

Improving the QRS detection for one-channel ECG sensor

Ervin Domazet* and Marjan Guse

Faculty of Computer Science and Engineering, University Ss Cyril and Methodius, Skopje, North Macedonia

Abstract. We analyzed several QRS detection algorithms in order to build a quality industrial beat detector, intended for a small, wearable, one channel electrocardiogram sensor with a sampling rate of 125 Hz, and analog-to-digital conversion of 10 bits. The research was a lengthy process that included building several hundred rules to cope with the QRS detection problems and finding an optimal threshold value for several parameters. We obtained 99.90% QRS sensitivity and 99.90% QRS positive predictive rate measured on the first channel of rescaled and resampled MIT-BIH Arrhythmia ECG database. Even more so, our solution works better than the algorithms for the original signals with a sampling rate of 360 Hz and analog-to-digital conversion of 11 bits.

Keywords: QRS detection, beat detector, electrocardiogram, wearable sensors, rescale, resample, optimization

1. Introduction

Advances in the Internet of Things (IoT) field have encouraged researchers to intensify their focus on Electrocardiogram (ECG) processing, especially for wearable devices. This field has become a popular research topic in biomedical engineering [1]. Our primary focus is, on building a quality industrial QRS detector for a wearable ECG sensor that is significantly better than existing robust algorithms for QRS detection, which are inconvenient for mobile devices with limited resources [2].

The performance of the QRS detector is evaluated by calculating how many real QRS peaks are found (QRS sensitivity, denoted as Q_{SE}), and how many of those detected QRS peaks are real beats (QRS positive predictive rate, denoted as Q_{+P}). The default testing database was the Massachusetts Institute of Technology – Beth Israel Hospital Arrhythmia database (MITDB) [3] with 48 records of 30 minute ECG measurements. The original signals are sampled with 360 Hz and 11 bit AD conversion. Our research target is a QRS detector for signals using a sampling frequency of 125 Hz and 10 bit AD conversion.

One of the most cited papers for QRS detection built for small devices with limited resources is the Pan and Tomkins algorithm [4]. Its robustness lies in the fact that it runs fast enough to be used in real time and can cope with noisy signals. However, the performance of this algorithm depends on bit resolution in the AD conversion. In our case, when using one channel ECG sensor and smaller sampling

*Corresponding author: Ervin Domazet, Faculty of Computer Science and Engineering, University Ss Cyril and Methodius, Skopje, North Macedonia. E-mail: ervin_domazet@hotmail.com.

31 frequencies, its performance was not satisfactory, especially for signals with smaller amplitudes. It was
32 not a good solution for us because of these factors.

33 Another alternative is the Hamilton algorithm [5]. Compared to the Pan and Tompkins algorithm, it is
34 quite similar but uses different filters and decision rules. It is a stable solution, but, still is unable to cope
35 with small amplitudes, or variations in consecutive amplitude levels, especially, when using a smaller
36 bit resolution in AD conversion.

37 Physionet.org [6] is a very comprehensive resource where one can find several QRS detection algo-
38 rithms, including *Wavedet*, *gqrss*, *wqrss*, and *sqrss*. They represent simple and fast algorithms demanding a
39 small number of resources and obtain high sensitivity and positive predictive rate values. However, they
40 lack the beat classification, and the obtained sensitivity and the positive predictive rate are also consid-
41 ered to be lower than the demands of a quality industrial QRS detector for a wearable one channel ECG
42 sensor.

43 After these initial efforts, the attention of researchers gradually focused on developing more sophisti-
44 cated QRS detection algorithms, including Machine Learning and other methods, as described in Sec-
45 tion 2 (related work). Although some of the new approaches achieved better performance, they generally
46 require computationally intensive algorithms, not suitable for smartphones that collect continuous ECG
47 data from wearable sensors.

48 In our research, we improve Hamilton's algorithm [5] in order to make it efficient for industrial appli-
49 cation. The improvement was a rather long process due to the exponential nature of the effort to improve
50 the algorithm. The closer you are to the margin of 100% the more effort is needed for a very small
51 improvement of the performance. We introduced several hundred rules to cope with the identified prob-
52 lems in QRS detection, and several thousand tests to tune parameters, and threshold values for identified
53 solutions. Some threshold values obtained good performance on some test file while performing badly
54 on others. When we fine-tuned some parameters, it so happened that some of the rules did not work on
55 other test datasets, which was even more challenging.

56 Finally, the results proved that we have realized our goals. We reached higher values than all other
57 published results, though we were working with almost three times less sampling frequency and half of
58 the AD conversion bit resolution.

59 The structure of this paper is organized as follows. Section 2 discusses the related work in the area of
60 QRS detection algorithms and Section 3 presents the background. Analysis of problems are presented in
61 Section 4, and our approach to improving the algorithms in Section 5. The results from the experiments
62 are evaluated in Section 6 and compared with other solutions. Section 7 is devoted to conclusions and
63 future work.

64 **2. Related work**

65 QRS detection algorithms generally follow the same routine [7], starting with Digital Signal Process-
66 ing (DSP) filtering that eliminates noise and baseline wander. Then the output is matched to a set of
67 thresholds. The algorithms mainly differ in the way they calculate the threshold values and implement
68 decision rules, which is another layer implemented after thresholding [8].

69 Long ECG recordings are generally occupied with noise in the form of a subtle deviation of the heart
70 rhythm. In addition, signal quality changes to alternating changes in the wave amplitude. Unless noise
71 is eliminated, detection of such QRS complexes becomes difficult, and decreases the accuracy [9]. It is
72 important to state that any algorithm designed for quality industrial QRS detection must be adaptable to
73 any type of noise.

74 There are several survey papers [9,10] which give a comprehensive overview of popular QRS detection
75 methods. In summary, published QRS detection algorithms are based on the following techniques:

- 76 – *Differentiation (derivation)*, where the difference between the current and previous samples is calcu-
77 lated, as a way of identifying the slope, and then it is compared to a given threshold value including
78 the Pan & Tompkins algorithm [4], Hamilton's algorithm [5], or other relevant approaches [11–13];
- 79 – *Pure DSP algorithms*, where fundamental DSP filters with different characteristics are combined to
80 produce a bandpass filter, with the aim to eliminate noise, and filter the signal so that a threshold
81 will determine the beats [14–18];
- 82 – *Pattern Recognition algorithms*, where the signal data is matched with predefined patterns and a
83 waveform is detected in case of similarity within given constraints of the amplitude and slopes [19–
84 23];
- 85 – *Neural Network*, Multilayer Perceptron (MLP), Radial Basis Function (RBF), and Learning Vector
86 Quantization (LVO) networks are used to adaptively predict the location of the next peak [21,24–
87 27];
- 88 – *Digital Wavelet Transformation (DWT)*, where the signal is decomposed to a certain level of scales,
89 and then recomposed, which effectively reduces noise. Then a threshold is applied to select proper
90 peaks [28–32];
- 91 – *Genetic Algorithms* are used to optimize the preprocessing polynomial filter. The ECG signal is
92 compared to an adaptive threshold and the parameters are optimized with a genetic optimization
93 approach [33];
- 94 – *Hidden Markov Models (HMM)*, used to train the probability function varying according to the
95 hidden Markov chain, and then the model predicts the current state, which could be QRS complex.
96 P and T waves can also be computed [34–36];
- 97 – *Hilbert Transform*, where Hilbert transform of ECG signal is calculated by Fast Fourier Transform
98 (FFT), and that is used for calculating the signal envelope [37–39]; and
- 99 – *Phasor Transformation*, where each ECG sample is converted into a phasor to correctly manage P,
100 and T waves, by definition having lower amplitudes than an R-peak with low computational cost,
101 and then compares it against a threshold [40].

102 High performance QRS detector directly affects the amount and the quality of valuable information
103 on ECG. QRS detection is the initial step for further ECG analysis.

104 3. Background

105 In this section, we will explain the evaluation metrics and give an overview of the original Hamilton's
106 QRS detector algorithm.

107 3.1. Performance measures

108 The benchmarks used in our testing methodology are the same used in the IEC 60601-2-47 standard
109 for particular requirements for the safety, including an essential performance of ambulatory electrocar-
110 diographic systems, and ANSI/AAMI EC57:2012 for Testing and Reporting Performance Results of
111 Cardiac Rhythm and ST Segment Measurement Algorithms. These standards use the MIT-BIH ECG
112 arrhythmia database [3], and the American Heart Association's (AHA database) [41].

113 MITDB contains half-hour ECG recordings for 48 anonymized persons, and only 44 records exclude
114 those that contain paced beats. These recordings are publicly available on the physionet.org web site [6].

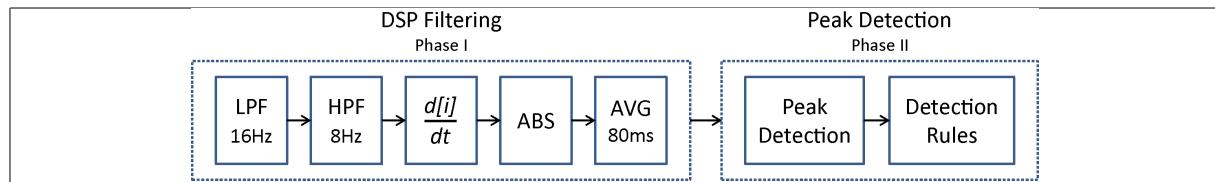


Fig. 1. Hamilton's approach for QRS detection.

