



Simple real-time QRS detector with the MaMeMi filter[☆]



David Castells-Rufas^{*}, Jordi Carrabina

Department of Microelectronics and Electronics Systems, Engineering School, Universitat Autònoma de Barcelona, Bellaterra 08193, Spain

ARTICLE INFO

Article history:

Received 11 December 2014
Received in revised form 29 May 2015
Accepted 1 June 2015
Available online 26 June 2015

Keywords:

Nonlinear filters
Biomedical signal processing
QRS detection
Real-time systems
Embedded software

ABSTRACT

Detection of QRS complexes in ECG signals is required to determine heart rate, and it is an important step in the study of cardiac disorders. ECG signals are usually affected by noise of low and high frequency. To improve the accuracy of QRS detectors several methods have been proposed to filter out the noise and detect the characteristic pattern of QRS complex. Most of the existing methods are at a disadvantage from relatively high computational complexity or high resource needs making them less optimized for its implementation on portable embedded systems, wearable devices or ultra-low power chips. We present a new method to detect the QRS signal in a simple way with minimal computational cost and resource needs using a novel non-linear filter.

© 2015 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

The function of the heart can be detected by sensing the voltage variations that occur on certain parts of the human body surface. An electrocardiograph (ECG) records those voltage changes over time for later analysis. ECG is a fundamental tool used by cardiologist to diagnose cardiac diseases. Moreover, heart rate monitoring has become an extremely popular method to control the evolution of physical training in sports.

The normal sinus rhythm of the heart is characterized by a sequence of some known phases such as P, Q, R, S, T as depicted in Fig. 1. Heart rate is inferred by detecting the QRS complexes and obtaining the period between consecutive R peaks. However, QRS detection is not as simple as it could seem after seeing this ideal situation. ECG signals are usually affected by several noise sources, like muscular contraction and respiration. Moreover, as a consequence of a disease or a temporal alteration, heart beats can have very different characteristic patterns [1]. Real ECG recording databases, like the MIT/BIH ECG database [2], include episodes with all these different phenomena. Such databases allow us to study and design QRS detectors that take them into account. On top of that, by providing simple metrics to compare the performance of different

algorithms, they have fostered the research for better and better detection methods.

There are large number of QRS detection proposals [3], and it could seem that the QRS detection problem is already saturated. Nevertheless, with the availability of portable embedded systems, wearable devices, and body area networks there is a renewed interest on analyzing energy efficient algorithms that allow QRS detection with minimal energy consumption so that they can extend battery life of portable equipment [4,6,7] or sports equipment.

In this paper we present a novel algorithm to detect QRS complexes in ECG signals with very low computational complexity. In Section 2 we review some of the methods described in the literature that should be avoided in order to reduce the computational complexity and energy consumption of a QRS detector. Then, Section 3 describes the various processing steps we apply to the signal to detect the heart beats, including the novel MaMeMi filter used to remove the low frequency noise caused by the baseline wander, and the following steps to finally detect the QRS peaks.

Once the method is presented, Section 4 describes the performance obtained by the algorithm when tested against the MIT/BIH ECG database. Section 5 gives more details about the low resource cost of the algorithm implementation. Finally Section 6 presents the conclusions.

2. Computationally costly methods

In this paper we analyze detectors that process digitized ECG data like those present in MIT/BIH. On such data detection is usually a two step process, first a filtering step to remove noise and enhance

[☆] This work is partially supported by the EU FP7 DocuMeet Project <http://www.documeet.eu/the-project>.

^{*} Corresponding author. Tel.: +34 935813563.

E-mail addresses: david.castells@uab.cat (D. Castells-Rufas), jordi.carrabina@uab.cat (J. Carrabina).

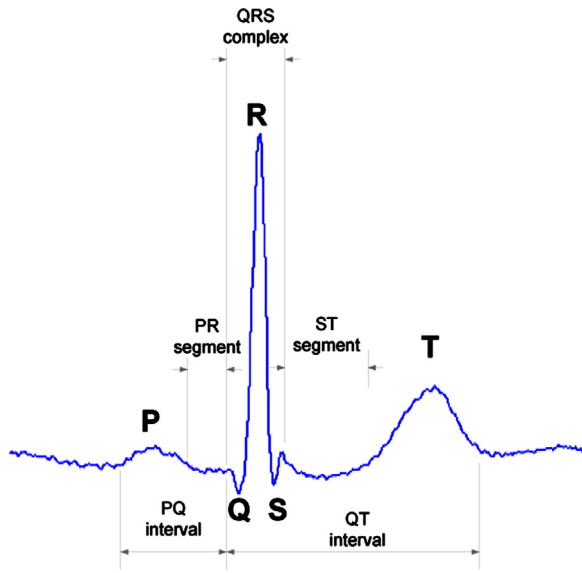


Fig. 1. Example of the parts of ECG (actual fragment of tape 103 from MIT/BIH database).

signal so that QRS complex become clean pulses, and then a second step that analyze the pulses to determine if they are QRS complexes.

The frequency components of the QRS complex range from 10 Hz to 25 Hz. So, most algorithms filter out high and low frequency noise.

Low frequency noise can be removed by different high pass filtering techniques [8–10]. High pass FIR filters for low cutoff frequencies require a large number (let be n) of taps. This kind of filters require n fixed point multipliers and adders that must compute for each sample. This means that at least $2 \cdot n$ operations are needed for each sample. On the other hand, IIR filters usually need floating point coefficients, hence require floating point multipliers. The downfall of this method is that it takes too much resources in a hardware design or too many clock cycles of a integer microcontroller. Another method to remove the baseline is the subtraction of the median over a window from the signal ([11]). This is an ineffective strategy since the sorting of a signal window must be done for every sample, so for a window of n samples n^2 operations are needed.

For high frequency removal similar problems can arise, although the number of taps is usually lower. A possible technique to remove high frequency noise is the usage of morphological operators as described in [15]. Their principle is simple, but they need to access several values of the input signal for each sample. Common designs combine up to 8 concurrent filters of this kind. So the number of needed operations per sample is not very low.

