

TEACHING OF DIGITAL MODULATION TECHNIQUES USING A SIMULATION ENVIRONMENT

Abul K. M. Azad

*Department of Technology, College of Engineering and Engineering Technology,
Northern Illinois University, USA.
Email: azad@ceet.niu.edu*

Introduction

Teaching of digital modulation techniques is an important part of a digital and data communication course. This becomes challenging when it needs to be incorporated within a technology course, where the focus is mainly with application rather than the theoretical aspects. The student needs to understand the impact of various modulation parameters towards the modulation and demodulation outcomes. In addition, there is limited time for course teachers to address all these issues within a course. It has been proven that in addition to traditional classroom teaching, a simulation environment can play an important role towards the understanding of subject matter (Tokhi *et al.*, 1999; Chen and Naughton, 2000).

In the past, academics have used high level languages, such as, FORTRAN, PASCAL, Modula-II, and C++, for the development of engineering and engineering technology simulation environments (Auslander *et al.*, 2002). As these languages are originally designed for general purpose applications, it is quite tedious to implement various engineering concepts in an efficient manner as well as for convenient presentation of the results. The development of modern engineering software packages (Matlab, Femlab, Matrix, Pspice, MathCAD, Electronic Workbench, etc.), along with the progress in computer hardware and operating systems, offers exciting opportunities for educators to develop computer simulation environments for delivering course materials (Penfield and Larson, 1996).

A number of researchers have used Matlab for the development of simulation environments for engineering systems (Azad and Tokhi, 2003; Chen and Naughton, 2000; Tokhi *et al.*, 1999). Towards this effort, a Matlab based simulation environment has been designed and developed for the implementation of digital modulation and demodulation techniques for an undergraduate technology course. The environment will be used by the student as a complement to the lecture and laboratory classes. The Matlab and its toolboxes work with the Guide in an interactive manner to provide a flexible simulation environment where the user can observe the behavior of an engineering system or process without going into detailed implementation of the algorithm (Marchand and Holland, 2003).

This paper presents the development of a highly interactive user-friendly environment for the simulation of digital modulation and demodulation techniques. After this introduction, the first

section describes the theoretical basis of various digital modulation techniques that are implemented within the environment. The second section describes the structure of the developed simulation environment along with a case study. These are followed by the conclusion and a list of references.

Simulated Digital Modulation Techniques

There are a number of digital modulation techniques that one can employ for electronic communication. In this paper, only four of these techniques that are being implemented within the simulation environment will be discussed. The techniques are: a) Binary Phase Shift Keying; b) Binary Frequency Shift Keying; c) Quadrature Phase Shift Keying; and d) Quadrature Amplitude Modulation. The rest of this section will describe the theoretical basis for the developed simulation environment for these modulation techniques.

Binary Phase Shift Keying

Binary Phase Shift Keying (BPSK) is one of the simplest forms of digital modulation techniques. With BPSK, two output phases are possible for a single carrier frequency. As the input digital signal changes state, the phase of the output carrier shifts between two angles that are 180° out of phase. Figure 1 shows the binary input data and subsequent modulated signal.

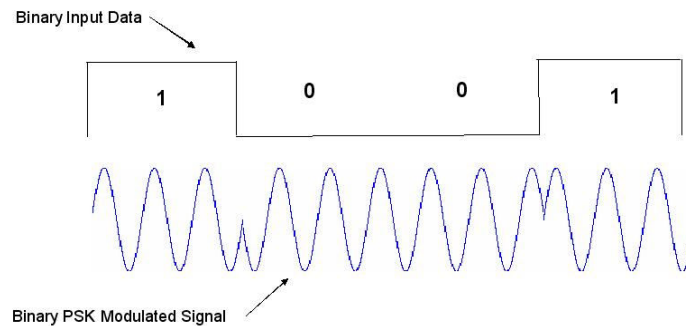


Figure 1: Input data and BPSK modulated signal.

