Neural PCA Controller based on Multi-Models

Luis Brito Palma[1,2], Fernando Vieira Coito[1,2] and Paulo Sousa Gil[1,3]

[1] Universidade Nova de Lisboa - FCT - DEE, Caparica 2829-516, Portugal
[2] Uninova CTS Research Institute, Caparica 2829-516, Portugal
LBP,FJVC@fct.unl.pt
[3] CISUC Research Institute, Coimbra 3030-290, Portugal
PSG@fct.unl.pt

Abstract. In this paper, a new approach to design nonlinear adaptive PI multi-controllers, for SISO systems, based on neural local linear principal components analysis (PCA) models is proposed. The PCA neural networks only implements the integral term of the PI multi-controller, a proportional term is added to obtain a PI structure. A modified normalized Harris performance index is used for evaluating the controller performance. Some experimental results obtained with a nonlinear three tank benchmark model are presented, showing the adaptive PI-PCA multi-controller performance compared to neural linear PI controllers.

Keywords: nonlinear adaptive PI control, principal component analysis, multi-models.

1 Introduction

Industrial plants and processes are typically nonlinear systems, most of them controlled by linear or nonlinear proportional-integral (PID) controllers, [1, 2, 30]. Most loops use in fact only PI control because the derivative action is not used very often. The PID controller is used for a wide range of control problems: process control, motor drives, flight control, robotic systems in automotive industries, etc. The PID controllers appears in different forms: as standard single-loop hardware controllers, as software modules in programmable logic controllers (PLC's) and also as distributed multi-loop control systems in industrial plants.
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, 20, 26, 27]. PCA can also be used for controller loop monitoring, [9], as well as for feedback control in the scores space, [8, 26, 27].
The PCA control formulation, in a reduced control scores space, is analogous to eigenvalue-assignment control. The great advantage of the control structure proposed in this work is the fact that only input-output process data is needed for controller tuning, resulting in a data-based controller tuning approach using neural networks.
In this paper, a nonlinear adaptive PI multi-controller based on linear local neural PCA models, to deal with nonlinear SISO systems in real-time applications,

is proposed. This work is based on the previous work focused on a linear neural PCA controller, [6].

2 Control Loop Performance Analysis (CLPA) Approach based on a Modified Normalized Harris Index

In a typical industrial plant there are dozens or even hundreds of controllers, most of them based on the PID structure. Even if these controllers initially perform well, various factors can contribute to their performance deterioration, including [16, 18]: actuator/sensor or process faults, sticking valves producing oscillations, external disturbances, to name just a few. Around 50% of all industrial controllers, or more, have some kind of performance problem, [14], [28].
Traditional performance measures such as overshoot, rise time, settling time, control error, integral error criteria, etc, have been used by some researchers for CLPA. The most widespread criterion considered for CLPA is the variance. Most of the reported industrial applications of control loop performance monitoring are based on the pioneer method suggested by Harris, [15], or some algorithms inspired on Harris approach, such as the approaches found in the references [3, 19], 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, [11].
The first indices for control loop performance assessment were proposed by Harris [15], Desborough and Harris [12] and Stanfelj et al [29]. A review of the status in control loop performance assessment (CLPA) technology and industrial applications was published in 2006, by Jelali [21]. Merits and drawbacks of each CLPA method are highlighted. The application of fuzzy logic and neural networks were also investigated in the CLPA field, [10].
The modified normalized Harris index used in this work is described next, summarily, [9]. In this work, the algorithms were implemented in discrete-time: at time instant t_k, the sample is k. 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 signal $\delta(k)$ between the measured control error $e(k)$ and a forward prediction of the control error $\hat{e}(k)$, described by

$$\delta(k) = e(k) - \hat{e}(k) \tag{1}$$

In a control loop that is performing well it is expected that the control error contains only random noise. The normalized CLPA index used in this work $\vartheta(k)$, computed on-line for a sliding window (time horizon), is described by (2), where *var(.)* is the variance and *mse(.)* is the mean-squared error. 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 $\hat{e}(k)$, [9]. 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. This CLPA index is used here for evaluating the controllers performance in closed-loop.

