Analysis of PPG Signals:

Filtering, Feature Extraction, and Peak Detection Theory

Photoplethysmogram (PPG) signals are a type of biosignal used to measure blood volume changes in the microvascular bed of tissue, primarily used to estimate heart rate and other cardiovascular parameters. They can be obtained using a variety of optical sensors, typically by measuring the light absorption or reflection in tissues. To analyze PPG signals, several processes such as filtering, feature extraction, and peak detection are critical for ensuring accurate interpretation. Let's break each of these down in theory:

1. Filtering of PPG Signals

PPG signals are often contaminated with noise from various sources, including motion artifacts, ambient light, and electronic noise. Filtering is therefore an essential preprocessing step to enhance the quality of the signal.

Types of Filters:

- **Low-pass filters:** PPG signals primarily contain frequencies in the range of 0.5–5 Hz, corresponding to the heart rate. To remove high-frequency noise, a low-pass filter is typically applied to eliminate components above 5 Hz, including power line interference (e.g., 50 or 60 Hz noise) or other high-frequency disturbances.
- **High-pass filters:** High-frequency components of the PPG signal often originate from motion artifacts. A high-pass filter is used to remove these low-frequency drifts or baseline shifts that can result from slow movements of the sensor or skin, usually around 0.5 Hz.
- **Band-pass filters:** Combining both high-pass and low-pass filters, a band-pass filter allows frequencies within a specific range (typically 0.5–5 Hz) to pass through while attenuating others. This filter is ideal for preserving the primary frequency range of the heart rate.
- **Notch filters:** These are used to specifically target and remove known interference frequencies such as 50 Hz or 60 Hz, caused by electrical power lines.

Goal of Filtering:

- To preserve the physiological information related to blood flow.
- To remove noise and artifacts that can distort the heart rate estimation.

2. Feature Extraction from PPG Signals

After filtering, the next step is to extract relevant features that can help characterize the PPG signal, especially for health and medical analysis. These features can be time-domain, frequency-domain, or time-frequency domain features.

Time-domain features:

- **Peak-to-peak interval:** The time between successive peaks in the PPG signal, which can be used to compute heart rate.
- **Pulse amplitude:** The magnitude of the PPG signal's peaks can give insights into the blood flow or the perfusion level.
- **Heart rate variability (HRV):** Variations in the time intervals between heartbeats (R-R intervals) which can be derived from the PPG peaks.

Frequency-domain features:

- **Power spectral density (PSD):** By performing a Fourier transform, the signal’s frequency components can be analyzed. Peaks in the PSD correspond to the frequency components of the heart rate, which typically lie between 0.5 and 5 Hz for most people.
- **Peak frequency:** This corresponds to the dominant frequency in the PPG signal and is often used to estimate the heart rate.

Time-frequency domain features:

- **Wavelet transform:** The continuous wavelet transform (CWT) is useful in capturing both the time and frequency characteristics of non-stationary signals like PPG. It helps detect transient events such as changes in pulse rate or amplitude.

Other features:

- **Autocorrelation:** Measuring the similarity between the PPG signal and a time-shifted version of itself can be useful in estimating periodicity.
- **Signal entropy:** Quantifies the complexity or randomness of the signal. High entropy can indicate a more chaotic signal, whereas low entropy may indicate more regularity or a stable rhythm.

3. Peak Detection in PPG Signals

Detecting peaks (or "detection of systolic peaks") is one of the most important tasks in analyzing PPG signals, as these peaks correspond to the pulses or beats of the heart. Accurate peak detection is critical for applications such as heart rate monitoring.

Theoretical Background:

- **Local maxima:** Peaks in the PPG signal correspond to local maxima, where the signal changes from increasing to decreasing values. Identifying these local maxima allows for the determination of heart rate and pulse characteristics.

Methods of Peak Detection:

- **Threshold-based methods:** Peaks can be detected when the signal exceeds a certain threshold. However, this is highly sensitive to noise and signal amplitude variations, so it’s used less frequently on its own.
- **Derivative methods:** The first derivative (rate of change) of the signal can be used to detect zero-crossings, where the signal changes direction. When the first derivative is positive, the signal is increasing, and when it's negative, the signal is decreasing. A peak is identified when the derivative changes sign from positive to negative.
- **Second derivative or curvature:** A more robust approach involves analyzing the second derivative of the signal. At a peak, the second derivative changes from positive to negative, indicating a local maximum.
- **Pan-Tompkins algorithm (for ECG, but adaptable):** Although initially developed for ECG, the Pan-Tompkins algorithm has been used in various biosignals for detecting peaks. It involves differentiating the signal, squaring it, and passing it through a moving average filter.
- **Wavelet-based peak detection:** This method can be more effective in handling noisy signals. By applying wavelet transforms, one can detect peaks by analyzing the scale and position of the wavelet coefficients.
- **Template matching:** Template matching techniques involve creating a model of a typical PPG pulse shape and then searching for matches or similarities to this template within the signal.