Binary Frequency Shift Keying

Binary Frequency Shift Keying (BFSK) is a form of constant-amplitude angle modulation similar to the frequency modulation, except that the modulating signal is a binary signal that varies between two discrete voltage levels. The general expression for BFSK can be written as (Tomasi, 2001):

$$BFSK_{out}(t) = V_c \cos\{2\pi[f_c + v_m(t)\Delta f]t\} \quad (1)$$

where,

$BFSK_{out}(t)$ = binary FSK output;

V_c = peak carrier amplitude;

f_c = carrier frequency;

Δf = peak frequency deviation;

$v_m(t)$ = binary modulating signal (+/- 1).

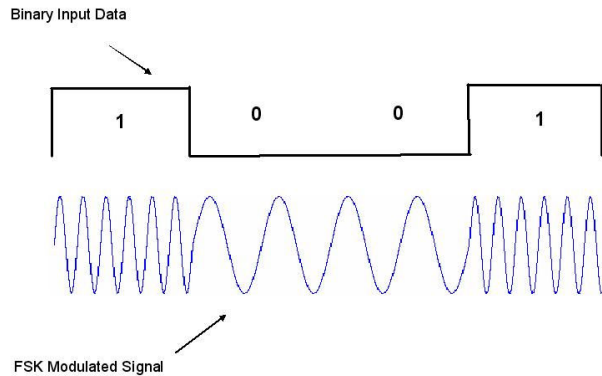


Figure 2: Waveform of a BFSK modulated signal.

As shown in equation 1, the peak shift in the carrier frequency Δf is proportional to the amplitude and the polarity of the binary input signal. The modulating signal $v_m(t)$ is either a +1 (logic 1) or a -1 (logic 0). As the binary signal shifts in logic, the output frequency shifts between two frequencies: mark frequency (logic 1) and space frequency (logic 0) (The mark and space frequencies are arbitrarily assigned, depending on the system design). Figure 2 shows the BFSK modulated waveform (continuous phase) for a binary signal. It can be seen that as the logic changes its state, the frequency shifts between the mark and space frequency values.

Quadrature Phase Shift Keying

Quadrature Phase Shift Keying (QPSK) is another form of angle-modulated, constant amplitude modulation. It is an M -ary encoding technique, where $M = 4$ (Miller and Beasley, 2002). There are four possible output phases for a single carrier frequency. The QPSK encodes two bits at a time, which results in 4 different phase-combinations. The binary input data is combined into groups of two bits called *dibits* and clocked into the modulator. For every *dibit*, a single output waveform is generated; therefore, the baud rate for QPSK is half the input bit rate.

The signal shifts in increments of 90 degrees from 45 to 135, 135 to -45, -45 to -135 or -135 to 45 degrees. These points are chosen as they can be easily implemented using an I/Q modulator. Only two I values and two Q values are needed, and this gives two bits per channel. There are four states for QPSK. It is, therefore, a bandwidth-efficient modulation compared to BPSK and potentially twice as efficient.

Quadrature Amplitude Modulation

Quadrature Amplitude Modulation (QAM) is a method for transmitting two separate (and uniquely different) channels of information using a single carrier. The two modulation inputs (analog or digital) are applied to two separate balanced modulators, each of which is supplied with the \sin or \cos carriers. That means one of the modulators is supplied with the \sin carrier and the other with the \cos carrier.

The outputs of both modulators are algebraically summed and result in a single waveform containing the I and Q information. The output waveform of an 8-QAM modulator is shown in Figure 3.

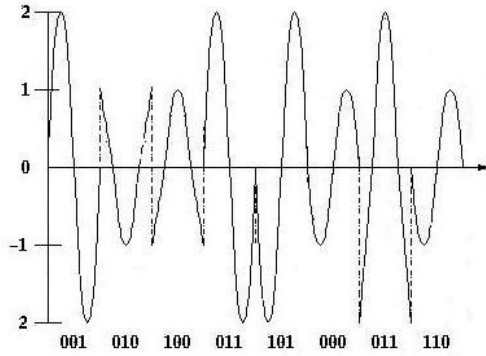


Figure 3: Typical 8-QAM modulated signal waveform.