The recording frequency is 360 samples per second, per channel, with a 11-bit resolution. Even though each recording contains two-channels, we used the first channel, identified as ML II, in most of the records.

In addition, we follow the requirements according to the standard IEC 60601-2-47:2012 for medical electrical equipment, particularly, requirements for essential performance of ambulatory electrocardiographic systems. According to these requirements, any calculated peak is considered as detected if it is at most 150 ms away from the real beat.

A detected QRS is denoted to be True Positive (TP), if the QRS detector has found a QRS closer than 150 ms from the one which is annotated. A False Negative (FN) is a missed QRS, or if the QRS detector has found a QRS outside the 150 ms perimeter, while a False Positive (FP) is an erroneously detected QRS (extra found).

The commonly used performance measures are sensitivity and positive predictive rate, calculated by Eq. (1).

$$SE = \frac{TP}{TP + FN} + P = \frac{TP}{TP + FP} \quad (1)$$

In addition, to find an optimal value of a parameter, we provided a lot of test experiments, and calculate the number of *Total Errors* by Eq. (2), as a sum of errors denoted by *FP* and *FN*. The smaller the errors are, the better performance is achieved.

$$Errors = FP + FN \quad (2)$$

3.2. Analysis of Hamilton's algorithm

EP Limited's open source software for arrhythmia detection serves as a basis for this research [5, 42]. It has a complete C-code implementation of Hamilton's algorithm, with three different detectors, and a simple beat classifier. Two of the detectors are for general-purpose, whereas the third one is for environments with a small amount of memory.

Algorithmic details are theoretically provided in their original work [5]. Figure 1 represents the conceptual level for the two phases, and the high-level steps conducted for each of them.

The algorithm starts with a low pass filter (LPF) with a cutoff frequency of 16 Hz. Then, the signal is passed through a high pass filter (HPF) of 8 Hz. Both filters have the effect of a band pass filter (BPF). Then, the slope of the signal is calculated with a differentiation method ($\frac{d[i]}{dt}$), followed by the calculation of an absolute value (ABS). The final step (AVG) consists of calculating a time average over an 80 ms moving window.

After eliminating the noise in the DSP filtering phase, the algorithm continues with the peak detection phase. It already has two thresholds for the AVG signal, classified as a

- static threshold with a fixed value, and
- dynamic adaptive threshold (DAT) which is affected by the amplitudes of the latest peaks.

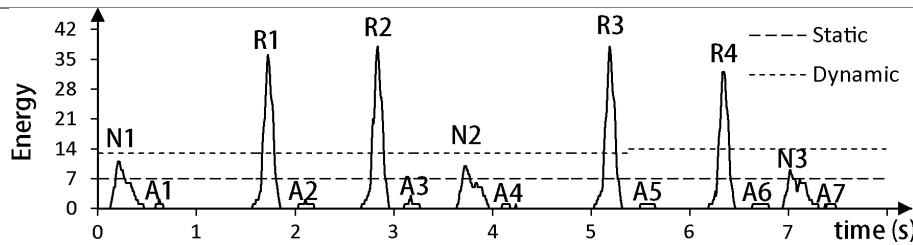


Fig. 2. Detecting artifacts, noise and real peaks based on values of static and dynamic thresholds in the original Hamilton's algorithm presented on signal extract over MITDB record 124 (1046 sec).

The original algorithm sets the static threshold (*STHR*) at value *MIN_PEAK_AMP* = 7. A general rule of thumb is that a lower value of the static threshold will find more peaks, but also detect lots of artifacts. On the other hand, a higher static threshold value yields fewer peaks, but a smaller number of artifacts.

When a new local peak is found, the *dynamic adaptive threshold* is calculated by taking the mean values for real peaks and noise peaks into account. The mean value is computed by Eq. (3).

$$\text{mean} = \frac{\sum_{n=1}^8 X_n}{8} \quad (3)$$

Let the mean value for real beats and noise peaks be denoted as *qmean* and *nmean* respectively, and also *TH* be the constant multiplier (with a default value of 0.5125), then the DAT is calculated by Eq. (4).

$$\text{DAT} = \text{nmean} + (\text{qmean}-\text{nmean}) * \text{TH} \quad (4)$$

When a new local maximum is detected by calculating an AVG value, both of the thresholds are compared to this value. If the value is higher than the static threshold, then it is considered to be a *potential peak*, otherwise, it is an *artifact*. It is classified as a *noise peak* if the calculated value is lower than the dynamic peak, and as a *real beat* if it is higher than the dynamic adaptive threshold.

Figure 2 presents both dynamic and static thresholds. Note, that detected peaks A1, A2, ..., A7 are categorized as artifacts (smaller than the static threshold), and R1, R2, R3 and R4 as real beats. N1, N2, and N3 are considered noise peaks since the local maxima of each label are smaller than the dynamically calculated threshold.

4. Identification of performance issues

A discrepancy in peak amplitudes may introduce bad detection. We have identified two cases when this happens:

- a sequence of low-amplitude peaks after an isolated high-amplitude peak;
- an isolated low-amplitude peak after a sequence of high-amplitude peaks.

Furthermore, apart from fine-tuning the threshold value, and expecting lower performance on cropped signals, we analyzed the segments where the algorithm showed lower sensitivity, and specificity even though the signal was not contaminated with noise. The conclusion was that lower performance was obtained for specific segments, and the problems can be classified as:

- a mixture of low and high-amplitude peaks;
- artifact elimination; and
- wrong R-peak location.

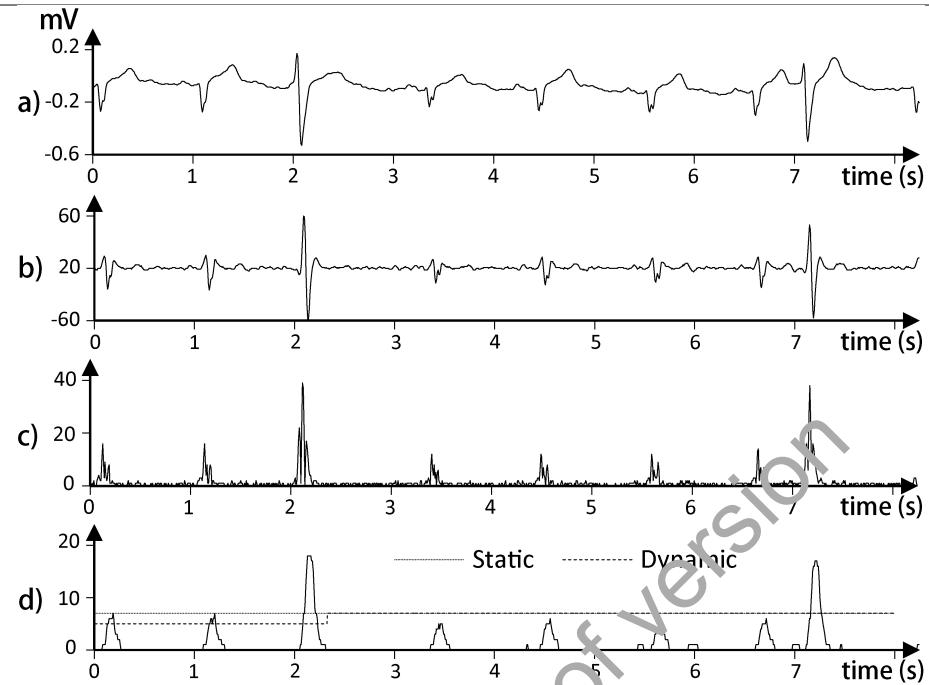


Fig. 3. Signal extracts of executing the Hamilton algorithm over the MITDB record 114 (240 sec) a) Original ECG signal; b) Output after bandpass filtering; c) Output after differentiation and absolute calculation; d) Output after average over an 80 ms window.

174 4.1. Bad detection of low-amplitude peaks

175 Figure 3 presents the case when an isolated high-amplitude peak is preceded and followed by a sequence of low-amplitude peaks. An 8-second segment extract of MITDB record 114 is displayed including the original signal and outputs after executing each of the processing steps BPF, ABS, and AVG.

178 Figure 3d identifies the static and dynamic thresholds and shows the case where the beats between the two high amplitude are considered as artifacts, although they should be real QRS beats.

180 The reason for bad detection of low-amplitude peaks after high-amplitude peaks is primarily due to the high level of static threshold. Even if one makes a correction by decreasing the static threshold value to include these peaks, there will still be a problem regardless of the fact that the peak will be treated as a candidate, and the dynamic threshold check will be applied. This is because the high-amplitude peak will increase the dynamic threshold value caused by the calculation of the mean, and so the peaks will be classified as noise peaks.

186 4.2. Isolated peaks after sequences of high-amplitude peaks

187 High-amplitude peaks directly affect the calculation of *dynamic adaptive threshold*. It is recalculated 188 each time a new local maximum is found with an amplitude higher than the *static threshold*. In this case, 189 it is considered as a potential peak, and the values lower than the dynamic threshold are considered as 190 a noise peak, while the others as real QRS peaks. The original algorithm buffers the latest eight peak 191 amplitudes and calculates a DAT value by Eq. (4).

