15th Triennial World Congress, Barcelona, Spain

ON LINE SEGMENTATION ALGORITHM FOR ICU CONTINUOUSLY MONITORED CLINICAL DATA

S. Charbonnier[(1)] G. Becq[(1)], L. Biot[(2)], P.Y. Carry[(2)], J.P. Perdrix[(2)]

[(1)]Laboratoire d'Automatique de Grenoble
BP 46, 38402 St Martin d'Hères France
tel : (33) 476-82-64-15 - fax : (33) 476-82-63-88 - email : Sylvie.Charbonnier@inpg.fr
[(2)]CHU Lyon SUD
Chemin du Grand Revoyet 69495 Pierre-Bénite Cedex FRANCE

Abstract : In this paper, an on-line segmentation algorithm, developed to preprocess continuously monitored data in Intensive Care Units, in a purpose of alarm filtering, is presented. The algorithm splits the signal monitored into linear affine functions of various lengths and determines on line when a new segment must be calculated. The delay of detection of a new linear function depends on the importance of the change : the more important the change, the quicker the detection.
The algorithm provides a good filtering of the data, without distortion, on simulated data as well as on clinical data recorded on patients admitted in ICU. The information returned by the algorithm can be used to extract on line information on the signal, like the trend of the signal, at short or long term. This useful information could be used in "more intelligent" alarm systems, like knowledge based systems.

Keywords : Biomedical engineering, data processing, monitoring element, filtering techniques, knowledge acquisition

1. INTRODUCTION

In intensive care units, monitoring systems are of grand interest for the medical staff. They measure and display on line physiological parameters, that give an information on the patient's state. An important task is to identify an abnormal physiological state as soon as possible and to warn the care giving personnel. To do so, monitoring systems are equipped with alarm systems, which consist commonly in a limit alarm system. When the value of the parameter monitored exceeds preset limits, an alarm is given. Unfortunately, these systems generate a large number of false alarms that are actually an extra burden to the care giving personnel, as reported in the literature (O' Carrol, 86). A reason for this is that variations can occur on the signal monitored that do not correspond to a physiological change but that are due to extraneous causes (measurement artefacts, patient turning in bed, cough …).

On the past decade, some work has been done to develop intelligent alarm systems for ICU, their goal being to assist clinicians in the interpretation of an alarm situation (Coiera 93, Uckun 94) The task of reducing the false alarms rate is an important part of it. Intelligent alarm systems require to extract the maximum information possible from the data available from the bed-side monitors: extraction of the trend in the signal monitored, pattern recognition on the signal (steady state, level change, slow increase ...), signal to symbol transformation so as to perform inferences with the information extracted from the clinical data (Avent and Charlton 90, Gordon 86, Imhoff et al 98)

Yet, a significant obstacle for using the monitored data in intelligent alarm systems is indeed the difficulty to extract reliable information from these data. On line recorded physiological parameters contain artefacts, natural fluctuations and transients that make the use of such techniques difficult in practise.
A common way to remove noise from on-line monitored data are linear filtering methods such as a low-pass filter or a moving averager. However, these tend to distort the signal during the transients. Another common method is the median filter, robust to artefacts. Yet, a good filtering requires large time windows that may create a delay that is too long in a purpose of alarm filtering (Makivirta et al 91).

This paper presents an on line segmentation algorithm developed to preprocess physiological data recorded continuously from the bed-side monitors. It splits the signal into linear segments of various lengths and returns useful information like the slope of the segment, its ordinate and its starting point. It provides a good filtering of the data, without distortion and is able to detect a quick change in the data. The information it returns can be used to extract on line information on the signal, like the trend of the signal, at short or long term.

In the first paragraph, a description of the algorithm is made. In the second paragraph, the results obtained on simulated signals are compared to those obtained by a moving average and a median filter. The results obtained on real data coming from different patients from ICU are analysed in the third paragraph.

2. DESCRIPTION OF THE SEGMENTATION ALGORITHM

Segmentation is a way to treat signals with quick non stationarities or sudden breaks and is a first step towards signal knowledge extraction. It consists in considering that the signal is composed of a succession of homogeneous segments of constant characteristics separated by abrupt transitions where the signal caracteristics change very quickly.

