Available online at www.sciencedirect.com
SCIENCE DIRECT®
Control Engineering Practice 14 (2006) 935–948
CONTROL ENGINEERING PRACTICE
www.elsevier.com/locate/conengprac

Advanced control algorithms embedded in a programmable logic controller

Samo Gerkšič[a,*], Gregor Dolanc[a], Damir Vrančić[a], Juš Kocijan[a,b], Stanko Strmčnik[a], Sašo Blažič[c], Igor Škrjanc[c], Zoran Marinšek[d], Miha Božiček[d], Anna Stathaki[e], Robert King[e], Mincho Hadjiski[f], Kosta Boshnakov[f]

[a]*Jožef Stefan Institute, Ljubljana, Slovenia*
[b]*Nova Gorica Polytechnic, Nova Gorica, Slovenia*
[c]*University of Ljubljana, Faculty of Electrical Engineering, Ljubljana, Slovenia*
[d]*INEA d.o.o., Ljubljana, Slovenia*
[e]*Computer Technology Institute, Athens, Greece*
[f]*University of Chemical Technology and Metallurgy Sofia, Sofia, Bulgaria*

Received 23 April 2004; accepted 15 May 2005
Available online 22 July 2005

Abstract

This paper presents an innovative self-tuning nonlinear controller ASPECT (advanced control algorithms for programmable logic controllers). It is intended for the control of highly nonlinear processes whose properties change radically over its range of operation, and includes three advanced control algorithms. It is designed using the concepts of agent-based systems, applied with the aim of automating some of the configuration tasks. The process is represented by a set of low-order local linear models whose parameters are identified using an online learning procedure. This procedure combines model identification with pre- and post-identification steps to provide reliable operation. The controller monitors and evaluates the control performance of the closed-loop system. The controller was implemented on a programmable logic controller (PLC). The performance is illustrated on a field test application for control of pressure on a hydraulic valve.

Keywords: Control engineering; Fuzzy modelling; Industrial control; Model-based control; Nonlinear control; Programmable logic controllers; Self-tuning regulators

1. Introduction

Modern control theory offers many control methods to achieve more efficient control of nonlinear processes than provided by conventional linear methods, taking advantage of more accurate process models (Bequette, 1991; Henson & Seborg, 1997; Murray-Smith & Johansen, 1997). Surveys (Takatsu, Itoh, & Araki, 1998; Seborg, 1999) indicate that while there is a considerable and growing market for advanced controllers, relatively few vendors offer turn-key products.

Excellent results of advanced control concepts, based on fuzzy parameter scheduling (Tan, Hang, & Chai, 1997; Babuška, Oosterhoff, Oudshoorn, & Bruijn, 2002), multiple-model control (Dougherty & Cooper, 2003; Gundala, Hoo, & Piovoso, 2000), and adaptive control (Henson & Seborg, 1994; Hägglund & Åstrom, 2000), have been reported in the literature. However, there are several restrictions for applying these methods in industrial applications, as summarised below:

1. Because of the diversity of real-life problems, a single nonlinear control method has a relatively narrow

*Corresponding author. Tel.: +386 1 477 3994; fax: +386 1 425 7009.
E-mail address: samo.gerksic@ijs.si (S. Gerkšič).

0967-0661/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.conengprac.2005.05.006

field of application. Therefore, more flexible methods or a toolbox of methods are required in industry.
2. New methods are usually not available in a ready-to-use industrial form. Custom design requires considerable effort, time and money.
3. The hardware requirements are relatively high, due to the complexity of implementation and computational demands.
4. The complexity of tuning (Babuška et al., 2002) and maintenance makes the methods unattractive to nonspecialised engineers.
5. The reliability of nonlinear modelling is often in question.
6. Many nonlinear processes can be controlled using the well-known and industrially proven PID controller. A considerable direct performance increase (financial gain) is demanded when replacing a conventional control system with an advanced one. The maintenance costs of an inadequate conventional control solution may be less obvious.

The aim of this work is to design an advanced controller that addresses some of the aforementioned problems by using the concepts of agent-based systems (ABS) (Wooldridge & Jennings, 1995). The main purpose is to simplify controller configuration by partial automation of the commissioning procedure, which is typically performed by the control engineer. ABS solve difficult problems by assigning tasks to networked software agents. The software agents are characterised by properties such as autonomy (operation without direct intervention of humans), social ability (interaction with other agents), reactivity (perception and response to the environment), pro-activeness (goal-directed behaviour, taking the initiative), etc. This work does not address issues of ABS theory, but rather the application of the basic concepts of ABS to the field of process systems engineering. In this context, a number of limits have to be considered. For example: initiative is restricted, a high degree of reliability and predictability is demanded, insight into the problem domain is limited to the sensor readings, specific hardware platforms are used, etc.

The ASPECT controller is an efficient and user-friendly engineering tool for implementation of parameter-scheduling control in the process industry. The commissioning of the controller is simplified by automatic experimentation and tuning. A distinguishing feature of the controller is that the algorithms are adapted for implementation on PLC or open controller industrial hardware platforms.

The controller parameters are automatically tuned from a nonlinear process model. The model is obtained from operating process signals by experimental modelling, using a novel online learning procedure. This procedure is based on model identification using the local learning approach (Murray-Smith & Johansen, 1997, p. 188). The measurement data are processed batch-wise. Additional steps are performed before and after identification in order to improve the reliability of modelling, compared to adaptive methods that use recursive identification continuously (Hägglund & Åstrom, 2000).

The nonlinear model comprises a set of local low-order linear models, each of which is valid over a specified operating region. The active local model(s) is selected using a configured scheduling variable. The controller is specifically designed for single-input, single-output processes that may include a measured disturbance used for feed-forward. Additionally, the application range of the controller depends on the selected control algorithm. A modular structure of the controller permits use of a range of control algorithms that are most suitable for different processes. The controller monitors the resulting control performance and reacts to detected irregularities.

The controller comprises the run-time module (RTM) and the configuration tool (CT). The RTM runs on a PLC, performing all the main functionality of real-time control, online learning and control performance monitoring. The CT, used on a personal computer (PC) during the initial configuration phase, simplifies the configuration procedure by providing guidance and default parameter values.

The outline of the paper is as follows: Section 2 presents an overview of the RTM structure and describes its most important modules; Section 3 gives a brief description of the CT; and finally, Section 4 describes the application of the controller to a pilot plant where it is used for control of the pressure difference on a hydraulic valve in a valve test apparatus.

2. Run-Time Module

The RTM of the ASPECT controller comprises a set of modules, linked in the form of a multi-agent system. Fig. 1 shows an overview of the RTM and its main modules: the signal pre-processing agent (SPA), the online learning agent (OLA), the model information agent (MIA), the control algorithm agent (CAA), the control performance monitor (CPM), and the operation supervisor (OS).

2.1. Multi-faceted model (MFM)

The ASPECT controller is based on the multi-faceted model concept proposed by Stephanopoulus, Henning, and Leone (1990) and incorporates several model forms required by the CAA and the OLA. Specifically, the MFM includes a set of local first- and second-order discrete-time linear models with time delay and offset,

