Teaching Convolutional Coding using MATLAB in Communication Systems Course

Davoud Arasteh

Department of Electronic Engineering Technology
Southern University, Baton Rouge, LA 70813, USA

Abstract

Convolutional codes are channel codes, which are extensively used in communication systems like GSM (global system for mobile communications) and Interim standard IS-95. We introduce a strategy to present convolutional codes to students learning wireless communication systems, digital communication or similar courses, without using mathematical structures. In this paper, we will discuss our method to implement convolutional encoding and Viterbi decoding to determine the bit error rate. We have compared its performance under different conditions. To exemplify and illuminate our approach, we have selected a communication channel with rate 1/3 convolutional code. Constrain length is three for both Viterbi hard decision decoding method and Viterbi soft decision decoding method. Results suggest that soft decision decoding has at least 3.24 dB improvements in SNR compared to hard decision decoding for same bit error rate. Also by using soft decision Viterbi decoding on AWGN channel, we have examined various convolutional codes for four different constrain lengths and rates. We find that the first code with rate 1/3 code and constrain length 3 has better bit error rate performance with respect to other coding methods. This means, it gives a lower bit error rate for the same value of signal to noise ratio used in other coding schemes. Thus, to achieve the same bit error rate the first code will require a lower SNR. A lower SNR means a lower transmitter power. However, if we use this code the bandwidth requirement is three times larger compared with an un-coded transmission. For the second coding scheme we have an increase in bandwidth by a factor of 2 and for the third and fourth coding scheme the increase factor is 1.5. We believe this approach and its corresponding results will expose students to conceptual and technical aspects of signal encoding and its analysis.

Introduction

Our main objective of using MATLAB® tools is to develop an applicable knowledge of the constituent components necessary to cover in digital communication or wireless communication courses. Generally in these courses the topics include source coding (Huffman, Arithmetic, and Dictionary Codes), signal detection in the presence of white noise, digital modulation and demodulation schemes, error performance analysis, channel coding (cyclic and convolutional codes) and spread spectrum techniques. For each topic we illustrate the basic notions through MATLAB simulation examples. In this paper we introduce a strategy to present convolutional codes to students learning wireless communication systems, digital communication or similar courses, without using mathematical structures. We will discuss our method to implement convolutional encoding and Viterbi decoding to determine the bit error rate to illustrate its application. We have compared its performance under different conditions. To exemplify and illuminate our approach, we have selected a communication channel with rate 1/3 convolutional code. Constrain length is three for both Viterbi hard decision decoding method and Viterbi soft decision decoding method.

Convolutional Coding

The main aim of a digital communication system is to transmit information reliably over a channel. The available amount of transmitter power and bandwidth are the major constraints in the design of a digital communication system. The channel can be coaxial cables, microwave links, or fiber optic. The channel is subject to various types of noise, distortion, and interference. Also some communication systems have limitation on transmitter power. All these may lead to errors. Consequently we may need some form of error control encoding to recover the information reliably. Convolutional codes are extensively used for real time error correction. The position of the channel encoder is shown in following block diagram of the elements of a digital communication system.

