Sensor Signal Preprocessing Techniques for Analysis and Prediction

Gustavo Monte
Universidad Tecnológica Nacional - Regional Académica Confluencia – Plaza Huincul - Argentina
gmonte@frn.utn.edu.ar

Abstract- **This paper presents a signal processing technique that employs oversampling and identification of important samples to determine signal behavior and tendency. Sensor signal windows of random lengths are vectorized and classified to fit into only eight predefined types, and in conjunction with time indexes vectors, they can predict future values, steady state value and an estimation of the sensor signal function. The techniques developed allow the representation of any class of sensor signal for further analysis. The computational cost is quite low so they can be implemented in real time into smart sensors with low cost microcontrollers. Therefore, it is also an ideal technique to preprocess the sensor signal to mark regions of interest to more sophisticated processes.**

I. INTRODUCTION

Today, connecting a sensor to a data acquisition system involves a complex process. Sensors have become smart in the way that involve signal processing capability at the point of measurement, together with the digital data transmission. Sensors are the most important components in any monitoring and control system; they measure physical parameters and convert them into signals [1]. Without reliable information from the sensors, the system can not take right decisions.

In order for the system to interpret the sensor data correctly, it must have access to a variety of parameters. These include the range, scale, and measurement units for the sensor, bandwidth, calibration details, and more. A lot of efforts have been carried out to standardize the connection of sensors. As a result, the IEEE 1451 standards were developped to automate this process by incorporating the necessary data as part of the sensor itself, having the sensor carry its own data allows the data acquisition system to automatically configure itself correctly [2]. [3].

The information that a smart sensor can transmit to the control and monitoring system goes beyond the measured signal value. In the same way that the IEEE 1451 standard, signal parameters themselves should be standardized. For example, it would be of great interest to know signal behavior like tendency, shape, future values, steady state value, presence of impulsive noise, optimal sampling frequency, variance and mean value. Also it would be important to know where signal segments are suitable for additional or time consuming processes. Examples of such processes are high order statistics, wavelet, spectrum analysis and any other signal processing technique that is not worth to process when, for example, the signal is constant.

There is an explosion of interest in mining time series databases [4] [6]. As with most computer science problems, representation of the data is the key to efficient and effective solutions. The proposed algorithm is well suited for data mining processes and it is adapted to sensor signals where a control algorithm is applied based on sensor information.

II. ALGORITHM DESCRIPTION

A. Segmentation algorithm

The signal preprocessing algorithm is based on the degree of redundancy that can be observed in an oversampled sensor signal. In a process of regular sampling, not all the samples have the same degree of importance [5]. Considering, at first glance, that a sample does not carry important information it is substituted by linear interpolated sample. This operation generates an error, i.e. the difference between the real and the interpolated sample. By computing the error for all the samples, the non-interpolated ones are tagged as significative when the error exceeds a threshold. Also, the sample time index is tagged and included as a new component of an index vector.

The algorithm starts interpolating the sample between two samples, i.e. interpolates sample 2 from 1 and 3. Then, the error, as the difference between the real sample and the interpolated sample, is computed. If the error is less than a prefixed value, the sample has no relative importance due to the fact that its value can be obtained from its neighbors. Now, the sample 3 is interpolated from 1 and 5 and so on. Two kinds of errors are computed. First, the replacement error and second, an accumulative error generated by the samples replacement process.

If the error exceed a threshold, the sample is tagged as important and the process starts again now from this sample. The interpolation is achieved by linear interpolation. If no important sample is found at modulus N, the last sample is tagged and the process starts from this sample.

To make the algorithm description more specific and clear, the pseudocode is presented as follow. Let $x(n)$, the sensor signal sampled at uniform sampling. The sampling rate chosen must greater than the Nyquist frequency, by at least a

factor of 10. The oversampling is necessary if the segment shape analysis is performed as it is shown in section B. Let $\mathcal{E}_{\max}$ the total permissive error and ε_i the sample interpolation error.