An M -QAM receiver is almost identical to the M -PSK receiver. The differences are that the PAM levels at the output of product detectors are different from those achievable by using PSK. Unlike PSK, QAM also has different amplitude levels for the bit combinations being transmitted.

The Simulation Environment

The simulation environment is developed by using Matlab, Guide, and other associated Matlab toolboxes (Mathworks, 2001). The Matlab and associated toolboxes provide the background computation for each of these implementations, while the user interacts with the environment through GUIs developed by using the Guide. Within the environment, four digital modulation and demodulation techniques are implemented. For each of these techniques, there are three GUIs. These are: a) Transmitter GUI; b) Channel GUI; and c) Receiver GUI. The implementation of the QPSK modulation and demodulation process using this environment is illustrated. At the very beginning of the process, the transmitter GUI can be activated by entering the "transmitter" at the Matlab command prompt.

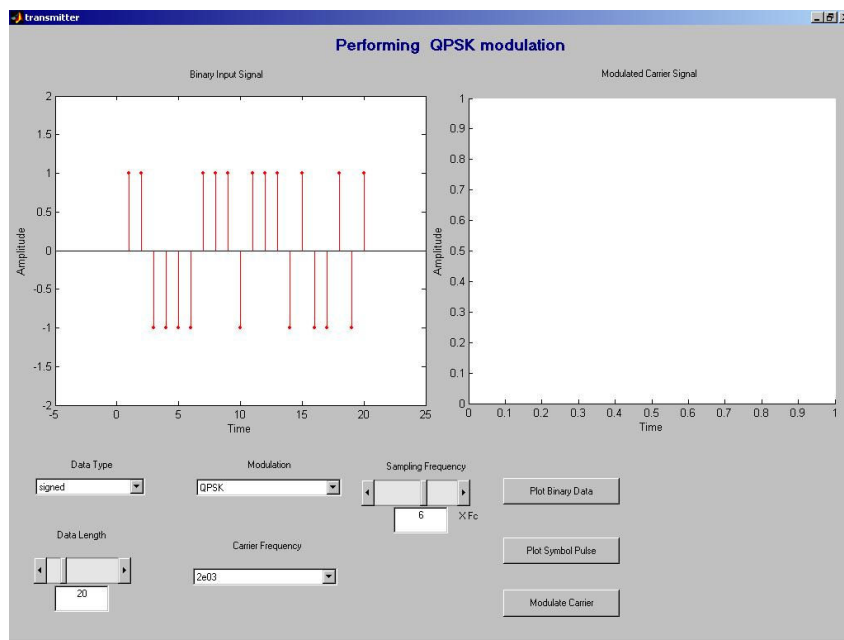


Figure 4: An image of the transmitter GUI with binary data input.

Binary data generation and display

After providing all the modulation parameters within the receiver GUI, the user can click on the Plot Binary Data button at the bottom right hand side of the GUI. This would generate the binary data input in accordance with the user inputs. A set of binary signals (top left side graph window) along with user inputs are shown in Figure 4. The binary data input will be used as the modulating signal.

