Title: Analysis and Processing of PPG Signal with Noise, Filtering, Normalization, and Peak Detection

Objectives:

The objective of this experiment is to analyze a Photoplethysmogram (PPG) signal by:

1. Simulating the PPG Signal
2. Apply filtering to remove noise.
3. Normalize the filtered signal to a standard range.
4. Detect the peaks representing heartbeats from the processed signal.

Theory:

Photoplethysmography (PPG) is a non-invasive optical measurement technique used to detect blood volume changes in the microvascular bed of tissue. It is widely used for heart rate measurement, oxygen saturation levels, and other physiological parameters. In this experiment, we analyze a simulated PPG signal, which consists of a clean PPG signal contaminated with noise. The goal of this report is to:

This process is essential to improve the signal quality and ensure accurate heart rate estimation, which is typically calculated by detecting the peaks of the PPG signal corresponding to heartbeats. The goal of this report is to:

1. Simulating the PPG Signal

To simulate the PPG signal, we create a clean signal that consists of two sinusoidal components:

- A **1 Hz** component representing the heart rate (approximately 60 beats per minute).
- A **0.05 Hz** component representing low-frequency baseline wander (e.g., movement or other environmental noise).

We then add Gaussian noise to simulate real-world conditions where PPG signals are often contaminated with various types of interference.

2. Bandpass Filtering

A **bandpass filter** is applied to the noisy PPG signal to remove noise outside the typical heart rate frequency range. The filter is designed to pass frequencies between **0.5 Hz** and **5 Hz**, which corresponds to the expected frequency range of the heart rate. The filter is created using a

Butterworth filter design with a filter order of 2, and the `filtfilt` function is used to apply the filter to the signal.

3. Normalization

Normalization of the signal is performed to scale the filtered signal to the range of [0, 1]. This is done by subtracting the minimum value of the signal and dividing by the range (max - min). Normalization helps in standardizing the signal for further analysis and comparison.

4. Peak Detection

The **peaks** in the filtered PPG signal are detected using the `find_peaks` function from the **SciPy** library. The function identifies the locations where the PPG signal reaches its maximum amplitude, which corresponds to the heartbeats. A minimum distance of **fs/2.5** (sampling rate divided by 2.5) is specified to avoid detecting multiple peaks within the same heartbeat cycle.

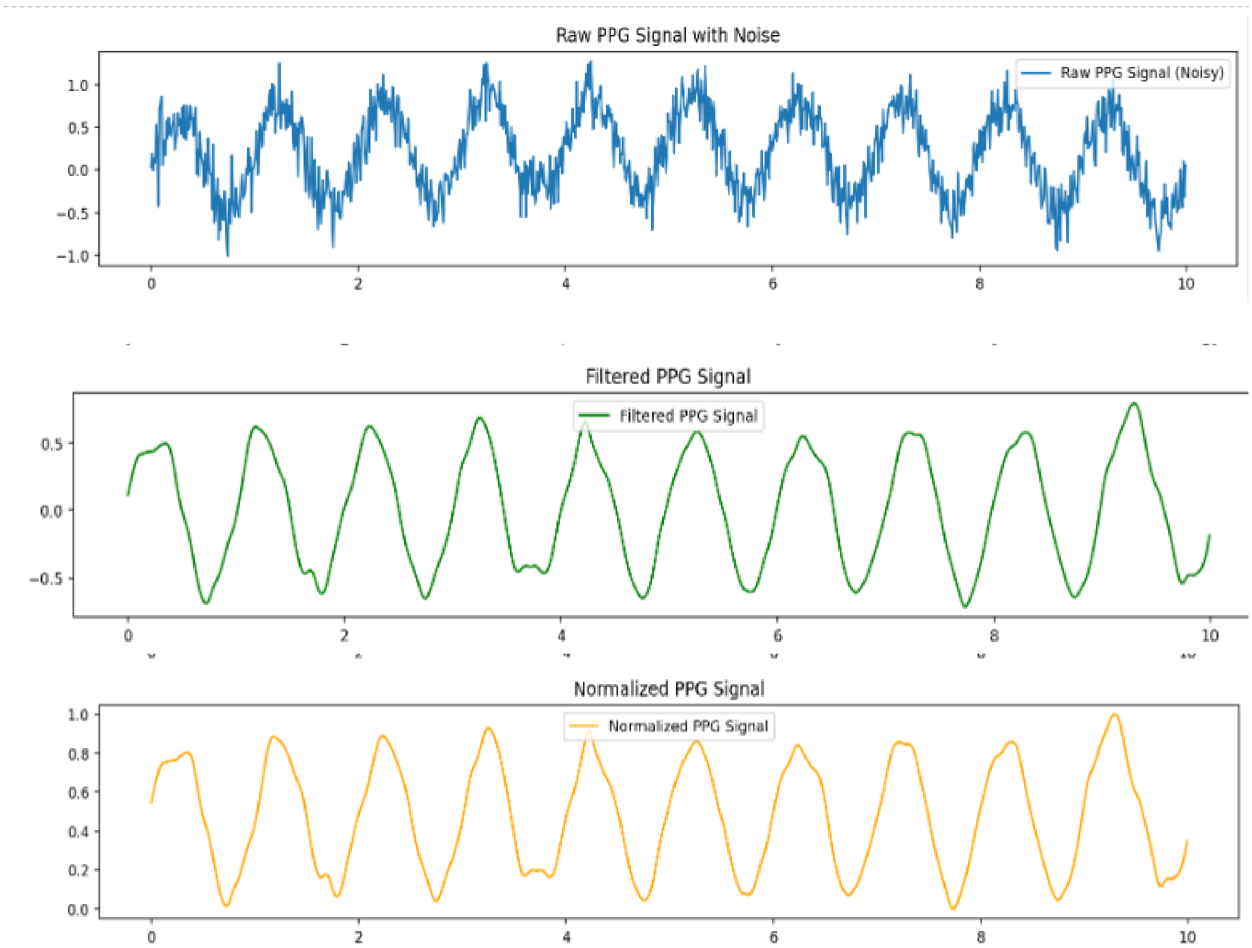
Source code:

