**CT304 – Digital Communication**

**Lab Exercise - 7**

**Name: Sakshi Patadiya**

**ID: 202101469**

**Group: 5**

**Date: 14th November, 2023**

1. **From [1], perform experiments 1 to 6 on Amplitude shift keying (ASK). The page numbers for the same are given in the table of contents. You need to follow the procedures given in the manual and observe the output as suggested in the observations.**

**Experiment 1**

|  |  |  |
| --- | --- | --- |
| Data Pattern | Data Frequency | Resulting Waveforms |
| 8-Bit | 2KHz | CH1: Input Data Clock (TP1) and  CH2: Input Data 8-Bit (TP2). |
| 8-Bit | 2KHz | CH1: Input Data 8-Bit (TP2) and  CH2: 1-bit encoded data (TP3). |

**Experiment 2**

|  |  |  |
| --- | --- | --- |
| Input Data Type | Data Clock Frequency | Resulting Waveforms |
| 8-Bit | 2KHz | CH1: Input Data Clock (TP1),  CH2: Input Data(TP2) |
| 8-Bit | 2KHz | CH1: Input Data (TP3),  CH2: Carrier Signal (TP4) |
| 8-Bit | 2KHz | CH1: Carrier signal (TP4),  CH2: ASK Output (TP5) |

**Experiment 3**

|  |  |  |
| --- | --- | --- |
| Input Data Type | Data Clock Frequency | Resulting Waveforms |
| 8-Bit | 2KHz | CH1: ASK output (TP5),  CH2: multiplier output (TP7) |
| 8-Bit | 8KHz | CH1: ASK output (TP5),  CH2: multiplier output (TP7) |

**Experiment 4**

|  |  |  |
| --- | --- | --- |
| Input Data Type | Data Clock Frequency | Resulting Waveforms |
| 8-Bit | 2KHz | CH1: ASK output (TP5),  CH2: Integrator output (TP8) |
| 8-Bit | 16KHz | CH1: ASK output (TP5),  CH2: Integrator output (TP8) |

**Experiment 5**

|  |  |  |
| --- | --- | --- |
| Input Data Type | Data Clock Frequency | Resulting Waveforms |
| 8-Bit | 2KHz | CH1: Input Data (TP2),  CH2: Comparator out (TP9) |

**Experiment 6**

|  |  |  |
| --- | --- | --- |
| Input Data Type | Data Clock Frequency | Resulting Waveforms |
| 8-Bit | 2KHz | CH1: Input Data (TP2),  CH2: Demodulator out (TP9) |
| 8-Bit | 2KHz | CH1: Input Data (TP2),  CH2: Demodulator out (TP9) |

1. **From [2], go through “Signal correlator” from section 5.2, solve illustrative problems 5.1 and 5.2. Then, reproduce the MATLAB script for illustrative problem 5.2. Thereafter, from the problems at the end of the chapter, solve problem 5.3.**

Ans:

**Q. 5.2**

**MATLAB CODE:**

% MATLAB script for Illustrative Problem 5.2

% Initialization:

K = 20; % Number of samples

A = 1;

% Signal amplitude

l = 0:K;

% Defining signal waveforms:

s\_0 = A \* ones(1, K);

s\_1 = [A \* ones(1, K/2), -A \* ones(1, K/2)];

% Initialize output signals:

r\_0 = zeros(1, K);

r\_1 = zeros(1, K);

% Case 1: noise-N(0, 0)

noise = randn(1, K);

% Sub-case s = s\_0:

s = s\_0;

r = s + noise; % received signal

for n = 1:K

r\_0(n) = sum(r(1:n) .\* s\_0(1:n));

r\_1(n) = sum(r(1:n) .\* s\_1(1:n));

end

% Plotting the results:

subplot(3, 2, 1)

plot(0:K, [0, r\_0], '-', 0:K, [0, r\_1], '--')

set(gca, 'XTickLabel', {'0', '5Tb', '10Tb', '15Tb', '20Tb'})

axis([0, 20, -5, 30])

xlabel('(a) \sigmaA2= 0 & S\_{0} is transmitted', 'fontsize', 10)

