15th Triennial World Congress, Barcelona, Spain

NONLINEAR INTERNAL MODEL CONTROL FOR MISO SYSTEMS BASED ON LOCAL LINEAR NEURO-FUZZY MODELS

Alexander Fink[*], **Oliver Nelles**[†], **and Rolf Isermann**[*]

[*]*Darmstadt University of Technology, Institute of Automatic Control*
Laboratory of Control Engineering and Process Automation
Landgraf-Georg-Str. 4, D-64283 Darmstadt, Germany
Phone: +49 6151 162114   Fax: +49 6151 293445
Email: {AFink, RIsermann}@iat.tu-darmstadt.de

[†]*SiemensVDO Automotive, AT PT DTS FDC*
Osterhofener Str. 14, D-93055 Regensburg, Germany
E-mail: Oliver.Nelles@gmx.de

Abstract: The internal model control (IMC) scheme has been widely applied in the field of process control. So far, IMC has been mainly applied to linear processes. This paper discusses the extension of the IMC scheme to nonlinear processes based on local linear models where the properties of linear design procedures can be exploited. The IMC scheme results in controllers that are comparable to gain-scheduled PI or PID controllers which are the standard controllers in process industry. In practice, the tuning of conventional PI or PID controllers can be very time-consuming whereas the IMC design procedure is very simple and reliable. In this paper, the design effort of the IMC and conventional controller design methods will be discussed and control results will be compared by application to nonlinear control of an industrial-scale heat exchanger. *Copyright © 2002 IFAC*

Keywords: Nonlinear Control, Internal Model Control (IMC), Local Linear Models, Neuro-Fuzzy Models, MISO Systems, Heat Exchanger

1. INTRODUCTION

In process control, internal model control (IMC) (Morari and Zafiriou, 1989) has gained high popularity due to the good disturbance rejection capabilities and robustness properties of the IMC structure. Furthermore, the controller design is simple and straightforward such that the controller can easily be tuned by the process engineer. The IMC controller design is theoretically well explored for linear processes. In practice, however, almost every process displays nonlinear behavior especially if it is driven in a wide operating range. Hence, the need emerges to extend the linear design procedure to nonlinear systems (Economou *et al.*, 1986).

Neural networks as well as fuzzy systems have been widely employed for the representation of nonlinear systems and the idea of internal model control can be combined with these types of models (Hunt *et al.*, 1992). In this paper, a local linear neuro-fuzzy model will be used (Murray-Smith, 1994; Nelles, 2000; Takagi and Sugeno, 1985), see Fig. 1. Here, the output of the model is calculated as an interpolation of locally valid linear models. Local model architectures have several advantages over conventional black-box approaches. On the one hand, their transparent architecture allows the combination of prior knowledge about the process with measured identification data. On the other hand, classical linear control design methods can be utilized for nonlinear controller design (Babuška

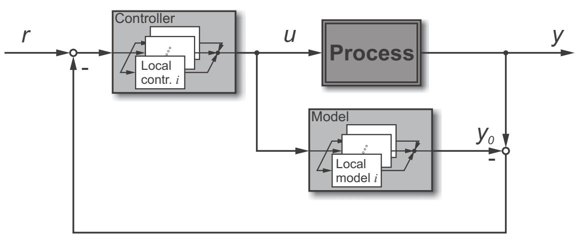


Fig. 1. Local model internal model control.

and Verbruggen, 1996; Hunt and Johansen, 1997). For the IMC approach, this local linearity can be exploited directly and a nonlinear gain-scheduled controller can be obtained.

It can be shown that for simple linear models, the IMC loop can be transferred to classical feedback structures resulting in e.g. PI or PID controllers (Morari and Zafiriou, 1989). In practice, the tuning of conventional PI or PID controllers can be very time-consuming compared to the straightforward IMC design. Here, the IMC scheme can be exploited to evaluate parameters for classical controllers.

The paper is organized as follows. First, local model networks are briefly introduced in Section 2. Then, the extension of linear design techniques to local model systems is discussed in Section 3. Here, the internal model control scheme will be described and compared to conventional PI and PID controller design. The reliability and effectiveness of the presented IMC method is shown by application to nonlinear temperature control of an industrial-scale heat exchanger in Section 4. A summary and conclusions are given in Section 5.