The segmentation algorithm developed here uses a segmentation well suited for trend extraction or pattern detection purposes. It consists in splitting the monitored data into successive affine linear functions of the form : $y(t)=p_i(t-t_{oi})+y_{oi}$

where t_{oi} is the time when the linear function begins, p_i is its slope and y_{oi} is the ordinate at time t_{oi}.

The principle of the segmentation algorithm is to determine on line the moment when the linear approximation is no longer acceptable and when to calculate the new linear function that now best fit the data. The technique used to detect if the linear approximation is still acceptable is the cumulative sum (CUSUM) technique. This technique, which consists in integrating the difference between the observed value and the current model, is very sensitive to changes of behaviour in the data. It makes the algorithm able to react quickly in front of sudden changes.

More details on the algorithm is given below (the algorithm is written in appendix):

Let us suppose that, at time t_1, the characteristics of a new linear function has been calculated, that is p_1, yo_1 and to_1.

k sample time later, $t_1+k\Delta t$, the model extrapolation is $\hat{y}(t_1 + k\Delta t) = p_1.(t_1 + k\Delta t - to_1) + yo_1$

and $e(t_1 + k\Delta t) = y(t_1 + k\Delta t) - \hat{y}(t_1 + k\Delta t)$ is the difference between the measurement and the extrapolation.

The cumulative sum of the difference calculated from time t_1 is

$$cusum\,(t_1 + k\Delta t) = cusum\,(t_1 + (k-1)\Delta t) + e(t_1 + k\Delta t)$$
$$= \sum_{j=0}^{k} e(t_1 + j\Delta t)$$

The absolute value of the cusum is compared, at each sampling time, to two thresholds, named th1 and th2.

If the absolute value of cusum($t_1 + k\Delta t$) is inferior to th1, the linear model is still acceptable.

If the absolute value of cusum($t_1 + k\Delta t$) is superior to th1, the signal value y($t_1 + k\Delta t$) and the corresponding time is stored in a block, named block of abnormal values.

If the absolute value of cusum($t_1 + k\Delta t$) is superior to th2, the linear model is no longer acceptable and a new linear function is calculated using least squares estimation on the values contained in the block of abnormal values, if the length of the block is superior to a certain number (at least 3). The values contained in the block of abnormal values correspond to the data $y(t_1 + k\Delta t)$, so that $th1 < abs(cusum(t_1 + k\Delta t)) \le th2$

Once a new linear function has been calculated, the cusum is reset to 0. An illustration of the technique is presented in figure 1.

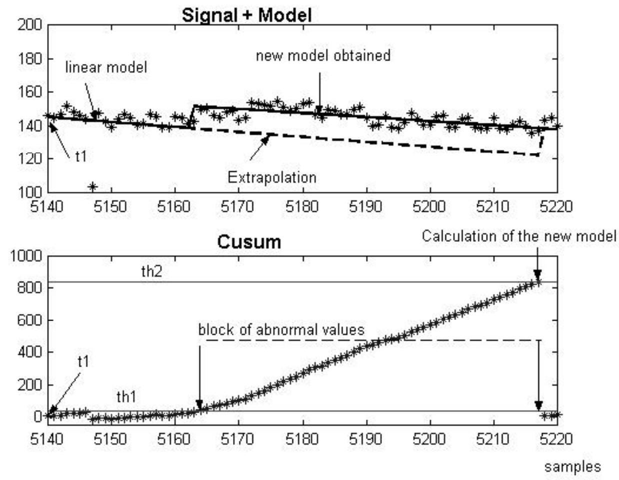

Figure 1 : Description of the technique used for segmentation

To prevent discontinuities in the filtered signal which would not correspond to a physiological behaviour, the algorithm analyses the continuity between two consecutive segments. When a new linear function has been calculated, the time of the intersection between this new function and the preceding one is immediatly calculated. If the time of intersection occurs after the current time or before the beginning of the preceding segment, the filtered signal is assumed to be discontinuous (ie there is a step variation between the two segments). The new segment starts at the instant when the value of the cusum crossed the first threshold (th1) for the last time. If the time of intersection occurs after the beginning of the preceding segment and before the current time, the beginning of the new linear function is the time of intersection, if the fit on the data, calculated with the CUSUM, is better this way.