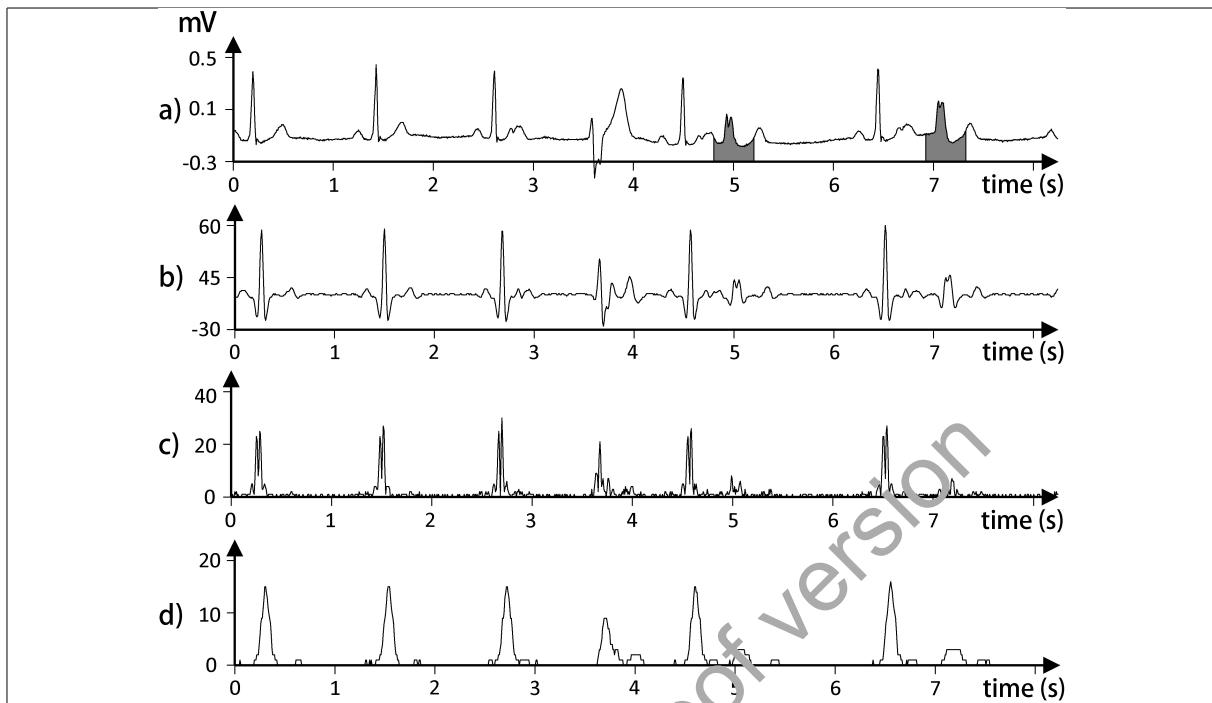


Fig. 4. Signal extracts of executing the Hamilton algorithm over the MITDB record 201 (424 sec) a) Original ECG signal; b) Output after bandpass filtering; c) Output after differentiation and absolute calculation; d) Output after average over an 80 ms window.

An extensive analysis of the MITDB record 201 shows too many misses, especially in cases of aberrated atrial premature beats (classified as *1 beat*) as illustrated in Fig. 4. Two of the beats highlighted by default cannot be captured due to the *dynamic adaptive threshold* and mean calculation, since most of the latest beats have a high amplitude. In this case, neither the static nor the dynamic adaptive threshold will work. The example is highlighted in Fig. 5 by peaks C and D, which should be classified as QRS peaks, but they are detected as artifacts because their value is lower than the static threshold.

4.3. Classification of artifacts

Dynamic adaptive thresholding identifies noise and real peaks. Although in most cases, the dynamic adaptive threshold reacts properly, there are still cases where a noise peak is incorrectly calculated as real. Correct classification of artifacts is of primary importance for a quality industrial QRS detector.

Increasing the *static threshold* directly decreases the number of artifacts, however, this drastically increases the missed beats. This also applies to the *dynamic adaptive threshold*. To find an optimum of the static and dynamic peaks means to search for a comprise that would reach high values of both sensitivity, and positive predictive rate.

An example is illustrated in Fig. 5. Peaks labeled as A, C, D, E and G are low amplitude peaks. However, peaks labelled as B and F are artifacts. With the default threshold, A and E peaks are considered as candidates for QRS, whereas the rest are considered as artifacts. For this particular case, decreasing the static threshold to 3 would catch all of the real peaks, but, will consider artifact peak F as a peak. On the other hand, keeping the static threshold at 4 will only detect G as a peak, and the other peaks will remain artifacts again.

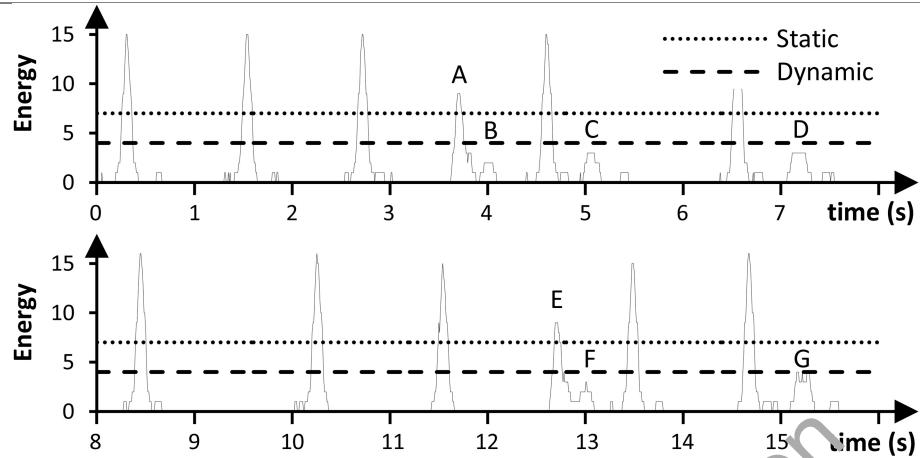


Fig. 5. Static and dynamic threshold values on the output after average over an 80 ms window on MITDB record 201 (424 sec).

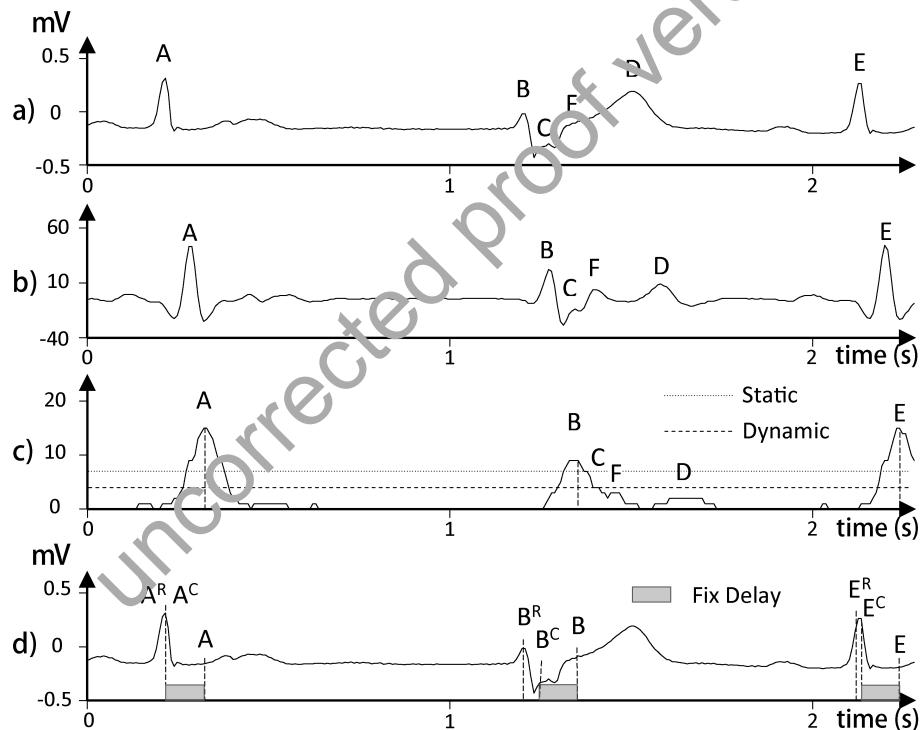


Fig. 6. Signal extracts and R-peak detection over the MITDB record 201 (426.4 sec): a) Original signal and local peaks A , B , C , D and E ; b) Output after bandpass filter; c) Output after average over an 80 ms window; d) Real and calculated R-peak locations with constant delay introduced by the filter.

212 4.4. Calculation of R-peak location

213 One of the issues in executing the original Hamilton algorithm is the proper detection of a QRS peak.
 214 Fig. 6 illustrates such a case, where local maxima are labeled with A , B , C , D , and E . Another peak

215 appears in the output after bandpass filter denoted by F , as seen in Fig. 6b). The proximity of marked
216 peaks B , C , and F cause two local peaks on the output of 80 ms average window, marked as B and C
217 on Fig. 6c. When static thresholding is applied to the time average over an 80 ms window, C , F , and
218 D peaks are detected to be artifacts, whereas A , B , and E are identified as potential peaks. Since the
219 dynamic threshold is below the static, these beats are classified as real.

