

# Simple Fuzzy Identification Implemented in Advanced Controller

UDK 004.896  
IFAC 2.5

Original scientific paper

The paper focuses on identification issues of the advanced controller ASPECT\* that is implemented on a simple PLC platform with an extra mathematical coprocessor and is intended for the advanced control of complex plants. The model of the controlled plant is obtained by means of experimental modelling using an online learning procedure that combines model identification with pre- and post-identification steps that provide reliable operation. It is shown that acceptable performance of the system is obtained despite difficult conditions that may arise during operation.

**Key words:** programmable logic controllers, fuzzy modelling, identification, nonlinear control, pH control

## 1 INTRODUCTION

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

One possibility is to apply adaptive control that was accommodated to treat not only linear time invariant plants but also nonlinear and time variant plants. Adaptive control is avoided in the practice due to the lack of robustness even though robust adaptive control was proposed to overcome this drawback [3]. Many successful applications of fuzzy controllers [6] have shown their ability to control nonlinear plants. Fuzzy controllers were later upgraded with the ability of constructing fuzzy model of the plant on-line and adjusting control parameters accordingly. The universal approximation theorem [8] provided a theoretical background for new fuzzy direct and indirect adaptive controllers [1] whose stability was proven using the Lyapunov theory.

The main drawback of most of the existing approaches for the control of nonlinear plants is that they are very complex, difficult to understand since they demand fairly good knowledge of mathematics and are thus avoided by practising engineers. Since our wish was to implement the controller on a simple platform, e.g. programmable logic controller, the algorithm should be kept as simple as possible. On the other hand, one of the most important goals was to choose the algorithm that would be

simple to tune, i.e. to select the necessary design parameters. In our case, Takagi-Sugeno fuzzy model of low order [7] was chosen. The model is obtained via experimental modelling using a special batch-wise on-line learning procedure combining model identification with pre- and post-identification steps providing reliable operation. Many different controller types are possible to use with this model, e.g. PID, predictive etc. The system was designed such that we are not confined to a certain controller type. Rather, many different controllers can be incorporated.

Within the ASPECT project, an efficient and user-friendly engineering tool for control of a certain practically very important class of non-linear plants was built. This paper focuses on model identification issues but will also depict some of the properties of the product.

In Section 2 the general overview of the controller will be given, Section 3 focuses on the module that is responsible for on-line identification (it is referred to as the on-line learning agent – OLA). Section 4 depicts the identification algorithm built in, while the results of the simulation tests are presented in Section 5. Section 6 gives some conclusions of the paper.

## 2 CONTROLLER OVERVIEW

Modular multi-agent structure of the controller enables a choice of several control algorithms suitable for different processes. The parameters of the

\* The ASPECT project was co-sponsored by the European Commission within its 5th Framework Program and the participating companies INEA d.o.o., Start Engineering JSCo, and INDELEC Europe S.A.

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 controller code is subdivided into the Run-Time Module (RTM), running on a PLC and performing all activities mentioned above, and the Configuration Tool (CT) that simplifies the initial configuration from a personal computer, providing guidance through the configuration procedure.

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 includes the following agents, as shown in Figure 1:

- Signal Preprocessing Agent (SPA) – provides the signals to the other agents,
- Online Learning Agent (OLA) – identifies the model of the plant (most of the paper will be devoted to this module),
- Model Information Agent (MIA) – maintains the active model (model-in-use) and its status information – the OLA module provides new models, but MIA module decides whether to accept them or not,
- Control Algorithm Agent (CAA) – includes functionality of an advanced industrial non-linear control algorithm and automatic tuning of its parameters from the model,

- 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.

