

UNIVERSITY OF MAURITIUS

FACULTY OF ENGINEERING

**DEPARTMENT OF ELECTRICAL AND ELECTRONIC
ENGINEERING**

ELEC 1200 VACATION TRAINING 2005

THE MATLAB COMMUNICATION TOOLBOX

TUTORIAL FOR

**BEng (Hons) Electrical and Electronic/ Electronic and
Communication Engineering/BSc (Hons) Information and
Communication Technologies/ BSc (Hons) Electronics and
Computer Science
Level 1 students**

Prepared by

Dr M.A. Hosany

MAY 2005

This tutorial aims at providing a brief overview of the MATLAB® Communications Toolbox with functions, plots, and a graphical user interface for exploring, designing, analyzing, and simulating digital modulation techniques of a digital communication system. The tutorial provides a comprehensive guide to understand how to use the digital modulation and demodulation functions. A short overview of how to plot constellations for visualizing communications signals as well as computing error rates of the designed system are given.

1. Modulation: An Introduction

One way to communicate a message signal whose frequency spectrum does not fall within that fixed frequency range, or one that is otherwise unsuitable for the channel, is to change a transmittable signal according to the information in the message signal. This alteration is called *modulation*, and it is the modulated signal that is transmitted. The receiver then recovers the original signal through a process called *demodulation*.

Modulation is a process by which a *carrier signal* is altered according to information in a *message signal*. The *carrier frequency*, denoted F_c , is the frequency of the carrier signal. The *sampling rate*, F_s , is the rate at which the message signal is sampled during the simulation.

The frequency of the carrier signal is usually much greater than the highest frequency of the input message signal. The Nyquist sampling theorem requires that the simulation sampling rate F_s be greater than two times the sum of the carrier frequency and the highest frequency of the modulated signal, in order for the demodulator to recover the message correctly.

1.1 Baseband Versus Passband Simulation

For a given modulation technique, two ways to simulate modulation techniques are called *baseband* and *passband*. Baseband simulation requires less computation. The MATLAB® Communication toolbox supports baseband simulation for digital modulation and passband simulation for analog modulation. In this tutorial, baseband simulation will be used.

1.2 Digital Modulation Techniques

1.2.1 Amplitude Shift Key (ASK) Modulation

In this method the amplitude of the carrier assumes one of the two amplitudes dependent on the logic states of the input bit stream. A typical output waveform of an ASK modulator is shown in Fig. 1.

