

Support Vector Based Novelty Detection for Fault Tolerant Control

Sergio Saludes and M.J. Fuente

Abstract—This article presents a method for achieving fault tolerant control in the frame of the nonlinear IMC control scheme. The method consists on the minimization of a cost function in order to compute new controller parameters values. In addition to this, a support vector based novelty detector is used to detect faults in the plant. Simulation results for a continuous stirred tank reactor are presented.

I. INTRODUCTION

The detection and isolation of faults in manufacturing processes is of great practical importance. The quick and correct fault diagnosis avoids production loss and product degradation. Manufacturing processes usually involve many variables that interact with one another. When any of these variables deviate beyond their specified limits, a fault occurs. The early approaches of fault detection are based on the threshold checking. Later, pattern classification and model identification were used. Currently, a large number of approaches are available such as machine learning and statistics [1], including those based on novelty detection [2].

Likewise, many fault tolerant control methods have been developed. These methods are capable of retaining satisfactory system performance in the presence of faults [3]. Fault tolerance in dynamic systems can be achieved either from system robustness to faults or from controller reconfiguration in response to specific faults. The former methodology, since no information about faults is utilized by the control system, can be referred as passive fault-tolerant control systems. By utilizing the fault information obtained from fault detection and identification (FDI) scheme, reconfigurable control modifies the control function in response to the faults. So, this is referred as active fault tolerant control [4].

As most plants are inherently nonlinear and the faults may often amplify the nonlinearities by driving the plants from a relatively linear operation point into a more nonlinear operation region, the study of fault tolerant control for nonlinear systems is important. In an attempt to solve this problem, methods such as neural networks, fuzzy systems and, more recently, support vector machines (SVM) [5], have been used due to their capabilities of making arbitrarily accurate approximations to any continuous nonlinear function.

A suitable control strategy that directly incorporates the plant model is provided by Internal Model Control (IMC). In this paper, fuzzy neural networks, called ANFIS [6], have been used for the construction of the plant model. Its

inverse have been implemented by means of an Elman neural network [7]. Both of them are used directly within the IMC control structure.

Besides this, a support vector based novelty detector is presented as a fault detection algorithm. The information provided by this algorithm is used to reconfigure the controller, that is, the inverse model of the system, which consists on a set of Elman neural networks. The key idea is to modify the output layer bias when a fault is detected. This bias is modified through the minimization of a cost function. Results for a continuous stirred tank reactor are presented.

This paper is organized as follows. In section II the support vector based novelty detector is presented. In section III the nonlinear IMC control scheme is described, together with the fault tolerant strategy. Next section gathers the results obtained related to control, fault detection and fault tolerance. Finally, section VI concludes the paper.

II. SUPPORT VECTOR BASED FAULT DETECTION

The principle of novelty detection offers an approach to the problem of fault detection which only requires the normal class to be defined. A model of normality is learnt by including normal examples only in the training data; abnormalities are then identified by testing for novelty against this description.

Novelty detection can be carried out by means of classical methods [8], and also through neural networks and others machine learning related methods [9].

One of the latest contributions to novelty detection comes from the kernel methods field. The present work follows the method proposed by [10], that proposes an algorithm which computes a binary function which is supposed to capture regions in the input space where the probability density is, i.e., a function such that most of the data will be in the region where the function is nonzero. Moreover, it is also applicable in cases where the density of the data distribution is no even well-defined.

A. v -SV Novelty Detection

Let X be the training data set

$$X = \{\mathbf{x}_1, \dots, \mathbf{x}_m\} \subset \mathcal{X} \quad (1)$$

where $m \in \mathbb{N}$ is the number of observations, and \mathcal{X} is some set. Let Φ be a feature map $\mathcal{X} \rightarrow \mathcal{H}$; i.e., a map into a dot product space \mathcal{H} such that the dot product in the image of Φ can be computed by evaluating some simple kernel

$$k(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle \quad (2)$$

Sergio Saludes is with CARTIF, Parque Tecnológico de Boecillo P205, 47151 Boecillo, Valladolid, Spain sersal@cartif.es

