2012 20th Mediterranean Conference on Control & Automation (MED)
Barcelona, Spain, July 3-6, 2012

Design of Adaptive Sliding Window PI-PCA Controller

L. Brito Palma, F. Vieira Coito, and P. Sousa Gil

Abstract—**In this paper an approach to design adaptive proportional-integral (PI) controllers, for SISO systems, based on sliding window principal components analysis (PCA) is presented. Closed-loop control can be formulated and implemented within the reduced space defined by a PCA model. This PCA controller, results in an integral controller, which can be used as an inferential controller for linear or nonlinear systems. The main contributions of the paper are: a) the proposed architecture and the adaptive mechanism based on online sliding window PCA; b) the incorporation of a proportional action and an integral anti-windup mechanism on the integral PCA controller. Some experimental results, obtained with a nonlinear system (three tank benchmark – Amira DTS200) are presented, showing the controller performance.**

I. Introduction

Modern industrial automation and distributed control systems make possible the collection of large quantities of process raw data. These data has to be processed to extract important information. Knowledge should be obtained from the significant information. Multivariate statistical methods (principal component analysis, factor analysis, discriminant analysis, etc) can be used to develop nonparametric models for process monitoring (including fault detection and diagnosis), ([1, 2, 3, 4]), for controller loop monitoring [22], as well as for feedback control in the scores space ([1, 2, 5, 6, 7].

There are a lot of approaches for controller tuning depending on the type of controller [18]. The focus here is on sliding window approaches for PI controller's tuning. Some references in this area are the following: a) a neural auto-tuner for PID controller [17], b) auto-tuning methods for PID controllers [15]; c) auto-tuning PID control with an on-line recursive identification algorithm (least squares support vector machines estimation algorithm) [19]; d) a control approach using a minimum variation controller and a linearized neural network model around its current operation region [20].

The proposed PCA control formulation, in a reduced control space, is analogous to modal control (also called eigenvalue-assignment control, ([8, 9]).

In [5], a controller based on static PCA, using an ARX model, for linear SISO systems, was proposed. An adaptive version using recursive PCA was proposed in [6].

In this paper, an adaptive controller based on sliding window PCA to deal with linear and nonlinear systems, in real-time applications, is proposed.

II. SVD / PCA Decomposition

Principal Component Analysis (PCA) is a multivariate statistical technique that can be implemented using singular value/vector decomposition (SVD) of the sample covariance matrix. Others statistical techniques include Principal Components Regression (PCR), Partial Least Squares (PLS or Projection to Latent Structures), etc, ([3, 4, 10, 11]).

PCA models can be used for data compression, process monitoring, fault detection and diagnosis and also for process control, as described in the next section.

PCA involves several steps. First, the original data $\mathbf{X}$o should be scaled, i.e., mean centered, and often normalized by the standard deviation. From the scaled data $\mathbf{X} \in \Re^{n \times m}$, the covariance matrix is computed by the relationship (1), where n is the number of observations (samples) and m is the number of process variables.

$$\mathbf{S} = \frac{1}{n\text{-}1}\,\mathbf{X}^{\mathbf{T}}\,\mathbf{X} \tag{1}$$

PCA captures the variability of the data $\mathbf{X}$. It determines loading vectors (orthogonal vectors) ordered by the amount of variance explained in the loading vector directions. The loading vectors are computed by solving the stationary points of an optimization problem ([3, 10, 11]), i.e., solving a singular value decomposition (SVD) of the sample covariance matrix $\mathbf{S}$:

$$\mathbf{S} = \frac{1}{n\text{-}1}\,\mathbf{X}^{\mathrm{T}}\,\mathbf{X} = \mathbf{V}\,\mathbf{\Lambda}\,\mathbf{V}^{\mathrm{T}} \tag{2}$$

The diagonal matrix $\mathbf{\Lambda} = \mathbf{\Sigma}^{\mathrm{T}}\,\mathbf{\Sigma}$, $\mathbf{\Lambda} \in \Re^{\mathrm{m \times m}}$, contains the non-negative real singular values of decreasing magnitude obeying the relation

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m \geq 0 \tag{3}$$

When the goal is to minimize the effect of random noise that corrupt the PCA representation, and to optimally capture the variations of data, then only the loading vectors corresponding to the "a" largest singular values must be

L. Brito Palma is with the Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, (corresponding author to provide phone: +351.212948339; fax: +351.212948532; e-mail: LBP@fct.unl.pt).