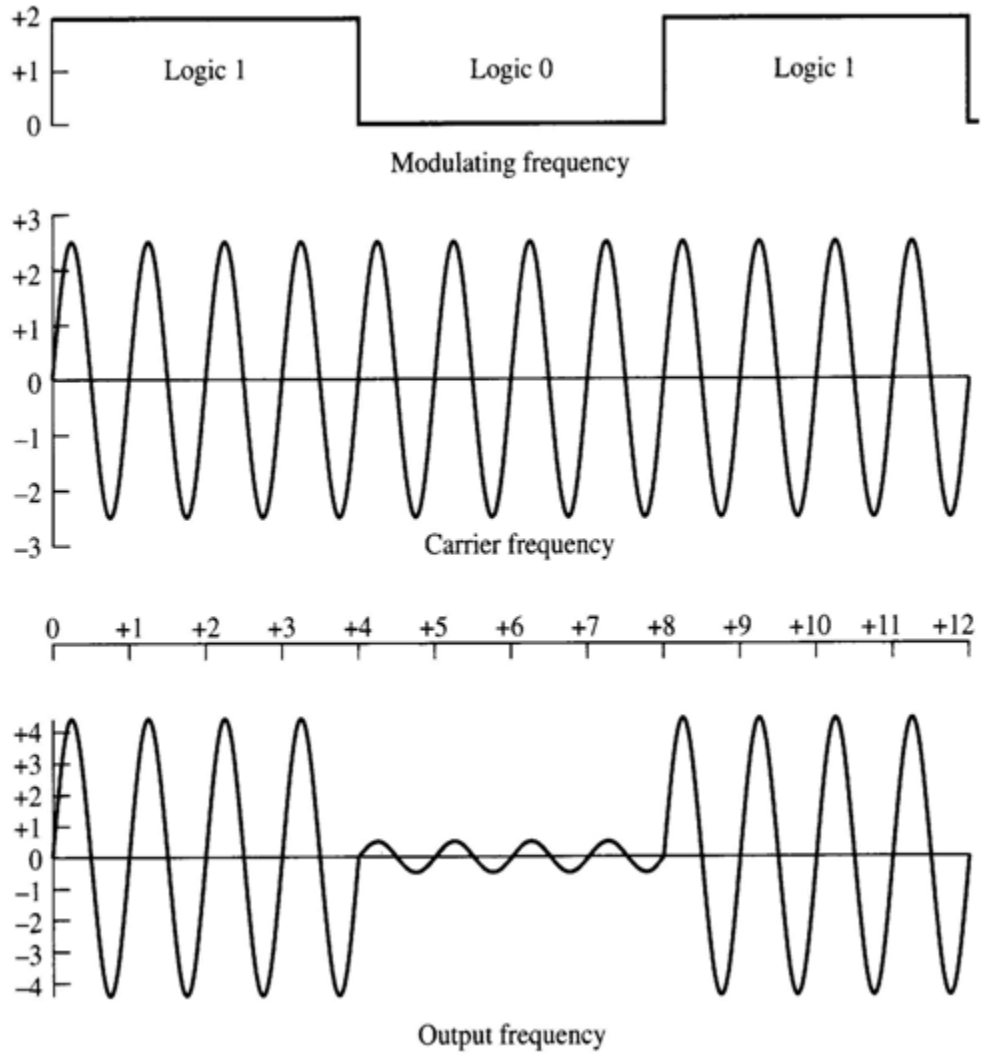


Fig. 1: ASK Modulation

1.2.2 Frequency Shift Key (FSK) Modulation

In this method the frequency of the carrier is changed to two different frequencies depending on the logic state of the input bit stream. The typical output waveform of an FSK is shown in Fig. 2. Notice that a logic high causes the centre frequency to increase to a maximum and a logic low causes the centre frequency to decrease to a minimum.

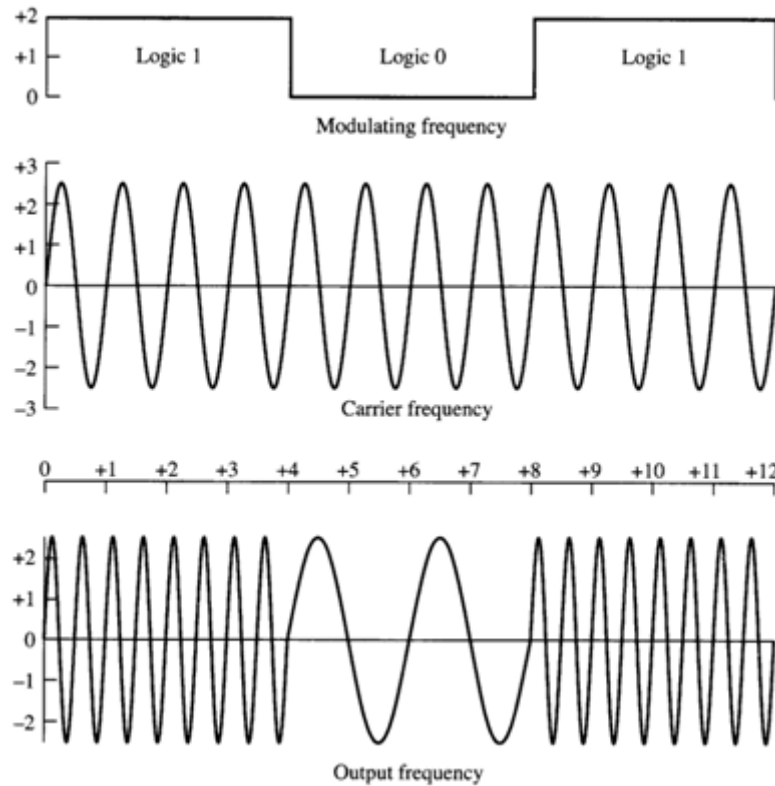


Fig. 2: FSK Modulation

1.2.3 Phase Shift Key (PSK) Modulation

With this method the phase of the carrier changes between different phases determined by the logic states of the input bit stream.

There are several different types of *Phase Shift Key* (PSK) modulators. These are:

- Two-phase (2 PSK)
- Four-phase (4 PSK)
- Eight-phase (8 PSK)
- Sixteen-phase (16 PSK)

Two-Phase Shift Key Modulation

In this modulator the carrier assumes one of two phases. A logic 1 produces no phase change and a logic 0 produces a 180° phase change. The output waveform for this modulator is shown in Fig. 3.