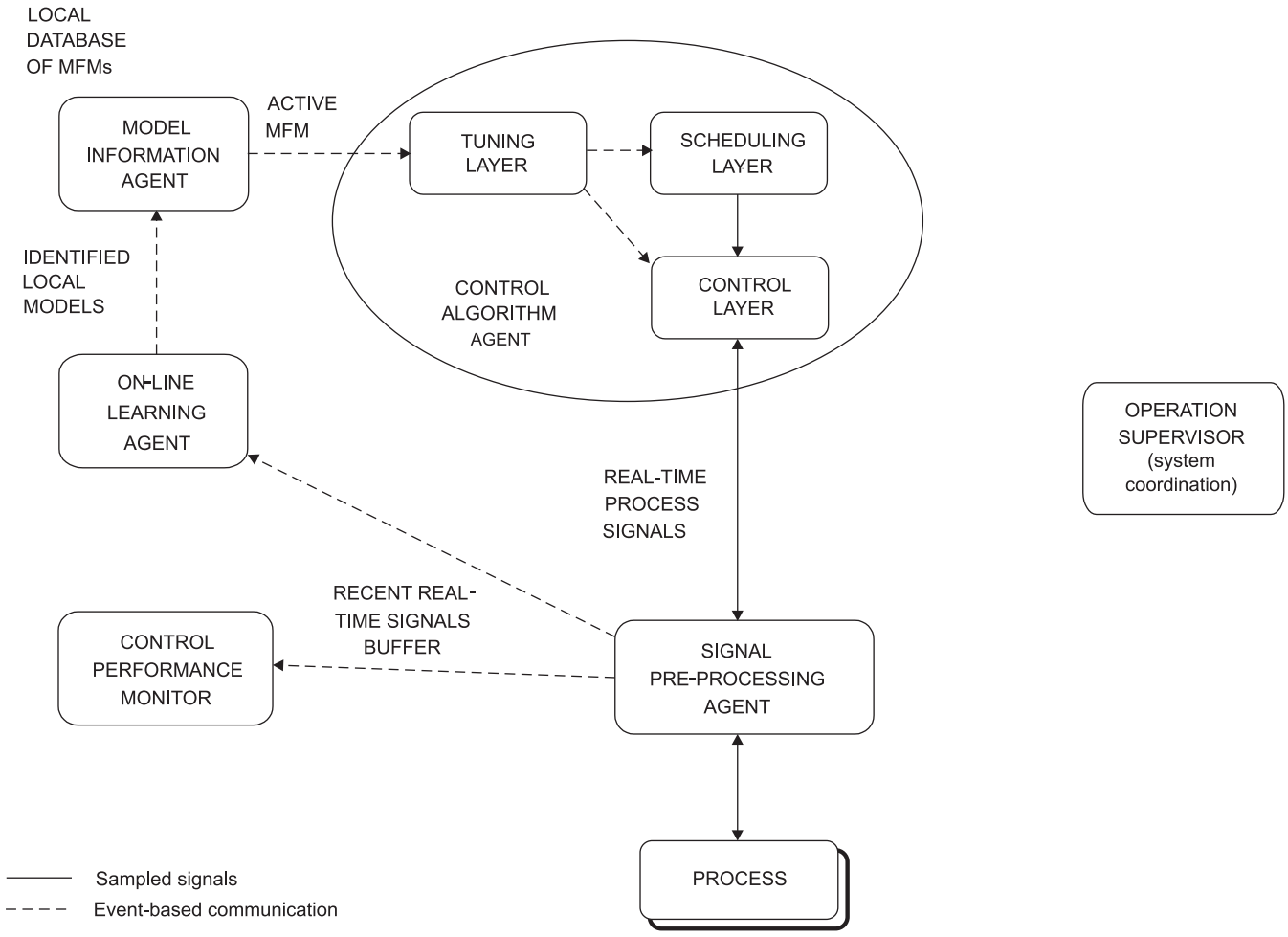

Fig. 1. Run-time module overview.

which are specified by a given scheduling variable $s(k)$. The model equation of first order local models is

$$y(k+1) = -a_{1,j}y(k) + b_{1,j}u(k-\mathrm{d}u_j) + c_{1,j}v(k-\mathrm{d}v_j) + r_j, \tag{1}$$

while the model equation of second order models is

$$\begin{aligned} y(k+1) = {} & -a_{1,j}y(k) - a_{2,j}y(k-1) + b_{1,j}u(k-\mathrm{d}u_j) \\ & + b_{2,j}u(k-1-\mathrm{d}u_j) + c_{1,j}v(k-\mathrm{d}v_j) \\ & + c_{2,j}v(k-1-\mathrm{d}v_j) + r_j, \end{aligned} \tag{2}$$

where k is the discrete time index, j is the number of the local model, $y(k)$ is the process output signal, $u(k)$ is the process input signal, $v(k)$ is the optional measured disturbance signal (MD), du is the delay in the model branch from u to y, dv is the delay in the model branch from v to y, and $a_{i,j}$, $b_{i,j}$, $c_{i,j}$ and r_j are the parameters of the jth local model.

The set of local models can be interpreted as a Takagi–Sugeno fuzzy model, which in the case of a second order model can be expressed in the following form:

$$\begin{aligned} y(k+1) = {} & -\sum_{j=1}^{m}\beta_j(k)a_{1,j}y(k) - \sum_{j=1}^{m}\beta_j(k)a_{2,j}y(k-1) \\ & + \sum_{j=1}^{m}\beta_j(k)b_{1,j}u(k-\mathrm{d}u_j) \\ & + \sum_{j=1}^{m}\beta_j(k)b_{2,j}u(k-1-\mathrm{d}u_j) \\ & + \sum_{j=1}^{m}\beta_j(k)c_{1,j}v(k-\mathrm{d}v_j) \\ & + \sum_{j=1}^{m}\beta_j(k)c_{2,j}v(k-1-\mathrm{d}v_j) + \sum_{j=1}^{m}\beta_j(k)r_j, \end{aligned} \tag{3}$$

where $\beta_j(k)$ is the value of the membership function of the jth local model at the current value of the scheduling variable $s(k)$. Normalised triangular membership functions are used, as illustrated in Fig. 2.

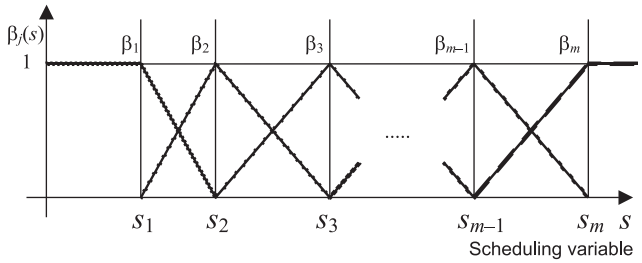

Fig. 2. Fuzzy membership functions of local models in the MFM.

The scheduling variable $s(k)$ is calculated using coefficients k_r, k_y, k_u, and k_v, using the weighted sum

$$s(k) = k_r r(k) + k_y y(k) + k_u u(k-1) + k_v v(k). \quad (4)$$

The coefficients are configured by the engineer according to the nature of the process nonlinearity.

2.2. Online Learning Agent (OLA)

The OLA scans the buffer of recent real-time signals, prepared by the SPA, and estimates the parameters of the local models that are excited by the signals. The most recently derived parameters are submitted to the MIA only when they pass the verification test and are proved to be better than the existing set.

The OLA is invoked upon demand from the OS or autonomously, when an interval of the process signals with sufficient excitation is available for processing. It processes the signals batch-wise and using the local learning approach. An advantage of the batch-wise concept is that the decision on whether to adapt the model is not performed in real-time but following a delay that allows for inspection of the identification result before it is applied. Thus, better means for control over data selection is provided.

A problem of distribution of the computation time required for identification appears with batch-wise processing of data (opposed to the online recursive processing that is typically used in adaptive controllers). This problem is resolved using a multi-tasking operation system. Since the OLA typically requires considerably more computation than the real-time control algorithm, it runs in the background as a low-priority task.

The following procedure, illustrated in Fig. 3, is executed when the OLA is invoked.

2.2.1. Copy signal buffer

The buffer of the real-time signals is maintained by the SPA. When the OLA is invoked, the relevant section of the buffer is copied for further processing.

2.2.2. Excitation check

A quick excitation check is performed at the start, so that processing of the signals is performed only when they contain excitation. If the standard deviations of the signals $r(k)$, $y(k)$, $u(k)$, and $v(k)$ in the active buffer are below their thresholds, the execution is cancelled.

2.2.3. Copy active MFM from MIA

The online learning procedure always compares the newly identified local models with the previous set of parameters. Therefore, the active MFM is copied from the MIA where it is stored. A default set of model parameters is used for the local models that have not yet been identified (see Section 2.3).

2.2.4. Select local models

A local model is selected if the sum of its membership functions $\beta_j(k)$ over the active buffer normalised by the active buffer length exceeds a given threshold. Only the selected local models are included in further processing.

2.2.5. Identification

The local model parameters are identified using the fuzzy instrumental variables (FIV) identification method developed by Blažič et al. (2003). It is an extension of the linear instrumental variables identification procedure (Ljung, 1987) for the specified MFM, based on the local learning approach (Murray-Smith & Johansen, 1997). The local learning approach is based on the assumption that the parameters of all local models will not be estimated in a single regression operation. Compared to the global approach it is less prone to the problems of ill-conditioning and local minima.

