Title: To understand the modulation procedure of Amplitude Shift Keying (ASK), Phase Shift Keying (PSK), and Frequency Shift Keying (FSK).

Objective:

The objective of this lab experiment is to investigate and compare the performance of three common modulation techniques - Amplitude Shift Keying (ASK), Phase Shift Keying (PSK), and Frequency Shift Keying (FSK) - in transmitting digital data over a communication channel. Through simulation using Google Collaboratory and NumPy, we aim to analyze and contrast the efficiency, robustness, and spectral characteristics of each modulation scheme.

Apparatus:

1. Google Collaboratory
2. NumPy library for numerical computations
3. Jupyter Notebook for code execution and documentation

Theory:

Amplitude Shift Keying (ASK): ASK is a digital modulation technique where the amplitude of a carrier signal is varied in accordance with the digital data being transmitted. The presence of the carrier wave at a particular amplitude represents one bit of digital data, while its absence represents the other bit.

Phase Shift Keying (PSK): PSK involves changing the phase of the carrier signal to represent different symbols or bits of digital data. This modulation technique is sensitive to phase changes and is widely used in digital communication systems.

Frequency Shift Keying (FSK): FSK involves modulating the frequency of the carrier signal based on the digital data being transmitted. Each frequency represents a different symbol or bit, and the receiver interprets these frequency changes to decode the transmitted information.

Procedure:

1. We generated digital data to be transmitted.
2. Then we implemented ASK, PSK, and FSK modulation schemes using Python and NumPy.
3. Then compared and contrast the visual representation of ASK, PSK, and FSK.

Simulation Procedure:

1. We generated a carrier signal and digital data.
2. Then we modulated the carrier signal using ASK, PSK, and FSK modulation techniques.
3. Then we plotted graphs and visualize the results for comparison.

Experimental Code & Graph:

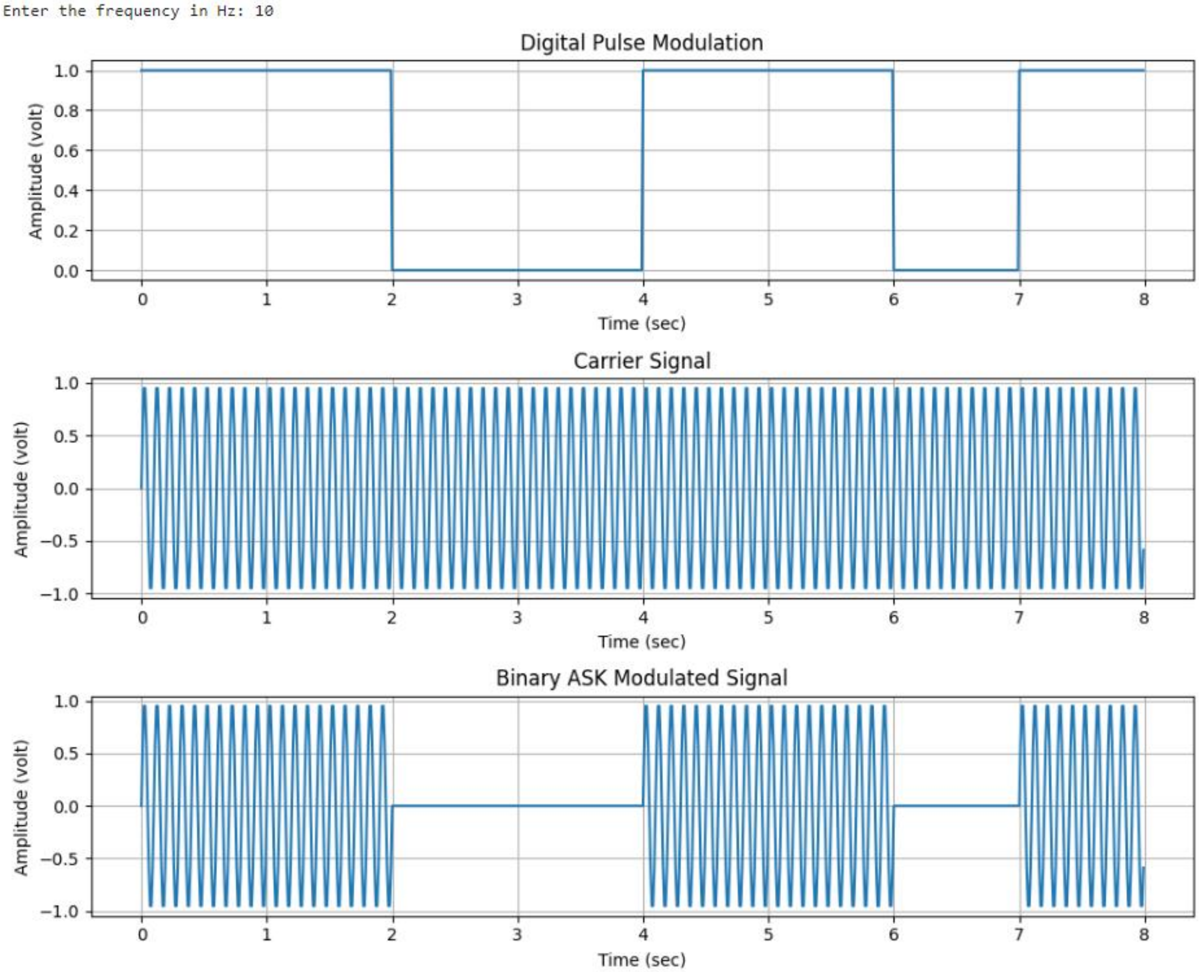
```
import numpy as np
import matplotlib.pyplot as plt
```