2. LOCAL LINEAR NEURO-FUZZY MODELS

In the following, the modeling of nonlinear dynamic processes using local linear neuro-fuzzy models is described. A large class of multiple-input single-output (MISO) nonlinear dynamic processes with measurable disturbances can be described in discrete-time domain by

$$y(k) = f(\boldsymbol{\varphi}(k)) \,. \tag{1}$$

The regression vector

$$\begin{aligned}\boldsymbol{\varphi}(k) = [&u_1(k-d_{u1}-1) \ \dots \ u_1(k-d_{u1}-nu_1) \ \dots \\ &u_p(k-d_{up}-1) \ \dots \ u_p(k-d_{up}-nu_p) \\ &n_1(k-d_{n1}-1) \ \dots \ n_1(k-d_{n1}-nn_1) \ \dots \\ &n_m(k-d_{nm}-1) \ \dots \ n_m(k-d_{nm}-nn_m) \\ &y(k-1) \ \dots \ y(k-ny)]^T\end{aligned} \tag{2}$$

is composed out of previous process inputs u_i, $i=1\dots p$, measurable disturbances n_j, $j=1\dots m$, as well as previous process outputs y. The dead times of the input and the disturbances are denoted by d_{ui} and d_{nj}, respectively. nu_i, nn_j, and ny are the dynamic orders of u_i, n_j, and y. The restriction to single-output systems in (1) and (2) is made with respect to the presented application. The extension to multi-output systems can be realized by simply adding more outputs to the regressors.

Throughout this contribution, the unknown nonlinear function $f(\cdot)$ in (1) is approximated by a local linear neuro-fuzzy model. This model can be interpreted as a local model neural network with linear models on the one hand (Murray-Smith, 1994; Nelles, 2000), or as a Takagi-Sugeno fuzzy model (Takagi and Sugeno, 1985) which is characterized by rule consequents that are linear functions of the input variables on the other hand. For the latter case, the rule base comprises M rules of the form

$$\begin{aligned}&R_j: \text{ IF } z_1 \text{ is } A_{j,1} \text{ AND } \dots \text{ AND } z_{nz} \text{ is } A_{j,nz} \\ &\text{THEN } y_j(k) = w_{j,0} + w_{j,1}x_1 + \dots + w_{j,nx}x_{nx} \,,\end{aligned} \tag{3}$$

$j = 1\dots M$, where $A_{j,i}$ is a fuzzy set defined on the universe of discourse of the input i. Both, the nz-dimensional vector $\mathbf{z}(k) = [z_1 \ z_2 \ \cdots \ z_{nz}]^T$ in the rule premise and the nx-dimensional vector $\mathbf{x}(k) = [x_1 \ x_2 \ \cdots \ x_{nx}]^T$ in the consequent contain subsets of the elements of $\boldsymbol{\varphi}(k)$ in (2). The rule consequents represent affine difference equations which are linear in the parameters $w_{j,i}$. The additional constants $w_{j,0}$ define the operating points.

By choosing the product operator as t-norm, the output $y(k)$ of the system with M local models can be aggregated as

$$y = \sum_{j=1}^{M} (w_{j,0} + w_{j,1}x_1 + \dots + w_{j,nx}x_{nx}) \cdot \Phi_j(\mathbf{z}) \tag{4}$$

where Φ_j denote the normalized validity functions

$$\Phi_j(\mathbf{z}, \mathbf{c}_j, \sigma_j) = \frac{\mu_j}{\sum_{j=1}^{M} \mu_j} \,. \tag{5}$$

The membership functions μ_j are chosen as Gaussian functions with centers $\mathbf{c}_j$ and standard deviations $\boldsymbol{\sigma}_j$

$$\begin{aligned}\mu_j = &\exp\left(-\frac{1}{2}\frac{(z_1 - c_{i,1})^2}{\sigma_{i,1}^2}\right) \cdot \cdots \\ &\cdots \cdot \exp\left(-\frac{1}{2}\frac{(z_{nz} - c_{i,nz})^2}{\sigma_{i,nz}^2}\right) \,.\end{aligned} \tag{6}$$

μ_j is the degree of fulfillment of rule R_j and Φ_j can be interpreted as the activation of the j-th local model in the sense of a local model network.