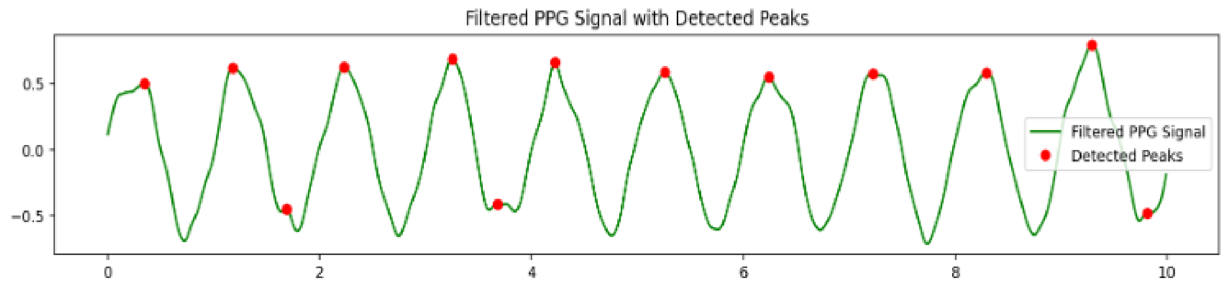
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import butter, filtfilt, find_peaks

# Function to design a bandpass filter
def butter_bandpass(lowcut, highcut, fs, order=2):
    nyquist = 0.5 * fs
    low = lowcut / nyquist
    high = highcut / nyquist
    b, a = butter(order, [low, high], btype='band')
    return b, a

# Function to apply the filter to the signal
def butter_bandpass_filter(data, lowcut, highcut, fs, order=2):
    b, a = butter_bandpass(lowcut, highcut, fs, order)
    return filtfilt(b, a, data)

# Function to normalize the PPG signal to the range [0, 1]
```

```
def normalize_signal(data):
    return (data - np.min(data)) / (np.max(data) - np.min(data))

# Simulate raw PPG signal with noise
fs = 100  # Sampling rate (Hz)
t = np.arange(0, 10, 1/fs)  # Time vector (10 seconds)
clean_signal = 0.6 * np.sin(2 * np.pi * 1 * t) + 0.3 * np.sin(2 * np.pi * 0.05 * t)  #
Clean signal (1 Hz for HR + 0.05 Hz for baseline wander)
noisy_signal = clean_signal + np.random.normal(0, 0.2, len(t))  # Add
Gaussian noise to the clean signal

# Apply bandpass filter to the noisy signal (0.5 Hz to 5 Hz)
lowcut = 0.5
highcut = 5.0
filtered_signal = butter_bandpass_filter(noisy_signal, lowcut, highcut, fs)

# Normalize the filtered PPG signal
normalized_signal = normalize_signal(filtered_signal)

# Peak detection for heart rate calculation
peaks, _ = find_peaks(filtered_signal, distance=fs/2.5)  # Minimum distance
between peaks

# Plotting the results
plt.figure(figsize=(12, 10))

# Raw noisy PPG signal
plt.subplot(4, 1, 1)
plt.plot(t, noisy_signal, label="Raw PPG Signal (Noisy)")
plt.title("Raw PPG Signal with Noise")
plt.legend()

# Filtered PPG signal
plt.subplot(4, 1, 2)
plt.plot(t, filtered_signal, label="Filtered PPG Signal", color='g')
plt.title("Filtered PPG Signal")
plt.legend()

# Normalized PPG signal
plt.subplot(4, 1, 3)
plt.plot(t, normalized_signal, label="Normalized PPG Signal", color='orange')
```

```
plt.title("Normalized PPG Signal")
plt.legend()

# PPG signal with detected peaks
plt.subplot(4, 1, 4)
plt.plot(t, filtered_signal, label="Filtered PPG Signal", color='g')
plt.plot(t[peaks], filtered_signal[peaks], 'ro', label="Detected Peaks")
plt.title("Filtered PPG Signal with Detected Peaks")
plt.legend()

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

Output:





Results and Discussion:

1. Raw PPG Signal

The raw PPG signal is simulated as a combination of sinusoidal components and noise. The noisy signal contains significant low-frequency noise and random variations, which make it difficult to accurately detect heartbeats and estimate heart rate.

2. Filtered PPG Signal

The filtered signal, after applying the bandpass filter, removes much of the noise that lies outside the heart rate frequency range. The result is a smoother signal that more closely resembles the true PPG waveform, with much of the noise, including baseline wander and high-frequency interference, removed.

3. Normalized PPG Signal

Normalization of the filtered signal scales it to a range between 0 and 1. This is useful for signal comparison and ensures that the amplitude of the signal is not influenced by its original scale. While this step does not affect the overall signal shape or the detection of peaks, it standardizes the signal for easier processing and interpretation.

4.Detected Peaks

The peaks corresponding to heartbeats are detected in the filtered signal. The `find_peaks` function successfully identifies these peaks, which correspond to the locations in the signal where the heart rate is at its maximum. The detected peaks are overlaid on the filtered signal to visually assess the accuracy of peak detection.