



Online fuzzy identification for an intelligent controller based on a simple platform

Sašo Blažič^{a,*}, Igor Škrjanc^a, Samo Gerkič^b, Gregor Dolanc^b, Stanko Strmčnik^b, Mincho B. Hadjiski^c, Anna Stathaki^d

^a Faculty of Electrical Engineering, University of Ljubljana, Tržaška 25, 1000 Ljubljana, Slovenia

^b Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia

^c UCTM Sofia, Kliment Ohridski Blvd. 8, Sofia, Bulgaria

^d Computer Technology Institute, Akteou and Pouloupoulou St. 11, Athens, Greece

ARTICLE INFO

Article history:

Received 12 April 2006

Received in revised form

15 April 2008

Accepted 2 March 2009

Available online 22 April 2009

Keywords:

Fuzzy identification

Programmable controllers

Online learning

Nonlinear control systems

pH neutralisation process

ABSTRACT

The paper presents the identification issues of the self-tuning nonlinear controller ASPECT (Advanced control algorithmS for Programmable logic controllers). The controller is implemented on a simple PLC platform with an extra mathematical coprocessor, but is intended for the advanced control of complex processes. The model of the controlled plant is obtained by means of experimental modelling. A special batch-wise algorithm that is based on the Takagi–Sugeno model and uses “fuzzy instrumental variables” technique is described in the paper. Many robustness problems of the classical adaptive approaches can be circumvented to some extent by the proposed batch-wise approach combined with a supervisory mechanism. The paper also includes some experimental results on the hydraulic pilot plant and some simulation case studies.

© 2009 Elsevier Ltd. All rights reserved.

1. Introduction

The problem of the control of nonlinear plants has received a great deal of attention in the scientific community. The problem itself is fairly demanding, but when the model of the plant is unknown or poorly known, obtaining the solution becomes considerably more difficult. Nevertheless, several approaches exist to solve the problem.

One possibility is to apply adaptive control that was adapted to treat not only linear time-invariant plants but also nonlinear and time-variant plants. Adaptive control is avoided in practice due to the lack of robustness, even though robust adaptive control was proposed to overcome this drawback (Ioannou and Sun, 1996). Since the adaptation is the key problem of the adaptive control, many papers deal with this topic. Fortescue et al. (1981) used variable forgetting factor to prevent blowing-up of the covariance matrix. Hilhorst et al. (1994) proposed a special switching mechanism that does not require a new identification if a previously visited operating point is recognised.

Many successful applications of fuzzy and neural network-based controllers (Sugeno and Nishida, 1985; Škrjanc and Matko, 2000; Precup et al., 2000, 2007) have shown their ability to control nonlinear plants. Fuzzy controllers were later upgraded

with the ability to construct a fuzzy model of the plant online and adjust control parameters accordingly (Layne and Passino, 1993; Škrjanc et al., 1997). The universal approximation theorem (Wang and Mendel, 1992) provided a theoretical background for new fuzzy direct and indirect adaptive controllers (Blažič et al., 2003; Tang et al., 1999) whose stability was proven using the Lyapunov theory.

There is also the possibility to design “pure” nonlinear controllers (Vidyasagar, 1993) if a very good model of the plant is available in advance, which is rarely the case.

The main drawback to most of the existing approaches for controlling nonlinear plants is that they are very complex and difficult to understand, since they demand a good knowledge of mathematics, and are thus avoided by practising engineers. Consequently, these methods are rarely applied to simple controllers such as programmable logic controllers. Indeed, reports from industrial practice show that the total number of plants using advanced control technology is relatively low. In Japanese industry the majority of the market share in advanced control is taken by advanced PID controllers, followed by auto-tuning controllers, gain-scheduling controllers, MPC, fuzzy controllers, etc. (Takatsu et al., 1998).

The area of nonlinear system identification has been a very active field of research in recent years. Many different approaches can be adopted to model nonlinear dependencies in the system, e.g., fuzzy systems, artificial neural networks, splines, the use of different basis functions, etc. In our case fuzzy systems are used.

* Corresponding author. Tel.: +386 1 4768763; fax: +386 1 4264631.
E-mail address: saso.blazic@fe.uni-lj.si (S. Blažič).

Wang and Mendel (1992) showed that fuzzy systems can be used to approximate arbitrary function that maps the compact set from the input space to the output with any desired degree of accuracy.

The goal of the ASPECT (Advanced control algorithmS for Programmable logiC conTrollers) project was to design a self-tuning nonlinear controller, intended for the control of a practically very important class of nonlinear plants whose dynamics change over the range of operation. The actual goal was to come up with a controller that outperforms classical conservatively tuned PID controller that is often used to control mildly nonlinear plants. What is even more important, the controller should be robust to disturbances and noise, so some safety mechanisms should be added. Moreover, the algorithms that are used should be simple to tune, i.e., the number of necessary design parameters should be kept low. The distinguishing property of this controller is that it runs on a simple platform, e.g., programmable logic or open controller. A very important feature is also that it adapts to changes in the environment. Since a classical adaptive nature is avoided in practice, the supervised batch-wise identification approach was chosen which is an alternative to using methods for preventing erroneous adaptation described above.

It is also very important to limit the use of the controller to a certain class of plants. The plants to be controlled should be stable, minimum phase plants, where the dominant dynamics are of the second order. Some delay can also be present in the plant. The nonlinearity of the plant should be a smooth function that depends only on one measured variable. Optionally, there can be one measured disturbance (MD) present in the system. The plant is also allowed to be slowly time variant. In such case the stability problems of switching do not arise (Narendra et al., 1995). The above-mentioned conditions can be slightly violated since the identification and the control algorithms are robust to reasonable deviations from the assumed properties. The described class of plants covers a lot of plants in the process industries. In many cases, such plants are currently controlled by conservatively tuned PID controllers. Our wish was to make a step further and offer a simple controller that is capable of adapting to not too complex nonlinear plants.

To fulfil the above-mentioned demands, the Takagi–Sugeno fuzzy model of low order (Takagi and Sugeno, 1985) was chosen. The model is obtained via experimental modelling using a special batch-wise online learning procedure. Very important part of the procedure are pre- and post-identification steps. In the former, the data for the identification are carefully monitored. Only if the excitation is satisfactory in a particular fuzzy domain, the identification procedure starts. After the identification, a special supervisory mechanism decides whether the model is accepted for the future use or not. Such an approach circumvents many traditional problems of classical adaptive control (where online testing of the data is very hard) thus providing reliable operation of the system. Many different controller types can be used with this approach, e.g., PID, predictive. The system was designed so that we are not confined to a certain controller type. Rather, many different controllers can be incorporated.

In Section 2 a brief overview of the controller is given, while in Section 3 the module for online learning is presented. Section 4 is devoted to the identification algorithm used in the controller. Sections 5 and 6 present the test results. In Section 7 the conclusions are stated.

