PI Controller for SISO Linear Systems based on Neural Linear PCA

L. Brito Palma, F. Vieira Coito
Universidade Nova de Lisboa, FCT - Dep. Eng. Electrotécnica, CTS - Uninova Research Institute
Portugal, Monte de Caparica, 2829-516
Email: [LBP,FJVC]@fct.unl.pt

P. Sousa Gil
Universidade Nova de Lisboa, FCT - Dep. Eng. Electrotécnica, CISUC Research Institute
Portugal, Monte de Caparica, 2829-516
Email: PSG@fct.unl.pt

***Abstract*—In this paper an approach to design proportional-integral (PI) controllers, for SISO systems, based on neural linear principal components analysis (PCA) is presented. Closed-loop control can be formulated and implemented within the reduced space defined by a PCA model. The neural linear PCA controller, results in an integral controller, which can be used as an inferential controller. The main contributions of the paper are: a) the proposed architecture with a classical proportional controller and a neural integral controller based on linear neural PCA; b) the evaluation of the controller performance using the Harris index. Some experimental results obtained with a DC motor linear model are presented, showing the controller performance.**

I. Introduction

The PID controller is the most dominating form of feedback system in use today in industries. Research on PID control will have certainly impact on the future, [1], [2], [3]. Most loops use in fact only PI control because derivative action is not used very often. The PID controller is used for a wide range of control problems: process control, motor drives, automotive, flight control, etc. The PID controllers appears in different forms: as standard single-loop controllers, as a software module in programmable logic controllers (PLC's) and also as distributed multi-loop control systems.

Principal component analysis (PCA) and other multivariate statistical methods (factor analysis, discriminant analysis, etc) can be used to develop nonparametric models for process monitoring (including fault detection and diagnosis), [4], [5], [6], [7]. PCA can also be used for controller loop monitoring, [8], as well as for feedback control in the scores space, [4], [5], [9], [10], [11].

The proposed PCA control formulation, in a reduced control space, is analogous to modal control, also called eigenvalue-assignment control. The great advantage of the control structure proposed here is the fact that only input-output process data is needed for controller tuning, so this is a data-based controller tuning approach.

In this paper, a PI controller based on linear neural PCA, to deal with linear SISO systems in real-time applications, is proposed.

II. Control based on PCA Models

Principal components analysis (PCA) can be used to design integral controllers, [4], [5], [9], [10], [11].
The Matlab notation is used here to define sub-matrices.
The architecture of the classical integral PCA controller is depicted in Fig. 1. Detailed information about the design of this kind of controllers can be found in [9], [10], [11]. The block **"Pa"** represents the plant, **"C"** is the PCA controller, and $\boldsymbol{Q}$ is given by (1). The algorithms were implemented in discrete-time. Let's assume that k is the k sample in the algorithms.

$$\boldsymbol{Q} = (\boldsymbol{U}^T)^+ \tag{1}$$

In (1), $+$ represents the pseudo-inverse of the matrix and T is the transposed. The matrix $\boldsymbol{U}$ is obtained from (2), where $x_d(k) = y(k)$ is the process output.

$$\boldsymbol{U} = (\boldsymbol{\Upsilon}^T\ x_d(k))^T \tag{2}$$

Assuming two principal components, $a = 2$, $\boldsymbol{\Upsilon}$ is the projection of the regression matrix $\boldsymbol{X}$ (input/output data) on the 2D PCA scores space, accordingly (3).

$$\boldsymbol{\Upsilon} = \boldsymbol{X}\ \boldsymbol{P} \tag{3}$$

Assuming that the training data matrix is $\boldsymbol{X} \in \Re^{n\times m}$, where n is the number of rows and m is the number of columns, the matrix $\boldsymbol{P}$ corresponds to the first a columns of the loadings (singular vectors) matrix $\boldsymbol{V}$, i.e. $\boldsymbol{V} = \boldsymbol{P}(:, 1:a)$,

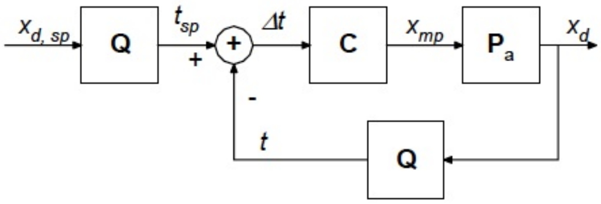

Fig. 1. Architecture of the classical integral PCA controller.

