Dynamic Principal Component Analysis in Multivariate Time-Series Segmentation

Zoltán Bankó, László Dobos, János Abonyi[1]

[1] Department of Process Engineering, University of Pannonia, POB. 158, Veszprém, H-8200, Hungary

Abstract. Principal Component Analysis (PCA) based, time-series analysis methods have become basic tools of every process engineer in the past few years thanks to their efficiency and solid statistical basis. However, there are two drawbacks of these methods which have to be taken into account. First, linear relationships are assumed between the process variables, and second, process dynamics are not considered. The authors presented a PCA based multivariate time-series segmentation method which addressed the first problem. The nonlinear processes were split into locally linear segments by using T^2 and Q statistics as cost functions. Based on this solution, we demonstrate how the homogeneous operation ranges and changes in process dynamics can also be detected in dynamic processes. Our approach is examined in detail on simple, theoretical processes and on the well-known pH process.

Keywords: time series segmentation, Dynamic Principal Component Analysis, multivariate dynamic processes

1 Introduction

As PCA based methods have gained reputation in the past few decades not only amongst process engineers but in almost every area of science, their limitations have become more obvious. PCA defines a linear projection of the data and it is not able to consider process dynamics. Various solutions were proposed to address limitations arise from linear projection (kernel-function approach (Webb, 1996), generative topographic mapping (Bishop et al., 1998), etc.). Other popular alternative of these global non-linear methods is to split the data to locally linear segments and to use the PCA model of these segments. However, as (Kivikunnas, 1998) reported, *although in many real-life applications a lot of variables must be simultaneously tracked and monitored, most of the segmentation algorithms are used for the analysis of only one time-variant variable*.

Using univariate time-series segmentation methods for multivariate time-series cannot be precise enough, since the correlation between the variables is not a negligible factor when a PCA based method is applied. In such cases, the correlation can be treated as a hidden process which carries the information. E.g. creation of a simplified model of a high-density polyethylene plant requires more than 10 process variables to be tracked. Under monitoring, the signals (polymer production intensity, hydrogen input, ethylene input, etc.) are compared to the stored patterns to detect any sign of malfunctions. However, comparing the monitored signals to their counterparts only are often not enough because deviation in one or more signals does not mean malfunction automatically. Thus, Principal Component Analysis (PCA) based similarity measures are used to compute and compare the hidden processes in real-time and the applied segmentation algorithm should follow the same logic, i.e. it has to put the focus on the analysis of this hidden process.

(Abonyi et al. 2005) were the first, as we are aware of, who utilized the two PCA related statistics (T^2 and Q) as cost functions in a bottom-up segmentation algorithm. With this approach, it was possible to segment multivariate time-series based on changes in the correlation structure between the variables, i.e. to extract locally linear segments.

Although this solution gave the possibility to segment multivariate time-series according to the needs of PCA, it shall not be forgotten that PCA is created for analyzing steady state processes, thus it is not able to handle any process dynamics. However, it seems this fact is rarely considered before any PCA based technique is applied. This conveys that changes in process dynamics are not discovered and, what is even worse, it can also lead to wrong conclusions.

To address such problem different solution were proposed. (Ku et al., 1995) suggested the application of PCA to an extended data matrix which contains past values of each variable, and named this method to dynamic PCA (DPCA). (Negiz & Cinar, 1997) proposed using a state space model based on Canonical Variate Analysis (CVA) and (Simoglou et al., 2002) presented an approach for the monitoring of continuous dynamic processes which involved CVA and Partial Least Squares (PLS). Although, (Negiz & Cinar, 1997) also pointed out that DPCA can be severely affected by process noise and requires more computational power, DPCA is still a powerful tool for environments with less noise and it provides a sound basis.

The aim of this paper is to develop the dynamic extension of our multivariate segmentation algorithm that can handle the dynamics of time-varying multivariate data and to segment these times series based on the changes in process dynamics.

In the next section we shortly review the basics that are necessary for our novel method and we show how the process dynamics can be considered using PCA based tools. We present our segmentation approach on two simple, theoretical examples and finally the segmentation of a the well-know pH process is examined in detail.

2 PCA Based Time-Series Segmentation

A multivariate time-series is much more than a set of univariate times-series considered in the same time horizon. Contrary to univariate time-series in which the variable represents the process itself and the processes can be analyzed based on it, multivariate time-series are not only described by their variables but the correlation between them. This correlation can be treated as a hidden process which carries the real information, thus the analysis of the underlying processes should be based on it.

The most frequently applied tool to discover such information is PCA (Tipping & Bishop, 1999). Fundamentally, it is a linear transformation that projects the original (Gaussian distributed) data to a new coordinate system with minimal loss of information. More information of its properties and application possibilities can be found in (Jackson, 1991).

To create a projection based on the correlation, PCA selects the coordinate axes of the new coordinate system one by one according to the greatest variance of any projection, while it does not consider the local changes in the correlation. Thus, application of the above mentioned non-linear methods or segmentation is required if such changes can be assumed.

