ECG Measurement and Analysis using Raspberry Pi

Sam Luby
School of Electrical andElectronic Engineering
University College Dublin
Dublin, Ireland
sam.luby@ucdconnect.ie

***Abstract*—This paper presents a method of collecting and analysing electrocardiograph (ECG) data from a patient in order to perform heart rate variability analysis. This methodology focuses on a low-cost implementation for heart monitoring and is based on the Pan Tomkins algorithm to perform feature extraction of the QRS complex of an ECG signal. This information is then used for analysis, particularly in the area of heart rate variability (HRV).**

***Index Terms*— electrocardiography, ECG, heart rate analysis, heart rate variability, HRV, raspberry pi**

I. Introduction

Cardiovascular diseases combined are the number one cause of death worldwide, representing approximately 17 million mortalities globally or 32% of all registered deaths [1]. There is a disproportionality here in that of these 17 million, more than 70% are in low or middle-income countries such as parts of Asia, Eastern Europe and North Africa [2]. It is clear that the less wealthy are more at risk because the majority have no access to proper healthcare.

Evidently, current implementations for monitoring and treatment are not enough. According to the Irish Heart Foundation, heart failure in Ireland is set to dramatically increase. Without proper healthcare treatment, heart failure can lead to cardiac arrest which can cause permanent brain damage after just 5 minutes, eventually leading to total shutdown of the bodys vital organs and eventual death. Regular monitoring and early diagnoses can help to expose underlying lingering issues in the circulatory system.

The problem this paper aims to address is how to properly and easily monitor a patients circulatory system to tackle the increasing issue of heart disease. The ability to automate the monitoring process could help to reduce cardiovascular disease mortality rate, as well as the reduction in healthcare costs in the public health service and in individual patient healthcare premiums. The aim of the project is to develop a novel and cost-effective solution for the heart monitoring process, to provide an initial diagnosis to uncover any abnormalities or underlying issues.

A. Commercial Solutions

Historically, patient heart monitoring is performed manually, either by using the traditional electrocardiogram machine that produces an ECG signal on a digital display which the cardiologist can analyse by eye, or by printing the signal on ECG paper. An alternative method which allows for a longer analysis is using sensors that the patient wears for 24-48 hours known as a Holter monitor. These methods usually require manual analysis by a cardiologist which can lead to high healthcare costs.

B. Similar Works

ECG monitoring has also been the focus of various research-based projects in recent years.

A research team at the Pukyong National University in South Korea created a non-contact system for ECG monitoring. The system uses capacitive coupled sensors placed on an office chair, which collects ECG data from the patient non-intrusively by measuring the capacitance formed between the body and electrodes. The system benefits from being non-intrusive, but the user is confined to the chair. Additionally, although the patient's data is uploaded to the cloud, minimal analysis is performed only heart rate monitoring [3].

Carleton University conducted research on an Internet of Things (IoT) based remote patient monitoring ECG system. The system uses an IOIO microcontroller board to collect cardiovascular data from a patient using ECG electrodes, and a mobile device which connects to the board via Bluetooth. The mobile device acts as a tether to the internet, and can also be used for visualisation. The solution fails to provide analysis on the ECG data, opting instead for visualisation [4].

II. Methodology

A. Hardware

The hardware approach used to collect electrocardiograph data is primarily done via a Raspberry Pi (RPI). An RPI is a low-power single-board computer with a low price point - typically around 30 euro - which makes it an attractive choice for low to middle income areas. Additionally, the open source Debian-based operating system for RPI Raspbian allows the installation of various software packages, including various programming languages such as Python which is used for this paper.

Since the RPI does not feature an on-board analogue-to-digital converter (ADC), an external dual in-line package (DIP) ADC is used. The Microchip MCP3008 10-bit ADC is used for conversion and was chosen due to its low price once again, as well as its easy compatibility with the RPI, with available

Python libraries to configure the serial peripheral interface (SPI) bus.

ECG data is obtained using three electrodes placed on the patient's skin. The three electrodes are connected to a single-lead ECG heart rate monitor specifically, the Analog Devices AD8232. The AD8232 is a signal conditioning block which is used to extract, filter and amplify the small biopotential signals found when acquiring ECG data.

B. Software

Python was primarily used during the making of this paper for the purpose of collecting and analysing ECG data. Python was chosen as there are numerous modules available to facilitate the process of analysing ECG data, including modules for SPI device communication and biosignal processing.