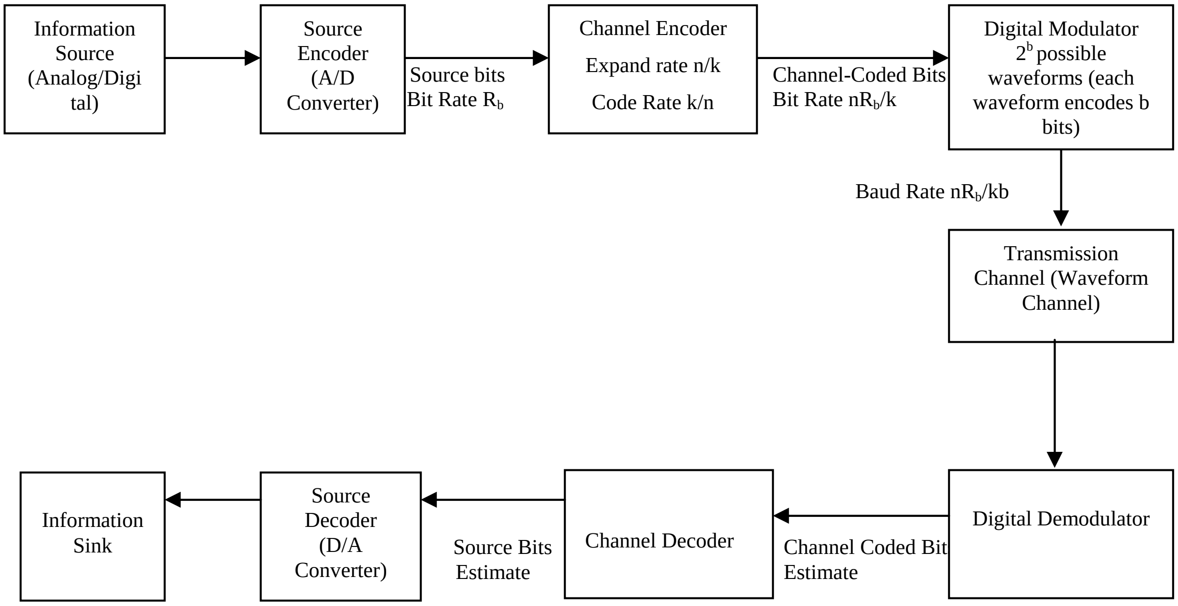

Figure 1 Channel encoder/decoder position in the block diagram of a digital communication system

In 1948, Claude Shannon proved that any communication channel could be characterized by maximum theoretical capacity, C. If the source information rate, R, is less than C ($R<C$), then there exist channel-encoding method such that information can be reliably transmitted. This theorem set a theoretical limit on possible information rate for achieving reliable (error-free) transmission through the channel by appropriate coding. On the other hand, if $R>C$, reliable transmission is not possible regardless of amount of signal processing performed at the transmitter and receiver. Thus if the required transmission rate R (measured in bits per second) is less than C, it is possible to design a communication system for that channel and with the help of error-control coding one can achieve a very small probability of output error for that channel. The capacity C in bits per second (b/s) depends on only two parameters, the channel bandwidth and the signal-to-noise ratio. In practice, it has proved to be remarkably difficult to find classes of constructive codes that can be decoded by feasible decoding algorithms at rates, which come at all close to the Shannon limit. Within the past decade there have been remarkable breakthroughs, principally the invention of turbo codes [1] and the rediscovery of Low-Density Parity Check codes [2], which have allowed the capacity of AWGN (Additive White Gaussian Noise) channels to be approached in a practical sense to the theoretical Shannon limit than any other code so far.

Error-control coding can be used for a number of different applications. Codes can be used to achieve reliable communication in presence of interference. In military applications error control codes are used to protect information from intentional enemy interference. In case of satellite communication, there are severe limitations on transmitter power. So with the help of error control coding we can correctly recover very weak messages. Even when the received signal power is close to thermal noise power, error control coding is used to achieve reliable communication. The deep-space communications application has been the arena in which most of the most powerful coding schemes for the power-limited AWGN channel have been first deployed, because the only noise is AWGN in the receiver front end; bandwidth is effectively unlimited; power fractions have huge scientific and economic value; and receiver (decoding) complexity is effectively unlimited.

Digital autopilots, digital process-control systems, digital switching systems, and digital radar signal processing all are systems that involve large amounts of digital data transfers between interconnected subsystems. In all these cases, error control coding is essential to maintain proper performance.

There are many different types of error control codes like BCH codes, Reed Solomon codes, Linear Block codes, Turbo codes, Convolutional codes. Different factors affect the choice of a particular coding scheme. Constraints like cost, power, bandwidth, type of channel, allowable Delay in Decoding, data rate and type of information play a major role in selection of a particular coding scheme. However, Reed Solomon codes, Turbo codes, and Viterbi decoded Convolutional codes are more frequently used with respect to other error control.