2.1 Times-series segmentation

An n-variable, m-element multivariate time-series, $X_n = [x_1, x_2, \ldots, x_n]$, is an m-by-n element matrix, where $x_i = [x_i(1), x_i(2), \ldots, x_i(m)]^T$ is the i^{th} variable and $x_i(k)$ is its k^{th} element. $X_n(k) = [x_1(k), x_2(k), \ldots, x_n(k)]$ is the k^{th} sample of X_n. The i^{th} segment of X_n is a set of consecutive time points, $S_i(a,b) = [X_n(a), X_n(a+1), \ldots, X_n(b)]^T$. The c-segmentation of time-series X_n is a partition of X_n to c non-overlapping segments, $S^c_{Xn} = [S_1(1,a), S_2(a+1,b), \ldots, S_c(k+1,m)]^T$. In other words, a c-segmentation splits X_n to c disjoint time intervals, where $1 \leq a$ and $k \leq m$.

Segmentation can be framed in several ways but its main goal is always the same: finding homogenous sections by the definition of a cost function, $cost(S_i(a,b))$. This function can be any arbitrary function which projects from the space of time-series to the space of non-negative real numbers. In practice, $cost(S_i(a,b))$ is usually based on the (Euclidean-) distances between the actual values of the time-series and the values of a simple function *f* (constant or linear function, a polynomial of a higher but limited degree) fitted to the data of each segment:

$$\mathrm{cost}(s_i(a,b)) = \frac{1}{b-a+1}\sum_{k=a}^{b}(d(X_n(k), f(X_n(k))) \qquad \mathbf{(1)}$$

Thus, the segmentation algorithms simultaneously determine the parameters of the models, which represent the actual segment, and the borders of segments by minimizing the sum of costs of individual segments:

$$\mathrm{cost}(s_{X_n}^{c}) = \min\left(\sum_{i=1}^{c}(\mathrm{cost}(S_i(a,b)))\right) \qquad \mathbf{(2)}$$

This segmentation cost can be minimized by dynamic programming, which is computationally intractable for many real datasets. Consequently, heuristic optimization techniques such as greedy top-down or bottom-up techniques are frequently used to find good but suboptimal *c*-segmentations:

- Bottom-Up: Every element of X_n is handled as a segment. The costs of the adjacent elements are calculated and two elements with the minimum cost are merged. The merging cost calculation of adjacent elements and merging are continued until some goal is reached.
- Top-Down: The whole X_n is handled as a segment. The costs of each possible split are calculated and the one with the minimum cost is executed. The splitting cost calculation and splitting is continued recursively until some goal is reached.
- Sliding Window: The first segment starts with the first element of X_n. This segment is grown until its cost exceeds a predefined value. The next segment then is started at the next element. The process is repeated until the whole time-series is segmented.

All of these segmentation methods have their own specific advantages and drawbacks. E.g. the sliding window method is not able to divide up a sequence into predefined number of segments but this is the fastest method. (Keogh et al., 2001) examined these heuristic optimization techniques in detail through the application of Piecewise Linear Approximation (PLA). Their results carried out if real-time (on-line) segmentation is not required, the best result can be achieved by Bottom-Up segmentation. Thus, we use this method throughout this paper.

The pseudo code of bottom-up algorithm is the following:

1 Create initial fine approximation.
2 Find the cost of merging for each pair of segments: $mergecost(i)=cost(S_i(a_i,b_{i+1}))$
3 while $min(mergecost)<maxerror$
  3.1 Find the cheapest pair to merge: $i=argmin_i(mergecost(i))$
  3.2 Merge the two segments, update the a_i, b_i boundary indices, and recalculate the merge costs:
    $mergecost(i)=cost(S_i(a_i,b_{i+1}))$
    $mergecost(i-1)=cost(S_{i-1}(a_{i-1},b_i))$

2.2 PCA based segmentation

Any PCA based method shall be applied only if the examined process is linear. Unfortunately, many processes are not linear thus PCA should not be used. To overcome this problem, the authors suggested the segmentation of the non-linear processes to locally linear ones by using Hotelling's T^2 statistics or the Q reconstruction error as the measure of homogeneity of segments, i.e. to construct the cost function.

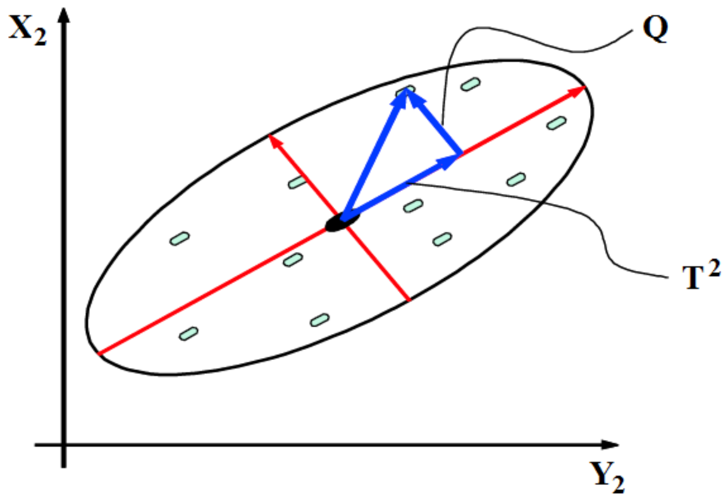

Fig. 1. Distance measures based on PCA model

Figure 1 shows the two distance measures in case of a time-series of 11 elements with 2-variables. The elements are represented with ellipses and the black spot marks the intersection of axes of principal components, i.e. the center of space which was defined by these principal components.