% Sub-case s = s\_1:

s = s\_1;

r = s + noise; % received signal

for n = 1:K

r\_0(n) = sum(r(1:n) .\* s\_0(1:n));

r\_1(n) = sum(r(1:n) .\* s\_1(1:n));

end

% Plotting the results:

subplot(3, 2, 2)

plot(0:K, [0, r\_0], '-', 0:K, [0, r\_1], '--')

set(gca, 'XTickLabel', {'0', '5Tb', '10Tb', '15Tb', '20Tb'})

axis([0, 20, -5, 30])

xlabel('(b) \sigmaA2= 0 & S\_{1} is transmitted', 'fontsize', 10)

% Case 2: noise-N(0, 0.1) noise =

0.1 \* randn(1, K);

% Sub-case s = s\_0:

s = s\_0;

r = s + noise; % received signal

for n = 1:K

r\_0(n) = sum(r(1:n) .\* s\_0(1:n));

r\_1(n) = sum(r(1:n) .\* s\_1(1:n));

end

% Plotting the results:

subplot(3, 2, 3)

plot(0:K, [0, r\_0], '-', 0:K, [0, r\_1], '--')

set(gca, 'XTickLabel', {'0', '5Tb', '10Tb', '15Tb', '20Tb'})

axis([0, 20, -5, 30])

xlabel('(c) \sigmaA2= 0.1 & S\_{0} is transmitted', 'fontsize', 10)

% Sub-case s = s\_1:

s = s\_1;

r = s + noise; % received signal

for n = 1:K

r\_0(n) = sum(r(1:n) .\* s\_0(1:n));

r\_1(n) = sum(r(1:n) .\* s\_1(1:n));

end

% Plotting the results:

subplot(3, 2, 4)

plot(0:K, [0, r\_0], '-', 0:K, [0, r\_1], '--')

set(gca, 'XTickLabel', {'0', '5Tb', '10Tb', '15Tb', '20Tb'})

axis([0, 20, -5, 30])

xlabel('(d) \sigmaA2= 0.1 & S\_{1} is transmitted', 'fontsize', 10)

% Case 3: noise-N(0, 1)noise = randn(1, K);

% Sub-case s = s\_0:

s = s\_0;

r = s + noise; % received signal

for n = 1:K

r\_0(n) = sum(r(1:n) .\* s\_0(1:n));

r\_1(n) = sum(r(1:n) .\* s\_1(1:n));

end

% Plotting the results:

subplot(3, 2, 5)

plot(0:K, [0, r\_0], '-', 0:K, [0, r\_1], '--')

set(gca, 'XTickLabel', {'0', '5Tb', '10Tb', '15Tb', '20Tb'})

axis([0, 20, -5, 30])

xlabel('(e) \sigmaA2= 1 & S\_{0} is transmitted', 'fontsize', 10)

% Sub-case s = s\_1:

s = s\_1;

r = s + noise; % received signal

for n = 1:K

r\_0(n) = sum(r(1:n) .\* s\_0(1:n));

r\_1(n) = sum(r(1:n) .\* s\_1(1:n));

end

% Plotting the results:

subplot(3, 2, 6)

plot(0:K, [0, r\_0], '-', 0:K, [0, r\_1], '--')

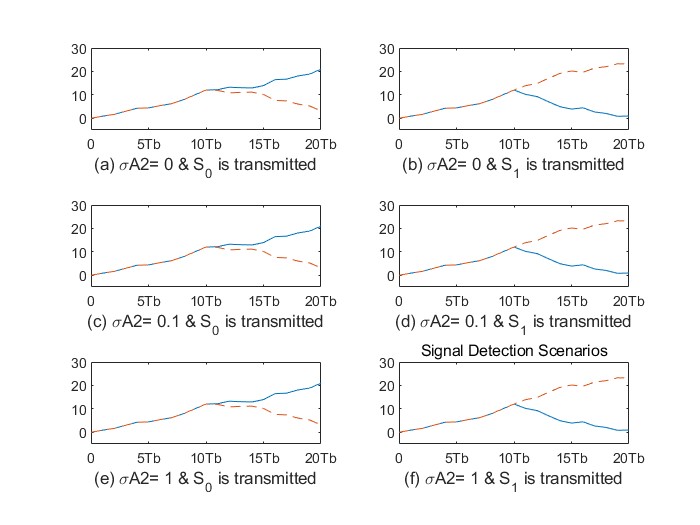
set(gca, 'XTickLabel', {'0', '5Tb', '10Tb', '15Tb', '20Tb'})