ASK

```
[2] # User input for frequency to form carrier signal
    frequency = float(input('Enter the frequency in Hz: '))

    # Binary sequence
    binary_sequence = np.array([1, 1, 0, 0, 1, 1, 0, 1])
    num_bits = len(binary_sequence)

    # Time array
    time = np.arange(0, num_bits, 0.01)

    # Creating the pulse train
    pulse_train = np.repeat(binary_sequence, 100)[:len(time)]

    # Generating carrier signal
    carrier_signal = np.sin(2 * np.pi * frequency * time)

    # Modulating the signal
    modulated_signal = pulse_train * carrier_signal

    # Plotting
    plt.figure(figsize=(10, 8))
```

```
plt.subplot(3, 1, 1)
plt.plot(time, pulse_train)
plt.title('Digital Pulse Modulation')
plt.xlabel('Time (sec)')
plt.ylabel('Amplitude (volt)')
plt.grid(True)

plt.subplot(3, 1, 2)
plt.plot(time, carrier_signal)
plt.title('Carrier Signal')
plt.xlabel('Time (sec)')
plt.ylabel('Amplitude (volt)')
plt.grid(True)

plt.subplot(3, 1, 3)
plt.plot(time, modulated_signal)
plt.title('Binary ASK Modulated Signal')
plt.xlabel('Time (sec)')
plt.ylabel('Amplitude (volt)')
plt.grid(True)

plt.tight_layout()
plt.show()
```