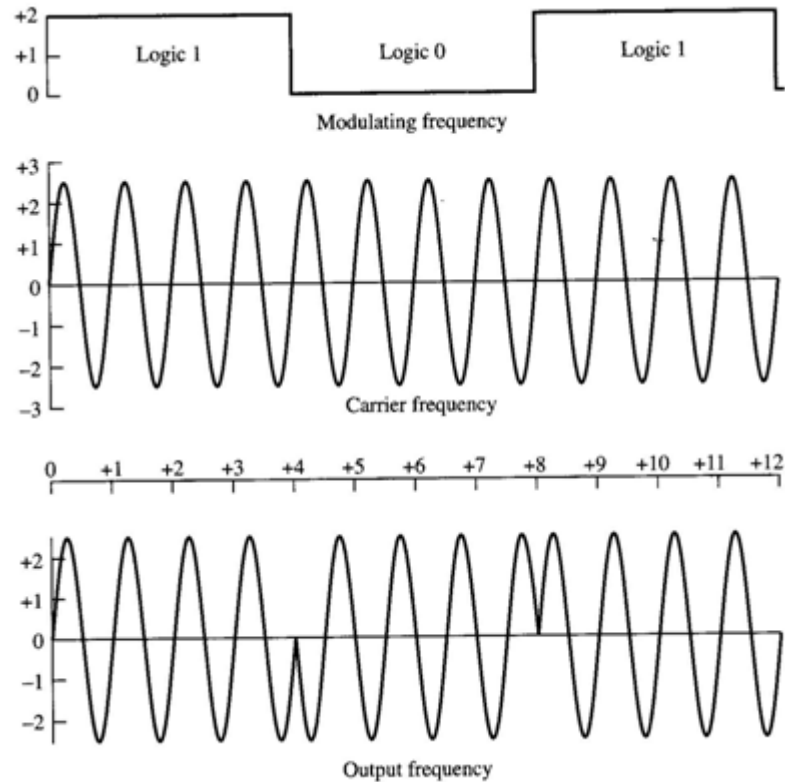


Fig. 3: 2PSK or BPSK Modulation

Four-Phase Shift Key Modulation

With *4 PSK or QPSK*, 2 bits are processed to produce a single-phase change. In this case each symbol consists of 2 bits. The actual phases that are produced by a 4 PSK modulator are shown in Table 1.

Table 1: Bits and Phases for 4 PSK modulation

Bits	Phase
00	45°
01	135°
10	315°
11	225°

From Table 1, a signal space diagram or signal constellation can be drawn as shown in Fig. 4. Note from Fig. 4 that from any two closest bits sequences, there is only one bit change. This is called Gray Coded scheme. For example, bit sequence '00' has one bit change for its closest bit sequences '01' and '10'.

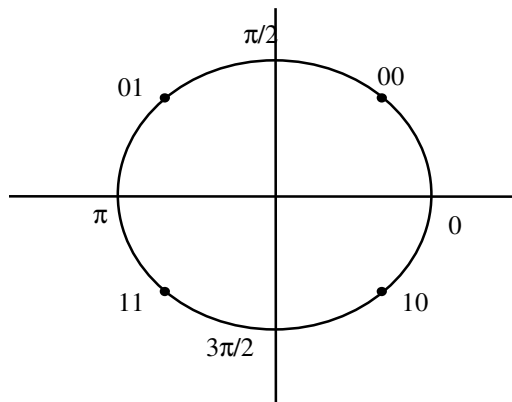


Fig. 4: 4PSK constellation

Eight-Phase Shift Key Modulation

With this modulator 3 bits are processed to produce a single-phase change. This means that each symbol consists of 3 bits. Fig. 5 shows the constellation and mapping of the 3-bit sequences onto appropriate phase angles.

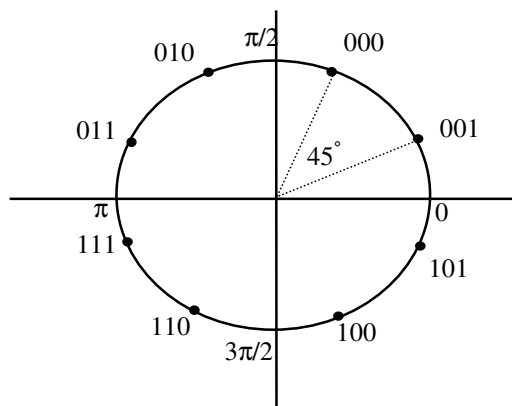


Fig. 5: 8 PSK signal constellation

Higher Order PSK modulation schemes

Modulation schemes like 16 PSK, 32 PSK and higher orders can also be designed and represented on a signal space diagram.

