EXERCISE 1: Random Signal

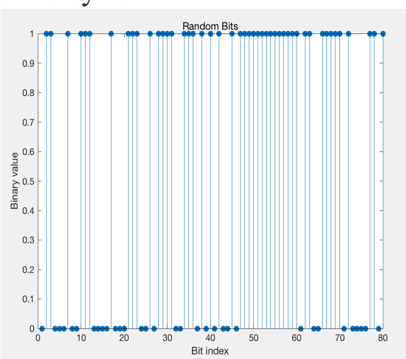
Milestone: Demonstrate your code and the output to a demonstrator.

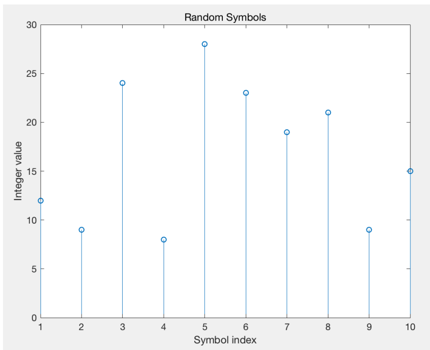
Objectives: Please write at least two points mentioning the purpose of this lab exercise.

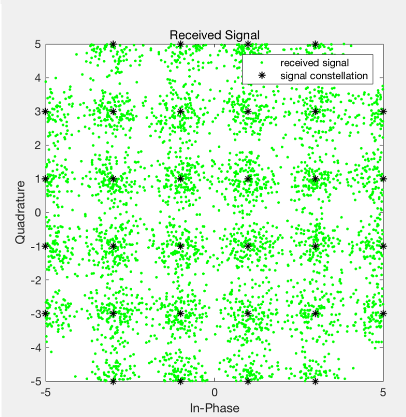
1. The bit error rate of the system is calculated and the transmitted and received signals are displayed in the scatter diagram. A communication system consisting of baseband modulator, channel and demodulator is used to process binary data flow.
2. Be familiar with the relevant communication toolbox functions. In this exercise, the baseband 32-qam is selected as the modulation scheme and the addition of Gaussian white noise is used as a channel model.

Procedure: Please briefly describe the exercise procedure. Also provide schematic diagram, output, figure and/or table, if any.

1. Generates a random binary stream of data. Use the randInt function to create a column vector that lists the consecutive values in the binary data stream.
2. Use 32 qam modulation. The modem.qammod object implements an M-ary QAM modulator. In this exercise, M is 32 (changed).
3. Add Gaussian white noise. When the AWGN function is applied to modulated signals, Gaussian white noise increases. Bit energy noise power spectral density ratio EB/N0 is arbitrarily set to 10dB (change).
4. Create a scatter plot. When you apply the scattering function to the signal sending and receiving, you can see what the signal constellation looks like and how the noise distorts the signal.
5. Use 32 qam demodulation. When the modem.qamdemod object's demodulation method is applied to the receiving signal, it is demodulated.
6. Calculate the system BER.








```
number_of_errors =
   440
bit_error_rate =
   0.0146666666666667
```

Code: Please write your code here. Do not forget to include comments.

```
M=32; % size of signal constellation--change1
k=log2(M); %number of bits per symbol
n=3e4; %number of bits to process
nsamp=1; %oversampling rate

x=randi([0 1],[n 1]); %random binary data stream
figure
stem(x(1:80),'filled');%--change2
title('Random Bits');
xlabel('Bit index');ylabel('Binary value');

% bit-to-symbol mapping
xsym=bi2de(reshape(x,k,length(x)/k).','left-msb');
figure
stem(xsym(1:10));
title('Random Symbols');
xlabel('Symbol index');ylabel('Integer value');

% modulation
y=modulate(modem.qammod(M),xsym);
% transmitted signal
ytx=y;
EbNo=10; % in dB %--chang3
snr=EbNo+10*log10(k)-10*log10(nsamp);
ynoisy=awgn(ytx,snr,'measured');
yrx=ynoisy;

% scatter polt
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]);
xlabel('In-Phase');ylabel('Quadrature');

%demodulation
zsym=demodulate(modem.qamdemod(M),yrx);

% symbol-to-bit mapping
z=de2bi(zsym,'left-msb');
z=reshape(z.',numel(z),1);

% BER computation
[number_of_errors, bit_error_rate]=biterr(x,z)
```