Since the CUSUM technique is very sensitive to artefacts, it is necessary to reject them before the calculus of a new segment. Artefacts occuring on biological signals correspond to a sudden and important variation in the signal that can last for a few samples.

The algorithm rejects the artefacts in the following way. If the variation of the signal between two consecutive samples is measured larger than a fixed threshold (Ar1), the value of the preceding sample and the corresponding cusum is stored. If the difference between the next values and the value stored still remains important (superior to a second threshold Ar2) after a given time interval (TimeArtefact), the variation is considered a step and the whole data is considered for segmentation. Else, if the value of the signal decreases under Ar2 before TimeArtefact, the variation is considered an artefact, the data are removed from the block of abnormal values and the cusum is reset to the value stored before the artefact.

The segmentation algorithm has 5 tuning parameters, th1 and th2 to tune the decomposition into linear segments, Ar1, Ar2 and TimeArtefact to reject artefacts. An analysis of their effect on the decomposition into segments is presented in the next part.

3.RESULTS AND DISCUSSION

In order to analyse the results obtained, the segmentation algorithm is tested at first on simulated data, then on real data recorded on ICU patients.

3.1.Results on simulated data

A set of data was created, composed of 1000 simulations. Each simulation was composed of 2000 samples, corresponding to three successive linear functions, the parameters of the functions changing at time 500 and 1000. The change at time 500 between the first and the second function was continuous, whereas the change between the second and the third function, at time 1000 was discontinuous. To simulate biological rythms that can be present in the clinical data, a sinusoidal function with a period of 20 samples was added to the data, in addition to white noise.

At each simulation, the parameters of the three linear functions (slopes and ordinates) were randomly chosen and so were the sinusoide and noise amplitude, which were respectively 3% and 10% of the value reached at time 1000 by the first function. Thus, the algorithm was tested with varying signal to noise ratios, depending on the parameters obtained for the first function. An example of a simulation is given in figure 2.

On the data of each simulation, were processed the segmentation algorithm, a moving average filter with a time window of 30 samples and 60 samples and a median filter with the same time windows. The euclidian distance between the filtered data obtained by each method and the three linear functions with the sinusoide and the noise removed (ie the signal to be extracted) averaged from the 60^{th} sample to the 2000^{th}, named D, was used to compare the results obtained by the three methods.

The tuning of the segmentation algorithm parameters was made automatically at each simulation. th1 was calculated on the first 60 samples, it corresponds to four times the maximum value obtained by the cusum on these samples and th2 was chosen as 10 times th1.

The median value of D, calculated from the 1000 simulations, is equal to 16.3 (standard deviation 13.7) when the segmentation algorithm is used. It is equal to 36.7 (standard deviation 16.1) with the 30 samples moving average and to 62.7 (standard deviation 29.2) with the 60 samples moving average. For the median filter, the results are respectively 39.5(standard deviation 18.0) with the 30 samples time window, 65.3 (standard deviation 30.2) with the 60 samples time window. The results obtained by the segmentation algorithm are better (D strictly smaller) than the other methods for at least 940 simulations over the 1000 (94%).

The results obtained on these simulated data show that the segmentation algorithm is able to correctly eliminate the sinusoidal and the random components of the signal without distortion of the deterministic part, when it is composed of linear parts.

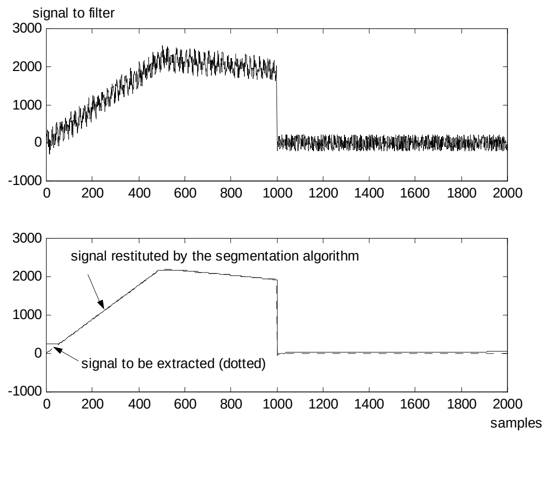

Figure 2 : An example of a simulation, and the result obtained with the segmentation algorithm

3.2. Results on real data

Clinical data were recorded at a frequency of 1Hz, on 18 patients admitted in the ICU service of Lyon-Sud Hospital Center. Each recording lasted 2 hours, during which the following signals were monitored : heart rate, systolic, diastolic and mean pressures, oxygen saturation and maximal pressure in the airways. The data were extracted from the analogical signals, with no prior treatment. During these recordings, a medical observer stayed bedside and analysed each alarm event that occurred.