The task of local model identification is to determine both, the nonlinear parameters of the Gaussian membership functions, namely the centers $\mathbf{c}_j$ and the standard deviations $\boldsymbol{\sigma}_j$, and the linear parameters of the local models. For that matter, various approaches have been proposed (Babuška and Verbruggen, 1996) which compute nonlinear dynamic models from input-output measurement data, e.g., fuzzy clustering, tree construction algorithms, or neuro-fuzzy approaches. In this paper, the LOLIMOT (local linear model tree) algorithm (Nelles, 2000) is applied for nonlinear model identification. This algorithm iteratively splits up the input space into regions where linear models are estimated.

3. LOCAL MODEL CONTROL DESIGN

In this section, the controller design procedure for local models will be described. Then, the design procedure of conventional controllers will be briefly discussed and the internal model control design procedure will be explained in more detail. Both design techniques can be extended to local model systems.

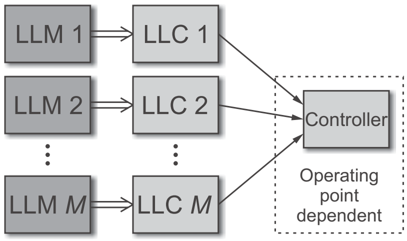


Fig. 2. Local model based controller design. (LLM: local linear model, LLC: local linear controller)

There are generally two control design approaches for local model networks that can be pursued, the linearization and the local model based controller design (Fink *et al.*, 1999). The linearization based design is a general approach and can be utilized for any nonlinear model. Here, the model is linearized for each sampling instant at the current, dynamic operating point and subsequently a linear controller is designed online based on that linearized model (Sousa *et al.*, 1997). For the local model based approach, linear controllers are designed offline for each local model separately and later merged to one operating point dependent controller by weighting the local linear controllers according to the actual operating point (Xie and Rad, 2000). This approach is pursued in this paper, see Fig. 2. The local model based concept is also called parallel distributed compensation (PDC). For this approach, powerful methods for proving closed-loop stability via solving linear matrix inequalities (LMIs) are available for local model control design (Wang *et al.*, 1996). The interpolation of the controller parameters closely follows the idea of model based adaptive control or the gain-scheduling control approach (Hunt and Johansen, 1997).

3.1 Conventional PI and PID Control Design

The standard controllers in process industry are linear PI and PID controllers (Isermann, 1981)

$$G_{\mathrm{PI}}=\frac{q_0+q_1 z^{-1}}{1-z^{-1}}\,,\quad G_{\mathrm{PID}}=\frac{q_0+q_1 z^{-1}+q_2 z^{-2}}{1-z^{-1}}\,. \tag{7}$$

These controllers can manually be tuned by changing the values of the controller parameters q_i. For the PID controller, for instance, three parameters have to be tuned. This can be a very lengthy procedure since the influence of the controller parameters on the control performance cannot easily be described. Also, the parameters itself do not have a physical interpretation which makes the tuning non-intuitive. In order to assist the process engineer, the controller parameters can be automatically optimized based on a given process model by minimizing a predefined cost function, e.g. the quadratic cost function

$$J=\sum_{k=0}^{\infty} e(k)^2+\beta\sum_{k=0}^{\infty}\Delta u(k)^2\,, \tag{8}$$

which expresses the control objectives. The first sum penalizes the control error $e(k) = r(k) - y(k)$. The second term saves the actuator by preventing the manipulated variable u from performing vast changes. β is the penalty factor for changes of the manipulated variable Δu. The optimization of the cost function (8), however, is in general nonlinear and computationally expensive. Also, the choice of the tuning parameter β is strongly dependent on the process to be controlled and hence difficult to adjust.

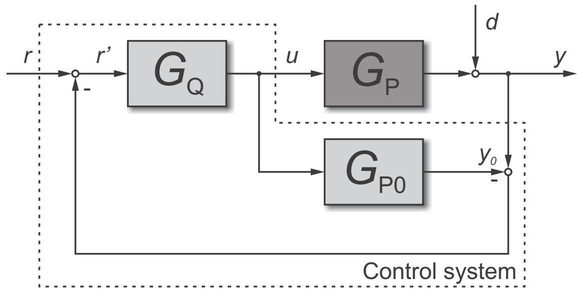


Fig. 3. Internal model control (IMC) scheme.

In case of local model systems, local conventional controllers can individually be designed and blended to an overall nonlinear gain-scheduled controller.