F. Vieira Coito is with the Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, (e-mail: FJVC@fct.unl.pt).

P. Sousa Gil is with the Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, (e-mail: PSG@fct.unl.pt).

retained in the PCA model; "a" is the number of principal components that captures the most important information, that can be computed according a certain explained variance (usually greater than 80 % or 90%).

PCA projects the observation space into two subspaces: the scores subspace and the residual subspace. Selecting the columns of the loading matrix $\mathbf{P} \in \Re^{m \times a}$ to correspond to the loading vectors $\mathbf{V} \in \Re^{m \times m}$ associated with the "a" largest singular values, the projections of the observation data $\mathbf{X} \in \Re^{n \times m}$ into the lower-dimensional space are contained in the scores matrix $\mathbf{T} \in \Re^{n \times a}$

$$\mathbf{T} = \mathbf{X}\,\mathbf{P} \tag{4}$$

The projection of $\mathbf{T}$ back into the m-dimensional observation space is given by

The residual matrix E is given by

The residual matrix $\mathbf{E}$ captures the variations in the observation space spanned by the loading vectors associated with the "m - a" smallest singular values. The subspaces eq \o(**X**;\s\up8(^)) and $\mathbf{E}$ are usually denominated scores space and residual space, respectively.

I. Process Control based on PCA Models

Many industrial plants (power plants, chemical processes, etc) have a large number of exogenous variables and a few manipulated ones. The exogenous variables indicate the process state, and the manipulated variables control directly the measured quantities and indirectly the unmeasured quantities. Some of these unmeasured quantities are related to the final product quality, so they should be carefully monitored and controlled.

In this work, the main idea is to develop a PCA model, using a sliding window, to represent the desired process region in the score space, and then design a controller in the score space that maintains operation within this region. The control moves in the score space are then mapped to the real variable space and implemented on the process. The process is kept within the desired region provided that the PCA model has correctly established the relationship between exogenous variables and manipulated variables.

A. *Design of Controller based on Static PCA Model*

The formulation presented here is based on the previous works [1, 2, 5]. Let $\mathbf{X}$ be composed of two types of variables (exogenous and manipulated)

$$\mathbf{X} = [\mathbf{X}_{ex} \,|\, \mathbf{X}_{mp}] \tag{7}$$

The development of a PCA model yields

$$\mathbf{X} = [\mathbf{X}_{ex} \,|\, \mathbf{X}_{mp}] = \mathbf{T}\,\mathbf{P}^{\mathrm{T}} + \mathbf{E} \tag{8}$$

The process output signal can be decomposed in

$$x_d = \mathbf{T}\,q^{\mathrm{T}} + f_v \tag{9}$$

where f_v is the output residual.

The equivalent controller set point in the scores space is obtained from (10), where Ψ is the pseudo-inverse.

$$t_{sp} = x_{d,sp}\,(q^{\mathrm{T}})^{\Psi} \tag{10}$$

The score vector, t, can be computed from the projection of x onto the matrix of eigenvectors $\mathbf{P}$, obeying

$$t = x\,\mathbf{P} \tag{11}$$

In the scores space the error between the desired set point and the online scores associated with x at discrete time k is given by

$$\Delta t = t_{sp} - t \tag{12}$$

The error in the scores space can be reconstructed as an error in the $\mathbf{X}$ space by

$$\Delta x = \Delta t\,\mathbf{P}^{\mathrm{T}} \tag{13}$$

The relationship between the exogenous and manipulated variables in the score space must be defined. Consider the partition of the matrix of eigenvectors $\mathbf{P}$ as

$$\mathbf{P}^{\mathrm{T}} = [\mathbf{P}_{ex} \,|\, \mathbf{P}_{mp}] \tag{14}$$

The relationship between exogenous and manipulated variables is given by (15), where $\mathbf{\Lambda}_{\mathrm{p}}$ is the matrix of coefficients that defines this relationship in the scores space.