If the second principal component is ignored, two distances can be computed for each element. The first one is the squared Euclidean distance between the original data point and its reconstructed value using the most significant principal component only. Arrow Q represents this lost information which can be computed for the *k*th data point of the times series X_n as:

$$Q(j) = (X_n(j) - \underline{X}_n(j))(X_n(j) - \underline{X}_n(j))^T = X_n(j)(I - U_{X_{n,p}} U^T_{X_{n,p}})X_n^T(j), \qquad \textbf{(3)}$$

where $\underline{X}_n(k)$ is the k^{th} predicted value of X_n, I is the identity matrix and $U_{Xn,p}$ is the matrix of eigenvectors. These eigenvectors belong to the most important *p-n* eigenvalues of covariance matrix of X_n, thus they describe the hyperplanes. Please note, the Q error based segmentation can be considered as the natural extension of Piecewise Linear Approximation presented by (Keogh & Pazzani, 1999): both of them define the cost function based on the Euclidean distance between the original data and its reconstructed value from a lower dimensional hyperplane.

The second measure which can be used to construct the cost function is Hotelling's T^2 statistic that shows the distance of each element from the center of the data, hence it signs the distribution of the projected data. Its formula is the following for the k^{th} point:

$$T^2(k) = Y_p(k)Y_p(k)^T, \qquad \textbf{(4)}$$

where $Y_p(k)$ is the lower (p) dimensional representation of $X_n(k)$:

$$Y_p^T(k) = U^T_{X_{n,p}} X_n(k) \qquad \textbf{(5)}$$

The cost functions can be defined as:

$$\mathrm{cost}_Q(s_i(a,b)) = \frac{1}{b-a+1}\sum_{k=a}^{b} Q(k) \tag{6}$$

$$\mathrm{cost}_{T^2}(s_i(a,b)) = \frac{1}{b-a+1}\sum_{k=a}^{b} T^2(k) \tag{7}$$

Using these functions, the segmentation can be framed considering the two typical applications of PCA models. The Q reconstruction error can be used to segment the time-series according to the direct change of the correlation between the variables, while the Hotelling's T^2 statistics can be utilized to segment the time-series based on the drift of the center of the operating region.

3 Dynamic Principal Component Analysis (DPCA) Based Segmentation

The time variant property of dynamic processes cannot be handled by standard PCA based tools since PCA assumes no time dependency exists between the data points. This drawback of PCA motivates *Ku* (1995) to dynamize PCA for the needs of dynamic processes. Consider the following process:

$$Y_g^T(k+1) = A_1 Y_g^T(k) + \ldots + A_{ta} Y_g^T(k-t_a) + B_1 U_h^T(k) + \ldots + B_{tb} U_h^T(k-t_b) + C, \tag{8}$$

where $A_1, \ldots, A_{ta}$ *and* $B_1, \ldots, B_{tb}$ are g x g and g x h dimensional matrices, C is column vector, t_a and t_b show time dependence, $U_h(k)$ is the k^{th} sample of the (multivariate) input and $Y_g(k)$ is the (multivariate) output in the same time. In case of standard (static) PCA the multivariate time-series of a process is formed by the inputs (flow rates, temperatures, etc.) and the outputs (properties of the final product):

$$X_n = \lfloor Y_g, U_h \rfloor \tag{9}$$

Ku pointed out that performing PCA on the above X_n data matrix preserves time dependence of the original series and this obviously reduces the performance of the applied algorithm. He suggested that the X_n data matrix should be formed by considering the process dynamics at every sample points. Generally speaking, every sample points were completed with the points they can be depended on, i.e. the past values:

$$\begin{bmatrix} Y_g(t) & U_h(t) & \cdots & Y_g(t-n) & U_h(t-n) \\ Y_g(t+1) & U_h(t+1) & \cdots & Y_g(t+1-n) & U_h(t+1-n) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ Y_g(m) & U_h(m) & \cdots & Y_g(t+m-n) & U_h(t+m-n) \end{bmatrix} \tag{10}$$

3.1 DPCA based segmentation algorithm

Due to the process dynamics, linear relations exist between inputs and outputs. These relations are preserved under PCA as auto and cross correlations of the scores. Performing PCA on the above modified data matrix moves these unwanted correlations to the noise subspace because the possible combinations of the shifted input and output variables are presented in the data matrix and only the most important combinations of these are selected by PCA. Moreover, the first zero (or close to zero) eigenvalue shows the linear relationship between variables revealed by the eigenvector belongs to this eigenvalue. Based on these facts, in this section we show how a multivariate time-series can be segmented considering the process dynamics.

As it was presented in the previous section, the system-wide time relevance (dynamics) can be handled with the help of DPCA; however, one can say that the problem of the non-linear systems still exist. The only difference compared to our previously presented algorithm is that the "problem of non-linearity" becomes the problem of the process dynamics.