2. Controller overview

The controller code is subdivided into the run-time module (RTM), running on a PLC (Mitsubishi A1S series PLC with an INEA

IDR SPAC20 coprocessor, based on the Texas Instruments DSP TMS320C32 at 40 MHz with 2 MB of RAM, and a Mitsubishi MAC E700 HMI unit), and the configuration tool (CT), which simplifies the initial configuration from a personal computer, providing guidance through the configuration procedure. This paper will only discuss the RTM. The modular multi-agent structure of the RTM enables a choice of several control algorithms suitable for different plants. The parameters of the control algorithms are automatically tuned from the model. The controller monitors the resulting control performance and may react to detected irregularities. A distinguishing feature of the controller is that the algorithms are adapted for implementation on low-cost industrial hardware platforms such as programmable logic or open controllers.

The code that resides in the controller (RTM) can be viewed as a multi-agent system where several independent agents (modules) interact with each other. The system comprises the following agents, as shown in Fig. 1:

- Signal preprocessing agent (SPA)—provides the signals to the other agents.
- Online learning agent (OLA)—identifies the model.
- Model information agent (MIA)—maintains the active model (model-in-use) and its status information.
- Control algorithm agent (CAA)—includes the functionality of an advanced industrial nonlinear control algorithm and automatic tuning of its parameters from the model; the tuning layer of the fuzzy parameter-scheduling controller is based on the magnitude optimum criterion implemented using the multiple integration method (Vrančić et al., 2001).
- Control performance monitor (CPM)—supervises the control performance and intervenes if appropriate.
- Operation supervisor (OS)—the main part of the program that binds the other agents.

In the following sections the OLA agent will be described in detail.

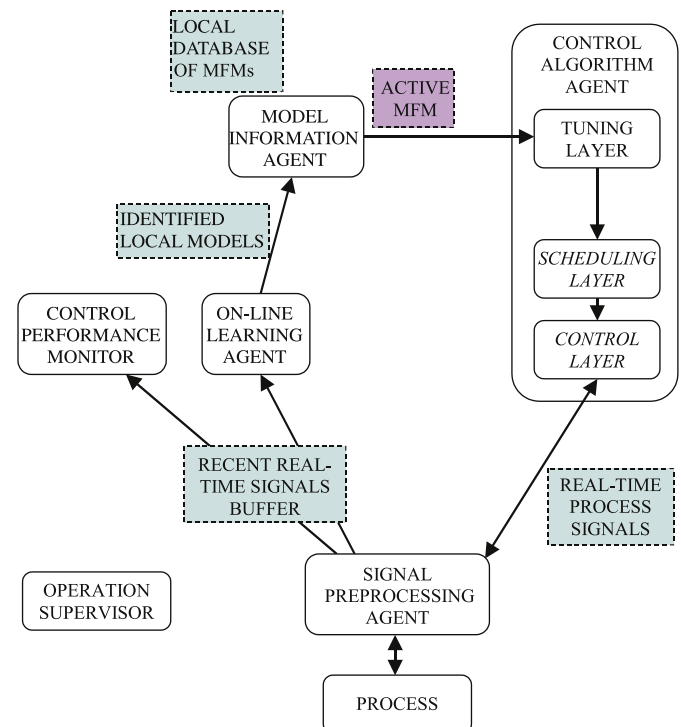


Fig. 1. Overview of the run-time module.

3. Online learning agent

The controller is model-based, founded on a multi-faceted model (MFM) that includes several model forms required by the online-learning mechanism and the control algorithm agents. The multi-faceted nature in our case means that the plant is described by a set of first-order and second-order local affine discrete models with delay. Correspondingly, two different controllers reside in the memory. Which controller is used in the closed loop (the one that is based on the first-order plant model or the one that is based on the second-order model) depends on the user choice or is selected automatically. As already mentioned, the local models are blended together, so that the resulting model can be seen as a Takagi–Sugeno model with m fuzzy domains. The second-order model of the plant has the following form:

$$y(k) = - \sum_{j=1}^m \beta_j a_{1j} y(k-1) - \sum_{j=1}^m \beta_j a_{2j} y(k-2) + \sum_{j=1}^m \beta_j b_{1j} u(k-1-d_{uj}) + \sum_{j=1}^m \beta_j b_{2j} u(k-2-d_{uj}) + \sum_{j=1}^m \beta_j c_{1j} v(k-1-d_{vj}) + \sum_{j=1}^m \beta_j r_j \quad (1)$$

where

- k is the discrete time index,
- j is the number of the local model,
- $y(k)$ is the process output signal (controlled variable, CV),
- $u(k)$ is the process input signal (manipulated variable, MV),
- $v(k)$ is the (optional) measured disturbance signal,
- r_j is a constant that defines the operating point in the j -th fuzzy domain (it compensates for the nonlinearity in the static characteristics),
- a_{1j} , a_{2j} , b_{1j} , b_{2j} , and c_{1j} are plant parameters in the j -th operating point,
- d_{uj} and d_{vj} are delays in the MV–CV and MD–CV paths, respectively,
- $\beta_j = \beta_j(\sigma)$ —the degree of fulfilment of the j -th membership function (it is a function of the scheduling variable σ).

The triangular membership functions (see Fig. 2) are used in the approach. They are chosen so that for each σ , $\sum_{j=1}^m \beta_j(\sigma) = 1$ and therefore form the fuzzy partition over the whole interval of the scheduling variable σ . Thus, the calculations are simplified with respect to the computational burden.

The scheduling variable is calculated in each time instant k as follows:

$$\sigma(k) = k_w w(k) + k_y y(k) + k_u u(k-1) + k_v v(k) \quad (2)$$

Signal $w(k)$ represents the set-point. Constants k_w , k_y , k_u , and k_v are chosen by the designer. The following options for the scheduling variable are most usual: $y(k)$, $v(k)$, $u(k-1)$, and a

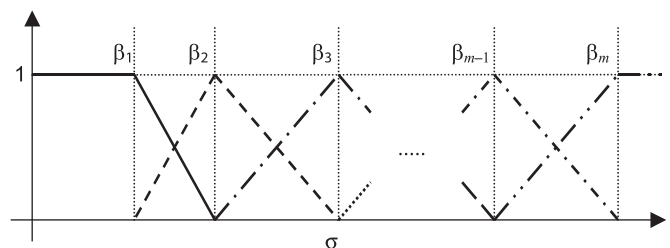


Fig. 2. Fuzzy membership functions of local models in the MFM.

linear combination of $y(k)$ and $w(k)$. The system does not have the ability to choose the scheduling variable automatically.

Note that only one parameter is identified in the numerator of the transfer functions in the MD–CV path. Some tests were carried out with two identified parameters in the numerator, but then several problems were encountered: the identifiability of the parameters was lower, the identified transfer functions were often of nonminimum phase, etc. It has to be emphasised that v is a measured disturbance that we cannot influence and it often has a low level of excitation. Since our wish was to design a very robust algorithm this simplification was introduced, even though that the class of the possible disturbance models was reduced.

During regular closed-loop operation the RTM gathers information about the controlled process. This information may be required to improve the plant model. It is very likely that the system is started with limited knowledge about the controlled plant. In order to improve the performance of the system it is necessary to obtain a better model of the plant. These tasks are performed by the OLA.