for a SVD decomposition as follows (4), assuming that $\boldsymbol{\Lambda}$ is the singular values matrix.

$$cov(\boldsymbol{X}) = \frac{1}{n-1}\, \boldsymbol{X}^T\, \boldsymbol{X} = \boldsymbol{V}\, \boldsymbol{\Lambda}\, \boldsymbol{V}^T \tag{4}$$

Assuming an ARX(2,2,1) model for the plant/process under study, the projection of the on-line regressor data vector $\boldsymbol{x}(k) = [y(k)\ y(k-1)\ y(k-2)\ u(k-1)\ u(k-2)]$ on the 2D scores space, is given by (5).

$$\boldsymbol{t}(k) = \boldsymbol{x}(k)\, \boldsymbol{P} \tag{5}$$

The classical integral PCA controller was initially formulated by Piovoso, [4], [5]. Some developments using ARX models were proposed later, for static PCA models and also for adaptive PCA models, [9], [10], [11]. In the incremental form, assuming $u(k) = x_{mp}(k)$, the control action of the classical integral controller based on static PCA is given by (6).

$$x_{mp}(k) = x_{mp}(k-1) + K_i\, \Delta x_m(k) \tag{6}$$

The incremental action depends on the manipulated matrix $\boldsymbol{P}_{mp}$ as described in (7).

$$\Delta x_m(k) = \Delta t(k)\, \boldsymbol{P}_{mp} \tag{7}$$

The $\boldsymbol{P}$ matrix is decomposed into two matrices, $\boldsymbol{P} = [\boldsymbol{P}ex|\boldsymbol{P}mp]$: exogenous matrix ("Pex", first r columns) and manipulated matrix ("Pmp", the last m-r columns). The variables m, r and m-r are, respectively, the length of $\boldsymbol{x}(k)$, the number of output process variables and input process variables in the regression vector $\boldsymbol{x}(k) = [y(k)\ y(k-1)\ y(k-2)\ u(k-1)\ u(k-2)]$; for an ARX(2,2,1) model the variables assume the values m = 5, r = 3 and m-r = 2.

III. Proposed Controller Structure based on Linear Neural PCA

The integral controller proposed here is based on neural linear PCA that can be implemented accordingly to the architecture depicted in Fig. 2, [12], [13]. Function G implements data compression and function H implements data decompression.

The main idea and contribution in the paper is the

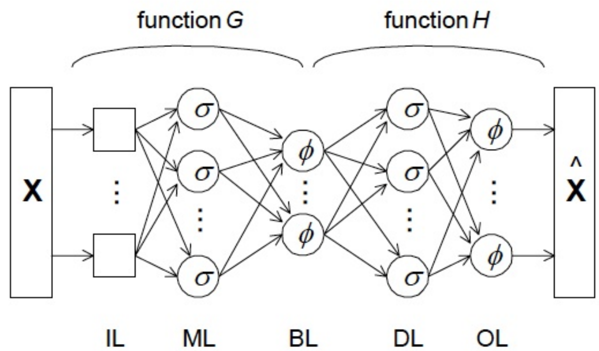

Fig. 2. Architecture of auto-associative neural network for PCA.

implementation of the equation $\boldsymbol{t}(k) = \boldsymbol{x}(k)\ \boldsymbol{P}$ using a neural structure. In fact, since to implement PCA on neural structures an auto-associative architecture is needed, the projection on the 2D scores space is obtained at the output of the bottleneck layer (BL), assuming two principal components, $a = 2$. In this work, only linear activation functions without bias were used, in order to guarantee a linear neural network, so both σ and ϕ are linear activation functions. As observed in Fig. 2, an auto-associative neural network implements a data compression at the bottleneck layer and a data decompression at the output layer.

