TABLE OF CONTENTS

1: Introduction and Principle of Operation.....................................

2: Data Sampling...

3: Objectives ....

4: Schematic Diagram…………………………………………

5: Implementation..

6: Code……………………………………………………

7: Flow Of Control……………………………………………………

8: Conclusion…………………………………………

INTRODUCTION

In the realm of healthcare and biomedicine, the advent of advanced sensor technologies has revolutionized the way we monitor and manage individual health metrics. Biomedical signal monitoring systems have become crucial tools in providing real-time insights into physiological parameters, enabling early detection of health issues and personalized healthcare management. This project aims to develop a comprehensive biomedical signal monitoring system using MATLAB, focusing specifically on the real-time monitoring of heart rate and body temperature.

Importance of Biomedical Signal Monitoring

The continuous monitoring of physiological signals such as heart rate and body temperature is essential for several reasons. Heart rate variability (HRV), for instance, not only indicates the overall cardiovascular health but also serves as a marker for stress levels and autonomic nervous system activity. Body temperature monitoring, on the other hand, is fundamental for detecting fever or hypothermia, which are critical indicators of various health conditions ranging from infections to metabolic disorders.

System Components and Overview

The proposed system integrates several key components to achieve its objectives:

1. **Arduino Uno**: Serving as the microcontroller, the Arduino Uno facilitates signal acquisition and initial processing from connected sensors. It acts as a bridge between the physical world (sensor data) and the digital realm (computer processing).
2. **ECG Sensor (AD8232)**: The ECG sensor captures the electrical activity of the heart, allowing the calculation of heart rate and potentially detecting abnormal heart rhythms or conditions.
3. **GSR Sensor**: The Galvanic Skin Response (GSR) sensor measures changes in skin conductivity, providing insights into stress levels and emotional arousal. This sensor aids in understanding the emotional and physiological responses under different conditions.
4. **Temperature Sensor (DS18B20)**: The DS18B20 sensor accurately measures body temperature, crucial for detecting fever or hypothermia, which are indicative of various health conditions.
5. **Bluetooth Module HC-05**: Enables wireless communication between the monitoring system and external devices such as smartphones or computers. This feature enhances mobility and accessibility of the monitoring system.
6. **MATLAB**: MATLAB serves as the primary platform for signal processing, data analysis, and graphical user interface (GUI) development. It leverages its Signal Processing Toolbox for advanced signal analysis and the Instrument Control Toolbox for communication with the Arduino.

Project Objectives

The main objectives of this project are:

- **Real-time Signal Acquisition**: Develop mechanisms to acquire continuous data from ECG, GSR, and temperature sensors in real-time.
- **Signal Processing and Analysis**: Implement algorithms in MATLAB to process and analyze acquired signals. This includes filtering, feature extraction (such as heart rate computation from ECG signals), and anomaly detection.
- **Health Insights and Monitoring**: Provide real-time feedback on heart rate variability, stress levels (via GSR), and body temperature trends. This information helps in monitoring overall health status and detecting abnormalities early.
- **Data Correlation and Visualization**: Utilize databases and external sources to correlate the monitored health parameters with known health conditions or physiological responses. Visualizations and reports generated from MATLAB aid in interpreting and presenting these correlations effectively.

Significance and Innovation

This project contributes to the field of biomedical engineering by demonstrating a practical implementation of a comprehensive monitoring system using readily available components and MATLAB software. It combines hardware integration with advanced signal processing techniques, making it a valuable tool for healthcare professionals, researchers, and individuals interested in personal health monitoring.

In summary, the biomedical signal monitoring system presented in this project integrates cutting-edge sensor technology with powerful computational tools to enable real-time monitoring of critical health parameters. Through MATLAB, it provides sophisticated data analysis and visualization capabilities, paving the way for enhanced healthcare management and proactive health monitoring strategies.

DATA SAMPLING

We use Bootstrapping which is a method to estimate the distribution of a statistic (e.g., mean, median) by resampling with replacement from the original data. This technique helps in estimating the accuracy (e.g.-> standard error, confidence intervals) of a statistic without making strong parametric assumptions.

1.Resampling: Randomly draw samples with replacement from the original dataset.

2. Statistic Calculation: Compute the statistic of interest (e.g., mean) for each resampled dataset.

3. Repetition: Repeat the resampling process many times to create a distribution of the statistic.

4. Estimation: Use the distribution to estimate properties like standard error or confidence intervals.

• Confidence Interval
Purpose: A confidence interval provides a range of values within which the true population parameter is expected to lie, with a certain level of confidence (e.g., 95%).