The online learning agent is a module that performs a structural and parametric identification of the plant online. As it is quite complex it is divided into smaller submodules:

- OLA main unit—it performs the parameter estimation; its inputs are the structure of the model (order and dead time of the plant, the position of membership functions); it outputs the set of identified parameters \mathbf{M} ;
- OLA verification unit—it immediately follows the OLA main unit and performs a verification of the parameters obtained in the main unit; on the basis of the parameter set \mathbf{M} the confidence index of the model is calculated—the latter is a measure for the quality of the model;
- Excitation unit—this is a very important unit that supervises the excitation; if the excitation is not sufficient, the estimation is disabled, otherwise it is enabled;
- Membership-functions unit—the module is used to determine if another fuzzy domain should be added to improve the overall model;
- Dead-time unit—the block is used to determine the dead time of the plant.

The OLA is invoked periodically or upon demand by the OS. Since it is computationally intensive, it runs as a low-priority task. It comprises the following steps:

Excitation check: If the variance of the signals $w(k)$, $y(k)$, $u(k)$, and $v(k)$ in the active buffer is below their specified thresholds, the execution is cancelled. Obviously, the variance is not the relevant measure of the excitation. The proper way would be to test the frequency content of the signals in the buffer. Since we were limited with the equipment, the choice of variance seemed to be an acceptable measure for testing the excitation.

Copy active MFM from MIA: The current model in use that resides in the database is copied into the working memory.

Select excited local models: If the average membership function fulfilment of the signals in the active buffer $\beta_j(\sigma(k))$ exceeds a certain threshold p , this local model is selected. This means that only fuzzy domains that have a relatively high excitation are considered. Further processing does not include other local models.

Identification: The parameters of the selected local models are identified using the novel fuzzy instrumental variables (FIV) identification method, an extension of the linear instrumental variables identification procedure (Ljung, 1987) for the specified MFM. It will be described in Section 4.

Verification/validation: In this step the comparison of the old and the new models is performed, and each model is given a certain confidence index that serves as a basis for the decision whether to replace the old model by the new one. More details will be given in Section 4.

4. Identification algorithm

The identification is batch-wise, i.e., signal buffers of a certain size, denoted by N (signal indexes run from 0 to $N - 1$), are analysed to obtain the plant parameters. The identification is performed in each sufficiently excited fuzzy domain. Index j denotes that the j -th fuzzy domain is taken into consideration. $\hat{\theta}_j$ is a vector of the estimates of the plant parameters $[\hat{a}_{1j}, \hat{a}_{2j}, \hat{b}_{1j}, \hat{b}_{2j}, \hat{c}_{1j}]^T$.

4.1. Fuzzy least squares (FLS) algorithm

In the first step the recursive version of the least squares method is used in a fuzzy form:

$$\begin{aligned}\hat{\theta}_j(k+1) &= \hat{\theta}_j(k) + \mathbf{P}_j(k+1)\psi_j(k+1)e(k+1) \\ e(k+1) &= D(\beta_j y(k+1) - \psi_j^T(k+1)\hat{\theta}_j(k)) \\ \mathbf{P}_j(k+1) &= \mathbf{P}_j(k) - \frac{\mathbf{P}_j(k)\psi_j(k+1)\psi_j^T(k+1)\mathbf{P}_j(k)}{1 + \psi_j^T(k+1)\mathbf{P}_j(k)\psi_j(k+1)}\end{aligned}\quad (3)$$

where

- $\psi_j(k+1)$ is the vector of measurements:

$$\psi_j(k+1) = [\beta_j[-y(k), -y(k-1), u(k-d_{uj}), u(k-1-d_{uj}), v(k-d_{vj})]^T]\quad (4)$$

- $D(\cdot)$ is the dead-zone operator with parameter d_{dead} :

$$D(x) = \begin{cases} x, & |x| > d_{dead} \\ 0, & |x| \leq d_{dead} \end{cases}\quad (5)$$

Obviously, in order to construct the vectors of measurements $\psi_j(k+1)$ from the given signal buffer, the starting index for k is $k_0 = \max(d_{uj} + 1, d_{vj})$, and k runs on the interval $k = k_0, \dots, N - 1$ in Eq. (3). The algorithm has to be initialised. This is done by copying the active model in use from MIA ($\hat{\theta}_{MIAj}$) to $\hat{\theta}_j(k_0)$ and selecting a large initial covariance matrix $\mathbf{P}(k_0) = 10^5 \mathbf{I}$. The final estimate of this algorithm $\hat{\theta}_j(N)$ is denoted by $\hat{\theta}_{FLSj}$.

4.2. Fuzzy instrumental variables algorithm

In the second step the “fuzzy instrumental variables” algorithm is used:

$$\begin{aligned}\hat{\theta}_j(k+1) &= \hat{\theta}_j(k) + \mathbf{P}_j(k+1)\chi_j(k+1)e(k+1) \\ e(k+1) &= D(\beta_j y(k+1) - \psi_j^T(k+1)\hat{\theta}_j(k)) \\ \mathbf{P}_j(k+1) &= \mathbf{P}_j(k) - \frac{\mathbf{P}_j(k)\chi_j(k+1)\psi_j^T(k+1)\mathbf{P}_j(k)}{1 + \psi_j^T(k+1)\mathbf{P}_j(k)\chi_j(k+1)} \\ k &= k_0, \dots, N - 1\end{aligned}\quad (6)$$

with the instrumental variables $\chi_j(k+1)$ defined as

$$\chi_j(k+1) = [\beta_j[-\hat{y}(k), -\hat{y}(k-1), u(k-d_{uj}), u(k-1-d_{uj}), v(k-d_{vj})]^T]$$

and \hat{y} is the simulated output:

$$\hat{y}(k) = \sum_{j=1}^m \psi_j^T(k)\hat{\theta}_j(k-1)\quad (7)$$

The FIV algorithm (6) is initialised by $\hat{\theta}_{FLSj}$ and the covariance matrix obtained in the last calculation of the FLS algorithm. The last estimate $\hat{\theta}_j(N)$ of the FIV algorithm (6) is denoted by $\hat{\theta}_{FIVj}$.

Remark 1. The dead zone (5) is included to prevent any drift of the parameters due to noise. This is a known solution in adaptive control (Peterson and Narendra, 1982) and is motivated by the fact that the dominant part in the small error $e(k)$ might be and often is due to the noise that would lead to a wrong correction of the parameters if the adaptation was not switched off. The logical consequence is that the optimal parameters are not obtained, but it is more important that the robustness is improved and the reliable operation of the system is obtained.

Remark 2. Note that a recursive algorithm is used that does not include matrix inversion, which is difficult to realise on a simple platform. The FLS algorithm is initialised by the model from MIA, while the FIV algorithm starts with the model obtained in the first (FLS) phase.