First of all, the algorithm needs input / output training data to build a neural linear PCA model. Let's assume that the training data is archived in the matrix $\boldsymbol{X} \in \Re^{n \times m}$. In this work, to build the data matrix, the regressor vector was inspired on the regressor of an ARX(2,2,1) model, as described in (8), without loss of generality. In this work, the simulations were done assuming $n = 3636, m = 5$, for a sampling period of $T_s = 0.11[s]$. .

$$\boldsymbol{x}(k,:) = [y(k)\ y(k-1)\ y(k-2)\ u(k-1)\ u(k-2)] \tag{8}$$

The training data should not contain very high frequency data since this can cause problems in the training algorithm of the neural network that was used for the PCA model. The Levenberg-Marquardt optimization algorithm was used for the training of the neural network, [14].

The integral controller based on linear neural PCA will be presented using a matrix formulation, inspired on the linear integral PCA controller described in section II.
In the algorithm $\boldsymbol{M}$ is the neural mapping matrix (first layer matrix ML, with weight matrix $\boldsymbol{W}_1$) and $\boldsymbol{B}$ is the neural bottleneck matrix (second layer matrix BL, with weight matrix $\boldsymbol{W}_2$), i.e., $\boldsymbol{M} = \boldsymbol{W}_1$ and $\boldsymbol{B} = \boldsymbol{W}_2$.

The $\boldsymbol{P}_n$ matrix, equivalent to the main singular vectors matrix, is obtained from the neural network weight matrices as described in (9).

$$\boldsymbol{P}_n = \boldsymbol{B}\ \boldsymbol{M} \tag{9}$$

The $\boldsymbol{P}_n$ matrix is decomposed into two matrices, $\boldsymbol{P}_n = [\boldsymbol{P}_{ex}|\boldsymbol{P}_{mp}]$: exogenous matrix (P_{ex}, first r columns) and manipulated matrix (P_{mp}, the last columns).

$$\boldsymbol{P}_{ex} = \boldsymbol{P}_n(:, 1:r) \tag{10}$$

$$\boldsymbol{P}_{mp} = \boldsymbol{P}_n(:, r+1:m) \tag{11}$$

The matrix $\boldsymbol{M}_\lambda$ is obtained from the $\boldsymbol{P}_n$ matrix, where + is the pseudo-inverse of the matrix.

$$\boldsymbol{M}_\lambda = (\boldsymbol{P}_{ex})^+ \boldsymbol{P}_{mp} \tag{12}$$

The error vector is given by

$$\Delta \boldsymbol{x}(k,:) = [e_c(k)\ e_c(k-1)\ e_c(k-2)] \tag{13}$$

The increment in the manipulated variable is expressed by

$$\Delta x_{mp}(k) = \Delta x(k,:)\ \boldsymbol{M}_\lambda(:,1) \tag{14}$$

The non-saturated control action is formulated as

$$u_{c0}(k) = u_c(k-1) + K_i\ \Delta x_{mp}(k) + a_w(k-1) \tag{15}$$

The saturated control action is given by

$$u_c(k) = satur(u_{c0}(k), ...) \tag{16}$$

The anti-windup term is expressed by

$$a_w(k) = K_{aw}\ [u_c(k) - u_{c0}(k)] \tag{17}$$

A proportional term was added to the integral controller in order to improve the overall controller performance. The new non-saturated control action is given by (18).

$$u_{c0}(k) = K_p\ e_c(k) + u_c(k-1) + K_i\ \Delta x_{mp}(k) + a_w(k-1) \tag{18}$$

IV. Control Loop Performance Monitoring based on Harris Index

In a typical process industry there are hundreds of controllers. Even if these controllers initially perform well, various factors can contribute to their performance deterioration, including [15]: sensor/actuator or process faults, sticking valves producing oscillations, to name just a few. Around 50% of all industrial controllers, or more, have some kind of performance problem, [16], [17].

Most of the reported industrial applications of control loop performance monitoring are based on the pioneer method suggested by Harris, [18], or some algorithms inspired on Harris approach, such as the approaches found in the references [19], [20], to name just a few. The great popularity of the methods based on the Harris performance index is due to both the conceptual and computational simplicity, as also the small amount of information required. Performance indices can be used for control loop performance monitoring or for control structure selection, [21].

The first indices for control loop performance assessment were proposed by Harris [18], Desborough and Harris [22] and Stanfelj et al [23]. A review of the status in control loop performance assessment (CLPA) technology and industrial applications was published in 2006, by Jelali [24]. Merits and drawbacks of each CLPA method are highlighted. The application of fuzzy logic and neural networks were also investigated in the CLPA field, [25].