$$\vartheta(k) = 1 - \frac{var(\delta(k))}{mse(e(k))} \tag{2}$$

3 Linear Integral Controller based on Classical PCA Model

In the past some approaches have been proposed for design classical integral controllers based on principal components analysis (PCA), [5, 7, 8, 26, 27]. 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 [5, 7, 8]. The block **"Pa"** represents the plant, **"C"** is the PCA controller, and $\boldsymbol{Q}$ is given by (3). Let's assume that k is the k sample in the algorithms.

$$\boldsymbol{Q} = (\boldsymbol{U}^T)^+ \tag{3}$$

In (3), + represents the pseudo-inverse of the matrix and T is the transposed. The matrix $\boldsymbol{U}$ is obtained from (4), where $x_d(k) = y(k)$ is the process output.

$$\boldsymbol{U} = (\boldsymbol{\Upsilon}^T \, x_d(k))^T \tag{4}$$

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 (5).

$$\boldsymbol{\Upsilon} = \boldsymbol{X} \, \boldsymbol{P} \tag{5}$$

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. $\boldsymbol{V} = (\boldsymbol{P})_{:,1\ldots a}$, for a SVD decomposition as follows (6), assuming that $\boldsymbol{\Lambda}$ is the singular values matrix.

$$cov(\boldsymbol{X}) = \frac{1}{n-1} \, \boldsymbol{X}^T \, \boldsymbol{X} = \boldsymbol{V} \, \boldsymbol{\Lambda} \, \boldsymbol{V}^T \tag{6}$$

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 (7).

$$\boldsymbol{t}(k) = \boldsymbol{x}(k) \, \boldsymbol{P} \tag{7}$$

The classical integral PCA controller was initially formulated by Piovoso, [26, 27]. Some developments using ARX models were proposed later, for static PCA models and also for adaptive PCA models, [5, 7, 8]. 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 (8).

$$x_{mp}(k) = x_{mp}(k-1) + K_i \, \Delta x_m(k) \tag{8}$$

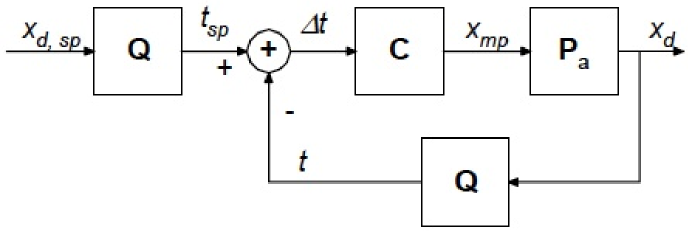

Fig. 1. Architecture of the classical integral PCA controller.

The incremental action depends on the manipulated matrix $\boldsymbol{P}_{mp}$ as described in (9).

$$\Delta x_m(k) = \Delta t(k)\ \boldsymbol{P}_{mp} \tag{9}$$

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

4 Linear Controller based on Neural PCA Model

The main ideas related to the design of integral controllers using neural PCA models can be found in [6]. The integral controller presented here is based on neural linear PCA that can be implemented accordingly to the architecture depicted in Fig. 2, [13, 22]. The auto-associative neural network implements data compression at the bottleneck layer (BL) using function G and data decompression at the output layer (OL) using function H.
The main idea is the 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.
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 (10), without loss of generality. The simulations were done assuming $n = 800, m = 5$, for a sampling period of $T_s = 1[s]$.