M.J. Fuente is with Department of Systems Engineering and Control. University of Valladolid, Science Faculty. Prado de la Magdalena s/n 47011 Valladolid, Spain maria@autom.uva.es

such as the Gaussian kernel

$$k(\mathbf{x}, \mathbf{x}') = e^{-\|\mathbf{x}-\mathbf{x}'\|^2/c} \quad (3)$$

The objective is to find a function f that takes the value +1 in a region capturing most of the data points, and -1 elsewhere. The strategy is to map the data into the feature space corresponding to the kernel, and to separate them from the origin with maximum margin. For a new point \mathbf{x} , the value $f(\mathbf{x})$ is determined by evaluating which side of the hyperplane it falls on, in feature space. Since there are several types of kernel functions, it is possible to obtain a variety of nonlinear estimators in the input space.

To separate the data set X from the origin, the following quadratic program must be solved:

$$\min_{w \in \mathcal{H}, \xi \in \mathbb{R}^m, \rho \in \mathbb{R}} \frac{1}{2} \|w\|^2 + \frac{1}{vm} \sum_i \xi_i - \rho \quad (4)$$

$$\text{subject to } (w \cdot \Phi(x_i)) \geq \rho - \xi_i, \xi_i \geq 0 \quad (5)$$

with $v \in (0, 1]$. In practice, the problem that is solved is the dual one of 4:

$$\min_{\alpha \in \mathbb{R}^m} \frac{1}{2} \sum_{ij} \alpha_i \alpha_j k(x_i, x_j) \quad (6)$$

$$\text{subject to } 0 \leq \alpha_i \leq \frac{1}{vm} \quad (7)$$

$$\text{and } \sum_i \alpha_i = 1 \quad (8)$$

Data \mathbf{x}_i with $\alpha_i \neq 0$ are called support vectors. Finally, ρ can be computed as

$$\rho = \sum_j \alpha_j k(x_j, x_i) \quad (9)$$

where the α_i associated to the \mathbf{x}_i holds that $0 < \alpha_i < 1/(vm)$.

The parameter v , as it is demonstrated in [11] has a clear meaning, whenever $\rho \neq 0$:

- 1) v is an upper bound on the fraction of novelties.
- 2) v is a lower bound on the fraction of support vectors.
- 3) Suppose the data were generated independently from a distribution $P(\mathbf{x})$ which does not contain discrete components. Suppose, moreover, that the kernel is analytic and non constant. With probability 1, asymptotically, v equals both the fraction of support vectors and the fraction of novelties.

III. THE NONLINEAR IMC

The basic idea of the linear Internal Model Control (IMC) is illustrated in Fig. 1. The key characteristic of this control design approach is the inclusion of a plant model within the control structure. If the model is a perfect representation of the process, the influence of the process output on the feedback signal vanishes. The feedback signal then only carries the influence of disturbances. However, in practice, the model of the plant is not a perfect description of the plant. The feedback signal then combines the model error with the disturbances. Based on this structure, perfect control is obtained if the controller C is chosen as the inverse of the internal model M . IMC controllers have been extensively

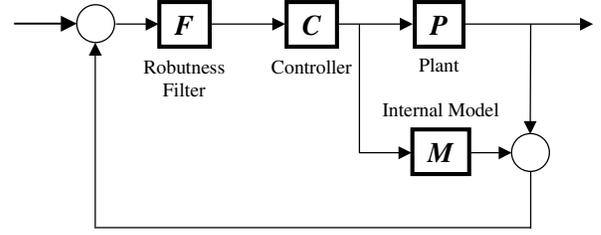


Fig. 1. Structure of the IMC control scheme

studied in the case of linear modeling of the process [12], and have been shown to have good robustness properties against disturbances and model mismatch.

The nonlinear version of the IMC is analogous to the linear one. The difference is that the model of the plant, M , and the controller, C , are nonlinear functions. Neural networks [13] and fuzzy systems [14], [15] have been used for those purposes. In this paper, a combination of neural networks and fuzzy systems is presented.