Next is described, summarily, the modified Harris index used in this work, [8]. The key variable for CLPA is the control error, $e(k) = r(k) - y(k)$. The control error should have no predictable component. Let's compute a residual $\delta(k)$ between the measured control error $e(k)$ and a forward prediction $\hat{e}(k)$, described by

$$\delta(k) = e(k) - \hat{e}(k) \tag{19}$$

In a control loop that is performing well it is expected that the control error contains only random noise. The normalized CLPA index used here $\vartheta(k)$, computed on-line for a sliding window (time horizon), is described by (20), where var(.) is the variance and mse(.) is the mean-squared error.

$$\vartheta(k) = 1 - \frac{var(\delta(k))}{mse(e(k))} \tag{20}$$

For typical data from process control loops an autoregression AR(*n*,*b*) time series model that makes predictions *b* steps ahead is suitable for modeling the forward prediction $\hat{e}(k)$.

A good performing loop has a value of the CLPA index $\vartheta(k)$ close to 0, while for a poorly performing loop is close to 1.

V. Experimental Results

In order to evaluate the performance of the proposed control structure, different process models were tested, namely first and second order models. Here, simulation results obtained with a DC motor model are presented. The algorithms were implemented in Matlab in discrete-time, k.

The DC motor model was simulated in continuous-time and sampled at each sampling period, $T_s = 0.11[s]$, using the ode45 function based on the Runge-Kutta method. In armature controlled DC motors, the applied voltage $u_a(t)$ controls the angular velocity $\omega_r(t)$ of the shaft. A simplified continuous-time transfer function of the DC motor can be given by (21) and (22), a second order system, [26]. K_m is the torque constant, L is the armature inductance, R is the armature resistance, J is the rotor inertia, K_f is

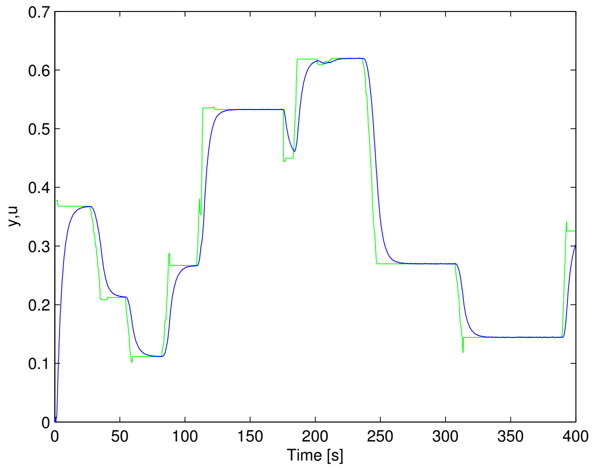

Fig. 3. Training data for neural linear PCA using manual control with the mouse.

the viscous-friction coefficient, and K_b is the back-emf constant. The values, in S.I. units, used for the simulations are: $K_m = K_b = 0.1[Nm/A]$, $L = 0.5[H]$, $R = 2$ $[\Omega]$, $J = 0.8$ $[Kgm^2/s^2]$, and $K_f = 0.2[Nms]$.

$$G_m(s) = \frac{\omega_r(s)}{u_a(s)} \tag{21}$$

$$G_m(s) = \frac{K_m}{LJ\ s^2 + (LK_f + RJ)\ s + RK_f + K_m K_b} \tag{22}$$

Fig. 3 contains the input/output data used to train the auto-associative neural network that implements the integral linear neural PCA controller. The control input data was generated by a human operator using a mouse, in closed-loop manual control.

In Fig. 4 can be observed an experiment done with the PI-PCA neural linear controller proposed in this work. From top to bottom can be observed the following signals: reference (r) and process output (y), control error (e), control action (u), proportional action (up) and integral action (ui), and finally the Harris index (ha).

The controller gains are the following: $K_p = 1, K_i = 0.2\ \eta$, with $\eta = mean(abs(t1))$, and $K_{aw} = 1$. The value $t1$ is the projection of the data $x(k)$ along the first singular vector, i.e., the first principal component, accordingly (5).