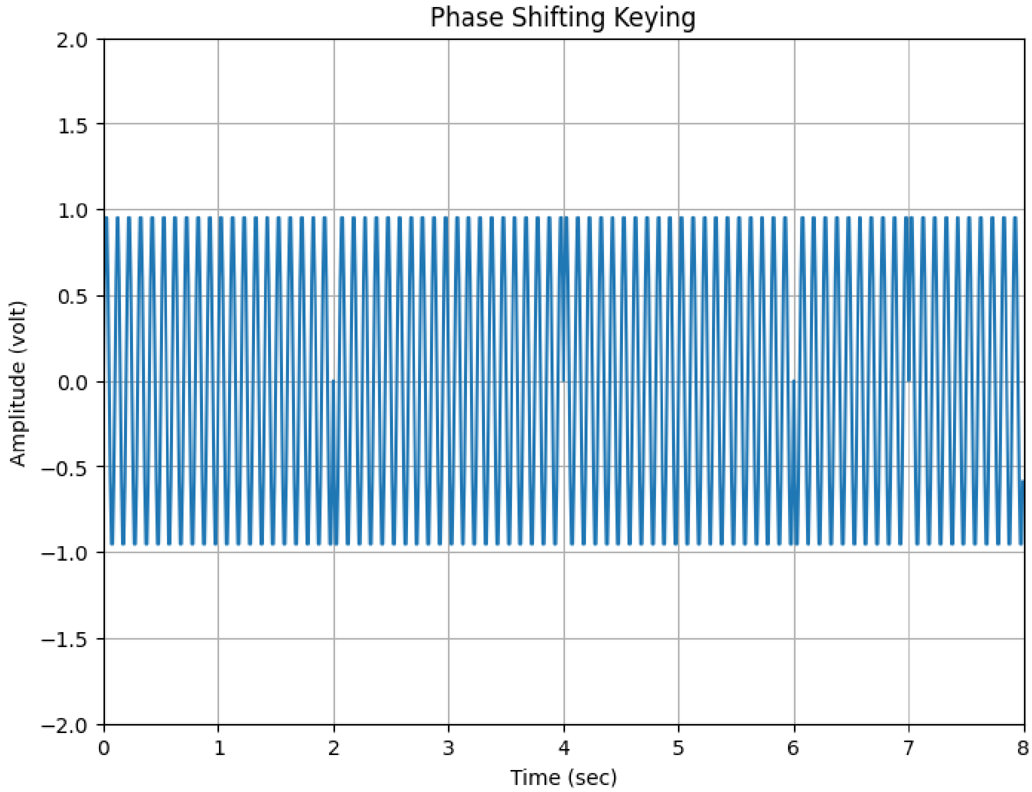
```
Enter the frequency in Hz: 10
```



PSK

```
[3] # Binary sequence
binary_sequence = np.array([1, 1, 0, 0, 1, 1, 0, 1])

# Convert binary sequence to ±1 modulation
nn = np.where(binary_sequence == 1, 1, -1)

# Generate time array
t = np.arange(0, len(binary_sequence), 0.01)

# Perform Phase Shifting Keying modulation
y = np.zeros_like(t)
i = 0
for j, tj in enumerate(t):
    if i < len(nn):
        y[j] = nn[i]
        if tj >= i + 1:
            i += 1

# Generate carrier signal
frequency = float(input('Enter the frequency in Hz: '))
carrier_signal = np.sin(2 * np.pi * frequency * t)

# Modulate the signal
modulated_signal = y * carrier_signal

# Plotting
plt.figure(figsize=(8, 6))

plt.plot(t, modulated_signal)
plt.title('Phase Shifting Keying')
plt.xlabel('Time (sec)')
plt.ylabel('Amplitude (volt)')
plt.grid(True)
plt.axis([0, len(binary_sequence), -2, 2])

plt.show()
```