220 Note that the filter makes a constant delay, which is deducted from the location of peak values (as
221 displayed on the signal average output). It correctly determines properly the previous QRS peak A , and
222 the next QRS peak E , but makes a mistake in determining the peak B .

223 The original Hamilton's algorithm detects the location of R-peaks to be at points A^C , B^C , and E^C ,
224 even though they have different real peak locations A^R , B^R , and E^R .

225 Hamilton's algorithm detects a peak based on a calculated amount of delay [42]. Although filters
226 produce a fixed delay, the author has pointed out that the detection delay can easily vary from 395 ms
227 to 1 sec, depending on the heart rate and detection rule. It is important to note that the *search back*
228 *detection* method [42] *can produce a fixed delay* if the search back algorithm does not report any local
229 peaks. Thus, we can conclude that the original Hamilton's algorithm provides the best likely position of
230 the R-peak, however, it is not always exact. This problem can particularly increase the total number of
231 FP's. This particular case is observed in almost all MITDB records. Even though the difference is not so
232 big, there are cases where the difference between the real and detected location is higher.

233 5. Algorithm improvement

234 Increasing the algorithm performance is directly related to fine-tuning thresholds and algorithm im-
235 provements.

236 5.1. Improving the detection of low-amplitude peaks

237 A careful analysis shows that the step (ABS) executed prior to time average, will not be able to
238 cope with the bad performance in detection of low-amplitude peaks. This is especially crucial in cases
239 when the signal is a mixture of one high amplitude beat, and then followed by several beats with low
240 amplitudes. We used the idea introduced in the Pan Tompkins algorithm [4] to square the signal, instead
241 of calculating the absolute value.

242 Figure 7 presents a case where a combination of a square mode and the optimized static threshold
243 will improve the detection of low energy peaks. The peaks labelled as A , C , E , F , H , and I are real R
244 peaks, whereas B and D are artifacts. The original algorithm, which uses the calculation of an absolute
245 value and average along with the static threshold, is not able to classify F , and I as real peaks. However,
246 the square mode and signal average in combination with a new (smaller) static threshold, is able to
247 detect that there is sufficient energy for a potential peak. The square average signal also marks B , and
248 D as potential peaks. Such peaks can be reduced with the dynamic threshold or by introducing rules for
249 artifact detection.

250 Nevertheless, there are side effects, especially in the calculation of the dynamic adaptive threshold.
251 This threshold increases due to increased amplitudes, and it becomes slightly difficult to adapt to sudden
252 changes in the amplitudes.

253 This operation behaves as an important amplifier especially if it followed by a calculation of an average
254 over an 80 ms moving window. It shows that this copes better with the identified problems. However,

10

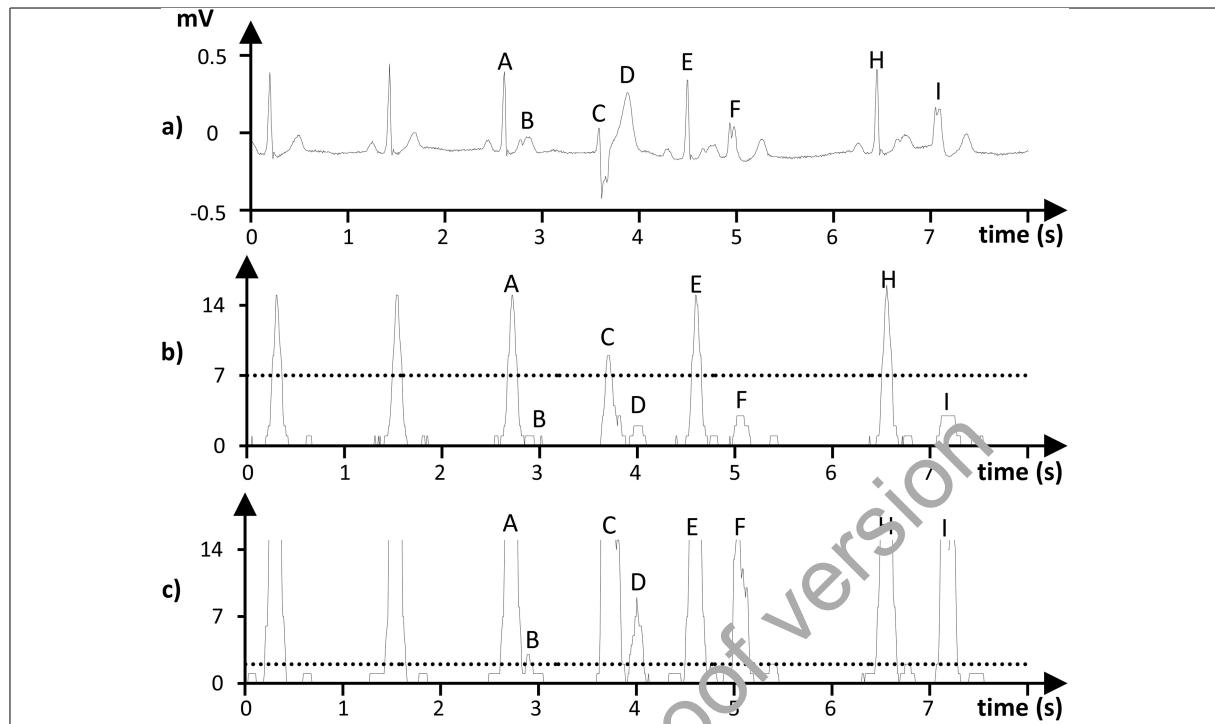
E. Domazet and M. Guse / Improving the QRS detection for one-channel ECG sensor

Fig. 7. Signal extracts of executing the Hamilton's algorithm over the MITDB record 201 (424 sec): a) Original signal; b) Output after average over a 80 ms window using the original Hamilton's algorithm; c) Output after square average over a 80 ms window with the optimized static threshold.

255 this is not enough, since this algorithm cannot perform with static threshold values, and needs dynamic
 256 calculation by other rules.

257 We have conducted several tests to experiment with threshold values $STHR$ from 2 to 50, to find the
 258 optimal threshold value for which the number of errors is minimal. The left part of Fig. 12 presents the
 259 false detections for the conducted experiment. The performance of the algorithm gradually decreases
 260 as the threshold increases, mainly due to the high values of FP. The best performance is obtained for
 261 $STHR = 2$. We noticed a decreased number of FN, which gives an idea for how to get better performance
 262 if we decrease the number of FPs through other methods.

263 5.2. Improving the calculation of the R-peak location

264 Since the Hamilton algorithm only gives an approximation where the peak is, one way to reduce the
 265 number of FP's that occur due to the determination of a proper R peak location, is to search the real
 266 peak in near proximity. The idea is to search the most convenient local maximum by analyzing the
 267 noise-eliminated output after bandpass filtering.

268 The original Hamilton's algorithm will determine the best approximate for the location of the R-peak.
 269 This is the starting point for the search of the local maximum, within a range, found $SearchL$ ms to the
 270 left, and $SearchR$ ms to the right of the approximated R-peak location. Once the local maximum is found
 271 on the output of the bandpass filter, we continue to find the local maximum on the actual signal, though
 272 the range for searching will be limited to 48 ms.

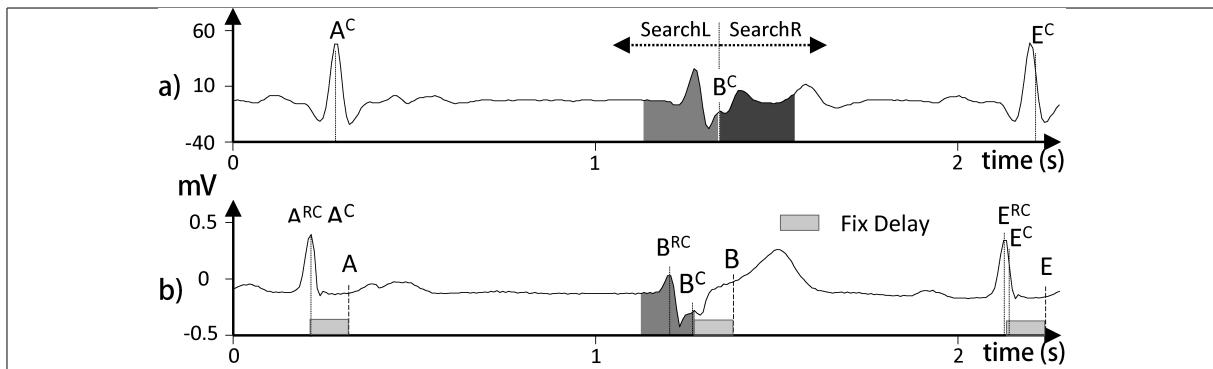


Fig. 8. Proper calculation of the R-peak location on the MITDB record 201 (426.4 sec): a) Calculated R-peak location B^C and the search intervals over the output of bandpass filter; b) The re-calculated R-peak location B^{RC} on the actual signal.