C. Pan Tompkins Algorithm

The typical ECG signal is represented in the form of a PQRST wave. The QRS complex is the most distinguishable characteristic of the waveform, representing the depolarization of the main mass of the heart's ventricles. The peak of the QRS complex is therefore often used as the focal point for ECG feature extraction for analysis purposes. Numerous algorithms exist for the detection of the QRS complexes of ECG signals.

One such algorithm is the Pan Tompkins Real-Time QRS Detection Algorithm. The algorithm is used for reliably recognising the QRS complexes of an ECG signal, performing signal processing techniques to yield an output pulse stream representing each of the identified QRS complexes in the ECG signal [5][6][7].

1) Pre-Processing: Feature scaling is performed on the ECG data before analysis occurs. Normalisation of the data allows different data sets to be used so, in theory, different ECG measuring tools can be used.

2) Bandpass Filter: The bandpass filter implemented using a low pass and high pass filter in series is used to reduce the interference of noise in the ECG signal. ECG recordings are often corrupted by noise in one of the following three ways:

- Baseline wander (BW) is a low-frequency artefact which is caused by movement or respiration of the patient.
- Power line interference is a high-frequency contamination due to electromagnetic fields from the power lines or other nearby equipment. The filter must be adjusted accordingly for the region - the line or mains frequency is 50Hz in Europe and 60Hz in the USA and parts of Asia.
- Electromyography (EMG) or muscle noise is caused by the contraction of other muscles near the heart, which is picked up by the electrodes of the ECG monitor.

3) Derivative: The signal is then differentiated which suppresses the low-frequency components of the filtered ECG signal, while amplifying the high-frequency components. The result is an ECG signal with attenuated P and T waves, with amplified QRS complexes.

4) Squaring: The squaring operation provides non-linear amplification of the ECG signal point by point. The resulting signal is all positive and emphasizes the high frequencies of the QRS complex, while suppressing the P and T waves once again.

5) Moving-Window Integration and Thresholding: A moving-window or rolling mean is used for smoothing of the output of the squaring operation. The averaging process produces a new signal used for thresholding. This thresholding is used for determining the position of R-peaks by rejecting any peaks which are below this average signal.

D. Heart Rate

The detected QRS peaks of the ECG signal are stored in a Python dictionary as the peak's sample number from the collected data. With this list of detected R-peak locations, the average heart rate can be calculated. The interval between successive detected peaks is calculated and the average heart rate is calculated from the average interval between successive R-peaks.

for *each detected R-peak* **do**
    RR-interval = R-peak(*n*) - R-peak(*n*-1);
    RR-interval-ms = (RR-interval / fs) * 1000
end
average heart rate = 60000 / average (RR-interval-ms)

E. Heart Rate Variability (HRV)

Heart rate variability (HRV) is the analysis of the heart's beat-to-beat variation, and is a good indicator of the overall heart health of the patient. A high HRV means that the heart can effectively change heart rate depending on the activity level, and indicates greater cardiovascular fitness and more resilience to stress or disease. On the contrary, a low HRV indicates that the patient is more susceptible to heart attacks, strokes etc.

In the time domain, there are several methods of measuring a patient's HRV, which are based on the beat-to-beat variance of the heart. *NN* is used interchangeably with *RR*, and simply denotes the interval between successive QRS peaks[8][9].

1) Root Mean Square of Successive Differences (RMSSD): The RMSSD measures the average interval between successive QRS peaks in the ECG signal, and is the most commonly used metric to access a patient's heart health.

$$RMSSD = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n-1}(RR_{i+1} - RR_i)^2} \tag{1}$$

2) Standard Deviation of Intervals (SDNN): The standard deviation of RR intervals measures the square root of the variance of ECG, and is dependent on the length of the ECG recording. SDNN is often calculated over a period of 24 hours.

$$SDNN = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(RR_i - \overline{RR})^2} \tag{2}$$

3) NNx and pNNx: NNx is a HRV metric which measures the number of successive RR intervals which differ by more than x where x is a millisecond interval. pNNx is the proportion of NNx compared to the total number of NN intervals.

$$NNx = \sum_{i=1}^{N}(|RR_{i+1} - RR_i| > x\ ms) \tag{3}$$

$$pNNx = \frac{NNx}{N} * 100 \tag{4}$$

III. Results

A. Data Acquisition

The implementation of the Raspberry Pi and the MCP3008 ADC provides fast and accurate data acquisition at the chosen sampling rate of 250Hz. This rate was chosen as it is a good balance between a high enough sampling rate for accurate ECG measurements, as well as being low enough so that the computational time is minimal.