1.2.4 Quadrature Amplitude Modulation (QAM)

QAM, is a method for sending two separate (and uniquely different) channels of information. The carrier is shifted to create two carriers namely the sine and cosine versions. The outputs of both modulators are algebraically summed and the result of which is a single signal to be transmitted, containing the In-phase (I) and Quadrature (Q) information. The set of possible combinations of amplitudes, as shown on an x-y plot, is a pattern of dots known as a *QAM constellation*.

Consider the 16 QAM modulation scheme. With this modulator, 4 bits are processed to produce a single vector. The resultant constellation consists of four different amplitudes distributed in 12 different phases as shown in Fig. 6.

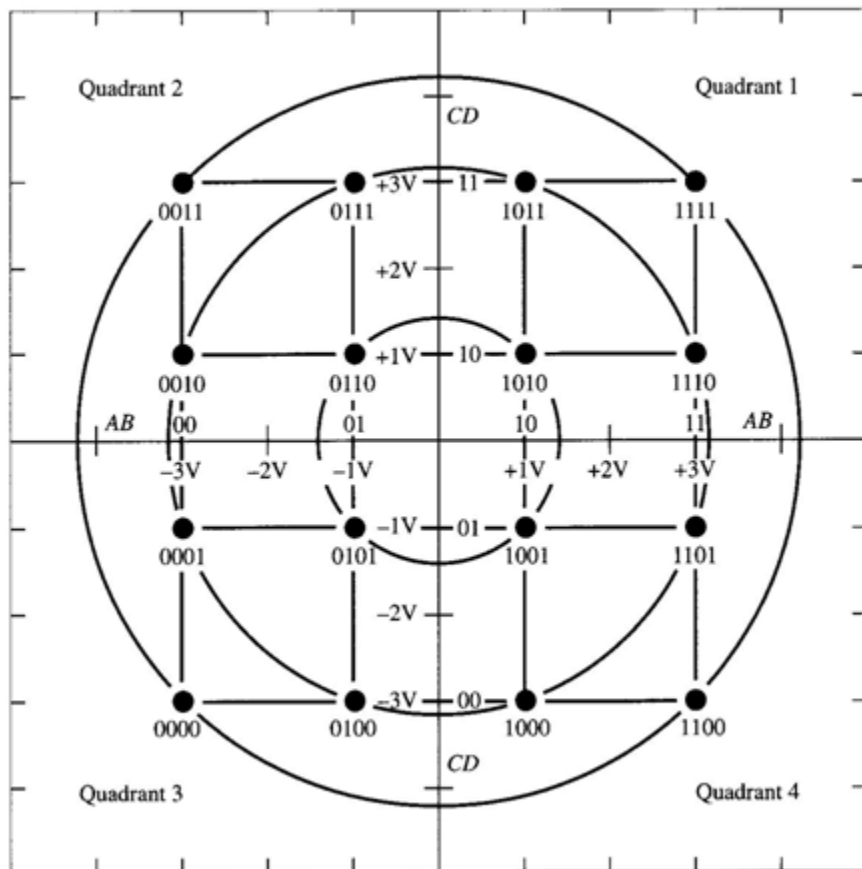


Fig. 6: 16 QAM Constellation

2. QAM Modulation and MATLAB®

To modulate a signal using digital modulation with an alphabet having M symbols, start with a real message signal whose values are integers between 0 and M. Represent the signal by listing its values in a vector, x. Alternatively, a matrix can be used to represent a multichannel signal, where each column of the matrix represents one channel.

For example, if the modulation uses an alphabet with 8 symbols, then the vector [1 2 3 1 0 4 4 2 5]' is a valid single-channel input to the modulator. As a multichannel example, the two-column matrix

```
[2 3;  
3 3;  
7 3;  
0 3;]
```

defines a two-channel signal in which the second channel has a constant value of 3.

Problem definition: A simulation study must be carried out for a binary data stream that has to be transmitted over a channel known as Additive White Gaussian Noise (AWGN) Channel using 16QAM modulation scheme.