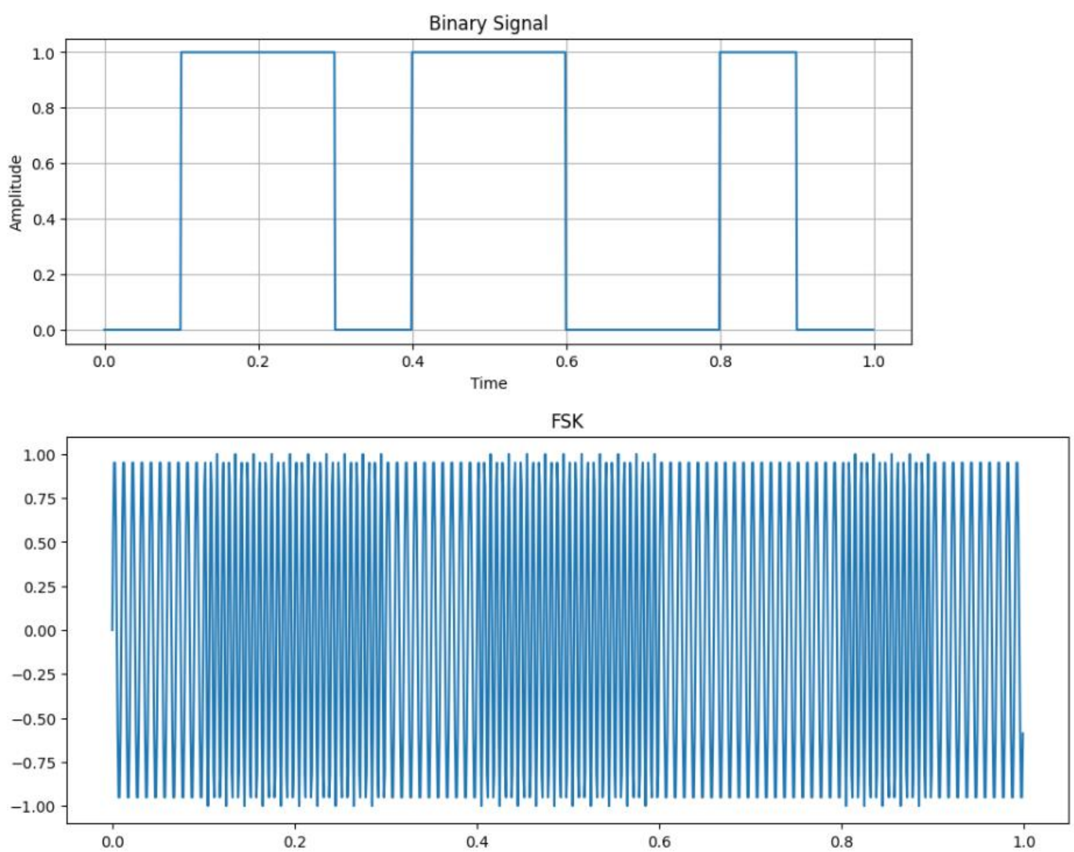
```
Enter the frequency in Hz: 10
```



FSK

```
import matplotlib.pyplot as plt
import numpy as np
from math import pi

# Generating random binary sequence
def binary(symbol, sym_len):
    # Generate random binary sequence
    rand_n = np.random.randint(0, 2, symbol)
    sig = np.zeros(int(symbol*sym_len))

    # Generate symbols
    for i in range(len(rand_n)):
        if rand_n[i] == 1:
            temp = int(i*sym_len)
            sig[temp:temp+sym_len] = 1
    return sig

# Carrier information
Fs = 1000
fc = 100   # Carrier frequency
T = 1      # Simulation time
t = np.arange(0, T, 1/Fs)

# Carrier wave
x = np.sin(2*pi*fc*t)

# Generating random binary signal
Td = 0.1    # Bit duration
samples = int(Td*Fs)
sym = int(np.floor(np.size(t)/samples))

sig = binary(sym, samples)
```

```python
plt.figure(figsize=(10, 8))


plt.subplot(2,1,1)
plt.plot(t, sig)
plt.xlabel('Time')
plt.ylabel('Amplitude')
plt.title('Binary Signal')
plt.grid()

plt.show()

plt.figure(figsize=(10, 8))
#FSK
Freq = fc + fc*sig/2
fsk = np.sin(2*pi*Freq*t)

plt.subplot(2,1,2)
plt.plot(t,fsk)
plt.title('FSK')

plt.tight_layout()
```



Discussion:

Through the experiment we got a coding understanding of modulation system. We got our basic idea about ASK,PSK,FSK.

We familiarized with them also with there NumPY code. We tried to understand how it works theoretically and then taking a signal we implemented it. At first some of the portion at multiplier of original signal and carrier signal was not visualized properly. So we took help of books and we got the solution how to sequence the carrier signal to get an proper modulated signal.