EXPERIMENT – 8

BINARY PHASE SHIFT KEYING BPSK

March 25 - 2025
B Siddharth Sekhar - EC22B1064

Aim:

To implement Binary Phase Shift Keying (BPSK) on a specified binary message sequence, decode the signal, and compare it with the original message. Additionally, analyze the effect of noise on the BPSK demodulation process and plot the corresponding waveforms.

Theory:

BINARY PHASE SHIFT KEYING (BFSK):

Binary Phase Shift Keying (BPSK) is a digital modulation technique where the phase of the carrier signal changes based on the binary data. The carrier frequency remains constant, while the phase shifts by 180 degrees depending on whether the bit is '1' or '0'.

Mathematical Representation

The BPSK modulated signal is given by:

$$s(t) = Acos(2\pi f_c t) \quad if\ b(t) = 1$$
$$s(t) = -Acos(2\pi f_c t) \quad if\ b(t) = 0$$

where: A is the carrier amplitude, f_c is the carrier frequency, b(t) is the binary message signal.

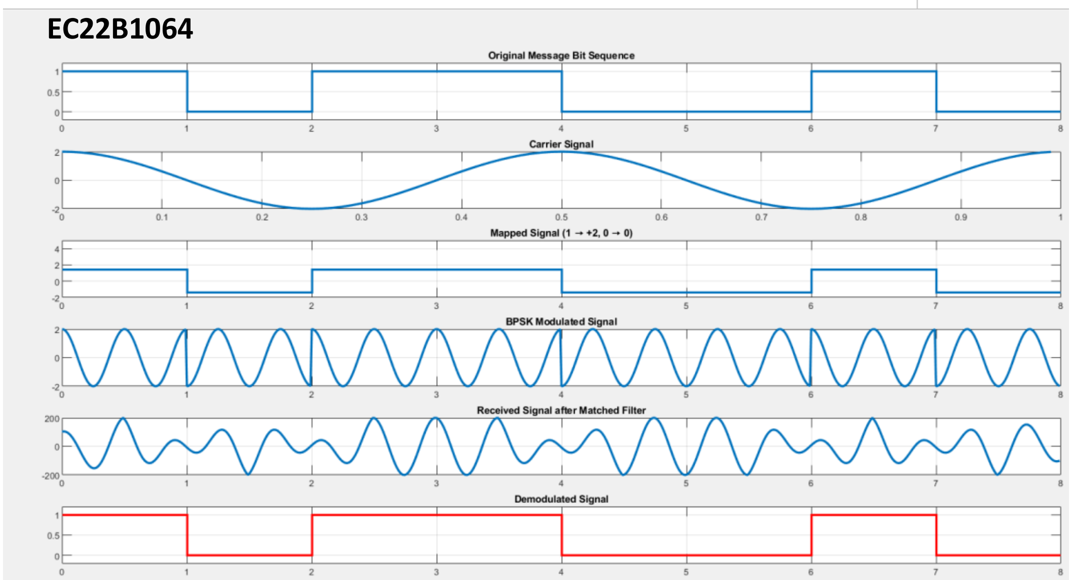
Advantages of BPSK:

High noise immunity compared to ASK and BFSK. Simple coherent detection using matched filtering. Disadvantages of BPSK: Requires coherent detection for demodulation. More complex circuitry compared to ASK.