In order to obtain the model of the inverse of the plant, several approaches have been proposed. Some authors [13] have proposed online training based on synthetic signals; while others [14] have proposed invertible formulations for M .

IV. FAULT TOLERANT IMC WITHOUT A PRIORI KNOWLEDGE

A methodology for achieving the fault tolerance in the IMC controller has been developed. The methodology is suitable both for the linear and non linear version of the IMC controller and the solely requirement is that the controller C follows the pattern shown below.

$$\mathbf{u}(t) = f(\mathbf{y}(t-1), \dots, \mathbf{y}(t-n), \theta) + \mathbf{b} \quad (10)$$

where θ and \mathbf{b} are tuning parameters and n is the embedded dimension. Note that \mathbf{b} is a bias. Function f is a nonlinear one when the controller is applied to a nonlinear system.

The fault-tolerant strategy is based on the adaptation of parameter \mathbf{b} in (10) by means of the minimization:

$$J = \min_{\tilde{\mathbf{b}} \in \mathbb{R}^m} \sum_{t=N_{e_1}}^{N_{e_2}} g(\tilde{\mathbf{b}}, \mathbf{e}(t) + \Delta) \quad (11)$$

where g is a quadratic function, $\mathbf{e}(t)$ is the control error, m is the dimensionality of the control vector, N_{e_1} and N_{e_2} bound a temporal window and Δ is the observed output deviation whose origin is the presence of a fault:

$$\Delta = \text{ref}(t_{fault} + \delta) - y(t_{fault} + \delta) \quad (12)$$

In (12), ref is the reference and t_{fault} is the instant in which the fault is detected. δ is a time delay introduced to allow the system to reach the stable fault state. Usually, δ can be neglected because of the fault detection delay.

The temporal window in (11), $[N_{e_1}, N_{e_2}]$, holds that $t_{fault} + \delta < N_{e_1}$. This is to avoid the influence of transients in the computation of cost function J .

The optimization problem is solved in parallel, i.e., the plant keeps running under faulty state while the new parameter \mathbf{b} is being computed. When the computation has finished, the new parameter \mathbf{b} is applied and the fault is overcome.

V. APPLICATION TO A NONLINEAR PROCESS

A. Plant description

The plant used in this study is the Continuous Stirred Tank Reactor (CSTR) described in [16]. It is a CSTR with a cooling jacket in which a Van Der Vusse chemical reaction type takes place. In this reaction, a product A reacts to produce B and C products. At the same time B produces D . The flow V fed to the reactor contains only substance A with concentration C_{A0} and temperature T_0 . The heat removal Q_K can be withdrawn from the coolant by an external heat exchanger. There is an unmeasurable perturbation in the temperature T_0 .

The dynamics of the reactor can be described by the following nonlinear differential equations that are derived from component balances for substances A and B and from energy balances for the reactor and cooling jacket:

$$\dot{C}_A = \frac{V}{V_R} (C_{A0} - C_A) - k_1(T)C_A - k_3(T)C_A^2 \quad (13)$$

$$\dot{C}_B = -\frac{V}{V_R} C_B + k_1(T)C_A - k_3(T)C_B \quad (14)$$

$$\begin{aligned} \dot{T} = & \frac{V}{V_R} (T_0 - T) - \frac{1}{\rho C_p} (k_1(T)C_A \Delta H_{R_{AB}} \\ & + k_2(T)C_B \Delta H_{R_{BC}} + k_3(T)C_A^2 \Delta H_{R_{AD}}) \\ & + \frac{k_w A_R}{\rho C_p V_R} (T_k - T) \end{aligned} \quad (15)$$

$$\dot{T}_K = \frac{1}{m_k C_{pK}} (Q_K + k_w A_R (T - T_K)) \quad (16)$$

The concentrations of substances A and B are C_A and C_B , respectively. The temperature in the reactor is denoted by T , while the temperature in the cooling jacket is given by T_K . The reaction velocities k_i are assumed to depend on the temperature via the Arrhenius law:

$$k_i(T) = k_{i0} \cdot \exp\left(\frac{E_i}{T + 273.15}\right), i = 1, 2, 3 \quad (17)$$