### 3 ON-LINE LEARNING AGENT (OLA)

The controller is model-based, founded on a multi-faceted model (MFM) that includes several model forms required by the online-learning mechanism and control algorithm agents. Takagi-Sugeno model of the plant was chosen with  $m$  fuzzy domains (maximal number is limited to 10 at the moment). In each fuzzy domain a local linear model is used. First and second order local models are considered. This means that identification of both models is performed and better model is used if not explicitly defined otherwise. The second order model is more general and will be described here. It takes the following form:

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

where  $k$  is the discrete time index,  $j$  is the number of the local model,  $y(k)$  is the plant output,  $u(k)$  is the plant input,  $v(k)$  is the (optional) measured disturbance,  $r_j$  is a constant that defines the operating point in the  $j$ -th fuzzy domain (it compensates for the nonlinearity in the static curve),  $a_{1j}$ ,  $a_{2j}$ ,  $b_{1j}$ ,  $b_{2j}$ ,  $c_{1j}$  – plant parameters in the  $j$ -th operating point,  $du_j$ ,  $dv_j$  – delays in the  $u$ - $y$  and  $v$ - $y$  paths, respectively,  $\beta_j$  – the degree of fulfilment of the  $j$ -th membership function (it depends upon the scheduling variable  $\sigma$ ), and

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

where predefined coefficients  $k_w$ ,  $k_y$ ,  $k_u$ , and  $k_v$  are used, and signal  $w(k)$  here represents the set-point.

Normalised triangular membership functions (MF) are used, i.e.  $\sum_{j=1}^m \beta_j = 1$ . They are illustrated in Figure 2.

Note that only 1 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

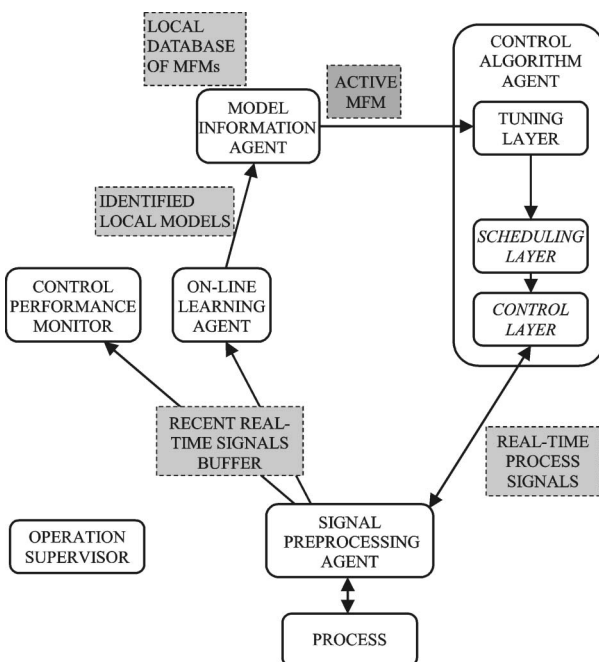


Fig. 1 Schematic representation of the controller

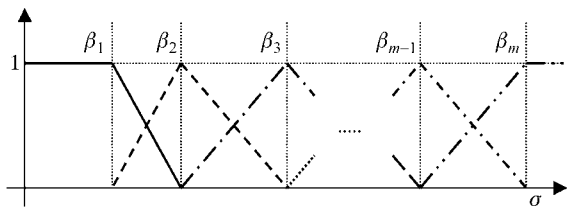


Fig. 2 Fuzzy membership functions of local models in the MFM

functions were often of non-minimum phase etc. It has to be emphasised that  $v$  is measured disturbance that we cannot influence and often has low level of excitation. Since our wish was to design very robust algorithm the model was simplified even though that it is sometimes not optimal.

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

The OLA agent is a software module that performs structural and parametric identification of the plant on-line. As it is quite complex it is divided into smaller submodules:

- OLA main unit – performs parameter estimation; its inputs are the structure of the model (order and dead time of the plant, the position of MFs) while it produces the set of identified parameters;
- OLA verification unit – performs verification of the parameters obtained in the main unit and calculates the confidence indexes that are used as model quality measures;
- excitation unit – this very important unit 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 over-all model;
- dead time unit – the block is used to determine the dead time of the plant.

The OLA agent is invoked periodically or upon demand by the OS. Since it is computationally intensive, it runs as a low-priority task. Whenever OLA is invoked, the procedure illustrated in Figure 3 is executed.