Thus, the DPCA based segmentation algorithm is simple but yet straightforward:

1. Create the "dynamized" data matrix of inputs and outputs in every sample point
2. Determine the number of latent variables for segmentation
3. Select the suitable PCA statistics (T^2 or Q) as segmentation cost
4. Apply our previously presented PCA based segmentation method

In the next two sections we show three different models on which our algorithm was examined. All of the models were created using Simulink and the algorithm itself was realized in Matlab environment.

4 Motivating Examples

Consider a first order, one variable, linear time variant system with the transfer function below.

$$G(s) = \frac{3}{20s+1} \tag{11}$$

Let the input of the system is a simple step function with some additional noise while the gain of the system (which is a proportional value that shows the relationship between the magnitude of the input to the magnitude of the output signal(s) at steady states, often called the sensitivity of the process) changes from *K=3* to *K=5* at *t=250*. Such input and the response of the system (i.e. the output) is shown in Figure 2.

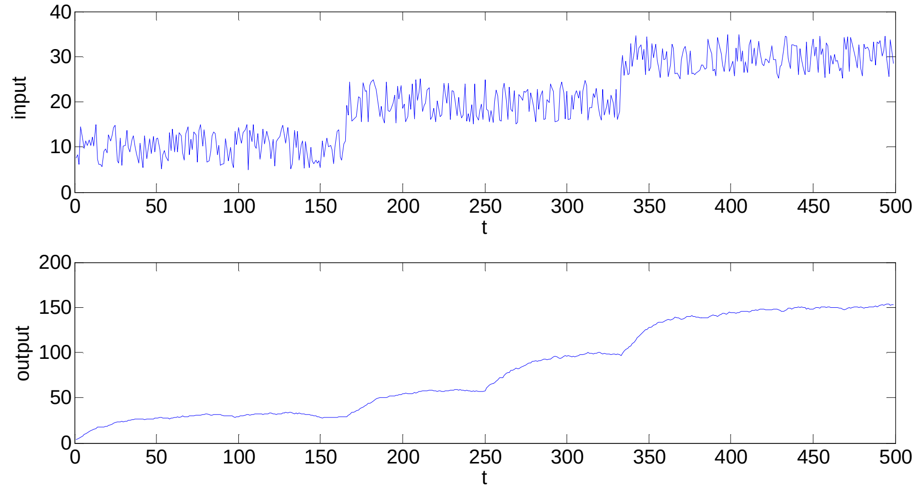

Fig. 2. Input and output of a first order, time variant system (gain changed at t=250)

To demonstrate the drawbacks of the standard (univariate) segmentation algorithms, time-series of Figure 2 was segmented with Piecewise Linear Approximation. As it can be noticed, there is no sense to segment input and output variables separately because the change of the gain can only be seen in the output, thus the input can be split to three segments while the output can be divided to four parts.

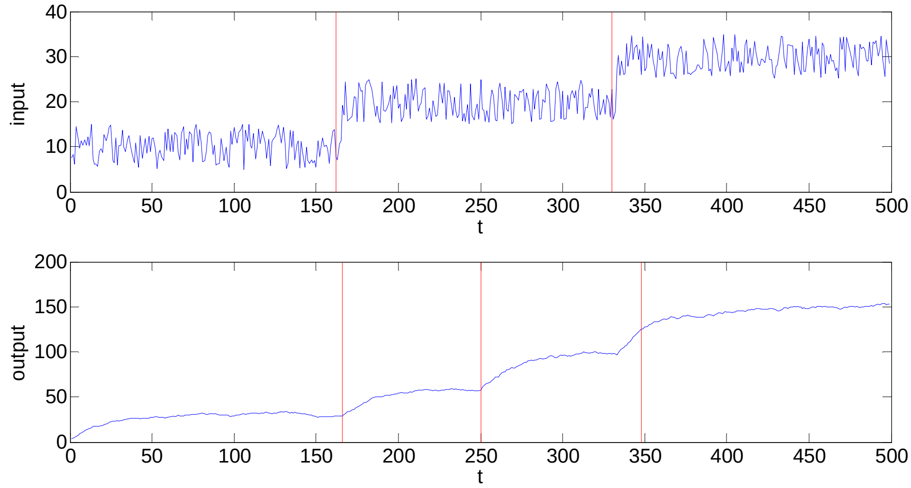

Fig. 3. Segmentation result using PLA

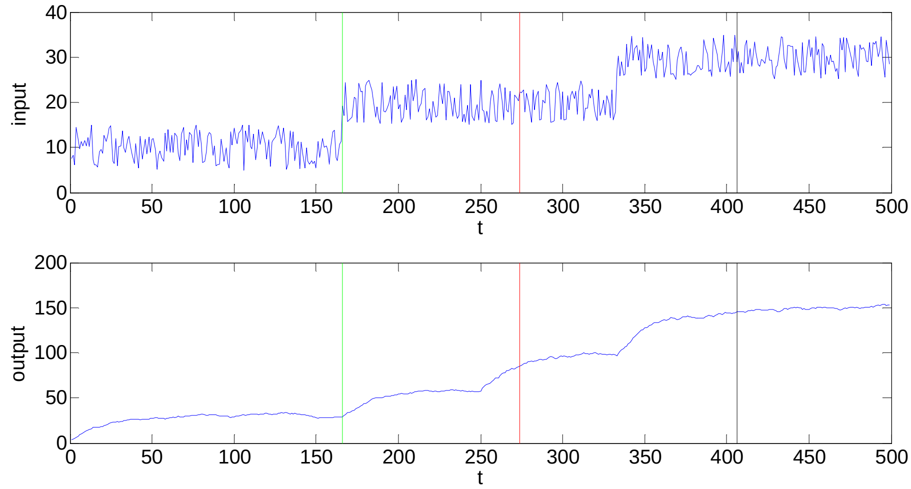

Fig. 4. Result of PCA based segmentation (1 latent variable and T^2 statistics were used). Red, green and blue colors show the limits of the segments according to their detection order, respectively.