OBJECTIVES

- Design and implement a comprehensive biomedical signal monitoring system using MATLAB

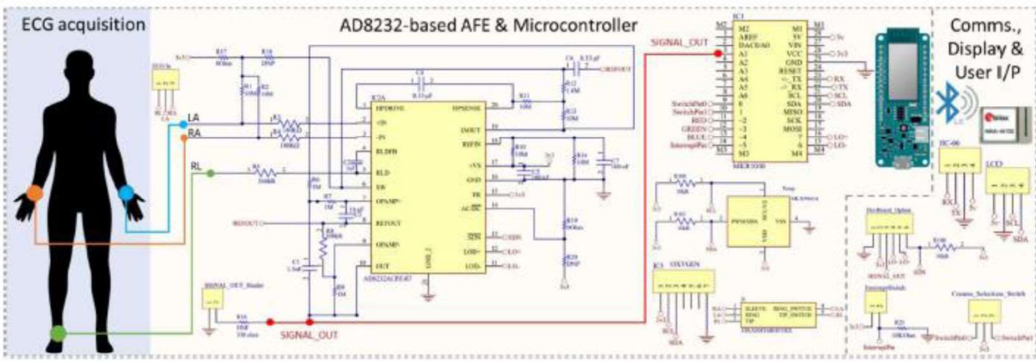
- focusing on real-time monitoring of heart rate and body temperature.

- The system should be capable of acquiring, processing, and analyzing signals from sensors, providing valuable insights into the individual's health status. While also analyzing databases from the internet to correlate