The values of the parameters can be found in [16]. The manipulated variables are V/V_R , where V_R is the tank volume, and Q_K . It is assumed that concentrations C_B and C_A and temperature T can be measured. Both C_B and T are the controlled variables, but only the concentration C_B is of technological interest, so it is in this output where the control effort must be put. The control tolerance for C_B is 0.02 mol/l and the sampling time is 18 seconds .

For the purpose of this work, sensor and actuator faults have been excluded. Only faults in the parameters E_i , which are related to the catalysts degradation, have been taken into

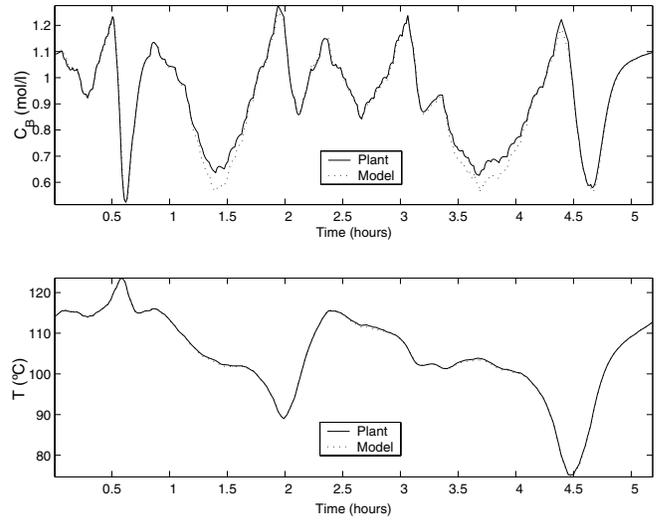


Fig. 2. ANFIS output for plant model M with validation data

consideration. Abrupt faults are the only type that has been studied.

B. Application of the nonlinear IMC

The three elements that conform the IMC controller are the internal model M , the controller C and the robustness filter F .

1) *Internal model:* In this paper, the nonlinear plant model proposed is the Adaptive Network Based Fuzzy Inference System (ANFIS), described in [6]. The ANFIS architecture implements a fuzzy inference system, equivalent to a Tagaki–Sugeno one. Once the ANFIS is trained, it plays the M operator role in the control structure. The ANFIS is trained in the classical way, i.e., the error signal used to adjust the network weights is the difference between the plant and network outputs.

Since the ANFIS can only have one output and the plant under experimentation has two outputs, two different ANFIS must be trained, one for the $C_B(k)$ output and another for the $T(k)$ output. For both of them, the input vector \mathcal{J}_M is

$$\mathcal{J}_M = \left\{ \frac{V}{V_R} (k-1), Q_K (k-1), C_B (k-1), T (k-1) \right\} \quad (18)$$

where k is the sampling period. Both ANFIS also share the input structure. They have four gaussian membership functions for each input. For training the nets, two step trains in the manipulated inputs with different amplitudes and frequencies were generated. The resulting ANFIS, one for each output, have four rules. Fig. 2 shows the validation results obtained after training.

The ANFIS has been trained by means of the Matlab Fuzzy Logic Toolbox.

2) *Controller:* In the IMC scheme described, the controller role is played by a model of the inverse of the plant dynamics. Some methods for training such a model have been mentioned in section III, but in practice, is not unusual