This method is well suited to the needs of industrial operation (intuitiveness, gradual building of the nonlinear model, modest computational demands). It enables inventory of the local models that are not estimated properly due to insufficient excitation. It is efficient and reliable in early configuration stages, when all local models have not been estimated yet. On the other hand, the convergence in the vicinity of the optimum is slow. Therefore, it is likely to yield a worse model fit than methods employing nonlinear optimisation. The following briefly describes the procedure.

Model identification is performed for each selected local model (denoted by the index j) separately. The initial estimated parameter vector $\hat{\boldsymbol{\theta}}_{j,\mathrm{MIA}}$ is copied from the active MFM, and the covariance matrix $\mathbf{P}_{j,\mathrm{MIA}}$ is initialised to 10^5 $\mathbf{I}$ (identity matrix). The FLS (fuzzy least-squares) estimates, $\hat{\boldsymbol{\theta}}_{j,\mathrm{FLS}}$ and $\mathbf{P}_{j,\mathrm{FLS}}$, are obtained using weighted least-squares identification, with $\beta_j(k)$ used for weighting. The calculation is performed recursively to avoid matrix inversion. The FIV (fuzzy instrumental variables) estimates, $\hat{\boldsymbol{\theta}}_{j,\mathrm{FIV}}$ and $\mathbf{P}_{j,\mathrm{FIV}}$, are calculated using weighted instrumental variables identification.

In order to prevent result degradation by noise, a dead zone is used in each step of FIV and FLS recursive

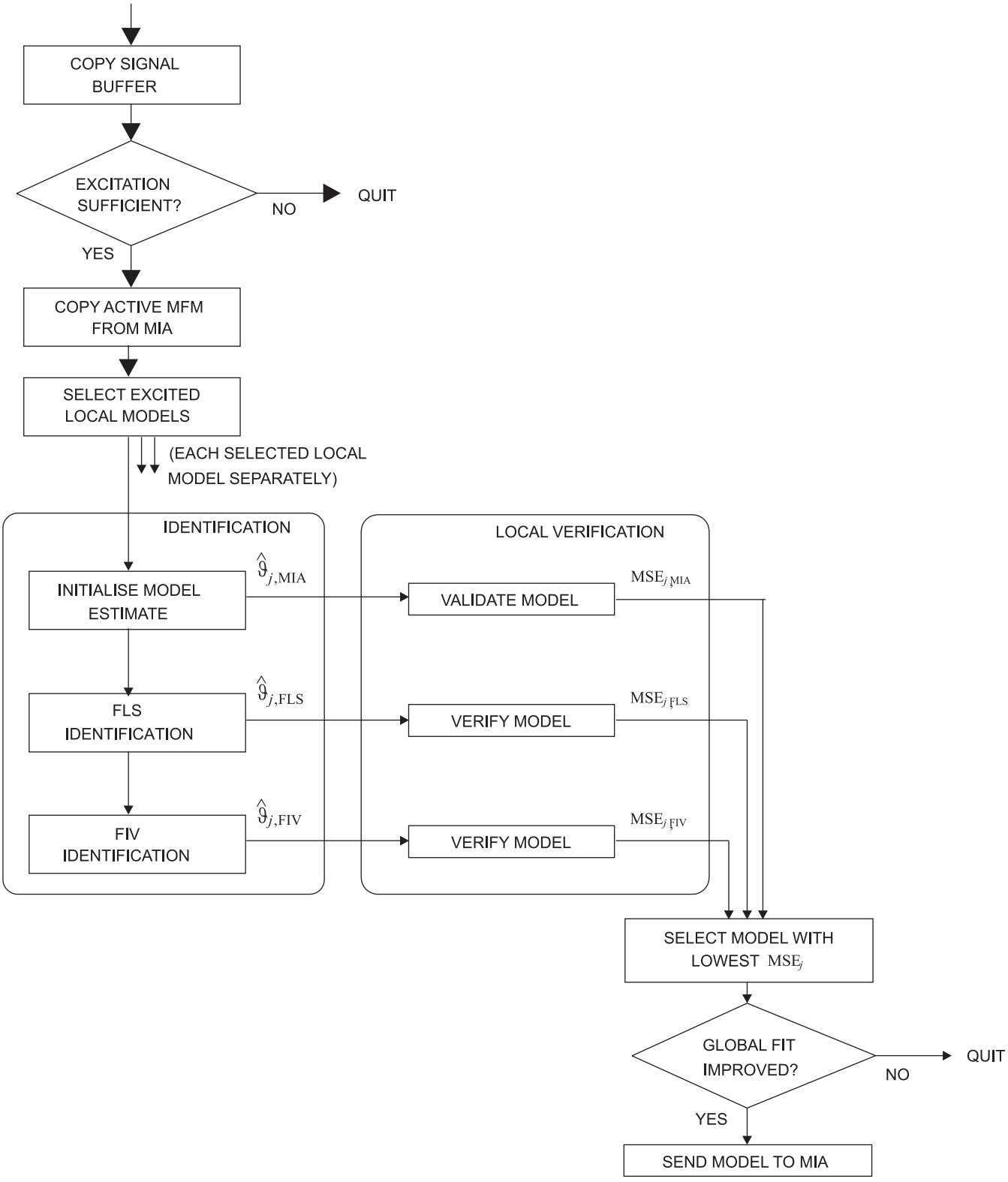

Fig. 3. Online learning procedure.

estimation. The vector of parameters and the covariance matrix are updated only if the absolute weighted difference between the process output and its prediction is above the configured noise threshold.

In case of lack of excitation in the branch from u to y or in the model branch from v to y (or when measured disturbance is not present at all), variants of the method with reduced parameter estimate vectors are used.

2.2.6. Verification/validation

This step is performed by comparing the simulation output of a selected local model with the actual process output in the proximity of the local model position. The normalised sum of mean square errors (MSE_j) is calculated. The proximity is defined by the membership functions β_j. For each of the selected local models, this step is carried out with three sets of model parameters: $\hat{\boldsymbol{\theta}}_{j,\mathrm{MIA}}$, $\hat{\boldsymbol{\theta}}_{j,\mathrm{FLS}}$, and $\hat{\boldsymbol{\theta}}_{j,\mathrm{FIV}}$. The set with the lowest MSE_j is selected.

Then, global verification is performed by comparing the simulation output of the fuzzy model including the selected set with the actual process output. The normalised sum of mean square errors (MSE_G) is calculated. If the global verification result is improved compared to the initial fuzzy model, the selected set is sent to the MIA as the result of online learning, otherwise the original set $\hat{\boldsymbol{\theta}}_{j,\mathrm{MIA}}$ remains in use.

For each processed local model, the MIA receives the MSE_j, which serves as a confidence index, and a flag indicating whether the model is new or not.

2.2.7. Model structure estimation

Two model structure estimation units are also included in the OLA. The dead-time unit (DTU) estimates the process time delay. The membership function unit (MFU) suggests whether a new local model should be inserted. It estimates an additional local model in the middle of the interval between the two neighbouring local models that are the most excited. The model is submitted to the MIA if the global validation of the resulting fuzzy model is sufficiently improved, compared to the original fuzzy model.

2.3. Model Information Agent (MIA)

The task of the MIA is to maintain the active MFM and its status information.

Its primary activity is processing the online learning results. When a new local model is received from the OLA, it is accepted if it passes the stability test and its confidence index is sufficient. If it is accepted, a "ready for tuning" flag is set for the CAA. Another flag indicates whether the local model has been tuned since start-up or not. If the model confidence index is very low, the automatic mode may be disabled.

The MIA contains a mechanism for inserting additional local models into the MFM. This may occur either by request or automatically, using the MFU of the OLA. The MIA may also store the active MFM to a local database or recall a previously stored one, which is useful for changing of modes.

At initial configuration, the MIA is filled with default local models based on the initial estimation of the process dynamics. They are not exact but may provide reliable (although sluggish) control performance, similar to the Safe mode. Using online learning through experiments or normal operating records (when the conditions are appropriate for closed-loop identification), an accurate model of the plant is estimated gradually by receiving identified local models from the OLA.

2.4. Control Algorithm Agent (CAA)

The CAA comprises an industrial nonlinear control algorithm and a procedure for automatic tuning of its parameters from the MFM. Several different CAAs may be used in the controller and may be interchanged in the initial configuration phase.

The following modes of operation are supported:

- *Manual mode*: open-loop operation (actuator constraints are enforced).
- *Safe mode*: a fixed PI controller with conservatively tuned parameters.
- *Auto mode* (or several auto modes with different tuning parameters): a nonlinear controller.