For the purposes of testing, an ECG signal generator was used. The SKX-2000 simulator generates a realistic ECG signal at an adjustable heart rate. The 3-lead AD8232 heart rate monitor connects to the generator.

B. Pan Tompkins Implementation

Pan Tompkins algorithm was implemented in Python and was used to process the collected ECG data from the signal generator. The output from each stage of the Pan Tompkins algorithm is displayed below.

The result of implementing the Pan Tompkins algorithm and processing the ECG data is an output of pulse streams representing the R-peaks of the signal, where the peak of each pulse is annotated to indicate a detected QRS complex. By annotating the peaks we can easily visualise the data and ensure there are no false positives or negatives detected by the algorithm.

For demonstration and testing purposes, the heart rate from the ECG signal generator was set at 60BPM. Analysis was performed on the gathered data over a period of 30 seconds and the results are shown below.

The results obtained in Figure 2 show a calculated heart rate of 59.93BPM, indicating the algorithm performed well on the given data.

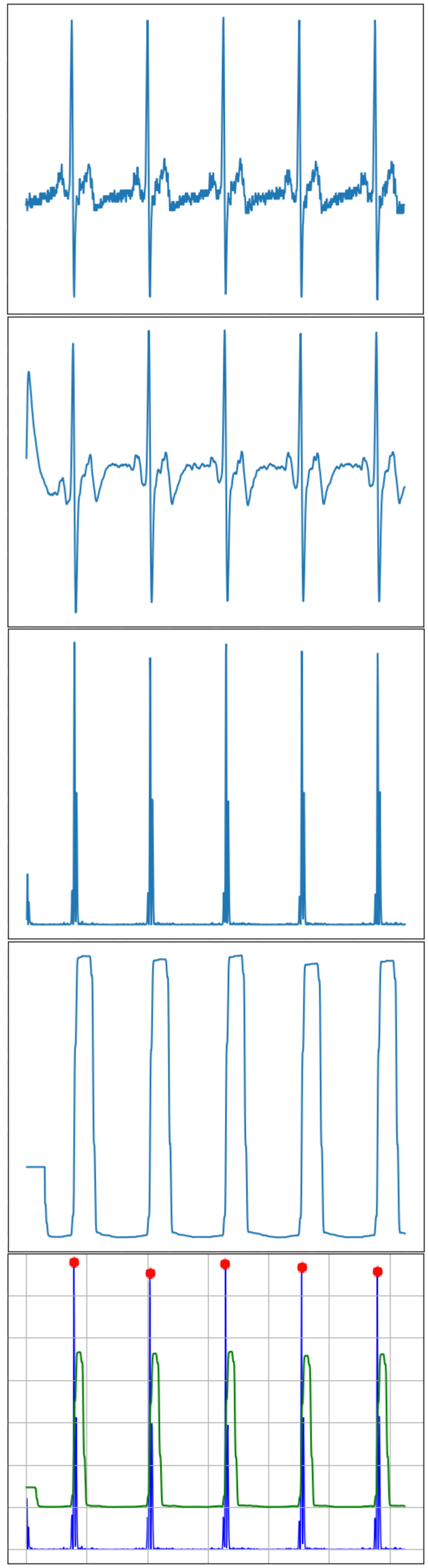

Fig. 1. shows the stages of PT algorithm: a) input signal, b) filtered, c) derivation & squaring, d) moving average, thresholding/decision making

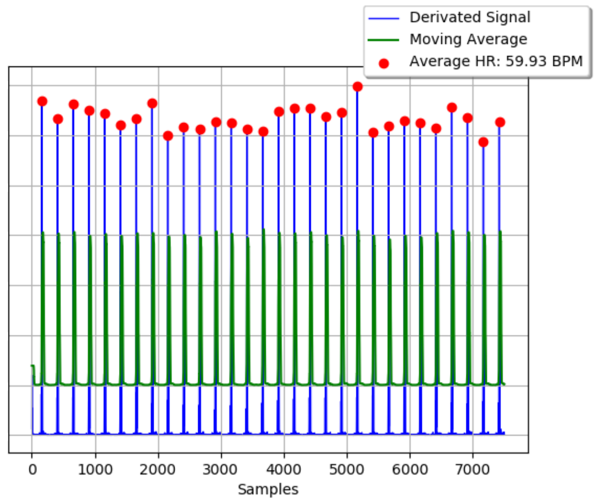

Fig. 2. shows the output of the implemented PT algorithm for an input rate of 60BPM over 30 seconds.