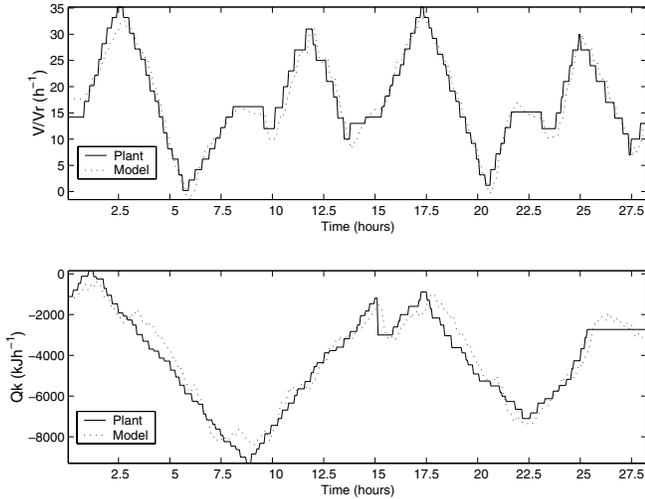


Fig. 3. Elman neural network output for inverse dynamics model C validation data

to use signals acquired in the plant. The plant outputs are used as inputs in the training set and plant inputs become the outputs in the training set [17]. This has been the method followed in this work for training an Elman neural network [7] with 200 logistic nodes in the hidden layer and two linear nodes in the output layer. One of the outputs corresponds to V/V_R and the other one to Q_K . The input vector is

$$\mathcal{J}_C = \{C_B(k-1), \dots, C_B(k-6), T(k-1), \dots, T(k-6)\} \quad (19)$$

where k is the sampling period. The results for the validation set are shown in Fig. 3.

The neural net was training using the backpropagation with momentum and adaptive learning rate algorithm that is implemented in the Matlab Neural Networks Toolbox and required 8940 epochs.

3) *Robustness filter*: One of the classical properties of the IMC structure is that if the controller is equal to the model inverse, the error between the plant output and its desired setpoint is asymptotically null, i.e., the IMC implements an integral action. However, an important error can generate a signal that may deteriorate performances or even cause instability. The addition of a low-pass filter adds robustness characteristics in the control structure. In this paper, the filter F is taken as

$$F(z^{-1}) = \frac{\Gamma z}{[z - (1 - \Gamma)]} \quad (20)$$

where $\Gamma = 1 - \alpha$ and, for ensuring stability, $0 < \alpha < 1$ must be held. The value that has been chosen is $\alpha = 0.9$.

4) *Control results*: Fig. 4 shows the plant outputs and the respective references for an experiment in the absence of perturbation. In this case, the mean error for C_B is -0.01 ; in case perturbation is present, the error is -0.0096 .

C. Fault Detection

A support vector novelty detector is used for detecting faults. To do this, a set of data representing all possible fault-

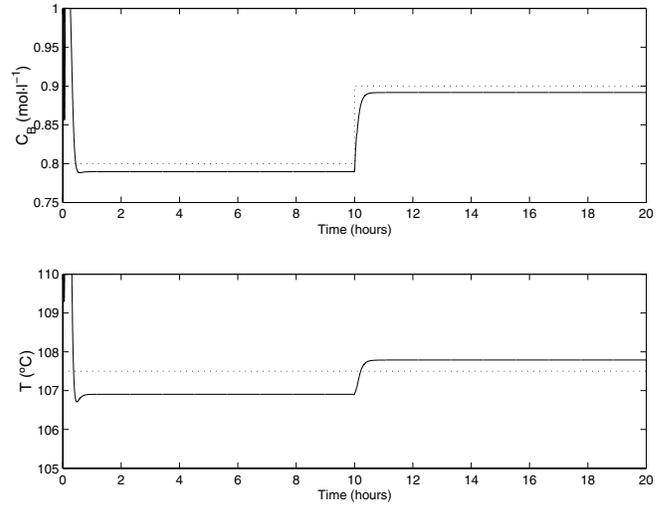


Fig. 4. Controlled plant C_B output in absence of faults and perturbations

free states have been gathered. Then, a One-Class SVM was trained using this data set. In this way, it is obtained a function f that takes the value $+1$ where fault-free states data live and -1 elsewhere, i.e., where faulty states data are supposed to be.

