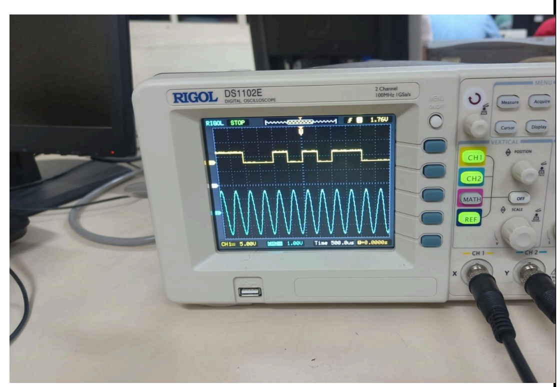
Lab-8

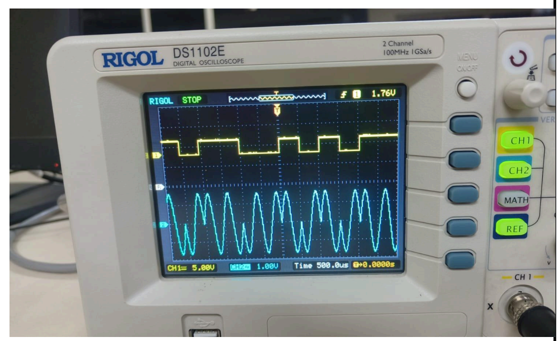
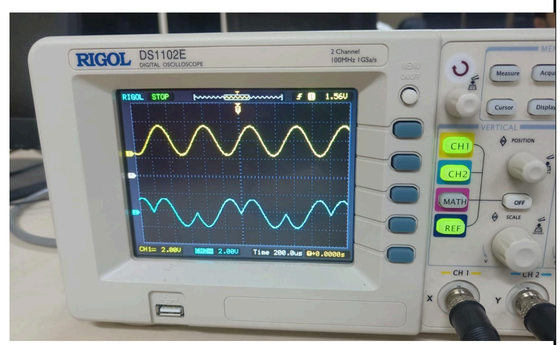
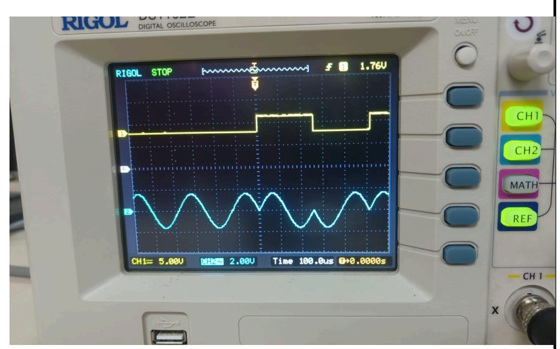
Raj-202201403, Bhoomish-202201414

Manthan-202201416, Rakshit-202201426

Binary Phase Shift Keying

Experiment 7

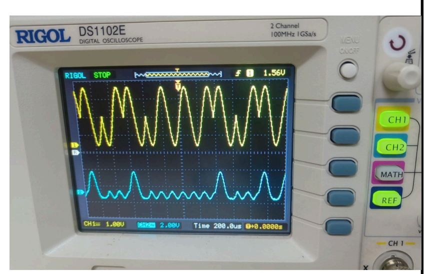
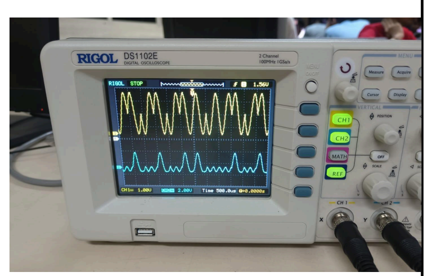
Data Pattern	**Data Frequency**	**CH1**	**CH2**	**Output**
8-Bit	2KHz	Input Data (TP10)	Carrier Signal (TP11)	

Data Pattern	**Data Frequency**	**CH1**	**CH2**	**Output**
8-Bit	2KHz	Input Data (TP10)	BPSK Out (TP12)	
32-Bit	4KHz	Carrier Signal (TP11)	BPSK Out (TP12)	
32-Bit	4KHz	Input Data (TP10)	BPSK Out (TP12)	

Conclusion:

1. BPSK modulation shifts the phase of the carrier signal by 180 degrees based on input data.
2. The modulator successfully encodes binary data into phase-shifted carrier signals, with clear output at TP12.
3. BPSK is ideal for robust data transmission in noisy environments.

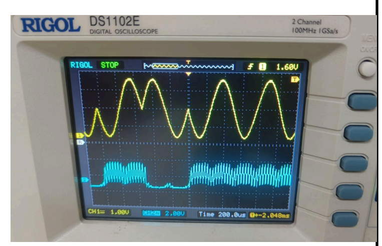
Experiment 8

Data Pattern	**Data Frequency**	**CH1**	**CH2**	**Output**
32-Bit	4KHz	BPSK Out (TP12)	Multiplier Out (TP14)	
8-Bit	2KHz	BPSK Out (TP12)	Multiplier Out (TP14)	

Conclusion:

1. The complex multiplier modulates the signal by multiplying the input data with a carrier.
2. Clear correlation is observed between BPSK output (TP12) and the multiplier output (TP14).
3. The experiment confirms the phase transition behavior of BPSK.