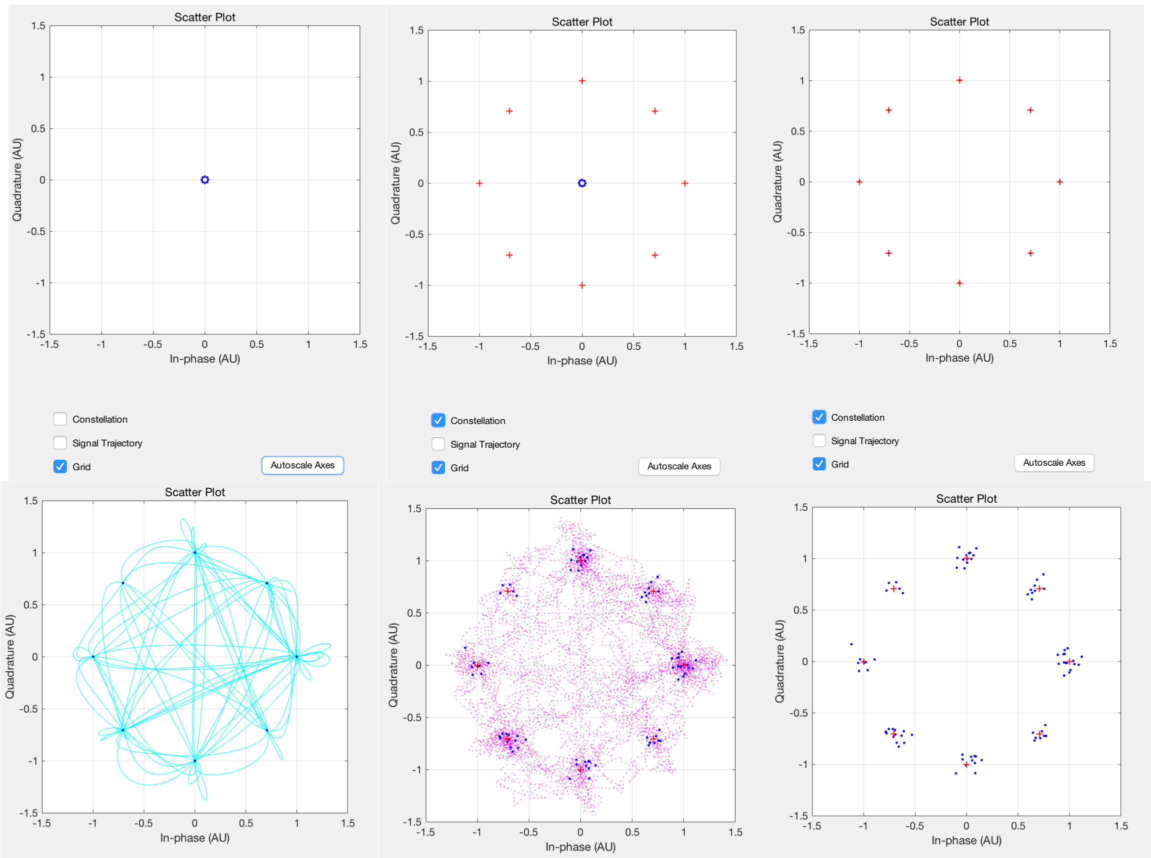
Learning Outcomes: Please mention at least two things that you have learnt while performing this exercise.

1. A binary data stream in a communication system consists of a baseband modulator, channel, and demodulator.
2. Familiar with MATLAB functions, like randint—generate a random binary data stream; awgn—add white Gaussian noise; and modulate and demodulate with modem.qamdemod object.

EXERCISE 2: Scatter plots

Milestone: Demonstrate your code and the output to a demonstrator.

Objectives: Please write at least two points mentioning the purpose of this lab exercise.
1. Create a scatter plot from the signal and use the scatterplot object. Scatter plots are often used to visualize the signal constellation associated with digital modulation. 2. Observe the received signal of the QPSK modulation system. The output symbol is pulse shaped using a cosine roll-off filter.
Procedure: Please briefly describe the exercise procedure. Also provide schematic diagram, output, figure and/or table, if any.
1. Create a QPSK modulator object and upsampling filter using 16 upsampling rates and transmitted signals. 2. Create a scatterplot and set the sample for each symbol to signal ascent sampling rate. 3. Set the constellation values in the scatter plot as expected. 4. Refilter the signal using a normalized filter. The reset range also resets the measurement delay counter and discards the temporary filter value. 5. Create a noisy signal by passing xmt to the AWGN channel. Send the received signal to the scatter plot. 6. Print scatterplots



Code: Please write your code here. Do not forget to include comments.

```
%%ex2
hMod=modem.pskmod('M',8,'phaseOffset',pi/4);%--change1

Rup=32;%--change2 %upsampling rate
hFilDesign=fdesign.pulseshaping(Rup,'Raised Cosine','Nsym,Beta',Rup,0.5);
hFil=design(hFilDesign);

d=randi([0 hMod.M-1],100,1);
sym=modulate(hMod,d);
xmt=filter(hFil,upsample(sym,Rup));

hScope=commscope.ScatterPlot;
hScope.SamplesPerSymbol=Rup;

hScope.Constellation=hMod.Constellation;

groupDelay=(hFilDesign.NumberOfSymbols/2);
hScope.MeasurementDelay=groupDelay/hScope.SymbolRate;

update(hScope,xmt)
hScope.PlotSettings.Constellation='on';

hFil.Numerator=hFil.Numerator/max(hFil.Numerator);

xmt=filter(hFil,upsample(sym,Rup));

reset(hScope)
update(hScope,xmt)

hScop.PlotSettings.SignalTrajectory='on';
hScope.PlotSettings.SignalTrajectoryStyle = ':m';

autoscale(hScope)

rcv = awgn(xmt, 20, 'measured'); % Add AWGN

reset(hScope)
update(hScope, rcv)

hScope.PlotSettings.SignalTrajectory = 'off';

hScope.PlotSettings.Constellation = 'on';
```

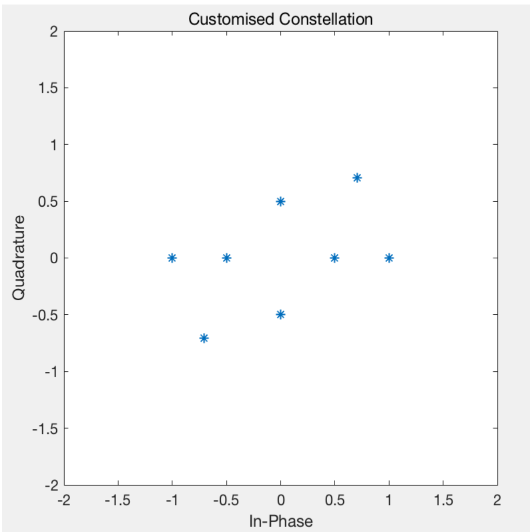
Learning Outcomes: Please mention at least two things that you have learnt while performing this exercise.

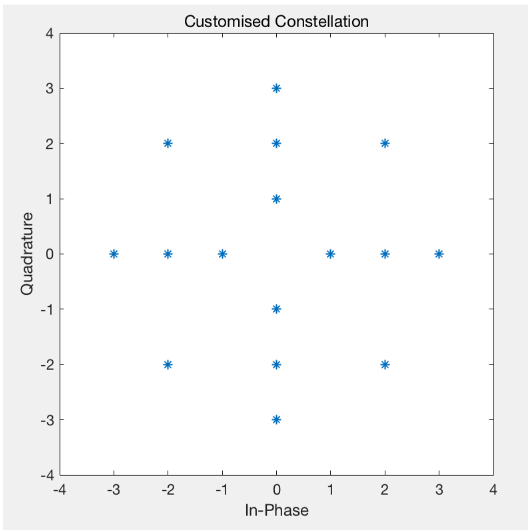
1. A scatter or constellation chart is used to display the constellation of a digitally modulated signal.
2. To generate a scatter plot from a signal, use the scatter plot function or use the comm. constellation system object.
3. This example shows how to use a constellation chart to see the signals sent and received by QPSK, which are pulses formed by a raised cosine filter.

