

# Computationally efficient time domain detection algorithm for characteristic points in non invasive continuous blood pressure measurements

Lerch D.<sup>1</sup>, Orglmeister R.<sup>2</sup>

<sup>1</sup>Electronics and Medical Signal Processing, TU Berlin, Berlin, Germany, dennis\_lerch@gmx.de

<sup>2</sup>Electronics and Medical Signal Processing, TU Berlin, Berlin, Germany, reinhold.orglmeister@tu-berlin.de

**Structure:** 1. Introduction / 2. Methods / 3. Results / 4. Conclusion / 5. References

## Abstract

In this paper a computationally efficient algorithm for continuous blood pressure curve segmentation is presented. It uses only methods in the time domain and can distinguish between systolic, diastolic values and values of calibration steps caused by the continuous blood pressure measuring technique or values of other artefacts. The detection of local extremes, necessary for systolic and diastolic points, is performed with smoothed first and second derivations. An adaptive threshold approach sorts out most of the false extremes, not belonging to the valid blood pressure curve. But only the following plateau detection is suitably reliable to detect local extremes which lie within a calibration step.

## 1 Introduction

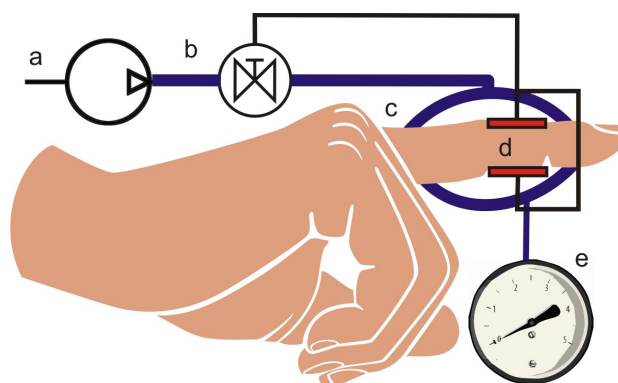
A lot of vital parameters, for example the heart rate, blood pressure, oxygen saturation and others, must be monitored in every day clinical environments. Interesting points like the R-peak, systolic and diastolic blood pressure must be automatically determined to support the medical staff in evaluating the biosignal curves. Not only for stationary use, but also for mobile applications the algorithms must be computationally efficient. In particular, home monitoring devices used for long-term recordings with low computing performance and a limited power source cannot handle complex memory and/or computationally intensive algorithms.

The continuous blood pressure measurement is sometimes preferred evaluating the morphology of the curve. Apart from providing the systolic and diastolic values, it enables advanced analysis. Frequently, the volume-clamp method is used, which was first introduced by J. Peñáz [1] and technically realised into Portapres® [2] device.

In Figure 1, the volume-clamp method is shown. It enables measuring of finger arterial blood pressure. A finger cuff combined with an automatically inflatable bladder controlled by a fast servo valve are used. With an infrared photoplethysmographic sensor, the actual volume of the finger artery is measured and the valve is set to compensate the expansion and adjust the volume to a predefined set point. Hence, the air pressure of the bladder is modulated by the arterial blood pressure.

To obtain a suitable set point and an appropriate blood pressure curve, the system must calibrate itself. During the calibration procedure, the transmural pressure must be determined, and the corresponding volume of the artery is used as set point. The transmural pressure is the equilibrium pressure between arterial blood and bladder air

pressure. Because of physiological effects like stress and changes in the tension of arterial muscles, the transmural pressure changes over time and the calibration must be done at certain intervals [2]. As a consequence, the measured air pressure curve, representing the blood pressure curve, contains stepped artefacts in addition to movement and other artefacts.



**Figure 1** Schematic of the volume clamp method. a) air-pump, b) servo valve, c) cuff with bladder, d) IR photoplethysmograph, e) bladder air pressure measuring. Hand illustration from Vectorportal.com.

Previous papers [3] and [4] suggest a computationally intensive wavelet transformation to find characteristic points, for example the systolic and diastolic pressure, and split them from stepped artefacts.