In the following the blocks in Figure 3 will be described:

- *Excitation check.* If the variance of the signals  $w(k)$ ,  $y(k)$ ,  $u(k)$ , and  $v(k)$  in the active buffer is

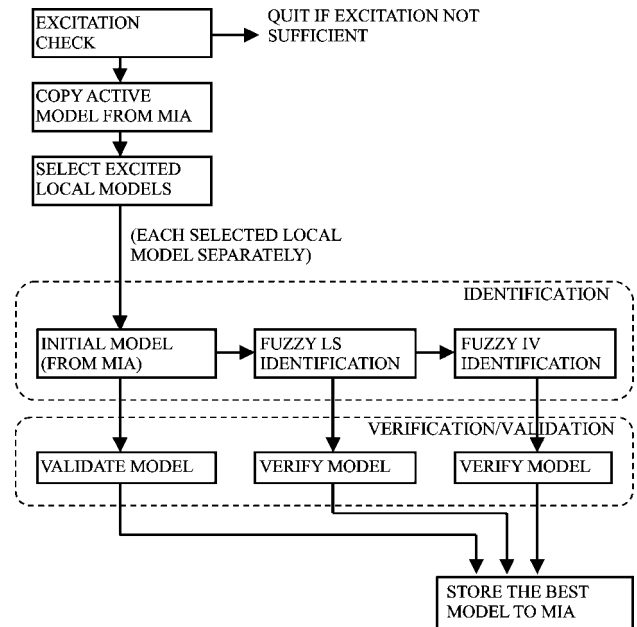


Fig. 3 The flowchart of tasks running in the OLA module

below their specified thresholds, the further estimation is cancelled.

- *Select local models.* The local models for which the sum of their corresponding membership functions  $\beta_i(s)$  over the active buffer normalised by the active buffer length exceeds certain threshold are selected. This means that only fuzzy domains that have 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 linear instrumental variables (IV) [4] for the specified MFM. It will be described in Section 4.
- *Verification/validation.* In this step the comparison of the old and new model is done 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.

#### 4 IDENTIFICATION ALGORITHM

The identification is batch-wise, i.e. signal buffers of the certain size are analysed to obtain plant parameters. The identification is performed in each sufficiently excited fuzzy domain. In general identification algorithm comprises of three phases. In the first phase fuzzy least squares algorithm is used that is initialised by the current model. The result of the first phase is used for initialising the second phase, i.e. fuzzy instrumental variables method. In the third phase the two identified models are veri-

fied and the old model is validated to come up with the model for the future use.

#### 4.1 The fuzzy least squares (FLS) algorithm

Fuzzy least squares method is used for obtaining the initial estimates of the plant parameters. Index  $j$  denotes that  $j$ -th fuzzy domain is taken into consideration.  $\hat{\theta}_j$  is a vector of the estimates of the plant parameters  $[a_{1j}, a_{2j}, b_{1j}, b_{2j}, c_{1j}]^T$ . The following algorithm is used on the signal buffer:

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

where:

– the vector of measurements  $\boldsymbol{\psi}_j(k+1)$  is

$$\boldsymbol{\psi}_j(k+1) = \beta_j \left[ -y(k), -y(k-1), u(k-du_j), \right. \\ \left. u(k-1-du_j), v(k-dv_j) \right]^T$$

– the simulated output  $\hat{y}$  is

$$\hat{y}(k+1) = \sum_{j=1}^m \boldsymbol{\psi}_j^T(k+1)\hat{\theta}_j(k) \quad (4)$$

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

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

The dead zone is included to prevent drift of parameters due to noise. This is a known solution in adaptive control [5] and is motivated by the fact that the dominant part in small error  $e(k)$  might be and often is due to the noise that would lead to wrong correction of 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.