$$\mathbf{P}_{mp} = \mathbf{P}_{ex}\,\mathbf{\Lambda}_{\mathrm{p}} \tag{15}$$

So, the matrix $\mathbf{\Lambda}_{\mathrm{p}}$ can be computed using the pseudoinverse Ψ

$$\mathbf{\Lambda}_{\mathrm{p}} = \mathbf{P}_{ex}^{\ \Psi}\,\mathbf{P}_{mp} \tag{16}$$

The architecture of the classical integral PCA controller used in this work is depicted in Fig. 1, [5]. The block $\mathbf{P}_{\mathrm{a}}$ represents the plant, $\mathbf{C}$ is the PCA controller, and $\mathbf{Q}$ is given by

$$\mathbf{Q} = (q^{\mathrm{T}})^{\Psi} \tag{17}$$

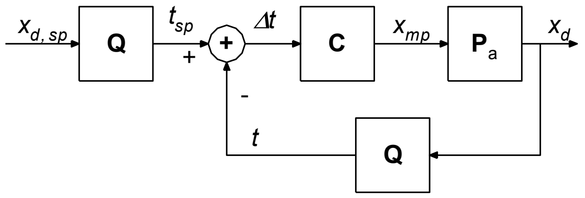

Fig. 1. Architecture of the classical integral PCA controller.

B. Implementation of the Classical Integral PCA controller

The PCA model, relating exogenous and manipulated variables, is a steady-state model that results in an integral PCA controller. The implementation presented will be detailed assuming, without loss of generality, a linear SISO process, modeled by an autoregressive ARX input-output model.

First, an ARX model structure should be selected by means of a system identification algorithm such as least-squares, principal components regression, etc [4]. In this work, a relay controller was used to capture the nominal data around a certain operating point. Let's assume a low order ARX model, ARX(na=2, nb=1, nd=1). The matrix $\mathbf{X} \in \Re^{n \times m}$, mean centered, will have lines of vectors (regressor for each discrete time k) given by the expression $x(k) = \eta(k) = [y(k)\ y(k\text{-}1)\ y(k\text{-}2)\ u(k\text{-}1)]$. In this work, n is the number of data samples and m is the number of columns in the regressor $\eta(k)$. The sampling time used was $T_S = 1$ s. The number of exogenous variables is $r = 3$, the number of manipulated variables is $p = m - r = 1$, and the number of principal components is $a = 2$.

The covariance matrix is computed from (1), and the SVD decomposition (2) gives the eigenvalues and the eigenvectors. The loading matrix $\mathbf{P} \in \Re^{m \times a}$ corresponds to the loading vectors $\mathbf{V} \in \Re^{m \times m}$ associated with the "a" largest singular values. The scores matrix $\mathbf{T} \in \Re^{n \times a}$ is computed based on (4). The process output is given by $x_d(k) = y(k)$ and q is given by

$$q = (\mathbf{T}^{\mathrm{T}}\, x_d)^{\mathrm{T}} \tag{18}$$

The decomposition of transposed $\mathbf{P}$ matrix permits to obtain the others matrices (14). The matrix $\mathbf{\Lambda}_{\mathrm{p}}$ is computed using (16).

The previous computations were obtained offline. In online operation the next computations should be performed for each discrete time k. First, assign the reference signal to $x_{d,sp}(k)$. Next, compute the set point in the scores space using (10). Compute the scores in the reduced 2D space from

$$t(k) = x_d(k)\, (q^{\mathrm{T}})^{\Psi} \tag{19}$$

The control error in the scores space is given by (12). The control error in the X space is given by (13).