Solution: Use MATLAB® Communication Toolbox to simulate the system. The latter will consist of a 16QAM baseband modulator, AWGN channel, and 16QAM demodulator. The system's bit error rate (BER) is computed and also the transmitted and received signals will be displayed in a scatter plot.

The table below indicates the key tasks in solving the problem, along with relevant functions from the MATLAB® Communications Toolbox. The solution arbitrarily chooses baseband 16QAM as the modulation scheme and AWGN (additive white Gaussian noise) as the channel model.

Task	Function
Generate a random binary data stream	<i>randint</i>
Modulate using 16-QAM	<i>dmodce</i>
Add white Gaussian noise	<i>awgn</i>
Create a scatter plot	<i>scatterplot</i>
Demodulate using 16-QAM	<i>ddemodce</i>
Compute the system's BER	<i>biterr</i>

The sections below describe each step in more detail, introducing M-code along the way. To view all the code in one editor window, enter the codes in the MATLAB Command Window.

edit commdoc_mod

2.1 Generate a Random Binary Data Stream

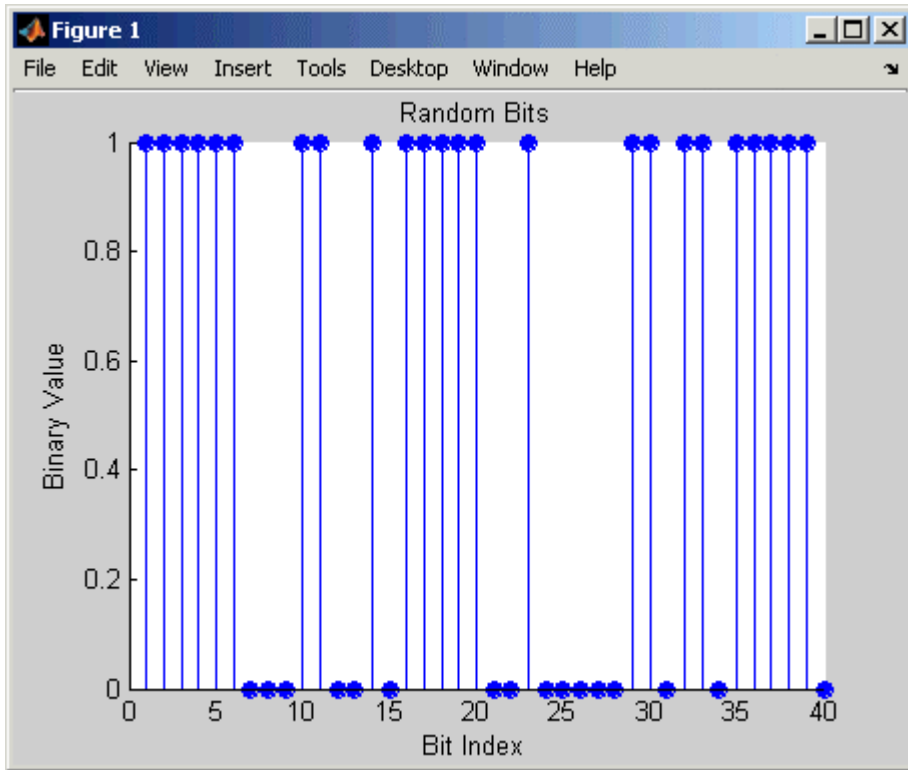
Use the *randint* function to create a column vector that lists the successive values of a binary data stream. Set the length of the binary data stream to 30,000.

The code below creates a stem plot of a portion of the data stream, showing the binary values. Figure 1 shows the stem plot of the input data stream.

```
%% Setup
% Define parameters.
M = 16; % Size of signal constellation
k = log2(M); % Number of bits per symbol
n = 3e4; % Number of bits to process = 30,000
Fd=1;Fs=1; %Input message sampling frequency, output message sampling
frequency
nsamp = 1; % Oversampling rate

%% Signal Source
% Create a binary data stream as a column vector.
x = randint(n,1); % Random binary data stream

% Plot first 40 bits in a stem plot.
stem(x(1:40), 'filled');
title('Random Bits');
xlabel('Bit Index'); ylabel('Binary Value');
```