Note that the recursive algorithm is used. One of the reasons is that matrix inversion is thus avoided since it is difficult to realise it on the simple platform. It is worth observing that the parameters of the model are calculated independently in each operating point in Equation (3). If all parameters were calculated at once, the information matrix would not be of the 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, all system parameters are not

identifiable as a single vector of unknown parameters. In Equation (3) data contribute to the estimated parameters of the fuzzy domain with the weight equal to the fuzzy membership of the corresponding domain. Thus the »fuzzy effect« is achieved. What also makes this identification nonlinear is the verification that uses nonlinear method for obtaining confidence index (the verification will be described in subsection 4.3).

#### 4.2 The fuzzy instrumental variables (FIV) algorithm

In the second step the »fuzzy instrumental variables« algorithm is used:

$$\begin{aligned}\hat{\theta}_j(k+1) &= \hat{\theta}_j(k) + \mathbf{P}_j(k+1)\boldsymbol{\chi}_j(k+1)e(k+1) \\ e(k+1) &= D\left[\beta_j y(k+1) - \boldsymbol{\psi}_j^T(k+1)\hat{\theta}_j(k)\right] \\ \mathbf{P}_j(k+1) &= \mathbf{P}_j(k) - \frac{\mathbf{P}_j(k)\boldsymbol{\chi}_j(k+1)\boldsymbol{\psi}_j^T(k+1)\mathbf{P}_j(k)}{1 + \boldsymbol{\psi}_j^T(k+1)\mathbf{P}_j(k)\boldsymbol{\chi}_j(k+1)}\end{aligned}$$

with the instrumental variables  $\boldsymbol{\chi}_j(k+1)$  defined as

$$\boldsymbol{\chi}_j(k+1) = \beta_j \left[ -y(k), -y(k-1), u(k-du_j), \right. \\ \left. u(k-1-du_j), v(k-dv_j) \right]^T.$$

The FIV algorithm is initialised with the model obtained in the first (FLS) phase.

#### 4.3 Verification/validation of the models

After the identification, a decision has to be made whether to use the old model (from MIA), the model obtained by FLS or the model obtained by FIV. The latter is sometimes very bad since the FIV algorithm does not assure that the estimates converge (when the level of noise is high and/or the initial estimate is bad).

The verification/validation is performed by simulating all three models with the actual plant input. Three simulated outputs ( $\hat{y}_{\text{MIA}}$ ,  $\hat{y}_{\text{FLS}}$ , and  $\hat{y}_{\text{FIV}}$  – obtained by Eq. 4) were used for calculating the corresponding mean square errors:

$$\begin{aligned}V_{\text{MIA}} &= \frac{1}{N} \sum_{k=0}^{N-1} (\hat{y}_{\text{MIA}}(k) - y(k))^2 \\ V_{\text{FLS}} &= \frac{1}{N} \sum_{k=0}^{N-1} (\hat{y}_{\text{FLS}}(k) - y(k))^2 \\ V_{\text{FIV}} &= \frac{1}{N} \sum_{k=0}^{N-1} (\hat{y}_{\text{FIV}}(k) - y(k))^2\end{aligned}$$

where  $N$  is the length of the signals in the buffer.

To prevent slow degradation of control performance we should be careful in selecting new model

of the plant. To prevent the drift of plant parameters, another »dead zone« is used in the identification algorithm. The actual model (that resides in MIA) is favoured if the confidence indexes of the old ( $V_{MIA}$ ) and the new model ( $V_{FIV}$  or  $V_{FLS}$ ) are similar. Only if the confidence index of the new model is much better (for a certain multiplicative constant), the model is replaced.

## 5 SIMULATION TESTS WITH THE RTM

The operation of the controller was tested on a simulated neutralisation process in Figure 4 [2]. 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 located downstream. The effluent pH is the controlled variable  $y$ , and the manipulated variable  $u$  is the flow of the base stream  $Q_3$ . The static curve  $u$ - $y$  of the plant is highly non-linear (open-loop gain of the plant changes by the factor 8). The consequence is that the conventional control (e.g. by PID controller) does not produce good results.