EXERCISE 3: Signal Constellation

Milestone: Demonstrate your code and the output to a demonstrator.

Objectives: Please write at least two points mentioning the purpose of this lab exercise.
1. Plot the signal constellations related to the modulation process, such as PSK, QAM, and customized constellations. 2. Gray coded signal placement. Label the points using binary numbers so that you can visually confirm that the constellation uses gray coding.
Procedure: Please briefly describe the exercise procedure. Also provide schematic diagram, output, figure and/or table, if any.
1. If the modulation process alphabet size is m, the signal [0: M-1] is created. This signal represents all inputs to the modulator. 2. Use the appropriate modulation function to modulate the signal. Scale the output as needed. The result is all points in a set of signal constellations. 3. Apply the scatterplot function to the modulation output to create a plot.





Code: Please write your code here. Do not forget to include comments.

```
%%ex3
M = 4;
x = [0:M-1];
inphase = [1/2 1 0 1/(2^(1/2))];
quadr = [0 0 1/2 1/(2^(1/2))];
inphase = [inphase; -inphase]; inphase = inphase(:);
quadr = [quadr; -quadr]; quadr = quadr(:);
const = inphase + j*quadr;
% Plot constellation.
scatterplot(const, 1, 0, '*');
hold on;
% circle=rectangle('Position',[-1,-1,2,2],'Curvature',[1,1]);
axis([-2 2 -2 2]);
title('Customised Constellation');
hold off;

M = 8;
x = [0:M-1];
inphase = [1 2 3 0 -2 0 2 0];
quadr = [0 0 0 1 2 2 2 3];
inphase = [inphase; -inphase]; inphase = inphase(:);
quadr = [quadr; -quadr]; quadr = quadr(:);
const = inphase + j*quadr;
% Plot constellation.
scatterplot(const, 1, 0, '*');
hold on;
% circle=rectangle('Position',[-1,-1,2,2],'Curvature',[1,1]);
axis([-4 4 -4 4]);
title('Customised Constellation');
hold off;
```

Learning Outcomes: Please mention at least two things that you have learnt while performing this exercise.

1. Process to draw the constellation for 16-PSK, 32-QAM, and also the customized QAM signals
2. Use the Gray-coded constellation to label the points using binary numbers.

EXERCISE 4: Using BERTool

Milestone: Demonstrate your code and the output to a demonstrator.

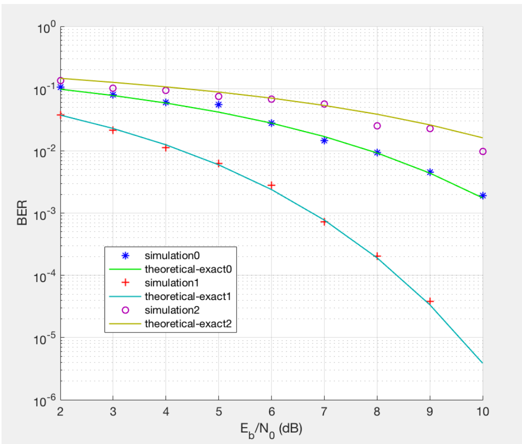
Objectives: Please write at least two points mentioning the purpose of this lab exercise.

1. Create advanced simulations with several techniques. Loop through a set of values for specific parameters such as Eb / N0, alphabet size, oversampling rate.

2. Run the simulation using BERTool. Plot BER as a function of EbNo using a logarithmic scale on the vertical axis. Then compare with the theoretical value.

Procedure: Please briefly describe the exercise procedure. Also provide schematic diagram, output, figure and/or table, if any.

1. Save the template in your own directory. Save the bertool template under the file name my commdoc bertool.
2. Copy the setup code to the template. Copy the simulation code to the template.
3. Update numbits and toterr. Insert the required code after the code pasted in the last step and before the end statement of the template.
4. Suppress previous drawings. Running multiple iterations will result in multiple graphs, which are not shown in this exercise for the sake of simplicity.
5. Omit direct allocation of ebno. When bertool calls an analog function, it specifies the value of ebno. The my commdoc bertool function cannot assign ebno directly. Therefore, delete or comment out the lines that you pasted into my commdoc bertool (within the channel section) directly assigned to ebno. %Ebno = 10;% in DB% is bertool comment out
6. Use bertool for simulation and drawing. Click the run button on bertool.
7. Compared with the theoretical results. To check that the results of the above exercise are correct, use bertool again. This time, its theoretical panel is used to draw theoretical BER results in the same window as the previous simulation results.