Remark 3. It is worth pointing out that the parameters of the model are calculated independently in each operating point in Eqs. (3) and (6). Each sample contributes to the estimated parameters of the fuzzy domain with a weight that is precisely equal to the fulfilment of the membership function of the corresponding domain, which makes this method very similar to weighted least squares. Such an approach can be seen as a “local learning”. There are several advantages to using a local approach in our context:

- transparent relation to linear techniques,
- moderate computational demand,
- control over the data selection (achieved by planning the experiments),
- possibility to modify a specific part of the model,
- less prone to the problems of ill-conditioning due to over-parametrisation and local minima; if all parameters were calculated at once, the information matrix would not be of full rank and even though the matrix inversion was not involved, the results would drift in the directions where there was no excitation; in other words, most often all system parameters are not identifiable as a single vector of unknown parameters.

Our method may be suboptimal, but it is extremely computationally efficient (suitable for PLCs and open controllers) and reliable in the initial phase of parameter estimation (far away from the optimum). Reliable performance far away from the optimum is extremely important during plant configuration, when some of the local models have not yet been tuned. “Global learning” methods typically do not consider this at all. There are also drawbacks. Most notably, uneven excitation in one fuzzy domain may result in model bias. But due to the use of global verification (which will be explained in the following), this cannot lead to model degradation.

Remark 4. Obviously, the procedure is slightly modified if the order of the plant is 1 instead of 2. The parameters a_2 and b_2 are 0 in that case and are not estimated. The whole procedure is modified in a straightforward manner.

Remark 5. In the case of a lack of excitation in the model branch from v (MD) to y (CV) (or when MD is not present at all) or the branch from u (MV) to y (CV), variants of the described method are used where the parameters connected with the unexcited path are not identified. Rather, they are calculated so that the same input–output gain is obtained.

and strategies. The main part of the plant consists of two pumps (denoted by PO1 and PO2 in the scheme in Fig. 3) that together with two controlled valves (V11 and V12) influence the water inflow into two tanks. The water levels in the tanks (LT1 and LT2) are measured and are usually the controlled variables. In our set up only half of the device was used, i.e., only one tank, one pump, and one control valve were employed. The manipulated variable in our case was the voltage applied to the pump, while the controlled variable was the water level in the tank. The analysis shows that the controlled plant is quite nonlinear and the use of the nonlinear controller is justified.

The time constants of the plant are very long and the experiments are lengthy. The RTM operated in the closed-loop setting for a few hours, while the reference signal was changing across the wide operating band. The signals obtained in the experiment are shown in Fig. 4. The RTM was initialised by a very

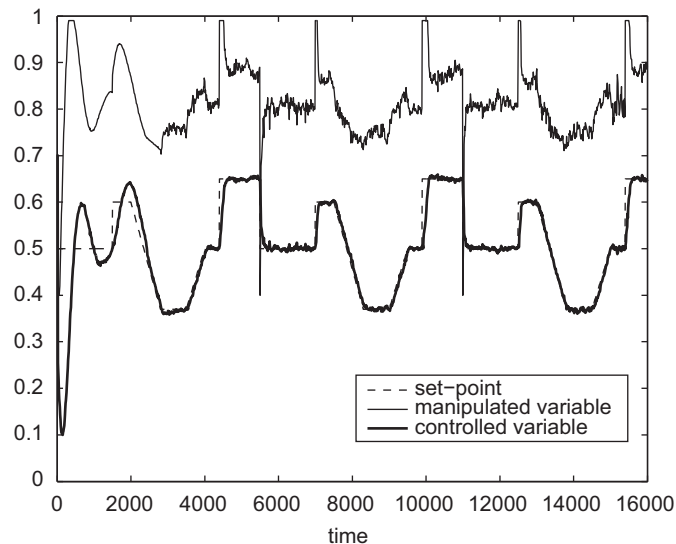


Fig. 4. Signals obtained during a real-time experiment on the hydraulic pilot plant.

conservatively tuned (linear) PI controller, i.e., the system was operating in safe mode. This happens when there is not enough information available about the controlled plant. The system does not operate optimally during the safe mode, which is clearly seen from Fig. 4 (the system operates in safe mode from the beginning until the time of 2810 s). At the time of 2810 s the OLA module was triggered for the first time (this happens when the signals for identification are appropriate). Fig. 5 shows the signals that the OLA module was using to obtain the first fuzzy model (three fuzzy sets were used with membership functions having peaks at 0.43, 0.5, and 0.57). In the first call of the OLA module only the fuzzy sets 0.5 and 0.57 had sufficient excitation. The upper part of Fig. 5 shows the responses of the real plant and of those linear models that were corrected. The old model (denoted by MIA) is a linear one, which means that all linear submodels are equal. Consequently, the responses of all submodels coincide, which can be seen from Fig. 5. The OLA module produced two affine submodels of the Takagi–Sugeno model (denoted by OLA 0.5 and OLA 0.57). The responses of these two models practically coincide with the response of the plant. The lower part of Fig. 5 shows the responses of the plant and both fuzzy models (denoted by MIA and OLA). The OLA fuzzy model is better than the MIA fuzzy model although it is not a good approximation to the real system. This is due to the fact that the model 0.43 is still a default model, which is far from the optimum one. But the new OLA model is very good in the other two fuzzy domains, and based on the model in these two domains two PID controllers were designed that are blended together as a fuzzy gain-scheduling controller. At the time of 3260 s the module was called again, all three models were estimated again, and the system decided to keep the newly obtained models 0.43 and 0.57. The responses of the affine models are shown in the upper part of Fig. 6, and the responses of the fuzzy models in the lower part of Fig. 6. It can be seen that the complete fuzzy model that now includes the models 0.43, 0.5, and 0.57 shows very good agreement with the actual plant. After the second call, the OLA module was called repeatedly, and only in a few instances was the new model kept (remember that the new model has to be much better in order to be accepted by the supervisory mechanism). The last correction was made at the time

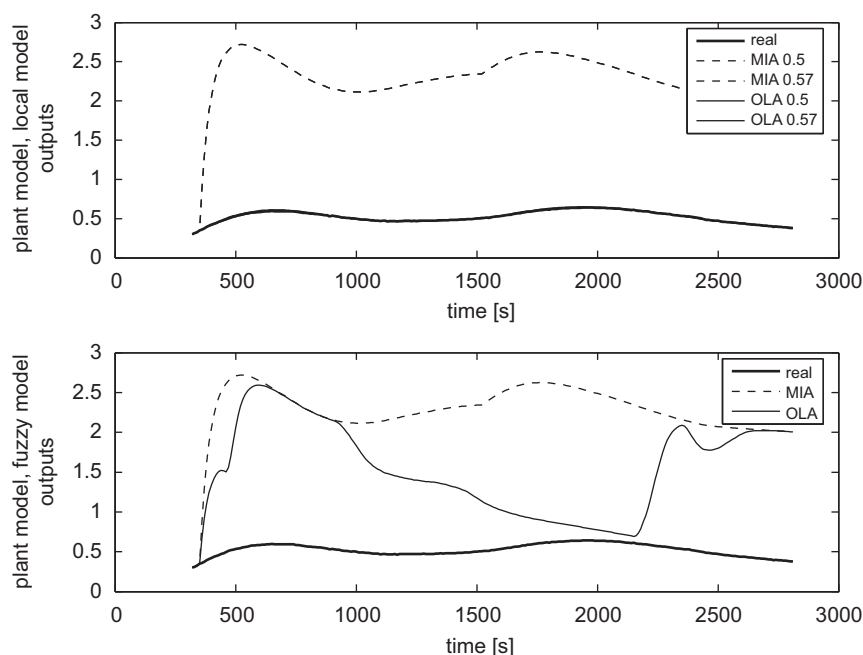


Fig. 5. Signals processed during the first call of the OLA module.