2.2 Prepare to Modulate

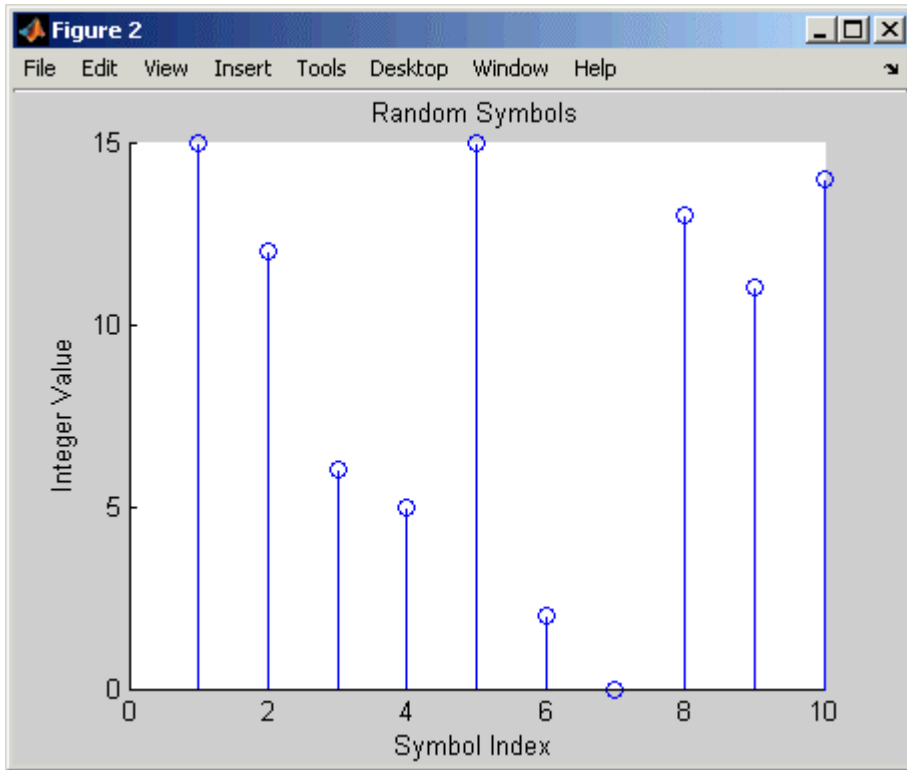
Each 4-tuple of values from x is arranged across a row of a matrix, using the *reshape* function in MATLAB, and then the *bi2de* function is applied to convert each 4-tuple to a corresponding integer. (The *.* characters after the *reshape* command form the unconjugated array transpose operator in MATLAB. Figure 2 shows the random symbols being generated.

```

%% Bit-to-Symbol Mapping
% Convert the bits in x into k-bit symbols.
xsym = bi2de(reshape(x,k,length(x)/k).', 'left-msb');

%% Stem Plot of Symbols
% Plot first 10 symbols in a stem plot.
figure; % Create new figure window.
stem(xsym(1:10));
title('Random Symbols');
xlabel('Symbol Index'); ylabel('Integer Value');

```



2.3 Modulate Using 16-QAM

The `dmodce` function implements a 16-QAM modulator. `xsym` from above is a column vector containing integers between 0 and 15. The `dmodce` function can now be used to modulate `xsym` using the baseband representation. Note that M is 16, the alphabet size.

```
%% Modulation
% Modulate using 16-QAM.
y = dmodce(xsym,Fd,Fs, 'qask',M);
```

The result is a complex column vector whose values are in the 16-point QAM signal constellation. A later step in this example will show what the constellation looks like.

2.4 Add White Gaussian Noise (AWGN) Channel

Applying the `awgn` function to the modulated signal adds white Gaussian noise to it. The ratio of bit energy to noise power spectral density, E_b/N_0 , is arbitrarily set at 10 dB. The expression to convert this value to the corresponding signal-to-noise ratio (SNR) involves k , the number of bits per symbol (which is 4 for 16-QAM), and `nsamp`, the oversampling factor (which is 1 in this example). The factor k is used to convert E_b/N_0 to an equivalent E_s/N_0 , which is the ratio of *symbol* energy to noise power spectral density. The factor `nsamp` is used to convert E_s/N_0 in the symbol rate bandwidth to an SNR in the sampling bandwidth.

```

%% Transmitted Signal
ytx = y;

%% Channel
% Send signal over an AWGN channel.
EbNo = 10; % In dB
snr = EbNo + 10*log10(k) - 10*log10(nsamp);
ynoisyy = awgn(ytx, snr, 'measured');

%% Received Signal
yrx = ynoisy;