Code: Please write your code here. Do not forget to include comments.
(code below)

```
function [ber, numBits] = my_commdoc_bertool(EbNo, maxNumErrs, maxNumBits)
%BERTOOLTEMPLATE Template for a BERTool simulation function.
%   This file is a template for a BERTool-compatible
%   simulation function.  To use the template, insert your
%   own code in the places marked "INSERT YOUR CODE HERE"
%   and save the result as a file on your MATLAB path.
%   Then use the Monte Carlo panel of BERTool to execute
%   the script.
%
%   For more information about this template and an example
%   that uses it, see the Communications System Toolbox documentation.
%
%   See also BERTOOL, VITERBISIM.

% Copyright 1996-2010 The MathWorks, Inc.

% Import Java class for BERTool.
import com.mathworks.toolbox.comm.BERTool;

% Initialize variables related to exit criteria.
totErr  = 0; % Number of errors observed
numBits = 0; % Number of bits processed
%% Setup
% Define parameters.
M = 32;                        % Size of signal constellation
k = log2(M);                  % Number of bits per symbol
n = 1000;                     % Number of bits to process
nSamp = 1;

% Simulate until number of errors exceeds maxNumErrs
% or number of bits processed exceeds maxNumBits.
while((totErr < maxNumErrs) && (numBits < maxNumBits))

   % Check if the user clicked the Stop button of BERTool.
   if (BERTool.getSimulationStop)
      break;
   end

%% Create Modulator and Demodulator
hMod = modem.qammod(M);          % Create a 16-QAM modulator
hMod.InputType = 'Bit';          % Accept bits as inputs
hMod.SymbolOrder = 'Gray';       % Accept bits as inputs
hDemod = modem.qamdemod(hMod);   % Create a 16-QAM based on the modulator

%% Signal Source
% Create a binary data stream as a column vector.
x = randi([0 1],n,1); % Random binary data stream

%% Modulation
% Modulate using 16-QAM.
y = modulate(hMod,x);

%% Transmitted Signal
yTx = y;

%% Channel
% Send signal over an AWGN channel.
% EbNo = 15; % In dB
SNR = EbNo + 10*log10(k) - 10*log10(nSamp);
yNoisy = awgn(yTx,SNR,'measured');

%% Received Signal
yRx = yNoisy;

%% Demodulation
% Demodulate signal using 16-QAM.
z = demodulate(hDemod,yRx);

%% 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);

%% Update totErr and numBits.
totErr = totErr + number_of_errors;
numBits = numBits + n;

end % End of loop

% Compute the BER.
ber = totErr/numBits;
```

Learning Outcomes: Please mention at least two things that you have learnt while performing this exercise.

1. Bertool was used to simulate the signals of different ebno, and compared with the theoretical values to observe the differences.

2. Processing data in smaller sets rather than in a larger set reduces memory requirements; dynamically determining how much data to process for reliable results, rather than trying to guess at the beginning.

EXERCISE 5: Preparing Simulink Models for Use with BERTool

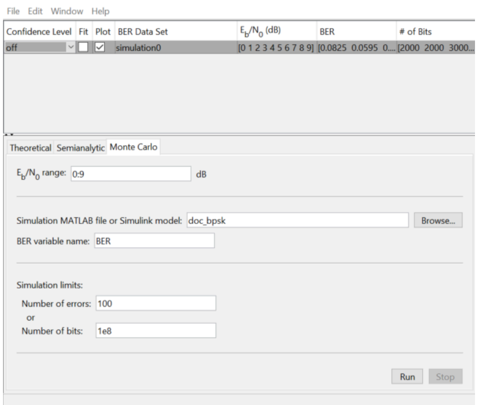
Milestone: Demonstrate your code and the output to a demonstrator.

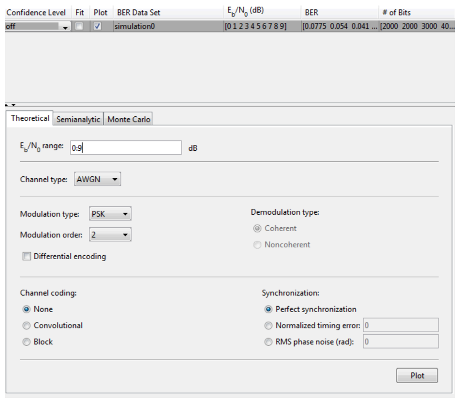
Objectives: Please write at least two points mentioning the purpose of this lab exercise.

1. The BER of the system is calculated and the transmitted and received signals are displayed in the scatter diagram. A communication system consisting of baseband modulator, channel and demodulator is used to process binary data flow.
2. Familiar with related communication toolbox functions. In this exercise, baseband 32- QAM is selected as the modulation method, and AGWN is selected as the channel model.

Procedure: Please briefly describe the exercise procedure. Also provide schematic diagram, output, figure and/or table, if any.

1. Enter bpskdoc command in matlab command window to open the model.
2. To initialize parameters in the MATLAB workspace, enter the required commands in the matlab command window
3. Click the block to open the AWGN channel block dialog box. In this particular model, ES / N0 is equivalent to EB / N0 because the modulation type is BPSK.
4. Add the error rate calculation block dialog box. Set the target error number to maxnomels, the maximum number of symbols to Max nomit.
5. Insert the signal into the work block into the model and connect it to the output of the error rate calculation block.
6. Open bertool and go to the Monte Carlo tab. Set the parameters on the Monte Carlo tab, as shown in the following figure.





7. To configure the new signal added to the work block, open its dialog box. Set variable name to BER, limit data points to last 1.
8. To compare these simulation results with theoretical results, go to the theory tab in bertool and set the parameters, as shown in the figure above.

Code: Please write your code here. Do not forget to include comments.