On the enhancement phase some techniques (used in [12]) include the squaring of the signal. The use of multipliers should be avoided if possible to minimize the resource and energy costs. Other methods (also used in [12]) use the integration of the signal over a window of n samples, which is again requiring n operations for each sample of the signal.

Wavelet transform based methods [13] are also computationally expensive. Hilbert transformation [14], that shows very good results, is also prohibitive since it requires computing FFT over a window for every sample.

3. Proposed design

We propose a novel algorithm to be applied in real-time over ECG signals. Like few similar proposals, it consists of a filtering procedure coupled with enhancement phase, followed by a

detection phase resulted after several applied criteria. A diagram of the process is shown in Fig. 2.

The differentiation factor from known methods in proposed one is the use of a new filter to reduce low frequency noise. The detection phase is similar to many other works [3]. The computational cost of most of the algorithms is devoted to the first filtering and enhancement phase. In our case the computational cost is highly reduced, making it very attractive for its application in battery operated devices.

3.1. Low frequency noise suppression

To remove the base wander, we have designed a high pass non-linear filter. We called it MaMeMi, from maximum mean minimum. The principle is very similar to [11], i.e. subtract a low-pass filtered version of the signal from the original signal to take the higher frequency components. As stated in Section 2, [11] proposes a median filter to get the signal to subtract, which is a computationally costly operation of the order n^2 operations per sample, being n the width of the window used to compute the median. Other works proposed a similar approach using a standard FIR or IIR filter to get the signal subtract. Again this is a computationally costly operation of the order $2 \cdot n$ per sample, being n the number of taps of the filter. In our case we propose to subtract the mean between the moving maximum and minimum values of the signal. Let $x(t)$ be the discrete time function of the digitized ECG input signal, the filter output $h(t)$ is defined by Eq. (1).

$$h(t) = x(t) - \frac{\max^*(t) + \min^*(t)}{2} \quad (1)$$

$$\max^*(t) = \begin{cases} x(t) & \text{if } t = 0 \\ \max^*(t-1) + \sigma \cdot \Delta & \text{if } x(t) > \max^*(t-1) \\ \max^*(t-1) - \Delta & \text{if } x(t) \leq \max^*(t-1) \end{cases} \quad (2)$$

$$\min^*(t) = \begin{cases} x(t) & \text{if } t = 0 \\ \min^*(t-1) - \sigma \cdot \Delta & \text{if } x(t) < \min^*(t-1) \\ \min^*(t-1) + \Delta & \text{if } x(t) \geq \min^*(t-1) \end{cases} \quad (3)$$

In order to obtain the real extrema of $x(t)$ over a window of n samples we would need n comparisons for each sample, and this is precisely the amount of computation that we wanted to avoid. We used pseudo-extrema functions instead that need just one comparison and one operation per sample. We called them \max^* and \min^* . They are defined in Eqs. (2) and (3) respectively. The principle of operation of the $\max^*(t)$ is as follows: we kept a variable with the current maximum, if the value of the input function $x(t)$ is equal or lower than the current maximum, we decreased the current maximum by a factor Δ . If the value of the input function $x(t)$ is higher we then increase the current maximum by a factor $\sigma \cdot \Delta$. We interpreted that $\max^*(t)$ tries to follow the maximum value of $x(t)$ with some decay over time. The higher the factor σ is, the faster $\max^*(t)$ will follow the maximum of $x(t)$. The higher the factor Δ is, the faster the decay will occur. Function $\min^*(t)$ is defined similarly. The selection of the best values for σ and Δ is not straightforward since the filter response depends, besides former parameters, on the amplitude of the signals being filtered. The MIT-BIH database was sampled with 11 bit resolution over a range of 10 mV. Heart beats appear as pulses of approximately 1 mV, which become pulses of around 200 units in the digitized signal.

The filter response for sinusoidal signals ranging from 0 to 50 Hz is shown in Fig. 3. To illustrate the different response to different signal amplitudes we show some discrete amplitude values: 10, 19, 55, and 208. Extrema of low amplitude signals are more easily followed and the pseudo-mean is closer to the input signal, so when removing it from the input signal the signal is highly

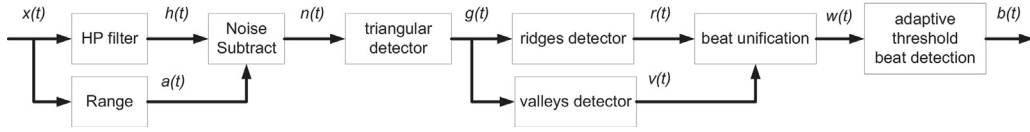


Fig. 2. Proposed algorithm. $x(t)$ is the digitized ECG signal. $h(t)$ is the output of the MaMeMi filter acting as a high pass filter to remove baseline wander. $a(t)$ is the signal range detected by the MaMeMi filter. $n(t)$ the signal after range is subtracted. $g(t)$ is the enhanced signal after triangular detector. $r(t)$ is the vector of peaks. $v(t)$ is the vector of valleys. $w(t)$ is the vector of potential beats after unification. $b(t)$ are the detected beats.

attenuated for a wide range of frequencies. Extrema of high amplitude signals is more difficult to follow. In this case the pseudo-mean changes more slowly with the input signal and by a smaller magnitude. For higher frequencies the change is small than the input signal is hardly attenuated. On the contrary for lower frequencies the pseudo-mean has enough time to follow the input signal, and the filter perfectly attenuates it. If we focus on the amplitude of the input signal, the higher the amplitude, the sharper the high-pass filter becomes, and the lower its cut-off frequency. If we increase the value of σ the cut-off frequency increases and the filter becomes smoother. Increasing the value of Δ , the filter becomes smoother without affecting much the cut-off frequency.

However, the analysis of the filter response to constant sinusoidal signals is not directly applicable in this case since ECG signals are characterized by pulsed activity. Moreover, non-linear filters have transient dynamics not captured by frequency response analysis. To find a good filter parameters we have empirically tested a set of combinations for σ and Δ and measured the performance of the whole QRS detector. The parameters that have shown best results are $\sigma = 2$ and $\Delta = 2$.

The MaMeMi filter can effectively remove the baseline wander as shown in Fig. 4. It also reduces the T wave, which is actually interesting to minimize the efforts to discriminate between R peaks and T peaks.