Algorithm ***LIN1***

Initialization:

$k = 2$
$n = 1$
$etotal = 0$; *accumulative error*
$mark(n) = x(n)$; *tagged samples vector*
$timepos(n) = n$; *time index vector*

Parameters:

$\mathcal{E}_{\max}$; *max accumulative error*
ε_i ; *max interpolation error*
N ; *max segment size*

Procedure:

```
LOOP FOREVER
    error=abs ( (x(n)+x(n+k))/2 – x(n+k/2))
    etotal=etotal + error

    IF (error < εi AND etotal < Emax AND k<N )
        k=k+2
    ELSE

        n=n+k/2
        mark(n)=x(n)
        timepos(n)=n
        k=2
        etotal=0

    END

END
```

If none of the error conditions were found for N consecutive samples, the last sample is tagged as important and the segment is consider finished. As a result of the algorithm, two vectors are obtained, the tagged samples vector $mark(n)$ and the time index vector $timepos(n)$.

The algorithm is extremely simple, nevertheless it allows a fast signal segmentation and a platform for the segment shape analysis. Only a division by 2 is involved in the average computation and in the sample index.

Two examples are shown. Fig 1. shows an oversampled sensor signal and the tagged samples obtained from the algorithm LIN1. Note that this signal portion can now be represented by two vectors of 7 elements, and the signal information is embedded in these vectors.

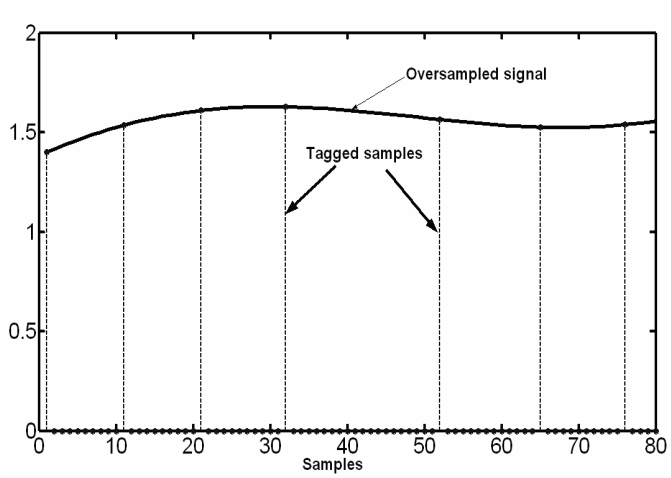


Fig 1. Oversampled sensor signal and tagged samples as result of algorithm LIN1 with $\varepsilon_i = 0.03$ and $\mathcal{E}_{\max} = 0.1$.

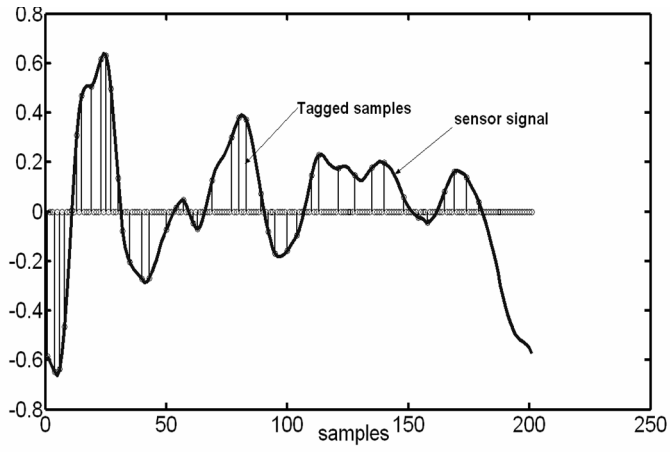


Fig 2. Signal sensor without oversampling and tagged samples as result of algorithm LIN1 with $\varepsilon_i = 0.03$ and $\mathcal{E}_{\max} = 0.1$.

The number of elements, the samples time indexes, the samples value and the first difference of samples is raw information that can be processed to determine signal behaviour as explained in section B.

In Fig 2. a non-oversampled sensor signal was processed by the algorithm. This example is presented to remark how the tagged samples point to shape changes in the signal. Note also that the signal could be rebuilt from its vectors.

B. Segment analysis algorithm

Once the signal was processed by the algorithm **LIN1**, a second process called Segment Analysis, **SA**, can be executed. The main reason for the previous process is to obtain the signal vectors for the **SA** process.

The goal of this algorithm is to analyze the signal behavior inside each segment. In this process the signal segments are considered independent from other segments. Let assume that the sensor signal was oversampled beyond its Nyquist frequency. Under this border conditions, the signal segment can experiment eight classes of behavior as shown in Fig 3.