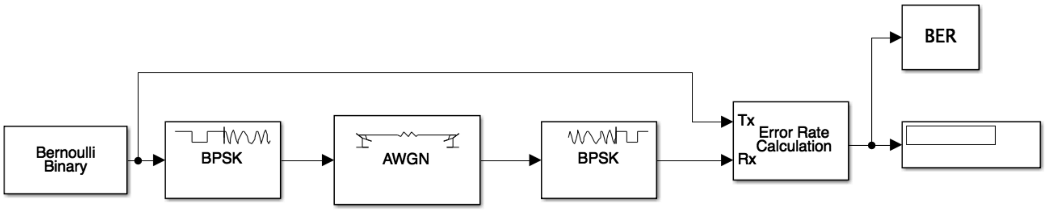


Figure 5.1 BPSK modulation diagram

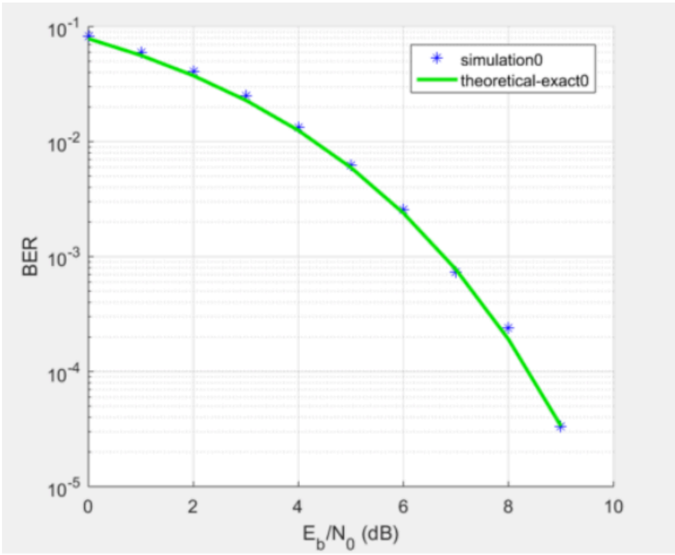


Figure 5.2 Simulation result and theoretical result

Learning Outcomes: Please mention at least two things that you have learnt while performing this exercise.

1. Bertol Command to start the BER tool application. Use this application to analyze BER communication system performance. Use the matlab function and Monte Carlo simulation of the Simulink model, or closed theoretical expression of the selected type communication system to analyze performance.

2.Simulation included in the MATLAB simulation function or Simulink model. After creating a function or model of the simulation system, the bertool iterates on the selected EB / N0 values and collects the results.

3. Draw one or more BER data sets on a single set of axes. For example, you can graphically compare simulation data from a number of similar models in a communication system with theoretical results or simulation data.

EXERCISE 6: Eye Diagram

Milestone: Demonstrate your code and the output to a demonstrator.

Objectives: Please write at least two points mentioning the purpose of this lab exercise.

In this exercise, we use eye scope to introduce the graph of the eye scope measurement, the comparison and the result of the print plot.

Procedure: Please briefly describe the exercise procedure. Also provide schematic diagram, output, figure and/or table, if any.

1. In the matlab command line, load nine eye map objects into the MATLAB workspace.

2. Type the eyescope on the matlab command line to start the eyescope tool.

3. Select eyeobj1 and click Import. Eyescope imports objects, displays images in object diagrams, and lists file names in the eye diagram objects list.

4. In the eyescope window, click the import button. Click to select eyeobj5, and then click the import button. Display a new drawing and add Eyeobj5 to the eye diagram objects list.

5. Change or delete measurements on the vision display

6. From the left side of the shuttle control, select crossing time and crossing amplitude, and then click Add to display the glasses with these new settings, click OK. The measuring area of the refractor shows the crossing time and amplitude at the bottom of the measuring section.

7. Change the order of the list so that the crossing time and crossing amplitude are displayed at the top of the list.

8. Import Eyeobj2, Eyeobj3, and Eyeobj4. Eyescope now contains eye diagram objects 1, 5, 2, 3, and 4 in the list.

9. To compare measurements from multiple eye chart objects, compare measurements view.

10. To include any data from a previously selected measurement selection during this process, go to the measurement selector and select the total jitter check box. The object plot updates to show additional measurements.

11. To print the plot display, plot to drawing. In the figure window, the image print.

Code: Please write your code here. Do not forget to include comments.

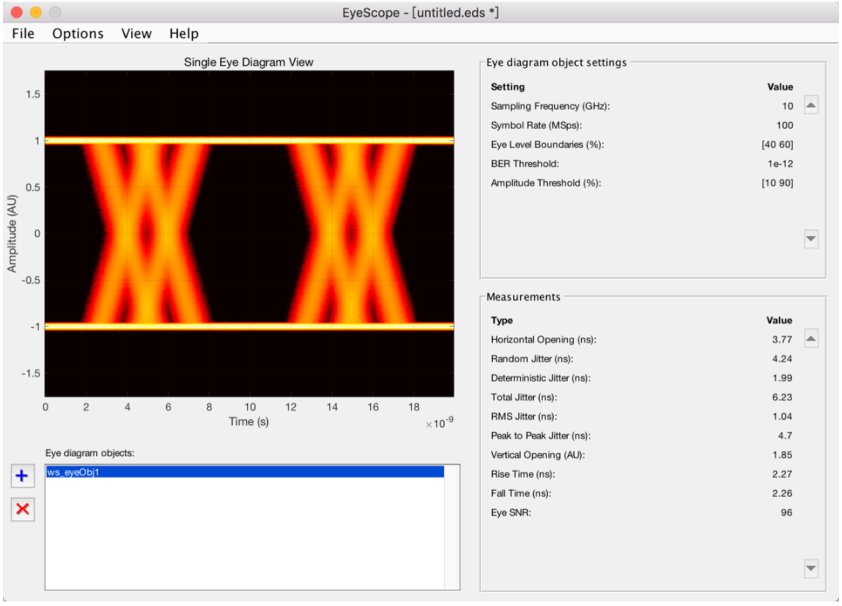


Figure 6.1 Eye diagram object settings and Measurements selections.

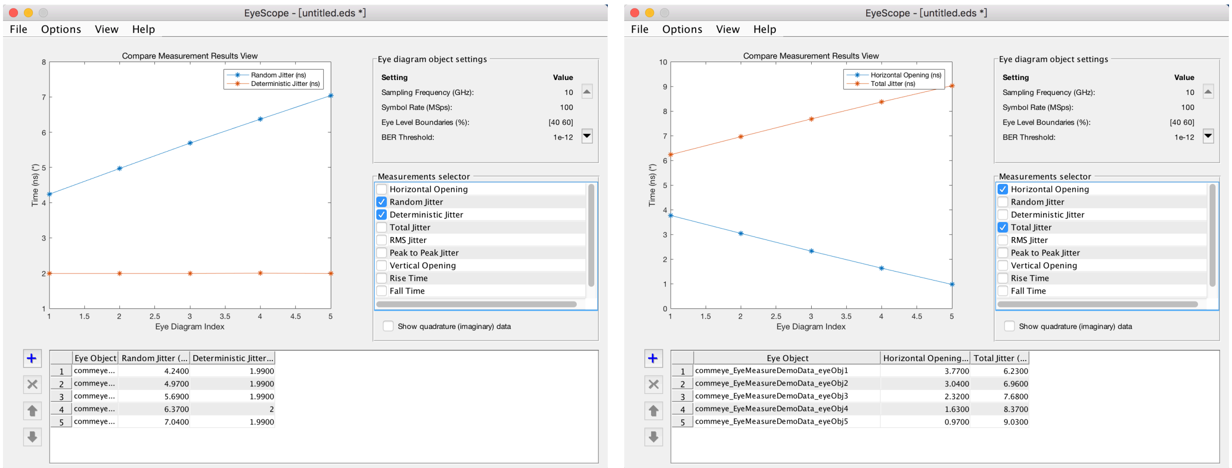


Figure 6.2 Compare Measurement Results View.