A small dither signal with variance 10^{-8} was added to the output, and also to the reference signal in order to guarantee persistent excitation conditions.

A comparison was done between the neural PI-PCA controller and a classical PI controller in incremental form, with gains $Kp = 1.69, Ti = 6.42$, [27]. The Harris index (control loop performance index) was used for measure the control loop performance in the experiments done, [8].

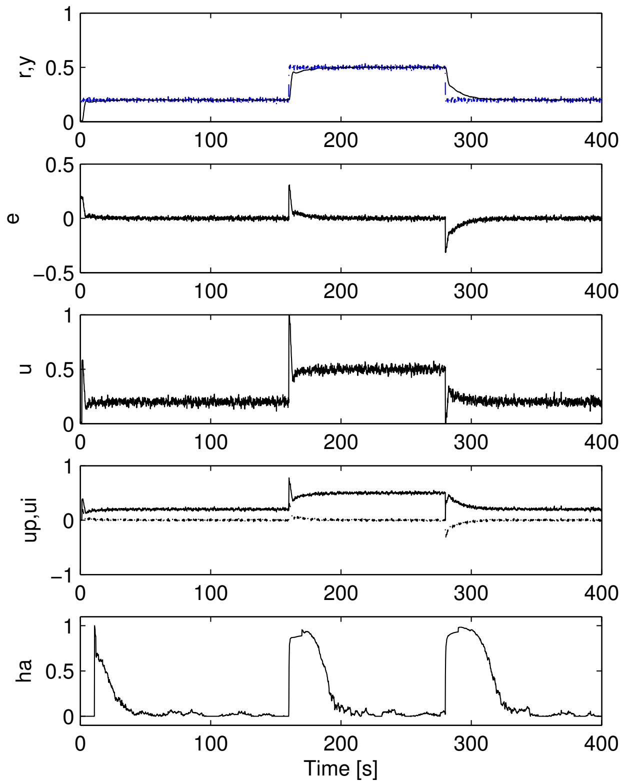

Fig. 4. PI-PCA linear neural controller.

The mean value of the Harris index, ϑ, was computed for two experiments, for all the experiment data, and the obtained values are: a) neural PI-PCA controller, $\vartheta = 0.190$; b) classical PI controller, $\vartheta = 0.157$. It is clear that the performance is similar. It should be enhanced that a better performance is obtained if the Harris index is lower.

In Fig. 5 can be observed an experiment done with the PI-PCA neural linear controller in a faulty situation, assuming that the sensor output tends abruptly to zero. It can be observed that the controller reacts well. For this situation the Harris index is $\vartheta = 0.265$ for the neural PI-PCA controller and is $\vartheta = 0.187$ for the classical PI controller.

As observed in Fig. 4 and Fig. 5, the Harris index clearly detects low performance situations, namely transient behaviours and faulty situations.

In Fig. 6 can be observed an experiment with a high integral gain, $K_p = 1, K_i = 1.0\ \eta$, i.e., 5 times greater than the other simulation (see Fig. 4). Accordingly to the Harris performance index, the control loop performance is not so good, $\vartheta = 0.255$, since ϑ it is higher. The control signal

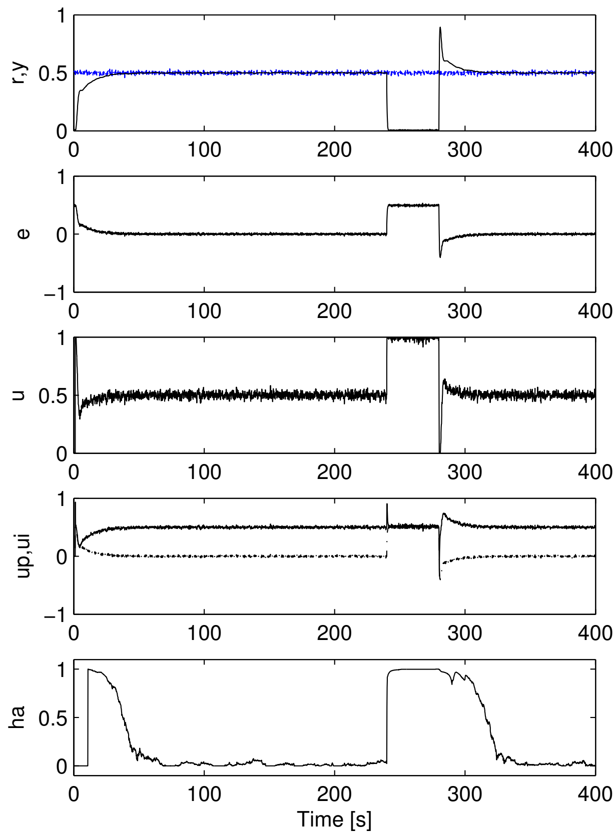

Fig. 5. Fault on sensor, y(k) = 0.