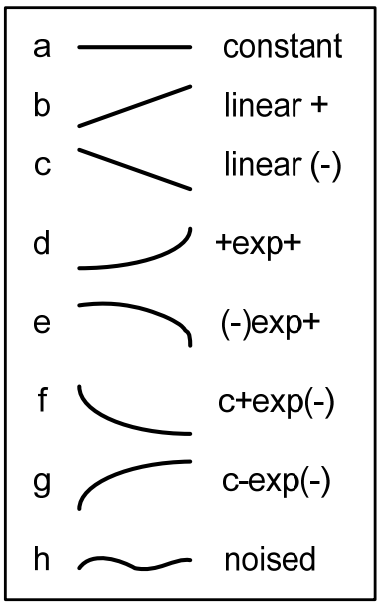


Fig 3. The eight classes of segment shape found in a oversampled sensor signal as a result of algorithm SA.

Fig 4. Algorithm SA for segment classification for obtaining eight predefined types.

The classes a, b and c are result of the segment size constraint of N samples. Therefore, in order to determine the class it is only necessary to calculate the difference between the left and right samples. For the rest of the classes, the difference between the interpolated sample and the real sample together with the difference between extreme samples are the key to classify them.

Fig. 4 shows the algorithm SA. It is an efficient process, and can be computed inside of the algorithm LIN1, therefore does not require extra calculation. The basic operation is based on the sign operator as shown in (1).

$$dif(i) = \sum_{k=k1}^{k2} sign[x'(n) - x(n)] \qquad (1)$$

Where $x'(n)$ is the interpolated signal; $k1, k2$ are the left and right sample indexes for segment i respectively, and $k2 - k1$ is the segment size.

With the signal segmentation and the segment classification processes, the sensor signal can be analyzed. In order to illustrate the algorithms, let analyze the signal in Fig. 5. Once processed, this signal can be represented by the following vectors:

$$timepos = [1, 6, 13, 23, 48, 67, 89, 118, 170]$$

$$mark = [2.05, 2.116, 2.174, 2.183, 2.11, 2.06, 2.03, 2.01, 2.00]$$

$$segment = [g, g, g, e, f, f, f, f]$$

With this raw information, a large amount of high-level processes can be carried out, like signal tendency or signal prediction. Moreover, the vectors could be the input to an artificial neural network, a Fuzzy logic system or to an artificial intelligent software.

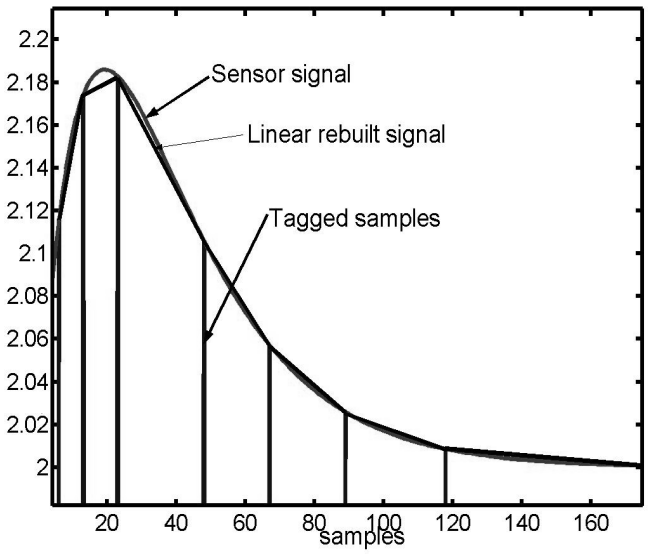


Fig 5. Algorithms LIN1 and SA applied to a test signal with $\varepsilon_i = 0.02$ and $\varepsilon_{\max} = 0.07$.

C. *Intra-segment analysis*

The techniques developed allow the representation of any class of sensor signal with simplicity while retaining a level

of detail. For further analysis of the signal, it is necessary to process the sequence of segments, taking into account both the duration of them and the class of segment. The relationship among segments provides us some clues about the overall behavior of the signal. For example, with IF….THEN rules, the smart sensor can turn on an alarm after finding any of the following conditions:

a) There are more than ***j*** consecutive class "**d**" segments **(+exp+)** with decreasing segments length.

b) There are more than ***k*** consecutive class "**e**" segments **(-exp+)** with decreasing segments length.

c) There are more than ***l*** consecutive class "**h**" segments **(noised)** (Probably a sensor failure).

d) There are more than ***m*** consecutive class "**g**" segments **(c-exp-)** with increasing length segments and the predicted steady state exceeds a prefixed limit.

e) There are more than ***n*** consecutive class "**f**" segments **(c+exp-)** with increasing length segments and the predicted steady state falls below a prefixed limit.