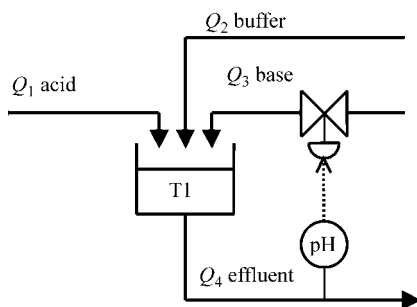


Fig. 4 The scheme of the neutralisation process

### 5.1 Behaviour of the RTM in the environment with measurable disturbance

The disturbance in the system is surely a possibility of adding extra burden on the control system. As already mentioned, OLA has got a very complex identification algorithm built in. In addition to having the possibility of identifying nonlinear (Takagi-Sugeno modelled) systems, systems with changing delays, membership functions and different combinations of the 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 two-input single-output plant (TISO). It is generally known that such plants are considerably more difficult to identify compared to 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 ex-

tracted. When the level of excitation of the disturbance is high enough, relatively good disturbance model can be obtained. But unfortunately the system would not behave very well even if the perfect disturbance model were obtained if the control algorithm does not take this information into account to suppress the negative influence of the disturbance.

The experiment was conducted to test the ability of the system to identify the disturbance model. The experiment was conceived that simulates the batch-wise operation of the pH process. The reference signal was changing according to the predefined periodic signal. Since the system was operating in the 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 signals used for 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 undesirably. Oscillations of manipulated and controlled variables can be noticed. It has to be stressed that the problems were not encountered in all operating points nor were 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).

In conclusion, it can be said that the option of identifying both models (the control one and the disturbance one) 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 high noise or a lot of the optional components in the OLA are enabled. The more possibilities are enabled in the RTM, the lower robustness of the system can be expected in general.

### 5.2 Behaviour of the RTM in the noisy environment when operating in the 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 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 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 product design the dead

zone was included into the identification procedure. This prevents the regressors that do not carry any new information about the plant to take part in the parameter calculation. The rationale behind the idea is that the main component of such regressors is probably noise that would lead the estimated parameters to the wrong direction.

Similar experiment as mentioned before (subsection 5.1) was conducted. The disturbance was not measured in this case but the level of noise was very high. Two identification algorithms were compared: the one with two »dead zones« and the other with only one (it did have the dead zone in the least squares algorithm but it did not prevent that only slightly better model be accepted).

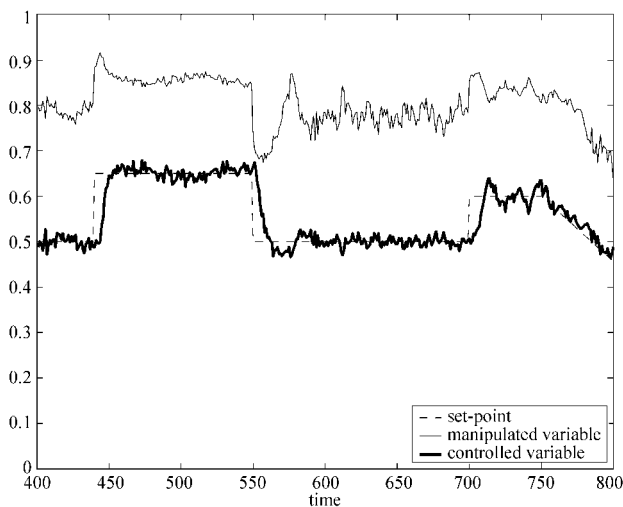


Fig. 5 Performance of the system in the beginning (without the second dead zone)

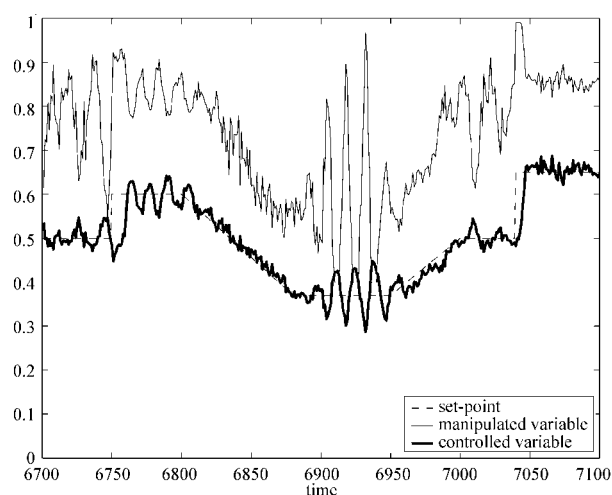


Fig. 6 Performance of the system after some time (without the second dead zone)