Attempts have been made to detect faults by three different methods:

- 1) First, a ν -SV has been trained using data from the plant. To do this, input vectors have been formed with plant input-output data,

$$\mathcal{J}_{method\ 1} = \left\{ \frac{V}{V_R}(k), Q_K(k), C_B(k), T(k) \right\} \quad (21)$$

The ν -SV had $\nu = 0.01$ and a gaussian kernel with $c = 0.01$. The training set had 900 samples and 10 support vectors were obtained. This is *method 1*.

- 2) Then, a different ν -SV has been trained, but this time two residuals have been used to form the input vector. These residuals were computed as the difference between the measured values of C_A and C_B , and the estimated values computed by means of the internal model of the plant M . The ν -SV parameter was $\nu = 0.01$ and the kernel was the gaussian one with $c = 0.03$. 586 samples were in the training set and 8 support vectors were obtained. This is *method 2*.

- 3) Finally, the third method was to apply a threshold to the residuals r_{C_A} and r_{C_B} . The threshold value was 0.15. The input used in these two last methods is

$$\mathcal{J}_{method\ 2,3} = \{r_{C_A}, r_{C_B}\} \quad (22)$$

where $r_{C_A} = C_A - \widehat{C}_A$ and $r_{C_B} = C_B - \widehat{C}_B$.

In order to determine the best approach, several experiments have been made. The results for abrupt faults in the parameters E_i , whose nominal values are $E_1 = E_2 = 9758.3\ K$ and $E_3 = 8560\ K$, are shown in table I. In this table, detection delay is the time, in seconds, between the fault induction and the time the alarm signal first changes to fault. False

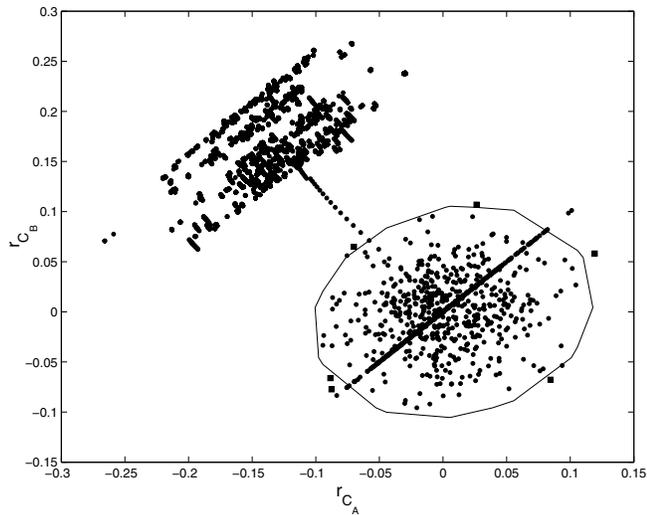


Fig. 5. Residuals for faulty and non-faulty state. The closed line is the boundary. The region inner to it corresponds to non-faulty states. The squared points are the support vectors.

TABLE I
DETECTION OF ABRUPT FAULTS IN THE PARAMETERS E_i

Change size	E_1 300	E_2 300	E_3 300	
Detection delay	192.63	4914	19593	Method 1
False alarms	0.0816	0.0821	0.0816	
False no-alarms	0.091	0.4577	0.9396	
Detection delay	36	36	108	Method 2
False alarms	0.005	0.005	0.005	
False no-alarms	0.0001	0.0001	0.003	
Detection delay	60.48	88.2	180	Method 3
False alarms	0.0	0.0	0.0	
False no-alarms	0.0022	0.0465	0.0256	

alarms is the ratio between the number of samples that are classified as faults and the number of samples before the fault is induced, ideally it should be 0. False no-alarms is the ratio between the number of samples that are classified as non-faulty and the number of them after the fault is induced, ideal value is 0.

The poorest results are obtained using method 1. The threshold method, method 3, reaches the 0 in false alarms ratio, but the method 2, i.e. residues processed with v -SV, has the closest to 0 false no-alarms value and it is the fastest detection method.

An example of fault detection by means of method 2 is shown in Fig. 5. The enclosed area is where the v -SV function takes the value 1 and the excluded region is where it takes the value -1 . So, all points laying outside the boundary are novelties, i.e., they are faults.