axis([0, 20, -5, 30])

xlabel('(f) \sigmaA2= 1 & S\_{1} is transmitted', 'fontsize', 10)

% Display the plots

subtitle('Signal Detection Scenarios');



**Q.5.3**

**MATLAB CODE:**

% MATLAB script for Illustrative Problem 5.3

% Initialization:

K = 20; % Number of samples

A = 1;

% Signal amplitude

l = 0:K;

% Defining signal waveforms:

s\_0 = A \* ones(1, K);

s\_1 = [A \* ones(1, K/2), -A \* ones(1, K/2)];

% Initialize output signals:

r\_0 = zeros(1, K);r\_1 = zeros(1, K);

% Noise variances:

noise\_variances = [0.1, 1, 3];

for var\_idx = 1:length(noise\_variances)

% Generate noise with different variances

noise = sqrt(noise\_variances(var\_idx)) \* randn(1, K);

% Sub-case s = s\_0:

s = s\_0;

r = s + noise; % received signal

for n = 1:K

r\_0(n) = sum(r(1:n) .\* s\_0(1:n));

r\_1(n) = sum(r(1:n) .\* s\_1(1:n));

end

% Plotting the results:

subplot(2, 3, var\_idx)

plot(0:K, [0, r\_0], '-', 0:K, [0, r\_1], '--')

set(gca, 'XTickLabel', {'0', '5Tb', '10Tb', '15Tb', '20Tb'})

axis([0, 20, -5, 30])

title(['\sigma^2 = ' num2str(noise\_variances(var\_idx))], 'fontsize', 10)

if var\_idx == 1

ylabel('Output', 'fontsize', 10)

end

% Sub-case s = s\_1:

s = s\_1;

r = s + noise; % received signal

for n = 1:K

r\_0(n) = sum(r(1:n) .\* s\_0(1:n));

r\_1(n) = sum(r(1:n) .\* s\_1(1:n));

end

% Plotting the results:

subplot(2, 3, var\_idx + 3)

plot(0:K, [0, r\_0], '-', 0:K, [0, r\_1], '--')

set(gca, 'XTickLabel', {'0', '5Tb', '10Tb', '15Tb', '20Tb'})

axis([0, 20, -5, 30])

if var\_idx == 1

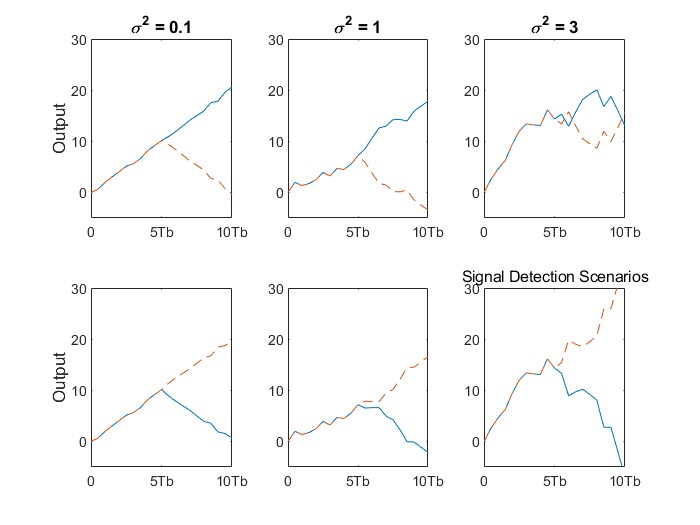
ylabel('Output', 'fontsize', 10)

end

end

% Display the plots

subtitle('Signal Detection Scenarios');



1. **From [2], go through “Matched filter” from sections 5.2, solve illustrative problems 5.3 and 5.4. Then, reproduce the MATLAB script for illustrative problem 5.4. Thereafter, from the problems at the end of the chapter, solve problem 5.4.**

Ans:

**Illustrative Problem 5.4**

**MATLAB CODE:**

% MATLAB script for Illustrative Problem 5.4.

