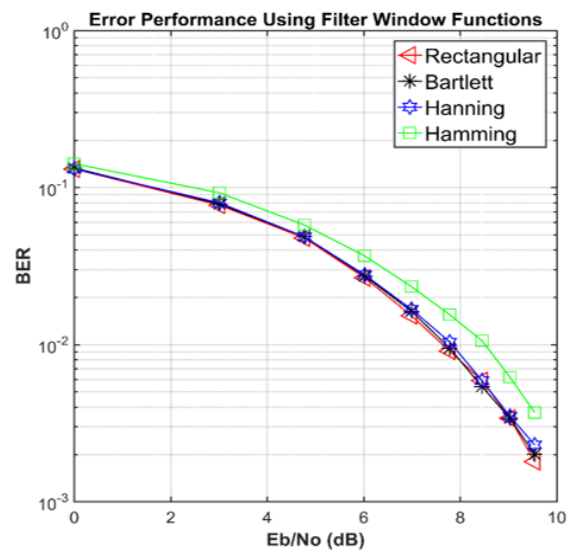


Figure 6. BER performance using filter design window functions

4. CONCLUSION

In this paper, a virtual laboratory software package for simulation of key data communication concepts has been presented. It models a typical communication systems laboratory, which teaches key communication concepts such as modulation, filtering and information coding. The system is easy to use, with a user-friendly graphical user interface, and the simulation results show that errors can be combated using coding schemes, which confirms with theoretical concepts. This package would facilitate the understanding of students taking courses in Communication Systems in tertiary institutions and other interested users. This virtual laboratory is also available at the convenience of learners and interested participants

ACKNOWLEDGEMENTS

This work was supported by Covenant University, Ota, Ogun State, Nigeria.

REFERENCES

- [1] O. A. Famade, et al., "Towards improved funding of tertiary institutions in Nigeria," *Asian Journal of Humanities and Social Sciences (AJHSS)*, vol. 3, no. 2, pp. 83-90, 2015.
- [2] P. A. Hatherly, et al., "Interactive screen experiments—innovative virtual laboratories for distance learners," *European Journal of Physics*, vol. 30, no. 4, pp. 751-762, 2009.
- [3] A. Alexiou, et al., "Virtual laboratories in education," in *IFIP World Computer Congress, TC 3*, pp. 19-28, 2004.
- [4] K. Muthusamy, et al., "Virtual laboratories in engineering education," *Asian Journal of Distance Education*, vol. 3, no. 2, pp. 55-58, 2005.
- [5] O. Okoyeigbo, et al., "Comparative Study of MIMO-OFDM Channel Estimation in Wireless Systems," *International Review on Modelling and Simulations*, vol. 11, no. 3, pp. 158-165, 2018.
- [6] O. Obinna, et al., "Comparative Analysis of Channel Estimation Techniques in SISO, MISO and MIMO Systems," *International Journal of Electronics and Telecommunications*, vol. 63, no. 3, pp. 299-304, 2017.
- [7] J. Sánchez, et al., "Easy java simulations: an open-source tool to develop interactive virtual laboratories using matlab/simulink," *International Journal of Engineering Education*, vol. 21, no. 5, pp. 798-813, 2005.
- [8] A. Spanias and V. Atti, "Interactive online undergraduate laboratories using J-DSP," *IEEE Transactions on Education*, vol. 48, no. 4, pp. 735-749, 2005.
- [9] A. Spanias, et al., "Using J-DSP to Introduce Communications and Multimedia Technologies to High Schools," in *The 33rd Annual Frontiers in Education Conference*, pp. F3A_22-F3A_27, 2003.
- [10] G. Gupta, et al., "PowerPC Processor with Floating Point Unit for Virtex-4 FX Devices," *Xilinx XAPP547 (v1.0.1)*, 2006.
- [11] S. Derouiche and B. Djedou, "Implementation Of The Development Of A Filtering Algorithm To Improve The System Of Hearing In Hearing Impaired With Cochlear Implant," *Proceedings of Advanced Information Technologies and Applications*, pp. 65-79, 2013.
- [12] O. Gunaydin and M. Ozkan, "Design of a brain computer interface system based on electroencephalogram (EEG)," in *4th European Education and Research Conference (EDERC 2010)*, pp. 150-154, 2010.
- [13] O. Oyetoke and O. E. Agboje, "Design and Implementation of A Java Based Simulation Package for Spectrum Analysis, Digital Filtration and Modulation as a Teaching Aid for Data Communication," *International Journal of Electrical and Electronics Research*, vol. 3, no. 1, pp. 30-41, 2015.
- [14] R. Priemer, "Introductory signal processing," *Advanced Series in Electrical and Computer Engineering, World Scientific Publishing Company*, vol. 6, 1990.
- [15] D. Mutz and K. George, "Costas loop and FFT based BPSK demodulation for pulsed radar receivers," in *2016 IEEE Aerospace Conference*, pp. 1-12, 2016.
- [16] J. G. Proakis and M. Salehi, "Digital communications," *McGraw-hill New York*, vol. 4, 2001.
- [17] S. Smith, "Digital signal processing: a practical guide for engineers and scientists," *Elsevier*, 2013.
- [18] B. Shamla and K. G. Devi, "Design and implementation of Costas loop for BPSK demodulator," in *2012 Annual IEEE India Conference (INDICON)*, pp. 785-789, 2012.
- [19] H. Yuan, et al., "Design and implementation of Costas loop based on FPGA," in *2008 3rd IEEE Conference on Industrial Electronics and Applications*, pp. 2383-2388, 2008.
- [20] P. Dondon, et al., "Design of a low cost BPSK modulator/demodulator for a practical teaching of digital modulation techniques," in *WSEAS: World Scientific and Engineering Academy and Society*, pp. 61-66, 2007.
- [21] H. Zumbahlen, "Linear circuit design handbook," *Newnes*, 2011.
- [22] L. Tan and J. Jiang, "Digital Signal Processing: Fundamentals and Applications," *Academic Press*, 2013.
- [23] K. S. Kumar, et al., "Performance Comparison of Windowing Techniques for ECG Signal Enhancement," *International Journal of Engineering Research*, vol. 3, no. 12, pp. 753-756, 2014.
- [24] G. F. Margrave, "Theory of nonstationary linear filtering in the Fourier domain with application to time-variant filtering," *Geophysics*, vol. 63, no. 1, pp. 244-259, 1998.
- [25] K. A. Stroud and D. J. Booth, "Advanced engineering mathematics," *Industrial Press*, 2003.