3.2. High frequency noise suppression

Some additional interesting properties of the max^* and min^* functions are that they do not react quickly to follow the maximum when a sharp pulse occurs, but they can be used to sense the amplitude of a high frequency signal. We can use the $a(t)$ range function as defined in Eq. (4) to sense the amplitude of such signals.

$$a(t) = max^*(t) - min^*(t) \tag{4}$$

Although this is not totally correct, we can interpret this signal as being the envelope of high frequency noise. Then, we can subtract this pseudo-noise amplitude from signal to try to reduce the effect of high frequency noise. The filtered signal $n(t)$ after noise reduction is defined by Eq. (5).

$$n(t) = \begin{cases} sign(h(t)) \cdot (|h(t)| - a(t)) & \text{if } a(t) \leq |h(t)| \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Since max^* and min^* do not react fast to sharp pulses like the R peak, this operation does not suppress the R peaks (Fig. 5).

3.3. Triangular detector

The QRS complex is usually described by a step positive pulse, i.e. a high peak in a neighborhood of lower values. The duration of

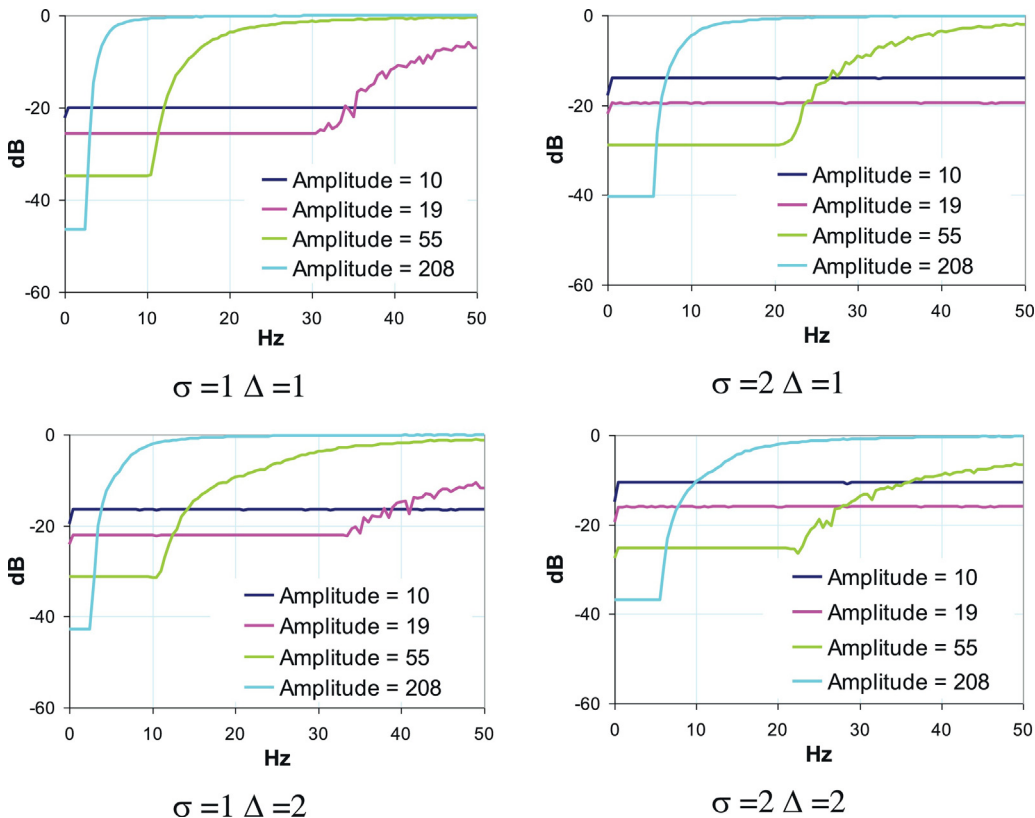


Fig. 3. Filter response to sinusoidal signals of various frequencies and amplitudes under various values of σ and Δ . Note that frequency response varies for different amplitudes.

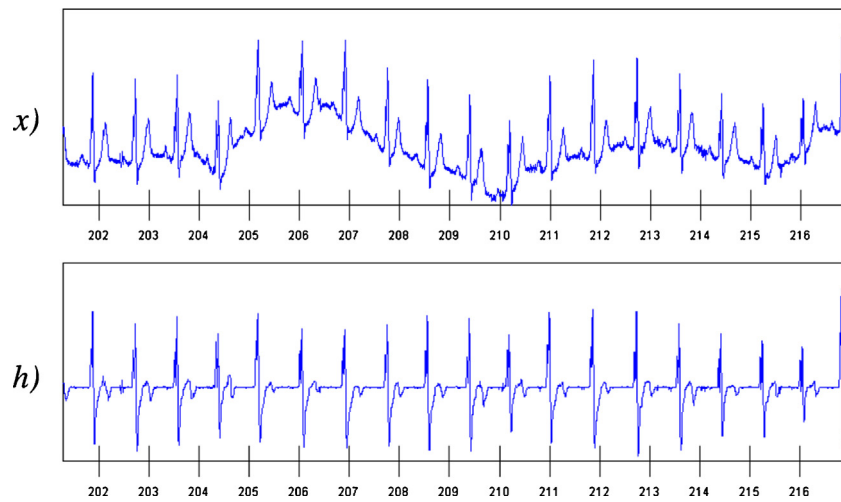


Fig. 4. (x) Fragment of tape 111 from the MIT/BIH ECG database. (h) The output of the MaMeMi filter, with $\sigma=2$ and $\Delta=2$.

