ELEC S421F Biomedical Informatics

Lab Session: ***Denoising and R-peak Detection in ECG Recording***

Student Name: ZHANG Hanhao

Student Number: 12399068

Introduction

ECG represents repetitive electric depolarization and repolarization pattern of heart muscle. Which characterized by P, Q, R, S, T waves and occasionally U wave is occurred. P wave is the result of depolarization of atrial, QRS complex (Q wave, R wave and S wave) is caused by ventricular depolarization and T wave shows ventricular repolarization. U wave peak also displays a portion of ventricular repolarization if present.

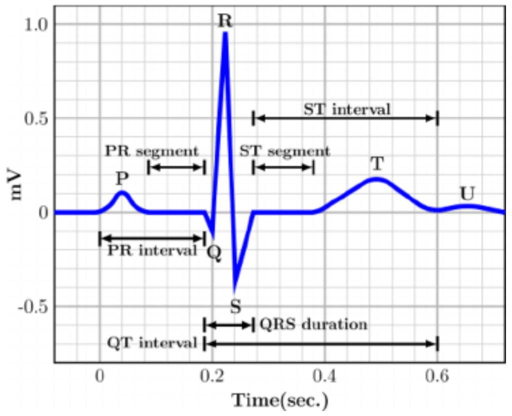


Fig.1 PQRST complexes in the ECG waveform [1]

However, there are serval noises may interfere with diagnosis of ECG signal such as baseline wander, powerline interference, EMG noise and electrode motion artifact. Embed a denoising algorithm in the program and let the raw ECG signal pass through can get a relatively accurate ECG signal. The preprocessing program can then be combined with the amplitude and 1st derivative algorithm to become a complete QRS detection system. In this report, Pan-Tompkins algorithm QRS detection program will be illustrated detailed.

Pre-process (Pan-Tompkins algorithm)

There are five steps the raw ECG signal should be passed through according to Pan-Tompkins algorithm. First is an integer coefficient bandpass filter composed of cascaded low-pass and high-pass filters. Its function is noise rejection. Next is a filter that approximates a derivative. After an amplitude squaring process, the signal passes through a moving-window integrator. Adaptive thresholds then discriminate the locations of the QRS complexes.[2] And the first thing to do is loading raw ECG signal/recording data.

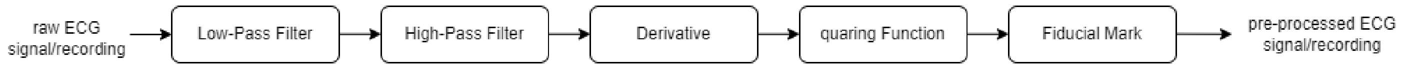
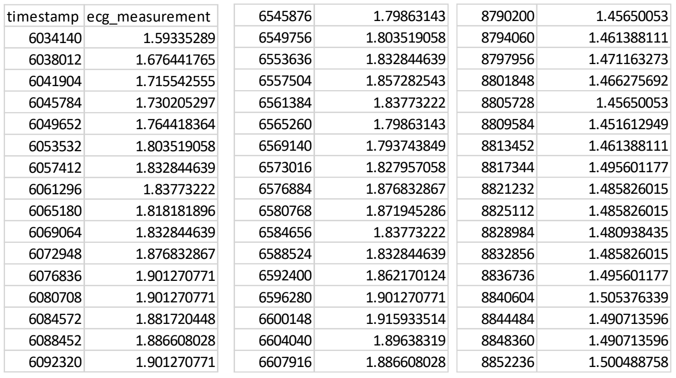


Fig.2 Pan-Tompkins algorithm Block Diagram

1. load the ECG signal



timestamp	ecg_measurement
6034140	1.59335289
6038012	1.676441765
6041904	1.715542555
6045784	1.730205297
6049652	1.764418364
6053532	1.803519058
6057412	1.832844639
6061296	1.83773222
6065180	1.818181896
6069064	1.832844639
6072948	1.876832867
6076836	1.901270771
6080708	1.901270771
6084572	1.881720448
6088452	1.886608028
6092320	1.901270771

6545876	1.79863143
6549756	1.803519058
6553636	1.832844639
6557504	1.857282543
6561384	1.83773222
6565260	1.79863143
6569140	1.793743849
6573016	1.827957058
6576884	1.876832867
6580768	1.871945286
6584656	1.83773222
6588524	1.832844639
6592400	1.862170124
6596280	1.901270771
6600148	1.915933514
6604040	1.89638319
6607916	1.886608028

8790200	1.45650053
8794060	1.461388111
8797956	1.471163273
8801848	1.466275692
8805728	1.45650053
8809584	1.451612949
8813452	1.461388111
8817344	1.495601177
8821232	1.485826015
8825112	1.485826015
8828984	1.480938435
8832856	1.485826015
8836736	1.495601177
8840604	1.505376339
8844484	1.490713596
8848360	1.490713596
8852236	1.500488758

Fig.3 Input ECG recording sample

As the figure shows above, the ECG recording are store in a xlsx file. Two columns: ‘time stamp’ and ‘ecg_measurement’.