For each clinical signals, a tuning set was proposed for the algorithm, that gave correct results for most patients.
The parameter th1 is a threshold that must be tuned taking into account the process noise. Indeed, it is an indicator of the moment when the current linear function is starting not to fit the data anymore. Assuming the noise is periodic, its value must be superior to the maximal value reached by the noise integral on a time interval at least equal to half a noise period. If the noise caracteristic (amplitude and period) is unknown, th1 may be estimated, for each patient, during the first minutes of the recording. The time window during which it is estimated must be long enough to acquire at least half a period of the slowest biological rythm and short enough for the linear estimation to be correct.

The parameter th2 determines the filtering effect. If it is small, a new segment will be calculated very often, and the filtering effect will be poor. Else, if it is too long, the algorithm will take a long time to detect a change in the data and some important variations may be filtered.
Because the algorithm will be used to process the data on line as a pre processor to an alarm filtering system, it is interesting to tune th2 in function of the delay required to detect a change in the data. It is an appreciation that can be given by clinicians. Chosen this way, th2 corresponds to the integral of the change to be detected on a time interval equal to the delay necessary to detect the change ($\Delta * T$ for a level change Δ in T samples, for instance).
For example, th2 was tuned to 600, when segmenting the heart rate. This means that the algorithm will take 60 samples (60s in this case) to detect a level change of 10 bpm. It will detect a slope change of 20 bpm in the same delay. However, it will take only 30 samples if it is a level change of 20 bpm. This is actually an advantage of the algorithm to adapt its reaction in function of the importance of the changes.

Parameters Ar1 and Ar2 are used to prevent the algorithm to react to artefacts. Their values can be tuned in function of the kind of artefacts that should be removed (amplitude and time of duration). For example, flushing the arterial catheter used to measure arterial pressure generates sudden variations in the systolic pressure that can last several seconds. Most of these artefacts were removed from the systolic pressure signal with the tuning Ar1=100mmHg, Ar2=90mmHg and TimeArtefact=8s.
The tuning proposed for each kind of signal is presented in table 1.

Table 1 : Set of tuning values proposed for each biological signal monitored

	PAS	PAM	PAD
th1	40	40	40
Δ level	14 mmHg	8mmHg	5mmHg
Time of detection (s)	60	60	60
th2	840	480	300
Ar1	100	45	45
Ar2	90	30	30
TimeArtefact	8	8	8

	HR	Spo2	Pmax
th1	12	1	10
Δ level	10 bpm	1%	1 mmH_2O
Time of detection (s)	60	60	180
th2	600	60	180
Ar1	150	11	60
Ar2	135	9	50
TimeArtefact	8	8	10

An example of the results obtained on real data is shown in figure 3. It presents a recording of the systolic and diastolic blood pressures, of the heart rate and pulse oxymetry, during 2 hours on the same patient, at a frequency of 1 Hz. We can see that the data are correctly filtered, and the deterministic variations in the signal are correctly extracted. At time 35 minutes, artefacts are correctly eliminated on the systolic and diastolic pressure signals. At time 110 minutes, the consequence of the flushing of the catheter can be observed on the signal of systolic and diastolic pressure. The segmentation algorithm eliminated it on the systolic pressure signal, but kept it as a level change on the diastolic pressure because the signal takes a few seconds to increase, which is not the definition for an artefact.
Using a limit alarm system with a preset limit fixed at 50 mmHg for the diastolic pressure generates several false alarms during the first hour of recording. The alarms would be detected as false (and would not ring) if the filtered data were used by the alarm system. Only one alarm would be ringed at time 70 minutes. This alarm corresponds to an effective decrease of the diastolic pressure, for which the care giving personnel had to react by giving medicine to the patient.