The CAAs share a common interface of interaction with the OS and a common modular internal structure, consisting of three layers:

1. The control layer offers the functionality of a local linear controller (or several local linear controllers simultaneously), including everything required for industrial control, such as handling of constraints with anti-windup protection, bump-less mode switching, etc.
2. The scheduling layer performs real-time switching or scheduling (blending) of tuned local linear controllers, so that in conjunction with the control layer a fixed-parameter nonlinear controller is realised.
3. The tuning layer executes the automatic tuning procedure of the controller parameters from the MFM when the MIA reports that a new local model is generated if auto-tuning is enabled. The parameters of the control layer and the scheduling layer are replaced in such a manner that real-time control is not disturbed.

Three CAAs have been developed and each has been proved effective in specific applications: the Fuzzy parameter-scheduling controller (FPSC), the dead-time compensation controller (DTCC), and the rule-based neural controller (RBNC). In this paper, only the concept of the FPSC is described briefly in the following subsection.

2.4.1. Fuzzy parameter-scheduling controller

An overview of the FPSC is shown in Fig. 4.

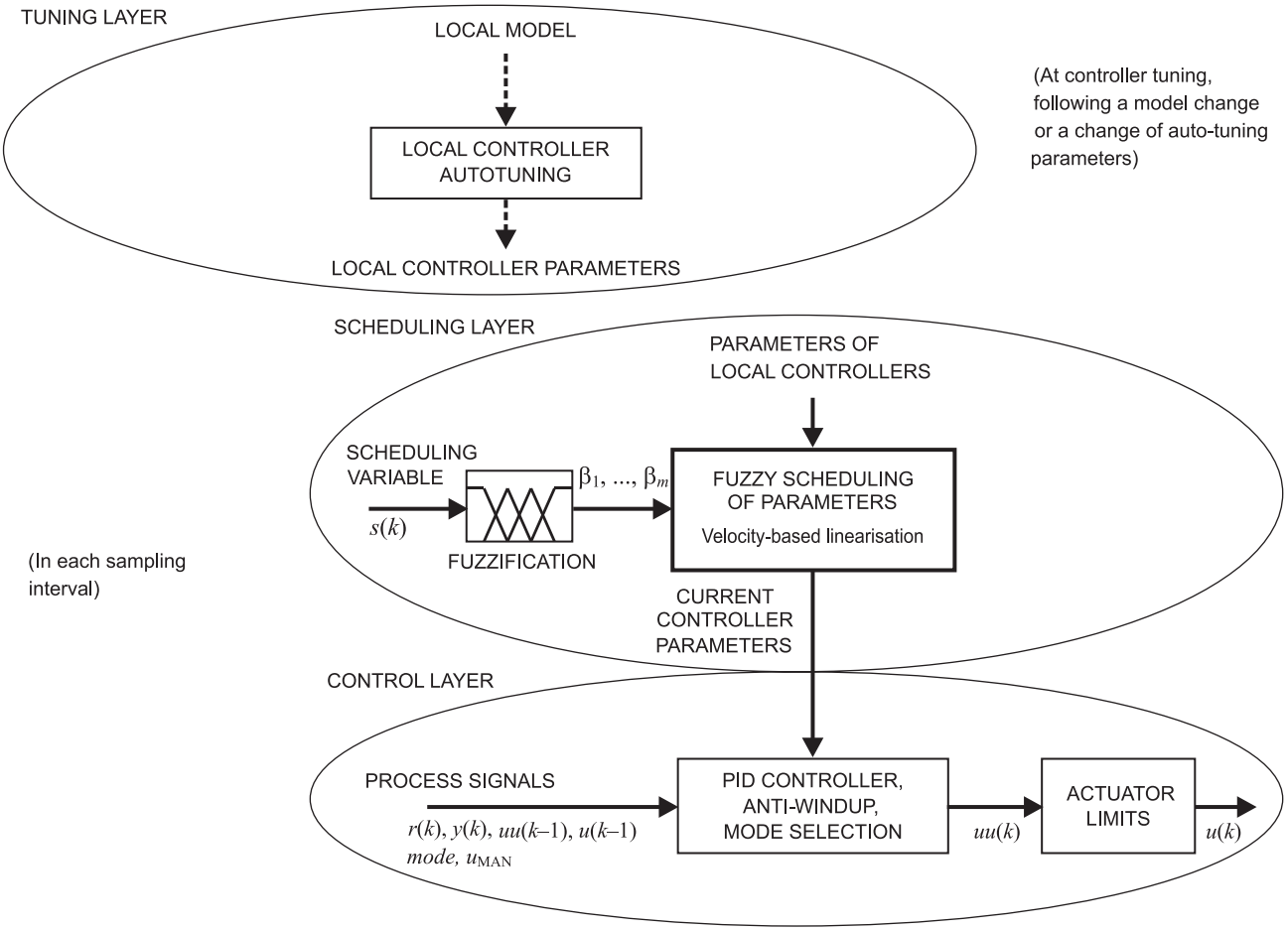

Fig. 4. FPSC overview.

The control layer of the FPSC includes a single PID controller in the form suitable for controller blending using velocity-based linearisation. It is equipped with anti-windup protection and bump-less transfer.

The scheduling layer of the FPSC performs fuzzy blending of the controller parameters (in the case of *Ti*, its inverse value) according to the scheduling variable $s(k)$ and the membership functions $\beta_j(k)$ of the local models. The instrument of velocity-based linearisation enables the dynamics of the blended global controller to be a linear combination of the local controller dynamics in the entire operating region, not just around equilibrium operating points. This provides the potential to improve performance with few local models and more transparent behaviour in off-equilibrium operating points (Leith & Leithead, 1998; Kocijan, Žunič, Strmčnik, & Vrančić, 2002).

The tuning layer of the FPSC is based on the magnitude optimum (MO) criterion implemented using the multiple integration (MI) method (Vrančić, Strmčnik, & Juričić, 2001). The MO criterion makes the magnitude (amplitude) of the system closed-loop set-point response as flat and close to unity as possible for a large bandwidth (Whiteley, 1946). This results in a relatively fast and nonoscillatory response of the closed-loop system. The expressions for calculating the PID controller parameters using the MO criterion are quite complex. However, the MI method significantly simplifies the equations and enables the calculation of the PID controller parameters directly from the process open-loop time response.

The auto-tuning procedure starts by receiving a discrete-time local model from the MIA. The model is converted into a continuous-time local model. Then, the so-called areas are calculated from the model using the MI method. Finally, the PI and PID controller parameters are calculated from the areas. Thanks to the transparent concept of the FPSC, an experienced engineer may choose to configure the control algorithm manually by specifying the local PID controller positions and parameters, without using the model-based tuning procedure.

2.5. Control performance monitor (CPM)

The CPM scans the buffer of recent real-time signals for recognisable events. When events are detected, it estimates the features of closed-loop control response

and an overall performance index. Like the OLA, it is invoked autonomously or upon demand from the OS and runs as a low-priority task. It consists of three modules: the buffer pre-processor (BP), the situation classifier (SC), and the performance estimator (PE), as shown in Fig. 5.

When the CPM is invoked, the BP copies the relevant section of the buffer of the real-time signals, which is maintained by the SPA. It checks if the process is in a steady-state; if so, it terminates processing. Otherwise, it filters the signals y and v and performs a low-level analysis. The SC scans the pre-processed buffer for the last recognisable event that may be evaluated, or is otherwise important, e.g., a step change of the reference signal, a step change of the measured disturbance signal, an occurrence of an unmeasured disturbance, or the presence of oscillation. If an event that may be evaluated is detected and the conditions for feature estimation are fulfilled (there is no excessive oscillation, the signal-to-noise ratio is sufficient, the process response has settled after the event and there was a period of steady-state before the event), the corresponding buffer interval is sent to the PE, otherwise the execution is terminated.

The PE may extract the following features of detected events: overshoot, settling time, rise time, oscillation decay rate, and tracking error measure or regulation error measure. Using a fuzzy evaluation procedure, an overall performance index (PI) is also calculated from the features.

The CPM results are sent to the human–machine interface. If poor performance is detected, the CPM triggers an automatic switchover to the Safe mode. Other automatic actions include, for example, blocking the OLA if oscillation is detected. Modifications of the CAA parameters based on CPM results are not generally performed because such actions are highly process-specific, but they may be implemented in specific applications.