Displaying raw ECG recording:

```
x1 = xlsread('Lab3_1.xlsx');
x1 = x1(:,2);
fs = 250;
N = length (x1);
t = [0:N-1]/fs;
figure(1)
subplot(2,1,1)
plot(t,x1)
xlabel('second');ylabel('Volts');title('Raw ECG Signal')
subplot(2,1,2)
plot(t(250:2250),x1(250:2250))
xlabel('second');ylabel('Volts');title('Input ECG Signal 1-9 second')
xlim([1 9])
```

line 1. Load the ECG signal from the file *Lab3_1.xlsx*, file name can be different as the format is the same.

line 2. Load the second column of the file, ecg_measurement.

line 3. An analog-to-digital converter (ADC) samples the ECG at a rate of 250 samples/s.

line 4. Signal length is the number of file rows.

line 5. Time index from 0 to the last row of the ecg_measurement.

line 6. Makes the figure specified by 1 and displays it on top of all other figures.

line 7. Divides the current figure into a 2 -by- 1 grid and creates axes in the position specified by 1.

line 8. Plot the data in x1 versus the corresponding values in t.

line 9. Adds the x-axis label to the ‘second’, y-axis label to ‘Volts’ and sets title as ‘Raw ECG Signal’.

line 10. Divides the current figure into a 2 -by- 1 grid and creates axes in the position specified by 2.

line 11. Plot the data in x1 from the 250th to 2250th versus the corresponding values in t from the 250th to 2250th.

Totally 10 ECG cycles

line 12. Adds the x-axis label to the 'second', y-axis label to 'Volts' and sets title as 'Input ECG Signal 1-9 second'.

line 13. Sets the x-axis limits for the current axes. Specify limits from 1 to 9.

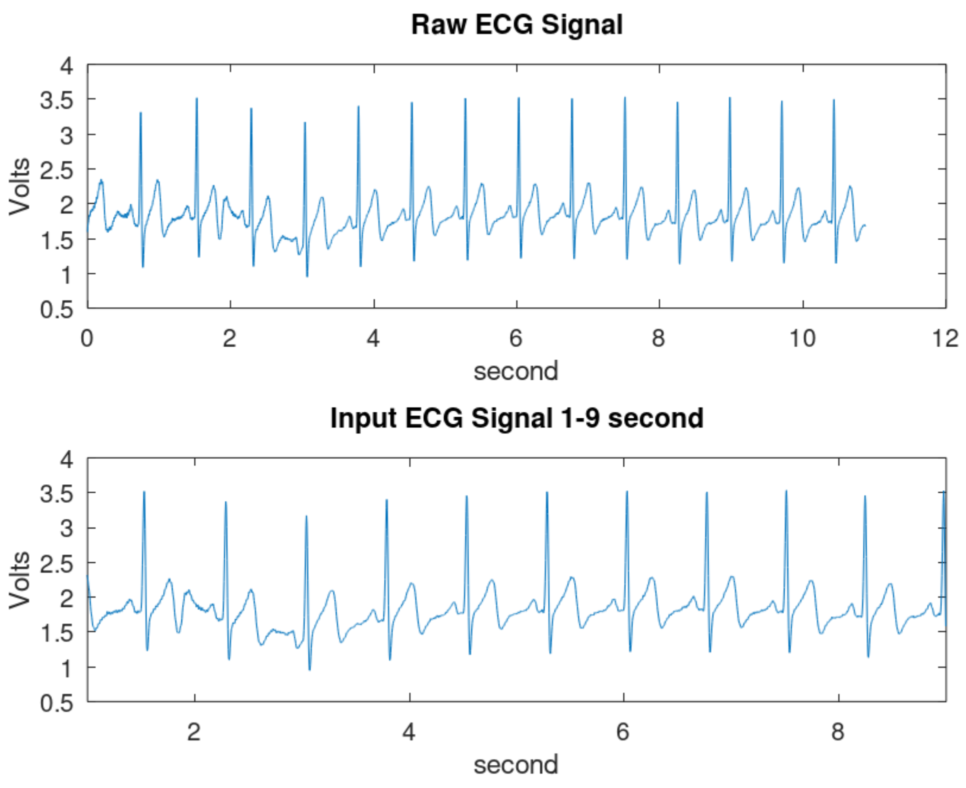


Fig.4 Row ECG Signal and Input ECG Signal 1-9 second

Cancellation DC Drift and Normalization:

The figure 4 display the raw ECG recording and its 10 cycles. There are a lot of noise and curve varies sharply. And the baseline drift appears in the first 3 second. The waveform shifted or offset from the '0' volts baseline apparently. The DC drift cancellation algorithm reduces noise in the ECG signal by matching the spectrum of the average QRS complex. This attenuates noise due to muscle noise, power line interference, baseline wander, T wave interference. [3] Normalization can convert an ECG signal into multiple template cycles, which are comparable between any two ECGs, no matter the sampling rates or health status. [4]

```
x1 = x1 - mean (x1);
x1 = x1/ max( abs(x1 ));
figure(2)
plot(t,x1)
xlabel('second');ylabel('Volts');title('Cancellation DC Drift and Normalization')
```

line 14. Cancel DC components

line 15. normalize to one

line 16. Makes the figure specified by 2 and displays it on top of all other figures.

line 17. Plot the data in x1 which has been cancelled DC drift and normalized versus the corresponding values in t.

line 18. Add x-axis, y-axis labels and set title.

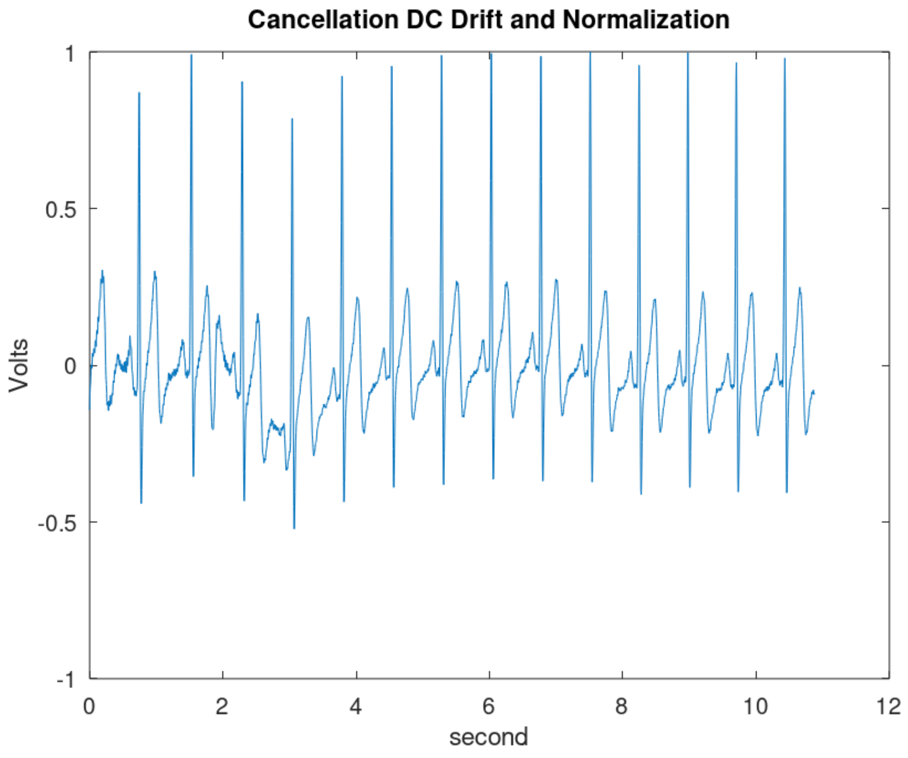


Fig.5 Cancellation DC Drift and Normalization

2. Bandpass Filter (Low-Pass Filter & High-Pass Filter)

The bandpass filter reduces the influence of muscle noise, 60 Hz interference, baseline wander, and T-wave interference. [2] In Pan-Tompkins algorithm, it cascaded the low-pass and high-pass filters described below to achieve a 3 dB passband from about 5-12 Hz.

The transfer function of the second-order low-pass filter is $H_{(z)} = \frac{(1-z^{-6})^2}{(1-z^{-1})^2}$

b=[1 0 0 0 0 0 -2 0 0 0 0 0 1];

a=[1 -2 1];

h_LP=filter(b,a,[1 zeros(1,12)]);

x2 = conv (x1 ,h_LP);

x2 = x2/ max(abs(x2));

line 19. Low pass filter

line 20. Low pass filter

line 21. Filters the input data [1 0 0 0 0 0 0 0 0 0 0 0 0] using a rational transfer function defined by the numerator and denominator coefficients b and a. Its value is [1 2 3 4 5 6 5 4 3 2 1 0 0]

line 22. Returns the convolution of vectors x1 and h_LP (low pass filter), which is equivalent to multiplying the two polynomials.

line 23. Normalize the data after passing through low-pass filter, for convenience.