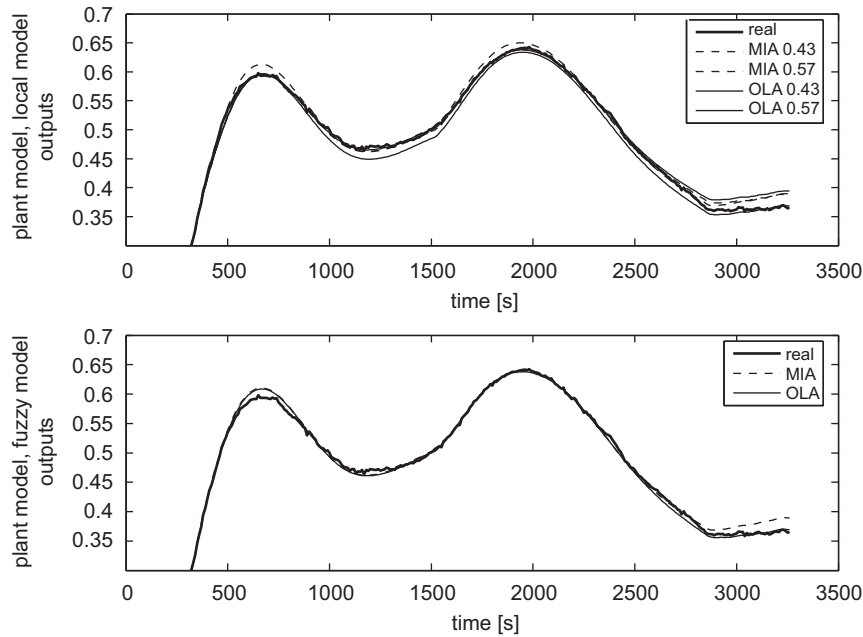


Fig. 6. Signals processed during the second call of the OLA module.

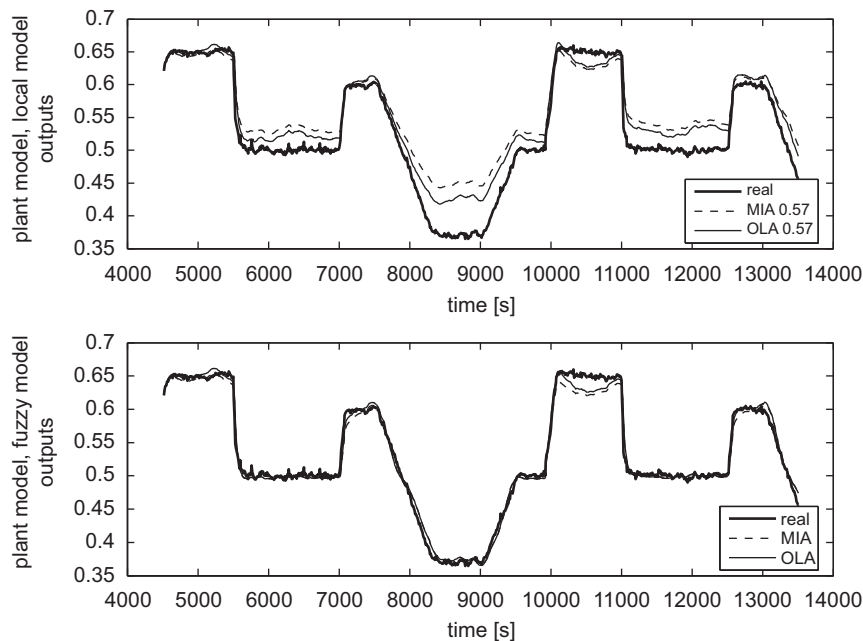


Fig. 7. Signals processed during the last call of the OLA module that resulted in a modified model of the plant.

of 13 510 s. The signals that the OLA module was processing in this call are shown in Fig. 7. Note that the responses of the affine submodels coincide with the real plant only in the area where the corresponding membership function has a large degree of fulfilment, while the response of the complete fuzzy model shows a high degree of agreement with the response of the plant. This is an obvious consequence of the nonlinear character of the plant. The system continued running in the closed loop after the time of 16 000 s, the OLA module was triggered several times more, but since the plant is more or less time invariant, no new model was found that could be regarded by the supervisory mechanism as a better one. All changes of the model parameters are collected in Table 1.

6. Robustness tests with the RTM

The next tests concerned the robustness of the RTM. In this experiment the robustness was not analysed formally. Rather, simulation tests were performed. The latter were chosen since it is much easier to influence the environment, and especially the disturbances and noises in the simulation experiments. The operation of the controller was tested on the simulated neutralisation process depicted in Fig. 8, described in detail by Henson and Seborg (1994). An acid stream Q_1 , a buffer stream Q_2 and a base stream Q_3 are mixed in a tank. The acid and base streams are equipped with flow control valves. The pH of the mixture is measured with a sensor that is located downstream. The effluent

Table 1
Changes of the model parameters.

Time	$\hat{a}_{1,1}$	$\hat{b}_{1,1}$	\hat{r}_1	$\hat{a}_{1,2}$	$\hat{b}_{1,2}$	\hat{r}_2	$\hat{a}_{1,3}$	$\hat{b}_{1,3}$	\hat{r}_3
2810	–	–	–	–0.9677	0.0592	–0.0319	–0.9678	0.0542	–0.0279
3260	–0.9762	0.0625	–0.0384	–	–	–	–0.9674	0.0557	–0.0290
4360	–0.9716	0.0622	–0.0364	–	–	–	–	–	–
5510	–	–	–	–	–	–	–0.9628	0.0550	–0.0247
9760	–0.9681	0.0655	–0.0373	–	–	–	–	–	–
12 760	–	–	–	–0.9642	0.0573	–0.0287	–	–	–
13 510	–	–	–	–	–	–	–0.9602	0.0648	–0.0315

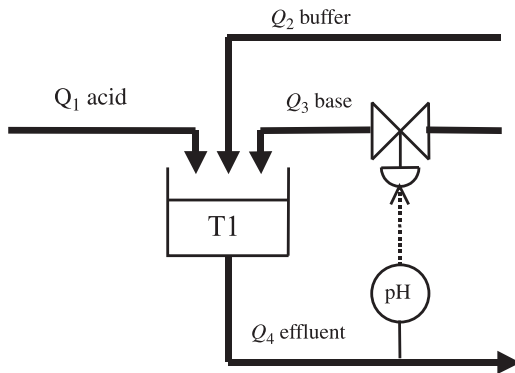


Fig. 8. Sample neutralisation process.

pH is the controlled variable y , and the manipulated variable u is the flow of the base stream Q_3 . The static curve $u-y$ is highly nonlinear and its open-loop gain changes by a factor of 8, which makes this plant very difficult to control with a conventional PID controller.