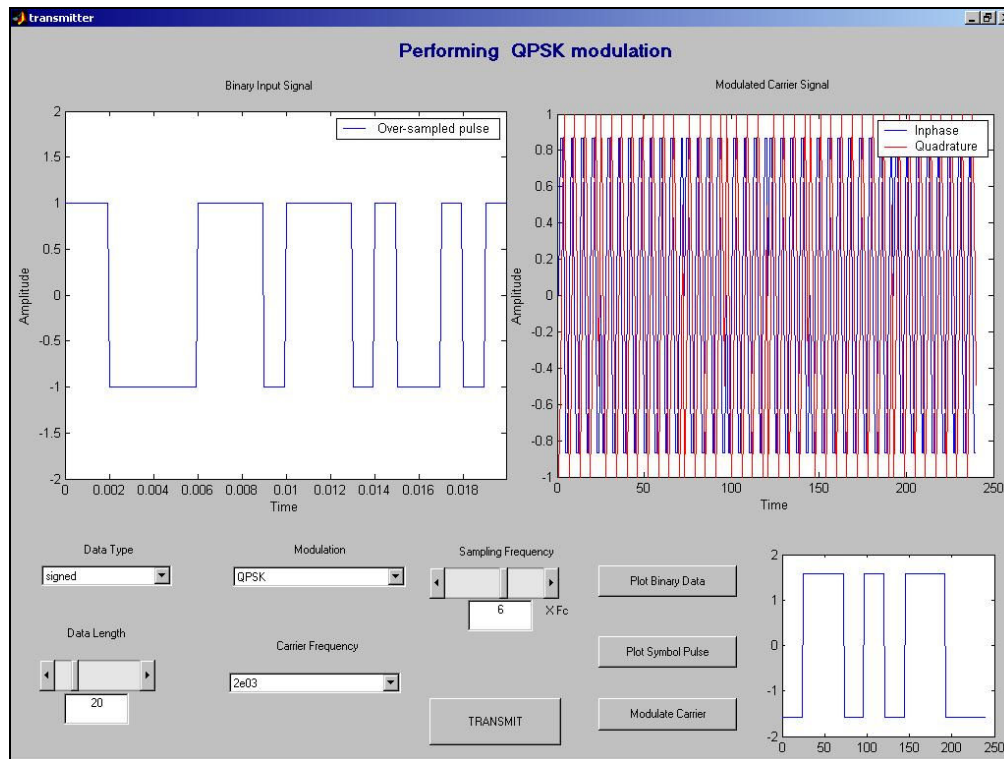


Figure 5: An image of the *Transmitter GUI* with all the waveforms.

Binary pulse shaping and modulation

To proceed with the modulation process, it is necessary to convert the bit pattern (shown in Figure 4) into a pulse. The non return to zero coding is used to convert the binary input signal to signal pulse. These signal pulses are then used to modulate the carrier. A click on the **Plot Symbol Pulse** will perform the conversion task. The pulse is shown on the left hand side graph window of Figure 5. A click on the **Modulate Carrier** button will perform the desired modulation. In this case, the QPSK has been selected within the GUI. The QPSK modulated waveform is shown on the right hand side graph window of Figure 5. Upon completion of these steps, two new features would appear on the GUI (Figure 5). One of these is a button called **Transmit**, and the other is a new graph window (bottom right hand side). The graph window is used to display the in-phase and quadrature components of the signal one after another.

Transmitting through channel

The next step is to transmit the modulated signal through the channel. This can be performed by clicking on the newly appeared **Transmit** button (Figure 5). This will open a new GUI window

for channel simulation (Figure 6). The purpose of the channel window is to provide the user with an option of introducing desired noise with the modulated signal. In this case, the QPSK modulated carrier signal is mixed with a white noise with the desired mean and variance. The signal to noise ratio in decibels can be calculated by clicking on the **Calculate SNR** button. After achieving the desired level of SNR, the user can click on the **Add Noise** button. The undistorted and distorted modulated carrier signals are shown in Figure 6. The left hand side graph is showing the undistorted waveform, while the right hand side graph is showing the distorted waveform. After introducing the desired level of noise, the user can pass the distorted modulated signal to the receiver by clicking on the **Receiver** button within the *Channel GUI*.