Even if we try to segment these series using PCA based segmentation, we have to face the results shown in Figure 4. The limits of the segments are marked by red, green and blue colors, according to their detection order. The change of the gain is not detected precisely, although, the closest segmentation point is considered as the most important. It is easy to see that the detection of the segments is influenced by both input and output time-series equally, i.e. there are seven segments where the correlation between the time-series are changed (not considering the noise): input was changed two times, output was also changed two times but with a delay of t=1 and its middle segment can be split because of the change of the gain. It is obvious this kind of method cannot be precise enough for dynamic processes.

In case of the introduced DPCA based segmentation method, the results are more close to the expected segmentation, i.e. the process dynamics is changed only one time thus the limits of segments should be found around this changing point:

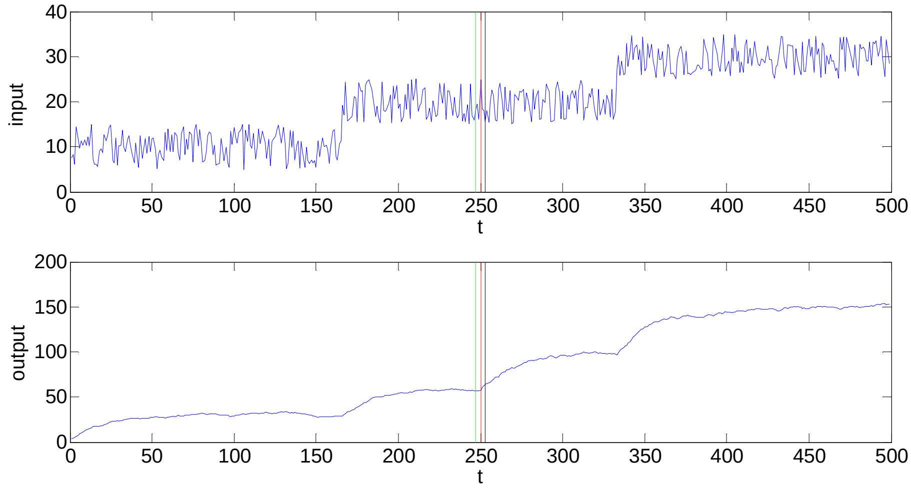

Fig. 5. Result of DPCA based segmentation (2 latent variables and T^2 statistics were used). Red, green and blue colors show the limits of the segments according to their detection order, respectively.

The segment was found first is marked by the red line as previously and it is "spot on". PCA based segmentation minimized the residuals or the average distance from the center of the data (depending on the selected segmentation cost). Using DPCA, the goal of segmentation becomes the problem of selecting the proper linear equation of a partial (local) system given by Equation 8. With the help of the bottom-up segmentation the initial segments are extended until their state equations properly describe them.

One of the biggest problems of every segmentation algorithm is to determine the number of segments. In the figure above, we can see an additional and very useful property of the DPCA based segmentation, namely the convergence of segments. In such case, the "unnecessary" segments are not located randomly, but close to the significant segments and their size is minimal. Usage of a segmentation algorithm with this property facilitates determining the number of significant segments.

As a second example, let us consider a second order, one variable, linear time variant system which was created by cascading two first order systems with the transfer functions:

$$G(s) = \frac{3}{10s+1} \tag{12}$$

$$G(s) = \frac{3}{40s+1} \tag{13}$$

The input of the system is the previous step function with some additional noise. The gain and the time constant of the first transfer function were changed from $K_1 = 3$ to $K_1=5$ and $\tau_1=10$ to $\tau_1=20$ at $t = 250$, while the same parameters were set from K_2=3 to K_2=5 and from τ_2=40 to τ_2=50 in case of the second system at t = *500*.

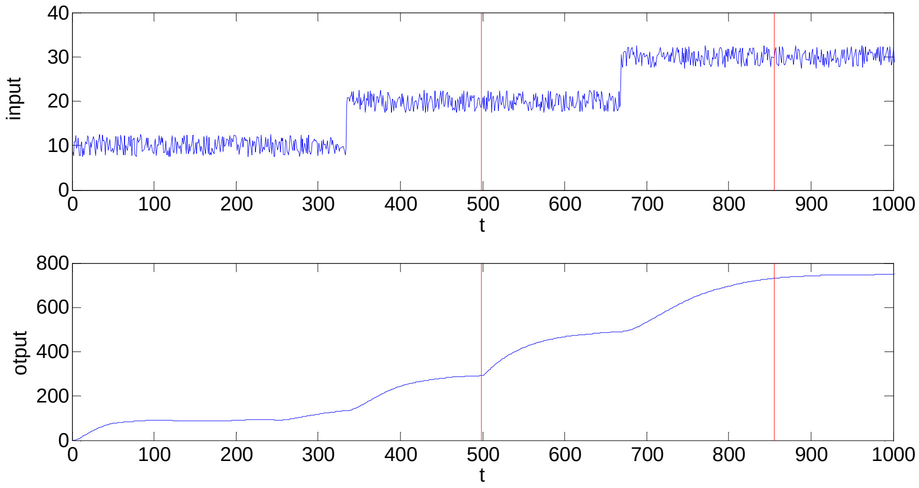

Fig. 6. Result of PCA based segmentation (1 latent variable and T^2 statistics were used)