Passing the information sequence to be transmitted through a linear finite state shift register generates a convolutional code. Additional combinatorial logic that performs modulo-two addition is also used. The input data to the encoder is shifted into and along the shift register, k bits at a time. The number of output bits for each k bits input sequence is n bits. Thus the code rate is k/n. In convolutional coding, an information frame together with the previous m information frames are encode into a single codeword frame. Hence successive frames are coupled together by the encoding procedure. Codes obtained this way are called tree codes and tree codes with additional properties of linearity and time invariance are called convolutional codes. Convolutional coding with Viterbi decoding has been the predominant FEC technique used in space communication, particularly in geostationary satellite communication networks, such as VSAT (very small aperture terminal) networks. The first large-scale application includes a rate-1/2 convolutional code with constraint length 20 for the Pioneer 1968 mission. The receiver used 3-bit soft decisions and sequential decoding implemented on a general-purpose 16-bit minicomputer with a 1 MHz clock rate. At 512 bps, the actual coding gain achieved at Pb(E)=0.005, was about 3.3 dB. In VSAT rate 1/2 convolutional coding with constraint length 7 is generally used. With this code, you can transmit BPSK or QPSK signals with at least 5 dB less power than you need without it. This is very useful in reducing transmitter and/or antenna cost or permitting increase data rates given the same transmitter power and antenna sizes. But there is a trade off if the modulation technique stays the same; the bandwidth expansion factor of a convolutional code will be n/k. As an example consider the encoder shown in figure 2.

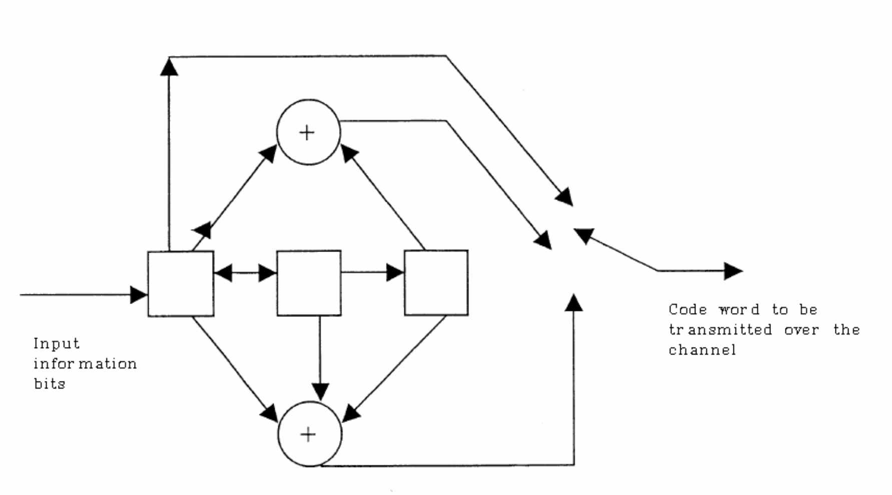

Figure 2 The convolutional encoder with rate (k/n) = 1/3, and constrain length K=3

This is a rate (k/n) = 1/3, with constrain length K=3 convolutional encoder. Here k is the number of parallel input information bits and n is the number of parallel output encoded bits at one time interval. The constraint length, K, of the convolutional encoder is defined by K=M+1, where M is the maximum number of memories in any shift register. Generally convolution codes are described by their generator polynomials, for this example they are:
$g_1(D)=D^2$
$g_2(D)=D^2+l$
$g_3(D)=D^2 + D+l$

Other useful methods used for their description are the *state transition diagram* and *Trellis structure* [3, 4]. At the receiver end the decoding strategy for convolutional codes is Viterbi algorithm. In next section we will use MATLAB capabilities to simulate convolutional encoding and Viterbi decoding [5].

Algorithm

The steps involved in simulating a communication channel using convolutional encoding and Viterbi decoding are as follows:

(1) Generating the data: The data to be transmitted through the channel is generated using *randn* function of Matlab in combination with the sign function. We have generated 4000 bits.

(2) Convolutionally encoding the data: This is done in two steps. In the first step, the *poly2trellis* function is used. It accepts a polynomial description of a convolutional encoder and returns the corresponding trellis structure description. In the next step the output of *poly2trellis* is suitable as an input to the *convene* and *vitdec* functions. The parameters for the *poly2trellis* function are “constraint length” and “code” generator polynomial”. The output of this is then used as one of the parameter for the *convene* function, along with the original data sequence. The convene function actually encodes the data bits.

(3) Adding noise to the transmitted symbols: The AWGN function adds white Gaussian noise to the channel symbols produced by the encoder. The parameters for this

function are the coded symbols, SNR (signal to noise ratio), the state, and the power type (whether in “dB” or “linear”).