$$\boldsymbol{x}(k,:) = [y(k)\ y(k-1)\ y(k-2)\ u(k-1)\ u(k-2)] \tag{10}$$

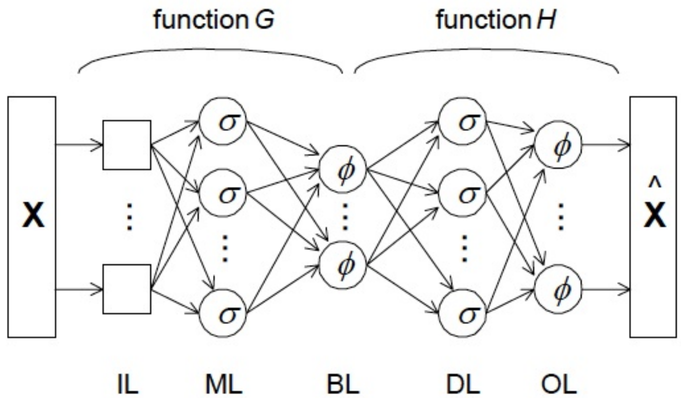

Fig. 2. Architecture of auto-associative neural network for PCA.

The training data should not contain very high frequency data since this can cause problems in the Levenberg-Marquardt optimization algorithm used in the training of the neural networks, [25].

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 3. 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 $\boldsymbol{W}_2$).

The $\boldsymbol{P}_n$ matrix, equivalent to the main singular vectors matrix, is obtained from the neural network weight matrices as described in (11).

$$\boldsymbol{P}_n = \boldsymbol{B}\ \boldsymbol{M} \tag{11}$$

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

$$\boldsymbol{P}_{ex} = (\boldsymbol{P}_n)_{:,1\ldots r} \tag{12}$$

$$\boldsymbol{P}_{mp} = (\boldsymbol{P}_n)_{:,r+1\ldots m} \tag{13}$$

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

$$\boldsymbol{M}_\lambda = (\boldsymbol{P}_{ex})^+ \boldsymbol{P}_{mp} \tag{14}$$

The error vector is given by

$$\Delta\boldsymbol{x}(k,:) = [e_c(k)\ e_c(k-1)\ e_c(k-2)] \tag{15}$$

The increment in the manipulated variable is expressed by

$$\Delta x_{mp}(k) = \Delta\boldsymbol{x}(k)\ (\boldsymbol{M}_\lambda)_{:,1} \tag{16}$$

The non-saturated control action is formulated as

$$u_{c0}(k) = u_c(k-1) + K_i \, \Delta x_{mp}(k) + a_w(k-1) \tag{17}$$

The saturated control action is given by

$$u_c(k) = satur(u_{c0}(k), ...) \tag{18}$$

The anti-windup term is expressed by

$$a_w(k) = K_{aw} \, [u_c(k) - u_{c0}(k)] \tag{19}$$

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 (20).

$$u_{c0}(k) = K_p \, e_c(k) + u_c(k-1) + K_i \, \Delta x_{mp}(k) + a_w(k-1) \tag{20}$$

5 Nonlinear Adaptive PI Multi-Controller based on Neural PCA Models

Inspired on the previous works [5, 6, 7, 8], a nonlinear adaptive PI multi-controller based on PCA multi-models, one PCA model for each setpoint range, is proposed here. The general architecture is depicted in Fig. 3. The supervisor, with a control-loop performance analysis module based on the modified Harris index as described in section 2, implements the adaptive mechanism used to compute the adaptive integral term. The block "Sat" is a saturation mechanism. As described in section 4 the integral term of the controller depends on the $\boldsymbol{P}_n$ matrix. This matrix is computed accordingly $\boldsymbol{P}_n = \boldsymbol{B}\,\boldsymbol{M}$. For each local linear neural PCA model a set of matrices $\{\boldsymbol{B}_i; \boldsymbol{M}_i\}$ is available after off-line training of the linear local PCA neural models. Assuming ζ local linear neural models for ζ setpoint ranges, each adaptive matrix $\{\boldsymbol{B}(k); \boldsymbol{M}(k)\}$ is computed on-line from a linear combination of the matrices of local models accordingly (21). The weights are computed accordingly (22), where $|e_i(k)|$ is the absolute value of the control error at each time instant k. This methodology weighs more nearby models.

