

# Faculty of Engineering & Technology Department of Information and Communication Engineering Lab report

Course name: Signal and systems

Course Code\_: ICE - 2204.

Experiment name: Extracting abnormalities from raw PPG

signal.

# Submitted By:

### Siam Khan

Roll: 220620

Session: 2021-22

2<sup>nd</sup> Year 2<sup>nd</sup> Semester

Department of ICE,

PUST

# Submitted To:

#### Dr. Md. Imran Hossain

Associate Professor

Department of Information and

Communication Engineering,

Pabna University of Science and

Technology, Pabna

Date of Submission: 14.02.2025

## Theory:

#### 1. Raw PPG Signal:

- A PPG signal is a reflection of blood volume changes in the microvascular bed of tissue, typically captured via optical sensors (like in a pulse oximeter).
- The signal is usually in the form of a time series of light intensity, which corresponds to the cyclic pumping of blood by the heart.

#### 2. Preprocessing the Signal:

- Filtering: PPG signals often contain noise from movement, ambient light, or other sources. Filters like bandpass (0.5-5 Hz) or lowpass filters can remove these high-frequency artifacts.
- Normalization: The signal is typically normalized to a range to help reduce variability due to sensor differences.

#### 3. Feature Extraction:

- From the filtered PPG signal, features such as heart rate, heart rate variability (HRV), pulse rate, and signal morphology are extracted.
- Peak Detection: Peaks correspond to heartbeats. The intervals between these peaks can be analyzed to detect abnormal rhythms.

#### 4. Anomaly Detection:

- Heart Rate Analysis: An abnormality is detected if the heart rate is significantly higher or lower than expected (e.g., tachycardia or bradycardia).
- Frequency Domain Analysis: By applying a Fourier transform, anomalies in the frequency spectrum, such as changes in the power of certain frequency bands, can indicate potential issues.
- Machine Learning: A classifier can be trained on labeled data to identify abnormal heart rhythms, e.g., arrhythmias.

#### 5. Abnormality Detection Methods:

- Statistical Methods: Comparing features like average heart rate or HRV to thresholds.
- Pattern Recognition: Detecting unusual patterns in the waveform, such as premature beats or irregular rhythm.

# Sorce code: Given Below import numpy as np import matplotlib.pyplot as plt from scipy.signal import find\_peaks, butter, filtfilt def bandpass\_filter(ppg\_signal, lowcut=0.5, highcut=5.0, fs=100, order=4): nyquist = 0.5 \* fslow = lowcut / nyquist high = highcut / nyquist b, a = butter(order, [low, high], btype='band') filtered\_signal = filtfilt(b, a, ppg\_signal) return filtered signal def detect\_peaks(ppg\_signal, distance=50): peaks, \_ = find\_peaks(ppg\_signal, distance=distance) return peaks def calculate heart rate(peaks, fs):

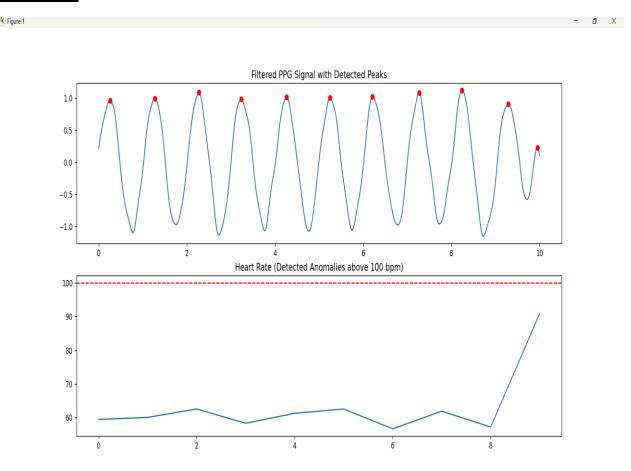
```
# Compute the time intervals between successive peaks (in
seconds)
  peak intervals = np.diff(peaks) / fs
  # Calculate heart rate (beats per minute)
  heart rate = 60 / peak intervals
  return heart rate
def detect anomalies(heart rate, threshold=100):
  abnormal heart rate = []
  for hr in heart rate:
    if hr > threshold: # For simplicity, consider HR > 100 as an
anomaly (tachycardia)
      abnormal heart rate.append(hr)
  return abnormal heart rate
# Sample PPG Signal (example data)
fs = 100 # Sampling frequency
time = np.linspace(0, 10, fs * 10) # 10 seconds of data
raw ppg signal = np.sin(2 * np.pi * 1 * time) + 0.2 *
np.random.randn(len(time)) # Sine wave with noise
# Apply the bandpass filter
filtered ppg signal = bandpass filter(raw ppg signal)
```

```
# Detect peaks (heartbeats)
peaks = detect_peaks(filtered_ppg_signal)
# Calculate heart rate from peaks
heart rate = calculate heart rate(peaks, fs)
# Detect anomalies based on a threshold
abnormal heart rate = detect anomalies(heart rate)
# Plotting the results
plt.figure(figsize=(12, 6))
plt.subplot(2, 1, 1)
plt.plot(time, filtered ppg signal)
plt.plot(time[peaks], filtered_ppg_signal[peaks], 'ro')
plt.title('Filtered PPG Signal with Detected Peaks')
plt.subplot(2, 1, 2)
plt.plot(heart rate)
plt.axhline(y=100, color='r', linestyle='--')
```

plt.title('Heart Rate (Detected Anomalies above 100 bpm)') plt.show()

print(f'Abnormal Heart Rate Episodes: {abnormal\_heart\_rate}')

# **Output:**



**☆ ← → | +** Q **=** | B

(x, y) = (1.880, 0.575)