

CONTENTS Close

521273S - Biosignal Processing I - Online Labs - Autumn 2024

Assignment 1 - Respiration analysis

Assignment 2 - ECG Filtering to Remove Noise

Assignment 3 - Adaptive filtering

Assignment 4 - Pan-Tompkins Algorithm for QRS Detection

Implement Pan-Tompkins filters

Assignment 5 - EMG and Muscular Force Analysis

Courses & Content | LMS Integration | Documentation & Support ▾

521273S - Biosignal Processing I - Online Labs - Autumn 2024 > Assignment 4 - Pan-Tompkins Algorithm for QRS Detection >

Implement Pan-Tompkins filters15 solutions submitted (max: Unlimited) | [View my solutions](#)[Open Problem in MATLAB Online](#)**Introduction**

The Pan-Tompkins algorithm identifies QRS complexes (see Fig. 1) based on analysis of the slope, amplitude and width of the QRS. The various stages of the algorithm are shown in Fig. 2.



Figure 1. Schematic representation of normal ECG



Figure 2. Block diagram of the Pan-Tompkins algorithm

The bandpass filter, formed using lowpass and highpass filters, reduces noise (such as muscle noise, 60 Hz interference and baseline drift) in the ECG signal. After that, the signal is passed through a differentiator to provide a large response at the high slopes that distinguish QRS complexes from low-frequency ECG components such as the P and T waves.

The next operation is the squaring operation, which emphasizes the higher values expected within QRS complexes and suppresses smaller values related to the P and T waves among noise in the output of the preceding stage. The squared signal is then passed through a moving-window integrator of window length $N = 30$ samples (for the sampling frequency of $FS = 200$ Hz). The expected result is a single smooth peak related to the QRS complex for each ECG cycle. The output of the moving-window integrator may be used to detect QRS complexes, measure RR intervals, and determine the duration of the QRS complex (see Fig. 3).

Read Section 4.3.2 of the [course book](#) for more details. The eBook is accessible within the university network, which you can also reach remotely with [VPN](#).

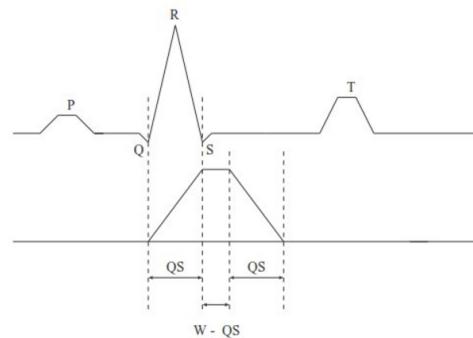


Figure 3. The relationship between a QRS complex and the moving-window integrator output

Data

The data file `ECG.txt` contains the input ECG sampled at 200 Hz rate. For the task, you are also supplied with a black box function named `findQRS` that finds the beginning and endings of the QRS complexes when supplied with the filtered signals, and detection thresholds.

Useful MATLAB commands`filter, ones`**Task**

Fill in the missing parts of the script to perform various filtering procedures that compose the Pan-Tompkins algorithm. Use the `filter` function for each step of signal processing; see Section 4.3.2 of the course book about details of the algorithm. **Note:** Yes, you need to read the chapter from the book at this point.

All the transfer functions of the filters are given in the course book. Before applying them with `filter` command, put them in the following format to get the coefficients `a` and `b`:

$$Y(z) = \frac{b(1) + b(2)z^{-1} + \dots + b(m+1)z^{-m}}{1 + a(2)z^{-1} + \dots + a(n+1)z^{-n}} X(z) \quad (1)$$

The amplitude of an ECG signal may start with a value other than zero. Consequently, the differentiator in the Pan-Tompkins algorithm will amplify the initial step, possibly resulting in an erroneous beat detection. In order to prevent this problem, we will subtract the value of its first sample from the entire ECG signal prior to processing by the Pan-Tompkins algorithm.