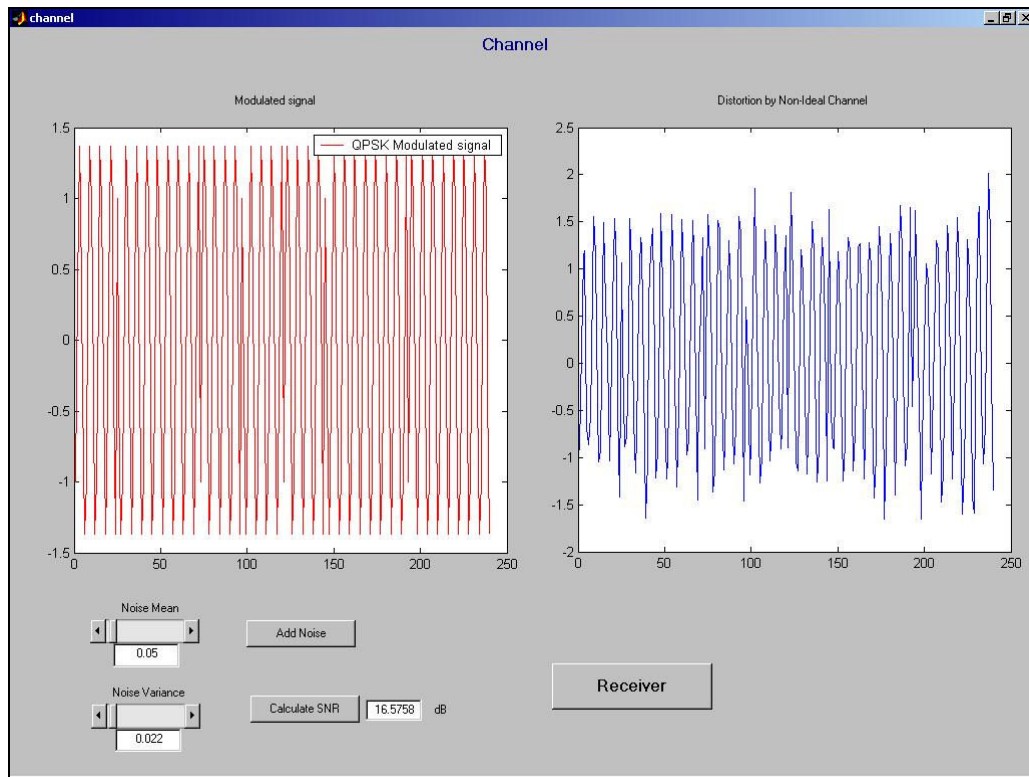


Figure 6: Image of a *Channel GUI* with undistorted and distorted QPSK waveform.

Receiver GUI for demodulation

The *Receiver GUI* window opens with the received noise-corrupted signal in the left hand side display graph (Figure 7). The user only has to click on the **Recover Signal** button to demodulate the information signal from the coded waveform. The algorithm used for the receiver side will automatically identify the implemented modulation technique and implement the appropriate demodulation technique. When a user presses the **Recover Signal** button, the right hand side graph will display the demodulated waveform (Figure 7). After the demodulation process, the user can return to the *Transmitter GUI* by clicking on the **Reset** button at the bottom of the *Receiver GUI*.