SCHEMATIC DIAGRAM

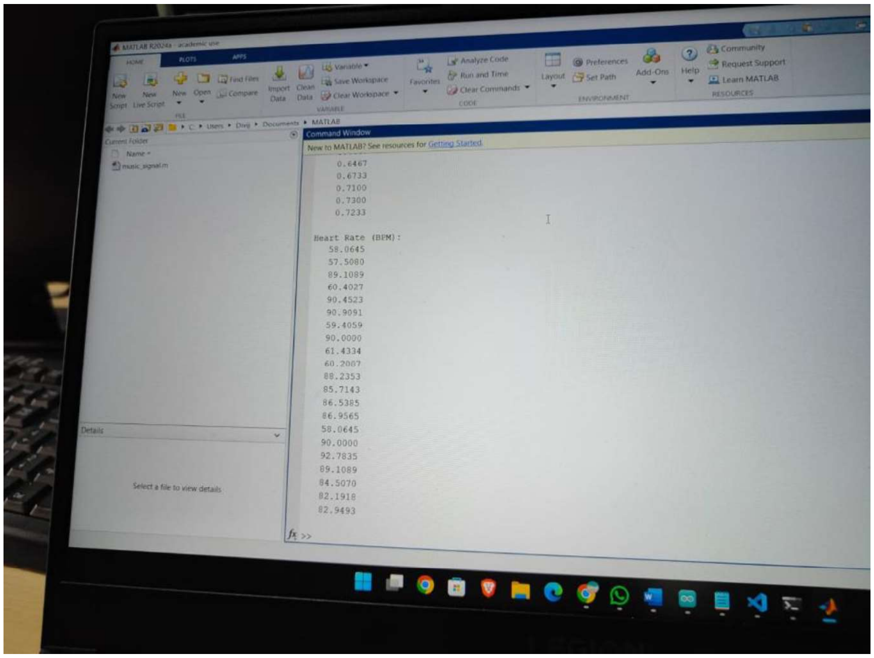
PHASE II

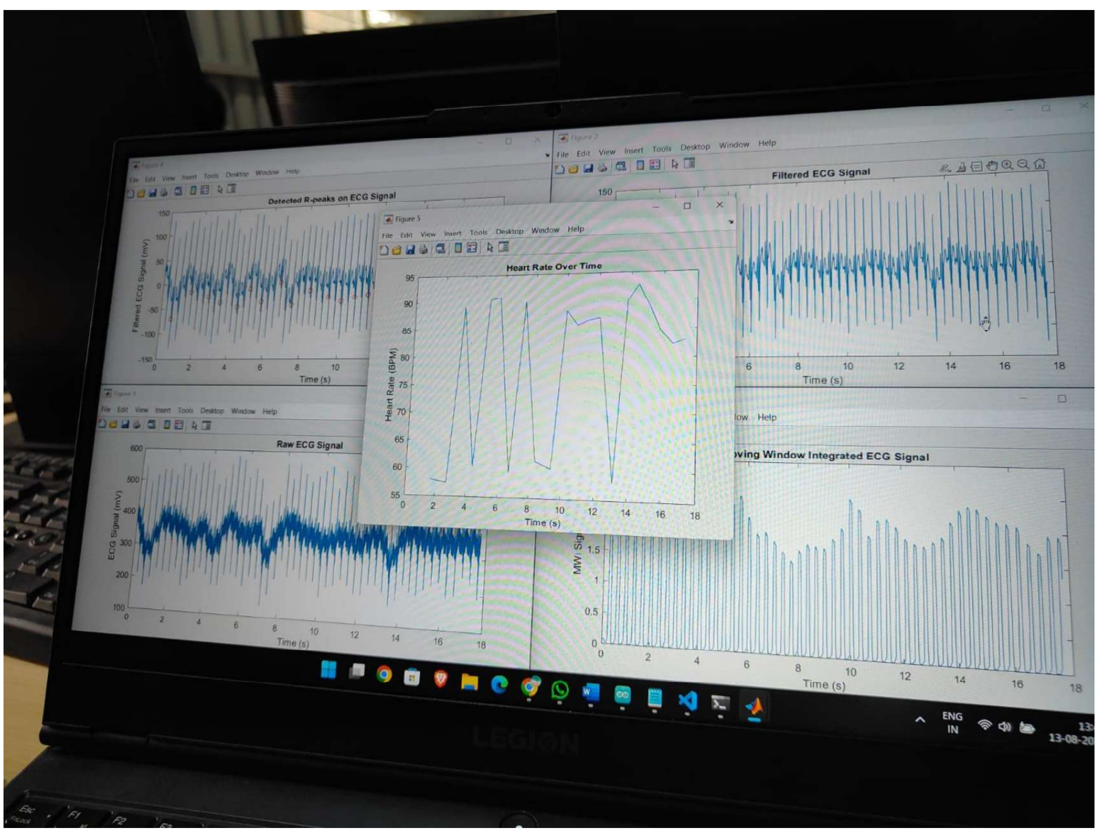
• FOR PHASE WE'LL DEVELOP A FULLY WORKING MODEL OF THE BIOMEDICAL SIGNAL MONITORING SYSTEM AND ALSO MATCH AND CORRELATE THE AQUIRED DATA TO IDENTIFY CONDITIONS VIA DATABASE PROVIDED BY WHO AND HEALTH MINISTRY ON THE INTERNET .