2.6. Operation supervisor (OS)

The OS coordinates the control, modelling, and tuning activities of the agents and user interaction through the hierarchical set of dialogue windows of the human-machine interface (HMI). The OS and the HMI include the functionality required for automatic user-friendly experimentation, which is usually required for controller tuning.

The controller commissioning procedure comprises the phases of basic settings, approximate estimation of the process dynamics for safe controller tuning, non-linear modelling and tuning of the scheduling controller, and configuring the regime for regular operation.

The OS supports the control engineer by automatic execution of experiments for identification of local models. These experiments consist of a series of step changes about the operating point of the model in either open or closed loop. In addition, the OS coordinates the OLA, the MIA and the CAA to automatically process the signals, build the model and tune the local controllers. This is the fastest and most reliable way of controller tuning when experimentation with the plant is allowed. Automatic conducting of experiments for closed-loop performance testing using the CPM is also supported.

Alternatively, scheduling of experiments may not be required if experimentation is not allowed. In this case, the controller is initialised in the safe mode, and processing of the signals for modelling and tuning is triggered by the OLA autonomously. However, it is required that sufficient excitation over the whole operating region is available during regular operation. The modelling progress is indicated by the status flags of the local models in the MIA. The flags show which models have been tuned and their confidence indices.

While initiative and suggestions of the agents are helpful during system configuration, this may not be desired during regular operation. Therefore, at the end of the commissioning procedure, the system may be reconfigured to simplify the operation as much as possible.

A range of operating regimes can be configured by enabling or disabling the agents and changing their configuration parameters. This results in a flexible control system that covers the requirements of a wide range of applications and may help diagnose problems. Thus, although designed for control of nonlinear processes, the ASPECT controller may also be used for adaptive control using a single linear model or as a tool for PID controller tuning. Some specific operating regime options are listed below:

- The OLA and/or the CPM may be invoked autonomously (during regular operation) or upon OS demand (following scheduled experiments), or both.
- The OLA may estimate the process dead-time continuously or not.
- The OLA may attempt to insert additional local models when appropriate, or estimate the local models at the fixed pre-selected positions only.
- Controller retuning may be triggered automatically immediately after each change of the model in MIA (“adaptive” operation), or following an engineer’s confirmation (“self-tuning” operation).
- Online learning may also be used for monitoring of the process dynamics without the intention of controller tuning, either by adaptation of the model or by cross-validation of a fixed model.

At start up, the system is initialised from a configuration file, which may put it into any phase of the

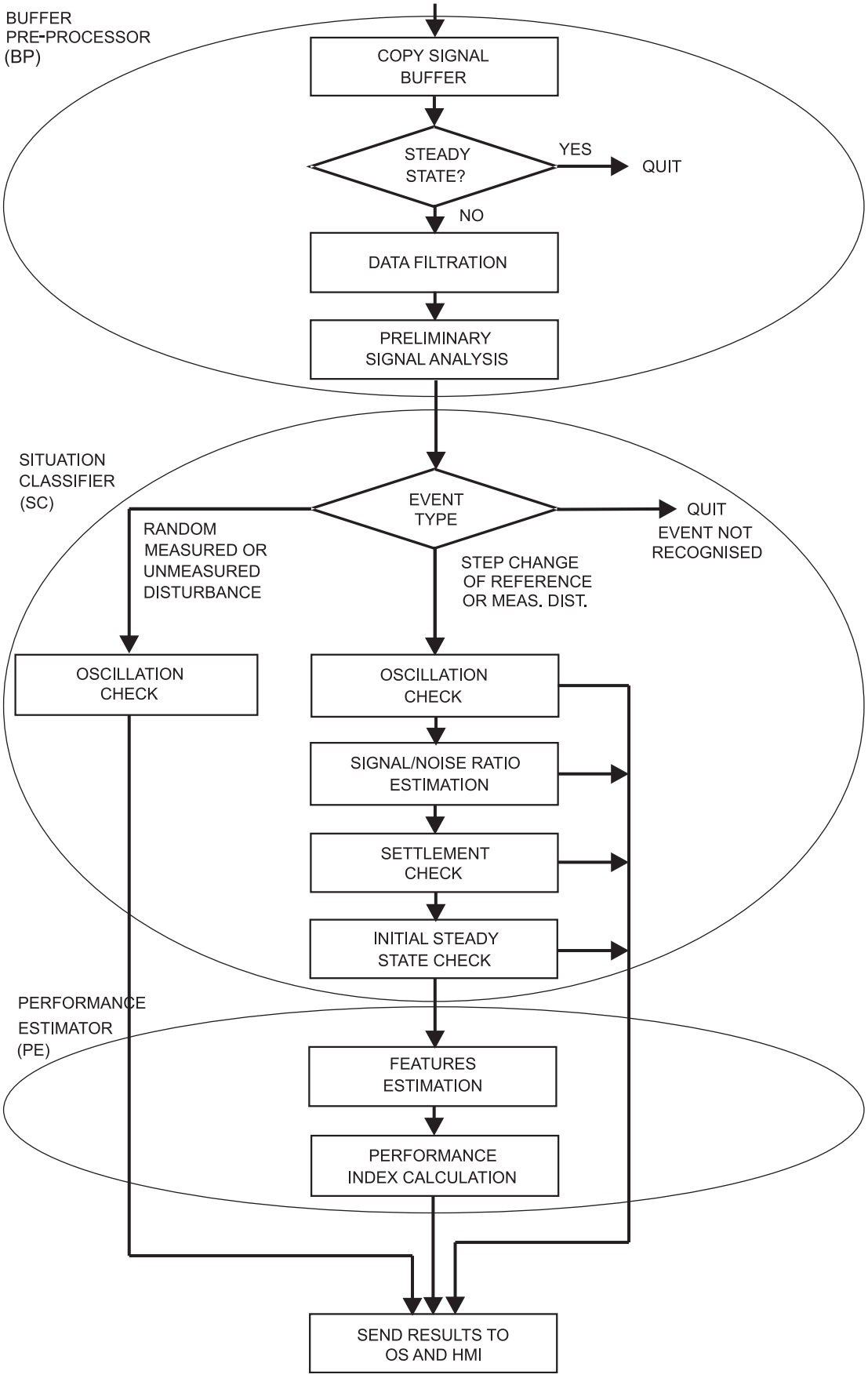

Fig. 5. CPM overview.

configuration procedure. Afterwards, the configuration procedure may be continued or repeated using the HMI dialogue windows.

3. Configuration tool

The CT is used on a PC, connected to the PLC running the RTM. The CT contains a configuration "wizard" that guides the engineer through the typical scheduling controller commissioning procedure. It is intended for less experienced users. Experienced engineers may find it more efficient to perform configuration only using the RTM, where other sequences of the tasks are possible.

The procedure is decomposed into small steps (25 dialogue windows). In each step, instructions are displayed and default values are suggested by using rules of thumb, based on the information already available. Inconsistency warnings may be displayed.

The main phases of the configuration procedure are:

- Basic settings: selection of the control signals, the signal limits, the sampling time, the control algorithm, the scheduling variable, the model order.
- Safe mode configuration: estimation of the process dynamics (experimentation and identification using the RTM may be used), self-tuning of the "safe" controller parameters (using the RTM), optional performance verification.
- Fuzzy model initialisation: initialisation of the model positions, initialisation of the local model parameters, display of the local model parameters and step responses.
- CAA settings: initialisation of the default values, advanced auto-tuning parameters.
- OLA settings: initialisation of the default values, advanced OLA settings.
- CPM settings: initialisation of the default values, advanced CPM settings.
- Experiment settings: initialisation of the default values, advanced automatic experimentation settings.
- Local controller tuning: sequence of automatic (open- or closed-loop) experimentation, online learning and tuning using the RTM around each local model position.
- Performance verification: sequence of automatic experimentation and performance evaluation using the RTM around each local model position.

The CT relies on the functionality of the RTM wherever possible. However, the PC development environment is more convenient for graphical user interface design and enables better visualisation of results, compared to typical PLC systems.