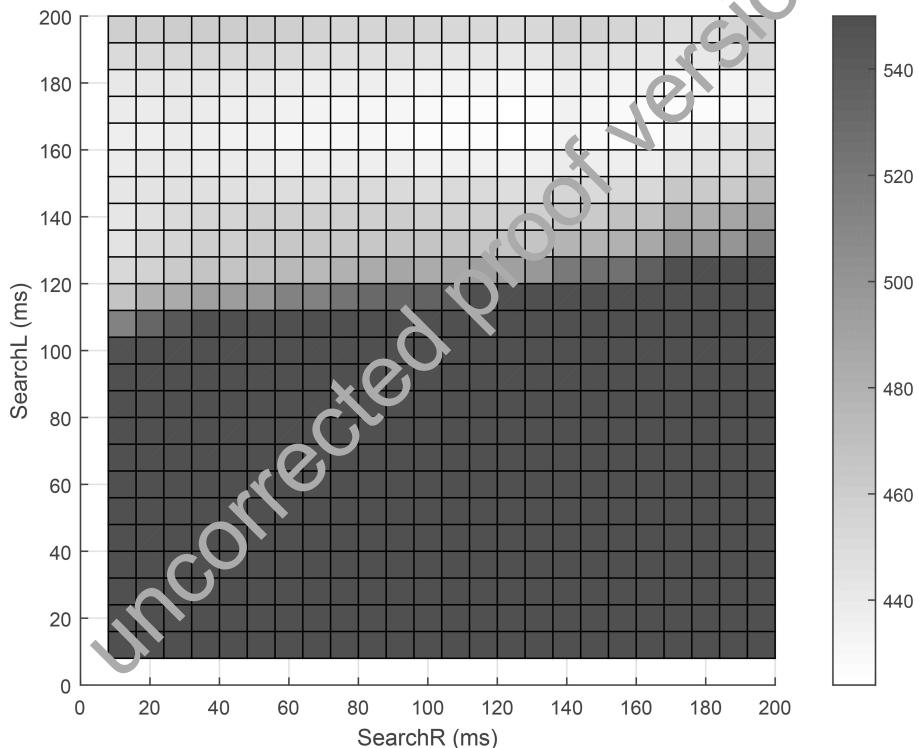


Fig. 9. False detections improving the calculation of the R-peak location.

273 Figure 8 illustrates the basic steps of this improvement algorithm in the example presented in Fig. 6.
 274 The corresponding search segments are marked on the original signal. The original Hamilton's algorithm
 275 detects the local peak B^C , and our improvement algorithm finds B^{RC} to be the real location.

276 We realized another experiment to locate the optimal values for $SearchL$, and $SearchR$. False detec-
 277 tions for thresholding values are plotted on a surface graph presented in Fig. 9, for different values for
 278 $SearchL$, and $SearchR$, using the static threshold value $STHR = 4$. The best results are obtained when
 279 $SearchL = 160$ ms, and $SearchR = 120$ ms.

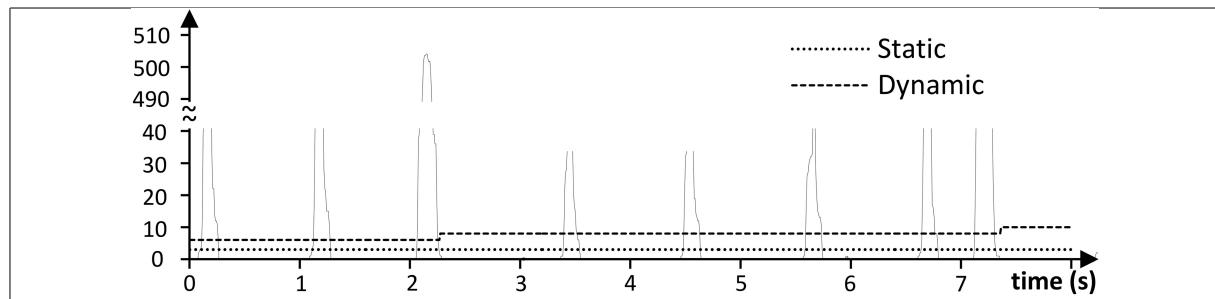


Fig. 10. Effect of a dynamic threshold to detect peaks after average of squared values over an 80 ms window over the MITDB record 114 (240 sec).

280 5.3. Improving the detection of low-amplitude peaks after sequences of high-amplitude peaks

281 Two functions, denoted as *mean* Eq. (3) and *thresh* Eq. (4), play a crucial role in the original QRS
282 detection algorithm.

283 We proposed a change in the mean function in order to alleviate the effect of high amplitude complex
284 proceeded by a significantly lower one. Instead of calculating the mean of the last 8 peaks, our algorithm
285 considers only half of this value when the peak amplitude is higher than the dynamic threshold (*DTHR*)
286 and the exact value in all other times, as defined by Eq. (5). This prevents a linear increase in the threshold
287 especially for high amplitude signals and solves the identified problem.

$$288 \text{mean} = \frac{\sum_{n=1}^8 \begin{cases} X_n, & \text{if } X_n < DTHR. \\ \frac{X_n}{2}, & \text{otherwise.} \end{cases}}{8} \quad (5)$$

288 In addition, we changed the *thresh* method. Previously, the calculated threshold was multiplied by the
289 constant *TH* = 0.3125. Since the square mode filter is used, we updated the multiplication constant by
290 its square, i.e. *TH* * *TH*. Thus the new DAT calculation is defined by Eq. (6). Both these interventions
291 enabled the detection of such beats.

$$292 \text{DAT} = n\text{mean} + (q\text{mean}-n\text{mean}) * TH^2 \quad (6)$$

292 A good performance is achieved on both cases with:

- 293 – a sequence of low-amplitude peaks after an isolated high-amplitude peak;
- 294 – an isolated low-amplitude peak after a sequence of high-amplitude peaks.

295 Figure 10 illustrates the improvement idea for the example presented in Fig. 3, where a sequence of
296 low-amplitude peaks is followed by a high-amplitude peak, which will increase the dynamic threshold
297 to a value where all consequent low-amplitude peaks are marked as noise peaks.

298 The figure demonstrates the output of an average of an 80 ms window realized on squared (not ab-
299 solute) values along with the static, and dynamic thresholds. Note that the detected potential peaks and
300 their absolute values presented in Fig. 3 are smaller than the original dynamic threshold value.

301 The effect of applying the new way to calculate the dynamic threshold can be also observed in Fig. 11,
302 presenting isolated, low-amplitude peaks after a sequence of high-amplitude peaks. The new minimum
303 threshold detects 17 candidate peaks, whereas the original algorithm with default dynamic threshold is
304 not able to capture the peaks labeled as *C*, *D* and *G*. The newly optimized threshold is able to capture
305 all 15 peaks correctly, and also classify *B*, and *F* peaks as noise.

Table 1
Parameter list for artifact detection rules

Parameter	Description
C	Current detected peak
P	Previous detected beat
CH	Current peak time average height
PH	Previous beat time average height
RR	Current beat to peak interval in ms
TAx	Time in ms optimizing Ax, $x \in \{0, 1, 2\}$
THRAx	Parameter optimizing Ax, $x \in \{1, 2\}$
MSxxx	xxx ms interval

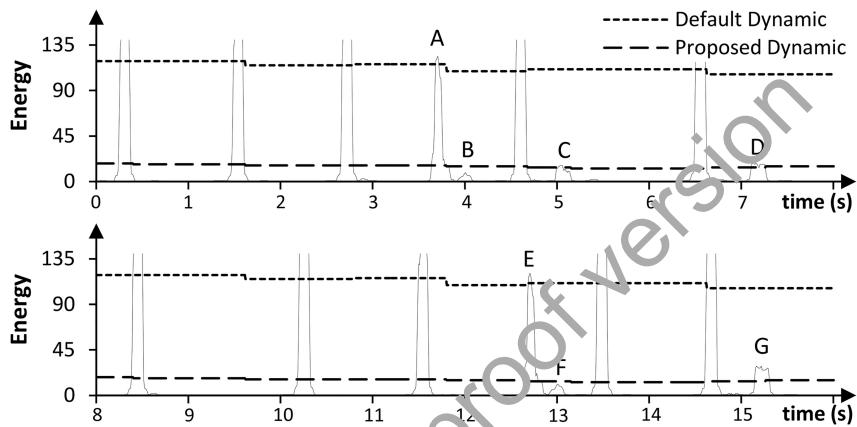


Fig. 11. Effect of a dynamic threshold to detect peaks after average of squared values over an 80 ms window over the MITDB record 201 (424 sec).

306 We have conducted a lot of experiments to determine the optimized value for the *DTHR* threshold
 307 value. The test cases included testing the threshold values of 100 up to 400. The middle part of Fig. 12
 308 presents false detection as a function of the threshold values. Threshold values near 200 are the best
 309 candidates for the most effective performance.

310 5.4. Improvement of artifact elimination

311 In the final step, we introduce a new *Classification* phase. The aim of this is to classify whether the
 312 calculated beat is real or an artifact. Three important decision rules decide whether the peak is an artifact.
 313 If none of these rules are satisfied, the beat is calculated as a real peak.