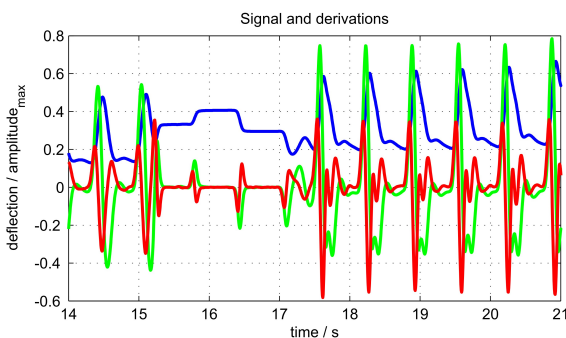
In this paper, an efficient time domain detection algorithm for characteristic points in non invasive continuous blood pressure measurements is introduced. It is based on an adaptive threshold algorithm and a plateau detection.

## 2 Methods

The algorithm can be divided into four steps. First, signal preprocessing is performed. In the second step, possible local extremes are determined which could be systolic or diastolic values respectively. In the third step, an adaptive threshold approach is used to sort out false maxima. Finally, the fourth step checks if the remaining local maxima lie in on a plateau. All calculation steps and their segmentation results will be visualised with the same signal segment. During the interval from 15 s to 17 s, calibration steps are shown (see Figure 2). For a better comparison between the individual curves themselves, the curves deflections are all scaled to their maximum value.

### 2.1 Signal preprocessing

To get only local extremes of systolic and diastolic blood pressure values, small noisy signal peaks must be smoothed. Local extremes are determined considering the zeros of the first derivation. A differentiated Gaussian filter kernel can do both, the smoothing and differentiation, at the same time. It is realized as a finite impulse response (FIR) filter. The weighted Gaussian averaging preserves the underlying signal course better than a comb filter because similar samples in the neighbourhood should have bigger influence on the actual sample amplitude than ones far away. In this application, a Gaussian filter kernel with a length of 51 samples and variance of 30 samples<sup>2</sup> fits our needs best. The filter has a linear phase response due to its symmetric impulse response and a normalized group delay of 25 samples. With the smoothed first derivation, a much easier filter kernel namely  $[0.5 \ 0 \ -0.5]$  can be used for calculating the second derivation. It is also a linear phase FIR filter and its normalized group delay equals 1 sample. The group delays are important to resynchronise the signals in time. Figure 2 visualises the resulting curves.



**Figure 2** Smoothed signal (blue) for a better visualisation and the first (green) and second derivation (red). Calibration steps from 15 s to 17 s are clearly visible.

### 2.2 Detection of local extremes

In the case of continuous curves, local extremes are detected where the first derivation becomes zero. In the discrete case, not only zeros of the first derivation can be

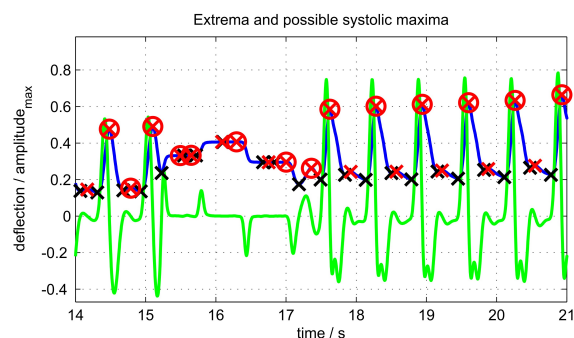
local extremes, but also zero crossing locations must be taken into consideration. Due to the discretisation, positions of local extremes may lie between two sampling moments. Every sample of the first derivation is checked for zero equality. A sign change is checked with the previous sample if it is not zero. If a sign change exists, in most cases it does not matter if the previous or the actual sample is used for an approximation of the real local extremes position. The error is negligibly small. To determine the type of local extremes, the second derivation is evaluated.

Sometimes it is very important to get the sub-sample position of an extremum generated by a zero crossing in the first derivation. If the second derivation is nearly zero and the zero crossing lies between a sign change in the second derivation, the selection of the position will strongly determine the local extremum type. It could be a local minimum, a local maximum or none of both. There are two possibilities to handle the situation.