On the Spo2 data, two episodes of hypoxemia occur, at time 5 minutes and 45 minutes. The first episode, though it lasts a short time, is a real event for the patient : the respirator was deconnected for a short time. It is correctly restituted by the segmentation algorithm which took 20s to detect this event. Then, at time 70 minutes, an artefact occurs which is filtered by the algorithm

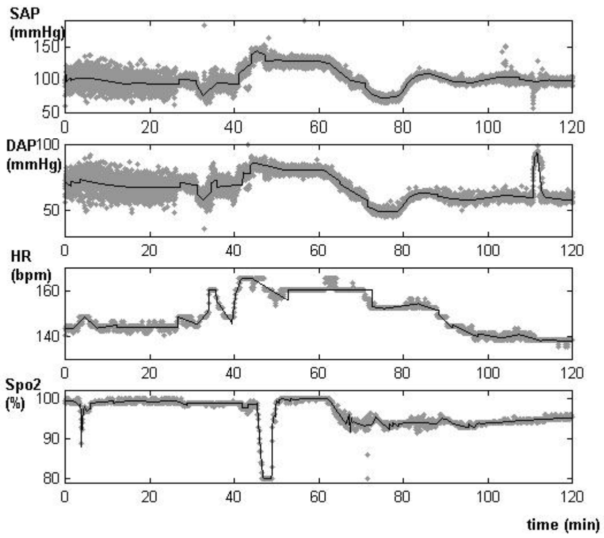

Figure 3 : Biological signals and the corresponding output from the segmentation algorithm

4. TREND EXTRACTION

The segmentation algorithm delivers information on the signal, that can be used to extract information on the signal variations. In this paragraph, it is shown how trends can be calculated on line with the algorithm.
We define the trend of the signal on the time interval T as the increase observed on the segmented data from time tnow-T to tnow, tnow being the current time, divided by the time window.

$$trend(ti) = \frac{1}{T} \sum_{k=t_i-T}^{k=t_i-1} \left[s(k+1) - s(k) \right]$$

Because the segmented data are not polluted with noise, it is possible to calculate the trend on any time window, even very short, which is difficult to do with classical methods.
The results obtained with the segmentation algorithm are compared to those obtained when the trend is estimated by a best-fit least square affine function, calculated on a moving block of length time T.
The trend is calculated on the Spo2 signal, which is a parameter for which deterministic variations can be very sudden, when a desaturation in oxygen occurs. It is can be interesting to know what was the evolution of the signal during the last minute, for example.

Results obtained are presented on figure 4. The trend calculated with the second method is affected by the noise corrupting the Spo2 signal, whereas the trend calculated with our method is not. The two hypoxemia episodes are clearly visible on both trends, but the trend calculated with the second method is also sensitive to an artefact occuring around 70 minutes. This would generate a false alarm if used in an alarm system.

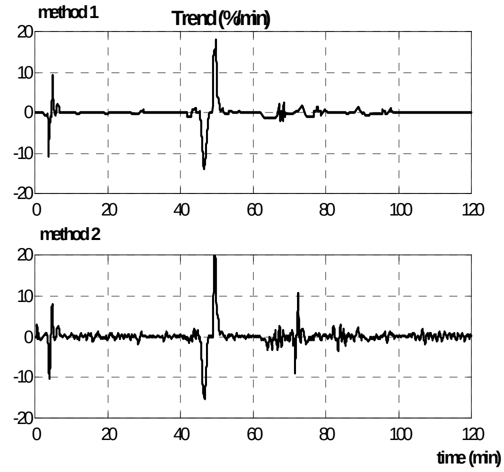

Figure 4 : Trend calculated with the segmentation algorithm and with best-fit least squares

5. CONCLUSION

The segmentation algorithm presented in this paper is an interesting tool to pre process on line monitoring data. It is able to correctly remove noise, periodic components, artefacts and transients corrupting the data and restitutes an undistorted filtered signal. It has the ability to react promptly in front of important changes, without being too reactive to transients. It is rather easy to tune, with two sets of parameters, one for the filtering effect, the other for artefact rejection. A set of tuning is proposed in this paper which gave good results for the 36 hours of recordings we dispose of.