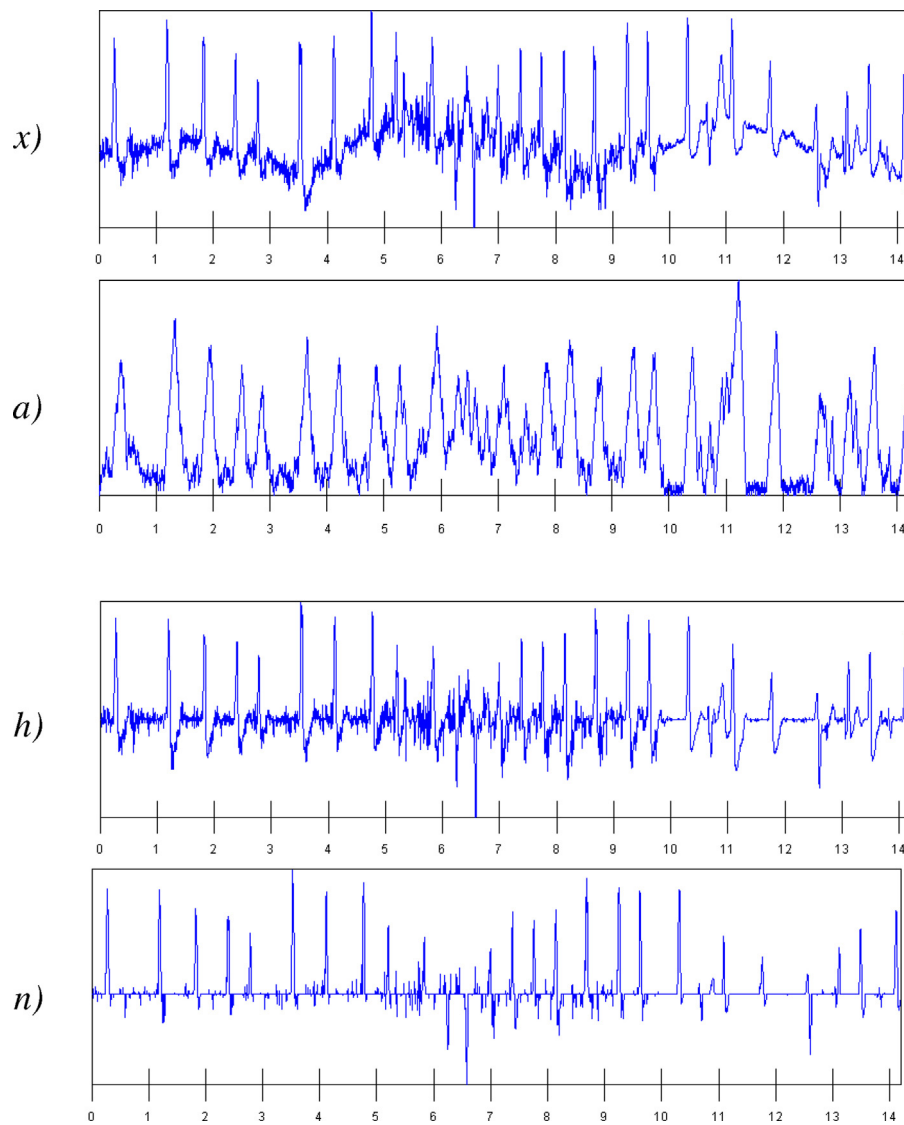


Fig. 5. (x) Fragment of tape 203 from the MIT/BIH ECG database. (a) Range from the MaMeMi extrema. (h) Output of the MaMeMi filter, with $\sigma=2$ and $\Delta=2$. (n) Signal after noise suppression.

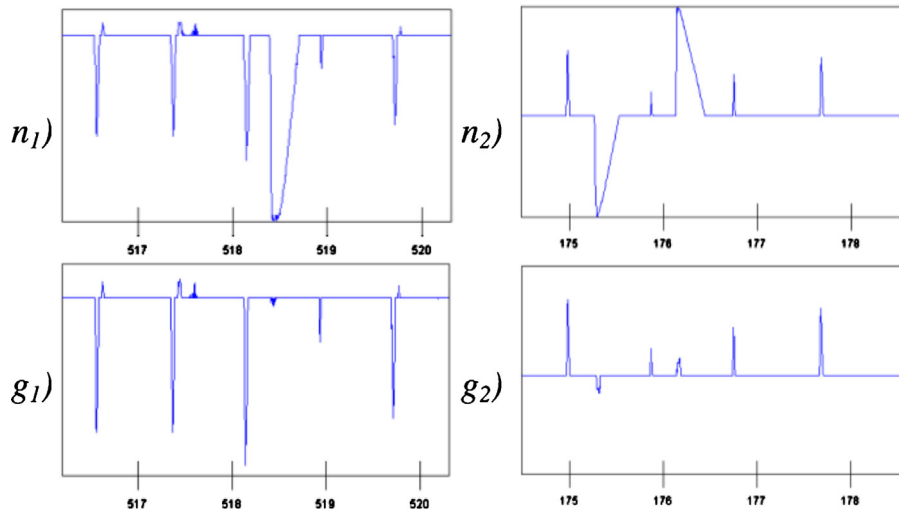


Fig. 6. (n_1) Fragment of tape 207 from the MIT/BIH ECG database after noise suppression. (g_1) Signal after triangular filter with $\beta = 15$. (n_2) Fragment of tape 101 for the MIT/BIH ECG database after noise suppression. (g_2) Signal after triangular filter with $\beta = 15$.

the pulse is determined by physical constraints, and it is commonly in the range of 0.06–0.12 s. To improve the detection of the pulse we propose a method that, for each sample, compares two samples situated β samples before and β samples after the current sample. If the three compared samples form a triangular peak on the positive range or a triangular valley in the negative one, we will measure its height. This logic is expressed in Eq. (6).

$$g(t) = \begin{cases} n(t) - \max(n(t - \beta), n(t + \beta)) & \text{if } (n(t) > 0) \wedge (n(t - \beta) < n(t) > n(t + \beta)) \\ n(t) + \min(n(t - \beta), n(t + \beta)) & \text{if } (n(t) < 0) \wedge (n(t - \beta) > n(t) < n(t + \beta)) \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

The formerly mentioned typical duration of heart beat pulses is equivalent to 21–42 samples, when sampling at 360 Hz. Since β should be close to the half of the loop, we measure empirically

which value of β between 10 and 21 minimizes the detection error rate of the whole QRS detector. The obtained value is $\beta = 15$.

As shown in Fig. 6, the triangular detector filter effectively reduces wide pulses, which are often caused by baseline drifts or big T waves.