(4) Decoding: The decoding is done for two different cases. The first one is “hard decision decoding” and the second is “soft decision decoding”. The *vitdec* function is used for this purpose. The input parameters required for this function are “code”, “trellis”, “tblen” (trace back length), “opmode”(operation mode) and “dectype”(decoder type). For “opmode” we have used the “trunc” option, i.e. truncated mode of operation. In this the encoder is assumed to have started at the all-zero state. The decoder traces back from the state with the best metric. The “dectype” can have three alternatives, “unquant”, “hard” and “soft”. For hard decision decoding we use ‘hard’, and for ‘soft’ for soft decision decoding. Besides selecting ‘soft’ for “dectype”, parameter “nsdec” need to be defined since our code consists of integers between 0 and $2^{(nsdec-1)}$. Before performing the decoding step, this can be determined by number of quantized levels for received channel symbols. For hard decision decoding, the symbols are quantized to one bit precision while for soft decision decoding, data bits are quantized to three or four bits of precision. In our simulation we have used three bits (i.e. eight levels). The selection of quantization levels is an important design decision because of its significant effect on the performance of the link.

(5) Bit error rate: The function *biterr* is used for calculating the number of errors and the bit error rate. The input parameters for this function are the “original data sequence” and “the decoded sequence”. Number of errors can be easily obtained by simply subtracting it bit by bit. The ratio of number of bit in error upon total number of bits gives us the bit error rate.

(6) Plot: The simulation is run several times for different SNR values (ranging from 1 to 20) for the same code. The results are plotted using *semilogy* function, which has a logarithmic scale for Y-axis. Finally the simulation has been run again for different codes. The combined plot is useful to compare all the codes. Also, hard and soft decision decoding is compared for one particular code. Here we have not included the steps of modulating the channel symbols onto a carrier and that of demodulating the received carrier to recover the channel symbols. This is because even if we omit these steps, still we can accurately model the effects of AWGN channel.

Results

Error performance analysis is performed by plotting the bit error-rate versus signal to noise ratio (SNR) for AWGN. Simulations were run for different codes. Figure (3) show the simulation results for rate 1/3 code with constrain length, K=3. Here we have compared the ‘Hard decision decoding’ to the ‘Soft decision decoding’. In literatures, it is believed that, soft decision decoding is always at least 2dB better than hard decision decoding [6]. For the code used, our simulation result shows that, the soft decision decoding is at least 3.24 dB better response compared to hard decision decoding.

