521273S Biosignal Processing I

Assignment 4. Pan-Tompkins Algorithm for QRS Detection

Deadline: Friday November 23th at 10.00

Learning outcomes

After this assignment, student can

- detect QRS complexes in ECG signal using the Pan–Tompkins algorithm

In order to pass the exercise, all correctly executed task results must be personally presented to a course assistant during the scheduled laboratory exercise.

Background

Read chapter "4.3.2 The Pan-Tompkins algorithm for QRS detection" from the course book. Participate in the lecture on Tuesday November 20th.

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.

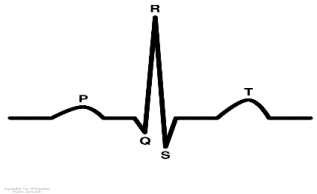

Figure 1. Schematic representation of normal ECG

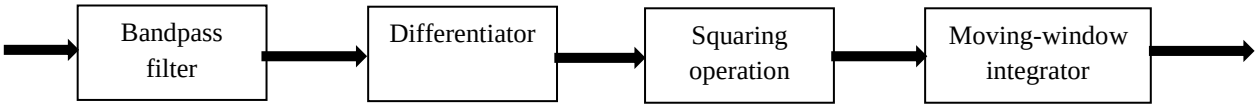

Figure 2. Block diagram of the Pan-Tompkins algorithm

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 F_S = 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.

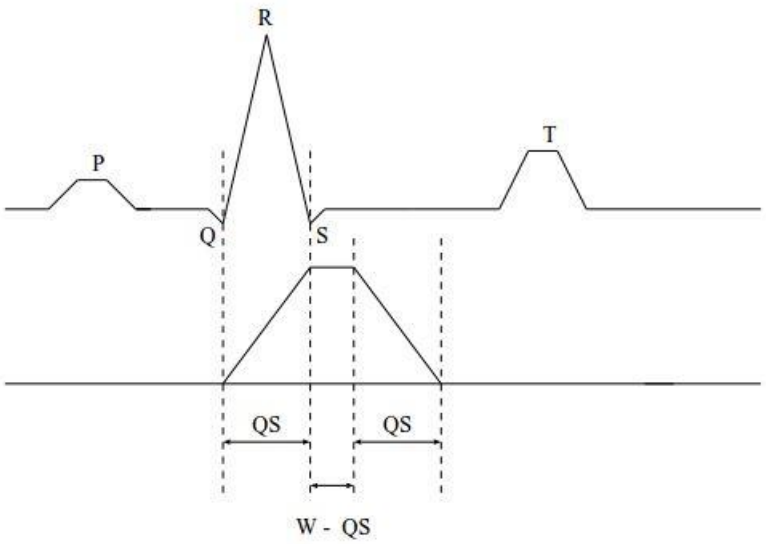

Figure 3. The relationship between a QRS complex and the moving-window integrator output

Data

Download the data file ***ECG.txt*** and black box function ***findQRS.p*** from Noppa. The sampling frequency of the ECG signal is 200 Hz.

Useful MATLAB commands

`filter, ones`

Tasks

Write your solution as a MATLAB script (m-file)!

Create a MATLAB script to perform various filtering procedures that compose the Pan-Tompkins algorithm. Use the filter command for each step; 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.

Plot the input (original ECG) and output signals at each stage of the program in same figure using subplot. As a result you should have a figure with 6 subplots: `subplot(6,1,x)`.

Note: Ensure that the axes of the plots are labeled in suitable units such as seconds. The amplitude of the signals may be shown in arbitrary units (AU). Remember also to put the titles on all plots!

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*:

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

Note: 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, subtract the value of its first sample from the entire ECG signal prior to processing by the Pan-Tompkins algorithm.

Steps of **Pan-Tompkins (P-T) algorithm** that you need to apply:

1. Subtract the value of the first sample of the ECG signal from the entire ECG
2. Low pass filter, equation 4.7, `subplot(6,1,2)`
3. High pass filter, equation 4.11, `subplot(6,1,3)`
4. Derivative filter `subplot(6,1,4)`
    - Notice: Formula 4.14 in the course book (2015, formula 4.13 2002) is incorrect!
    - Correct formula:

$$y(n) = \frac{1}{8}\left[x(n) + 2x(n-1) - 2x(n-3) - x(n-4)\right]$$

5. *(A)* Squaring and *(B)* integration, formula 4.15 (2015, formula 4.14 2002), N = 30
   ➔ output of the Pan-Tompkins algorithm, `subplot(6,1,5)`
6. Plot the same ECG signal as in `subplot(6,1,1)` again to `subplot(6,1,6)`

Next step is to detect QRS complexes. Use the provided *`findQRS`* blackbox function for that. Notice, *`findQRS.p`* is a protected function so its content is obfuscated.

- `[QRSStart, QRSEnd] = findQRS(data,blankingInterval,treshold1,treshold2)`
- Inputs:
    - `data`: P-T output from which you want to detect the QRS complexes (point 5. 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
    - `treshold1`: Q-wave begins here
    - `treshold2`: S-wave
- Outputs:
    - `QRSStart`: beginning points of QRS complexes
    - `QRSEnd`: ending points of QRS complexes
- Use blanking interval *250 ms* (convert to the number of samples for the function *`findQRS`*!).
- Select the thresholds by yourself **using the Fig. 3 on the previous page of this handout** as help. Mark the detected peak locations on the output of the integrator (`subplot(6,1,5)`) with red stars (`'r*'`) for starting points and red circles (`'ro'`) for ending points.
- Also mark the corresponding original ECG signal with the same signs in the last subplot (`subplot(6,1,6)`). After taking filtering delay into account, the markings should be found at the beginning and at the end of each QRS complex in `subplot(6,1,6)`.
    - Remember to take into account delays introduced by the filters (found from the course book)!
    - Adjust your threshold values observing `subplot(6,1,5)` for so long that all QRS complexes are found.

Additional task (if you want to do more):

Include steps in your code to compute the following parameters for ECG signal:

- Total number of beats detected
- Average RR interval (in *ms*)
- Standard deviation of RR intervals (in *ms*)
- Heart rate in beats per minute