Learning Outcomes: Please mention at least two things that you have learnt while performing this exercise.

1. In digital communication, digital provides a visual representation of how noise affects system performance.

2. Use the eyescope tool to verify the data contained in the object. Eyescope displays visual chart and measurement results in a unified graphic environment. You can import multiple pad objects and compare them with the imported results.

EXERCISE 7: Simulating a Simple Mode

<u>Milestone: Demonstrate your code and the output to a demonstrator.</u>

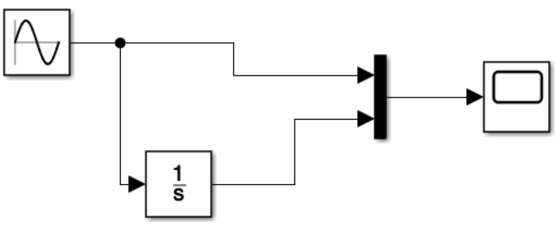
Objectives: Please write at least two points mentioning the purpose of this lab exercise.

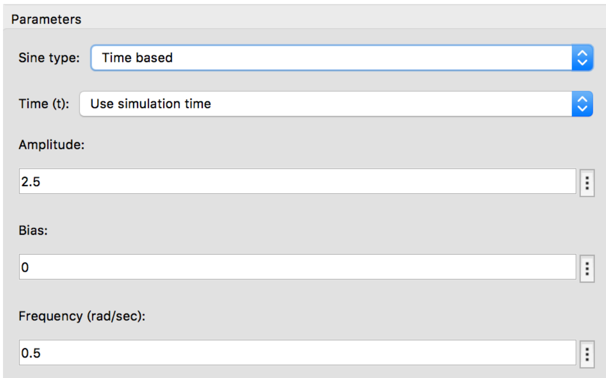
1. A simple model is built with Simulink software and simulated.

2. The model described in this exercise integrates sine waves and displays the results with the original waves.

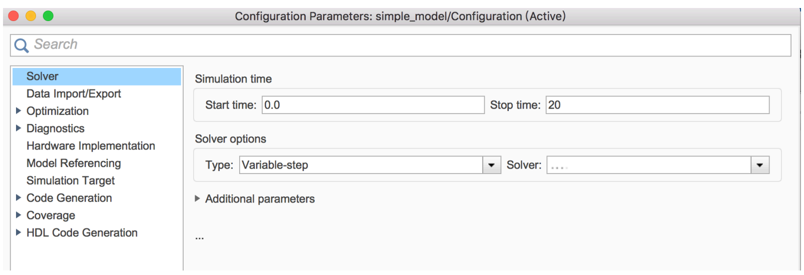
Procedure: Please briefly describe the exercise procedure. Also provide schematic diagram, output, figure and/or table, if any.

1. complete the model, and save the model with the file name simple_model.mdl. For the sine resource, we set the parameters as follows:





2. Set simulation options the start and stop time, and the type of solver that Simulink software uses to solve the model at each time step.



3. Run the simulation, observe by Scope block in the model window. The Scope window displays the simulation results.

Code: Please write your code here. Do not forget to include comments.

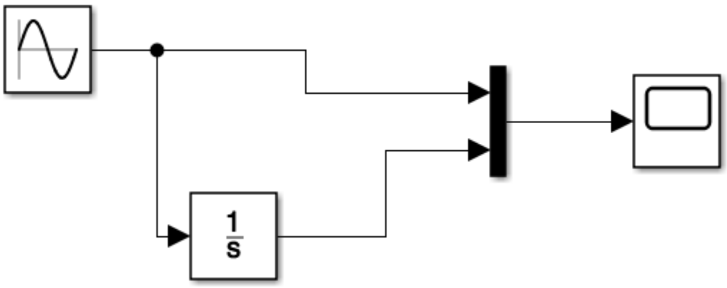


Figure 7.1 The system diagram

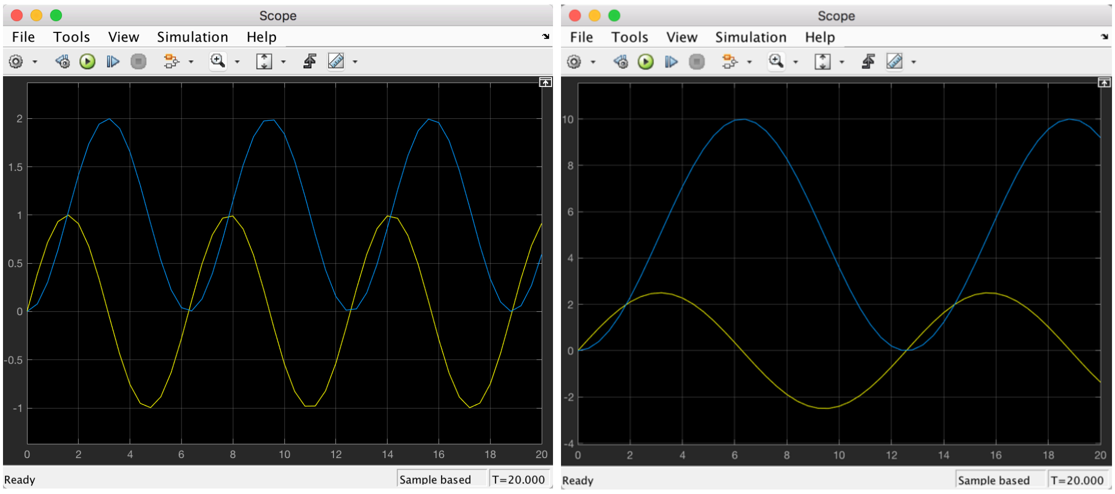


Figure 7.2 The simulation result with different input signals

Learning Outcomes: Please mention at least two things that you have learnt while performing this exercise.