Challenges in Peak Detection:

- **Motion artifacts:** Rapid movements or vibrations can create spurious peaks in the signal.

- **Low signal-to-noise ratio (SNR):** Weak signals in noisy environments may make it difficult to detect peaks accurately.
- **Irregularities in pulse shapes:** In cases of arrhythmias or irregular heartbeats, the PPG signal may present irregular peaks, making detection challenging.

Refinement Techniques:

- **Peak validation:** Detected peaks can be validated based on their timing (e.g., ensuring that peaks are spaced appropriately for the heart rate) and amplitude to reduce false positives.
- **R-wave synchronization (for multimodal signals like ECG/PPG fusion):** Synchronizing detected PPG peaks with ECG R-waves can help improve peak detection and provide more reliable heart rate estimations.

Conclusion:

To sum up, the analysis of PPG signals involves the following core steps:

1. **Filtering** to remove noise and artifacts.
2. **Feature extraction** to quantify important physiological parameters.
3. **Peak detection** to identify heartbeats and measure metrics like heart rate.

By applying these techniques, PPG signals can be reliably analyzed for monitoring cardiovascular health, diagnosing conditions, or integrating into wearable devices.

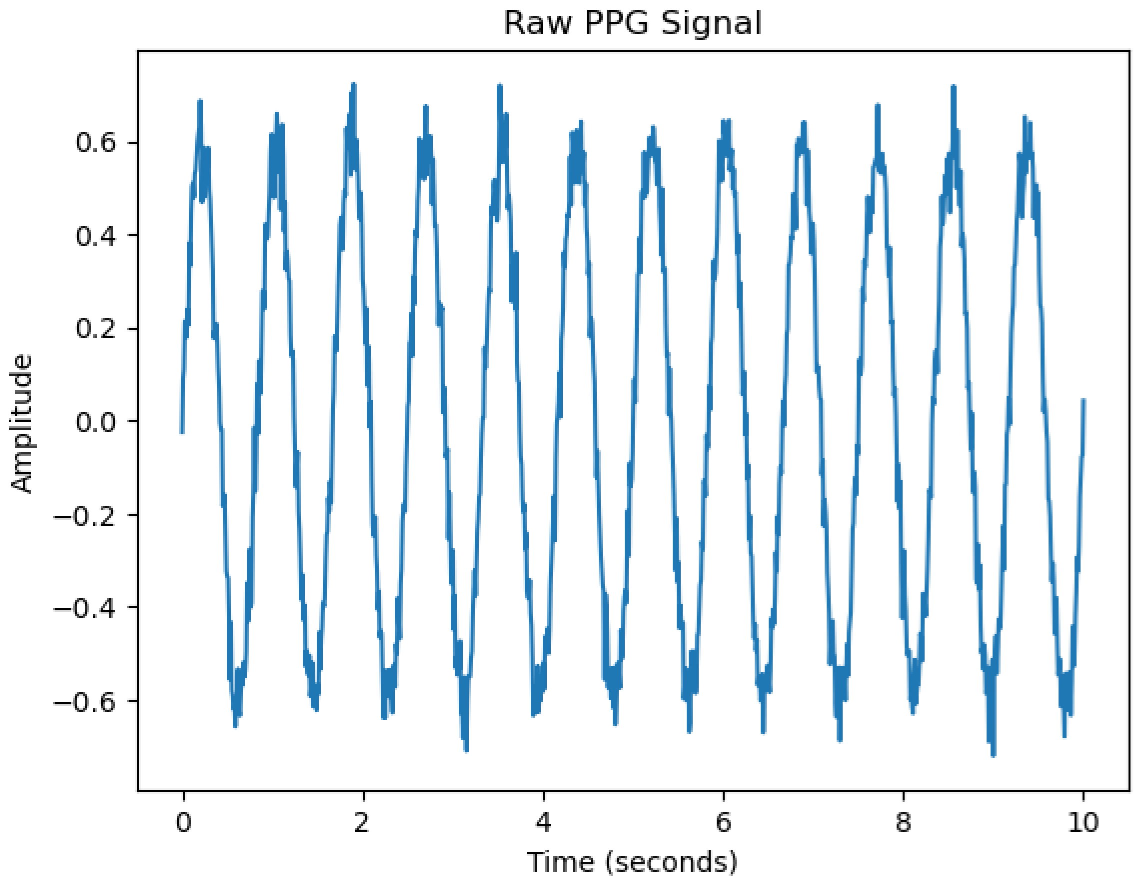
Sorce Code:

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

# Simulating a PPG signal (replace with actual data)
fs = 100  # Sampling rate (Hz)
t = np.linspace(0, 10, fs * 10)  # 10 seconds of data
ppg_signal = 0.6 * np.sin(2 * np.pi * 1.2 * t) + np.random.normal(0, 0.05,
len(t))

# Plotting the raw PPG signal
plt.plot(t, ppg_signal)
plt.title("Raw PPG Signal")
plt.xlabel("Time (seconds)")
plt.ylabel("Amplitude")
```

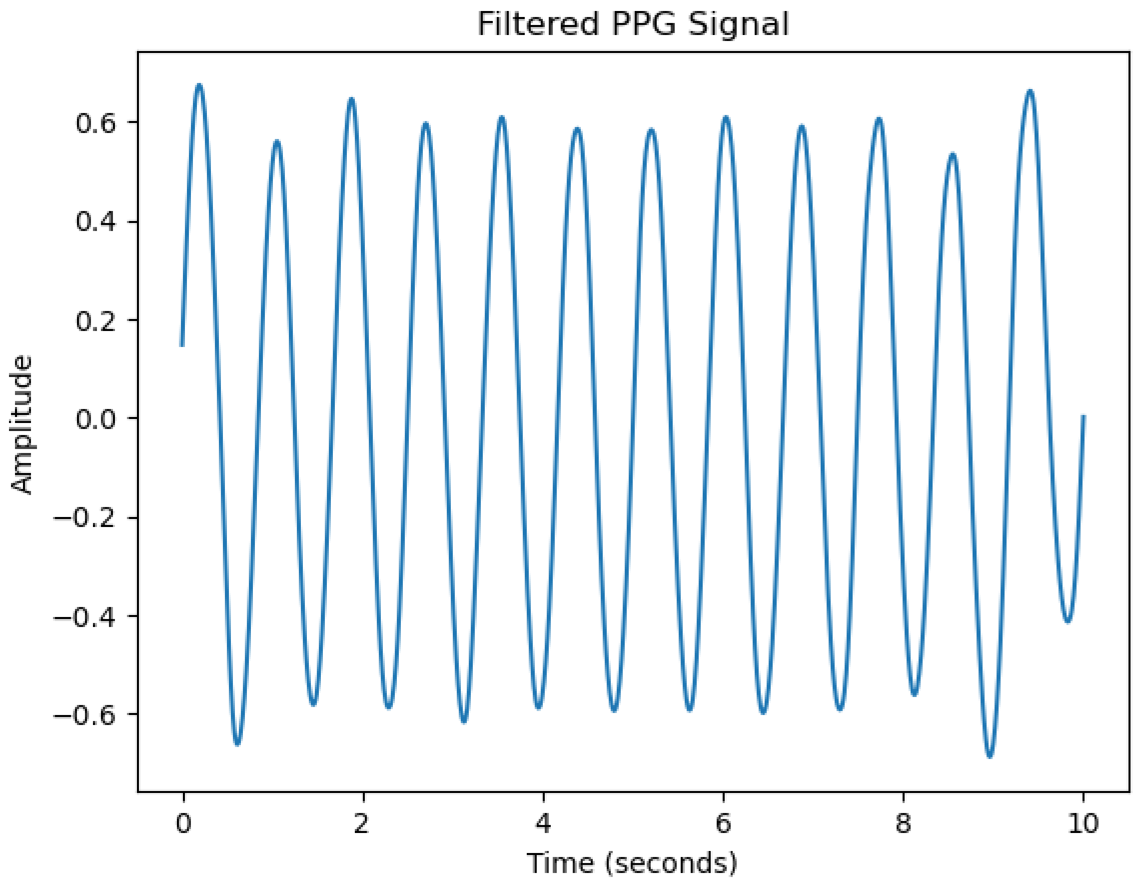
```
plt.show()
```



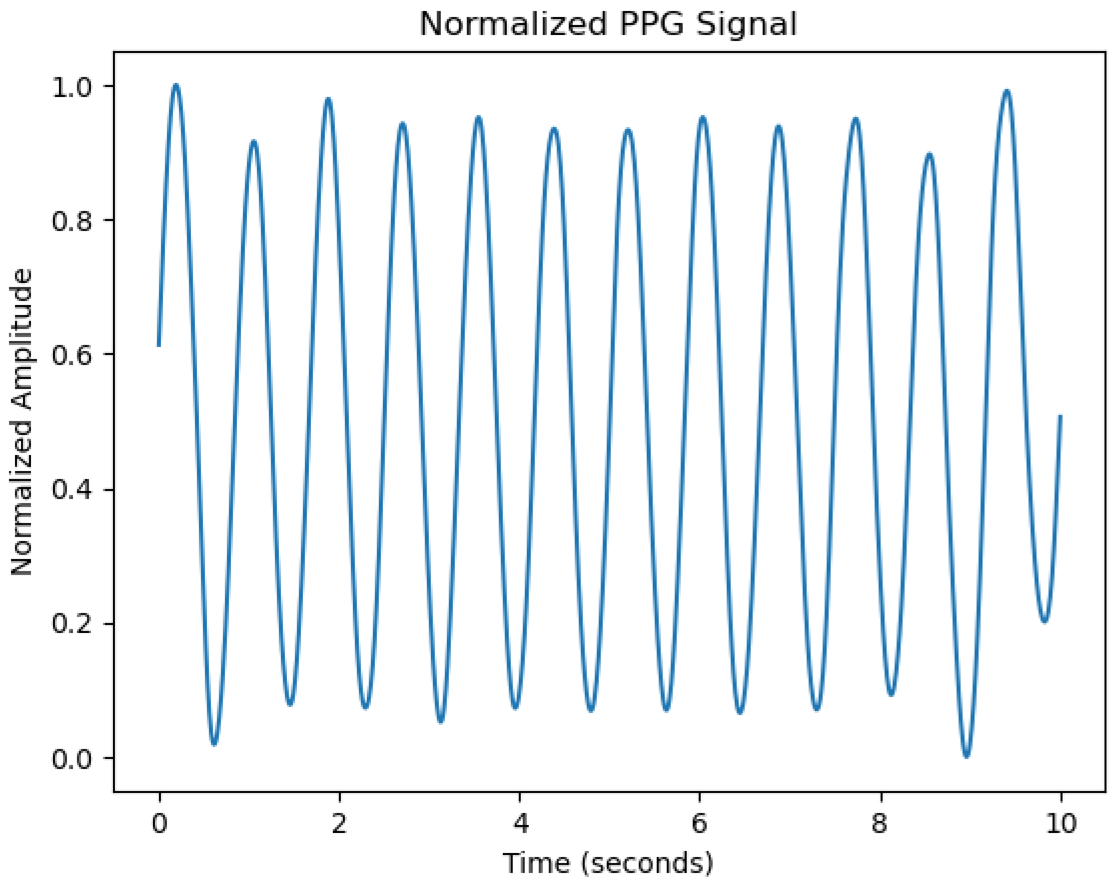