Finally, the increment in the manipulated variable can be computed from

$$\Delta x_m(k) = \Delta t(k)\, \mathbf{P}_{mp} \tag{20}$$

In the incremental form, assuming $u(k) = x_{mp}(k)$, the control action is given by

$$x_{mp}(k) = x_{mp}(k\text{-}1) + K_c\, \Delta x_m(k) \tag{21}$$

The controller gain, K_c, was incorporated in [5], in order to allow tuning the closed-loop dynamics. This control structure is an integral controller. In order to improve the controller performance, an anti-windup mechanism and a proportional action should be incorporated in the PCA controller; this will be done in section III.D.

C. Design of Adaptive PCA Controller based on Recursive PCA

It is possible to implement adaptive PCA controllers using a recursive PCA algorithm, as described in [6]. The main idea is to compute online, recursively, the eigenvector matrix $\mathbf{P}$(k) and the eigenvalue matrix $\mathbf{\Lambda}(k)$. The eigenvector matrix is computed accordingly (22), were $\mathbf{I}$ is an identity matrix, $\mathbf{P}_{\mathrm{v}}$ is a perturbation matrix and $\mathbf{N}_{\mathrm{p}}(k)$ is a diagonal normalization matrix, [6, 12, 13].

$$\mathbf{P}(k) = \mathbf{P}(k\text{-}1)\, (\mathbf{I} + \mathbf{P}_{\mathrm{v}})\, \mathbf{N}_{\mathrm{p}}(k) \tag{22}$$

D. Design of PI Controller based on Sliding Window PCA

The main contribution of the paper is presented next. A proportional-integral (PI) controller using sliding-window PCA will be detailed. The proposed architecture of the adaptive PI-PCA controller is depicted in Fig. 2.

An adaptive controller needs supervisory functions in order to function well in an industrial environment [14]. Here, the supervisor should adjust the adaptive mechanism of the sliding window PCA algorithm that implements the integral component of the PI controller.

The main advantage of this approach is the fact that the integral component of the PI controller is based on a PCA model build on-line using input-output correlated data acquired from the process. The persistent excitation conditions should be verified on-line. At the startup a fixed PCA model should be used, build off-line using manual control or any other stabilizing controller (relay, PID, etc).

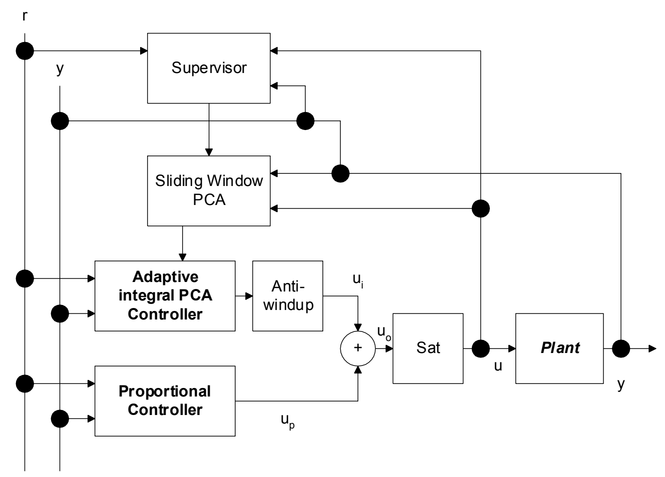

Fig. 2. Architecture of the adaptive sliding window PI-PCA controller.

The PI-PCA controller equations are presented next.

The control action is the sum of two control actions: one due to the proportional controller and another due to the adaptive integral PCA controller. The control action is saturated by the block "Sat" in order to guarantee a range between "0" and "1".

Here, the adaptive mechanism based on sliding window PCA algorithm is used to compute the adaptive integral action.