The results in the latter case were discouraging to some extent since they have shown the quality of the model can decrease with time. The deterioration was not instantaneous. Rather, the drift in the parameters can be observed. Such drift would not be possible if identification was performed in the 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 by a very conservatively tuned controller. The bandwidth of the system is therefore low and the manipulated variable is relatively slow (almost open loop). This is why a good model is obtained in the beginning (see Figure 5). The procedure of the controller tuning is such that results in high-gain controller. This is why the manipulated variable is much more vivid and also correlated with the controlled variable. The obtained model is worse because of that. The next controller results in an even more oscillating system. Consequently, the quality of signals used for identification is very low (more or less only one frequency is present). This cycle leads to the deterioration of the performance. In Figure 6 the signals in the system are depicted after a period of time when the system is run with the OLA agent active.

The problem lies in the very core of the approach. Usually, one part of the signal is used for identification and the other for validation. Since the fuzzy model is used it is possible that the system changes the fuzzy domain approximately in the moment when the »validation period« starts. In such case the validation would be completely wrong.

It is obvious that something has to be done to prevent very frequent changes of the model. The other algorithm (with two »dead zones«) achieved precisely that. Figures 7 and 8 represent the behaviour of the system in the same time intervals as Figures 5 and 6 in the previous experiment. It can be observed that the gradual deterioration of the performance is prevented. However, this solution is not absolute. It implicitly prevents small changes in the model. Consequently, it is hard for the algorithm to reach global optimum. As always, a trade-off between the performance and the robustness is performed. In this case, our standpoint is that robustness is more important than the optimal performance.

Some experiments have also been made with dead time unit and membership functions unit. Both of them are called periodically (after a certain number of parameter identifications) if enabled. Both of them are not very robust and demand the 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

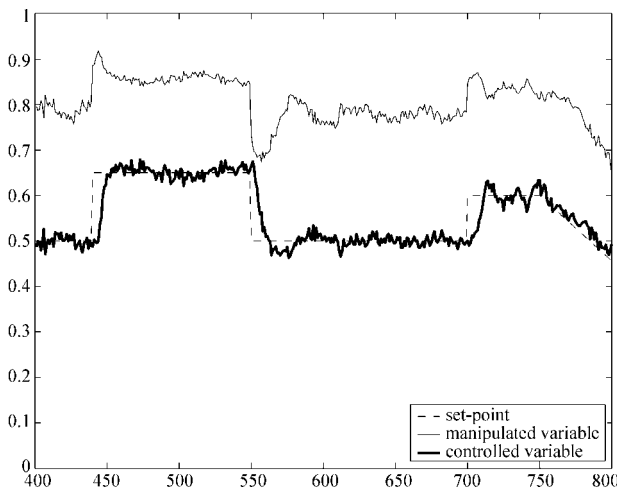


Fig. 7 Performance of the system in the beginning (with the second dead zone)

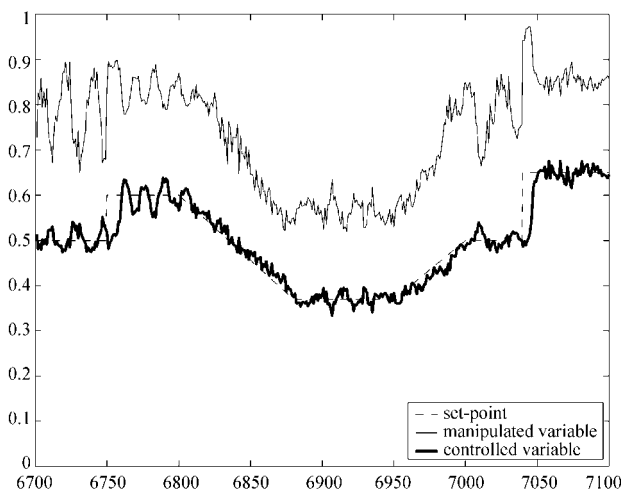


Fig. 8 Performance of the system after some time (with the second dead zone)