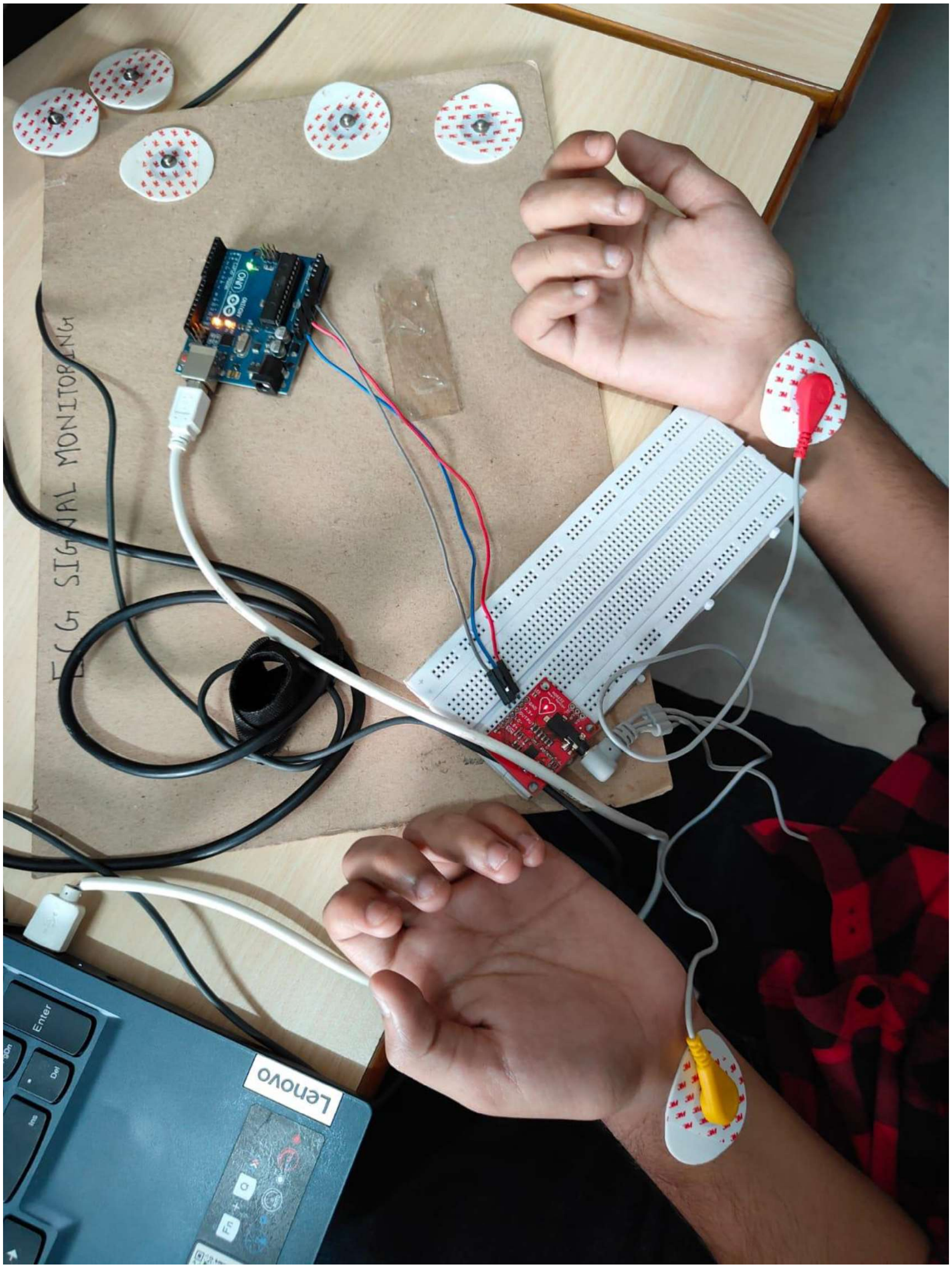
Sunday, 14 July 2024 RV COLLEGE OF ENGINEERING 10

IMPLEMENTATION





- **Top Left Plot**
    - This plot shows the ECG signal with the R-peaks detected and highlighted. The R-peaks are the prominent upward spikes in the ECG waveform and are crucial for determining heart rate.
- **Top Right Plot**
    - This plot displays a filtered version of the raw ECG signal. Filtering is typically done to remove noise and artifacts from the ECG signal, allowing for clearer analysis of the heart's electrical activity.
- **Center Plot**
    - This plot represents the heart rate over time, typically calculated by measuring the time intervals between consecutive R-peaks in the ECG signal. The y-axis likely represents the heart rate in beats per minute (BPM), while the x-axis represents time.
- **Bottom Left Plot**
    - This is the raw ECG signal before any processing. It shows the electrical activity of the heart as recorded by the ECG device, with possible noise and other artifacts.
- **Bottom Right Plot**
    - This plot seems to show an integrated ECG signal using a sliding window approach. This technique is often used to smooth the ECG signal or to calculate a moving average, providing a more stable view of the signal over time.