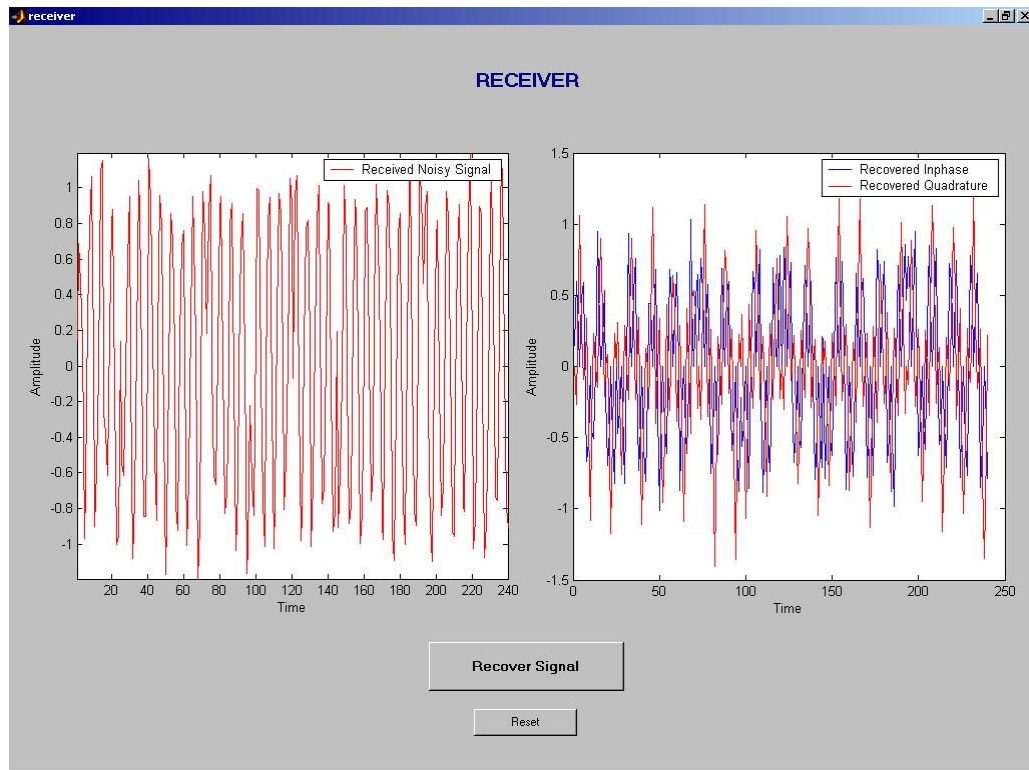


Figure 7: An image of the *Receiver GUI* with received BPSK and demodulated signal.

Conclusions

The development of an interactive and user-friendly environment for digital modulation systems has been presented. The use of the developed system has also been demonstrated through a case study. This environment has proven to be a valuable educational tool for understanding the effect of various modulation parameters for different digital modulation techniques. In this manner, the package can easily be used as a computer-aided teaching facility in a digital and data communication course, where the visual stimulus is important in grasping the underlying principles and concepts. This environment can also be used in conjunction with handouts and PowerPoint presentations to make the ideas and fundamental concepts more clear. It also removes the need for a technology student to write software code to implement the modulation techniques, which are sometimes difficult to implement and time consuming. Instead, the students can utilize their time to learn the physical implications of various modulation parameters.

The design of the developed environment is made flexible to accommodate any future modifications and enhancements to customize various teaching scenarios. Features, like bandwidth calculations, prompting the user for a particular input and reading data files from memory can be incorporated to make this GUI much more resourceful. The Matlab also has a facility (Matlab web server) to provide the environment over the Internet, so that students can also exploit this through the web at their convenience.

References

- Auslander, D.M., Ridgely, J. R. and Ringgenberg, J. D. (2002). *Control Software for Mechanical Systems: Object-Oriented Design in a Real-Time World*, Prentice-Hall.
- Azad, A. K. M. and Tokhi, M. O. (2003). An interactive environment for simulation and control of flexible manipulator systems, *Proceedings of the 9th IEEE International Conference on Emerging Technologies and Factory Automation*, Calouste Gulbenkian Foundation, Lisbon, Portugal, 16-19 September.
- Chen, Y-C and Naughton, J. M. (2000). An undergraduate laboratory platform for control system design, simulation, and implementation, *IEEE Control System Magazine*, June 2000, 12-20.
- Mathworks (2001). *Creating Graphical User Interface*, MathWorks Incorporated, MA, USA.
- Marchand, R. and Holland, O. T. (2003). *Graphics and GUIs with MATLAB*, Chapman and Hall/CRC.
- Miller, G. M. and Beasley, J. S. (2002). *Modern Electronic Communication*, 7th Edition, Prentice Hall, New Jersey.
- Penfield, P. and Larson, R. C. (1996). Education via advanced technologies, *IEEE Transactions on Education*, **39**(3), 436-443.
- Tokhi, M.O., Azad, A.K.M. and Powrwanto, H. (1999). SCEFMAS: A simulink environment for dynamic characterization and control of flexible manipulators, *International Journal of Engineering Education*, **15**(3), 213-226.
- Tomasi, W. (2001). *Electronic Communication Systems- Fundamentals through advanced*, 4th Edition, Prentice-Hall, New Jersey.

Biography

ABUL K M AZAD

Obtained PhD (control engineering) from the University of Sheffield (UK) in 1994. He is now an Assistant Professor with the Engineering Technology Department of NIU. He has worked at various academic and industrial establishments since graduation in 1987. His research interests include mechatronics, vibration control, real-time computer control of engineering systems, adaptive/intelligent control, and mobile robotics. Dr Azad has over 60 papers in these areas. He has active membership and involvement in several learned societies including the IEE, IEEE, ASEE, Epsilon Pi Tau, and a senior member of ISA.