3.2 Internal Model Control Design

Figure 3 shows the standard linear IMC scheme (Morari and Zafiriou, 1989) where the process model G_{P0} plays an explicit role in the control structure compared to the standard control loop. The IMC structure has some advantages over conventional feedback control loops. For the nominal case $G_{\mathrm{P}} = G_{\mathrm{P0}}$, for instance, the feedback is affected only by the disturbance d such that the system is effectively open loop and hence no stability problems can arise. Also, if the process G_{P} is stable, which is true for most industrial processes, the closed loop will be stable for any stable controller G_{Q}. Furthermore, the controller G_{Q} can simply be designed as a feedforward controller in the IMC scheme.

The IMC design procedure consists of two steps (Morari and Zafiriou, 1989). Since G_{Q} can be designed as an open-loop controller, the ideal choice for the controller is the inverse of the process model

$$G'_{\mathrm{Q}} = G_{\mathrm{P0}}^{-1}\,, \tag{9}$$

which will yield good tracking and disturbance rejection. Then, the controller is detuned for robustness to account for a possible plant-model-mismatch

$$G_{\mathrm{Q}} = G'_{\mathrm{Q}} G_{\mathrm{F}}\,. \tag{10}$$

In this second step, the controller is augmented with the low-pass filter

$$G_{\mathrm{F}} = \left(\frac{1-\lambda}{z-\lambda}\right)^n \tag{11}$$

to reduce the loop gain for high frequencies. The filter order n is chosen such that the controller G_{Q} becomes causal as the pure inverse of the model is usually not physically realizable. The inversion of the process model may also lead to unstable controllers in case of

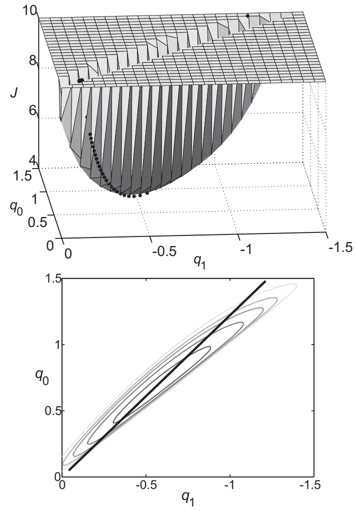


Fig. 4. Cost function for a PI controller and the IMC. (The function values are saturated at 10.)

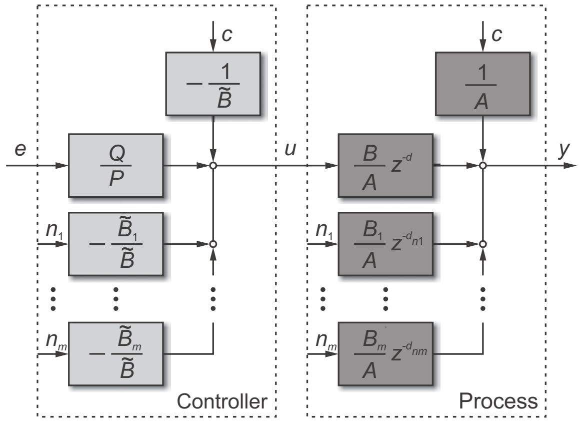


Fig. 5. Controller with offset compensation and compensation of measurable disturbances for one local linear model.

unstable zeros in the model. For this matter, several approaches have been developed to invert the model, e.g. following the H_2-optimality principle (Morari and Zafiriou, 1989) or performing the inversion such that there is zero phase-error tracking (Tomizuka, 1987). In this paper, the inversion (9) is performed according to the H_2-optimality criterion such that there is no amplitude error but a phase error. The time constant λ of the filter G_{F} is the only parameter that has to be tuned in the IMC design procedure. With this parameter, the desired closed-loop behavior can be defined. Consequently, the IMC design process is very simple compared to other design methods.

For nonlinear systems, the IMC approach can be extended to nonlinear models (Economou *et al.*, 1986; Hunt *et al.*, 1992). In general however, the inversion of nonlinear models is more involved and analytical solutions may not exist such that solutions have to be found numerically. In the case of local linear networks, the linearity can be exploited directly and an analytical solution can be obtained (Babuška and Verbruggen, 1996; Sousa *et al.*, 1997; Xie and Rad, 2000). Following the local model approach, a separate inversion of each local linear model is performed. The control output is then calculated as a weighted sum of locally valid linear controllers yielding a globally nonlinear gain-scheduled controller. Clearly, the weighted average of the local inverse is generally not equal to the global inverse. For this application, however, the difference is negligible and the computation effort of the local model approach is smaller than calculating the global inverse (Fink *et al.*, 1999).