The supervisor should guarantee a minimum dwell-time associated with switching between matrices in order to guarantee stability, [23, 24]. In this work a rounding approach was applied to the weights α_i in order to guarantee a minimum dwell-time.

$$\boldsymbol{M}(k) = \sum_{i=1}^{\zeta} \boldsymbol{\alpha}_i(k) \boldsymbol{M}_i(k) \quad \boldsymbol{B}(k) = \sum_{i=1}^{\zeta} \boldsymbol{\alpha}_i(k) \boldsymbol{B}_i(k) \quad \sum_{i=1}^{\zeta} \alpha_i(k) = 1 \tag{21}$$

$$\alpha_i(k) = \frac{\Gamma(k)}{\sum_{i=1}^{\zeta} \Gamma(k)} \quad \Gamma(k) = 1 - \frac{|e_i(k)|}{\sum_{i=1}^{\zeta} |e_i(k)|} \quad |e_i(k)| = |r_i(k) - y(k)| \tag{22}$$

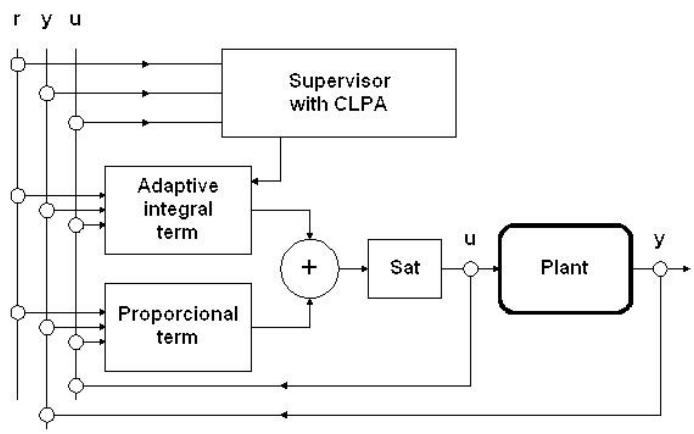

Fig. 3. Architecture for nonlinear neural PI-PCA multi-controller.

6 Experimental Results

In order to evaluate the performance of the proposed PI multi-controller structure, experimental results obtained with a simulation model of a three-tank system benchmark are presented, [17], as depicted in Fig. 4. The water levels (h1, h2 and h3) in each tank are the process state variables. In this benchmark, usually, the main goal is to maintain the water level, in the central tank, around a certain predefined set-point. The experiments show the water level evolution in the left tank T1. It is assumed that valve V13 and the outflow valve in the central tank T3 are open. The algorithms were implemented in Matlab in discrete-time k. The sampling time is 1 s. The input/output variables were normalized to the range [0;1].

In Fig. 5 can be observed an experiment done with the neural PCA multi-controller proposed, for two set-points $r1 = 0.15/0.6$ and $r2 = 0.3/0.6$. 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), high-pass filtering of reference signal (hpfr), weights α_i (wx), and finally the modified Harris index (ha). The controller gains are the following: $K_p = 1, K_i = 0.2\ \eta$, with $\eta = mean(abs(t1))$. The value $t1$ is the projection of data along the first singularvector, i.e., the first principal component.

The adaptive PCA multi-controller (AMC) was compared with two linear neural controllers (C1 & C2) tuned for each set-point $(r1, r2)$, for the same profile of set-points depicted in Fig. 5. The mean squared error of the modified Harris index obtained for each controller are the following, respectively, for the controllers AMC, C1 and C2: 0.368, 0.390 and 0.738. The multi-controller (AMC) reveals the best performance.