```

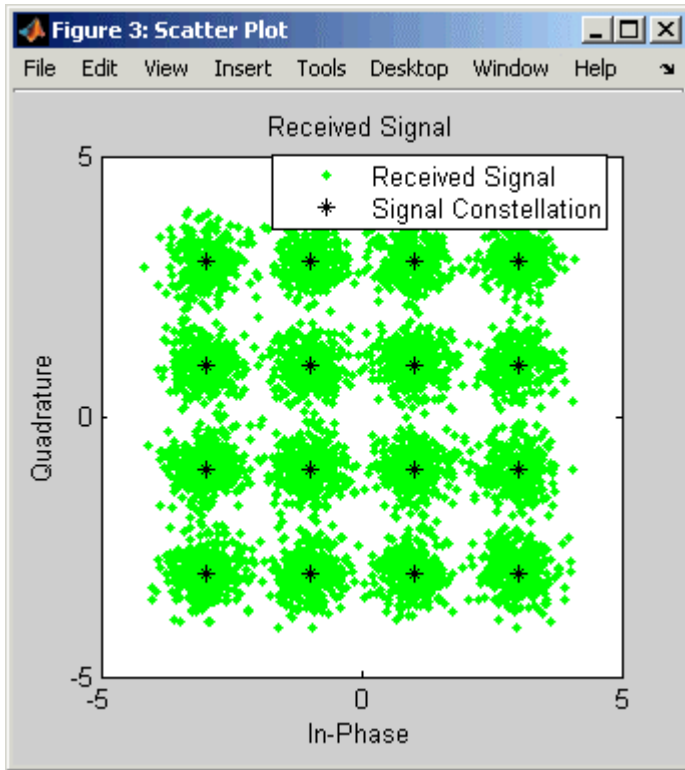
2.5 Create a Scatter Plot

The *scatterplot* function is applied to the transmitted and received signals. This shows how the signal constellation looks like and how the noise distorts the signal. In the plot, the horizontal axis is the In-phase (I) component of the signal and the vertical axis is the Quadrature (Q) component. The code below also uses the title, legend, and axis functions in MATLAB® to customize the plot. Figure 3 shows the received signal being distorted.

```

%% Scatter Plot
% Create scatter plot of noisy signal and transmitted
% signal on the same axes.
h = scatterplot(yrx(1:nsamp*5e3), nsamp, 0, 'g. ');
hold on;
scatterplot(ytx(1:5e3), 1, 0, 'k*', h);
title('Received Signal');
legend('Received Signal', 'Signal Constellation');
axis([-5 5 -5 5]); % Set axis ranges.
hold off;

```



2.6 Demodulate Using 16-QAM

Demodulation of the received 16-QAM signal is done by using the `ddemodce` function. The result is a column vector containing integers between 0 and 15.

```
%% Demodulation
% Demodulate signal using 16-QAM.
zsym = ddemodce(yrx,Fd,Fs, 'qask', M);
```

2.7 Convert the Integer-Valued Signal to a Binary Signal

The previous step produced `zsym`, a vector of integers. To obtain an equivalent binary signal, use the `de2bi` function to convert each integer to a corresponding binary 4-tuple along a row of a matrix. Then use the `reshape` function to arrange all the bits in a single column vector rather than a four-column matrix.

```
%% Symbol-to-Bit Mapping
% Undo the bit-to-symbol mapping performed earlier.
z = de2bi(zsym,'left-msb'); % Convert integers to bits.
% Convert z from a matrix to a vector.
z = reshape(z.',prod(size(z)),1);
```

2.8 Compute the System's BER

The *biterr* function is now applied to the original binary vector and to the binary vector from the demodulation step above. This yields the number of bit errors and the bit error rate.

```
%% BER Computation
% Compare x and z to obtain the number of errors and
% the bit error rate.
[number_of_errors, bit_error_rate] = biterr(x, z)
```

The statistics appear in the MATLAB Command Window. Results might vary because the example uses random numbers.

```
number_of_errors = 71
```

```
bit_error_rate = 0.0024
```