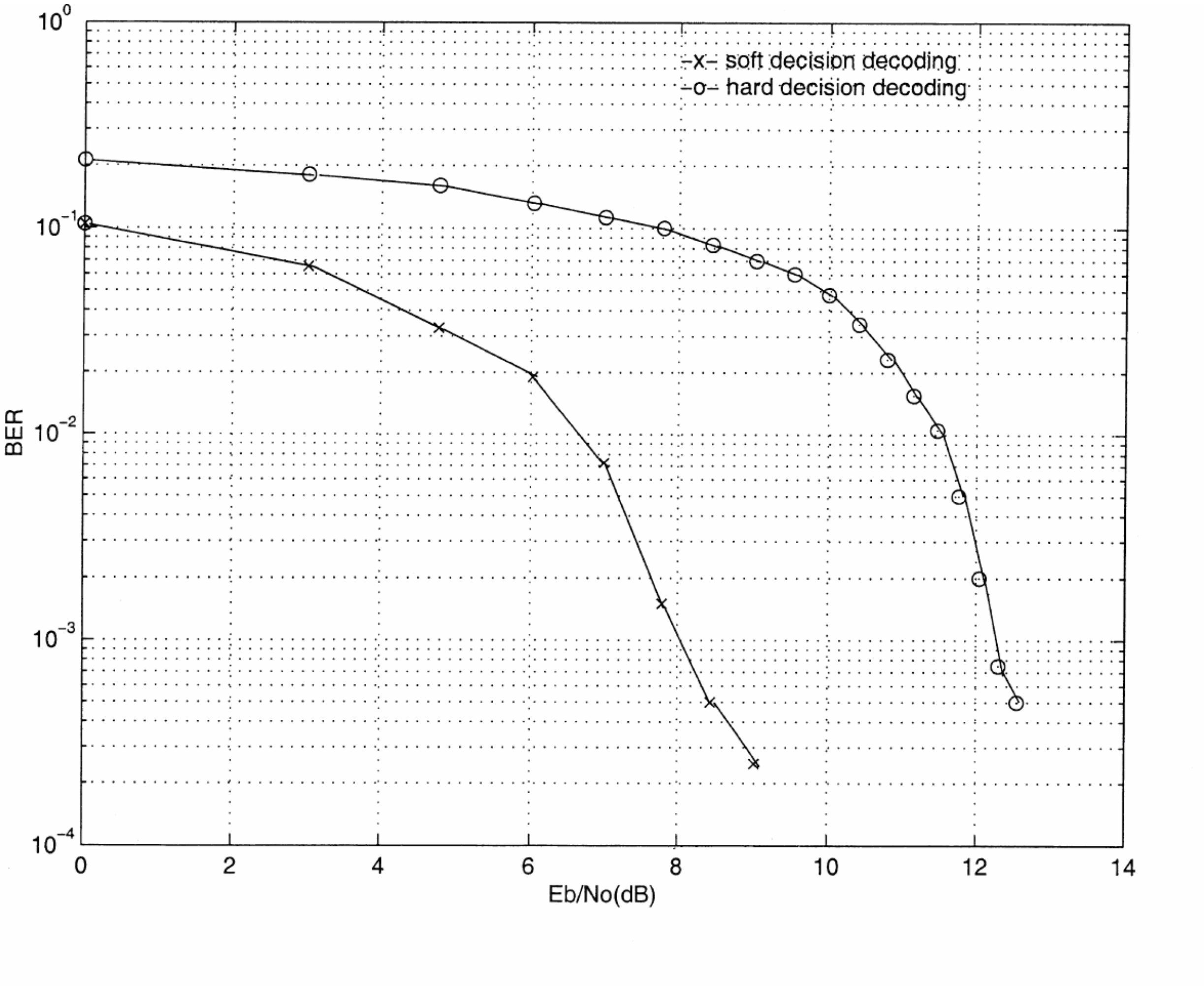

Figure 3 Simulation Results for Rate 1/3 Convolutional code (constrain length=3) with Viterbi (hard decision and soft decision) decoding on AWGN channel.

Figure (4) shows simulation results for various codes. The four codes compared are,

(1) Rate 1/3 code with constrain length=3 and generator polynomial=[4, 5, 7].
(2) Rate 1/2 code with constrain length=3 and generator polynomial = [6, 7].
(3) Rate 2/3 code with constrain length=[4, 3] and
generator polynomial =[4 5 17; 7 4 2].

(4) Rate 2/3 code with constrain length=[5, 4] and
generator polynomial =[23 35 0; 0 5 13].