The v -SV have been trained using LibSVM, that is described in [18]. The parameters ν and c have been chosen by grid partition.

D. Fault Tolerant Control

To test the ability for tolerating abrupt faults in parameters E_i , an experiment has been carried out. The fault affects parameter E_1 and consists in a diminution of 75 K. The

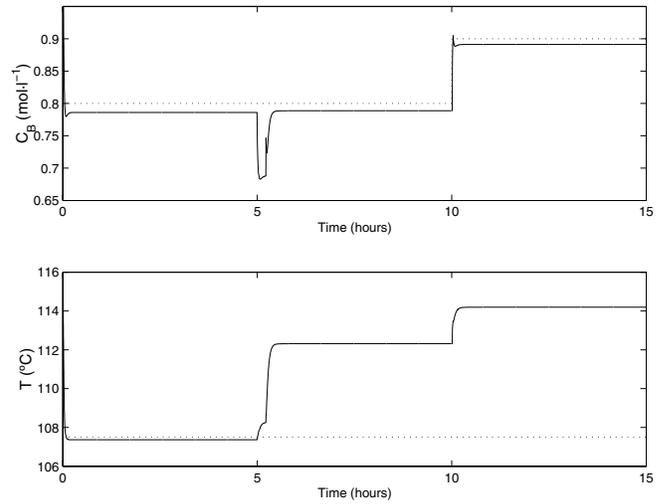


Fig. 6. Accommodation of an abrupt fault in E_1

TABLE II
ACCOMMODATION RESULTS FOR TWO DIFFERENT DETECTION METHODS

	v -SV	threshold
Detection delay	90	144
Bias computation time	710.28	771.12
Time to accommodation	800.28	1044

method used for detecting the fault is the v -SV with inputs shown in (22), i.e., method 2. The result is shown in Fig. 6. Besides this, and for comparing purposes, the same fault has been accommodated using as detection method the threshold evaluation over the same inputs, i.e., method 3. The results for this comparison are in table II and Fig. 7. In these experiments, δ in (12) is zero and function g in (11) is $\|\mathbf{ref}(k) - \mathbf{y}(k)\|^2$, where $\mathbf{ref}(k)$ is the reference and $\mathbf{y}(k)$ is the plant output. The minimization is performed using the Nelder-Mead simplex method.

The fault tolerant method based on v -SV is faster than the one based on threshold evaluation. The main reason is the higher ability of the v -SV for detecting faults, as it was shown in table I.

VI. CONCLUSION

A fault tolerant nonlinear IMC is presented in this paper. The nonlinear IMC is achieved by means of two nonlinear and machine learning based models. The internal model of the plant is implemented by means of an ANFIS and the plant dynamics inverse is modeled with an Elman neural network. This network is trained directly from input-output data gathered from the plant.

For the purpose of fault detection, these are understood as novelties regarding to the normal behavior. So, a novelty detector is used for detecting faults. To detect novelties, a one-class SVM classifier is used. Comparative results with two features and threshold and SVM based classifiers are presented. The one based on a one-class SVM that classifies the residues, is the one with a low false no-alarms rate and it is the fastest detection method.

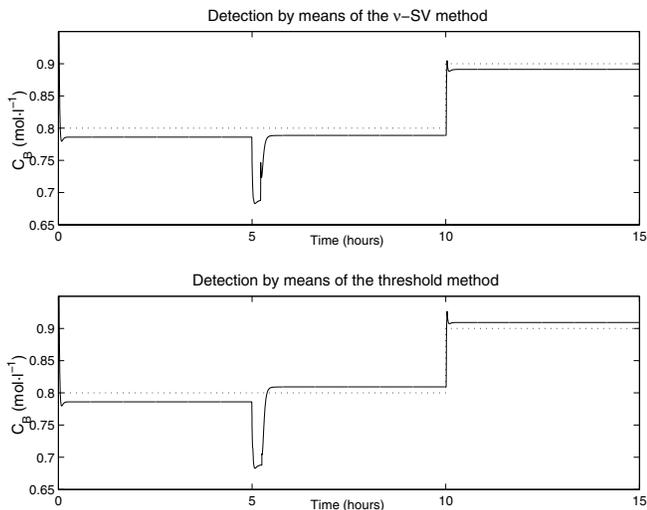


Fig. 7. Accommodation of an abrupt fault in E_1 by method v -SV and threshold method