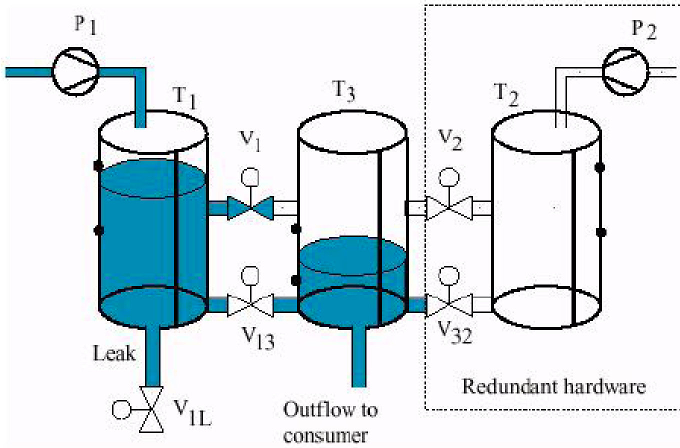

Fig. 4. Three-tank benchmark system.

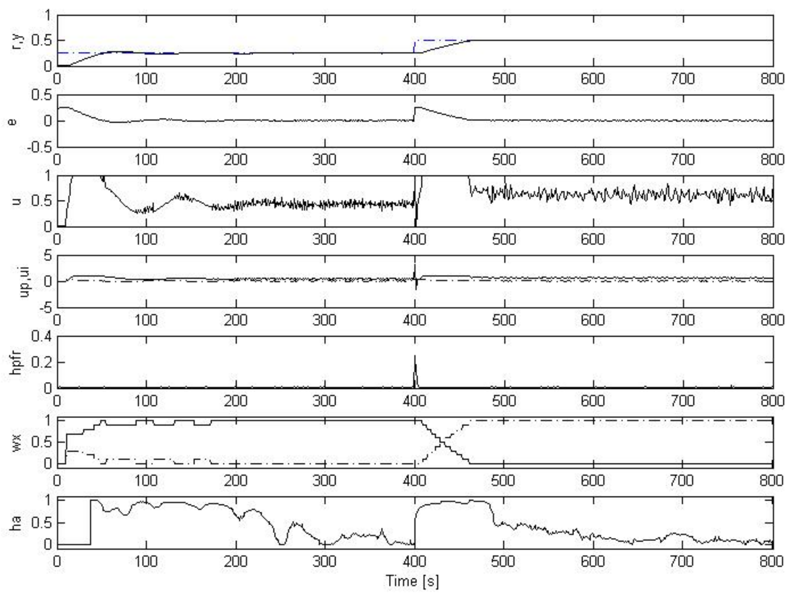

Fig. 5. Experiment with the multi-controller for two set-points, $r1$ and $r2$.

7 Conclusions

In this paper, a new approach for design nonlinear adaptive PI multi-controllers based on PCA multi-models, for SISO systems, was proposed.

Since neural networks are very sensitive to data, high frequency data should be avoided in the training phase of each PCA neural network. Only data around each set-point should be considered for the training of each local linear PCA model used to build the integral term of the PI multi-controller. The behaviour of the integral neural PCA controller, in terms of overdamped or underdamped responses, depends on the spectral content of the data used in the training phase of the neural networks. The obtained results are very promising. A good performance was obtained for different simulations as observed in the behaviour of the modified normalized Harris index. More tests should be done to evaluate this approach with other classes of systems.
Some future research pointers are: a) improvement of the approaches for better tuning the controller gains; b) discover the classes of systems that are apropriate for this kind of controllers; c) the comparison with classical adaptive PI controllers based on gain scheduling; d) the robustness of the multi-controller against faults and failures should also be analysed; e) the generalization for MIMO systems should also be investigated.

Acknowledgments.