3.4. Peak and valley detection

Not all the samples in the $g(t)$ signal are significant to detect the QRS complexes. In fact, due to the various morphologies of QRS

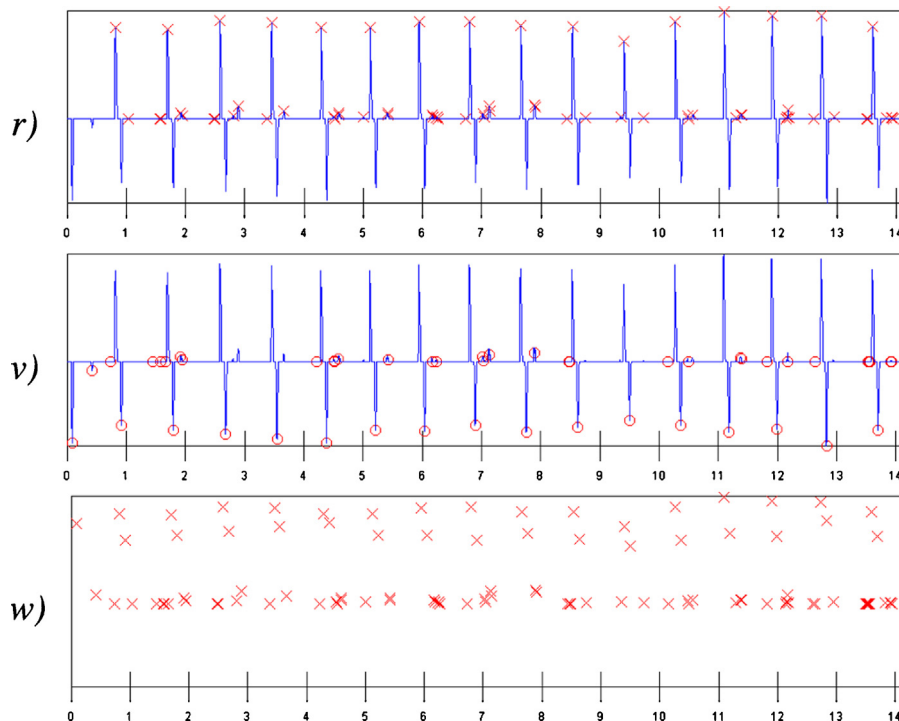


Fig. 7. (r) Detected peaks (crosses) on fragment of tape 107 from the MIT/BIH ECG database after triangular filter $g(t)$. (v) Detected valleys (circles) on the same fragment of $g(t)$. (w) Unification of peaks and valleys.

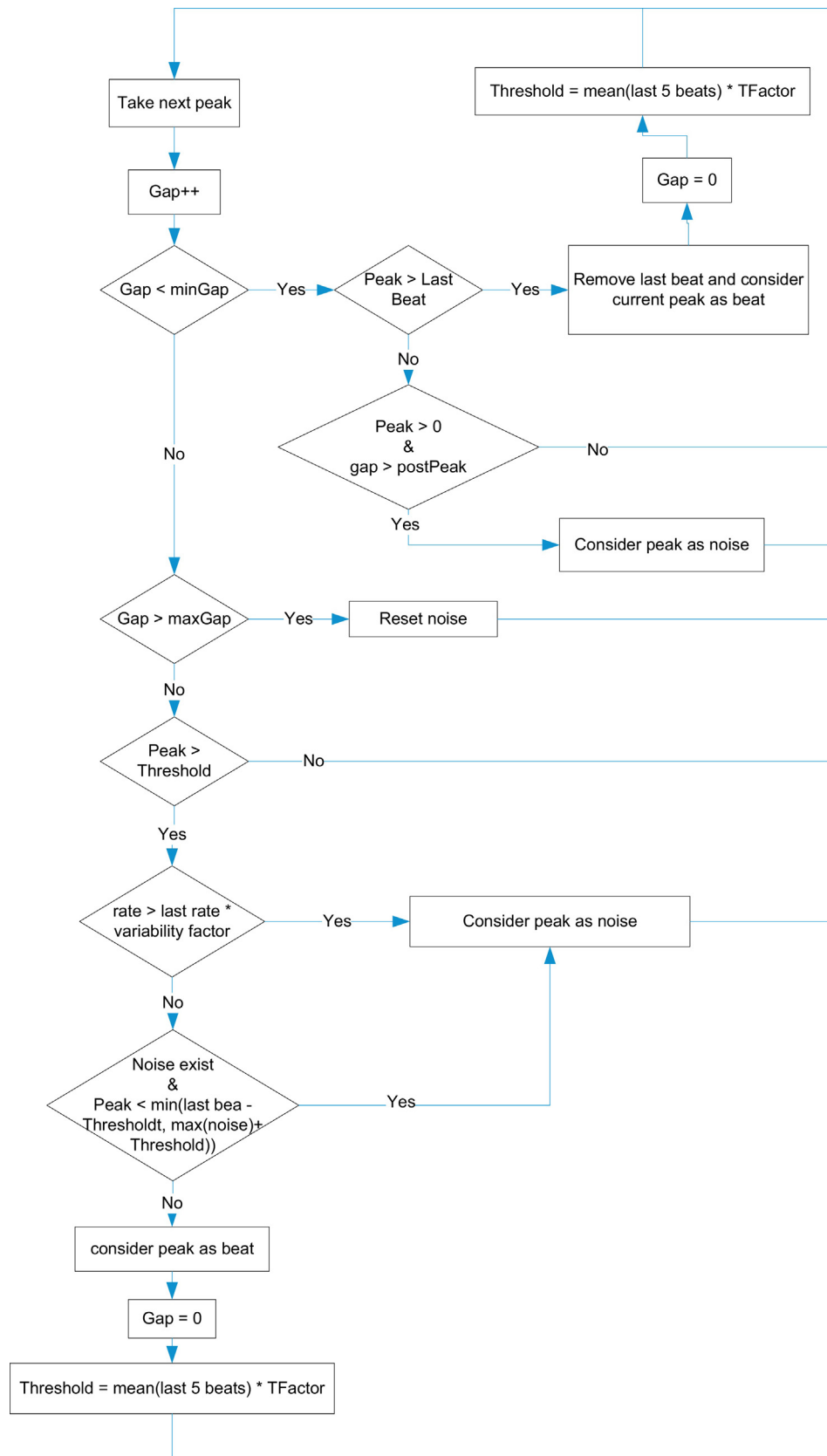


Fig. 8. Flow-chart of the heart beat detection algorithm.

complexes the R peaks can come as positive pulses or negative ones. We can filter out the values that are not interesting, and focus only on the potential R peaks. Potential R peaks are defined by being either local maxima (peaks) in the positive range, or local minima (valleys) in the negative range. Since the definition of local extreme for discrete functions can be ambiguous we defined exactly what points will be considered maxima and minima for the $g(t)$ time function.