First, the sliding window data $\mathbf{X}_k$=$\mathbf{X}(:,:,k) \in \Re^{n \times m}$ for "n" samples should be obtained from input-output process data:

$$X(:,:,k) = \begin{bmatrix} y(k-n+1) & \dots & & u(k-n) \\ & & \dots & \\ y(k) & y(k-1) & y(k-2) & u(k-1) \end{bmatrix} \quad (23)$$

Then the SVD decomposition is applied to the covariance matrix and finally the on-line $\mathbf{P}_k$ matrix is obtained:

$$\begin{aligned} \mathbf{X}_k &\rightarrow \mathbf{S}_k \\ \mathbf{V}_k &= \mathrm{SVD}(\mathbf{S}_k) \\ \mathbf{P}_k &= \mathbf{V}_k(:,1:\mathrm{a}) \end{aligned} \quad (24)$$

The manipulated part of the $\mathbf{P}_k$ matrix, $\mathbf{P}_{mp,k}$, is obtained from

$$\mathbf{P}^{\mathrm{T}}_{k} = [\mathbf{P}_{ex,k} \mid \mathbf{P}_{mp,k}] \quad (25)$$

Finally, the incremental action, $\Delta u_{sw}(k)$, computed based on the sliding window PCA, is given by

$$\Delta u_{sw}(k) = \Delta t(k)\, \mathbf{P}_{mp,k} \quad (26)$$

The adaptive integral control action, with anti-windup ("aw") mechanism, is given by (27), inspired on the classical integral action (21):

$$u_i(k) = u_i(k\text{-}1) + K_i\, \Delta u_{sw}(k) + \mathrm{K}_{aw}\, (u(k) - u_0(k)) \quad (27)$$

The proportional control action is expressed by the conventional law:

$$u_p(k) = \mathrm{K}_p\, e(k) = \mathrm{K}_p\, (r(k) - y(k)) \quad (28)$$

II. Training and Performance Evaluation of the Sliding Window PCA Controller

In order to evaluate the performance of the PI-PCA controller, a comparison with others controllers will be presented in the next section, using the mean-squared control error (MSE) criterion.

A. Training Data

Input-output open-loop or closed-loop data is necessary to train the classical integral PCA controller, [7]. In this work, closed-loop data captured using a relay controller was used to build the PCA model, assuming a sampling time equal to 1 second, as depicted in Fig. 3.

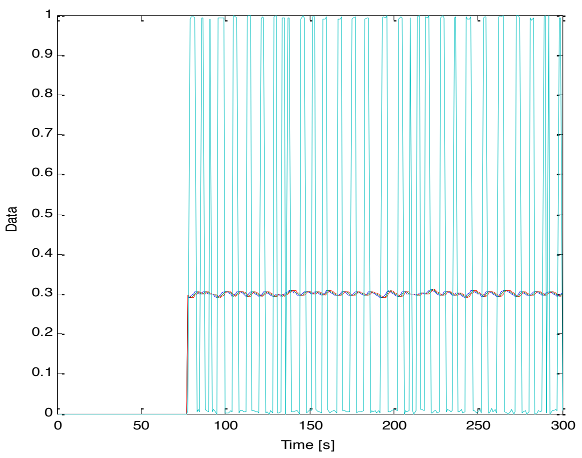

Fig. 3. Input-output training data for integral PCA controller.

B. Closed-Loop Behavior

The incorporation of the anti-windup mechanism and the proportional action on the classical integral PCA controller contribute to improve the closed-loop behavior, as shown on some experiments.

In the next section, a table showing the mean square control error (MSE) for a set of controllers is presented.

III. Experimental Results

Experimental results obtained with the DTS200 benchmark are presented here. A small dither (10^{-5}) was added to the reference signal in order to try to guarantee persistent excitation conditions.

A. DTS200 benchmark

The DTS200 benchmark system (Fig. 4, Fig. 5), manufactured by the Amira® Company, has been used, all over the world, to test nominal and fault tolerant controllers. Here the goal is to control the level in tank T1. The nonlinear models of this setup can be found in [21]. The configuration used in the experiments is the following: a) control action "u" – pump 1 control signal; b) sensor output "y" – tank 1 level; c) opened valves (V13, V32, V2out); d) leakage valves closed (V1L, V3L, V2L).