Q1)Perform Binary Phase Shift Keying (BFSK) and decode the signal.

```
Tb = 1;
fc = 2;
fs = 100;
t = 0:1/fs:Tb-1/fs;
A = sqrt(2/Tb);
Eb = 2;

message = [1 0 1 1 0 0 1 0];
N = length(message);
t_total = 0:1/fs:N*Tb-1/fs;

carrier = A * sqrt(Eb) * cos(2*pi*fc*t);

mapped_signal = sqrt(Eb) * (2*message - 1);
bpsk_signal = [];
```

```
for i = 1:N
    if message(i) == 1
        bpsk_signal = [bpsk_signal, carrier];
    else
        bpsk_signal = [bpsk_signal, -carrier];
    end
end

matched_filter = fliplr(carrier);

received_signal = conv(bpsk_signal, matched_filter, 'same') ;

demodulated_bits = zeros(1, N);
for i = 1:N
    segment = received_signal((i-1)*fs+1:i*fs);
    correlation = sum(segment .* carrier);
    demodulated_bits(i) = correlation > 0;
end
figure;
subplot(6,1,1);
stairs((0:N)*Tb, [message message(end)], 'LineWidth', 2);
title('Original Message Bit Sequence'); ylim([-0.2 1.2]); grid on;

subplot(6,1,2);
plot(t, carrier,'LineWidth', 2);
title('Carrier Signal'); grid on;

subplot(6,1,3);
stairs((0:N)*Tb, [mapped_signal mapped_signal(end)], 'LineWidth', 2);
title('Mapped Signal (1 → +2, 0 → 0)'); ylim([-2 5]); grid on;

subplot(6,1,4);
plot(t_total, bpsk_signal, 'LineWidth', 2);
title('BPSK Modulated Signal'); grid on;

subplot(6,1,5);
plot(t_total, received_signal,'LineWidth', 2);
title('Received Signal after Matched Filter'); grid on;


subplot(6,1,6);
stairs((0:N)*Tb, [demodulated_bits demodulated_bits(end)], 'r', 'LineWidth', 2);
title('Demodulated Signal'); ylim([-0.2 1.2]); grid on;


disp('Original Message:');
disp(message);
disp('Demodulated Bits:');
disp(demodulated_bits);
```



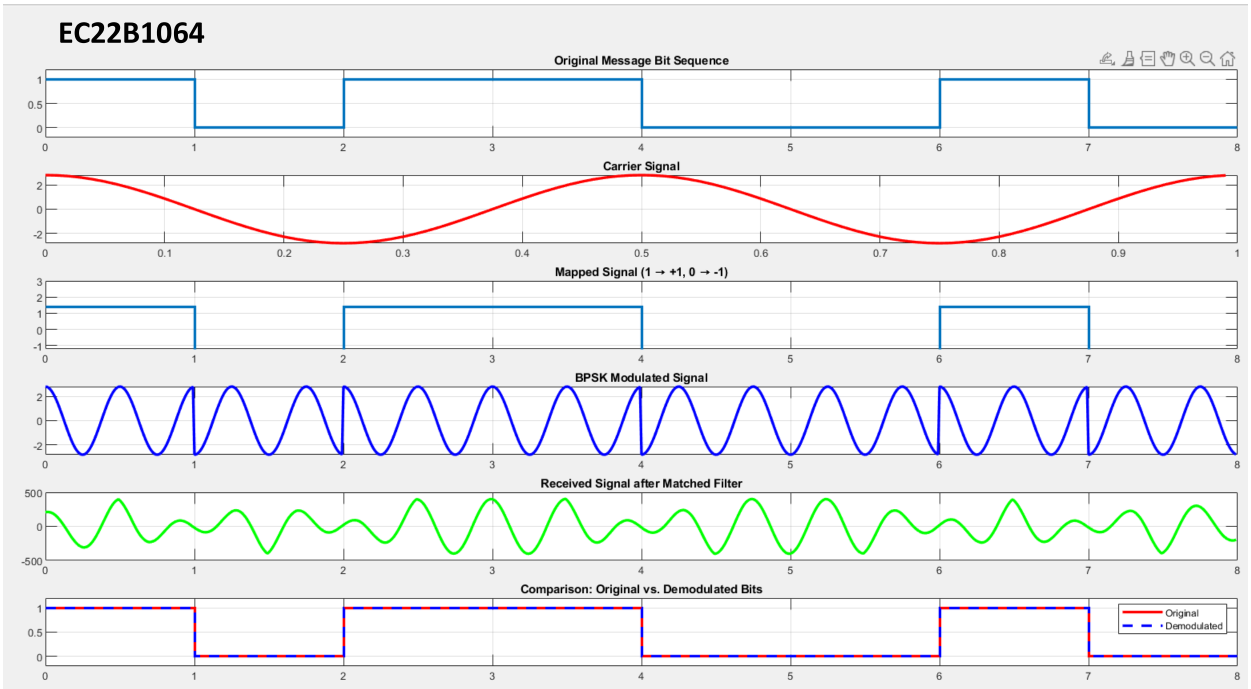
```
Original Message:
  Columns 1 through 7

     1     0     1     1     0     0     1

  Column 8

     0

Demodulated Bits:
  Columns 1 through 7

     1     0     1     1     0     0     1

  Column 8

     0
```