The design of the high-pass filter is based on subtracting the output of a first-order low-pass filter from an all-pass filter (i.e., the samples in the original signal). The transfer function for such a high-pass filter is $H_{(z)} = \frac{(-1+32z^{-16}+z^{-32})}{(1-z^{-1})}$. [2]

b = [-1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 32 -32 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1];
a = [1 -1];
h_HP=filter(b,a,[1 zeros(1,32)]); % impulse response of HPF
x3 = conv (x2 ,h_HP);
x3 = x3/ max(abs(x3));

line 24. High pass filter
line 25. High pass filter
line 26. Impulse response of High pass filter. Filters the input data [1 0 0 0 … 0 0 0 (32 zeros)] using a rational transfer function defined by the numerator and denominator coefficients b and a. Its value is [-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 31 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -10].
line 27. Returns the convolution of vectors x1 and h_HP (high pass filter), which is equivalent to multiplying the two polynomials.
line 28. Normalize the data after passing through low-pass filter, for convenience.

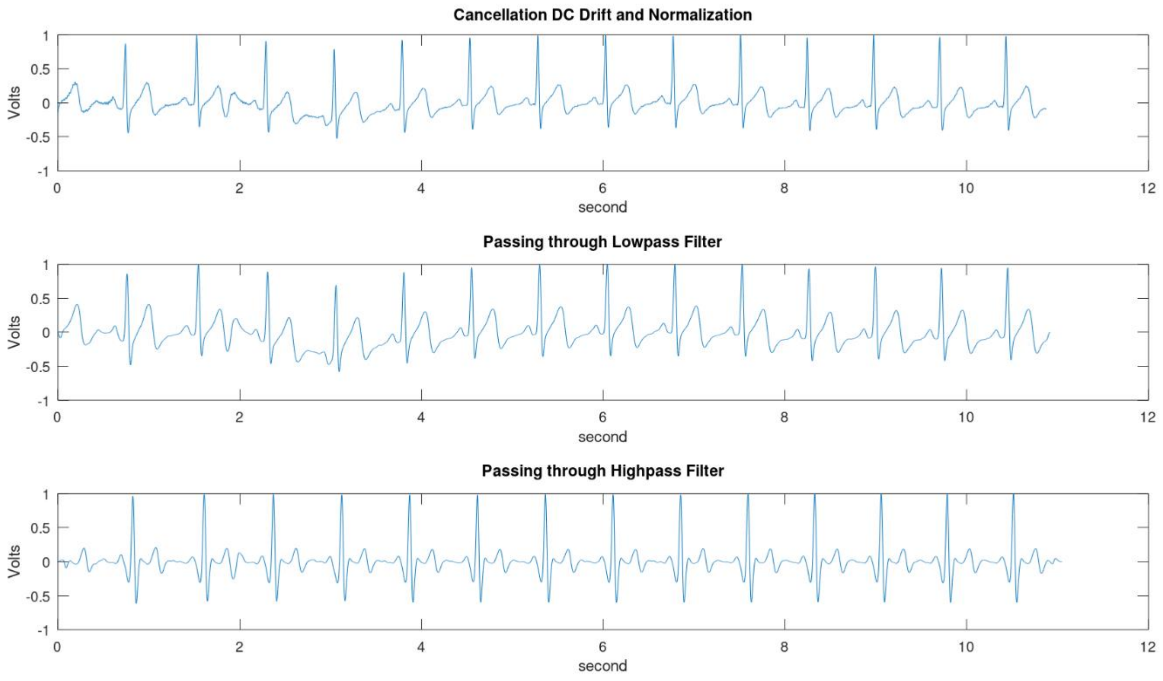


Fig.6 Low pass filter & High pass filter

The figure 6 shows that the signal is differentiated to provide the QRS complex slope information after filtering.

3. Derivative

The derivative shows the rate of change in the ECG and is small for slowly changing low frequency wave forms.

$$H_{(z)} = \left(\frac{1}{T}\right)(-z^{-2} - 2z^{-1} + 2z + z^{2})$$

h = [-1 -2 0 2 1]/8;

x4 = conv (x3 ,h);

x4 = x4 (2+[1: N]);

x4 = x4/ max(abs(x4));

line 29. Make impulse response

line 30. Returns the convolution of vectors x3 and impulse response which is equivalent to multiplying the two polynomials.

line 31. Frequency response of this derivative is nearly linear between dc and 30 Hz (i.e., it approximates an ideal derivative over this range). Its delay is two samples. So, we offset the delay.

line 32. Normalize the data.