Experiment 9

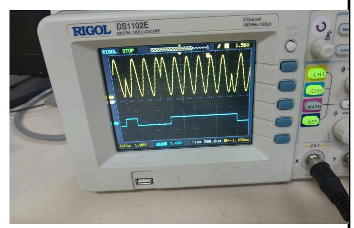
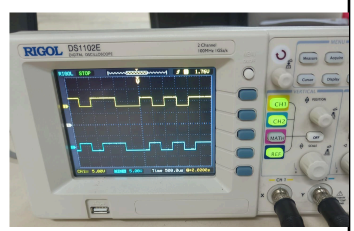
Data Pattern	**Data Frequency**	**CH1**	**CH2**	**Output**
32-Bit	2KHz	BPSK Out (TP12)	Integrator Out (TP15)	

Conclusion:

1. The integrator smooths the BPSK signal and provides a continuous signal.
2. Output at TP15 confirms successful signal integration, helping to filter high-frequency noise.
3. This process is crucial in recovering clean signals for

demodulation.

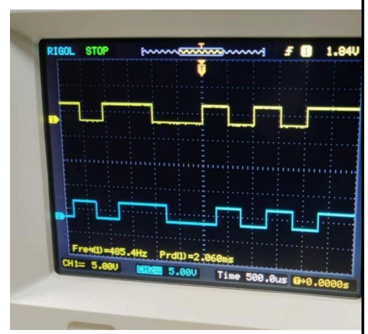
Experiment 10

Data Pattern	Data Frequency	CH1	CH2	Output
32-Bit	2KHz	BPSK Out (TP12)	Comparator Out (TP16)	
8-Bit	2KHz	Encoded Input Data (TP10)	Comparator Out (TP16)	

Conclusion:

1. The comparator successfully recovers digital data by comparing the integrator's output against a threshold.
2. The output at TP16 demonstrates accurate signal comparison for 1s and 0s.
3. The process aids in the final recovery of binary data from the BPSK modulated signal.

Experiment 11

Data Pattern	**Data Frequency**	**CH1**	**CH2**	**Output**
8-Bit	2KHz	Encoded Input Data (TP10)	Demodulator Out (TP16)	

Conclusion:

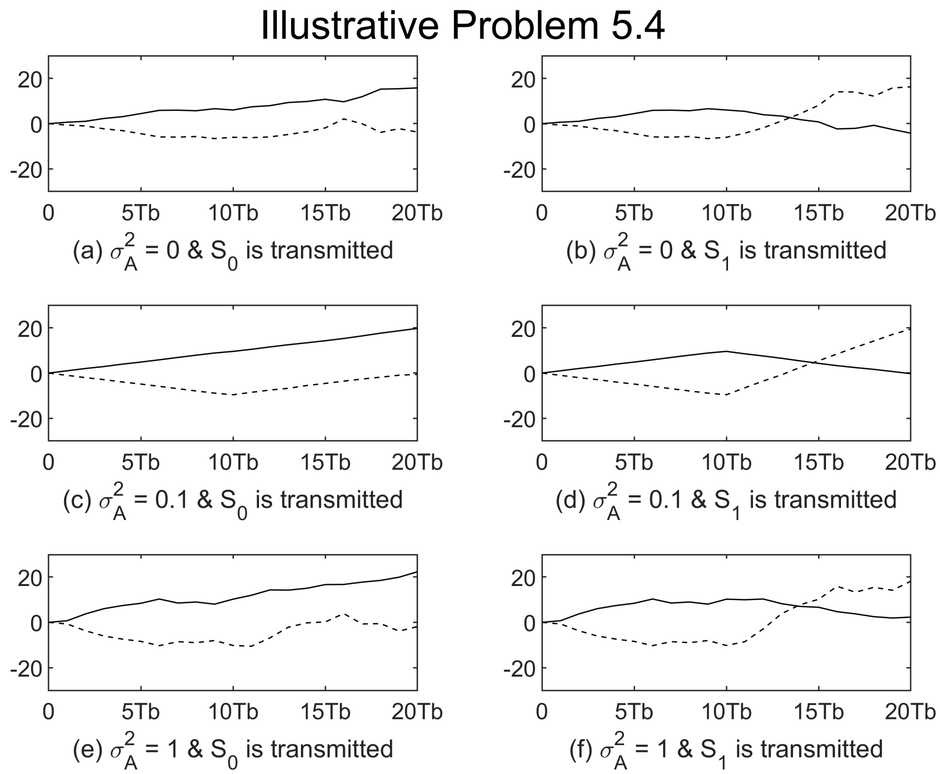
1. The BPSK demodulator accurately demodulates the phase-shifted signal, recovering the input binary data.
2. Output at TP16 confirms successful demodulation.
3. This experiment demonstrates the efficiency of BPSK in noise-resistant data recovery.