ECG SIGNAL MONITORING
Lenovo

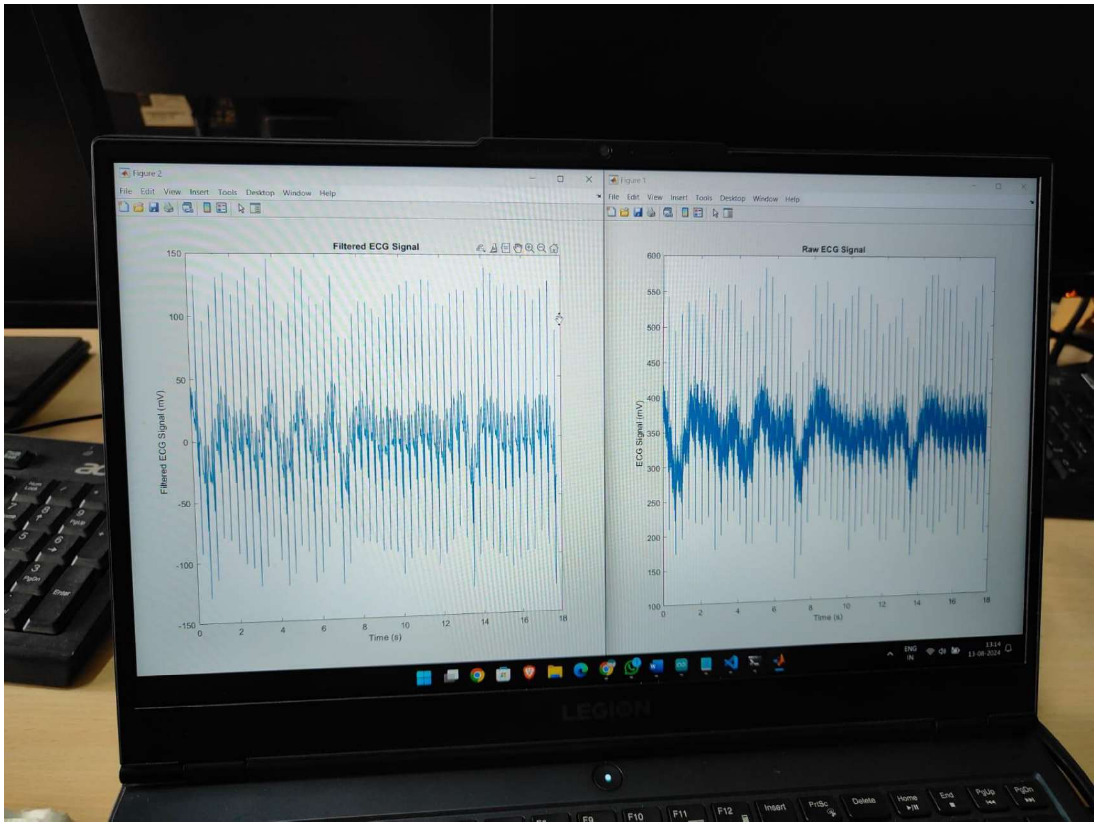


Figure 2
File Edit View Insert Tools Desktop Window Help
Filtered ECG Signal
Filtered ECG Signal (mV)
Time (s)
Figure 1
Raw ECG Signal
ECG Signal (mV)
Time (s)
LEGION

CODE

```
ecg_data = load('ecg_data.txt');
fs = 300;

t = (0:length(ecg_data)-1) / fs;

figure;
plot(t, ecg_data);
xlabel('Time (s)');
ylabel('ECG Signal (mV)');
title('Raw ECG Signal');

[b, a] = butter(2, [0.5 40] / (fs / 2), 'bandpass');
filtered_ecg = filtfilt(b, a, ecg_data);

figure;
plot(t, filtered_ecg);
xlabel('Time (s)');
ylabel('Filtered ECG Signal (mV)');
title('Filtered ECG Signal');

diff_ecg = diff(filtered_ecg);
squared_ecg = diff_ecg .^ 2;

window_size = round(0.12 * fs);
mwi_ecg = movsum(squared_ecg, window_size);

figure;
plot(t(2:end), mwi_ecg);
xlabel('Time (s)');
ylabel('MWI Signal');
title('Moving Window Integrated ECG Signal');

min_peak_distance = round(0.6 * fs);
threshold = 0.1;
[~, r_peaks] = findpeaks(mwi_ecg, 'MinPeakHeight', threshold, 'MinPeakDistance', min_peak_distance);

figure;
plot(t, filtered_ecg);
hold on;
plot(r_peaks / fs, filtered_ecg(r_peaks), 'ro');
xlabel('Time (s)');
ylabel('Filtered ECG Signal (mV)');
title('Detected R-peaks on ECG Signal');

rr_intervals = diff(r_peaks) / fs;
heart_rate = 60 ./ rr_intervals;

figure;
plot(r_peaks(2:end) / fs, heart_rate);
xlabel('Time (s)');
ylabel('Heart Rate (BPM)');
title('Heart Rate Over Time');

disp('Detected R-peaks at indices:');
disp(r_peaks);

disp('RR intervals (in seconds):');
disp(rr_intervals);

disp('Heart Rate (BPM):');
disp(heart_rate);
```

FLOW OF CONTROL

1. Placement of ECG Pads:

- **Electrode Positioning:** ECG pads are typically placed on the chest, arms, and legs of a patient. The standard 12-lead ECG setup uses 10 electrodes: 4 limb electrodes (one on each arm and leg) and 6 chest electrodes (placed in specific locations around the heart).
- **Electrical Contact:** The pads contain a conductive gel that helps create a good electrical contact between the skin and the electrodes, ensuring that the heart's electrical signals can be captured effectively.