314 From the preliminary analysis, we observed that artifacts generally follow a real beat and are closer
 315 than 320 ms away. The second important issue is that an artifact obviously has lower energy when com-
 316 pared to the previously detected beat. The original Hamilton's algorithm eliminates artifacts closer than
 317 195 ms. Our findings show that this value can also be optimized. Table 1 describes some parameters used
 318 in our optimization approaches. We introduce the following optimization rules for artifact elimination:

$C = \text{Artifact if}$

319 **A0:** $RR \leq TA0 = MS250$

320 **A1:** $RR \leq TA1 = MS260 \& PH/CH > THRA1$

321 **A2:** $RR \leq TA2 = MS320 \& PH/CH \geq THRA2$

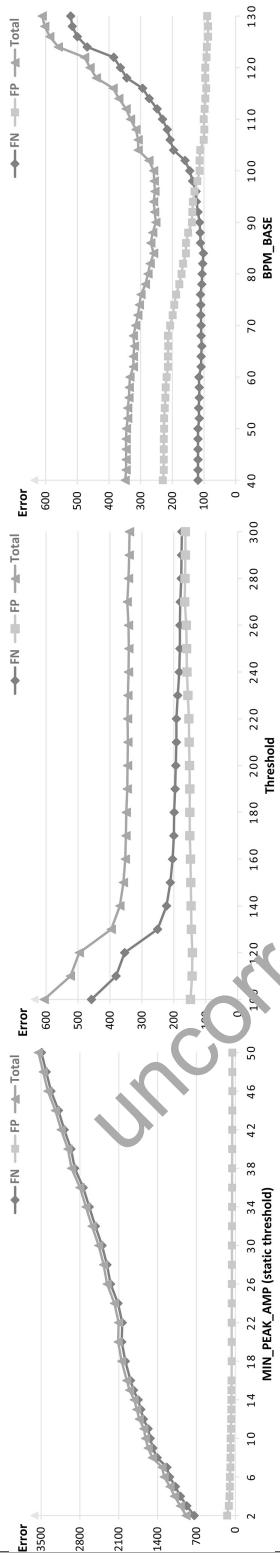


Fig. 12. False detections of optimization approaches for low amplitude peaks (left); sequences of high-amplitude peaks (middle), beat rate impact (right).

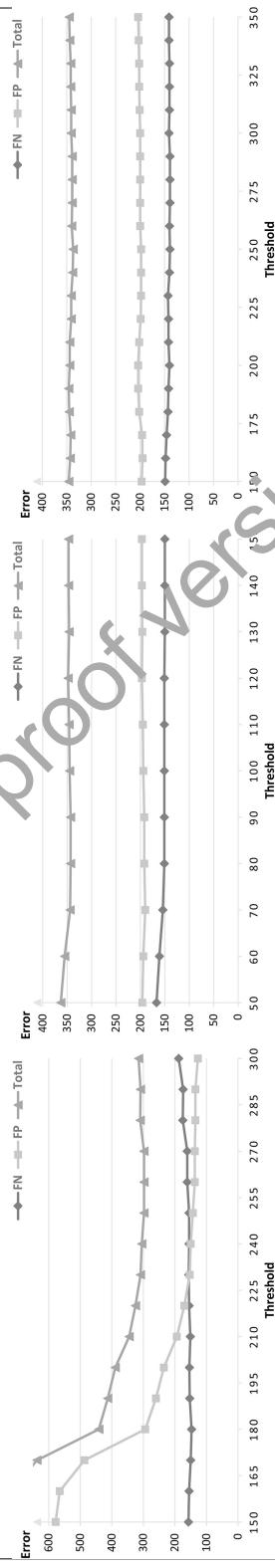


Fig. 13. False detections of optimization approaches A0 (left), A1 (middle), and A2 (right).

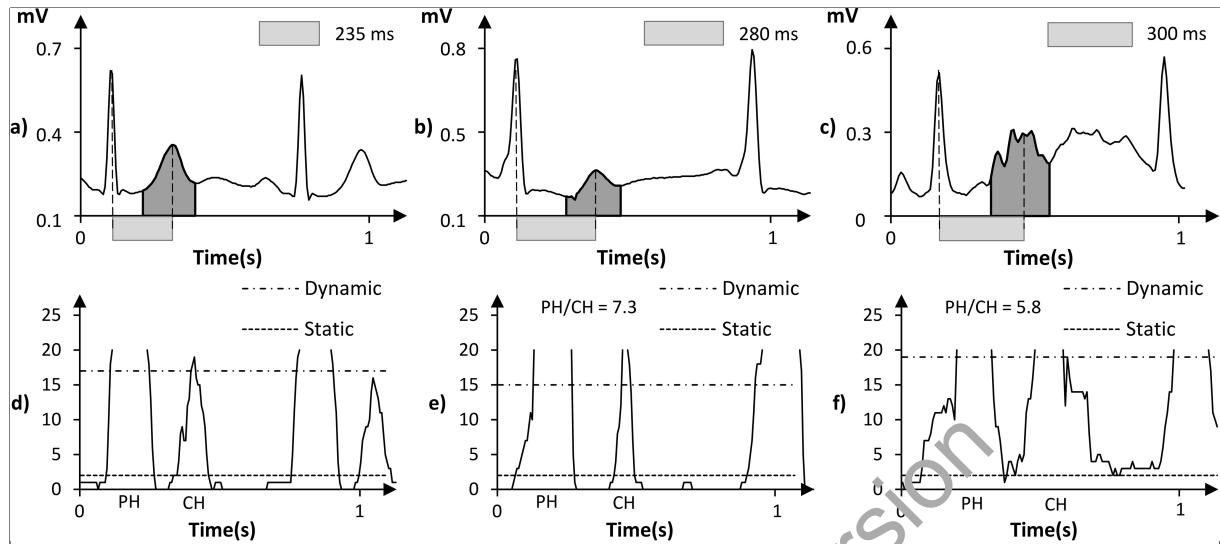


Fig. 14. Signal extracts and outputs after squared average over 80 ms window of executing our algorithm over MITDB: a) signal and d) output for A0 type artifact in record 103 (1304.4 sec); b) signal and e) output for A1 type artifact in record 124 (413.3 sec); c) signal and f) output for A2 type artifact in record 101 (132.2 sec).

322 Examples of different types of detected artifacts are presented in Fig. 14. The identified segments
 323 demonstrate that the detected peaks are artifacts, since their energy is higher than static, and dynamic
 324 thresholds, but satisfies one of the A0, A1, and A2 rules. Otherwise, if none of these rules are satisfied,
 325 then the detected beat C is not an artifact.

326 The results of the experiment using the A0 optimization approach, are presented in the left part of
 327 Fig. 13. A threshold value of $TA0 = MS250 = 250$ ms yields promising results, due to the lowest level
 328 of false detections.

329 The impact of threshold values on the A1 approach is demonstrated in the middle part of Fig. 13. The
 330 x -axis denotes values which are multiplied by 100 and the optimized value 80 for $THRA1$ corresponds
 331 to $80/100 = 0.8$.

332 The right part of Fig. 12 shows how the threshold parameter impacts the performance of the A2
 333 optimization method. The x -axis denotes values that are multiplied by 100, and the optimized value 250
 334 corresponds to $250/100 = 2.5$ for $THRA2$.

335 5.5. Beat rate impact on artifacts

336 Our experiments have shown that the beat rate affects the intervals in the A0-A2 optimization ap-
 337 proaches. Let RR_{avg} be the average value of the last R to R intervals, and f_s the sampling frequency.
 338 Then beat rate BPM is measured by beats per minute, and calculated by Eq. (7).

$$339 BPM = \frac{60}{(RR_{avg}/f_s)} = \frac{60f_s}{RR_{avg}} \quad (7)$$

340 To classify whether a peak is an artifact, we have set three time constants $TA0 = 250$ ms, $TA1 =$
 341 260 ms and $TA2 = 320$ ms. Usually, the peaks closer than 250 ms are considered as peaks (corresponds
 342 to a beat rate higher of 240 BPM). The second and third thresholds correspondingly check if the peak is
 closer than 260 ms (corresponds to 230 BPM) or 320 ms (corresponds to 188 BPM).

Table 2
Comparison of algorithm performance over MIT-BIH Arrhythmia database

Algorithm	MIT-BIH Arrhythmia database						All 48 records			No paced records (44)			
	f_S (Hz)	Bits	Scale (mV)	Total peaks	TP	FP	FN	Tot err	Q_{SE}	Q_{+P}	Tot err	Q_{SE}	Q_{+P}
Our Work	125	10	6	109494	109382	110	112	222	99.90	99.90	194	99.91	99.90
Ghaffari [43]	360	11	10	109428	109327	129	101	230	99.91	99.88	N/A	N/A	N/A
Bahouraa [29]	250	11	10	116137	109625	133	174	307	99.83	99.88	303	99.82	99.88
Elgendi [44]	360	11	10	109985	109775	82	247	329	99.78	99.92	322	99.76	99.92
Martinez [40]	360	11	10	109428	109111	35	317	352	99.71	99.97	N/A	N/A	N/A
Martinez [31]	360	11	5	109428	109208	153	220	373	99.80	99.86	N/A	N/A	N/A
Cvikl [45]	250	N/A	N/A	109494	109294	200	200	400	99.82	99.82	373	99.81	99.82
Chiarugi [46]	360	11	10	109494	109228	210	266	476	99.76	99.81	443	99.75	99.81
Lee [47]	N/A	N/A	N/A	109481	109146	137	335	472	99.69	99.87	459	99.68	99.87
Zidemal [48]	360	N/A	N/A	109494	109101	193	393	586	99.64	99.82	540	99.64	99.83
Hamilton [5]	360	11	10	109267	108927	248	340	588	99.69	99.71	569	99.68	99.76
Choi [49]	360	11	N/A	109494	109118	218	376	594	99.66	99.80	561	99.65	99.79
GQRS [6]	360	11	10	109494	109196	302	298	600	99.73	99.72	562	99.72	99.72
Christov [50]	360	N/A	5	109855	109615	386	288	674	99.74	99.65	670	99.72	99.62
Arzeno [10]	360	11	10	N/A	109099	405	354	759	99.69	99.63	N/A	N/A	N/A
Pan Tompkins [4]	200	11	10	116137	109532	507	277	784	99.75	99.54	771	99.73	99.50
Paoletti [51]	360	11	10	109809	109430	565	379	944	99.65	99.49	924	99.64	99.45
Poli [33]	120	11	10	109963	109522	545	441	985	99.60	99.51	N/A	N/A	N/A
Elgendi [52]	360	11	10	109493	109397	97	1715	1812	98.31	99.92	1798	98.33	99.91
Chouakri [53]	360	11	10	109488	108043	3068	1446	4514	98.68	97.24	4155	98.63	97.29