% Initialization:

K = 20; % Number of samples

A = 1;

% Signal amplitude

l = 0:K;

% Defining signal waveforms:

s\_0 = A \* ones(1, K);

s\_1 = [A \* ones(1, K/2) -A \* ones(1, K/2)];

% Initializing output signals:

y\_0 = zeros(1, K);

y\_1 = zeros(1, K);

% Case 1: noise - N(0, 0)

noise = random('Normal', 0, 0, 1, K);

% Sub-case s = s\_0:

s = s\_0;

y = s + noise; % received signaly\_0 = conv(y, fliplr(s\_0));

y\_1 = conv(y, fliplr(s\_1));

% Plotting the results:

subplot(3, 2, 1);

plot(l, [0 y\_0(1:K)], '-k', l, [0 y\_1(1:K)], '--k');

set(gca, 'XTickLabel', {'0', '5Tb', '10Tb', '15Tb', '20Tb'});

axis([0 20 -30 30]);

xlabel('(a) \sigma^2 = 0 & S\_0 is transmitted', 'fontsize', 10);

% Sub-case s = s\_1:

s = s\_1;

y = s + noise; % received signaly\_0 = conv(y, fliplr(s\_0));

y\_1 = conv(y, fliplr(s\_1));

% Plotting the results:

subplot(3, 2, 2);

plot(l, [0 y\_0(1:K)], '-k', l, [0 y\_1(1:K)], '--k');

set(gca, 'XTickLabel', {'0', '5Tb', '10Tb', '15Tb', '20Tb'});

axis([0 20 -30 30]);

xlabel('(b) \sigma^2 = 0 & S\_1 is transmitted', 'fontsize', 10);

% Case 2: noise - N(0, 0.1)

noise = random('Normal', 0, 0.1, 1, K);

% Sub-case s = s\_0:

s = s\_0;

y = s + noise; % received signaly\_0 = conv(y, fliplr(s\_0));

y\_1 = conv(y, fliplr(s\_1));

% Plotting the results:

subplot(3, 2, 3);

plot(l, [0 y\_0(1:K)], '-k', l, [0 y\_1(1:K)], '--k');

set(gca, 'XTickLabel', {'0', '5Tb', '10Tb', '15Tb', '20Tb'});

axis([0 20 -30 30]);

xlabel('(c) \sigma^2 = 0.1 & S\_0 is transmitted', 'fontsize', 10);

% Sub-case s = s\_1:

s = s\_1;

y = s + noise; % received signal

y\_0 = conv(y, fliplr(s\_0));

y\_1 = conv(y, fliplr(s\_1));

% Plotting the results:

subplot(3, 2, 4);

plot(l, [0 y\_0(1:K)], '-k', l, [0 y\_1(1:K)], '--k');

set(gca, 'XTickLabel', {'0', '5Tb', '10Tb', '15Tb', '20Tb'});

axis([0 20 -30 30]);

xlabel('(d) \sigma^2 = 0.1 & S\_1 is transmitted', 'fontsize', 10);

% Case 3: noise - N(0, 1)

noise = random('Normal', 0, 1, 1, K);

% Sub-case s = s\_0:

s = s\_0;

y = s + noise; % received signal

y\_0 = conv(y, fliplr(s\_0));

y\_1 = conv(y, fliplr(s\_1));

% Plotting the results:

subplot(3, 2, 5);

plot(l, [0 y\_0(1:K)], '-k', l, [0 y\_1(1:K)], '--k');

set(gca, 'XTickLabel', {'0', '5Tb', '10Tb', '15Tb', '20Tb'});

axis([0 20 -30 30]);

xlabel('(e) \sigma^2 = 1 & S\_0 is transmitted', 'fontsize', 10);