When a fault is detected, the Elman neural network output layer bias are changed in order to accommodate the fault. This change is performed minimizing a cost function which depends on the controller parameters and the error induced by the fault.

The proposed fault tolerant IMC has been checked in simulation in a continuous stirred tank reactor, which is a highly nonlinear plant.

The results show that the best performance is obtained when the fault is detected by means of a one-class SVM with the residues as inputs.

VII. ACKNOWLEDGEMENT

This work was supported by the National Research Agency of Spain (CICYT), through project DPI2003-09373.

REFERENCES

- [1] J. Chen and R. J. Patton, *Robust Model-Based Diagnosis for Dynamic Systems*. Kluwer Academic Publishers, 1999.
- [2] L. Tarassenko, A. Nairac, N. W. Townsend, I. Buxton, and P. Cowley, "Novelty detection for the identification of abnormalities," *International Journal of Systems Science*, vol. 31, no. 11, pp. 1427-1439, 2000.
- [3] R. J. Patton, "Fault tolerant control: The 1997 situation," in *Proc. IFAC SAFEPROCESS*, Hull, U.K., 1997, pp. 1029-1051.
- [4] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and Fault-Tolerant Control*. Springer, 2003.
- [5] M. Ge, G. Zhang, R. Du, and Y. Xu, "Application of support vector machine based fault diagnosis," in *Proc. of the 15th IFAC World Congress*, Barcelona (Spain), 2002.
- [6] J. Jang, "ANFIS: Adaptive-network-based fuzzy inference system," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 23, pp. 665-685, 1993.
- [7] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, pp. 179-211, 1990.
- [8] M. Markou and S. Singh, "Novelty detection: A review. part 1: Statistical approaches," *Signal Processing*, vol. 83, no. 12, pp. 2481-2497, December 2003.
- [9] —, "Novelty detection: A review. part 2: Neural network based approaches," *Signal Processing*, vol. 83, no. 12, pp. 2499-2521, December 2003.
- [10] B. Schölkopf, R. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt, "Support vector method for novelty detection," *Advances in Neural Information Processing Systems*, vol. 12, pp. 582-588, 2000.
- [11] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol. 13, pp. 1443-1471, 2001.
- [12] M. Morari and E. Zafriou, *Robust Process Control*. Prentice-Hall, 1989.
- [13] K. J. Hunt and D. Sbarbaro, "Neural networks for nonlinear internal model control," in *IEE Proceedings-D*, IEE, Ed., vol. 138. IEE, Septiembre 1991, pp. 431-438.
- [14] S. G. Reda Boukezzoula and L. Foulloy, "Nonlinear internal model control: Application of inverse model based fuzzy control," *IEEE Transactions on Fuzzy Systems*, vol. 11, no. 6, pp. 814-829, December 2003.
- [15] S. Saludes and M. J. Fuente, "Fault tolerant fuzzy IMC control in a pH process," in *Proc. of the European Control Conference*, Cambridge, UK, 2003.
- [16] H. Chen, A. Kremling, and F. Allgöwer, "Nonlinear predictive control of a benchmark CSTR," in *Proceedings of the 3rd European Control Conference*, Rome, September 1995, pp. 3247-3252.
- [17] M. Araúzo, E. Gómez, J. Cano, M. López, and A. Collados, "A control of the penicillin production with adaptive IMC using fuzzy neural networks," in *14th IFAC World Congress*, Beijing, China, July 1999, pp. 499-504.
- [18] C.-C. Chang and C.-J. Lin, "LibSVM: a library for support vector machines," 2001, available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.