4. Squaring

After differentiation, the signal is squared point by point. Squaring can make all data points positive and does nonlinear amplification of the output of the derivative emphasizing the higher frequencies (i.e., predominantly the ECG frequencies). The equation of this operation is

$$y(nT) = [x(nT)]^{2}$$

x5 = x4 .^2;

x5 = x5/ max(abs(x5));

line 33. Squaring function.

line 34. Normalize the data.

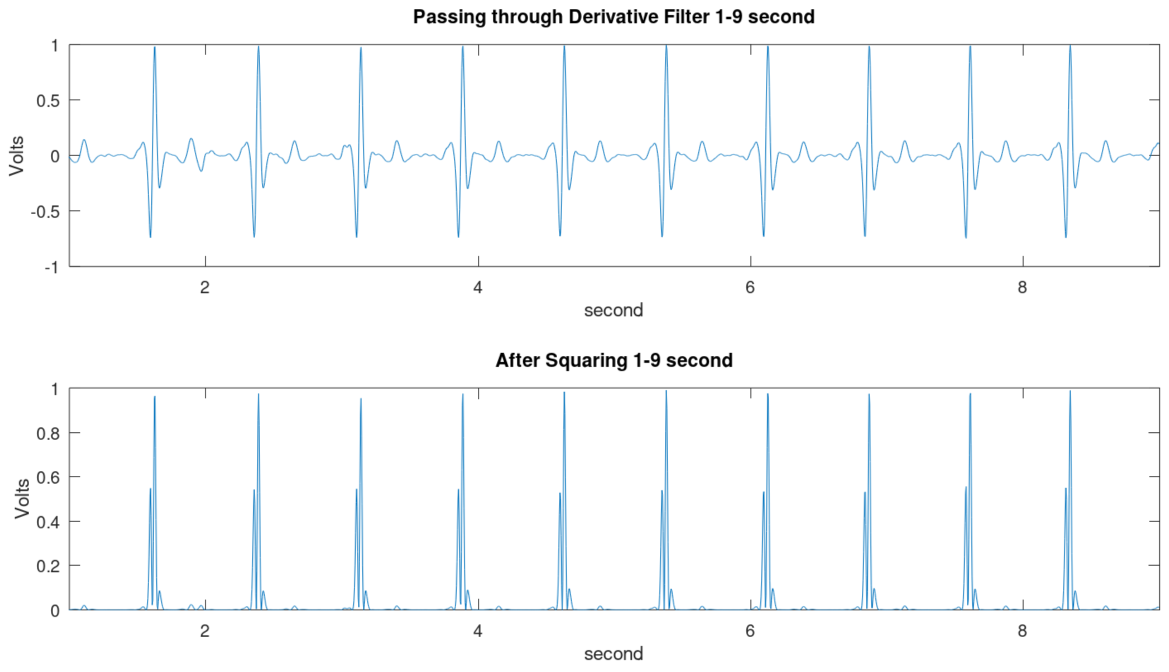


Fig.7 Signal pass through derivative filter and Squaring

5. Moving-Window Integration

The purpose of moving-window integration is to obtain waveform feature information in addition to the slope of the R wave. It is calculated from $y(nT) = \left(\frac{1}{N}\right)[x(nT-(N-1)T + x(nT-(N-2)T + \cdots + x(nT)]$. The sample rate of the recording is 250 samples/s, the window is 30 samples wide

```
h = ones (1 ,31)/31;
x6 = conv (x5 ,h);
x6 = x6 (15+[1: N]);
x6 = x6/ max( abs(x6 ));
figure(2)
plot([0:length(x6)-1]/fs,x6)
xlabel('second');ylabel('Volts');title('ECG Signal After Applying Pan Tomkins Algorithm')
```

line 35. Make impulse response [0.032258, …., 0.032258] (1x31)

line 36. Returns the convolution of vectors x5 and impulse response h, which is equivalent to multiplying the two polynomials.

line 37. Make up for the delay (15 samples wide) according to FIR filter

$$\frac{N-1}{2F_s}$$

Where F_s is sample frequency.

line 38. Normalize the data.

line 39. Makes the figure specified by 2 and displays it on top of all other figures.

line 40. Plot the data in x6 versus the corresponding values in its time range.

line 41. Adds the x-axis label to the 'second', y-axis label to 'Volts' and sets title as 'ECG Signal After Applying Pan Tomkins Algorithm'.