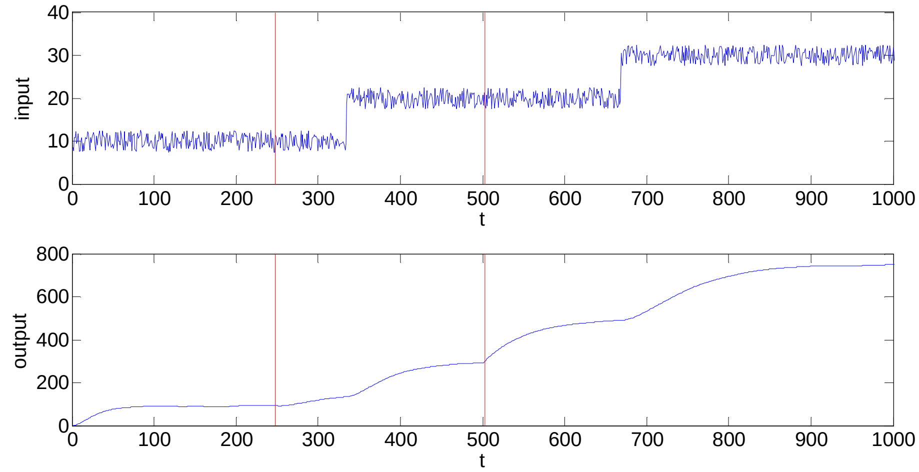

Fig. 7. Result of DPCA based segmentation (4 latent variables and T^2 statistics was used)

Our goal is to find these changes in the system dynamics using segmentation. Figure 6 shows the segment achieved by PCA based segmentation. Although the second changing point was recognized, the obvious correlation between the parallel ending sections created a segment instead of the change in the dynamics. Contrary to this, DPCA based segmentation meet our expectation and the changing points were detected precisely. To show the convergence of segments, we segmented the time-series to a higher number of segments (7) and, as it can be seen in Figure 8, the number of segments can be easily determined again using DPCA.

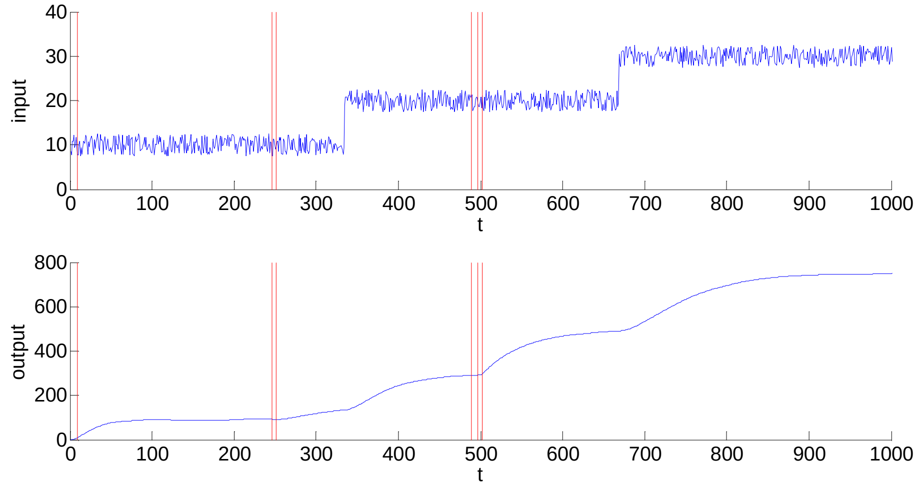

Fig. 8. Convergence of segments using DPCA based segmentation (4 latent variables and T^2 statistics were used). Although the process was changed two times only from dynamics point of view (at t=250 and at t=500), we are still able to determine these changes even if higher number of segments is used.

5 pH process: a case study

The modelling of pH (the concentration of hydrogen ions) in a continuous stirred tank reactor (CSTR) is a well-known problem that presents difficulties due to large variation in process dynamics. The CSTR is shown schematically in Figure 9.