MATLAB Exercise Lab 8

Illustrative Problem 5.4

```
N = 20;
A = 1;
t = 0:N;

s0 = A*ones(1,N);
s1 = [A*ones(1,N/2), - A*ones(1,N/2)];

y0 = zeros(1,N);
y1 = zeros(1,N);

%Case 1 noise ~ N(0,0)
noise = randn(1,N);

%Sub case 1 s = s0:
y = s0 +noise;
y0 = conv(y,fliplr(s0));
y1 = conv(y,fliplr(s1));

subplot(3,2,1);
plot(t,[0,y0(1:N)],'-k',t,[0,y1(1:N)],'--k')
set(gca,'XTickLabel',{'0','5Tb','10Tb','15Tb','20Tb'})
axis([0 20 -30 30])
xlabel('(a) \sigma_A^2 = 0 & S_0 is transmitted')


%Sub case 2 s = s1:
y = s1 +noise;
y0 = conv(y,fliplr(s0));
y1 = conv(y,fliplr(s1));

subplot(3,2,2);
plot(t,[0,y0(1:N)],'-k',t,[0,y1(1:N)],'--k')
set(gca,'XTickLabel',{'0','5Tb','10Tb','15Tb','20Tb'})
axis([0 20 -30 30])
xlabel('(b) \sigma_A^2 = 0 & S_1 is transmitted')


%Case 2 noise ~ N(0,0.1)
noise = randn(1,N)*0.1;

%Sub case 1 s = s0:
y = s0 +noise;
y0 = conv(y,fliplr(s0));
y1 = conv(y,fliplr(s1));

subplot(3,2,3);
plot(t,[0,y0(1:N)],'-k',t,[0,y1(1:N)],'--k')
set(gca,'XTickLabel',{'0','5Tb','10Tb','15Tb','20Tb'})
axis([0 20 -30 30])
```

```
xlabel('(c) \sigma_A^2 = 0.1 & S_0 is transmitted')


%Sub case 2 s = s1:
y = s1 +noise;
y0 = conv(y,fliplr(s0));
y1 = conv(y,fliplr(s1));

subplot(3,2,4);
plot(t,[0,y0(1:N)],'-k',t,[0,y1(1:N)],'--k')
set(gca,'XTickLabel',{'0','5Tb','10Tb','15Tb','20Tb'})
axis([0 20 -30 30])
xlabel('(d) \sigma_A^2 = 0.1 & S_1 is transmitted')



%Case 3 noise ~ N(0,1)
noise = randn(1,N);

%Sub case 1 s = s0:
y = s0 +noise;
y0 = conv(y,fliplr(s0));
y1 = conv(y,fliplr(s1));

subplot(3,2,5);
plot(t,[0,y0(1:N)],'-k',t,[0,y1(1:N)],'--k')
set(gca,'XTickLabel',{'0','5Tb','10Tb','15Tb','20Tb'})
axis([0 20 -30 30])
xlabel('(e) \sigma_A^2 = 1 & S_0 is transmitted')


%Sub case 2 s = s1:
y = s1 +noise;
y0 = conv(y,fliplr(s0));
y1 = conv(y,fliplr(s1));

subplot(3,2,6);
plot(t,[0,y0(1:N)],'-k',t,[0,y1(1:N)],'--k')
set(gca,'XTickLabel',{'0','5Tb','10Tb','15Tb','20Tb'})
axis([0 20 -30 30])
xlabel('(f) \sigma_A^2 = 1 & S_1 is transmitted')

sgtitle('Illustrative Problem 5.4');
```



Problem 5.4

```
N = 20;
A = 1;
t = 0:N;

s0 = A * ones(1, N);
s1 = [A * ones(1, N/2), -A * ones(1, N/2)];

variances = [0.1, 1, 3];
num_variances = length(variances);

figure;

for idx = 1:num_variances
sigma2 = variances(idx);

% Case: noise ~ N(0, sigma2)
noise = randn(1, N) * sqrt(sigma2);

% Sub-case s = s0:
y0 = s0 + noise;
output0 = conv(y0, fliplr(s0));

% Sub-case s = s1:
```

```
y1 = s1 + noise;
output1 = conv(y1, fliplr(s1));


subplot(3, 2, idx * 2 - 1);
plot(t, [0, output0(1:N)], '-k', t, [0, output1(1:N)], '--k');
title(['\sigma^2 = ' num2str(sigma2) ' & S_0 is transmitted']);
set(gca, 'XTickLabel', {'0', '5Tb', '10Tb', '15Tb', '20Tb'});
axis([0 20 -30 30]);
xlabel('Time (Tb)');

subplot(3, 2, idx * 2);
plot(t, [0, output0(1:N)], '-k', t, [0, output1(1:N)], '--k');
title(['\sigma^2 = ' num2str(sigma2) ' & S_1 is transmitted']);
set(gca, 'XTickLabel', {'0', '5Tb', '10Tb', '15Tb', '20Tb'});
axis([0 20 -30 30]);
xlabel('Time (Tb)');
end

sgtitle('Exercise Problem 5.4');
```