We define a new operator $\hat{<}$ in Eq. (7) to determine that $f(i) < f(j)$ and that between those two points in time there is no higher value of function f than $f(j)$. Similarly we have defined the $\hat{>}$ operator in Eq. (8) to determine that $f(i) > f(j)$ and that between those two points in time there is no higher value of function f than $f(i)$.

$$f(i)\hat{<}f(j) = \begin{cases} \text{true} & \text{if } i < j \wedge f(i) < f(j) \wedge f(k) \leq f(j) \quad \forall k \in (i, j) \\ \text{false} & \text{otherwise} \end{cases} \quad (7)$$

$$f(i)\hat{>}f(j) = \begin{cases} \text{true} & \text{if } i < j \wedge f(i) > f(j) \wedge f(k) \geq f(j) \quad \forall k \in (i, j) \\ \text{false} & \text{otherwise} \end{cases} \quad (8)$$

Given previous definitions local maxima and local minima are defined by Eqs. (9) and (10) respectively. In our definition, a local maximum of a discrete time function is a sample such that the next sample is lower, and that there exists a lower previous sample with no higher samples in between. The benefit from this definition is that it supports situations where the maximum value remains constant for several samples, which is a situation that can easily happen in sampled signals.

$$r(t) = \begin{cases} g(t) & \text{if } \exists i > 0 | g(t-i)\hat{>}g(t) > g(t+1) \\ \emptyset & \text{otherwise} \end{cases} \quad (9)$$

$$v(t) = \begin{cases} g(t) & \text{if } \exists i > 0 | g(t-i)\hat{<}g(t) < g(t+1) \\ \emptyset & \text{otherwise} \end{cases} \quad (10)$$

Fig. 7 shows how detected peaks (crosses) and valleys (circles) are detected from the input signal $g(t)$.

3.5. Beat unification

As we recently stated in previous subsection QRS complexes can be positive peaks or negative valleys. So we can discard peaks in the negative range and valleys from the positive range. In addition, we can unify them by changing the sign of valleys in the negative range to become positive peaks. This simplifies the following detection step as it has considered a single threshold in the positive range. In some proposed algorithms (like [12]) this is done by squaring the signal. In our prototype we want to avoid the multiplication operation as it has a higher cost than adding or subtracting. The simple unification time function is described by Eq. (11).

$$w(t) = \begin{cases} r(t) & \text{if } r(t) > 0 \\ -v(t) & \text{if } v(t) < 0 \end{cases} \quad (11)$$

3.6. Beat detection

Heart beats can be identified by peaks of $w(t)$ that are higher than a certain threshold. The amplitude of the beat pulses is normally constant in the range of some mV. But, this amplitude can be altered because of physiological causes or even because of the processes that have been applied and that were originally intended to enhance the input signal. Most QRS detection algorithms use an adaptive threshold that is able to detect the range of the pulses

Table 1
Detector results for the MIT-BIH database.

Tape	Beats	Errors	FP	FN	DER%
100	2273	1	0	1	0.04
101	1865	0	0	0	0.00
102	2187	1	0	1	0.05
103	2084	0	0	0	0.00
104	2229	12	6	6	0.54
105	2572	74	40	34	2.88
106	2027	16	1	15	0.79
107	2137	37	0	37	1.73
108	1763	86	34	52	4.88
109	2532	0	0	0	0.00
111	2124	37	7	30	1.74
112	2539	0	0	0	0.00
113	1795	1	0	1	0.06
114	1879	19	5	14	1.01
115	1953	0	0	0	0.00
116	2412	31	1	30	1.29
117	1535	3	3	0	0.20
118	2278	4	3	1	0.18
119	1987	1	1	0	0.05
121	1863	2	0	2	0.11
122	2476	0	0	0	0.00
123	1518	0	0	0	0.00
124	1619	10	0	10	0.62
200	2601	24	12	12	0.92
201	1963	40	0	40	2.04
202	2136	11	0	11	0.51
203	2980	201	19	182	6.74
205	2656	5	0	5	0.19
207	1860	201	188	13	10.81
208	2955	30	4	26	1.02
209	3005	3	2	1	0.10
210	2650	32	3	29	1.21
212	2748	2	2	0	0.07
213	3251	4	0	4	0.12
214	2262	10	0	10	0.44
215	3363	3	0	3	0.09
217	2208	5	1	4	0.23
219	2154	4	0	4	0.19
220	2048	0	0	0	0.00
221	2427	5	0	5	0.21
222	2483	1	0	1	0.04
223	2605	11	0	11	0.42
228	2053	19	7	12	0.93
230	2256	3	3	0	0.13
231	1571	0	0	0	0.00
232	1780	12	11	1	0.67
233	3079	4	0	4	0.13
234	2753	2	0	2	0.07
Total	109,494	967	353	614	0.88

and modify the threshold accordingly. In our algorithm we use an adaptive threshold. Peaks below that threshold are considered as “noise”.

The width of the QRS complex is usually considered constant, and there is some probability factor that the heart rate does not vary very significantly from beat to beat. We have information on the maximum and minimum possible rates of the human heart. With all this information we define the following criteria:

(1) Criterion 1

A peak higher than the threshold is considered as potential detectable heart beat. Lower peaks are considered noise. Threshold is adaptively updated as the mean of the last 5 detected beats.

(2) Criterion 2

QRS pulses are usually from 0.06 to 0.12 s wide. After a peak is detected if another higher peak is detected inside this time window, the higher one is considered as a valid heart beat, discarding the previous one.

