Department of
Information & Communication Engineering

Course Title: Signals and Systems
Course Code: ICE-2204

Lab Name: Extracting abnormality from raw PPG signal.

Submitted To:
Dr. Md. Imran Hossain
Associate Professor
Department of ICE
Pabna University of Science &Technology

Submitted By:
Mst. Tahomina Kabir Rany
Roll: 210642
Session: 2021-22
2nd Year 2nd Semester
Dept. of Information & Communication Engineering
Pabna University of science & Technology

Date of Submission: 14.02.2025

Introduction to PPG (Photoplethysmography)

Photoplethysmography (PPG) is an optical technique used to measure blood volume changes in the microvascular tissue. It is widely employed in wearable health devices and medical monitoring systems to track heart rate (HR), blood oxygen levels (SpO_2), and other cardiovascular parameters.

PPG signals are acquired using light-emitting diodes (LEDs) and photodetectors, which detect changes in light absorption due to pulsatile blood flow.

Extracting Raw PPG Signals

The process of obtaining a raw PPG signal involves several key steps:

Hardware Components

- **Light Source (LEDs):** Typically, red, green, or infrared (IR) LEDs are used. Green light is often preferred for wrist-based devices due to better penetration in blood-rich tissues.
- **Photodetector:** Measures the intensity of reflected or transmitted light, capturing pulsatile variations in blood volume.

Signal Acquisition

- The raw signal is an optical intensity waveform representing blood volume changes.
- Typically, a sample rate of **100–1000 Hz** is used for high-resolution data.

Preprocessing of PPG Signals

Since raw PPG signals contain noise and artifacts, preprocessing is crucial:

Signal Filtering:

    - **Low-pass filtering** (e.g., Butterworth, moving average) to remove high-frequency noise.
    - **High-pass filtering** to eliminate baseline drift.
    - **Notch filtering** to remove power line interference (50/60 Hz).

Motion Artifact Removal:

    - Adaptive filtering or signal decomposition techniques like **wavelet transform, Empirical Mode Decomposition (EMD)**.
    - **Accelerometer-based correction** for wearable devices.

Normalization & Smoothing:

- Normalization to scale the signal amplitude.
- Smoothing using **Savitzky-Golay filters** or median filtering.

Detecting Abnormalities in PPG Signals

PPG signals can be analyzed to detect cardiovascular abnormalities using various signal processing and machine learning techniques.

Heart Rate and Heart Rate Variability (HRV) Analysis

- Extracting peaks from the PPG waveform provides heart rate (HR).
- HRV analysis (time and frequency domain) helps detect arrhythmias and autonomic dysfunction.

Abnormality Detection Techniques

1. **Arrhythmia Detection:**
    - Irregular peak intervals indicate atrial fibrillation (AF) or bradycardia.
    - Machine learning classifiers (e.g., Random Forest, CNNs) can detect arrhythmias from PPG waveforms.
2. **Blood Pressure Estimation:**
    - Pulse Transit Time (PTT) and machine learning models estimate systolic and diastolic BP.
3. **SpO_2 (Oxygen Saturation) Estimation:**
    - Dual-wavelength PPG (red & IR) is used.
    - Abnormally low SpO_2 (<90%) may indicate hypoxia or respiratory distress.
4. **Respiratory Rate (RR) Estimation:**

    - Respiratory-induced modulations in PPG amplitude and frequency are analyzed.
5. **Detecting Shock or Hypovolemia:**
    - Decreased amplitude and variations in PPG signal morphology may indicate shock or hypovolemia.

Machine Learning and AI in PPG Abnormality Detection

With advancements in AI, deep learning-based approaches improve accuracy:

- **Convolutional Neural Networks (CNNs):** Extract features from PPG signals.
- **Recurrent Neural Networks (RNNs, LSTMs):** Capture temporal dependencies in time-series data.