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] Ang, K., Chong, G., Li, Y.: Pid control system analysis, design, and technology. IEEE Transactions on Control Systems Technology 13(4), 559–576 (2005)
[2] Astrom, K., Hagglund, T.: The future of pid control. Control Engineering Practice 9, 1163 – 1175 (2001)
[3] Bezergianni, S., Georgakis, C.: Controller performance assessment based on minimum and open-loop output variance. In: Control Eng. Practice. vol. 8, pp. 791–797 (2000)
[4] Brito Palma, L.: Fault Detection, Diagnosis and Fault Tolerance Approaches in Dynamic Systems based on Black-Box Models. Ph.D. thesis, Universidade Nova de Lisboa, Portugal (2007)
[5] Brito Palma, L., Coito, F., Gil, P.: Design of adaptive sliding window pi-pca controller. In: IEEE 20th Mediterranean Conference on Control and Automation (MED). Barcelona - Spain (July 3-6 2012)
[6] Brito Palma, L., Coito, F., Gil, P.: Pi controller for siso linear systems based on neural linear pca. In: European Control Conference (paper accepted). Strasbourg - France (June 24-27 2014)
[7] Brito Palma, L., Coito, F., Gil, P., Neves-Silva, R.: Process control based on pca models. In: 15th IEEE Int. Conf. on Emerging Technologies and Factory Automation (ETFA). Bilbao - Spain (Sept 13-16 2010)
[8] Brito Palma, L., Coito, F., Gil, P., Neves-Silva, R.: Design of adaptive pca controllers for siso systems. In: 18th World Congress of the International Federation of Automatic Control (IFAC WC). Milano - Italy (Aug28-Sept02 2011)

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

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

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

[12] Desborough, L., Harris, T.: Performance assessment measures for univariate feedback control. Canadian Journal of Chemical Engineering 70, 1186–1197 (1992)

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

[14] Ender, D.: Process control performance: not as good as you think. Control Engineering 180 (1993)

[15] Harris, T.: Assessment of control loop performance. Canadian Journal of Chemical Engineering 67, 856–861 (1989)

[16] Harris, T., Seppala, C., Desborough, L.: A review of performance assessment and process monitoring techniques for univariate and multivariate control systems. In: IFAC ADCHEM Conference. Canada (1997)

[17] Heiming, B., Lunze, J.: Definition of the three-tank benchmark problem for controller reconfiguration. In: European Control Conference. Karlsruhe - Germany (1999)

[18] Horch, A.: Condition Monitoring of Control Loops. Ph.D. thesis, Royal Institute of Technology - Sweden (2000)

[19] Horch, A., Isaksson, A.: A modified index for control loop performance assessment. In: ACC Americal Control Conference. USA - Philadelphia (1998)

[20] Jackson, J.: A User's Guide to Principal Components. Wiley (2003)

[21] Jelali, M.: An overview of control performance assessment technology and industrial applications. Control Engineering Practice 14, 441–466 (2006)

[22] Kramer, M.: Nonlinear principal component analysis using auto-associative neural networks. AICHE Journal 37(2), 233 – 243 (1991)

[23] Lemos, J., Rato, L., Mosca, E.: Progress in Systems and Control Theory, chap. Integrating predictive and switching control: Basic concepts and an experimental case study. Birkhäuser Verlag (2000)

[24] Liberzon, D.: Switching in Systems and Control. Birkhauser Boston-MA (2003)

[25] Norgaard, M., Ravn, O., Poulsen, N., Hansen, L.: Neural Networks for Modelling and Control of Dynamic Systems. Springer (2003)

[26] Piovoso, M.: The Control Handbook, chap. The Use of Multivariate Statistics in Process Control, pp. 561 – 573. CRC Press (1996)

[27] Piovoso, M., Kosanovich, K.: Applications of multivariate statistical methods to process monitoring and controller design. Int. Journal of Control 59(3), 743 – 765 (1994)

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

[29] Stanfelj, N., Marlin, T., MacGregor, J.: Monitoring and diagnosing process control performance: the single loop case. Ind. Eng. Chem. 32, 301–314 (1993)

[30] Vilanova, R., Visioli, A. (eds.): PID Control in the Third Millennium - Lessons Learned and New Approaches. Springer (2012)