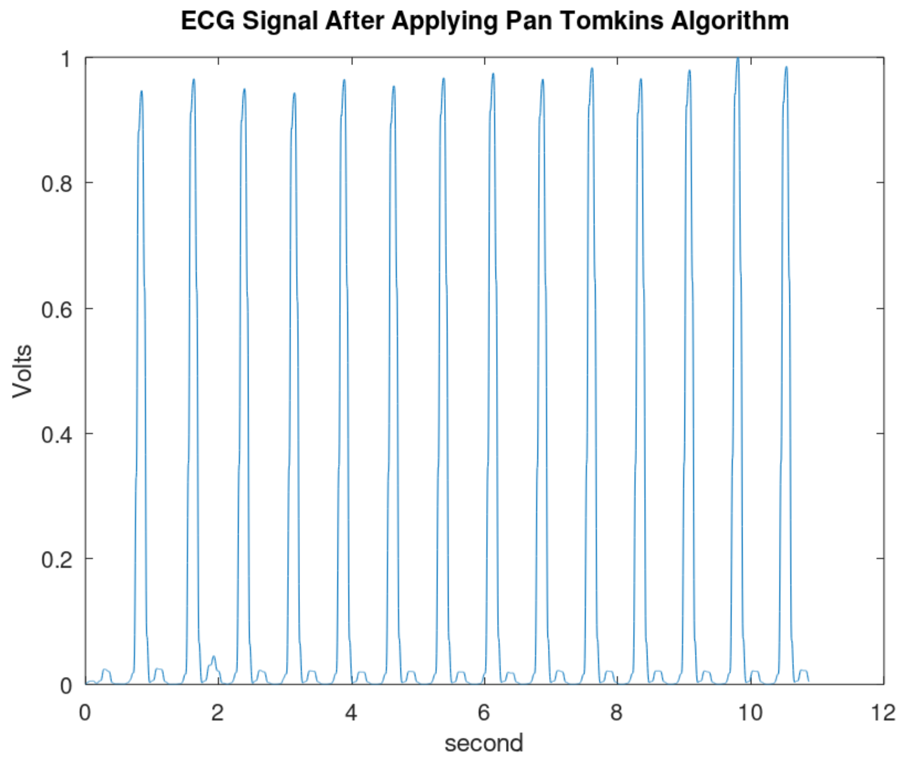


Fig.8 ECG Signal After Applying Pan Tomkins Algorithm

After applying Pan-Tomkins algorithm, it makes all the recording point positive and reduces the influence of muscle noise, 60 Hz interference, baseline wander, and T-wave interference. The Moving-Window Integration also helps us detect R wave peak through slope change. The detect algorithm can be applied later.

R peak detection

Here are two approaches to find the R peak, the first is 'find peak'. Codes are shows below.

```
y=[0:length(x6)-1]/fs;
[pks,locs]=findpeaks(x6,'MinPeakDistance',150);
figure(3);
subplot(2,1,1);
plot([0:length(x6)-1]/fs,x6,y(locs),pks,'o','MarkerFaceColor','g','MarkerSize',5);
title('Plotting The R Peak by Using FINDPEAK Function');
```

line 42. Define y as every sample of x-axis.

line 43. Use findpeak function to get the peak value (pks) and its location sample (locs). It requires a minimum amplitude difference of 150 samples between a peak and its neighbors.

line 44. Makes the figure specified by 3 and displays it on top of all other figures.

line 45. Divides the current figure into a 2 -by- 1 grid and creates axes in the position specified by 1

line 46. Plot the data in x6 versus the corresponding values in its time range. Mark the peak point in green dot.

line 47. Set the figure title as Plotting The R Peak by Using FINDPEAK Function.

The peak point is determined by the minimum amplitude difference, here are four examples with different minimum amplitude difference. (100, 150, 200, 400)