<u>Source Code:</u> Given below

```
import numpy as np

import scipy.signal as signal

import matplotlib.pyplot as plt

from scipy.signal import find_peaks, butter, filtfilt


# Generate synthetic PPG signal (replace this with your raw PPG data)

def generate_synthetic_ppg(length=1000, sampling_rate=100):

   t = np.linspace(0, length / sampling_rate, length)

   ppg = 0.5 * np.sin(2 * np.pi * 1.2 * t) + 0.05 *
np.random.randn(length)

   return t, ppg
```

```
# Bandpass filter
def bandpass_filter(data, lowcut, highcut, fs, order=4):
    nyquist = 0.5 * fs
    low = lowcut / nyquist
    high = highcut / nyquist
    b, a = butter(order, [low, high], btype="band")
    filtered_data = filtfilt(b, a, data)
    return filtered_data

# Detect peaks (heartbeats)
def detect_peaks(ppg, sampling_rate, distance=50):
    peaks, _ = find_peaks(ppg, distance=distance)
    return peaks

# Calculate heart rate variability (HRV)
def calculate_hrv(peaks, sampling_rate):
    rr_intervals = np.diff(peaks) / sampling_rate  # Time between
consecutive peaks
    mean_rr = np.mean(rr_intervals)
    std_rr = np.std(rr_intervals)
    return rr_intervals, mean_rr, std_rr
```

```
# Abnormality detection based on HRV thresholds
def detect_abnormalities(rr_intervals, mean_rr, std_rr, threshold=0.2):
    abnormalities = []
    for i, rr in enumerate(rr_intervals):
        if abs(rr - mean_rr) > threshold * std_rr:
            abnormalities.append(i)
    return abnormalities

# Main function
def main():
    # Parameters
    sampling_rate = 100  # Hz
    lowcut = 0.5       # Hz
    highcut = 5.0      # Hz
    threshold = 0.2    # Abnormality detection threshold

    # Generate or load PPG data
    t, raw_ppg = generate_synthetic_ppg(length=3000,
sampling_rate=sampling_rate)

    # Preprocess PPG data
```

```
    filtered_ppg = bandpass_filter(raw_ppg, lowcut, highcut,
sampling_rate)

    # Detect peaks
    peaks = detect_peaks(filtered_ppg, sampling_rate, distance=int(0.6 *
sampling_rate))  # Min distance ~600ms

    # Calculate HRV
    rr_intervals, mean_rr, std_rr = calculate_hrv(peaks, sampling_rate)

    # Detect abnormalities
    abnormalities = detect_abnormalities(rr_intervals, mean_rr, std_rr,
threshold=threshold)

    # Plot results
    plt.figure(figsize=(12, 6))
    plt.plot(t, filtered_ppg, label="Filtered PPG")
    plt.scatter(t[peaks], filtered_ppg[peaks], color="red", label="Detected
Peaks")
    plt.title("PPG Signal with Detected Peaks")
    plt.xlabel("Time (s)")
    plt.ylabel("Amplitude")
    plt.legend()
```

```
    plt.grid()

    plt.figure(figsize=(12, 6))
    plt.plot(rr_intervals, label="RR Intervals (s)")
    plt.scatter(abnormalities, np.array(rr_intervals)[abnormalities],
color="red", label="Abnormal Intervals")
    plt.title("RR Intervals with Abnormalities")
    plt.xlabel("Interval Index")
    plt.ylabel("RR Interval (s)")
    plt.legend()
    plt.grid()

    plt.show()

    # Print summary
    print(f"Mean RR Interval: {mean_rr:.2f} s")
    print(f"Standard Deviation of RR Intervals: {std_rr:.2f} s")
    print(f"Number of Abnormalities Detected: {len(abnormalities)}")

if __name__ == "__main__":
    main()
```

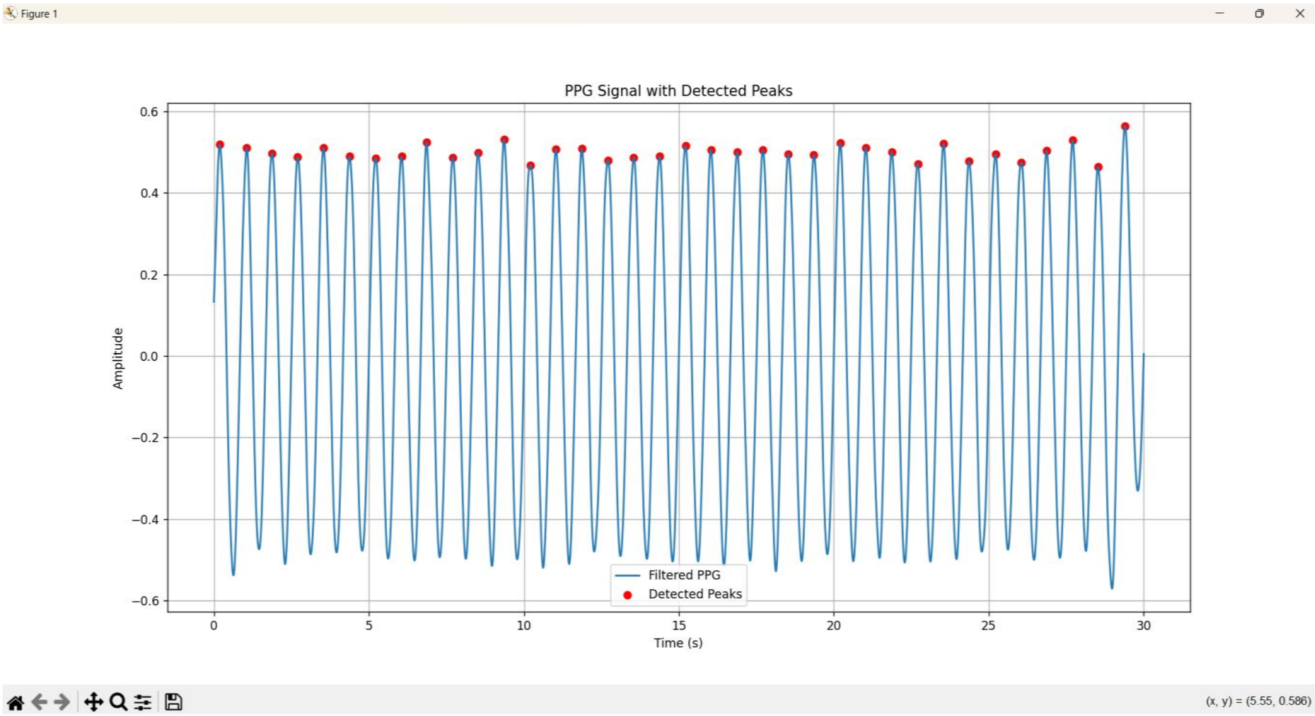
Figure 1
PPG Signal with Detected Peaks
Amplitude
Time (s)
Filtered PPG
Detected Peaks
(x, y) = (5.55, 0.586)