4. Field test

The ASPECT controller has been tested in several pilot applications, for example on a pH control benchmark (Blažič et al., 2003) and a gas–liquid separator (Kocijan et al., 2003). This section presents a pilot application on an apparatus for testing hydraulic valves, located in a hydraulic equipment production plant. A simplified scheme of the apparatus is shown in Fig. 6. It comprises a boiler with local temperature control, three pumps, a pressure sensor, a valve test stand with a pressure difference sensor, three flow meters that may be connected alternately for different measurement ranges, and an expansion vessel. The pumps P1, P2 and P3 are connected in parallel and may be activated in different combinations so that different flow ranges may be achieved. They are equipped with frequency converters; when switched on, all of them receive the same control signal u.

The apparatus is used to test the valves in a range of controlled operating conditions. The most important control task is to control the pressure difference on the tested valve (p_v) by adjusting the control signal u that is connected to the active pumps. The process is nonlinear and time-varying due to the following:

(a) the steady-state relation between the pressure difference on the valve p_v and the mass flow through the valve Q_m (related to the pump rotation speed ω) is quadratic,
(b) the openness of the valve S_v can be changed during a test, while the signal S_v is generally not available (manual valves), and
(c) different pumps (or combinations of pumps) can be used, according to the size of the valve.

These factors severely affect the process dynamics, which results in the unsatisfactory performance of a previously existing control system based on a fixed PI controller.

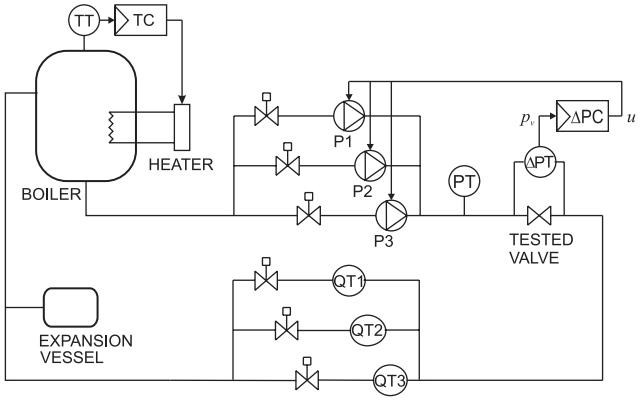

Fig. 6. Apparatus for testing of hydraulic valve (simplified).

Scheduling variable selection is a crucial step when applying a parameter-scheduling controller. While the nonlinearity (a) alone may be easily solved using scheduling from p_v, the condition (b) makes the problem considerably more difficult. Process modelling was used to find a suitable scheduling variable[1].

4.1. Process modelling

A simplified model of the process is a hydraulic loop that comprises a pump (subscript p) as a flow generator, two resistive components, the examined valve (subscript v) and the resistance of the pipeline (subscript l). The inertia of the liquid in the pipeline acts as an inductive component. Using Newton's second law, this is described as

$$\left(p_p - p_v - p_l\right) S = m\frac{\mathrm{d}v}{\mathrm{d}t} = \frac{m}{\rho S}\frac{\mathrm{d}Q_m}{\mathrm{d}t}, \tag{5}$$

where p_p, p_v, and p_l are the pressure differences at the pump, the valve and the pipeline, S is the pipeline cross-section, and m, v, ρ, and Q_m are the mass, velocity, density, and mass flow of the liquid. Applying local linearisation about the operating point $(\overline{p}_p, \overline{p}_v, \overline{p}_l, \overline{Q}_m)$, differentiation of Eq. (5) yields

$$\Delta p_p - \Delta p_v - \Delta p_l = \frac{m}{\rho S^2}\frac{\mathrm{d}\Delta Q_m}{\mathrm{d}t}. \tag{6}$$

Assuming the valve and pipeline resistance characteristic equations

$$Q_m = k_v S_v \sqrt{p_v}, \tag{7}$$

$$Q_m = k_l S \sqrt{p_l}, \tag{8}$$

where k_v and k_l are the valve and pipeline resistance coefficients, Δp_v and Δp_l are calculated using partial derivation about the operating point (Karba, 1999)

$$\Delta p_v = \frac{2\overline{p}_v}{\overline{Q}_m}\Delta Q_m + \frac{-2\overline{p}_v}{\overline{S}_v}\Delta S_v = R_v \Delta Q_m - |b_v|\Delta S_v, \tag{9}$$

$$\Delta p_l = \frac{2\overline{p}_l}{\overline{Q}_m}\Delta Q_m = R_l \Delta Q_m, \tag{10}$$

where the coefficients R_v, R_l and b_v are introduced for shorter notation (b_v is negative). Similarly, R_p and b_p are introduced to describe the pump and may be determined from pump characteristics

$$\Delta p_p = -\left|R_p\right|\Delta Q_m + b_p \Delta\omega. \tag{11}$$

[1]The ASPECT controller is designed to be tuned empirically through experimentation, and process modelling is not generally required.

Inserting Eqs. (9)–(11) into Eq. (6), the dynamic response of the hydraulic loop is described by

$$\frac{1}{\left|R_p\right| + R_v + R_l}\frac{m}{\rho S^2}\frac{\mathrm{d}\Delta Q_m}{\mathrm{d}t} + \Delta Q_m = \frac{1}{\left|R_p\right| + R_v + R_l}\left(b_p\Delta\omega + |b_v|\Delta S_v\right) \tag{12}$$

or, replacing the internal variable ΔQ_m by the process output Δp_v using Eq. (9), by

$$\begin{aligned}\frac{1}{\left|R_p\right| + R_v + R_l}\frac{m}{\rho S^2}\frac{\mathrm{d}\Delta p_v}{\mathrm{d}t} + \Delta p_v &= \frac{R_v}{\left|R_p\right| + R_v + R_l} b_p \Delta\omega \\ &- |b_v|\frac{\left|R_p\right| + R_l}{\left|R_p\right| + R_v + R_l} \\ &\times \left(\frac{1}{\left|R_p\right| + R_l}\frac{m}{\rho S^2}\frac{\mathrm{d}\Delta S_v}{\mathrm{d}t} + \Delta S_v\right).\end{aligned} \tag{13}$$

Additionally, the relation between the pump control signal u and speed ω, describing the dynamics of the pump motor with the frequency converter, must be given. It may be described by a first order lag

$$T_{p1}\frac{\mathrm{d}\Delta\omega}{\mathrm{d}t} + \Delta\omega = b_{p1}\Delta u, \tag{14}$$

where b_{p1} is the gain coefficient and T_{p1} the time constant. Therefore, second order dynamics of the controlled process is assumed.

Observing Eq. (13), $R_v = 2\overline{p}_v/\overline{Q}_m$ is a good scheduling variable candidate because it represents a property of the examined valve at the current operating point that affects both the gain and the time constant of the controlled process. It may be calculated directly from the available process measurements. To improve the stability of the quotient computation at low measurement values, for the actual scheduling variable in the implemented control system the Q_m measurement was replaced by the control signal u that was filtered according to the model in Eq. (14); after division, safety constraints were applied.

4.2. Experimental Trial

Once the scheduling variable is selected, the commissioning of the controller is an empirical procedure, supported by the automatic experimentation functionality of the OS. Firstly, a conventional PI controller is tuned for the Safe mode so that it maintains stable control over the operating region. Then, the local model/controller positions are selected. In this application, a default equidistant distribution of six positions over the operating range of s is used.

Because experimentation with the process is allowed, the typical procedure involving experiments around each local model position is used. In practice, this is the simplest way to ensure proper excitation of the signals. Using the Safe mode, the process is consecutively brought to each of the positions, where auto-tuning experiments are activated by the push of a button. The OS conducts a mode switch (open-loop experiments are preferred), injects the excitation signal containing four step changes, invokes the identification-tuning procedure at the end of the experiment and restores the original mode. For the first two local models, the excitation signal amplitude is 4%. Due to lower process gain it is increased to 8% for other local models, to improve the signal-to-noise ratio. An overview of the MIA status shows that all local models have been identified successfully. The Auto mode is configured after the engineer confirms the new controller

Table 1
Scheduling variable positions, discrete-time local model parameters, derived local model parameters, and local controller parameters

Local model positions	Local model parameters OLA model parameters ($T_{\text{samp}} = 0.5$ s)						Derived parameters[a]				Local controller parameters	
$s = \alpha\frac{\bar{y}}{\bar{u}}$	du	b_1	b_2	a_1	a_2	r	K_{ol}	$T_{90\%}$	T_1	T_2	Kp	Ti
0.10	1	0.002	0.009	−1.212	0.256	−0.002	0.25	20.0	8.01	0.38	5.78	8.03
0.26	1	0.003	0.014	−1.369	0.410	−0.002	0.41	16.5	6.55	0.61	2.58	6.56
0.42	1	0.004	0.017	−1.392	0.432	−0.005	0.52	16.5	6.41	0.66	1.95	6.29
0.58	1	0.003	0.022	−1.401	0.442	−0.005	0.61	15.5	6.09	0.68	1.54	6.12
0.74	1	0.001	0.026	−1.471	0.508	−0.008	0.73	15.0	5.77	0.85	1.14	5.92
0.90	1	-0.001	0.044	−1.397	0.440	−0.019	1.00	15.0	5.80	0.69	0.86	5.97

[a] $T_{90\%}$ is the rise time from 0 to 90% of open-loop step response in s, T_1 and T_2 are the denominator time constants of the continuous-time equivalent model in s, and K_{ol} is the open-loop model gain. $\alpha = 1.89$ is a scaling factor.