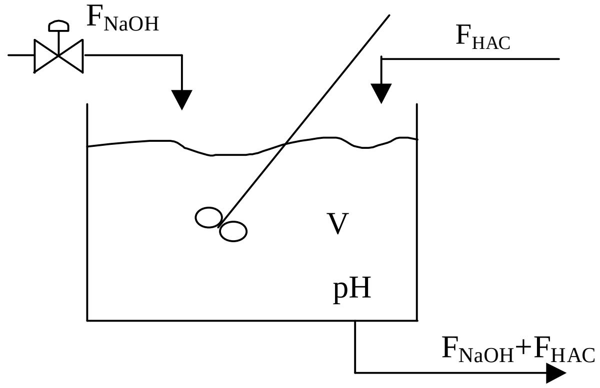

Fig. 9. Continuous stirred tank reactor

The CSTR has two input streams, one contains sodium hydroxide and the other is acetic acid. By writing material balances on [Na^+] and total acetate [$HAC+AC^-$] and assuming that acid-base equilibrium and electroneutrality relationships hold one we get:

- Total acetate balance:

$$F_{HAC}[HAC]_{in} - (F_{HAC} + F_{NaOH})[HAC + Ac^-] = V\frac{d[HAC + Ac^-]}{dt} \quad \textbf{(14)}$$

- Sodium ion balance:

$$F_{NaOH}[NaOH]_{in} - (F_{HAC} + F_{NaOH})[Na^+] = V\frac{d[Na^+]}{dt} \quad \textbf{(15)}$$

- HAC equilibrium:

$$\frac{[Ac^-][H^+]}{[HAC]} = K_a \quad \textbf{(16)}$$

- Water equilibrium:

$$[H^+][OH^-] = K_w \quad \textbf{(17)}$$

- Electroneutrality:

$$[Na^+] + [H^+] = [OH^-] + [Ac^-] \quad \textbf{(18)}$$

Equations 14 through 18 are a set of five independent equations which completely describe the dynamic behavior of the stirred tank reactor. The pH can be calculated from equations 16-18 as follows:

$$[H^+]^3 + [H^+]^2(K_a + [Na^+]) + [H^+]([Na^+]K_a - [HAC + Ac^-]K_a - K_w) - K_w K_a = 0 \quad \textbf{(19)}$$

$$pH = -\lg[H^+] \quad \textbf{(20)}$$

The pH process was modeled in Simulink environment and for simplicity, we set parameters according to (Bhat & Mcavoy, 1990):

Table 1. Parameters used in the simulation

Parameter	***Description***	***Nominal Value***
V	Volume of the tank	1000 l
F_{HAC}	Flow rate of acetic acid	81 l/min
F_{NaOH}	Stady flow rate of NaOH	515 l/min
[NaOH]in	Inlet concentration of NaOH	0.05 mol/l
[HAC]in	Inlet concentration of acetic acid	0.32 mol/l
[Na+]	Initial concentration of sodium in the CSTR	0.0432 mol/l
[HAC+Ac-]	Initial concentration of acetate in the CSTR	0.0432 mol/l
Ka	Acid equilibrium constant	$1.753 \cdot 10^{-5}$

When the input flow rate of the acetic acid is fixed (81 l/min in our simulation) the value of pH can be regulated by the input flow of NaOH only. Using a linearly increasing flow rate of NaOH, the process dynamics can be easily determined.

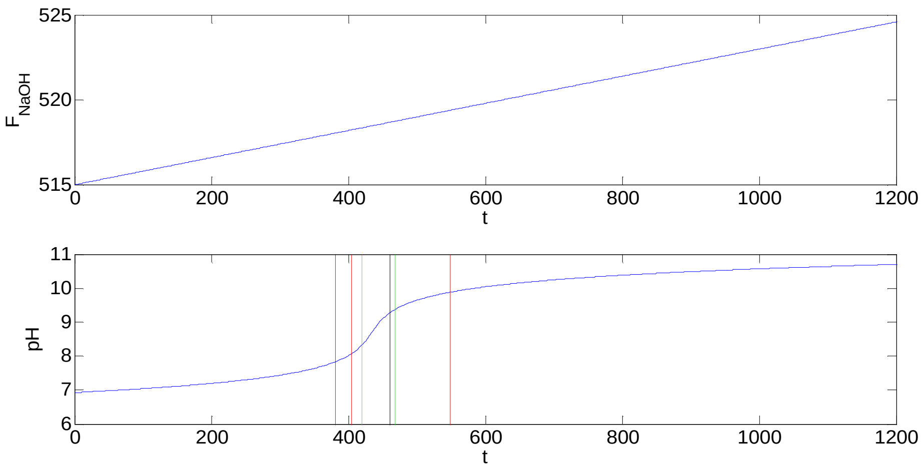

Fig. 10. Connection between the flow rate of NaOH (input) and pH (output). Segments found using DPCA, PCA and PLA are marked by black, red and green colors, respectively.

As it is shown in Figure 10, the gain of pH process changes two times. Obviously, these changes can be easily recognized with any of the above mentioned segmentation methods. The limits of the determined segments using PLA[1], PCA and DPCA methods are also presented in Figure 10. We can say that the found segments more or less met our expectations independently of the applied segmentation method.

1 The time-series of pH was segmented only. We could do this because the input is a linear function, thus it has no effect on PCA or DPCA based segmentation. However, when a non-linear input is used, it affects both multivariate methods, thus PLA was not computed because it cannot take both time-series into account.