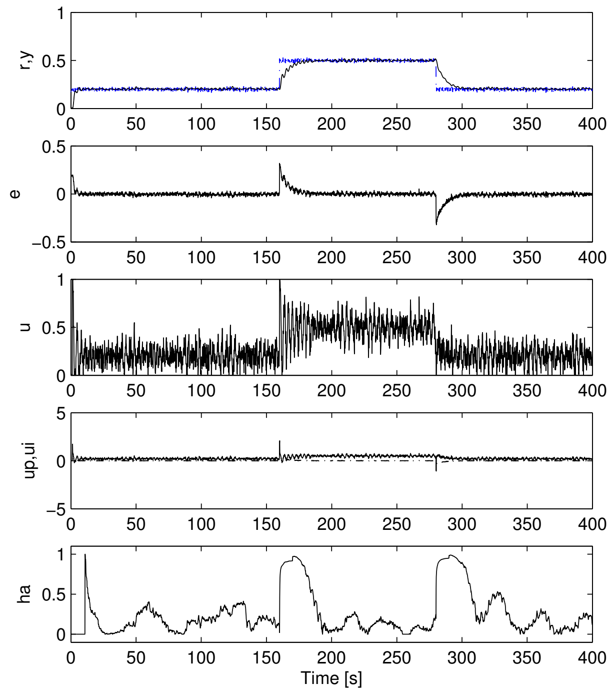

Fig. 6. Experiment with high controller gain Ki.

reveals a high variance and some oscillations appear on the output signal.

When evaluating PID controllers it is of interest to have a set of test examples so that different control schemes can be evaluated. Over the years, Astrom and Hagglund have collected a large number of test examples which has been used for research and also for evaluation of commercial PID controllers, [28]. In a near future, the proposed neural PI-PCA control scheme will be evaluated with this set of test examples, containing 11 transfer functions such as: multiple equal poles, fourth order system, right half plane zero, time delay and lag, time delay and double lag, heat conduction, fast and slow modes, conditionally stable system, oscillatory system, unstable pole and systems with integral action. Classical PID control is not well suited for all of them.

VI. Conclusions

In this paper, a new approach for design integral controllers based on linear neural PCA in the reduced 2D scores space, for SISO systems, was proposed.

In order to improve the overall control loop performance, a proportional control action was incorporated in the control structure.

The neural network is used to capture the integral controller behaviour. Since neural networks are very sensitive to data, high frequency data should be avoided in the training phase of the PCA neural network. Typically overdamped behaviours are obtained with this kind of controllers, due to the fact that the integral part of the controller is implemented with a neural network and the training data usually do not contain high frequency data.

The obtained results are very promising. More tests should be done to evaluate this approach with other classes of systems. Some future research pointers are: a) develop an approach to automatically adjust the controller gains, using optimization techniques; b) extend the approach for nonlinear controllers using a multi-models approach; c) more test examples should be evaluated in order to discover the classes of systems that are apropriate for this kind of controllers.

Acknowledgment

This work has been supported by Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, by Uninova-CTS, by CISUC-Coimbra and by national funds through FCT - Fundação para a Ciência e a Tecnologia (project PEst-OE/EEI/UI0066/2011). The authors would like to thank all the institutions.

References

[1] R. Vilanova and A. Visioli, Eds., *PID Control in the Third Millennium - Lessons Learned and New Approaches*. Springer, 2012.

[2] K. Ang, G. Chong, and Y. Li, "Pid control system analysis, design, and technology," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 4, pp. 559–576, 2005.

[3] K. Astrom and T. Hagglund, "The future of pid control," *Control Engineering Practice*, vol. 9, pp. 1163 – 1175, 2001.