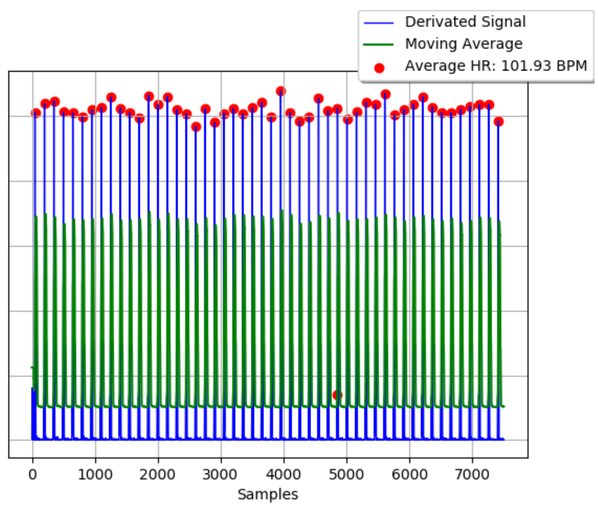

Fig. 3. shows the output of the implemented PT algorithm for an input rate of 100BPM over 30 seconds.

Repeating the analysis for a given heart rate of 100BPM, the measured rate was 101.93BPM as shown in Figure 3. The algorithm performed well locating all of the R-peaks, but one detected false positive peak indicates that there is room for improvement in the algorithm.

Below is a table containing the measured heart rate (in BPM) for different input heart rates from the ECG signal generator.

Input HR	Measured HR
60	59.92
80	79.89
100	99.85
120	119.83
140	139.99
160	160.20
180	180.45
200	199.71

C. *HRV Analysis*

Heart Rate Variability analysis was performed on the ECG results, using the three aforementioned metrics. However, since the ECG signal generator provides a constant fixed heart rate, we would expect the HRV metrics to all be close to zero, since there is no variability in the beat-to-beat intervals.

This was confirmed during the phase of collecting results.

IV. Conclusion

This paper outlines a method by which ECG monitoring can be performed, outlining the components necessary for measuring a patients ECG signal, as well as the steps involved in analysis of the collected data. One of the main focuses for this method was the implications it could have on low-income areas, driving down the costs of this kind of analysis to make it more affordable. The overall cost of this method amounts to around 50 euro, making it very affordable for low to middle income countries.

Although the algorithm performed well by positively identifying all R-peaks an ECG signal, some improvements can certainly be made by removing false positives. These false positives can skew the results of analysis and in the healthcare industry, the performance of these monitoring techniques is imperative.

Acknowledgment

The author would like to thank his supervisors Dr. Deepu John and Dr. Barry Cardiff for their continued help and support throughout the duration of this project.

References

[1] Roth, G.A., Huffman, M.D., Moran, A.E., Feigin, V., Mensah, G.A., Naghavi, M. and Murray, C.J., 2015. Global and regional patterns in cardiovascular mortality from 1990 to 2013. Circulation, 132(17), pp.1667-1678.

[2] Benziger, C.P., Roth, G.A. and Moran, A.E., 2016. The global burden of disease study and the preventable burden of NCD. Global heart, 11(4), pp.393-397.

[3] Fong, E.M. and Chung, W.Y., 2013. Mobile cloud-computing-based healthcare service by noncontact ECG monitoring. Sensors, 13(12), pp.16451-16473.

[4] Mohammed, J., Lung, C.H., Ocneanu, A., Thakral, A., Jones, C. and Adler, A., 2014, September. Internet of Things: Remote patient monitoring using web services and cloud computing. In Internet of Things (iThings), 2014 IEEE International Conference on, and Green Computing and Communications (GreenCom), IEEE and Cyber, Physical and Social Computing (CPSCom), IEEE (pp. 256-263). IEEE.

[5] Pan, J. and Tompkins, W.J., 1985. A real-time QRS detection algorithm. IEEE transactions on biomedical engineering, (3), pp.230-236.

[6] Bawa, K. and Sabherwal, P., 2014. R-peak detection by modified Pan-Tompkins algorithm. International Journal of Advancements in Research Technology, 3(5).

[7] Singh, S and Gandhi, N., 2010. Pattern Analysis of different ECG signal using Pan-Tompkin's algorithm. International Journal on Computer Science and Engineering, 2(7), pp.2502-2505

[8] Wang H. and Huang S., 2012, SDNN/RMSSD as a Surrogate for LF/HF: A Revised Investigation, Modelling and Simulation in Engineering

[9] Vollmer M., 2016, "A robust, simple and reliable measure of Heart Rate Variability using relative RR intervals", Computing in Cardiology Conference (CinC)