Table 2
Comparison with high quality detectors.

Algorithm	Year	Se (%)	+P (%)	DER%
Geometrical matching [20]	2007	97.94	99.13	2.92
Zero crossing [7]	2014	98.89	99.4	1.71
Moving avg., wavelet denoising [21]	2006	99.55	99.49	0.96
Proposed algorithm without triangle		99.41	99.63	0.95
Full proposed algorithm		99.43	99.67	0.88
Pan & Tompkins [12]	1985	99.76	99.56	0.68
New Dual slope [5]	2014	99.82	99.55	0.62
Dual slope [4]	2011	99.82	99.63	0.55
Combined adaptive threshold [22]	2004	99.78	99.78	0.44
Morphological filtering (VLSI) [16]	2012	99.76	99.82	0.43
Multiscale morphological filtering [15]	2009	99.81	99.80	0.39
Two moving averages [18]	2013	99.78	99.87	0.35
Shannon energy envelope est. [17]	2012	99.93	99.88	0.20

(3) *Criterion 3*

Highest possible heart rate is around 220 beats per minute. After a QRS pulse is detected, it assumed it is physiologically impossible for another beat to occur before 0.27 s. Peaks detected after the maximum QRS width and before this refractory period expires are considered noise.

(4) *Criterion 4*

The period between consecutive beats can vary to a certain degree, but it is usually limited to a range. We empirically observed this variability to 85% of the previous period. So peaks that happened out of this allowed variability range are considered noise.

(5) *Criterion 5*

If noise was detected from the last pulse, then consider beat as noise if the beat is smaller than the maximum noise peak plus the threshold, or smaller than the last detected beat minus the threshold.

All the previous criteria are taking into account and described by the flow chart in Fig. 8.

4. Experimental results

The MIT/BIH arrhythmia database [2] is used to evaluate the performance of the algorithm. It contains 48 half-hour two-channel ambulatory ECG tape recordings recorded at 360 Hz sampling frequency with 11 bit resolution over ±5 mV range. The database also includes annotations with the occurrences of all heart beats and other interesting information.

To evaluate the detection performance, true positive (TP), false positive (FP), false negative (FN) are used. A true positive happens when an annotated beat is correctly detected for a certain instant. A false positive happens when a beat has been detected in a certain instant, but no beat has been annotated in the database for that instant. A false negative happens when the database reports a beat for a certain instant but the algorithm fails to detect it. With this values we derive the metrics of sensitivity (Se), positive prediction (+P), and detection error, which are defined by Eqs. (12), (13) and (14) respectively.

$$Se(\%) = \frac{TP}{TP + FN} \tag{12}$$

$$+P(\%) = \frac{TP}{TP + FP} \tag{13}$$

$$DER(\%) = \frac{FP + FN}{Total\ QRS} \tag{14}$$

Table 1 reports the performance of the algorithm on all the MIT/BIH database recordings. We have considered all annotated beats in the annotation files from the MIT/BIH. They give a total number of 109,494 annotated beats.

The detector gives very good results on most records, except on records 108, 203, and 207. The detection errors on record 108 are mainly caused by high frequency noise. Record 203 is also affected by some high frequency noise but also by some sudden changes of the amplitude of some beat pulses that do not reach threshold levels, producing false negatives. Finally we get significant false positive errors in record 207 during the flutter episodes.

Table 2 reports the sensibility, positive prediction and detection error rate for the proposed algorithm and compares it with other high quality algorithms. It must be noted that our method is above the 99% threshold for Se and +P that is used in [3] to classify good quality algorithms. We report the performance of the full algorithm but also the performance obtained when the triangular filter is not used. The reason for this is that the triangular filtering uses more resources than the rest of the algorithm, and, if we focus on a very low power implementation, the performance loss is not that much significant considering the resource savings. Moreover, without the triangular filter the algorithm is still considered a high quality one.

5. Implementation

All the methods are suitable for its implementation on portable embedded systems. The resource needs in terms of computational elements and storage elements of every processing step are described in Table 3.

The algorithm can be effectively implemented in hardware. The MaMeMi filter basically needs two registers to store the *max** and *min** values and some comparators to check if the input signal is higher or lower than the registered values. The range-module actually reuse these values to compute the output. Triangular filter is costly in terms of storage because a delay line must be build to compare the samples to the right and left of the sample of interest. Nevertheless, it could be sacrificed if the lower detection performance can be tolerated. Peak and valley detection is simply implemented by using comparators and simple finite state machines. Beat unification is straightforward. Finally beat detection is implemented with a simple algorithmic machine.

Table 3
Resource needs of every algorithm step.

Algorithm	Reg	Add	Sub	Count	Comp	Mult
MaMeMi Filter	2	3	1		2	
Range	1		2		1	
Triangular	30		1		1	
Peak & Valley	2				2	
Beat Uni.	1		1		1	
Beat detection	5			1	7	1
Total	41	3	5	1	14	1

Table 4

Comparison with other detectors.

Algorithm	Reg	Add	Sub	Count	Comp	Mult
[19]	21	23	0	3	1	1
[4]	n.a.	14	0	0	15	5
Proposal w/o. triangle	11	3	4	1	13	1
Full proposal	41	3	5	1	14	1

Compared with state of the art portable implementations (see Table 4), our algorithm is competitive. The number of resources is similar to or lower than the alternative proposals, although detection performance is slightly lower. We include the work from [19] as a reference although it has not been tested against the MIT-BIH.

High performance detectors that are not specifically targeting portable devices use some of the techniques mentioned in Section 2, consuming much more resources. For instance, just a step of the Pam & Tompkins algorithm [12] needs to compute the square root, which need much more resources than our full implementation.

6. Conclusion

We have presented the MaMeMi filter, a non-linear high pass filter useful to remove ECG baseline wander. We have created a simple QRS detector based on this filter with minimal resource needs. The algorithm is easily implementable both in software and in hardware.

The algorithm is tested against the MIT/BIH database achieving a detection rate of 99.22%, a sensitivity of 99.43% and a positive prediction of 99.67%.