```
from scipy.signal import butter, filtfilt

# Bandpass filter design (0.5 to 5 Hz for heart rate detection)
def bandpass_filter(signal, lowcut, highcut, fs, order=4):
    nyquist = 0.5 * fs
    low = lowcut / nyquist
    high = highcut / nyquist
    b, a = butter(order, [low, high], btype='band')
    return filtfilt(b, a, signal)

# Filtered PPG signal
filtered_ppg = bandpass_filter(ppg_signal, 0.5, 5, fs)
plt.plot(t, filtered_ppg)
plt.title("Filtered PPG Signal")
plt.xlabel("Time (seconds)")
plt.ylabel("Amplitude")
plt.show()
```




```
normalized_ppg = (filtered_ppg - np.min(filtered_ppg)) / (np.max(filtered_ppg) -
np.min(filtered_ppg))
plt.plot(t, normalized_ppg)
plt.title("Normalized PPG Signal")
plt.xlabel("Time (seconds)")
plt.ylabel("Normalized Amplitude")
plt.show()
```



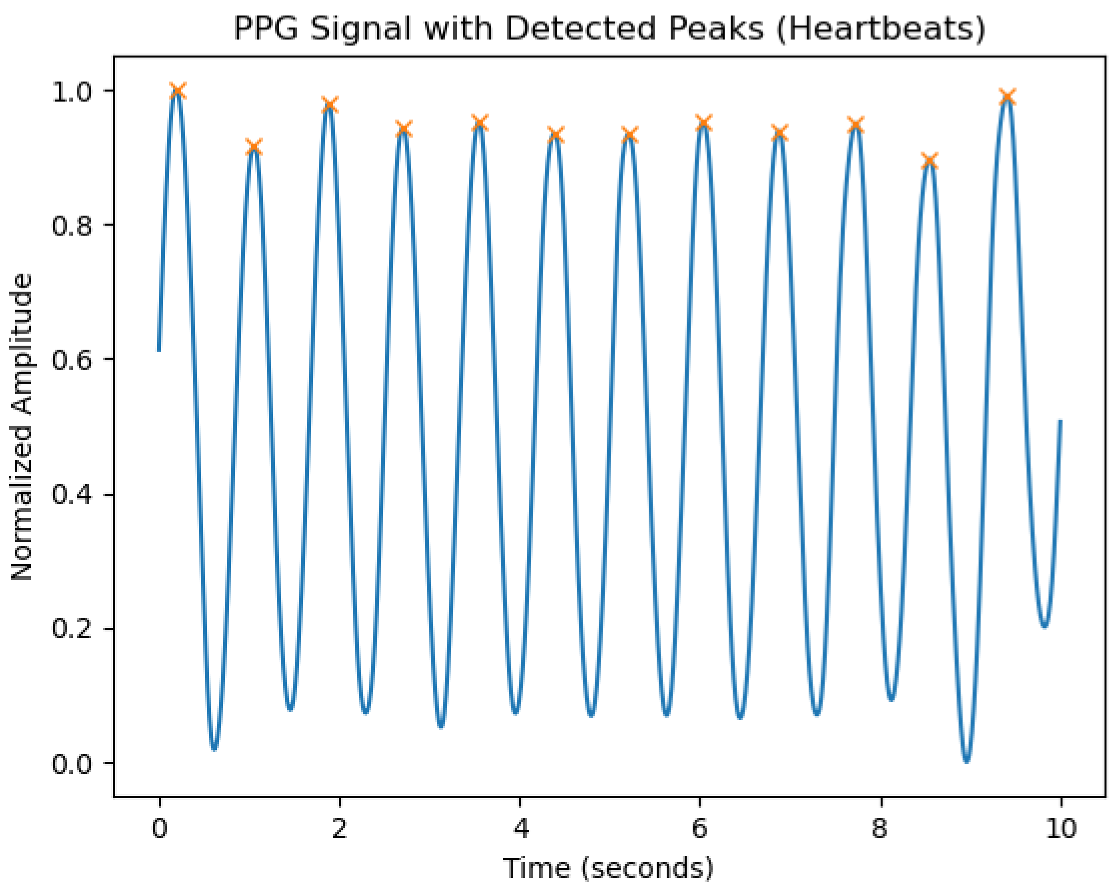

```python
from scipy.signal import find_peaks

# Detect peaks in the PPG signal
peaks, _ = find_peaks(normalized_ppg, distance=fs*0.6)  # Minimum distance of 0.6 seconds between peaks (for HR < 100 BPM)

# Calculate Heart Rate (BPM)
ibi = np.diff(peaks) / fs  # Inter-beat interval in seconds
heart_rate = 60 / ibi  # Convert to beats per minute (BPM)

# Plot the PPG signal with detected peaks
plt.plot(t, normalized_ppg)
plt.plot(t[peaks], normalized_ppg[peaks], "x")
plt.title("PPG Signal with Detected Peaks (Heartbeats)")
plt.xlabel("Time (seconds)")
plt.ylabel("Normalized Amplitude")
plt.show()

print("Heart Rate: ", np.mean(heart_rate), " BPM")
```




```
#Oxygen Saturation (SpO2) Calculation
# Assume red and infrared PPG signals (simulated)
red_ppg = 0.7 * np.sin(2 * np.pi * 1.2 * t) + np.random.normal(0, 0.05, len(t))
infrared_ppg = 0.6 * np.sin(2 * np.pi * 1.2 * t) + np.random.normal(0, 0.05,
len(t))

# SpO2 estimation (simplified version)
ratio = np.mean(red_ppg) / np.mean(infrared_ppg)
SpO2 = 110 - 25 * ratio  # Formula depends on device calibration

print("Estimated SpO2: ", SpO2, "%")

import matplotlib.pyplot as plt

# Simulated heart rate and SpO2 data
time = np.arange(0, len(heart_rate))  # Time index for heart rate
SpO2_values = np.random.normal(95, 1, len(time))  # Simulated SpO2 values

# Plotting heart rate and SpO2
plt.subplot(2, 1, 1)
```

```python
plt.plot(time, heart_rate, label='Heart Rate (BPM)')
plt.ylabel('BPM')
plt.title('Heart Rate and SpO2 Over Time')

plt.subplot(2, 1, 2)
plt.plot(time, SpO2_values, label='SpO2 (%)', color='red')
plt.ylabel('SpO2 (%)')
plt.xlabel('Time (seconds)')

plt.show()
```