It can be shown that the IMC loop in Fig. 3 can be transferred into a classical feedback control loop (Morari and Zafiriou, 1989). For instance, if the process model is a first order time-lag system, the resulting controller behavior is a PI controller and if the model is of second order, it will become a PID controller. Hence, the IMC design procedure can be utilized to design and tune conventional controllers. In practical applications, the controller tuning process should be as simple and straightforward as possible. However, manual tuning of conventional controllers can be very time-consuming since the dependence of the control performance on the controller parameters is not intuitive. On the other hand, the IMC design procedure is very simple and reliable since only one parameter has to be tuned. Additionally, this parameter is the time constant of the low-pass filter and has a physical meaning. It describes the desired closed-loop dynamics of the system. Figure 4 demonstrates the differences of the manual tuning process. It shows a plot of the cost function (8) for the two PI controller parameters q_0 and q_1. In order to obtain a PI controller also for the IMC design, the system used in this example was chosen as a first order time-lag system. For a good control performance, one has to adjust the two controller parameters in order to reach the minimum of the cost function. In comparison to that, only one parameter has to be tuned for the IMC design. The crosses on the surface plot mark the positions of the resulting parameters if the time constant of the filter is changed. The bottom plot shows the contour lines of the cost function. The straight line depicts the controller parameters that can be reached with the IMC procedure. It can be seen that this line passes through the vicinity of the minimum of the cost function. Hence, the manual tuning effort can be considerably reduced by using the IMC design procedure. It should be noted that the minimum that can be reached with the IMC procedure depends on the quality of the model. Hence, some effort should be put into finding a good model.

In case of systems with measurable disturbances and operating point dependent offsets, these influences have to be compensated in the control loop. In particular for the IMC scheme, this compensation has to be implemented in the inversion procedure. Figure 5 shows the complete compensation scheme for one local linear model. On the right, the transfer functions represent the process described by (2) and (4) for one local linear model. The polynomials B, B_i, and A and the offset c are constructed out of the parameters $w_{j,i}$ in (4). With respect to the later application example

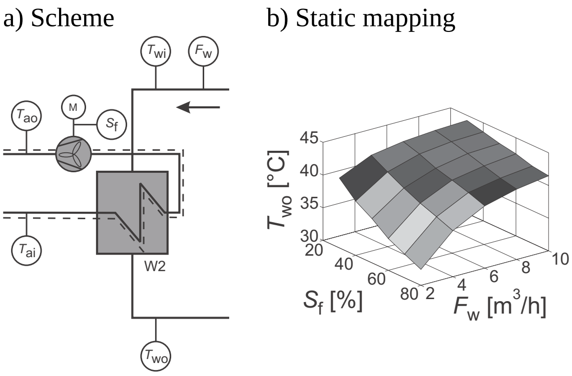


Fig. 6. Water-air cross-flow heat exchanger.

but without loss of generality, the process has only one control input but several measurable disturbances. In order to compensate the offset and the disturbances, the scheme depicted on the left in Fig. 5 has to be implemented. Here, the transfer function Q/P represents the controller which can either be a conventional controller or the IMC G_{Q}. Hence, the signal e is the control error $r - y$ for the conventional controller or the modified reference r' for the IMC. It should be noted that for the IMC the parallel path with the model G_{P0} is not shown in this figure. The other blocks in the controller build the compensator. To obtain stable and causal transfer functions also for these blocks, the inversion routines described for the IMC can be used to generate e.g. B_i/B, the results of this procedure are denoted by a tilde. For the global nonlinear process, the compensators for the local models have to be blended similar to the local controllers.

4. EXPERIMENTAL RESULTS

The discussed nonlinear controller design strategies will be compared by application to temperature control of an industrial-scale heat exchanger. Both, static and dynamic behavior of the heat exchanger strongly depend on the flow rates of the primary and the secondary media. Therefore, the temperature controller encounters strongly varying gains and time constants. Both controller design methods described in Section 3 are used to control the outlet temperature of a water-air heat exchanger.

4.1 Plant Description

Figure 6a depicts the water-air cross-flow heat exchanger under investigation. The incoming water (temperature T_{wi}) is cooled down by cold air (temperature T_{ai}) that is passed through the heat exchanger by a fan. The outlet water temperature T_{wo} is to be controlled by manipulating the fan speed S_{f}. Besides on the manipulated variable S_{f}, the control variable T_{wo} also depends on the measurable disturbances: inlet temperature T_{wi}, air temperature T_{ai}, and water flow rate F_{w}. The latter one highly influences the static behavior as can be seen from the static mapping in Fig. 6b. It is a major challenge to design a temperature controller for T_{wo} when the flow rates vary in a wide range since the process gain changes by a factor of about four. Moreover, different water flow rates induce varying time constants and varying dead times between T_{wi} and T_{wo}.