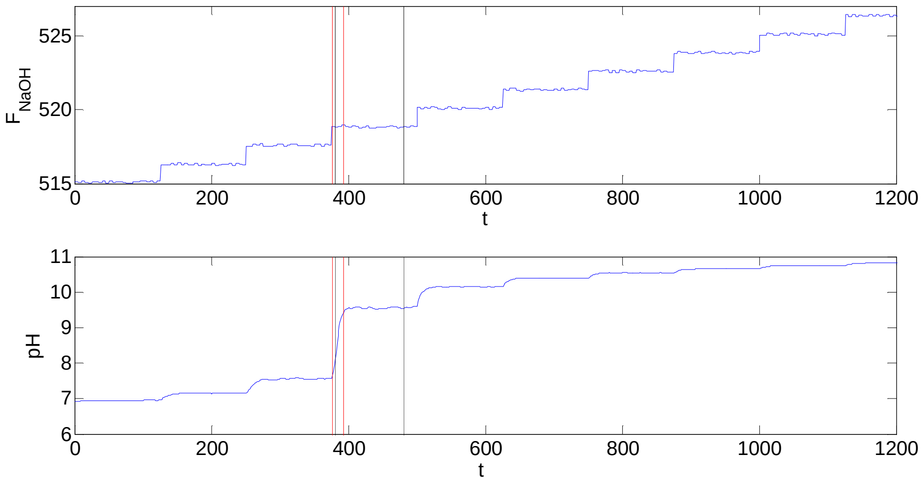

Fig. 11. Result of DPCA (black lines) and PCA based (red lines) segmentation (1 latent variable and T^2 statistics were used)

However, when the input is changed to another operation value, determining the changes in process dynamics is more difficult. Thus, we changed the linearly increasing input to a step function (with some additional noise) and executed the segmentation methods again, except PLA. Figure 11 shows how quickly the system reacts when the flow rate of NaOH was rapidly increased in the high gain interval. It can been also seen that the standard PCA based method focuses the rapidly growing output only; however, the higher gain is presented until the input is not increased again. This higher gain section was detected properly by the novel DPCA based segmentation. (Please compare Figure 10 and Figure 11 considering the detected high gain intervals.)

6 Conclusions

In this paper, a novel segmentation algorithm has been presented for PCA based methods. Our main goal was to create an algorithm which does not only address the problem of non-linear processes but to also provide solid basis for PCA based methods where process dynamics has to be considered. The presented solution is based on the covariance driven segmentation we introduced previously, i.e. where homogeneity measures have been used as cost functions for segmentation which are corresponding to the two typical applications of PCA models. The Q reconstruction error can be used to segment the time series according to the direct change of the correlation between the variables, while the Hotelling's T^2 statistics can be utilized to segment the time series based on the drift of the center of the operating region.

Utilizing Ku’s dynamized PCA, we were able to segment first and second order time-variant processes according to the changes in their gain. Our method was also verified on a (CSTR) pH modell where the high gain interval was detected properly. In addition, the presented method has two other useful properties. First, it makes possible to identify the locally linear models directly, and second, the number of the segments can be defined easily thanks to the convergence of the limits of the segments.

Acknowledgements This work was supported by the TAMOP-4.2.1/B-09/1/KONV-2010-0003 and TAMOP-4.2.2/B-10/1-2010-0025 projects and the E.ON Business Services Kft.

References

Abonyi J., Feil B., Németh S., Árva P. 2005. Modified Gath-Geva clustering for fuzzy segmentation of multivariate time-series. Fuzzy Sets and Systems.149(1): 39-56.

Bhat, N., Mcavoy, T. J. 1990. Use of Neural Nets for Dynamic Modelling and Control of Chemical Process Systems. Computers Chem. Eng. 14(4/5): 573-583.

Bishop, C. M., Svensen, M., Williams C. K. I. 1998. GTM: the Generative Topographic Mapping. Neural Computation. 10 (1): 215-234.

Jackson, J. E. 1991. A User's Guide to Principal Components. Wiley. New York

Keogh, E. J., Chu, S., Hart, D., Pazzani, M. J. 2001. An online algorithm for segmenting time-series. ICDM.

Keogh, E. J., Pazzani M. 1999. Scaling up dynamic time warping to massive datasets. 3rd European Conference on Principles and Practice of Knowledge Discovery in Databases. Vol. 1704. Springer. Prague. Czech Republic

Kivikunnas, S. 1998. Overview of Process Trend Analysis Methods and Applications. ERUDIT Workshop on Applications in Pulp and Paper Industry

Ku, W., Storer, R. H., Georgakis C. 1995. Disturbance Detection and Isolation by Dynamic Principal Components Analysis. Chemometrics and Intelligent Laboratory Systems. 30:179-196.

Negiz, A., Cinar, A. 1997. Statistical monitoring of multivariable dynamic processes with state space models. American Institute of Chemical Engineering Journal. 43(8):2002-2020

Simoglou, A., Martin, E.B., Morris A.J. 2002. Statistical performance monitoring of dynamic multivariate processes using state space modelling. Computers and Chemical Engineering 26:909-920

Tipping, M. E., Bishop, C. M. 1999. Mixtures of probabilistic principal components analysis. Neural Computation. 11: 443-482.

Webb, A. R. 1996. An approach to nonlinear principal components analysis using radially symmetrical kernel functions. Statistics and Computing 6(2): 159-168.