1. Learn and describe how to build a simple model using Simulink software and how to simulate the model.
2. We can change the input signal to get different results
3. Simulation can add some requirements, such as start and stop time, sampling frequency and so on.

EXERCISE 8: Understanding Simulink blocks

Milestone: Demonstrate your code and the output to a demonstrator.

Objectives: Please write at least two points mentioning the purpose of this lab exercise.

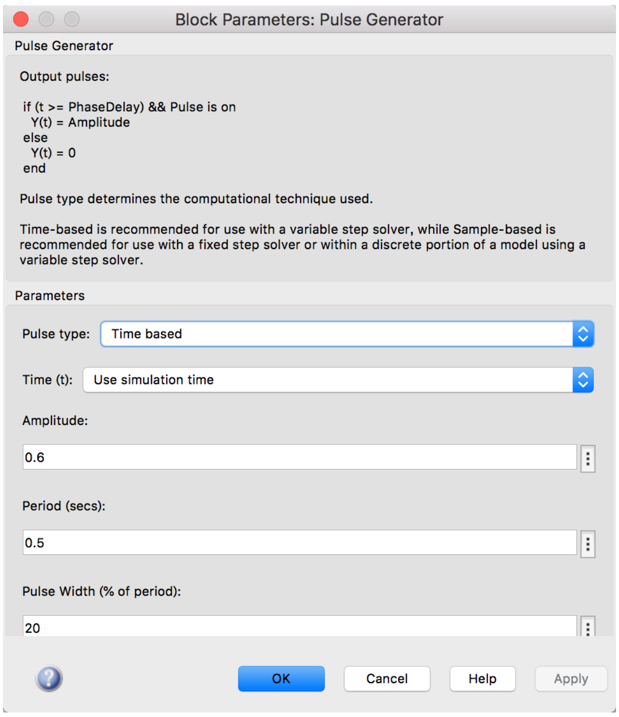
1. understand Simulink blocks, such as signal generator, digital clock, step, uniform random number, pulse generator, and sine wave function.
2. set the desired parameters to each block.
3. set the desired parameters to simulation.

Procedure: Please briefly describe the exercise procedure. Also provide schematic diagram, output, figure and/or table, if any.

1. Study following Simulink blocks using scope: signal generator, digital clock, step, uniform random number, pulse generator, and sine wave function.

2. start the simulation and observe the result.

3. design a multiplier with a square-wave input. Select the Simulation menu and then run the simulation by clicking Start. Save the model file as a Matlab mdl-file.



Code: Please write your code here. Do not forget to include comments.

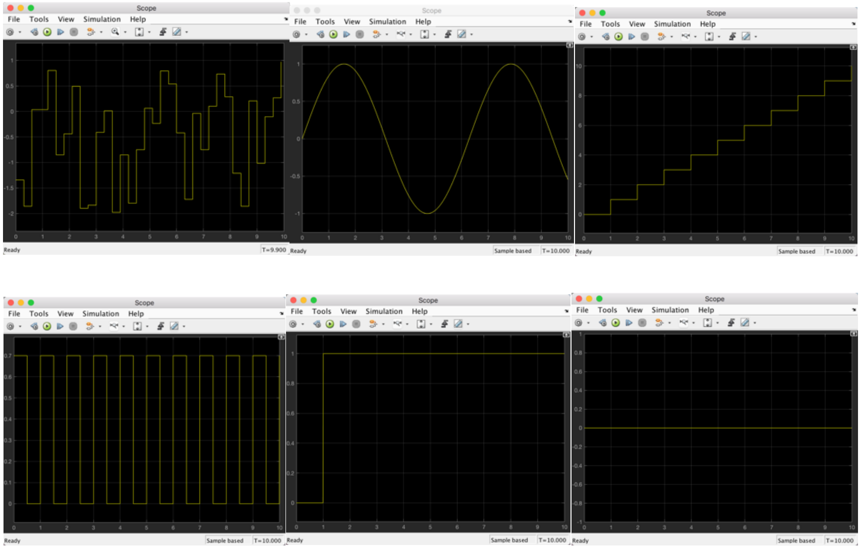
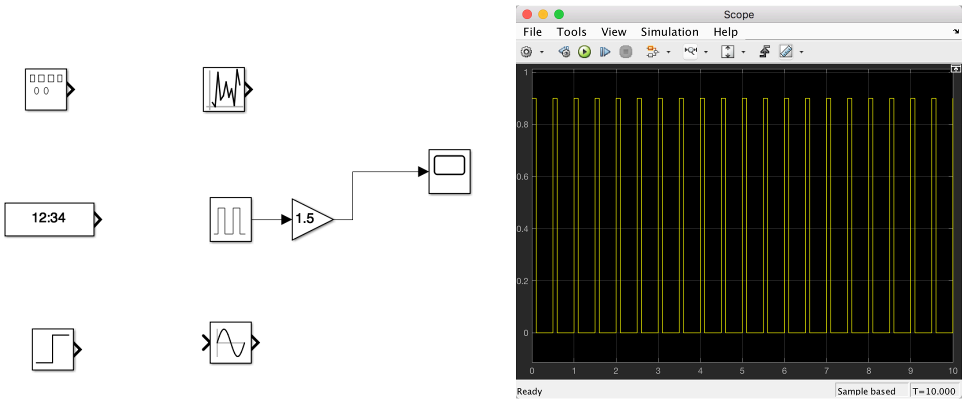


Figure 8.1 The simulation result with different input signals



(a) (b)

Figure 8.2 (a) The system diagram (b) The simulation result of a multiplier with a square-wave input

Learning Outcomes: Please mention at least two things that you have learnt while performing this exercise.