Using the algorithm as a pre processor to alarm systems seems interesting. By removing artefacts and transients on the data, it can reduce the number of false alarms in limit alarm systems. The information it delivers can be used to determine on line the trend of the signal, at any time length, even very short,. This is an information very useful for “more intelligent” alarm systems, like knowledge-based systems, which take into account the evolution in time of the signal. We are now studying a methodology for pattern extraction, using the segmentation algorithm.

The segmentation algorithm is also a powerful tool to compress data, in a purpose of storage. Indeed, the output of the segmentation algorithm consists in three parameters (the slope, the ordinate and the starting time of the linear function), which enable the reconstruction of the filtered signal. The size of a file containing the 5 monitored data at 1hz during 2 hours is about 300 ko, it decreases to 7 ko when the successive affine functions parameters are stored.

ACKNOWLEDGMENT

This work is supported by the national network for health technologies 2000-2003, from the french research ministary

REFERENCES

Avent R, Charlton J (90) A critical review of trend-detection methodologies for biomedical systems *Critical reviews in Biomedical engineering* **17**, 621-659

Coiera E (93) Intelligent monitoring and control of dynamic physiological systems *Artificial intelligence in medicine* **5**, 1-8

Gordon K. (86) The multi sate Kalman filter in medical monitoring *Computer methods and programs in Biomedicine* **23**, 147-154

Imhoff M., Bauer M, Gather U, Lohlein D (98) Statistical pattern detection in univariate time series of intensive care on_line monitoring data *Intensive care med.* **24**, 1305-1314

Makivirta A, Koski E, Kari A, Sukuvaara T (91)The median filter as a preprocessor for a patient monitor limit alarm system in intensive care *Computer methods and programs in Biomedicine* **34**, 139-144

O'Carrol T (86), Survey of alarms in an intensive therapy units *Anaesthesia* **41**, 742-744

Uckun S.(94) Intelligent systems in patient monitoring and therapy management A survey of research projects *Int. Journal of clinical monitoring and computing* **11,** pp. 241-25

APPENDIX

```
i= current time
Acquisition of the signal
   signal(i)=[i,value(i)];
Calculation of the first derivative
      der(i)=value(i)-value(i-1)
Research of artefact
   if abs(der(i))>Ar1&(n==0)
&(fla~=1),
      CUS1=cusum;
      n=1;
      d=value(i-1);
   end
   fla=0;
   if (n>=1)&(n<TimeArtefact),
      n=n+1;
      if abs(d-value(i))<Ar2,
         cusum=CUS1;
         n=0;
         valartefact=[];
         fla=1;
      end
   end
   if n==TimeArtefact,
      n=0;
      fla=1;
Abpts=[Abpts;valartefact]
      valartefact=[];
   end
         Calculus of cusum
   extrapolation=p1*(i-xo1)+yo1;
   difference=value(i)-
extrapolation;
   cusum=cusum+difference;
        if cusum > threshold1
if (abs(cusum)>th1)&(n==0),
Abpts=[Abpts;signal(i)]
 nAbpts=nAbpts+1;
        if cusum > threshold2
if(abs(cusum)>th2)
&(nAbpts>nAbptsmin),
    calculation of the new linear
              function
[p2,yo2]=LinearApproximation(Abpts(
:,2));
   calculation of the intersection
       between the 2 functions
xo2=Abpts(1,1);
xo2p=floor((yo2-yo1+p1*xo1-
p2*xo2)/(p1-p2));
determination of the new parameters
in function of the intersection and
        the fit on the data
if
(xo2p>Abpts(length(Abpts(:,1)),1))|
(xo2p<xo1)|(xo2p<=0),
p1=p2;
xo1=Abpts(1,1);
yo1=yo2;
else
   if (xo2p>=xo1)&
(xo2p<=Abpts(1,1)),
  calculation of the best fit with
             the cusum.
      Decision of continuity or
      discontinuity between the
        consecutive functions
    else
if
(xo2p>Abpts(1,1))&(xo2p<=Abpts(leng
th(Abpts(:,1)),1)),
  calculation of the best fit with
             the cusum.
      Decision of continuity or
      discontinuity between the
        consecutive functions
end
end
end
     Resetting of the parameters
              cusum=0;
              nAbpts=0;
              Abpts=[];
end
       else
        If cusum <threshold1,
           nAbpts=0;
           Abpts=[];

end
```