Analysis of electrocardiographic (ECG) signals

Žiga Trojer

Abstract - for the purpose of the first seminar paper in the subject of OBSS, I implemented the procedure of detection of QRS complex in ECG signal according to the article 'A Real-Time QRS Detection Algorithm' [1] [2], [3]. The algorithm works well as it uses different approaches such as special digital bandpass filter and automatic adjustment of threshold and parameters according to different ECG changes. When using real-time detection, the evaluation of the algorithm on the MIT-BIH database [3] [4] gives 66,5 % sensitivity and 97,6 % positive predictivity. When not using real-time detection, the evaluation of the algorithm on the same database gives 93,8 % sensitivity and 98,7 % positive predictivity

Introduction

QRS detection is a very difficult problem as there are many factors present that make detection difficult. These are, for example, the variability of the QRS complex and various noises that are present in the signal - muscle noises (occurs due to the movement of the electrodes). There is also a T - wave, which can be misidentified with a QRS complex. The approach described in the article reduces the impact of these noises with digital filters.

The data processing involves several important steps. These steps are: use of bandpass filter, derivative and moving window integrator. Then the signal is squared - this is a nonlinear transformation. Then, the T-wave exclusion principle and the adaptive threshold principle are used in the decision-making process.

Methods

I implemented the algorithm in Matlab. I also used Cygwin64 for data preparation purposes. I downloaded the data from the Physionet, where I used the MIT-BIH Arrhythmia Database [2] [4] . This one contains half-hour ECG signals from 47 subjects - I used 41 of them, as 6 records were causing me problems in preparation. Due to this problem, I also included 5, 24 hour recordings from another database - MIT-BIH Long Term ECG Database [5]. Because I wanted to have all the parameters the same as in the article, I transformed all the signals so that the final sampling frequency was equal to 200 Hz (the signals from the first database had a sampling frequency of 250 Hz and the signals from the second database had a sampling frequency of 128 Hz). I then had to transform all the signals to the appropriate Matlab format. Below are parts of the code I used for transformation.

Listing 1. Code for transforming signals

```
xform -i 100 -a atr
wfdb2mat -r 100
```

After a successful data transformation, I started implementing the algorithm. Figure 1 is showing the raw signal of sample 100 from the database [4] - 10 seconds of a record.

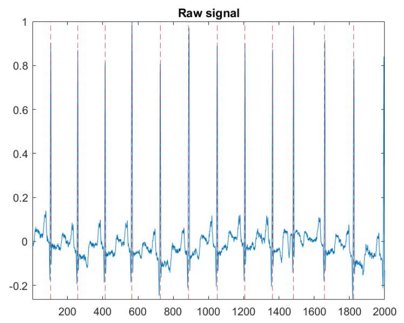

Figure 1. Raw signal from sample 100. This is the 10 seconds of signal with sampling frequency 200 Hz. Red vertical lines represents detected QRS complexes.

Bandpass filter

I first implemented a digital bandpass filter consisting of a high-pass and a low-pass filter. The filters are defined by the following equations

$$H(z) = \frac{(1-z^{-6})^2}{(1-z^{-1})^2}, \tag{1}$$

$$H(z) = \frac{(-1+z^{-16}+z^{-32})}{(1+z^{-1})}, \quad (2)$$

where equation 1 represents a transfer function for a low-pass filter and equation 2 represents a transfer function for a high-pass filter. The delay of the first filter is equal to 5 and the delay of the second filter is equal to 16 (we rounded it up from 15.5). Delay is very important as we will have to shift the detected beats on the transformed signal exactly by a delay to the left to get the correct positions.

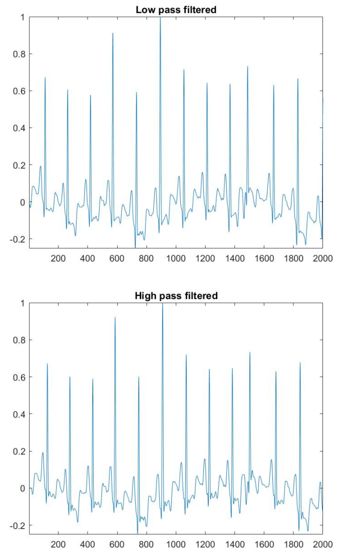

Figure 2. Signal after the low-pass and the high-pass filters.

Derivative and Squaring function

We need to differentiate a filtered signal to obtain slope information. Five-point derivative with transfer function

$$H(z) = (1/8T)(-z^{-2} - 2z^{-1} + 2z^{1} + z^{2}), \quad (3)$$

is used. T represents sampling period. The delay is 2.

After above transformation, the signal is squared (point by point). This transformation makes all data points positive. We can see from the second graph on the Figure 3 that there is no doubt that the QRS signal is very nicely seen. The only problem is that T - complex is right behind the QRS complex - if we look very closely in the peaks, there are 2. First one is QRS complex, the other one is T complex.