1. Understand the following Simulink blocks and their graphics: signal generator, digital clock, step, uniform random number, pulse generator and sine wave function.

2. Set different parameters to achieve the desired results. And use the simulation module to design the system we want.

EXERCISE 10: Realising Power Spectrum

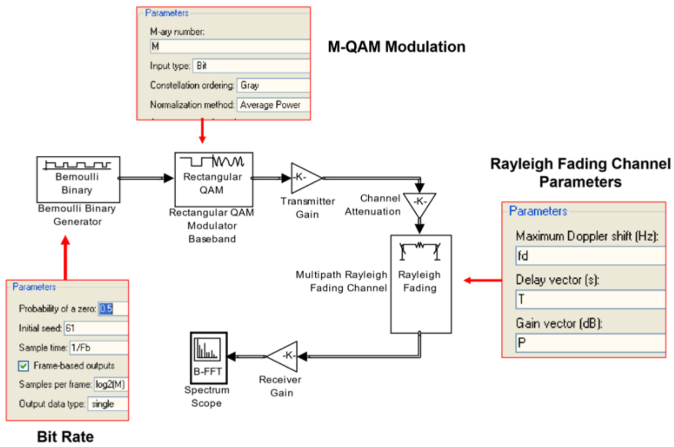
Milestone: Demonstrate your code and the output to a demonstrator.

Objectives: Please write at least two points mentioning the purpose of this lab exercise.

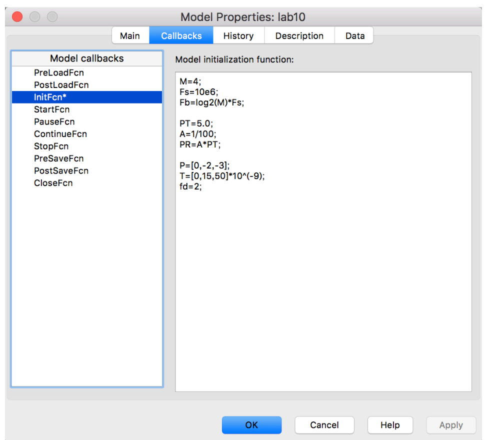
1. Draw the power spectrum of the signal with the block “Spectrum Scope”.
2. Understand what is Rayleigh fading channel and the rectangular QAM modulator.
3. Compare differences between various Doppler frequencies.

Procedure: Please briefly describe the exercise procedure. Also provide schematic diagram, output, figure and/or table, if any.

1. Design the following System.



2. Use the following channel characterisation.



3. Realise the received power spectrum.
4. change the parameter configurations as shown below and observe the difference in the received power spectrum.

Code: Please write your code here. Do not forget to include comments.

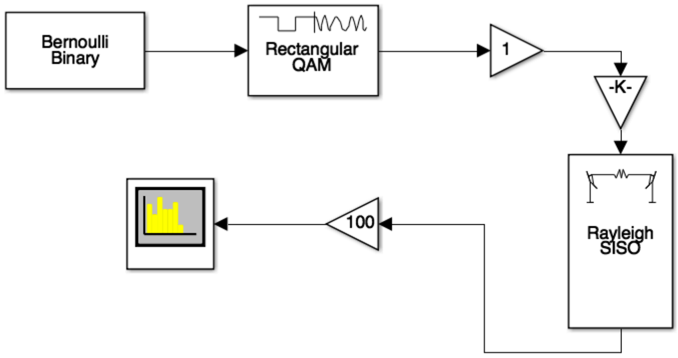
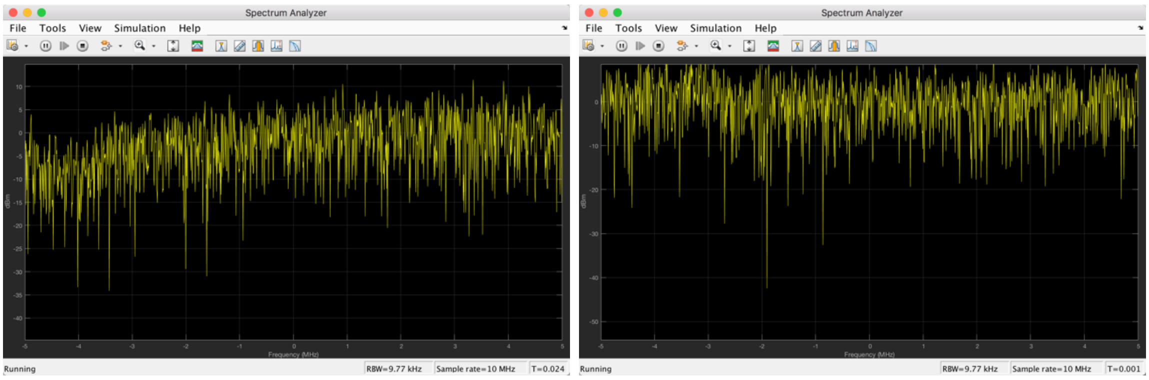


Figure 10.1 The desired system diagram



(a) (b)

Figure 10.2 The power spectrum with Doppler freq. different: (a) 70 Hz (b) 2 Hz

From the figure 10.2, we can see that the power spectrum is more stable when doppler frequency is changing from 70 Hz to 2 Hz.

Learning Outcomes: Please mention at least two things that you have learnt while performing this exercise.

1. Study how to come out the power spectrum of signals. Add the block "Spectrum Scope" to show the result.
2. With doppler frequency decreasing, the spectrum is more stable.
3. Signal processing contains the steps: digital modulation, transmitted and received.
4. In mobile communication, receiving a multipath signal in a receiver. Environmental interference will cause the signal to increase or decrease in phase.