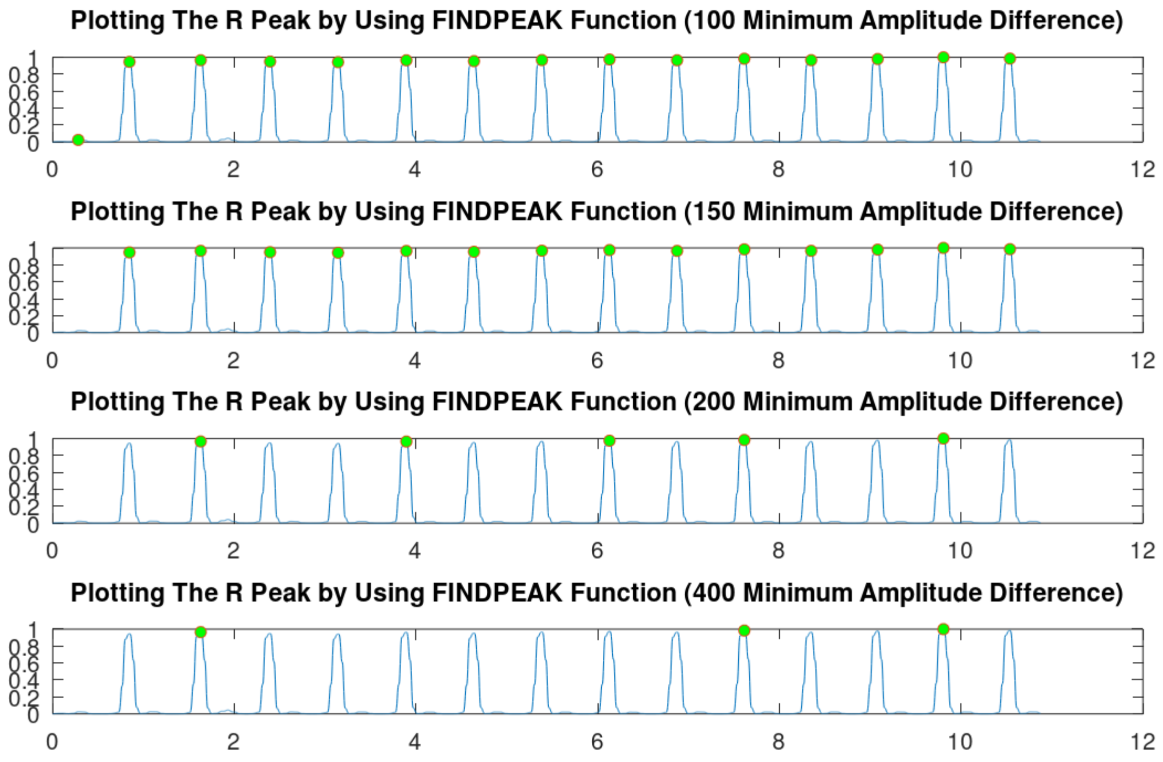


Fig.9 Plotting The R Peak by Using FINDPEAK Function with Different Difference

Figure 9 shows the difference when using different minimum amplitude difference. The proper minimum amplitude difference is around 150. If the window is too narrow, R peak detect algorithm may detect wrong point. If it is too wide, many ECG cycles will be skipped. This is the biggest disadvantage of using *‘findpeak’* function. If we use 100 as minimum amplitude difference, the first peak point drops on a T wave. And the second peak point is correct because it restricts the acceptable peak-to-peak separations to values greater than 100. However, using 400 as an acceptable peak-to-peak separations, the displayed results are logically different from that using 200. It skips the peaks from 4 seconds to 6 seconds. Although 400 cannot be used as a suitable minimum separation, it must be very unstable to use *‘findpeak’* function to find peak. Thus, finding the right window should through trial and error. Here is another approach to detect R peak by using threshold.

```
max_h = max(x6);
thresh = mean (x6 );
poss_reg =(x6>thresh*max_h)';
left = find(diff([0 poss_reg])==1);
right = find(diff([poss_reg 0])==-1);
left=left-(15);
right=right-(15);
for i=1:length(left)
    [R_value(i) R_loc(i)] = max( x6(left(i):right(i)) );
    R_loc(i) = R_loc(i)-1+left(i); % add offset
end
R_loc=R_loc(find(R_loc~=0));
figure(3);
subplot(2,1,2);
plot([0:length(x6)-1]/fs,x6,y(R_loc),R_value,'o','MarkerFaceColor','r','MarkerSize',5);
title('Plotting The R Peak by Using Threshold');
```

line 48. Find the maximum value of the pre-processed ECG recording.
line 49. Find the mean value of the pre-processed ECG recording.
line 50. Set the threshold as the mean value multiple the maximum value, if the data value is larger than the threshold value, it return 1. Otherwise, 0 will be returned
line 51. Let *poss_reg[N + 1] - poss_reg[N]*, find the position with a difference of 1. Find the first point above the threshold. 15 sample delay occurred because of LP and HP
line 52. Let *poss_reg[N + 1] - poss_reg[N]*, find the position with a difference of -1. Find the last point above the threshold. 15 sample delay occurred because of LP and HP
line 53. Cancel delay
line 54. Cancel delay
line 55. Create a for loop, from 1 to the number of the data above threshold
line 56. Find the peak point value and position for every interval (from the left point above threshold to the right point above threshold) that above the threshold
line 57. Add offset
line 58. End the loop
line 59. Make sure there is no selective way
line 60. Makes the figure specified by 3 and displays it on top of all other figures.

line 61. Divides the current figure into a 2 -by- 1 grid and creates axes in the position specified by 2
line 62. Plot the data in x6 versus the corresponding values in its time range. Mark the peak point in red dot.
line 63. Set the figure title as Plotting The R Peak by Using Threshold.