You will need to accomplish the following steps:

1. Load the data and calculate the sample time
2. Subtract the value of the first sample of the ECG signal from the entire ECG
3. Low pass filter, equation 4.7
4. High pass filter, equation 4.11
5. Derivative filter (Notice: Formula 4.14 in the course book (2015, formula 4.13 2002) is incorrect! Correct formula: $y(n) = \frac{1}{8} [x(n) + 2x(n-1) - 2x(n-3) - x(n-4)]$)
6. Squaring and Integration, formula 4.15 (2015, formula 4.14 2002), $N = 30$
7. Detecting the QRS complexes using the provided `findQRS` blackbox function. Use blanking interval of 250 ms, and the detection thresholds given in the code template
8. Calculate the combined filtering delay

The `findQRS` function call syntax is as follows: `[QRSSStart, QRSEnd] = findQRS(data, blankingInterval, threshold1, threshold2)` where

Inputs:

- data: P-T output from which you want to detect the QRS complexes (point 6. above)

- blankingInterval: number of samples not processed after a QRS complex found, i.e. new start of QRS is not allowed within this distance from previous hit
- threshold1: Q-wave begins here
- threshold2: S-wave

Outputs:

- QRSSstart: beginning points of QRS complexes
- QRSEnd: ending points of QRS complexes

Script ?

Save

Reset

MATLAB Documentation

[Open Problem in MATLAB Online](#)

```

1 % The sampling rate is 200 Hz
2 FS = 200;
3
4 % Calculate the sample interval from FS
5 T = 1 / FS;
6
7 % Load the ECG from the file 'ECG.txt'
8 ECG = load('ECG.txt');
9
10 % Subtract the first sample value to prevent P-T to amplify initial step
11 ECG = ECG - ECG(1);
12
13 % Lowpass filter the ECG
14 b_lowpass = (1/32) * [1 0 0 0 0 0 -2 0 0 0 0 0 1];
15 a_lowpass = [1 -2 1];
16 ECG_filtered1 = filter(b_lowpass, a_lowpass, ECG);
17
18 % Highpass filter the lowpass filtered ECG
19 b_highpass = (1/32)*[-1, zeros(1, 15), 32, -32, zeros(1, 14), 1];
20 a_highpass = [1 -1];
21 ECG_filtered2 = filter(b_highpass, a_highpass, ECG_filtered1);
22
23 % Differential filter the high- and lowpass filtered ECG
24 b_diff = (1/8)*[1, 2, 0, -2, -1];
25 a_diff = 1;
26 ECG_filtered3 = filter(b_diff, a_diff, ECG_filtered2);
27
28 % Square the derivative filtered signal
29 ECG_filtered4 = ECG_filtered3 .^ 2;
30
31 % Moving window integrator filter the squared signal
32 % Window size
33 N = 30;
34 b_integ = ones(1, N) / N;
35 a_integ = 1;
36 ECG_filtered5 = filter(b_integ, a_integ, ECG_filtered4);
37
38 % Set the blanking interval to 250 ms, but convert it to samples for the findQRS function
39 blankingInterval = (250 * FS) / 1000;
40
41 % The amplitude threshold for QRS detection are set to these
42 threshold1 = 500;
43 threshold2 = 2650;
44
45 % Call the findQRS function
46 [QRSSstart_ECG, QRSEnd_ECG] = findQRS(ECG_filtered5, blankingInterval, threshold1, threshold2);
47
48 % Calculate the cumulative filter delays (in samples)
49 |delays = 21;

```

Run Script

Previous Assessment: All Tests Passed

[Run Pretest](#)

[Submit](#)

Correct sample interval

Correctly loaded ECG

Correct lowpass filter coefficients b

Correct lowpass filter coefficients a

Correctly lowpass filtered ECG

Correct highpass filter coefficients b

Correct highpass filter coefficients a

Correctly highpass filtered lowpass filtered ECG

Correct differential filter coefficients b

Correct differential filter coefficients a

- Correctly differentially filtered ECG
- Correctly squared signal
- Correct integrator filter coefficients b
- Correct integrator filter coefficients a
- Correctly integrator filtered signal
- Correct blanking interval
- Correct QRS beginning detections
- Correct QRS ending detections
- Correct filter delays
- > Plotting - not a real test (Open to view) (Pretest)

Output

```
ECG =
```

```
0
-2
3
39
54
69
83
83
96
127
163
161
172
158
123
91
75
64
60
38
10
-42
-33
-29
-40
-32
-26
-22
25
142
219
450
832
833
536
244
8
-225
-268
-168
-61
-3
-1
-6
13
15
15
15
12
32
29
24
47
48
43
47
64
```