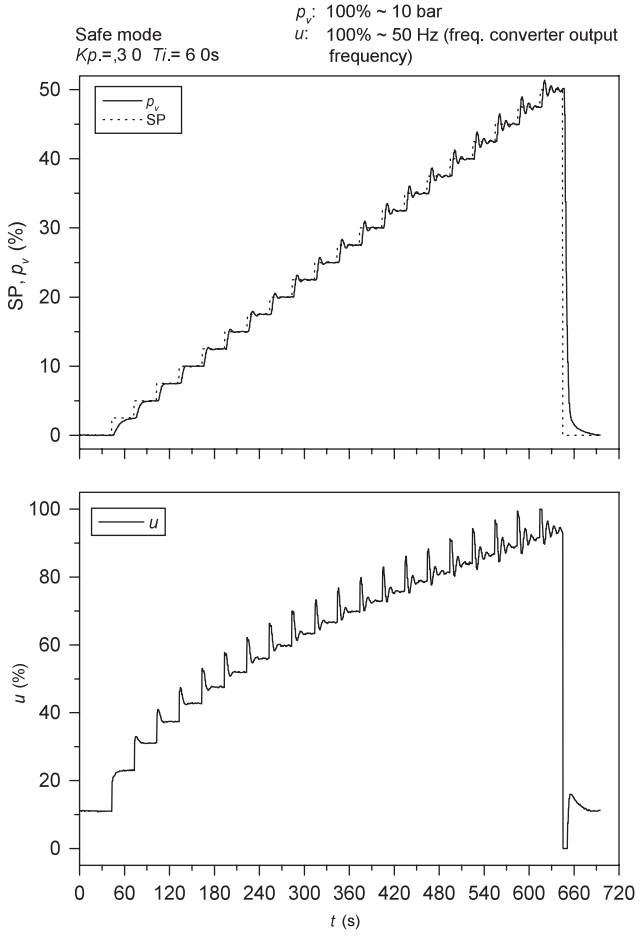

Fig. 7. Control of pressure difference p_v using the PI controller.

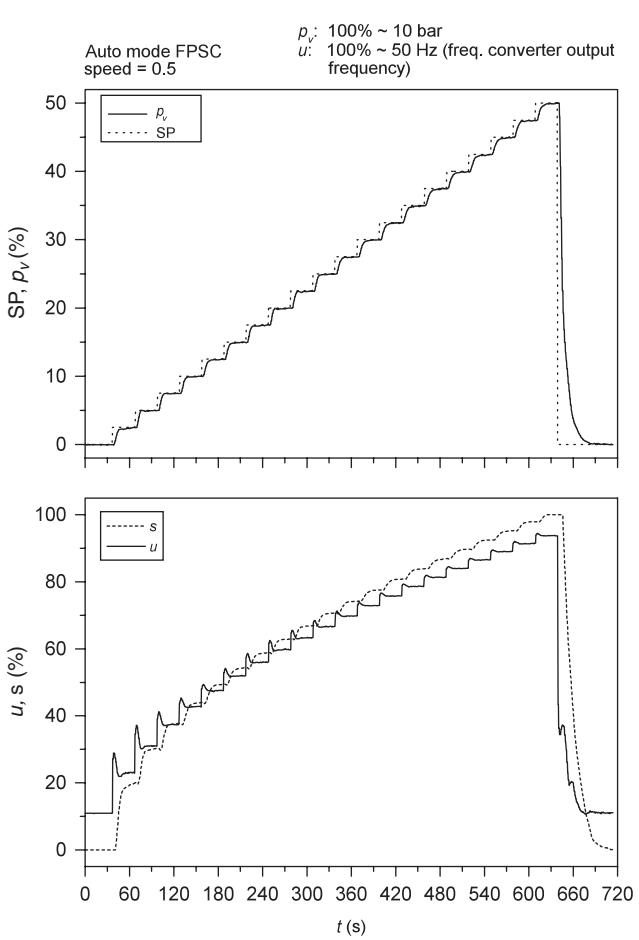

Fig. 8. Control of pressure difference p_v using the FPSC algorithm.

parameters. Table 1 displays the results of this experimental tuning procedure on the process.

The open-loop gain of the local models obviously rises with s (and R_v) in full accordance with the model in Eq. (13), which results in decreasing of Kp. The changes of the identified time constant T_2 are less obvious and do not appear to be consistent with the dependence of the time constant in Eq. (13) on R_v, however R_l and $|R_p|$ vary as well but are not measurable. The variation of Ti mostly results from the changes in T_1, which is associated wtih the pump dynamics T_{f1}. There is considerable but acceptable difference in Kp between the first two local controllers. The differences between the local controller parameters in the higher range of s are small; fewer local controllers could be used in that region.

The control performance is shown for the PI controller, realised using the safe mode of the ASPECT controller, and the FPSC controller. Fig. 7 shows the measured process response to a sequence of step changes of the set-point signal over the whole operating range when using the PI controller. The pressure difference at the valve p_v and the set-point (SP) are shown above, and the pump control signal u below. The parameters of the PI controller were determined so that optimal response was achieved at lower values of p_v. As the pressure increased, the response became oscillatory.

Fig. 8 shows the response when using the FPSC control algorithm. In addition to the signals shown in Fig. 7, the scheduling variable s is also shown in the bottom graph of Fig. 8. In comparison to the PI controller of Fig. 7, this response is optimal over the entire operating region.

4.3. Test platform

Despite the careful selection and modification of the algorithms to reduce computational demand, the OLA and CPM modules are not suitable for implementation in typical PLCs. A DSP or open controller add-on module tends to be a more cost-effective solution than an upper-market PLC. The test platform running the RTM of the ASPECT controller in this pilot application consisted of a Mitsubishi A1S series PLC with an INEA IDR SPAC20 coprocessor, based on the Texas Instruments DSP TMS320C32 at 40 MHz with 2MB of RAM, and a Mitsubishi MAC E700 HMI unit.

The RTM is an extension for INEA IDR BLOK, a graphical development tool for closed-loop control applications in the process industry using Mitsubishi