Lets observe that the rules involve both the type and length of the segment. The segment length is an important parameter because it measures how the signal departs from a linear behavior, i.e. it measures the speed of change in the signal.

More rules can be established depending on the specific application. For example, rules based on signal slope, impulsive noise and a specific patterns of segment classes.

II EXAMPLES OF APPLICATION

This section presents the signal analysis commonly found in monitoring and control systems. The most important case is exponential behavior because sensors related to measurement of analog signals found in a control systems show this class of signal shape.

A. Function type exp(-n)

This type of signal is detected by the occurrence of consecutive segments type "g" or "f" with increasing (exponentially) lengths. Fig. 6 shows an example of this kind of function. The steady state value can be predicted, and it is an important value for control systems [7].

Once the exponential behavior has been detected, the steady state value can be predicted using the tagged samples and the slope in these points as shown in Fig. 7.

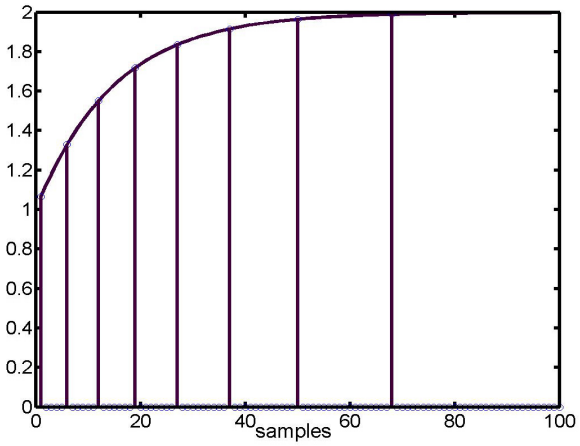


Fig. 6. The exponential behavior is detected through a sequence of segments class "g" of increasing length.

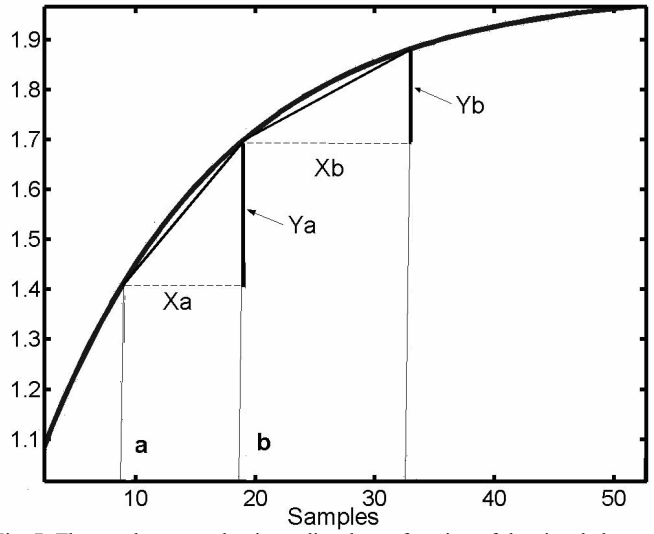


Fig. 7. The steady state value is predicted as a function of the signal slope in points ***a*** and ***b.***

A general expression for the curve in Fig 7. is

$$x(t) = Xss - (Xss - Xi)\, e^{-t/T} \qquad (2)$$

Where *Xss* is the steady state value, and *Xi* the initial value. The derivative of *x(t)* respect to *t* at *t=0* is

$$\frac{dx}{dt}\Big|_{t=0} = \frac{Xss - Xi}{T} \qquad (3)$$

The slope of the segments can be used as an approximation of the derivative in (3) applied at the points **a** and **b**. If we consider the derivative at point **a,** and using this point as the initial value we obtain

$$\frac{Xss - Xia}{T} = Ma \qquad (4)$$

Where $Ma = Ya/Xa$ is the slope in point **a**. Repeating the analysis now in point **b** and using this point as the initial value we obtain

$$\frac{Xss - Xib}{T} = Mb \qquad (5)$$

Where $Mb = Yb/Xb$ is the slope in point b. Solving (4) and (5) for Xss yields (6). Observe that Xss is computed without knowing the time constant ***T.***

$$Xss = \frac{Ma\ Xib - Mb\ Xia}{Ma - Mb} \qquad (6)$$

Therefore, the steady state value can be predicted using segment parameters like the left and right segment samples and the signal value at the segment extremes. These values are outputs from the algorithm LIN1 i.e. the $mark(n)$ and the time index $timepos(n)$ vectors, and we need only two segments.