The figures below clearly proves that two methods of detecting R peak are feasible. Therefore, Pan-Tompkins algorithm is effective for real-time ECG denoising and R-peak detection within the current format of ECG recording.

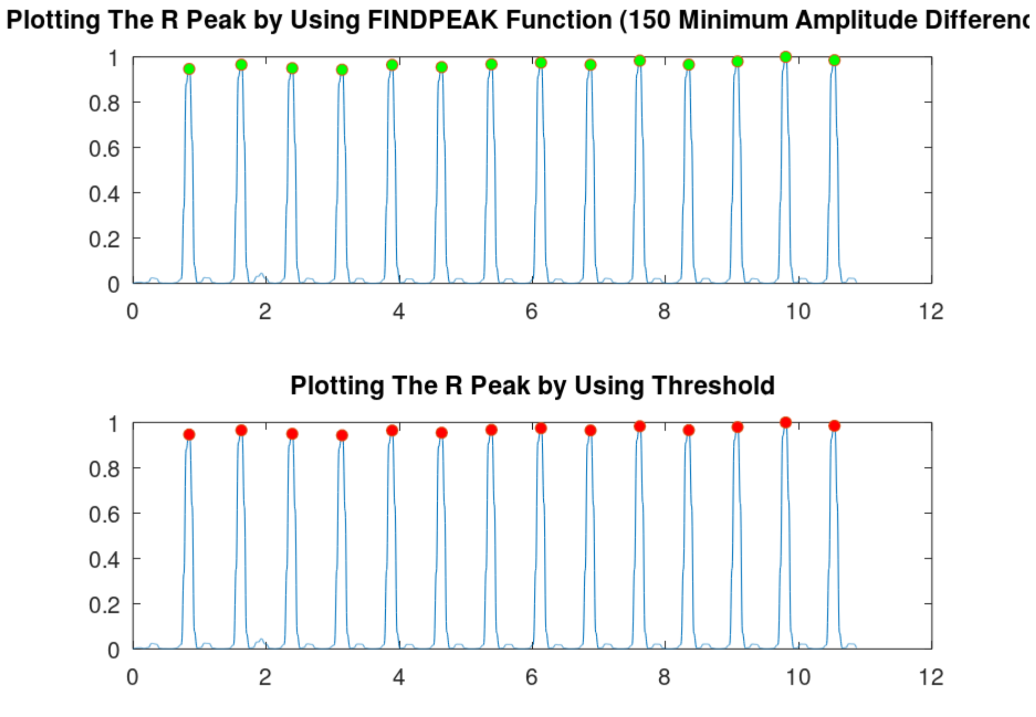


Fig.10 Plotting The R Peak by different approaches

Discussion

The Pan-Tompkins Algorithm is the most widely used QRS complex detector for the monitoring of many cardiac diseases. The bandpass filter can reduce the influence of muscle noise, 60 Hz interference, baseline wander, and T-wave interference effectively. This method provides a good detection performance with high-quality clinical ECG signal data.[5] The ECG recording data used in this experiment is perfect. If we use a low-quality ECG signal recording, the R peak detection accuracy should be further investigated. Here we introduced two different detection approaches. Using '*findpeak*' function in MATLAB to get the peak is required to select a suitable window because it may detect wrong point or skips valuable points. Although the use of threshold to detect R peak does not require constant trial and error, it still needs to offset the delay caused by low pass and high pass filters manually.

Appendix

[1] Chen, Shuo-Tsung & Guo, Yuan-Jie & Huang, Huang-Nan & Kung, Woon-Man & Tu, Shu-Yi. (2014). Hiding Patients Confidential Datainthe ECG Signal viaa Transform-Domain Quantization Scheme. Journal of medical systems. 38. 54. 10.1007/s10916-014-0054-9.

[2] J. Pan and W. J. Tompkins, "A Real-Time QRS Detection Algorithm," in IEEE Transactions on Biomedical Engineering, vol. BME-32, no. 3, pp. 230-236, March 1985, doi: 10.1109/TBME.1985.325532.

[3] H. Myo Tun, “Analysis of Computer Aided Identification System for ECG characteristic points,” International Journal of Biomedical Science and Engineering, vol. 3, no. 4, p. 49, 2015.

[4] M. Yang, B. Liu, M. Zhao, F. Li, G. Wang, and F. Zhou, “Normalizing electrocardiograms of both healthy persons and cardiovascular disease patients for biometric authentication,” PLoS ONE, vol. 8, no. 8, 2013.

[5] M. A. Fariha, R. Ikeura, S. Hayakawa, and S. Tsutsumi, “Analysis of Pan-Tompkins algorithm performance with noisy ECG signals,” Journal of Physics: Conference Series, vol. 1532, no. 1, p. 012022, 2020.