6.1. Behaviour of the RTM in the environment with measurable disturbance

The disturbance in the system is a possible additional burden on the control system. As already mentioned, the OLA has a very complex identification algorithm built in. In addition to having the possibility to identify nonlinear (Takagi–Sugeno modelled) systems, systems with changing delays, membership functions and different combinations of those mentioned, it also provides the possibility to estimate the disturbance model when the latter is measured. From the system theory aspect, the plant can be seen as a two-input single-output plant (TISO). It is generally known that such plants are considerably more difficult to identify than single-input single-output (SISO) plants, especially in the case when one of the inputs (disturbance in our case) cannot be influenced. It is obvious that in the case of a constant disturbance no information on the disturbance model can be extracted. When the level of excitation of the disturbance is high enough, a relatively good disturbance model can be obtained.

The experiment was conducted to test the ability of the system to identify the disturbance model. The experiment was conceived to simulate the batch-wise operation of the pH process. The reference signal changed according to the predefined periodic signal. Since the system was operating in a closed loop during the experiment additional troubles can be expected. The disturbance was constant most of the time. There were, however, some step-like changes of relatively high amplitude. The quality of the signals used for the identification is very low since they were obtained in the closed-loop operation from the TISO system. As expected, the responses show that in certain operating regions the system starts to behave undesirable. Oscillations of the manipu-

lated and controlled variables can be seen. It has to be stressed that the problems were not encountered in all operating points nor were they fatal for the performance of the system. The encouraging fact is that the desired behaviour of the system restored after the changes in the disturbance had stopped (actually, some time had passed before the “inconstant” disturbance left the identification window). (Note also that the control performance monitor is included in the controller. If bad control performance is detected, system automatically switches to the safe mode.)

In conclusion, it can be said that the option of identifying both models (the control model and the disturbance model) can be used, but one has to be aware of the fact that the behaviour might not always be as expected. This is especially true if the system is highly nonlinear, possesses a lot of noise or a lot of the optional components in the OLA are enabled. The more possibilities are enabled in the RTM, the less robust the system will be in general.

6.2. Behaviour of the RTM in the noisy environment when operating in a closed loop

Another difficulty in the identification is the presence of noise (or immeasurable disturbance) in the plant. Because of the closed loop the noise propagates to the manipulated variable causing a correlation between the latter and the controlled variable. That problem is solved to some extent by incorporating the instrumental variables (FIV) into the identification procedure. Nevertheless, the problem of noise in the identification in the closed loop is not completely circumvented by the FIV alone. Additional steps have to be taken to suppress the influence of noise. In the design phase the dead zone was included into the identification procedure. This prevents the regressors that do not carry any new information about the plant from taking part in the parameter calculation. The rationale behind this idea is that the main component of such regressors is probably noise that would lead the estimated parameters in the wrong direction.

A similar experiment to that mentioned before (Section 6.1) was conducted. The disturbance was not measured in this case, but the noise is present at the plant input and the plant output (zero-mean white noise with standard deviations 0.02 and 0.01, respectively). The input noise only affects the plant simulation while the controller and the identification algorithms operate with the noise-free variable. Two identification algorithms were compared: the one that selects the model with the lowest mean square error and the other that penalises the mean square errors obtained by the FLS and FIV algorithms. In other words, in the first experiment the supervisory mechanism, explained in Section 4.4, was turned off, while in the second experiment it was turned on ($C_{sup} = 1.1$).

The courses of the identified parameters are shown in Figs. 9, 10 and 11 for the first, the second, and the third fuzzy domain, respectively. The parameters r_j ($j = 1, 2, 3$) are not shown to improve legibility of the plots (they are more or less constant

throughout the experiment; they also do not influence the controller parameters). It can be noticed that the parameters remain almost constant if the supervisory mechanism is turned on (numbers of changes in the respective fuzzy domains: 2, 2, 3), while they change very often if the supervisory mechanism is turned off (numbers of changes in the respective fuzzy domains: 40, 26, 67). It can also be seen that in the latter case after these drift periods the parameters return back to the values around the ones obtained by the supervised identification.

Our wish was to confirm the idea of the supervisory mechanism. Indeed, it turned out that the quality of the model can decrease with time in the absence of the supervisory mechanism. The deterioration was not instantaneous. Rather, the drift in the parameters can be observed. Such a drift would not be possible if the identification was performed in an open loop. When the system operates in the closed loop and the noise is present, the scenario is as follows. The system starts with no information about the plant at all and with a very conservatively tuned controller. The bandwidth of the system is therefore low

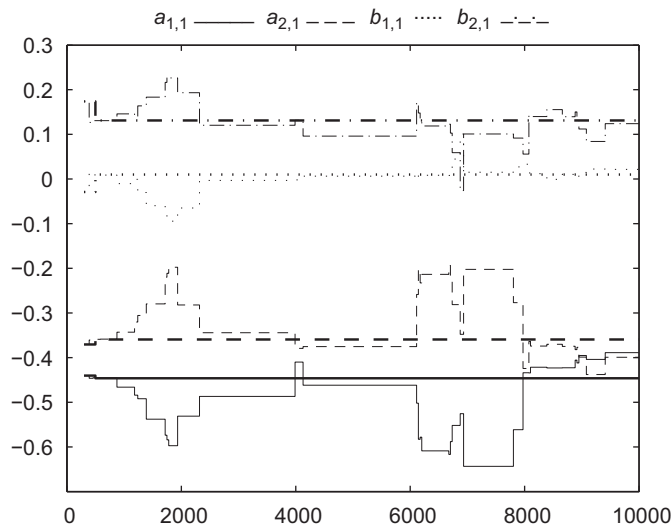


Fig. 9. Time plots of the identified parameters $\hat{a}_{1,1}$, $\hat{a}_{2,1}$, $\hat{b}_{1,1}$, $\hat{b}_{2,1}$ (thick line—supervisory mechanism is on, thin line—supervisory mechanism is off).

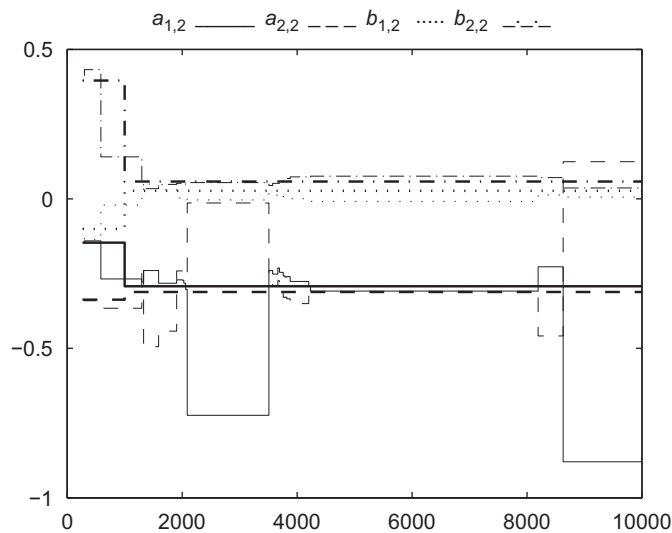


Fig. 10. Time plots of the identified parameters $\hat{a}_{1,2}$, $\hat{a}_{2,2}$, $\hat{b}_{1,2}$, $\hat{b}_{2,2}$ (thick line—supervisory mechanism is on, thin line—supervisory mechanism is off).