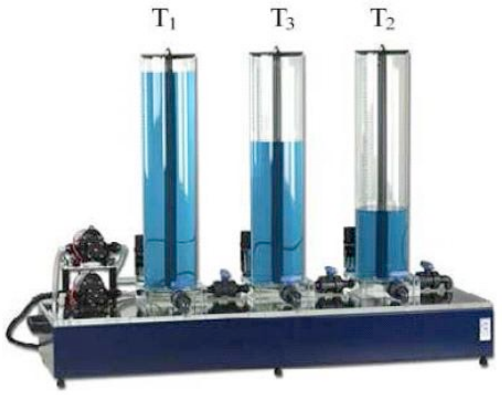

Fig. 4. The DTS200 benchmark setup (picture), [21].

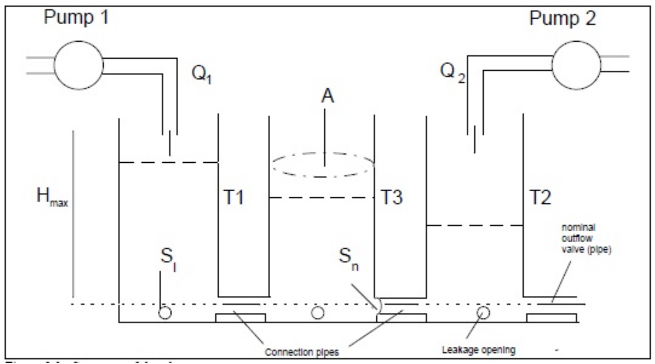

Fig. 5. The DTS200 benchmark setup (schematic diagram), [21].

B. Experimental Results

Experimental results are shown for a set of discrete time controllers, implemented on the Matlab environment using a National Instruments USB-6009 data acquisition board. The sampling time is 1 second. Figure 6 depicts the experimental results obtained with the classical PID controller, with anti-windup mechanism and the following controller parameters, obtained using the auto-tuning relay feedback method [15]: $K_p = 13.42$, $T_i = 6.43$ s, and $T_d = 0.96$ s. Figure 6 depicts the reference signal "r" (red), the control action "u" (green) and the sensor output "y" (blue). These labels are the same for all figures.

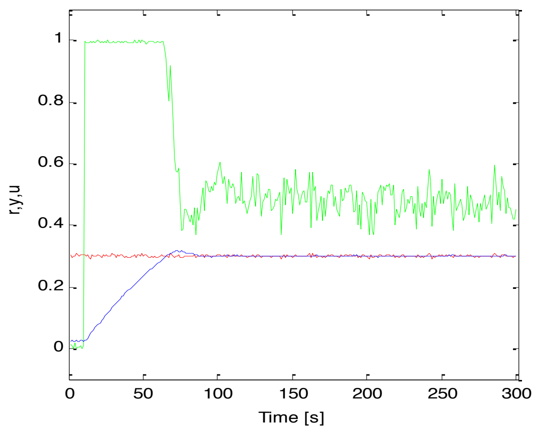

Fig. 6. Experiment with the classical PID controller for a fixed set-point (r = 0.3).

An adaptive LQG polynomial controller ([16]) was also tested on the benchmark system, as shown in Fig. 7. The controller gain is equal to 0.1, and an adaptive ARX(2,1,1) model was used for on-line system identification.

The results obtained with the proposed adaptive PI-PCA controller are depicted in Figure 8. The controller gains (design parameters) used for the experiments are the following: $K_p = 10$, $K_i = 33$, $K_{aw} = 0.1$. The initial value of the sliding window length ("n_0") was fixed on 60 seconds. The proportional action is depicted in a dashed line, and the integral adaptive action appears as a solid line. The choice of the controller gains obeys the following criteria. The proportional gain K_p was chosen to be equal to the classical PID controller. The integral gain K_i and the anti-windup gain were chosen to guarantee good performance in terms of convergence. The length of the sliding window ("n") should contain enough samples to guarantee a good PCA model obtained from the SVD decomposition of the data covariance matrix; here "n" obeys the equation (29), where "e" is the control error (in the normalized range $[0;1]$).