B. Function type exp(n)

In this case, the detection is based on the occurrence of consecutive segments of type "d" or "e" with decreasing (exponentially) lengths as shown in Fig. 8.

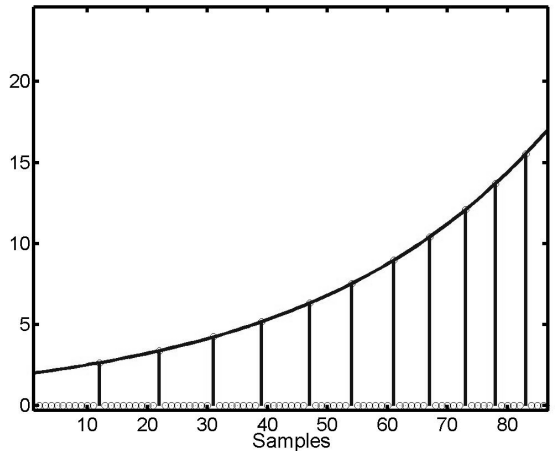


Fig. 8. The exponential behavior is detected through a sequence of segments class "d" of decreasing length.

C. Optimal sampling

In a time varying signal, the best way to deal with the correct sampling frequency is to filter beyond the region of interest in order to avoid aliasing. In most cases, the sampling frequency will be excessive respect to the signal bandwidth. In the proposed scheme, it is possible to find the lowest sampling frequency that avoids aliasing. Given a fixed error ε_i the optimal sampling for the sensor signal over a period of time is the sampling frequency multiplied by the shortest segment length.

D. Shot noise detection

Since the signal is oversampled, the presence of noise could be caused by an interference signal or a sensor failure.

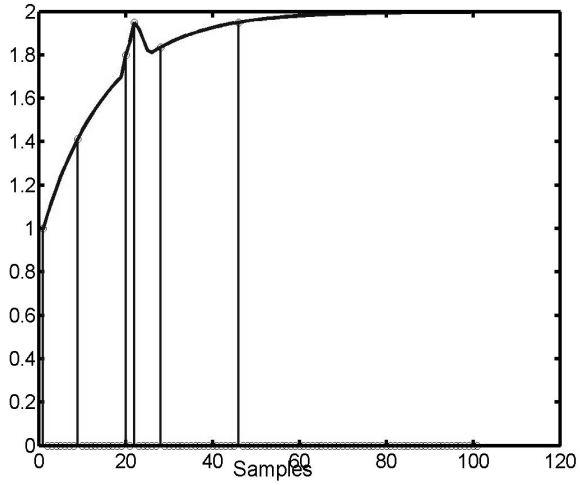


Fig. 9. The detection of shot noise is carried out through an abrupt reduction in the segment length.

Shot noise is detected by analyzing the sequence of segment lengths. Observe in Fig. 9 that the presence of impulsive noise causes an abrupt reduction of the segment length, and it can be easily detected.

E. Mean and variance estimation

The right and left samples represent the signal inside the segment. Considering that all the segments are class "b" or "c", an approximation of the mean of the segment will be the average between the right and the left samples, i.e. it allows a fast mean computation.

The variance can be approximated computing only the left and right samples because the inner samples are linear combination of them and do not add "randomness" to the signal.

F. Noise detection

A number of consecutive segments class "h" of short length could point out that the sensor signal is corrupted by noise.

III EXPERIMENTAL RESULTS

In order to test the algorithms, real sensor data were processed. We used a digital temperature sensor DS18S20 with a resolution of 0.1 ºC. The temperature inside a tank was used to test the algorithms. In Fig. 10 and Fig. 11 we can observe the heating and cooling curves, where the long term values were 90 ºC and 27.5 ºC respectively. Note that there is a noise signal that introduces errors in the predicted values. Even then the prediction errors are below 3%. The segment classification algorithm worked correctly, but if noise is present, we must allow that some of the samples have the incorrect sign i.e. the determination of the classes must be

based on the sign of the majority inner samples of the segment.