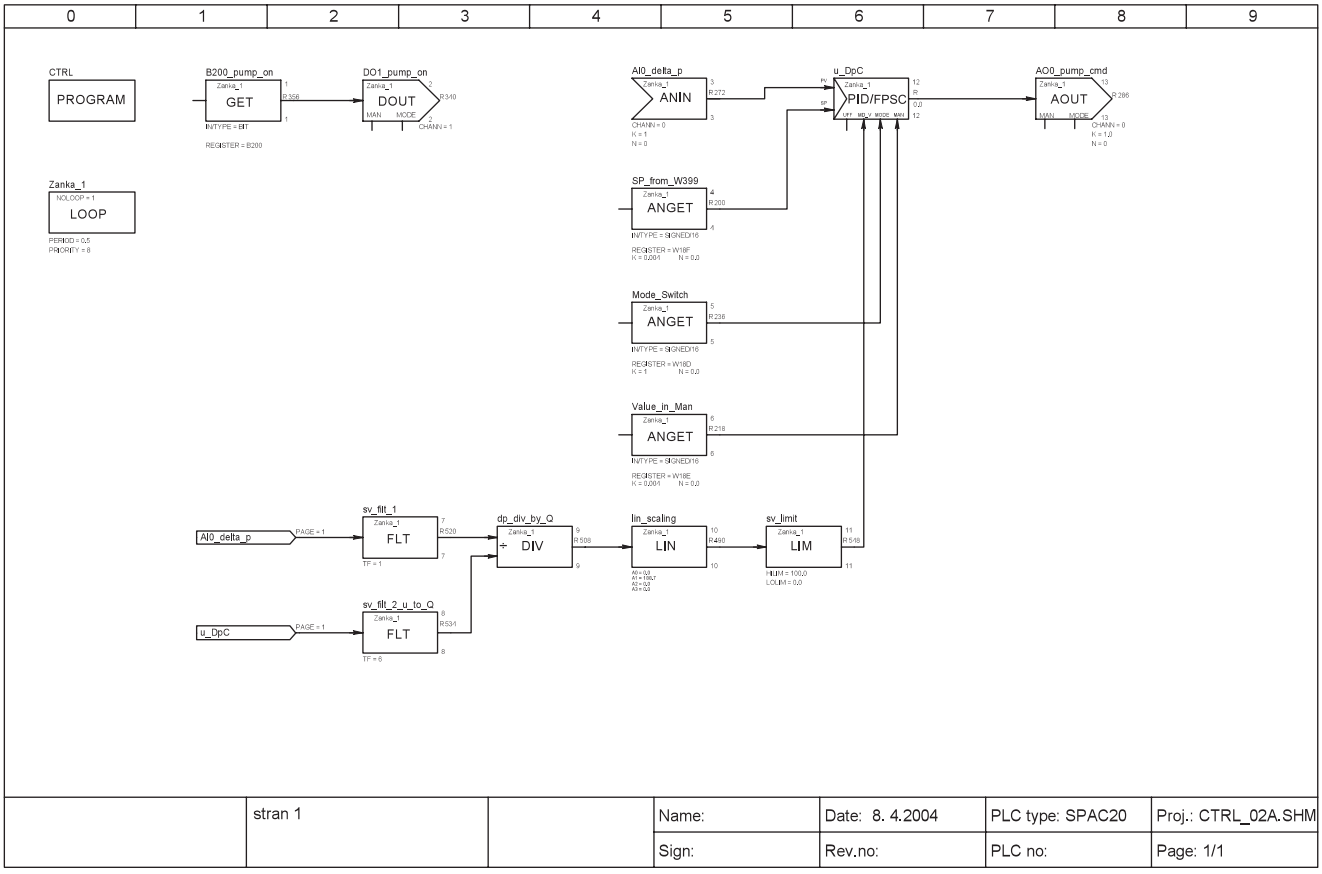

Fig. 9. Valve pressure control scheme designed using IDR BLOK.

Electric MELSEC AnSH PLC controllers (INEA d.o.o., 2001). The FPSC algorithm is included as an additional controller block "PID/FPSC". Other RTM components are implemented as separate PLC tasks, coded in C and downloaded to the PLC in compiled form. They are supervised using the HMI unit through a hierarchical set of menus (such as: Operator display, Trends display, Settings overview, Experiment settings, Online learning settings, Model parameters, Model status, FPSC settings, FPSC parameters, etc.).

Fig. 9 displays the valve pressure control scheme designed in the IDR BLOK environment. Unfortunately, the current version does not yet indicate the association of the PID/FPSC controller block with the supervisory tasks of the RTM.

5. 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 PI controller, an expected considerable improvement in the control performance was achieved using the FPSC control algorithm. Moreover, this performance was easily achieved in practice with self-tuning using the online learning procedure, by performing a sequence of short experiments around a few operating points. The modular multi-agent structure contributes to remarkable flexibility of the control system, so that it is easily reconfigured for various requirements. The work is currently directed towards further improvements in the simplicity of use and ease of application to new processes. A promising future research direction is the incorporation of the pattern recognition techniques from the CPM in data evaluation in the online learning procedure, which may result in a positive contribution to the reliability of adaptive control.

Acknowledgment

The authors would like to acknowledge the contributions of all other project team members. The ASPECT project was financially supported by EC under contract IST-1999-56407. ASPECT ©2002 software is the property of INEA d.o.o., Indelec Europe S.A., and Start Engineering JSCo. Patent pending PCT/SI02/00029.

References

Babuška, R., Oosterhoff, J., Oudshoorn, A., & Bruijn, P. M. (2002). Fuzzy self-tuning PI control of pH in fermentation. *Engineering Applications of Artificial Intelligence*, *15*, 3–15.

Bequette, B. W. (1991). Non-linear control of chemical processes: A review. *Industrial & Engineering Chemistry Research*, *30*, 1391–1413.

Blažič, S., Škrjanc, I., Gerkšič, S., Dolanc, G., Strmčnik, S., Hadjiski, M. B., & Stathaki, A. (2003). On-line Fuzzy Identification for Advanced Intelligent Controller. Proceedings of the IEEE international conference on industrial technology ICIT 2003. Maribor, Slovenia, (pp. 912–916).

Dougherty, D., & Cooper, D. J. (2003). A practical multiple model adaptive strategy for multivariable model predictive control. *Control Engineering Practice*, *11*, 649–664.

Gundala, R., Hoo, K. A., & Piovoso, M. J. (2000). Multiple Model Adaptive Control Design for a Multiple-Input Multiple-Output Chemical Reactor. *Industrial & Engineering Chemistry Research*, *39*, 1554–1564.

Hägglund, T., & Åström, K. J. (2000). Supervision of adaptive control algorithms. *Automatica*, *36*, 1171–1180.

Henson, M. A., & Seborg, D. E. (1994). Adaptive non-linear control of a pH neutralisation process. *IEEE Transactions on Control Systems Technology*, *2*, 169–182.

Henson, M. A., & Seborg, D. E. (1997). *Non-linear process control*. Upper Saddle River NJ: Prentice-Hall PTR.

INEA d.o.o. (2001). *IDR BLOK Process control tools for Mitsubishi electric PLC's, user's manual*, *Ver. 4.20*. Ljubljana: INEA d.o.o.; http://www.inea.si/index.php?kategorija=158.

Kocijan, J., Žunič, G., Strmčnik, S., & Vrančić, D. (2002). Fuzzy gain-scheduling control of a gas+liquid separation plant implemented on a PLC. *International Journal of Control*, *75*(14), 1082–1091.

Kocijan, J., Vrančić, D., Dolanc, G., Gerkšič, S. Strmčnik, S., Škrjanc, I., Blažič, S., Božiček, M., Marinšek, Z., Hadjinski, M. B., Boshnakov, K., Stathaki, A., & King, R. (2003). Auto-tuning non-linear controller for industrial use. *Proceedings of the IEEE international conference on industrial technology ICIT 2003*. Maribor, Slovenia, (pp. 906-910).

Karba, R. (1999). Modeliranje procesov = *process modelling (in Slovenian)*. Ljubljana: Faculty of Electrical Engineering.

Leith, D. J., & Leithead, W. E. (1998). Appropriate realisation of MIMO gain-scheduled controllers. *International Journal of Control*, *70*(1), 13–50.

Ljung, L. (1987). *System Identification*. Englewood Cliffs, NJ: Prentice-Hall.

Murray-Smith, R., & Johansen, T. A. (Ed.) (1997). *Multiple model approaches to modelling and control*. London, Bristol: Taylor & Francis.

Seborg, D. E. (1999). A perspective on advanced strategies for process control (revisited). In P. M. Frank (Ed.), *Advances in control, highlights of ECC'99,* (pp. 103–134). London: Springer.

Stephanopoulus, G., Henning, G., & Leone, H. (1990). Model LA—a modelling language for process engineering, II. Multifaceted modelling of processing systems. *Computers & Chemical Engineering*, *14*, 847–869.

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.

Tan, S., Hang, C.-C., & Chai, J.-S. (1997). Gain scheduling: From conventional to neuro-fuzzy. *Automatica*, *33*(3), 411–419.

Vrančić, D., Strmčnik, S., & Juričić, Đ. (2001). A magnitude optimum multiple integration tuning method for filtered PID controller. *Automatica*, *37*, 1473–1479.

Whiteley, A. L. (1946). Theory of servo systems with particular reference to stabilization. *The Journal of IEE, Part II*, *93*(34), 353–372.

Wooldridge, M., & Jennings, N. R. (1995). Intelligent agents theory and practice. *Knowledge Engineering Review*, *10*(2), 115–152.