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 it finds it appropriate. Again, it calculates the performance indexes with and without a new fuzzy domain added. It only tries to add

a new fuzzy domain if the augmented model is much better (e.g. for a multiplicative constant 2) from the original model.

## 6 CONCLUSION

The initial tests have shown that the ASPECT controller is capable of controlling the nonlinear plants, the plants with delay and slowly varying parameters. The identification algorithm is designed such that it prevents parameter drift due to noise especially when run in the closed-loop. It is also capable of dealing with systems with (variable) delay. This possibility is not suggested for the use in closed-loop since it does not show high level of robustness. The algorithm can also find the nonlinearity in the system and adds an extra membership function when appropriate.

## ACKNOWLEDGEMENT

The authors would like to acknowledge the contributions of all other project team members from Jožef Stefan Institute, Univ. 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

- [1] S. Blažič, I. Škrjanc, D. Matko, **Globally Stable Direct Fuzzy Model Reference Adaptive Control**. *Fuzzy Sets & Syst.*, 139, 1, 3–33, 2003.
- [2] M. A. Henson, D. E. Seborg, **Adaptive Non-linear Control of a pH Neutralisation Process**. *IEEE Trans. CST*, vol. 2, 169–182, 1994.
- [3] P. Ioannou, J. Sun, **Robust Adaptive Control**. Prentice-Hall, 1996.
- [4] L. Ljung, **System Identification**, Englewood Cliffs NJ: Prentice Hall, 1987.
- [5] B. B. Peterson, K. S. Narendra, **Bounded Error Adaptive Control**, *IEEE Transactions on Automatic Control*, AC-27, 6, 1161–1168, 1982.
- [6] I. Škrjanc, D. Matko, **Predictive Functional Control Based on Fuzzy Model for Heat-exchanger Pilot Plant**. *IEEE Transactions on Fuzzy Systems*, 8, 6, 705–712, 2000.
- [7] T. Takagi, M. Sugeno, **Fuzzy Identification of Systems and its Applications to Modelling and Control**. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-15, 1, 116–132, 1985.
- [8] L. X. Wang, J. M. Mendel, **Fuzzy Basis Functions, Universal Approximation, and Orthogonal Least-squares Learning**. *IEEE Transactions on Neural Networks*, 3, 5, 807–81, 1992.

**Jednostavna neizrazita identifikacija implementirana u naprednom regulatoru.** Ovaj se rad usredotočuje na problematiku identifikacije na osnovi naprednog regulatora ASPECT\* implementiranog na jednostavnoj PLC platformi s dodatnim matematičkim koprocesorom, koji se želi koristiti za naprednu regulaciju složenih postrojenja. Model reguliranog postrojenja dobiva se eksperimentalnim modeliranjem, pri čemu se koristi on-line procedura učenja s pred- i post-identifikacijskim koracima koji osiguravaju pouzdan rad. Pokazano je da se prihvatljive performance sustava dobivaju unatoč teškim uvjetima koji se mogu pojaviti tijekom rada.

**Ključne riječi:** programirajući logički regulatori, neizrazito modeliranje, identifikacija, nelinearna regulacija, regulacija pH

#### AUTHORS' ADDRESSES

**S. Blažič, I. Škrjanc**

University of Ljubljana, Faculty of Electrical Engineering,  
Tržaška 25, Ljubljana, Slovenia

**S. Gerkšič, G. Dolanc, S. Strmčnik**

Jožef Stefan Institute, Jamova 39, Ljubljana, Slovenia

**M. B. Hadjiski**

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

**A. Stathaki**

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

Received: 2004–11–30