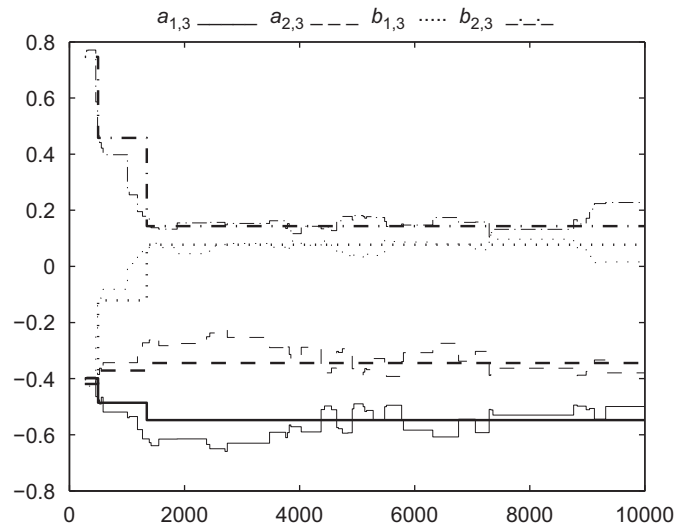


Fig. 11. Time plots of the identified parameters $\hat{a}_{1,3}$, $\hat{a}_{2,3}$, $\hat{b}_{1,3}$, $\hat{b}_{2,3}$ (thick line—supervisory mechanism is on, thin line—supervisory mechanism is off).

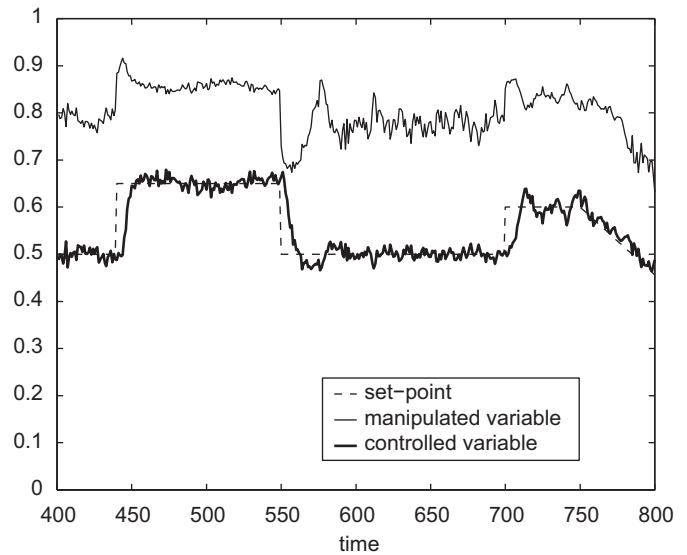


Fig. 12. Performance of the system in the beginning (without the supervisory mechanism).

and the manipulated variable is relatively slow (almost open loop). This is why a good model is obtained in the beginning (see Fig. 12). The procedure of controller tuning is such that it results in a high-gain controller. This is why the manipulated variable is much more dynamic and also correlated with the controlled variable. The obtained model is worse because of that. The next controller results in a system that oscillates even more. Consequently, the quality of the signals used for the identification is very low (more or less only one frequency is present). This cycle leads to a deterioration of the performance. In Fig. 13 the signals in the system are depicted after a period of time when the system is run with the OLA active. The biggest oscillations occur when the system is in the first fuzzy domain. Drastic changes of the identified parameters can be seen in Fig. 9 around the time interval from Fig. 13. After a certain period the identified parameters recover from the drift caused by the noise, thus resulting in a better control performance. This undesirable bursting phenomena can be avoided by turning on the supervisory mechanism.

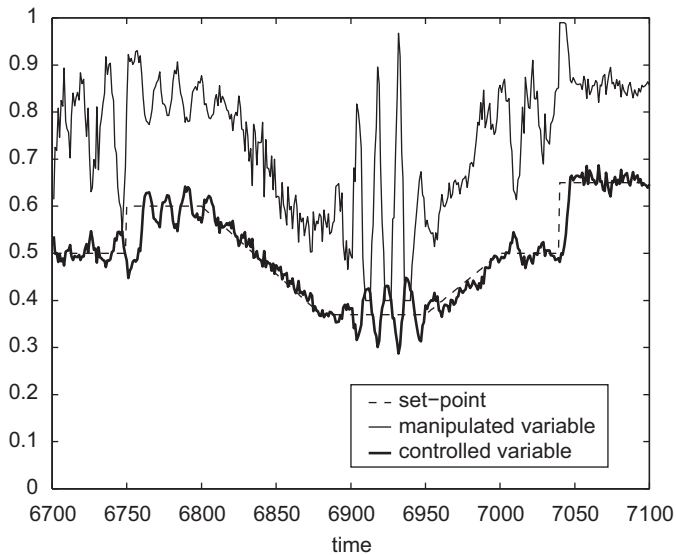


Fig. 13. Performance of the system after some time (without the supervisory mechanism).

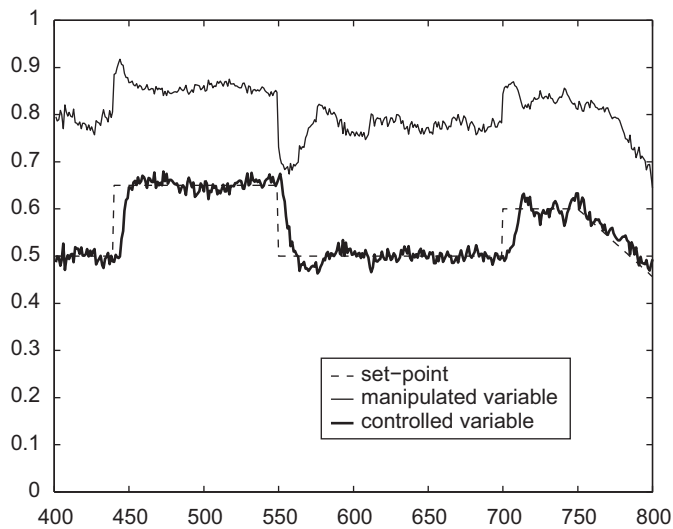


Fig. 14. Performance of the system in the beginning (with the supervisory mechanism).

Figs. 14 and 15 represent the behaviour of the system in the same time intervals as Figs. 12 and 13 in the previous experiment, but with the supervisory mechanism active. The results of the experiment show that the gradual deterioration of the performance is prevented by turning on the supervisory mechanism. However, this solution is not absolute. It implicitly prevents small changes in the model. Consequently, it is hard for the algorithm to reach a global optimum. As always, a tradeoff between the performance and the robustness is performed. In this case, our standpoint is that robustness is more important than optimum performance.