Either, the local extremum can be discarded because it is so little marked as not to be an interesting point. But two local extremes of the same type can be sequenced and the following steps of the algorithm might have problems.

Therefore, the second possibility was chosen. A linear interpolation of the first and second derivation are performed to get a sub-sample local extremum position and the corresponding type.

For all local maxima, the span to the previous local minima must lie in the interval  $[0.1, \dots, 0.35]$  s or  $[30, \dots, 70]$  samples to be marked as possible systolic extremes. See Figure 3 to verify the found possible systolic extremes in this step.



**Figure 3** Smoothed signal (blue), first derivation (green), local maxima (red), local minima (black), possible systolic values (red circled).

The algorithm identifies some of the local maxima during the calibration steps as possible systolic values marked with a red circle. These false ones must be discarded in the following steps.

But other extremes are already discarded in this preselection.

### 2.3 Adaptive threshold

Maximum amplitudes  $x_{max}$  of the first derivation between all preselected possible diastolic minima and systolic

maxima are determined. An adaptive threshold based on the implementation in [5] is used to sort out false pairs of dias-/systolic extremes. In equation (1), the threshold  $\tau(n)$  is adaptively adjusted for each pair with index  $n$ .

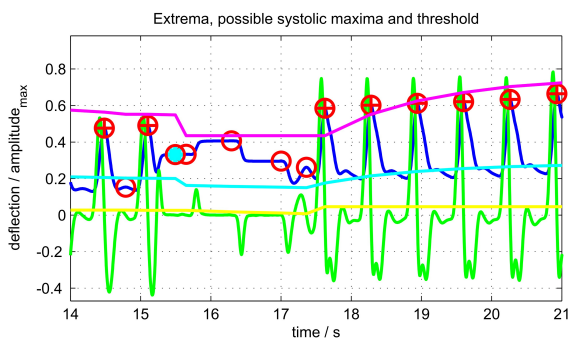
$$\tau(n+1) = \nu(n) + \frac{\sigma(n) - \nu(n)}{3} \quad (1)$$

The two values  $\sigma$  and  $\nu$  are separately adjusted depending on true or false detected pairs. If a true pair is detected,  $\sigma$  will be adjusted and otherwise  $\nu$  according to

$$\begin{aligned} \sigma(n+1) &= \frac{3}{8} x_{max}(n) + \frac{5}{8} \sigma(n) \\ \nu(n+1) &= \frac{3}{8} x_{max}(n) + \frac{5}{8} \nu(n). \end{aligned} \quad (2)$$

Empirically found initial thresholds are used at the beginning.

Especially for huge amplitudes of false pairs, the threshold can grow too strongly and following valid pairs can wrongly be discarded. To prevent huge thresholds, an upper limit is specified. If it is exceeded, the initial threshold value will be reset.



**Figure 4** Smoothed signal (blue), first derivation (green), possible systolic values (red circled), sigma (magenta), tau (cyan), nu (yellow), valid systolic values (red cross) and one misclassified systolic value (cyan dot).

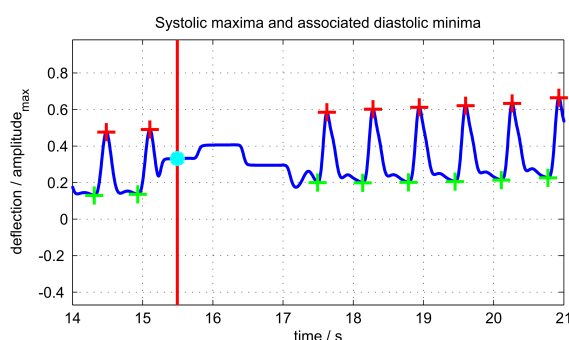
In Figure 4, the time course of the threshold is shown. For true possible systolic values  $\sigma$  is adjusted and for false values  $\nu$ , as it can be clearly seen between 17 s and 18 s. Only one misclassified systolic value still exists at 15.5 s which lies on the plateau of a calibration step.

## 2.4 Plateau detection

In most cases, possible local extremes caused by artefacts were discarded after applying the adaptive threshold approach. Only some extremes in calibration steps (as shown in Figure 5) can show properties similar to valid dias-/systolic pairs. A plateau detection sorts out these false pairs.