$$n = n_0(1+|e|), n_0 = 60, e = r - y. \tag{29}$$

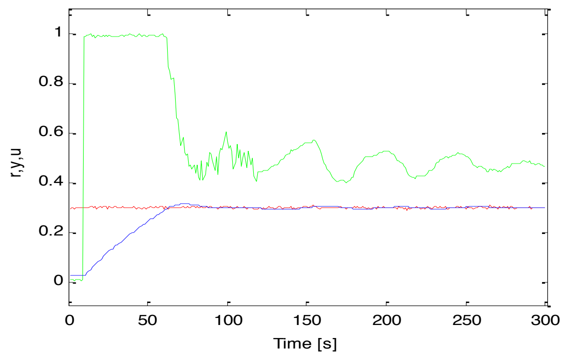

Fig. 7. Experiment with the adaptive LQG controller for a fixed set-point (r = 0.3).

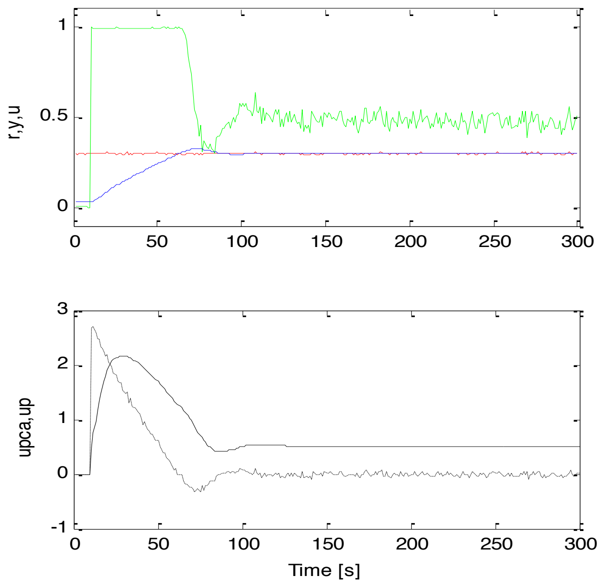

Fig. 8. Experiment with the adaptive PI-PCA controller for a fixed set-point (r = 0.3).

In Table I a comparison of controller's performance is presented, showing the mean squared control error (MSCE). The proposed PI-PCA controller reveals a good performance, for fixed (r = SP = 0.3) and variable set-points. The PCA controller is the classical integral PCA controller (21), without proportional action and without anti-windup mechanism.

TABLE I
COMPARISON OF CONTROLLERS PERFORMANCE (MSCE).

MSE (10^{-3})	SP: 0.3	SP: 0.3 → 0.4
PID controller	6.82	6.85
LQG controller	6.36	6.82
PCA controller	6.54	6.91
PI-PCA controller	6.15	6.80

IV. CONCLUSIONS

An adaptive PI controller based on sliding window principal component analysis (SW-PCA) was proposed in this paper, showing a performance similar to other kinds of controllers, such as the classical PID controller and the adaptive LQG controller. The main advantage of this approach is the fact that the integral component of the PI controller is based on a PCA model build on-line, using input-output correlated data acquired from the process. The persistent excitation conditions should be verified on-line.

In order to improve the performance of the PI-PCA controller, an optimization procedure that adjust on-line the sliding window length and guarantee the closed-loop stability should be implemented.

The extension of this kind of controller to act as a fault tolerant controller will be also investigated.

ACKNOWLEDGMENT

This work has been supported by Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa and by national funds through FCT - Fundação para a Ciência e a Tecnologia (project PEst-OE/EEI/UI0066/2011).

REFERENCES

[1] M. Piovoso, K. Kosanovich, "Applications of Multivariate Statistical Methods to Process Monitoring and Controller Design", in *Int. Journal of Control*, Vol. 59, No. 3, 1994, pp. 743-765.