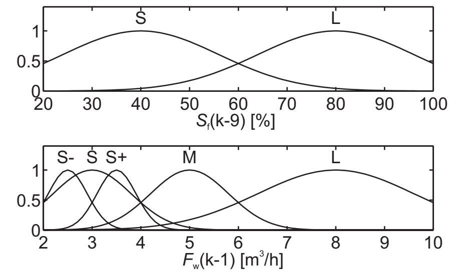


Fig. 7. Membership functions.

4.2 Local Model Neuro-Fuzzy Network

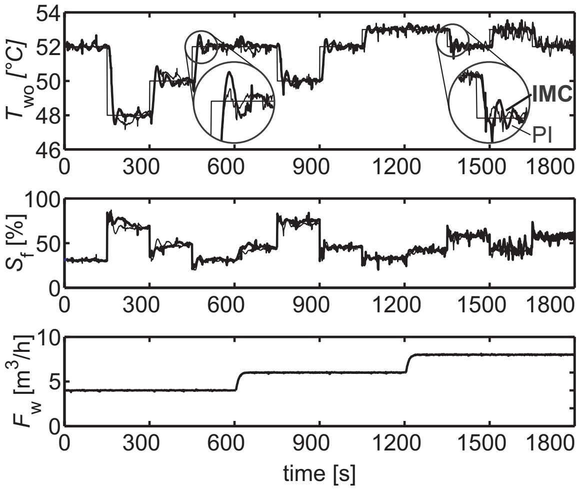
The local model network of the heat exchanger is obtained by using the LOLIMOT algorithm (Fischer *et al.*, 1997). The water inlet temperature is kept at $T_{\mathrm{wi}} = 55°\mathrm{C}$ and the air temperature T_{ai} is about constant at $5°\mathrm{C}$ during the experiment. In the local model network, the water outlet temperature T_{wo} depends on the actuation signal S_{f}, the water flow rate F_{w}, and the water inlet temperature T_{wi}. The LOLIMOT algorithm was provided with the regressors

$$\begin{aligned}\mathbf{x}(k) = [&S_{\mathrm{f}}(k-9)\\ &F_{\mathrm{w}}(k-1)\\ &T_{\mathrm{wi}}(k-2)\ T_{\mathrm{wi}}(k-4)\ \ldots\ T_{\mathrm{wi}}(k-20)\\ &T_{\mathrm{wo}}(k-1)\ 1]^T\end{aligned} \tag{12}$$

for the fuzzy rule consequents. By incorporating $S_{\mathrm{f}}(k-9)$ and $T_{\mathrm{wo}}(k-1)$ in $\mathbf{x}$, first order dynamics with a dead time of $d_1 = 8$ are assumed based on knowledge from prior step response experiments. The regressor 1 in $\mathbf{x}$ represents the offset of the local models. The LOLIMOT algorithm constructed a model with five rules with the following antecedents

$$\begin{aligned}&R_1 : \text{IF } F_{\mathrm{w}}(k-1) = \text{S-} \text{ AND } S_{\mathrm{f}}(k-9) = \text{S}\\ &R_2 : \text{IF } F_{\mathrm{w}}(k-1) = \text{S+} \text{ AND } S_{\mathrm{f}}(k-9) = \text{S}\\ &R_3 : \text{IF } F_{\mathrm{w}}(k-1) = \text{S} \text{ AND } S_{\mathrm{f}}(k-9) = \text{L}\\ &R_4 : \text{IF } F_{\mathrm{w}}(k-1) = \text{M}\\ &R_5 : \text{IF } F_{\mathrm{w}}(k-1) = \text{L}\,.\end{aligned} \tag{13}$$

The corresponding membership functions are shown in Fig. 7. It can be observed that the membership functions are denser for lower water flow rates. This reflects the fact that the nonlinearity of the process is stronger in this operating regime. Furthermore, the manipulated variable S_{f} has significant nonlinear influence only for low flow rates, see rules R_1, R_2, and R_3. This interpretation conforms to the static mapping in Fig. 6b. For more sophisticated control approaches, e.g. model predictive control, more complex neuro-fuzzy models have been identified (Fischer *et al.*, 1997). In this study, however, the described model is sufficient because of the simple control structure of the IMC and PI controller.



a) control variable: water outlet temperature T_{wo},
b) control input: fan speed S_{f},
c) disturbance: water flow rate F_{w}.