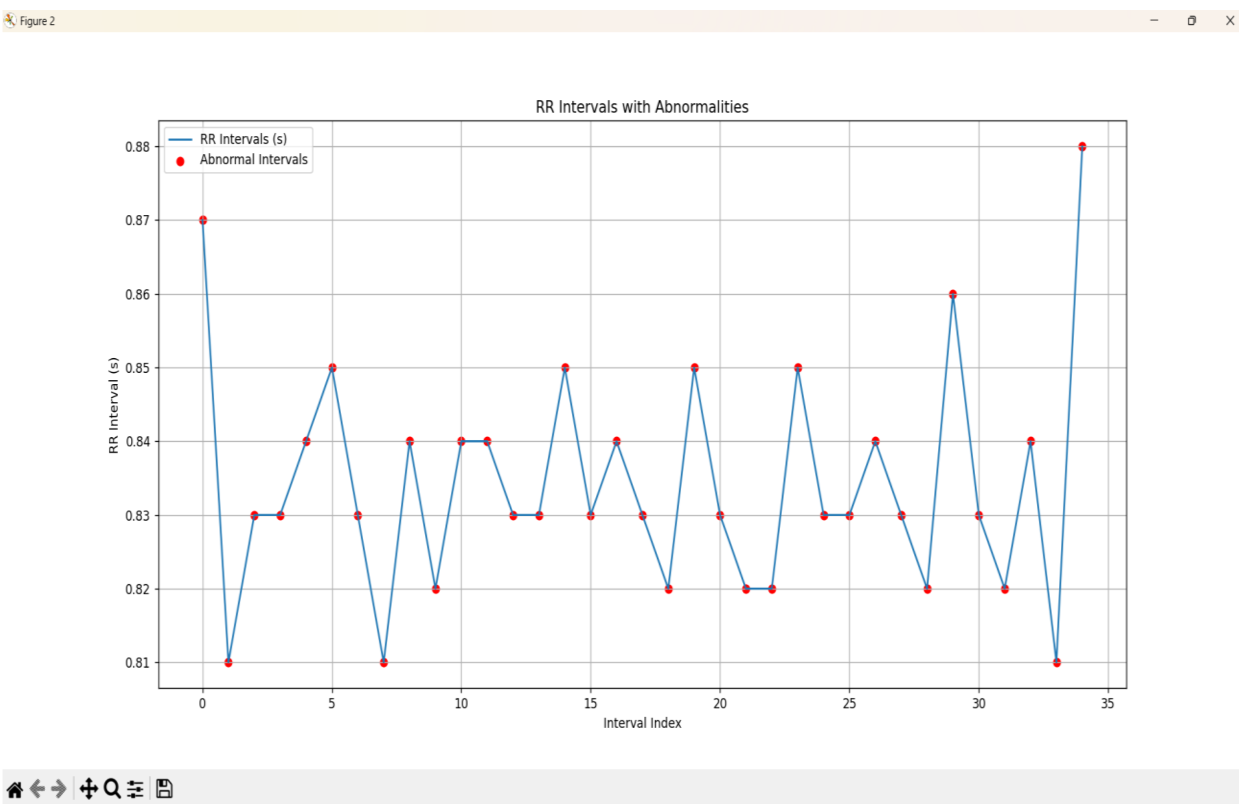
Figure 2
RR Intervals with Abnormalities
RR Intervals (s)
Abnormal Intervals
RR Interval (s)
Interval Index