**Aim ↬ To perform PSK [Phase Shift Keying] using a trainer kit & verify by MATLAB.**

**Apparatus Required ↬** MATLAB.

**Theory ↬**

**Phase Shift Keying (PSK)** is a digital modulation technique where the phase of the carrier signal is varied to represent binary or multi-level data. Unlike Amplitude Shift Keying (ASK) or Frequency Shift Keying (FSK), PSK keeps both the amplitude and frequency of the carrier signal constant and encodes information in the phase changes of the carrier.

In **Binary Phase Shift Keying (BPSK)**, the simplest form of PSK, two phases are used to represent binary data:

Binary '1' is represented by a phase of 0°.

$$s(t) = Acos(2\pi f_c t)$$

Binary '0' is represented by a phase of 180° (or a phase shift of π).

$$s(t) = Acos(2\pi f_c t + \pi) = -Acos(2\pi f_c t)$$

Here, A is the amplitude of the carrier, $f_c$ is the carrier frequency, and t is time.

For higher data rates, **Quadrature Phase Shift Keying (QPSK)** is used, which employs four distinct phase shifts (0°,90°,180°,270°) to encode two bits per symbol. The general representation is:

$$s(t) = Acos(2\pi f_c t + \phi)$$

where ϕ can be 0,π/2,π, or 3π/2, depending on the bit pair.

PSK signals require less bandwidth compared to FSK for the same data rate. The bandwidth for BPSK is approximately equal to the bit rate, B≈bit rate, making it spectrally efficient.

PSK, especially BPSK, is robust against noise because it encodes data in phase changes less affected by amplitude noise. However, it is sensitive to phase noise and requires accurate synchronization between the transmitter and receiver.

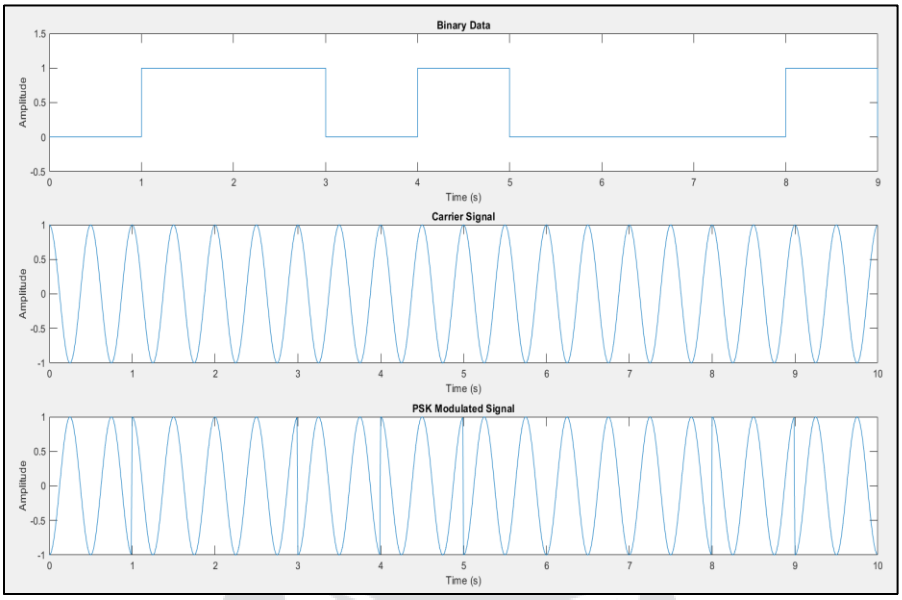
PSK is widely used in Satellite communication, Wireless LANs (e.g., Wi-Fi standards using QPSK), Cellular networks, RFID systems, etc. PSK balances bandwidth efficiency, noise immunity, and implementation complexity, making it a popular choice in modern digital communication systems.

**Code ↬**

```
%Phase Shift Keying
clc;
clear;
close all;

Fs = 100;
f0 = 2;
ph = pi;
bit_rate = 1;
Tb = 1/bit_rate;
t = 0:1/Fs:Tb-1/Fs;

data = [0 1 1 0 1 0 0 0 1 0];
N = length(data);

psk_signal = [];

for i = 1:N
    if data(i) == 1
        fsk_bit = cos(2 * pi * f0 * t);
    else
        fsk_bit = cos(2 * pi * f0 * t + ph);
    end
    psk_signal = [psk_signal fsk_bit];
end

t_total = 0:1/Fs:Tb*N-1/Fs;

figure;
subplot(3,1,1);
stairs([0:N-1]*Tb, data);
xlabel('Time (s)');
ylabel('Amplitude');
title('Binary Data');
ylim([-0.5 1.5]);

subplot(3,1,2);
plot(t_total, cos(2 * pi * f0 * t_total));
xlabel('Time (s)');
ylabel('Amplitude');
title('Carrier Signal');

subplot(3,1,3);
plot(t_total, psk_signal);
```

```
xlabel('Time (s)');
ylabel('Amplitude');
title('PSK Modulated Signal');
```

## Output ↬




## Result ↬

The PSK modulation was successfully demonstrated in MATLAB, where phase shifts in the carrier signal effectively represented binary values.

## Conclusion ↬

The experiment confirms that PSK is a robust and bandwidth-efficient method of modulating binary data by varying the phase of the carrier signal. It is particularly resilient to amplitude-based noise.

## Precautions ↬

- Ensure accurate phase shifts are implemented to represent the binary data correctly.
- Verify that MATLAB appropriately sets the carrier frequency, amplitude, and modulation parameters to generate the desired PSK signal.
- Check the MATLAB plots for proper time and phase scale alignment to visualize the phase transitions clearly.
- Maintain precise synchronization between transmitter and receiver to prevent errors in phase detection.