Moving-Window Integration

Purpose of this transformation is to obtain waveform feature information, which comes handy in distinguishing the QRS complex from the T-wave. The equation is

$$y(nT) = (1/N)(x(nT-(N-1)T)+...+x(nT)) \quad (4)$$

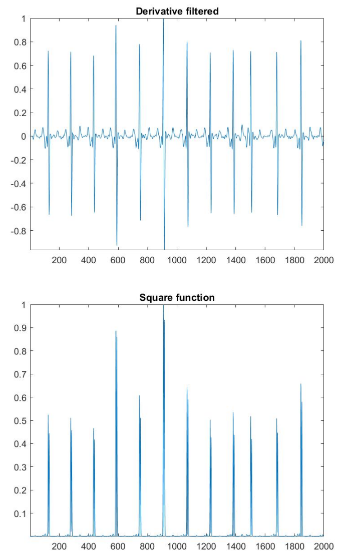

Figure 3. Signal after the derivative filter and after the squaring.

where N is the number of samples in the width of the integration window. For our sample rate of 200 samples/s, the window is 30 samples wide. This is a very important parameter, because if it is too narrow, some QRS will produce several peaks in the integration waveform - problems in the QRS detection process. On the other hand, too wide will merge QRS and T complex together and detection will not be accurate. This is also the reason, we transformed all the signals into appropriate form (appropriate N is already chosen). In the Figure 4 we see detected QRS complexes (red vertical lines). These lines coincide with local peaks. When the peak is detected, there is a condition that there is some time between the peaks - it is not physically possible for the QRS to happen so quickly one after another.

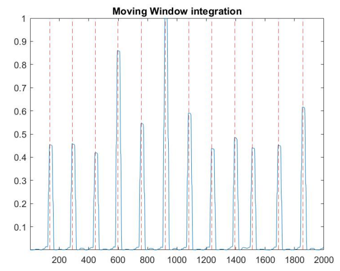

Figure 4. Signal after the Moving-Window Integration

Adaptive threshold

The thresholds automatically adapt over time to float above the noise. We have two thresholds - upper and lower. The lower threshold is used only when there is no signal above the upper threshold in some period - then we go back looking for a signal above the lower threshold. This period is predetermined in a way that a QRS complex should happen in that interval and it adapts over time (we look at the last eight beats). Also, the T-wave must be excluded when the RR time interval is between 200 ms and 360 ms. Here, the slope comes in handy. Appropriate literature was used for implementation of the adaptive threshold [6].

Results

We used the MIT-BIH database [4] and MIH-BIH long database [5] for evaluation. The table 1 contains the results - the algorithm was tested on 47 records, with 42 records half an hour long (MIT-BIT [4]) and 5 records 24 h long (MIT-BIH long [5]). Together, we tested the algorithm on 141 hours of ECG data. We used 2 metrics: sensitivity, which is defined as $\mathrm{Se} = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$ and positive predictivity $+\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP+FP}}$. The average sensitivity is 66,5 %, and the average positive predictivity is 97,6 %. Our algorithm works good on 'nice' records - records where a patient is without any serious illness that would have a major impact on QRS complexes - counterexample is record 14046, where the sensitivity is only 8 % and positive predictivity is 63 %. While implementing the algorithm, we noticed that the algorithm could be improved with a better way of detecting peaks. Our implementation is based on real time detection, so peak detection is not so accurate. For this purpose, we tested how the algorithm works when we use Matlab function "findpeaks" to find peaks with a certain QRS complex distance - we used this because it can't get any better than that. It turns out that the results are much better - on the same data, the sensitivity was 93,8 %, and the positive predictivity was 98,7 % - the last two columns of the table 1 represents the results with the usage of this function. We know that with this approach, the algorithm no longer works in real time because this function takes the entire record as an argument. With this approach it is easier to determine all the peaks.We also tested how changing different filters and parameters affect the results. The bandpass filter could be further improved, but we think it's OK for now as it works very well considering it's low complexity. Different powers instead of squaring the signal were also tested: $y(nT) = [x(nT)]^k$. Here we noticed that in different records the metrics sensitivity and positive predictivity change - in some it is better, in others it is worse. We got the best results with $k = 2.1$ (also absolute number of this) - sensitivity and positive predictivity improved by 0.2 percentage points. We also tried a different (general) filter for derivative: $H(z) = (1/T)(1 - z^{-1})$, but the results were not any better - the reason is that the original derivative filter is the best option, because of it's construction. It approximates an ideal derivative over the range where most information from QRS complex is needed.

Table 1. The performance of the algorithm.