% Sub-case s = s\_1:

s = s\_1;

y = s + noise; % received signal

y\_0 = conv(y, fliplr(s\_0));

y\_1 = conv(y, fliplr(s\_1));

% Plotting the results:

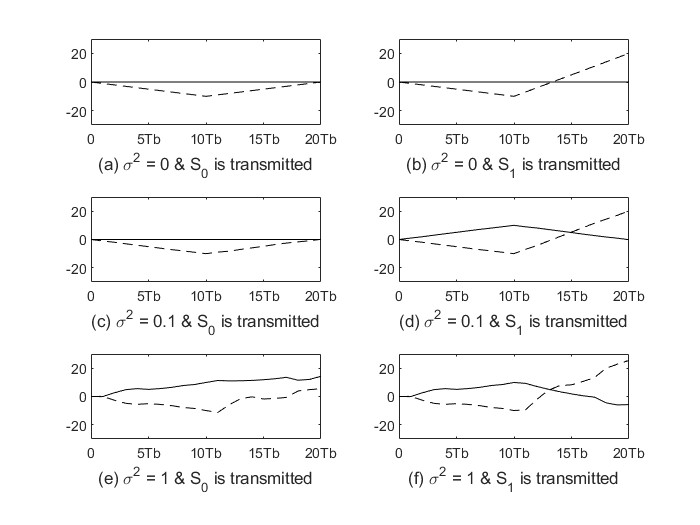
subplot(3, 2, 6);

plot(l, [0 y\_0(1:K)], '-k', l, [0 y\_1(1:K)], '--k');

set(gca, 'XTickLabel', {'0', '5Tb', '10Tb', '15Tb', '20Tb'});

axis([0 20 -30 30]);

xlabel('(f) \sigma^2 = 1 & S\_1 is transmitted', 'fontsize', 10);



**Problem 5.4**

**MATLAB CODE:**

% MATLAB script for Problem 5.4.

% Initialization:

K = 20; % Number of samples

A = 1;

% Signal amplitude

l = 0:K;

% Defining signal waveforms:

s\_0 = A \* ones(1, K);

s\_1 = [A \* ones(1, K/2) -A \* ones(1, K/2)];

% Different noise variances:

variances = [0.1, 1, 3];

for var\_idx = 1:length(variances)

% Initializing output signals:

y\_0 = zeros(1, K);y\_1 = zeros(1, K);

% Noise for the current variance:

variance = variances(var\_idx);

noise = random('Normal', 0, variance, 1, K);

% Sub-case s = s\_0:

s = s\_0;

y = s + noise; % received signal

y\_0 = conv(y, fliplr(s\_0));

y\_1 = conv(y, fliplr(s\_1));

% Plotting the results for the current variance:

subplot(2, length(variances), var\_idx);

plot(l, [0 y\_0(1:K)], '-k', l, [0 y\_1(1:K)], '--k');

set(gca, 'XTickLabel', {'0', '5Tb', '10Tb', '15Tb', '20Tb'});

axis([0 20 -30 30]);

title(['\sigma^2 = ' num2str(variance)], 'fontsize', 10);

end

% Sub-case s = s\_1:

s = s\_1;

for var\_idx = 1:length(variances)

% Initializing output signals:

y\_0 = zeros(1, K);

y\_1 = zeros(1, K);

% Noise for the current variance:

variance = variances(var\_idx);

noise = random('Normal', 0, variance, 1, K);

y = s + noise; % received signal

y\_0 = conv(y, fliplr(s\_0));

y\_1 = conv(y, fliplr(s\_1));

% Plotting the results for the current variance:

subplot(2, length(variances), length(variances) + var\_idx);plot(l, [0 y\_0(1:K)], '-k', l, [0 y\_1(1:K)], '--k');

set(gca, 'XTickLabel', {'0', '5Tb', '10Tb', '15Tb', '20Tb'});

axis([0 20 -30 30]);

title(['\sigma^2 = ' num2str(variance)], 'fontsize', 10);

end