2. Capture of Electrical Signals:

- **Heart's Electrical Activity:** The heart generates electrical impulses that control the contraction and relaxation of the heart muscles. These impulses are created by the sinoatrial (SA) node, which acts as the heart's natural pacemaker, and they propagate through the heart muscle, causing it to contract in a coordinated manner.
- **Signal Detection:** The electrodes (ECG pads) detect the small changes in voltage on the skin that occur due to the electrical activity of the heart. These voltages are very small, typically in the range of microvolts (µV) to millivolts (mV).

3. Generation of the ECG Signal:

- **Signal Amplification:** The electrical signals detected by the ECG pads are very weak, so they are amplified by an ECG machine to make them suitable for recording and analysis.
- **Recording the Signal:** The amplified signals are recorded over time, producing a waveform that represents the electrical activity of the heart. This waveform is the ECG signal.
- **Leads and Views:** Each pair of electrodes produces a different view (or lead) of the heart's electrical activity. A standard 12-lead ECG provides 12 different views of the heart, each showing the electrical activity from a different angle.

4. Key Components of the ECG Signal:

- **P Wave:** Represents the electrical activity associated with the atrial depolarization (the contraction of the atria).
- **QRS Complex:** Represents the rapid depolarization of the ventricles, leading to their contraction. The R-peak is the most prominent feature of this complex and is crucial for determining heart rate.
- **T Wave:** Represents the repolarization of the ventricles (the relaxation phase).

5. Processing the ECG Signal in MATLAB:

- **Raw Signal:** The raw ECG signal, as seen in one of the plots, is the direct output from the ECG machine before any processing. It includes the P wave, QRS complex, and T wave, along with any noise or artifacts.
- **Filtering:** The raw signal is often noisy due to muscle activity, electrical interference, or movement. MATLAB can apply filters to remove this noise, resulting in a cleaner signal that focuses on the relevant heart activity.
- **R-Peak Detection:** The R-peaks, which are the most prominent points in the QRS complex, are detected using algorithms in MATLAB. These peaks are critical for calculating the heart rate.
- **Heart Rate Calculation:** The time intervals between consecutive R-peaks are used to calculate the heart rate, typically expressed in beats per minute (BPM).

6. MATLAB Analysis:

- MATLAB can be used to visualize, filter, and analyze the ECG data. The software can automatically detect features like R-peaks and calculate metrics such as heart rate over time. MATLAB's robust signal processing capabilities make it an ideal tool for analyzing ECG data and extracting meaningful information about heart health.

CONCLUSION

The process of recording and analyzing ECG (Electrocardiogram) signals is a cornerstone of modern cardiology, providing critical insights into the electrical activity of the heart. The journey begins with the placement of ECG pads on the skin, carefully positioned to capture the heart's electrical impulses from multiple angles. These impulses, originating from the heart's natural pacemaker, the sinoatrial (SA) node, orchestrate the rhythmic contraction and relaxation of the heart muscles. As these electrical signals travel through the heart, they generate small voltage changes on the skin's surface, which are detected by the electrodes. These raw electrical signals are typically very weak, often measured in microvolts (μV) to millivolts (mV). Therefore, they require amplification to be recorded and analyzed effectively. The amplified signals are then displayed as a continuous waveform, known as the ECG signal, which visually represents the electrical activity occurring within the heart over time. The ECG waveform includes several key components: the P wave, QRS complex, and T wave, each corresponding to different phases of the heart's electrical cycle.

However, the raw ECG signal is not without its challenges. It is often contaminated by noise and artifacts, which can obscure the true electrical activity of the heart. Sources of noise can include muscle activity, movement, and external electrical interference. To address this, signal processing techniques are employed, often using specialized software like MATLAB. MATLAB provides powerful tools for filtering the raw ECG signal, removing unwanted noise, and enhancing the clarity of the waveform. In conclusion, the process of recording, processing, and analyzing ECG signals is a testament to the advances in medical technology and computational analysis. From the initial capture of the heart's electrical activity using simple ECG pads to the complex analysis performed by MATLAB, every step is crucial in ensuring accurate and meaningful interpretations of heart health. This technology not only aids in diagnosing and treating heart conditions but also pushes the boundaries of what is possible in preventive and personalized medicine. As technology continues to evolve, the integration of ECG analysis with other diagnostic tools and the development of more advanced algorithms will likely lead to even greater breakthroughs in cardiology, ensuring that healthcare providers can offer the best possible care to their patients.