[2] M. Piovoso, "The Use of Multivariate Statistics in Process Control", in *The Control Handbook* (edited by W. Levine), 1996, pp. 561-573, CRC Press & IEEE Press.

[3] J. Jackson, *A User's Guide to Principal Components*, Wiley, 2003.

[4] L. Brito Palma, *Fault Detection, Diagnosis and Fault Tolerance Approaches in Dynamic Systems based on Black-Box Models*, Phd Thesis, Universidade Nova de Lisboa, Portugal, 2007.

[5] L. Brito Palma, F. Vieira Coito, P. Sousa Gil, R. Neves-Silva, "Process Control based on PCA Models", *15th IEEE Int. Conf. on Emerging Technologies and Factory Automation*, Setp. 13-16, Univ. of the Basque Country, Bilbao – Spain, 2010.

[6] L. Brito Palma, F. Vieira Coito, P. Sousa Gil, R. Neves-Silva, "Design of Adaptive PCA Controllers for SISO Systems", *18th World Congress of the International Federation of Automatic Control (IFAC)*, Aug.28-Sept.02, Milano – Italy, 2011.

[7] L. Brito Palma, F. Vieira Coito, "Tuning PCA Controllers based on Manual Control Data", *IEEE Emerging Technologies and Factory Automation (ETFA)*, Sept. 5-9, Toulouse – France, 2011.

[8] G. Feng, R. Lozano, *Adaptive Control Systems*, Newnes, 1999.

[9] J. Magni, *Robust Modal Control with a Toolbox for Use with Matlab*, Kluwer, 2002.

[10] L. Chiang, E. Russell, R. Braatz, *Fault Detection and Diagnosis in Industrial Systems*, Springer-Verlag, 2001.

[11] I. Jolliffe, *Principal Component Analysis*, Springer, 2002.

[12] H. Peddaneni, D. Erdogmus, Y. Rao, A. Hedge, J. Principe, "Recursive Principal Components Analysis using Eigenvector Matrix Perturbation", *Proc. Of IEEE Workshop on Machine Learning for Signal Processing*, São Luís – Maranhão, Brazil, 2004.

[13] J. Yao, X. Liu, X. Zhu, "Reduced Dimension Control Based on Online Recursive Principal Component Analysis", *Proc. of American Control Conference*, St. Louis, USA, 2009.

[14] T. Hagglund, K. Astrom, "Supervision of Adaptive Control Algorithms", in *Automatica*, Vol. 36, 2000, pp. 1171-1180.

[15] K. Astrom, T. Hagglund, *Automatic Tuning of PID Controllers*, Instrument Society of America, 1988.

[16] F. Lewis, "Optimal Control", in *The Control Handbook*, editor W. Levine, CRC Press, 2000.

[17] R. Swiniarski, "Neural Network Based Self-Tuning PID Controller with Fourier Transformation of Temporal Patterns", *16th Conf. of IEEE Industrial Electronics Society*, 1990, pp. 1227-1232.

[18] W. Levine, *The Control Handbook*, editor W. Levine, CRC Press, 2000.

[19] J. Zhao, P. Li, X. Wang, "Intelligent PID Controller Design with Adaptive Criterion Adjustment via Least Squares Support Vector Machine", *Proc. of Combined 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, Shanghai, China, 2009.

[20] J. Chen, T. Huang, "Applying Neural Networks to On-Line Updated PID Controllers for Nonlinear Process Control", *Journal of Process Control*, no. 14, 2004, pp. 211–230.

[21] Gurski-Schramm, DTS200 Manual - Laboratory Setup Three-Tank System, Amira, Duisburg - Germany, 2009.

[22] C. Viveiros, L. Brito Palma, J. Manuel Igreja, "Fault Tolerant Switching Control Based on Adaptive LQG and Fuzzy Controllers", *Intelligent Systems, Control and Automation: Science and Engineering Book series*, Springer, ISBN: 978-94-007-4721-0, 2012.