343 However, our analysis has shown that premature beats might appear closer than these values. This is
 344 why we introduced a scaling factor to the previous improvement, and use the time thresholds calculated
 345 by Eq. (8), which are multiplied by the scaling factor BPM_BASE and heart rate BPM .

$$RR \leq TAx \frac{BPM_BASE}{BPM} x \in \{0, 1, 2\} \quad (8)$$

346 The right part of Fig. 12 shows how the scaling factor BPM_BASE impacts the performance. The x -axis
 347 shows the values of the scaling factor BPM_BASE in a range from 40 to 130, with increments of 2,
 348 and y -axis the number of errors.

349 We observe that a value of $BPM_BASE = 90$ minimizes the errors.

350 6. Evaluation and discussion

351 The following parameter combination achieves the best overall performance: $STHR = 2$, $SearchL =$
 352 152 ms, $SearchR = 56$ ms, $DTHR = 200$, $TA0 = 250$ ms, $TA1 = 260$ ms, $TA2 = 320$ ms, $THRA1 = 0.8$,
 353 $THRA2 = 2.5$, and $BPM_BASE = 90$.

354 Table 2 gives an overview of the obtained results and shows that our algorithm has reached better-
 355 combined values of sensitivity and positive predictive rate.

356 One can face several problems comparing any QRS detection method to other published papers as
 357 summarized below:

- 358 – no source code provided to check other approaches;
- 359 – no info about positive predictive rate; or
- 360 – no info about the number of errors.

When analyzing the performance, only a small number of papers give information on the achieved positive predictive rate and they usually target achieving higher sensitivity values. However, it is very easy to achieve a higher sensitivity value and capture most of the results you would like to include in your algorithm by relaxing the constraints on the optimization parameters, but, at the same time, this will produce many extra generated peaks that do not represent a QRS peak. This is why it is very important to address both the sensitivity and positive predictive to evaluate performance.

This means that there is no direct comparison method. To cope with this problem, we have analyzed the number of errors as a performance measure (as defined by *Errors* in Eq. (2)). Although this performance measure can be achieved by calculation the sum of FP and FN, it can also be calculated through the harmonic mean (*HM*) of the QRS sensitivity and positive predictive rate by Eq. (9).

$$\text{Errors} = \text{TotalQRS} * \left(\frac{1}{Q_{SE}} + \frac{1}{Q_{+P}} - 2 \right) \quad (9)$$

We used Eq. (10) to evaluate this relation. In addition, we assumed that the number of extra detected (false negative) peaks is a lot smaller than the number of correctly detected QRS peaks, i.e. $TP \gg FN$, leading to $\text{TotalQRS} \approx TP$.

$$\begin{aligned} \frac{1}{Q_{SE}} + \frac{1}{Q_{+P}} &= \frac{TP + FN}{TP} + \frac{TP + FP}{TP} \\ &= 2 + \frac{FP + FN}{TP} \end{aligned} \quad (10)$$

A number of discrepancies were found while analyzing related work. Table 1 of [4] shows that the number of errors is 782 although it is 784. The calculation of total beats is the most unambiguous. For example, $TP + FN$ is greater than TB in [52]. Lee publishes two papers [47,54] providing correspondingly 109486 and 109481 total beats. The former one has 6 additional beats, which are from files 118, 201, 205, 220, 221 and 233, whereas the latter one has 1 additional beat at record 114.

We found that various authors have used a different number of detected peaks. They should use the total number of detected beats, since the total number of peaks includes also non-beat annotations, for example, locations where there is a rhythm change. That is why we use the total number of annotated beats in the MIT-BIH Arrhythmia database 109494.

Table 2 also shows that researchers focusing on QRS detection algorithms mostly tend to use the original sampling frequencies, and resolution of reference ECG databases. Running the algorithm on 125 Hz means that nearly 3 times less data is feed to the QRS detector, thus the execution times are deduced accordingly. Moreover, the adjustments that we proposed increased the performance metrics, which in total yields a better QRS detector intended for a small wearable one channel ECG sensor.

7. Conclusion

We have introduced several methods to improve the QRS detection in differentiation-based algorithms. Even though our approach is demonstrated on Hamilton's QRS detection algorithm, it can also be implemented in other algorithms. The results show superior performance over other published results. The improvements were efficiently built in on an industrial QRS detector, for a wearable ECG monitor, where the sampling frequency is 125 Hz, with 10-bit resolution of the AD converter.

Tuning the threshold values might increase the performance, but one needs to develop new relations to generate the thresholds, such as finding

- 396 – how many previous beats might be analyzed to estimate the mean value,
397 – which peaks will be classified as QRS beats since they are very similar in shape, and
398 – the impact of the beat rate on classifying the artifacts.

399 For this purpose, we have introduced several new rules for 1) threshold calculation to better classify
400 QRS peaks; 2) elimination of QRS-like artifacts, and 3) artifact elimination based on beat rate.

401 Due to rescaling, loose ECG contacts, and noise caused by muscles, 62 beats cannot be detected by an-
402 alyzing the first ECG channel, without the analysis of a second channel. This gives a higher performance
403 of the QRS sensitivity to 99.96%, and our algorithm has reached a 99.90% QRS sensitivity, with 99.90%
404 positive predictive rate when all records are analyzed in the MITDB, and 99.91% QRS sensitivity for 44
405 records without paced beats.

406 Generally, the published papers on QRS detection algorithms do not offer their source codes, and
407 only some of them are validated with referenced ECG databases, such as the MIT-BIH Arrhythmia
408 database. Most of the algorithms give a brief description of design issues without implementation details,
409 addressing only theoretical related results. This is why one cannot directly compare results. Different
410 approaches generally do not achieve results as the ones achieved in a real implementation.

411 This study can facilitate possible QRS detection algorithms to consider rescaling, and resampling on
412 reference ECG databases in return for better performances. Our findings show that the adjusted version
413 of Hamilton's QRS detection algorithm yields better results with 125 Hz data on nearly three times
414 shorter execution times.

415 Our future work will be in the direction of classifying QRS detection errors, and eventually eliminating
416 them, as well as for the purpose of modeling the dependence on sampling rate, and bit resolution. Addi-
417 tionally, we are considering building a quality industrial heart beats classifier, with higher performance,
418 on top of the detection algorithm.

419 References

- 420 [1] Böhm A, Brüser C, Leonhardt SL. A novel BCG sensor-array for unobtrusive cardiac monitoring. *Acta Polytechnica*.
421 2013; 53(6).
- 422 [2] Álvarez RA, Penín AJM, Sobrino XAV. A comparison of three QRS detection algorithms over a public database. *Proce-
423 dedia Technology*. 2013; 9: 1159-1165.
- 424 [3] Moody GB, Mark RG. The impact of the MIT-BIH arrhythmia database. *IEEE Engineering in Medicine and Biology
425 Magazine*. 2001; 20(3): 45-50.
- 426 [4] Pan J, Tompkins WJ. A real-time QRS detection algorithm. *IEEE transactions on biomedical engineering*. 1985; (3):
427 230-236.
- 428 [5] Hamilton PS, Tompkins WJ. Quantitative investigation of QRS detection rules using the MIT-BIH arrhythmia database.
429 *IEEE transactions on biomedical engineering*. 1986; (12): 1157-1165.
- 430 [6] Goldberger AL, Amaral LA, Glass L, Hausdorff JM, Ivanov PC, Mark RG, et al. Physiobank, physiotoolkit, and physi-
431 onet. *Circulation*. 2000; 101(23): e215-e220.
- 432 [7] Pahlm O, Sörnmo L. Software QRS detection in ambulatory monitoring: A review. *Medical and Biological Engineering
433 and Computing*. 1984; 22(4): 289-297.
- 434 [8] Friesen GM, Jannett TC, Jadallah MA, Yates SL, Quint SR, Nagle HT. A comparison of the noise sensitivity of nine
435 QRS detection algorithms. *IEEE Transactions on biomedical engineering*. 1990; 37(1): 85-98.
- 436 [9] Kohler BU, Hennig C, Orglmeister R. The principles of software QRS detection. *IEEE Engineering in Medicine and
437 Biology Magazine*. 2002; 21(1): 42-57.
- 438 [10] Arzeno NM, Deng ZD, Poon CS. Analysis of first-derivative based QRS detection algorithms. *IEEE Transactions on
439 Biomedical Engineering*. 2008; 55(2): 478-484.
- 440 [11] Ahlstrom ML, Tompkins WJ. Automated high-speed analysis of Holter tapes with microcomputers. *IEEE Transactions
441 on Biomedical Engineering*. 1983; (10): 651-657.
- 442 [12] Fraden J, Neuman M. QRS wave detection. *Medical and Biological Engineering and computing*. 1980; 18(2): 125-132.

