Aim ↬ To perform FSK [Frequency Shift Keying] using MATLAB.

Apparatus Required ↬ MATLAB.

Theory ↬

Frequency Shift Keying (FSK) is a digital modulation technique where the frequency of the carrier signal is varied to represent binary data. Unlike Amplitude Shift Keying (ASK), where the amplitude changes, FSK uses two distinct frequencies to denote binary values. This method is less susceptible to noise than ASK since it relies on frequency changes rather than amplitude.

In FSK, binary data is represented by alternating the frequency of a continuous carrier signal:

- **Binary '1'** is represented by a high frequency f_1.

$$s(t) = Acos(2\pi f_1 t + \phi)$$

- **Binary '0'** is represented by a lower frequency f_0.

$$s(t) = Acos(2\pi f_0 t + \phi)$$

This is known as **Binary Frequency Shift Keying (BFSK)**, the simplest form of FSK, where only two frequencies are used. However, more advanced FSK systems can use multiple frequencies for greater data throughput, a technique known as **Multiple Frequency Shift Keying (MFSK)**.

FSK signals occupy a bandwidth given by:

$$B = 2(f_1 - f_0) + 2 \cdot bit\, rate$$

In practical applications, the spacing between f_1 and f_0 must be wide enough to avoid interference between the two frequencies, which ensures a clear distinction for each binary value.

FSK is less susceptible to amplitude noise, as data is encoded in frequency rather than amplitude. It is also more robust in environments with varying signal strength. Demodulation is achieved by detecting the frequency changes, which is more reliable in noisy channels compared to amplitude or phase changes.

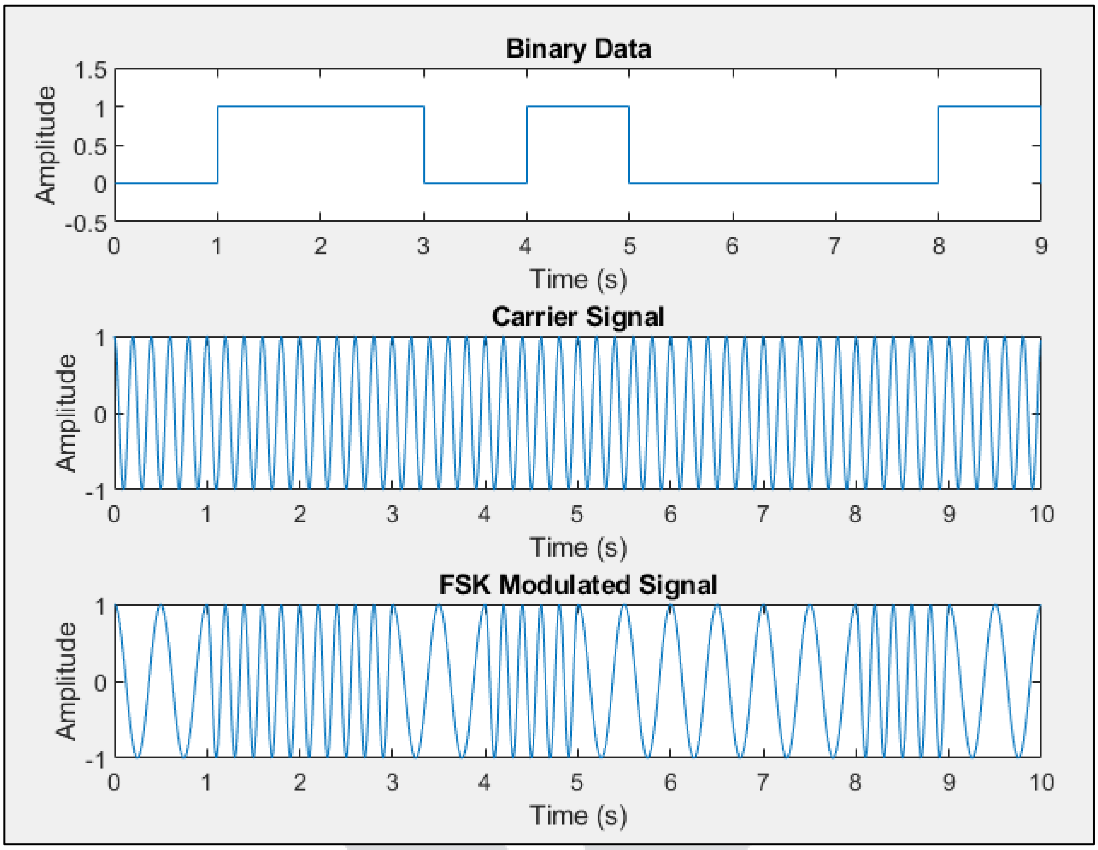
FSK is used in modem communication, low-speed wireless communication, and areas where noise immunity is critical.

Code ↬

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

Fs = 100;
f1 = 5;
f0 = 2;
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);

fsk_signal = [];

for i = 1:N
    if data(i) == 1
        fsk_bit = cos(2 * pi * f1 * t);
    else
        fsk_bit = cos(2 * pi * f0 * t);
    end
    fsk_signal = [fsk_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 * f1 * t_total));
xlabel('Time (s)');
ylabel('Amplitude');
title('Carrier Signal');

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

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

Output ↬



Result ↬

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

Conclusion ↬

The experiment confirms that FSK provides an efficient way to modulate binary data by varying frequency, making it robust against amplitude-based noise.

Precautions ↬

- Ensure the carrier frequencies for binary '0' and '1' are selected to avoid signal distortion.
- Check that the binary data and modulation settings are correctly inputted into MATLAB for accurate signal generation.
- Verify the time and amplitude scales on the MATLAB plot to visualize the modulated signal clearly.