6.3. Testing of the dead-time unit and the membership-functions unit

Some experiments were also made with the dead-time unit and the membership-functions unit. Both of them are called periodically (after a certain number of parameter identifications) if enabled. Neither of them are very robust and they demand

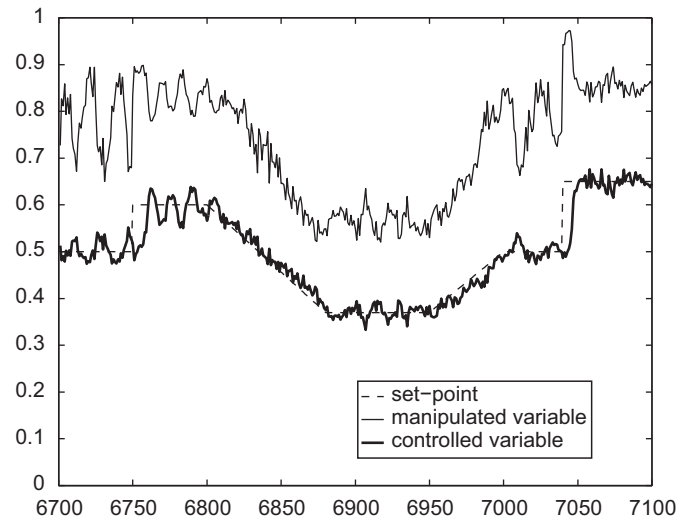


Fig. 15. Performance of the system after some time (with the supervisory mechanism).

signals of high quality (high level of information). The dead-time unit tries to fit the drastic changes in the output by changing the delay in the model. It is highly advisable to enable it only in open loop—usually this is done when an open-loop experiment is being conducted and there are some step-like changes in the system input. The membership-functions unit will add a new fuzzy domain if the mean square error of the new model (with one fuzzy domain more) is much better (for a certain multiplicative constant) than the original model.

7. Conclusion

An advanced self-tuning nonlinear controller has been successfully implemented on an industrial PLC platform. Several pilot applications, including the one presented in this paper, have also been successfully completed. Compared to the industry standard PID controller, an expected considerable improvement in the control performance was achieved using the on-line identification of the Takagi–Sugeno model and tuning of the Takagi–Sugeno PID-type controller.

The identification algorithm is based on the Takagi–Sugeno least-squares approach where several details are included to make the algorithm more robust: the identification is performed in batches (thus the excitation can be measured and the parameter changes can be disabled if the excitation is not sufficient), the instrumental variables are used instead of the ordinary least squares, dead zone is included to prevent parameter drift. The last and the most important technique is the use of the supervisor that performs the verification/validation of the new model and the old one. The new model is favored only if it is much better than the old one. This technique is also analysed in the simulation case study. The results show that the inclusion of the supervisor improves robustness of the system considerably. Consequently, also the control performance is better.

Acknowledgements

The ASPECT project was co-sponsored by the European Commission, as part of its 5th Framework Program, and the participating companies INEA d.o.o. (Slovenia), Start Engineering JSCo (Bulgaria), and INDELEC Europe S.A. (Greece). The authors would like to acknowledge the contributions of all the other

project team members from Jožef Stefan Institute, University of Ljubljana, INEA d.o.o., UCTM Sofia, Start Engineering JSco, Computer Technology Institute, and INDELEC Europe S.A. Patent pending PCT/SI02/00029.

References

- Blažič, S., Škrjanc, I., Matko, D., 2003. Globally stable direct fuzzy model reference adaptive control. *Fuzzy Sets and Systems* 139 (1), 3–33.
- Fortescue, T.R., Kershenbaum, L.S., Ydstie, B.E., 1981. Implementation of self-tuning regulators with variable forgetting factors. *Automatica* 17 (6), 831–835.
- Henson, M.A., Seborg, D.E., 1994. Adaptive nonlinear control of a pH neutralization process. *IEEE Transactions on Control Systems Technology* 2 (3), 169–182.
- Hilhorst, R.A., van Amerongen, J., Lohnberg, P., Tulleken, H.J.A.F., 1994. A supervisor for control of mode-switch processes. *Automatica* 30 (8), 1319–1331.
- Ioannou, P.A., Sun, J., 1996. *Robust Adaptive Control*. Prentice-Hall PTR, Englewood Cliffs, NJ.
- Layne, J.R., Passino, K.M., 1993. Fuzzy model reference learning control for cargo ship steering. *IEEE Control Systems Magazine* 13 (6), 23–34.
- Ljung, L., 1987. *System Identification*. Prentice-Hall, Englewood Cliffs, NJ.
- Narendra, K.S., Balakrishnan, J., Ciliz, M.K., 1995. Adaptation and learning using multiple models, switching, and tuning. *IEEE Control Systems Magazine* 15 (3), 37–51.
- Peterson, B.B., Narendra, K.S., 1982. Bounded error adaptive control. *IEEE Transactions on Automatic Control* Ac-27 (6), 1161–1168.
- Precup, R.E., Doboli, S., Preitl, S., 2000. Stability analysis and development of a class of fuzzy control systems. *Engineering Applications of Artificial Intelligence* 13 (3), 237–247.
- Precup, R.E., Preitl, S., Korondi, P., 2007. Fuzzy controllers with maximum sensitivity for servosystems. *IEEE Transactions on Industrial Electronics* 54 (3), 1298–1310.
- Škrjanc, I., Matko, D., 2000. Predictive functional control based on fuzzy model for heat-exchanger pilot plant. *IEEE Transactions on Fuzzy Systems* 8 (6), 705–712.
- Škrjanc, I., Kavšek-Biasizzo, K., Matko, D., 1997. Real-time fuzzy adaptive control. *Engineering Applications of Artificial Intelligence* 10 (1), 53–61.
- Sugeno, M., Nishida, M., 1985. Fuzzy control of model car. *Fuzzy Sets and Systems* 16 (2), 103–113.
- Takagi, T., Sugeno, M., 1985. Fuzzy identification of systems and its applications to modelling and control. *IEEE Transactions on Systems, Man, and Cybernetics* SMC-15 (1), 116–132.
- Takatsu, H., Itoh, T., Araki, M., 1998. Future needs for the control theory in industries—report and topics of the control technology survey in Japanese industry. *Journal of Process Control* 8 (5–6), 369–374.
- Tang, Y., Zhang, N., Li, Y., 1999. Stable fuzzy adaptive control for a class of nonlinear systems. *Fuzzy Sets and Systems* 104 (2), 279–288.
- Vidyasagar, M., 1993. *Nonlinear Systems Analysis*, second ed. Prentice-Hall, Englewood Cliffs, NJ.
- Vrančić, D., Strmčnik, S., Juričić, D., 2001. A magnitude optimum multiple integration tuning method for filtered PID controller. *Automatica* 37, 1473–1479.
- Wang, L.X., Mendel, J.M., 1992. Fuzzy basis functions, universal approximation, and orthogonal least-squares learning. *IEEE Transactions on Neural Networks* 3 (5), 807–814.