2.9 Plotting Signal Constellations

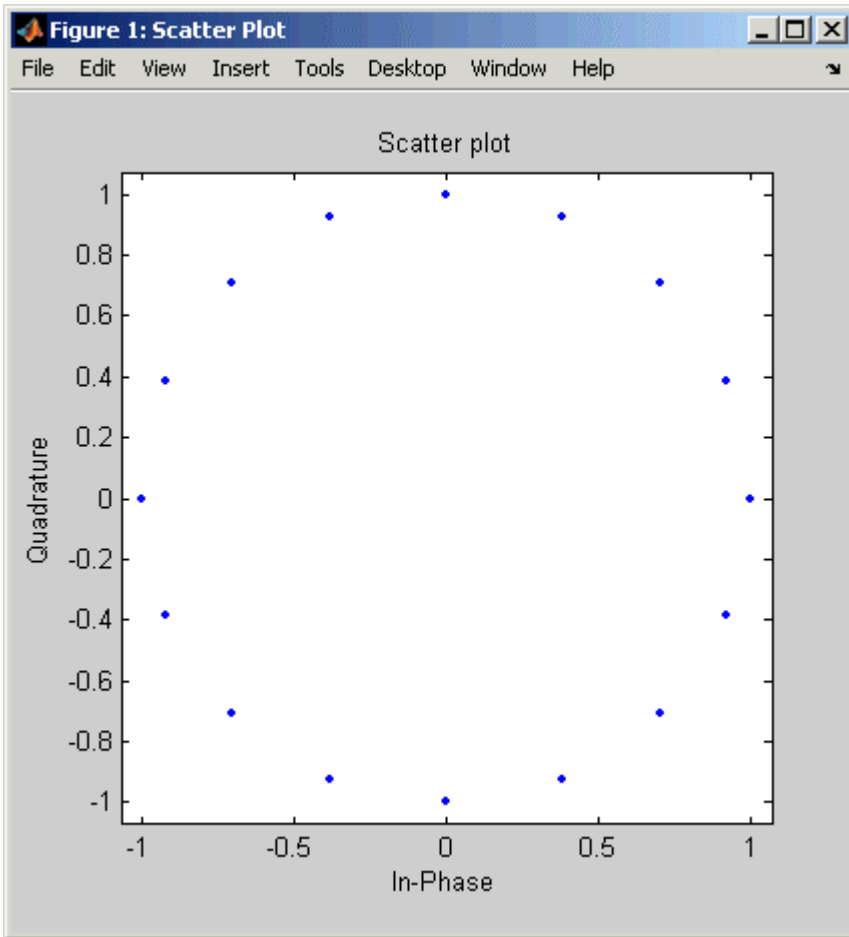
To plot the signal constellation associated with a modulation process, follow these steps:

1. If the alphabet size for the modulation process is M , then create the signal $[0:M-1]$. This signal represents all possible inputs to the modulator.
2. Use the appropriate modulation function to modulate this signal. If desired, scale the output. The result is the set of all points of the signal constellation.
3. Apply the *scatterplot* function to the modulated output to create a plot.

Constellation for 16-PSK

The code below plots a PSK constellation having 16 points. Figure

```
M = 16;
x = [0:M-1];
Fd=1;Fs=1;
scatterplot(dmodce(x, 1, 1, 'qask', M));
```



3. Exercises

NOTE: Help can be obtained from the MATLAB command window by typing the following at the command prompt: help /comm

- (a) For section 2 above, modify the parameters F_d , F_s , E_b/N_0 as well as the number of bits transmitted in the simulation and compute the Bit Error Rates (BER). Comment on all the BER results obtained in relation to the parameters being modified.
- (b) A digital communication system has as input a binary data stream that must be transmitted over an AWGN channel using 16 PSK modulation. Simulate the system using the MATLAB® Communication Toolbox. Your simulation must flow as given in section 2. Compute the BER as well as Symbol Error Rate (SER) for values of E_b/N_0 ranging from -3 to 20 dB.
- (c) As explained in section 2.9, write Matlab programs and display signal constellations for BPSK, QPSK, 8 PSK, Gray-coded 8-QAM as well as 16-QAM and 32-QAM modulation schemes.

4. Selected Bibliography

- [1] Jeruchim, Michel C., Philip Balaban, and K. Sam Shanmugan, *Simulation of Communication Systems*, New York, Plenum Press, 1992.
- [2] Proakis, John G., *Digital Communications*, 3rd ed., New York, McGraw-Hill, 1995.
- [3] Sklar, Bernard, *Digital Communications: Fundamentals and Applications*, Englewood Cliffs, N.J., Prentice-Hall, 1988.
- [4] The MathWorks, Inc.; www.mathworks.com