Record	TP	FP	FN	**Se**	**+P**	*Se*	*+P*
100	1826	1	76	0,96	1,00	1,00	1,00
101	1437	0	86	0,94	1,00	1,00	1,00
102	1566	176	255	0,86	0,90	0,92	0,92
103	1506	1	223	0,87	1,00	1,00	1,00
104	909	13	948	0,49	0,99	0,97	0,99
105	1220	42	935	0,57	0,97	0,97	0,97
106	1285	4	411	0,76	1,00	0,98	1,00
107	833	44	951	0,47	0,95	0,97	1,00
108	979	26	501	0,66	0,97	0.85	0.97
109	1774	1	325	0,85	1,00	0,99	1,00
111	1112	6	664	0,63	0,99	0.90	1,00
112	845	1	1266	0,40	1,00	0,98	1,00
113	943	0	563	0,63	1,00	1,00	1,00
115	1507	0	130	0,92	1,00	0,99	1,00
116	1622	6	395	0,80	1,00	0,98	0,99
117	499	13	785	0,39	0,97	0,97	1,00
118	1428	4	488	0,75	1,00	0,99	1,00
121	887	1	673	0,57	1,00	0,97	1,00
122	1373	1	681	0,67	1,00	1,00	1,00
123	1232	0	37	0,97	1,00	1,00	1,00
124	1215	8	152	0,89	0,99	0,97	1,00
201	1072	41	449	0,70	0,96	0,92	0,91
202	1334	0	537	0,71	1,00	0,99	1,00
205	1859	25	342	0,84	0,99	0,98	0,99
208	1546	96	891	0,63	0,94	0,98	1,00
209	1265	13	1254	0,50	0,99	0,99	1,00
210	1378	29	826	0,63	0,98	0,93	0,99
212	1143	4	1142	0,50	1,00	0,99	1,00
213	1490	15	1210	0,55	0,99	0,99	0,99
214	1380	33	499	0,73	0,98	0,99	1,00
217	858	44	987	0,47	0,95	0,99	1,00
219	1312	0	461	0,74	1,00	1,00	1,00
220	991	0	703	0,59	1,00	1,00	1,00
221	1583	25	437	0,78	0,98	0,99	1,00
222	1342	11	774	0,63	0,99	0,86	0,99
223	1324	1	875	0,60	1,00	0,96	1,00
228	466	21	1237	0,27	0,96	0,17	0,97
230	1653	9	206	0,89	0,99	1,00	1,00
231	993	0	285	0,78	1,00	0,99	0,99
232	971	5	514	0,65	0,99	0,99	1,00
233	1372	232	1189	0,54	0,86	0,92	0,96
234	1609	0	682	0,70	1,00	1,00	1,00
14134	32142	124	17180	0,65	1,00	0,99	1,00
14172	38606	201	27094	0,59	0,99	0,99	1,00
14157	76019	717	11748	0,87	0,99	0,99	1,00
14149	84741	0	59564	0,59	1,00	0,98	1,00
14046	9608	5659	105284	0,08	0,63	0,10	0,77
47 records	239646	2195	109698	0,665	0,976	0,938	0,987

Discussion

The algorithm seems to be very well designed, as there is a simple logic behind it - the fact that in times when technology was not yet so powerful, the whole algorithm could be implemented on 8-bit microprocessor (Zilog Z80). We were impressed by how the threshold for QRS complex detection adapts over time. It flows over the noise nicely. It also enables the look-back, when there is no QRS complex found in a expected time. As we mentioned before, our algorithm does not work the best on the records where there are a lot of non-normal heartbeats. We think the reason is that some heartbeats are so different than normal heartbeats - the algorithm then fails to recognize it a QRS complex. We think that with today's technology we could greatly improve the algorithm. A very good tool that could be used is a neural network. It would act as a classifier - it would tell us the information about the cycle: is it a QRS complex, P wave or T wave etc. This has already been implemented and documented - the 1-D CNN structure has been used - see [7]. The authors state that after the evaluation the algorithm on the same database [4], the following results were obtained: 99.81 % sensitivity and a 99.93 % positive predictivity ([7]). These are really good results, as the algorithm detects QRS complexes well even in records, which causes problems with our algorithm. I think there is still a lot of room for improving the algorithm by combining different approaches. Our algorithm could be refined with some other approaches as described above.

References

[1] J. Pan and W. J. Tompkins. A real-time qrs detection algorithm. *IEEE Transactions on Biomedical Engineering*,

BME-32(3):230–236, 1985.

[2] Moody GB and Mark RG. The impact of the mit-bih arrhythmia database. *IEEE*, Eng in Med and Biol 20(3):45–50, (May - June 2001).

[3] Goldberger A., Amaral L., Glass L., Hausdorff J., Ivanov P. C., Mark R., and Stanley H. E. Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals. circulation. *Circulation [Online]*, 101(23):e215–e220, 2000.

[4] Mit-bih arrhythmia database. https://physionet.org/content/mitdb/1.0.0/. Accessed: 2020-12-07.

[5] Mit-bih long-term ecg database. https://www.physionet.org/content/ltdb/1.0.0/. Accessed: 2020-12-07.

[6] Hooman Sedghamiz. Matlab implementation of pan tompkins ecg qrs detector., 03 2014.

[7] M. Šarlija, F. Jurišić, and S. Popović. A convolutional neural network based approach to qrs detection. In *Proceedings of the 10th International Symposium on Image and Signal Processing and Analysis*, pages 121–125, 2017.