- [13] Morizet-Mahoudeaux P, Moreau C, Moreau D, Quarante J. Simple microprocessor-based system for on-line ECG arrhythmia analysis. *Medical and Biological Engineering and Computing*. 1981; 19(4): 497-500.
- [14] Afonso VX, Tompkins WJ, Nguyen TQ, Luo S. ECG beat detection using filter banks. *IEEE transactions on biomedical engineering*. 1999; 46(2): 192-202.
- [15] Borjesson PO, Pahlm O, Sörnmo L, Nygård ME. Adaptive QRS detection based on maximum a posteriori estimation. *IEEE Transactions on Biomedical Engineering*. 1982; (5): 341-351.
- [16] Engelse W, Zeelenberg C. A single scan algorithm for QRS-detection and feature extraction. *Computers in Cardiology*. 1979; 6(1979): 37-42.
- [17] Fancott T, Wong DH. A minicomputer system for direct high speed analysis of cardiac arrhythmia in 24 h ambulatory ECG tape recordings. *IEEE Transactions on Biomedical Engineering*. 1980; (12): 685-693.
- [18] Okada M. A digital filter for the QRS complex detection. *IEEE Transactions on Biomedical Engineering*. 1979; (12): 700-703.
- [19] Gusev M, Ristovski A, Guseva A. Pattern recognition of a digital ECG. In: *International Conference on ICT Innovations*. Springer; 2016; p. 93-102.
- [20] Ciaccio E, Dunn S, Akay M. Biosignal pattern recognition and interpretation systems. *IEEE Engineering in Medicine and Biology Magazine*. 1993; 12(3): 89-95.
- [21] Maglaveras N, Stamkopoulos T, Diamantaras K, Pappas C, Strintzis M. ECG pattern recognition and classification using non-linear transformations and neural networks: a review. *International journal of medical informatics*. 1998; 52(1-3): 191-208.
- [22] Steinberg C, Abraham S, Caceres C. Pattern recognition in the clinical electrocardiogram. *IRE Transactions on Bio-Medical Electronics*. 1962; 9(1): 23-30.
- [23] Trahanias P, Skordalakis E. Syntactic pattern recognition of the ECG. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1990; 12(7): 648-657.
- [24] Xue Q, Hu YH, Tompkins WJ. Neural-network-based adaptive matched filtering for QRS detection. *IEEE Transactions on Biomedical Engineering*. 1992; 39(4): 317-329.
- [25] Barro S, Fernandez-Delgado M, Vila-Sobrino J, Regueiro C, Sanchez E. Classifying multichannel ECG patterns with an adaptive neural network. *IEEE Engineering in Medicine and Biology Magazine*. 1998; 17(1): 45-55.
- [26] Dokur Z, Ölmez T, Yazgan E, Ersoy OK. Detection of ECG waveforms by neural networks. *Medical Engineering and physics*. 1997; 19(8): 738-741.
- [27] Hu YH, Tompkins WJ, Urrusti JL, Afonso VX. Applications of artificial neural networks for ECG signal detection and classification. *Journal of electrocardiology*. 1993; 26: 66-73.
- [28] Li C, Zheng C, Tai C. Detection of ECG characteristic points using wavelet transforms. *IEEE Transactions on biomedical Engineering*. 1995; 42(1): 21-28.
- [29] Bahoura M, Hassani M, Hubin M. DSP implementation of wavelet transform for real time ECG wave forms detection and heart rate analysis. *Computer methods and programs in biomedicine*. 1997; 52(1): 35-44.
- [30] Shambu J, Tandon S, Bhatt R. Using wavelet transforms for ECG characterization. *IEEE Engineering in Medicine and Biology*. 1997; p. 77-83.
- [31] Martínez JP, Almeida R, Olmedo S, Rocha AP, Laguna P. A wavelet-based ECG delineator: evaluation on standard databases. *IEEE transactions on biomedical engineering*. 2004; 51(4): 570-581.
- [32] Milchevski A, Gusev M. Improved pipelined wavelet implementation for filtering ECG signals. *Pattern Recognition Letters*. 2017; 95: 85-90.
- [33] Poli R, Cagnoni S, Vardi G. Genetic design of optimum linear and nonlinear QRS detectors. *IEEE Transactions on Biomedical Engineering*. 1995; 42(11): 1137-1141.
- [34] Coast DA, Stern RM, Cano GG, Briller SA. An approach to cardiac arrhythmia analysis using hidden Markov models. *IEEE Transactions on biomedical Engineering*. 1990; 37(9): 826-836.
- [35] Andreão RV, Dorozzi B, Boudy J. ECG signal analysis through hidden Markov models. *IEEE Transactions on Biomedical engineering*. 2006; 53(8): 1541-1549.
- [36] Cost A, Cano GG. QRS detection based on hidden Markov modeling. In: *Engineering in Medicine and Biology Society, 1989. Images of the Twenty-First Century, Proceedings of the Annual International Conference of the IEEE Engineering in IEEE*; 1989. p. 34-35.
- [37] Song-Kai Z, Jian-Tao W, Jun-Rong X. The real-time detection of QRS-complex using the envelope of ECG. In: *Engineering in Medicine and Biology Society, 1988. Proceedings of the Annual International Conference of the IEEE*. IEEE; 1988. p. 38s.
- [38] Nygård ME, Sörnmo L. Delineation of the QRS complex using the envelope of the ECG. *Medical and Biological Engineering and Computing*. 1983; 21(5): 538-547.
- [39] Benitez D, Gaydecki P, Zaidi A, Fitzpatrick A. The use of the Hilbert transform in ECG signal analysis. *Computers in biology and medicine*. 2001; 31(5): 399-406.

- 500 [40] Martínez A, Alcaraz R, Rieta JJ. Application of the phasor transform for automatic delineation of single-lead ECG
501 fiducial points. *Physiological measurement*. 2010; 31(11): 1467.
- 502 [41] <http://www.heart.org>. American Health Association, AHA database.
- 503 [42] Hamilton P. Open Source ECG Analysis Software Documentation; 2002.
- 504 [43] Ghaffari A, Homaeinezhad M, Akraminia M, Atarod M, Daevaeiha M. A robust wavelet-based multi-lead electrocardio-
505 gram delineation algorithm. *Medical Engineering and Physics*. 2009; 31(10): 1219-1227.
- 506 [44] Elgendi M, Mohamed A, Ward R. Efficient ECG Compression and QRS Detection for E-Health Applications. *Scientific
507 Reports*. 2017; 7(1): 459.
- 508 [45] Cvikel M, Jager F, Zemva A. Hardware implementation of a modified delay-coordinate mapping-based QRS complex
509 detection algorithm. *EURASIP Journal on Applied Signal Processing*. 2007; 2007(1): 104-104.
- 510 [46] Chiarugi F, Sakkalis V, Emmanouilidou D, Krontiris T, Varanini M, Tollis I. Adaptive threshold QRS detector with best
511 channel selection based on a noise rating system. In: *Computers in Cardiology, IEEE*; 2007, p. 157-160.
- 512 [47] Lee J, Jeong K, Yoon J, Lee M. A simple real-time QRS detection algorithm. In: *Engineering in Medicine and Biology
513 Society, 1996. Bridging Disciplines for Biomedicine. Proceedings of the 18th Annual International Conference of the
514 IEEE*. vol. 4. IEEE; 1996. p. 1396-1398.
- 515 [48] Zidelmal Z, Amirou A, Adnane M, Belouchrani A. QRS detection based on wavelet coefficients. *Computer Methods and
516 Programs in Biomedicine*. 2012; 107(3): 490-496.
- 517 [49] Choi S, Adnane M, Lee GJ, Jang H, Jiang Z, Park HK. Development of ECG beat segmentation method by combining
518 lowpass filter and irregular R-R interval checkup strategy. *Expert Systems with Applications*. 2010; 37(7): 5208-5218.
- 519 [50] Christov II. Real time electrocardiogram QRS detection using combined adaptive threshold. *Biomedical engineering
520 online*. 2004; 3(1): 28.
- 521 [51] Paoletti M, Marchesi C. Discovering dangerous patterns in long-term ambulatory ECG recordings using a fast QRS
522 detection algorithm and explorative data analysis. *Computer Methods and programs in biomedicine*. 2006; 82(1): 20-30.
- 523 [52] Elgendi M, Jonkman M, DeBoer F. Frequency bands effects on QRS detection. *Pan*. 2010; 5: 15 Hz.
- 524 [53] Chouakri S, Berekci-Reguig F, Taleb-Ahmed A. QRS complex detection based on multi wavelet packet decomposition.
525 *Applied Mathematics and Computation*. 2011; 217(23): 9508-9525.
- 526 [54] Lee JW, Kim KS, Lee B, Lee B, Lee MH. A real time QRS detection using delay-coordinate mapping for the microcon-
527 troller implementation. *Annals of biomedical Engineering*. 2002; 30(9): 1140-1151.