Acknowledgments

This work is partially supported by the EU FP7 DocuMeet Project <http://www.documeet.eu/the-project>. We would like to express our thanks to Dr. Chit Cheng Yeoh for her support and fruitful discussions.

References

- [1] A. Gacek, W. Pedrycz, *ECG Signal Processing, Classification and Interpretation: A Comprehensive Framework of Computational Intelligence*, Springer, 2012.
- [2] G. Moody, R. Mark, The impact of the MIT-BIH Arrhythmia Database, *Eng. Med. Biol. Mag. IEEE* 20 (2001) 45–50.
- [3] B.-U. Kohler, C. Hennig, R. Orglmeister, The principles of software QRS detection, *Eng. Med. Biol. Mag. IEEE* 21 (2002) 42–57.
- [4] Y. Wang, C. Deepu, Y. Lian, A computationally efficient QRS detection algorithm for wearable ECG sensors, in: *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE, 2011*, pp. 5641–5644.
- [5] M. Arefin, R. Fazel-Rezai, Computationally efficient QRS detection analysis based on dual-slope method, in: *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE, 2014*, pp. 2274–2277.
- [6] M. Elgendi, B. Eskofier, S. Dokos, D. Abbott, Revisiting QRS detection methodologies for portable, wearable, battery-operated, and wireless ECG systems, *PLOS ONE Public Libr. Sci.* 9 (2014) e84018.
- [7] N. Ravanshad, H. Rezaee-Dehsorkh, R. Lotfi, Y. Lian, A level-crossing based QRS-detection algorithm for wearable ECG sensors, *IEEE J. Biomed. Health Inform.* 18 (2014) 183–192.

- [8] M. Kaur, B. Singh, Seema, Comparison of different approaches for removal of baseline wander from ECG signal, in: *Proceedings of the International Conference & Workshop on Emerging Trends in Technology, 2011*, pp. 1290–1294.
- [9] S. Rani, A. Kaur, J.S. Ubhi, Comparative study of FIR and IIR filters for the removal of Baseline noises from ECG signal, *Int. J. Comput. Sci. Inf. Technol.* 2 (3) (2011) 1105–1108.
- [10] H. Chen, S. Chen, A moving average based filtering system with its application to real-time QRS detection, *Comput. Cardiol.* 2003 (2003) 585–588.
- [11] L. Keselbrener, M. Keselbrener, S. Akselrod, Nonlinear high pass filter for R-wave detection in ECG signal, *Med. Eng. Phys.* 19 (1997) 481–484.
- [12] J. Pan, W.J. Tompkins, A real-time QRS detection algorithm, *IEEE Trans. Biomed. Eng.* BME-32 (1985) 230–236.
- [13] S. Kadambe, R. Murray, G.F. Boudreaux-Bartels, Wavelet transform-based QRS complex detector, *IEEE Trans. Biomed. Eng.* 46 (1999) 838–848.
- [14] D. Benitez, P.A. Gaydecki, A. Zaidi, A.P. Fitzpatrick, The use of the Hilbert transform in ECG signal analysis, *Comput. Biol. Med.* 31 (September) (2001) 399–406.
- [15] F. Zhang, Y. Lian, QRS detection based on multiscale mathematical morphology for wearable ECG devices in body area networks, *IEEE Trans. Biomed. Circuits Syst. IEEE* 3 (2009) 220–228.
- [16] C. Zhang, T.-W. Bae, VLSI friendly ECG QRS complex detector for body sensor networks, *J. Emerg. Sel. Top. Circuits Syst. IEEE* 2 (2012) 52–59.
- [17] M. Manikandan, K. Soman, A novel method for detecting R-peaks in electrocardiogram (ECG) signal, *Biomed. Signal Process. Control* 7 (2012) 118–128.
- [18] M. Elgendi, Fast QRS detection with an optimized knowledge-based method: evaluation on 11 standard ECG databases, *PLOS ONE, Public Libr. Sci.* 8 (2013) e73557.
- [19] T.-T. Hoang, J.-P. Son, Y.-R. Kang, C.-R. Kim, H.-Y. Chung, S.-W. Kim, A low complexity, low power, programmable QRS detector based on wavelet transform for implantable pacemaker IC, in: *2006 IEEE International SOC Conference, 2006*, pp. 160–163.
- [20] K. Suarez, J. Silva, Y. Berthoumieu, P. Gomis, M. Najim, ECG beat detection using a geometrical matching approach, *IEEE Trans. Biomed. Eng.* 54 (2007) 641–650.
- [21] S.-W. Chen, H.-C. Chen, H.-L. Chan, A real-time QRS detection method based on moving-averaging incorporating with wavelet denoising, *Comput. Methods Progr. Biomed.* 82 (2006) 187–195.
- [22] I.I. Christov, Real time electrocardiogram QRS detection using combined adaptive threshold, *Biomed. Eng. Online* 3 (2004) 28.



David Castells-Rufas was born in Manresa, Spain, in 1971. He received the B.S. degree in Computer Science Engineering in 1994 and the M.S. degree in Microelectronics Design in 2009 from Universitat Autònoma de Barcelona (UAB). From 2003 he is working as Research Assistant with CEPHIS/CAIAC research center of the UAB, where he also teaches as associate lecturer. His research interests include reconfigurable systems, networks on chip, and many-core architectures. He founded the technology-based companies Histeresy and Creanium in 1998 and 2001 respectively.



Jordi Carrabina was born in Manresa, Catalonia in 1963. He graduated in Physics in 1986, and received M.Sc. (1988) and Ph.D. (1991) degrees in Microelectronics (inside the Computer Science Program), at the University Autònoma de Barcelona. In 1986 he joined CNM, with a national fellowship where he was collaborator researcher until 1996. Since 1990, he is associate professor at the Computer Science Department of the University Autònoma de Barcelona, heading the research group Embedded Computation in HW/SW Platforms and Systems Laboratory, and the CEPHIS technology transfer node from the catalan IT network. Main interests are embedded platform-based design using hardware software co-design and SoC/NoC

Architectures facing integrated and reconfigurable devices. He has given courses in several universities in Spain, Europe and South America. He has been consultant for companies as Hewlett-Packard Barcelona, Mitsubishi Electric, Epson Electronics, Technimagén and many SME in Spain.