[4] M. Piovoso and K. Kosanovich, "Applications of multivariate statistical methods to process monitoring and controller design," *Int. Journal of Control*, vol. 59, no. 3, pp. 743 – 765, 1994.

[5] M. Piovoso, *The Control Handbook*. CRC Press, 1996, ch. The Use of Multivariate Statistics in Process Control, pp. 561 – 573.

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

[7] L. Brito Palma, "Fault detection, diagnosis and fault tolerance approaches in dynamic systems based on black-box models," Ph.D. dissertation, Universidade Nova de Lisboa, Portugal, 2007.

[8] L. Brito Palma, J. Moreira, P. Gil, and F. Coito, "Hybrid approach for control loop performance assessment," in *5th International Conference on Intelligent Decision Technologies (KES-IDT)*, Sesimbra - Portugal, June 26-28 2013.

[9] L. Brito Palma, F. Coito, P. Gil, and R. Neves-Silva, "Process control based on pca models," in *15th IEEE Int. Conf. on Emerging Technologies and Factory Automation (ETFA)*, Bilbao - Spain, Sept. 13-16 2010.

[10] ——, "Design of adaptive pca controllers for siso systems," in *18th World Congress of the International Federation of Automatic Control (IFAC WC)*, Milano - Italy, Aug.28-Sept.02 2011.

[11] L. Brito Palma, F. Coito, and P. Gil, "Design of adaptive sliding window pi-pca controller," in *IEEE 20th Mediterranean Conference on Control and Automation (MED)*, Barcelona - Spain, July 3-6 2012.

[12] M. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AICHE Journal*, vol. 37, no. 2, pp. 233 – 243, 1991.

[13] K. Diamantaras, *Principal Component Neural Networks: theory and applications*. Wiley, 1996.

[14] M. Norgaard, O. Ravn, N. Poulsen, and L. Hansen, *Neural Networks for Modelling and Control of Dynamic Systems*. Springer, 2003.

[15] T. Harris, C. Seppala, and L. Desborough, "A review of performance assessment and process monitoring techniques for univariate and multivariate control systems," in *IFAC ADCHEM Conference*, Canada, 1997.

[16] D. Ender, "Process control performance: not as good as you think," *Control Engineering*, vol. 180, 1993.

[17] A. Siemens, "How to improve the performance of your plant using the appropriate tools of simatic pcs7 apc portfolio - white paper." Siemens AG, 2008.

[18] T. Harris, "Assessment of control loop performance," *Canadian Journal of Chemical Engineering*, vol. 67, pp. 856–861, 1989.

[19] A. Horch and A. Isaksson, "A modified index for control loop performance assessment," in *ACC Americal Control Conference*, USA - Philadelphia, 1998.

[20] S. Bezergianni and C. Georgakis, "Controller performance assessment based on minimum and open-loop output variance," in *Control Eng. Practice*, vol. 8, 2000, pp. 791–797.

[21] S. Choobkar, A. Sedigh, and A. Fatehi, "Input-output pairing based on the control performance assessment index," in *IEEE Int. Conf. on Advanced Computer Control (ICACC)*, China - Shenyang, 2010.

[22] L. Desborough and T. Harris, "Performance assessment measures for univariate feedback control," *Canadian Journal of Chemical Engineering*, vol. 70, pp. 1186–1197, 1992.

[23] N. Stanfelj, T. Marlin, and J. MacGregor, "Monitoring and diagnosing process control performance: the single loop case," *Ind. Eng. Chem.*, vol. 32, pp. 301–314, 1993.

[24] M. Jelali, "An overview of control performance assessment technology and industrial applications," *Control Engineering Practice*, vol. 14, pp. 441–466, 2006.

[25] J. Cano-Izquierdo, J. Ibarrola, and M. Kroeger, "Control loop performance assessment with a dynamic neuro-fuzzy model (dfasart)," *IEEE Trans. on Automation Science and Engineering*, vol. 9, pp. 377–389, 2012.

[26] B. Kuo, *Automatic Control Systems*. Prentice-Hall, 1995.

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

[28] ——, "Benchmark systems for pid control," in *IFAC PID 2000 Conference*, Spain, 2000, pp. 181 – 182.