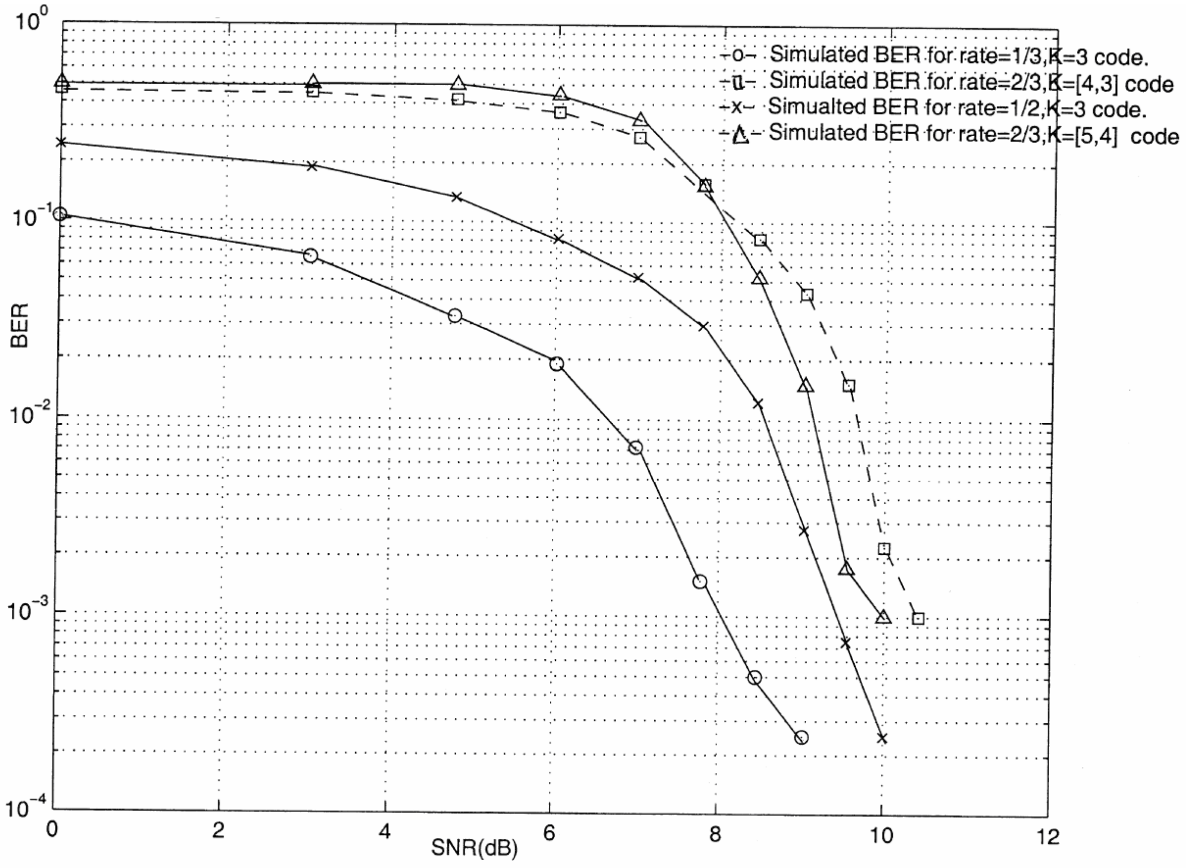

Figure 4 Simulation results for various Convolutional codes (different constrain lengths and rate) using Soft decision Viterbi Decoding on AWGN channel

We find that the first code performance is better than the other three codes. It gives a lower BER for the same value of SNR. Thus to achieve the same BER, the first code will require a lower signal to noise ratio, i.e. lower transmitter power, compared to other three codes. However if we use this code the bandwidth requirement is three times more compared with uncoded transmission (as the rate is 1/3). For the second code we have an increase in bandwidth by a factor ‘2’ and for the third and fourth it is ‘1.5’.

Convolutional coding is an effective method for trading bandwidth and implementation complexity against transmitter power. Convolutional codes are highly suitable for AWGN channels, where soft decision decoding is relatively straightforward. However, many types of conditions give rise to non-Gaussian conditions where the soft decision decoding may need to adapt to the channel conditions and where the channel coherence may mean that Viterbi decoding is no longer the maximum likelihood solution. Also Turbo codes are bandwidth efficient means of achieving similar coding gains.

Conclusion

The contribution of this paper is to provide a simulating tool that teaches efficiently the convolutional encoding in digital communication, and wireless communication courses. This method can be effectively use in similar simulation practices and can be run to compare various codes and decide which code to use for a specific application in communication systems. Simulations for the various types of convolutional encoding have been presented to determine the bit error rate for each type in conjunction with the use of the encoding strategy.

References

1. C. Berrou, A. Glavieux, and P. Thitimajshima, “Near Shannon Limit Error- Correcting Coding and Decoding: Turbo Codes,” Proceeding of the IEEE International Conference on Communications, Geneva, Switzerland, May 1993. [
2. R. G. Gallager, Low-Density Parity-Check Codes. Cambridge, MA: MIT Press, 1962.
3. J. G. Proakis, *Digital Communication,* fourth Edition, McGraw Hill, 1998.
4. D. P. Agrawal, Q. Zeng, *Introduction to Wireless and Mobile Systems*, Second Edition, Thomson, 2006.
5. J. G. Proakis, M. Salehi, G. Bauch, “*Contemporary Communication Systems Using MATLAB® and Simulink®*”, Second Edition, Thomson, 2004.
6. A. J. Viterbi, “*Error bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm,*” IEEE Transaction on Information Theory, Vol. IT-13, 1967.

DAVOUD ARASTEH
Davoud Arasteh serves as an assistant professor of Electronic Engineering Technology at Southern University of Baton Rouge. He has an extensive experience in curriculum development on senior level and laboratory environments. His research interests include Mobile Computing, Network Security, Nonlinear Dynamical Systems, Computer Vision, Condense Matter Physics, and Technology Based Engineering Education. He is the chair of departmental curriculum committee and is a member of ASEE, IEEE, IEEE Computer Society and SPIE.