If at least 30 sequenced samples lie within 3 mmHg amplitude range, a plateau is detected. The detection is only done in the neighbourhood of a local maximum and not for each sample. Starting from the local maximum,

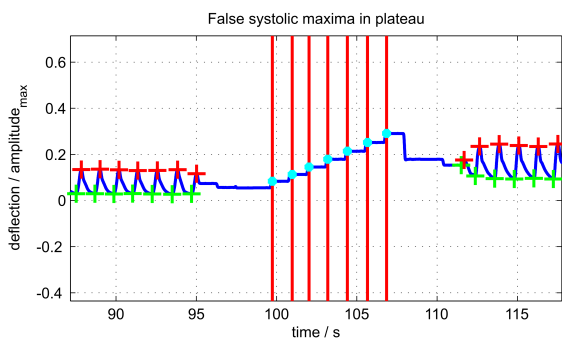
previous samples are investigated if they lie in the 3 mmHg amplitude range. If 30 samples are found, the local maximum will be marked as a plateau local maximum and the whole dias-/systolic pair is discarded. Otherwise, the following samples starting from the local maximum are checked, if they lie in the already spanned 3 mmHg amplitude range. As soon as a following sample is out of range and the range already contains at least 30 samples, the local maximum is also marked as a plateau local maximum. If there are less samples left, the new sample is used as new upper range limit. All entries which do not lie within this new spanned range will be excluded. This procedure is repeated as long as the local maximum is excluded or the range includes a number of samples greater or equal 30. In the latter case, the local maximum will be also marked as plateau local maximum.



**Figure 5** Smoothed signal (blue), systolic values (red cross), diastolic values (green) and systolic value in plateau (cyan dot with red line).

## 3 Results

The algorithm is tested for several continuous blood pressure curves, and the segmentation results are visually verified. In the second step, local maxima of artefacts with shapes differing from ordinary blood pressure curves are very reliably discarded. Valid dias-/systolic pairs show high deflections in the first derivation because of the steep rise in the continuous blood pressure curve. The rises of valid pairs are often even steeper than rises by calibration steps. An adaptive threshold algorithm can determine these differences very reliably. Only a few false remaining maxima must be detected within a plateau yet. Especially for long calibration periods (as shown in Figure 6), false systolic values after the adaptive threshold approach can occur.



**Figure 6** Smoothed signal (blue), systolic values (red cross), diastolic values (green) and systolic values in plateaus (cyan dot with red line).

## 4 Conclusion

Good results are achieved with less computing resources compared to methods using for example wavelet transformation. Typical calibration steps and other movement artefacts could be distinguished from characteristic continuous blood pressure values, like systolic and diastolic values.

In further implementations, not only the proximity of time between local minima and local maxima can be taken into consideration, but also the differences of their deflections. It is quite obvious that diastolic and systolic values differ

by about a few mmHg. But nevertheless, a plateau detection will be necessary for similar looking dias-/systolic pairs caused by calibration steps.

With slight modifications, the algorithm can be used for other signals as well. For example, the plethysmogram has a similar shape and only the interval boundaries and perhaps the threshold calculation must be adapted.

## 5 References

- [1] Penaz, J.: Photoelectric measurement of blood pressure, volume and flow in the finger. Digest of the 10th international conference on medical and biological engineering, 104, 1973
- [2] Finapres Medical Systems B.V.  
<http://www.finapres.com/site/page/2/13/Technology/>. - last checked 26.06.2012
- [3] Baas, T.; Koehler, H.; Malberg, H.; Doessel, O.: Automatic blood pressure segmentation algorithm for analysing morphology changes. In: Biomedizinische Technik / Biomedical Engineering, Proceedings BMT2010, 2010, 168–171
- [4] De Melis, M.; Morbiducci, U.; et al.: Blood pressure waveform analysis by means of wavelet transform. In: Medical, Biological Engineering and Computing 47, 2009, nr. 2, 65–173
- [5] Pan, J.; Tompkins, W.: A real time QRS Detection-Algorithm. In: Biomedical Engineering 32, 1985, 230-236