EC22B1064

Q2) Analyze the effect of noise on BPSK demodulation process by adding AWGN to the BPSK Modulated signal

```
Tb = 1;
fc = 2;
fs = 100;
t = 0:1/fs:Tb-1/fs;
A = sqrt(2/Tb);
Eb = 2;

message = [1 0 1 1 0 0 1 0];
N = length(message);
t_total = 0:1/fs:N*Tb-1/fs;

carrier = A * Eb * cos(2*pi*fc*t);

mapped_signal = sqrt(Eb) * (2*message - 1);

bpsk_signal = [];
for i = 1:N
    if message(i) == 1
        bpsk_signal = [bpsk_signal, carrier];
    else
        bpsk_signal = [bpsk_signal, -carrier];
    end
end

snr_db = 20;
bpsk_signal_noisy = awgn(bpsk_signal, snr_db, 'measured');

matched_filter = fliplr(carrier);
received_signal = conv(bpsk_signal_noisy, matched_filter, 'same');


demodulated_bits = zeros(1, N);
for i = 1:N
    segment = received_signal((i-1)*fs+1:i*fs);
    correlation = sum(segment .* carrier);
    demodulated_bits(i) = correlation > 0;
end

figure;
subplot(6,1,1);
```

```
stairs([0:N]*Tb, [message message(end)], 'LineWidth', 2);
title('Original Message Bit Sequence'); ylim([-0.2 1.2]); grid on;

subplot(6,1,2);
plot(t, carrier, 'r', 'LineWidth', 2);
title('Carrier Signal'); grid on;

subplot(6,1,3);
stairs([0:N]*Tb, [mapped_signal mapped_signal(end)], 'LineWidth', 2);
title('Mapped Signal (1 → +1, 0 → -1)'); ylim([-1.2 3]); grid on;

subplot(6,1,4);
plot(t_total, bpsk_signal, 'b', 'LineWidth', 2);
title('BPSK Modulated Signal'); grid on;

subplot(6,1,5);
plot(t_total, received_signal, 'g', 'LineWidth', 2);
title('Received Signal after Matched Filter'); grid on;

subplot(6,1,6);
stairs([0:N]*Tb, [message message(end)], 'r', 'LineWidth', 2); hold on;
stairs([0:N]*Tb, [demodulated_bits demodulated_bits(end)], 'b--', 'LineWidth', 2);
title('Comparison: Original vs. Demodulated Bits'); ylim([-0.2 1.2]); grid on;
legend('Original', 'Demodulated');

disp('Original Message:');
disp(message);
disp('Demodulated Bits:');
disp(demodulated_bits);
ber = sum(demodulated_bits ~= message) / N;
disp(['Bit Error Rate (BER): ', num2str(ber)]);
```



Inference:

In BPSK, binary symbols are represented by phase shifts, making it highly robust against noise. The presence of noise (AWGN) affects the demodulation process, leading to potential bit errors. Trade-off: BPSK offers better noise performance but requires phase-coherent detection.

Conclusion:

Binary Phase Shift Keying (BPSK) is an efficient modulation technique widely used in digital communication. BPSK provides better noise immunity than BFSK but requires coherent demodulation. The experiment successfully demonstrates BPSK modulation and demodulation, verifying the transmitted and received signals.

References:

[1] Simon Haykins, Communication systems, 2nd ed. (New York John Wiley and Sons, 2005).