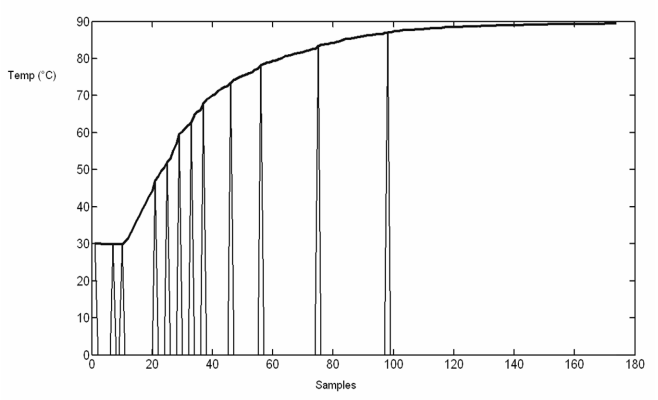
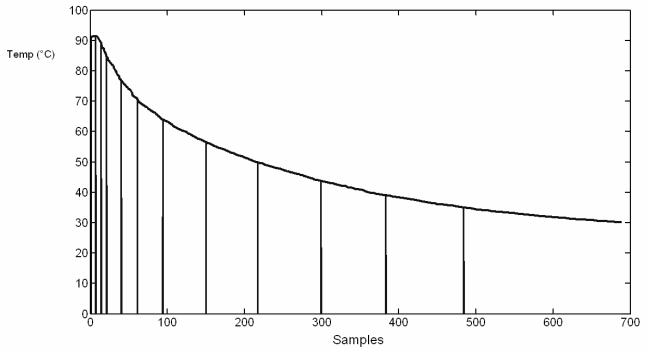


Fig. 10. The heating curve of a temperature sensor placed in a tank at 90



ºC. Also the tagged samples detecting exponential behavior with $\varepsilon_i = 1$ and $\varepsilon_{\max} = 10$.

Fig. 11. The cooling curve of a temperature sensor taked out from a tank at 90 ºC. Also the tagged samples detecting exponential behavior with $\varepsilon_i = 1$ and $\varepsilon_{\max} = 10$.

IV CONCLUSIONS

A technique for preprocessing a sensor signal was presented. The proposed algorithms are simple but powerful and offer a variety of parameters that extract important embedded information from the sensor signal. These algorithms can be easily executed into smart sensors since the computational cost is quite low.

A high degree of simplification is achieved with the segmentation of the signal into only eight classes. This is the key process to allow more sophisticated processes to be carried out. Some of them were presented in section II. Also, it is an excellent platform for data mining processes. Oversampling is necessary to get a high degree of correlation among samples. Under this condition, the signal segments are classified into only eight classes.

The algorithms detect signal regions were further analysis is not worth to be carried out. So, it is well suited to mark subsets where other processes could be executed.

The results in this paper emphasizes the fact that, in the same way that the IEEE 1451 standard was achieved, it should be standardized signal preprocessing techniques to supplement the information provided by smart sensors to the control and monitoring system.

ACKNOWLEDGMENT

The author is grateful to Mr. Gustavo Poblete for the data logging of temperature signals.

REFERENCES

[1] R. Wynn. " The TEDS smart sensor revolution". *IEEE Computing and Control Engineering*. August/September 2004. Vol 15, Issue 4, pp 25-27.

[2] B. Betts. " Smart sensors". *IEEE Spectrum on line*. April 2006.

[3] P. Cleaveland. "What is a smart sensor". *Control Engineering*. January 1, 2006.

[4] E. Keogh, Chu,S. and Hart,D. "An online algorithm for segmenting time series". Proceeding IEEE International Conference on Data Mining ICDM 2001. pp 289-296.

[5] Marten D. van der Laan. "*Signal sampling techniques for data acquisition in process Control"*. Thesis Rijksuniversiteit Groningen. - 1995. ISBN 90-367-0502-9

[6]]E. Keogh, M Pazzani "An Enhanced Representation of Time Series Which Allows Fast and Accurate Classification, Clustering and Relevance Feedback". *Fourth International Conference on Knowledge Discovery and Data Mining (KDD'98)* . pp 239-241, 1998.

[7] F. Bertling, S. Soter, "Real-time prediction of the steady state temperature of circuit components as a tool for power electronic circuit testing". *PCIM Europe 2007 Conference, Nuremberg, Germany.*