Fig. 8. Closed-loop control performance.

4.3 Comparison of the controllers

Fig. 8 shows the closed-loop control performance for the IMC and the conventional controller. For a first order system, the IMC results in a PI controller. Hence, a standard PI controller is used for the conventional controller in this comparison. In this example, however, the process has a dead time which is accounted for in the IMC scheme. This automatically results in a more complex controller similar to the idea of the smith-predictor-controller for the IMC design. In Fig. 8a, the control variable T_{wo} and the reference trajectory are plotted. The corresponding control input S_{f} is shown in Fig. 8b. The disturbance F_{w} shown in Fig. 8c covers the operating range from $4\mathrm{m}^3/\mathrm{h}$ to $8\mathrm{m}^3/\mathrm{h}$. For both controllers, the control variable follows the reference well and the manipulated variable has overshoots for steps in the reference. For changes of the water flow rate F_{w}, the controller has to compensate the influence of this disturbance. The control performance in this case is satisfactory, too. The IMC was tuned by changing the time constant of the low-pass filter. For the PI design, the parameters had to be numerically optimized based on the cost function (8). Here, the penalty factor β had to be tuned in many iterations. This again underlines that the IMC approach is an alternative to conventional controller design by achieving comparable or even better control performance.

5. CONCLUSIONS

In this paper, the extension of the internal model control (IMC) scheme to local linear neuro-fuzzy models has been discussed. It has been shown that the well developed linear design techniques can easily be adapted to this type of model yielding a gain-scheduled controller. The IMC structure can be converted into a standard control loop where the resulting controller is mainly a PI or PID controller. Consequently, the internal model control approach can be utilized to design and tune conventional controllers since the tuning of conventional controllers by a process engineer can be very time-consuming in practice. The IMC design procedure itself has only one tuning parameter and hence is very simple and reliable.

REFERENCES

Babuška, R. and H.B. Verbruggen (1996). An overview of fuzzy modeling for control. *Control Engineering Practice* **4**(11), 1593–1606.

Economou, C.G., M. Morari and B. O. Palsson (1986). Internal model control: Extension to non-linear systems. *Ind. Eng. Chem. Process Des. Dev.* **25**, 403–411.

Fink, A., O. Nelles and M. Fischer (1999). Linearization based and local model based controller design. In: *European Control Conference (ECC)*. Karlsruhe, Germany.

Fischer, M., O. Nelles and R. Isermann (1997). Exploiting prior knowledge in fuzzy model identification of a heat exchanger. In: *IFAC Symposium on Artificial Intelligence in Real-Time Control (AIRTC)*. Kuala Lumpur, Malaysia. pp. 445–450.

Hunt, K.J. and T.A. Johansen (1997). Design and analysis of gain-scheduled control using local controller networks. *International Journal of Control* **66**(5), 619–651.

Hunt, K.J., D. Sbarbaro R. Zbikowski and P.J. Gawthrop (1992). Neural networks for control systems – a survey. *Automatica* **28**(6), 1083–1112.

Isermann, R. (1981). *Digital Control Systems*. Springer. Berlin, Germany.

Morari, M. and E. Zafiriou (1989). *Robust Process Control*. Prentice Hall. Englewood Cliffs, USA.

Murray-Smith, R. (1994). A Local Model Network Approach to Nonlinear Modeling. PhD thesis. University of Strathclyde. Strathclyde, UK.

Nelles, O. (2000). *Nonlinear System Identification*. Springer Verlag. Heidelberg, Germany.

Sousa, J.M., R. Babuška and H.B. Verbruggen (1997). Internal model control with a fuzzy model: Application to an air-condition system. In: *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. Barcelona, Spain. pp. 207–212.

Takagi, T. and M. Sugeno (1985). Fuzzy identification of systems and its application to modelling and control. *IEEE Transactions on Systems, Man, and Cybernetics* **15**(1), 116–132.

Tomizuka, M. (1987). Zero phase error tracking algorithm for digital control. *ASME Journal of Dynamic Systems, Measurement, and Control* **109**(1), 65–68.

Wang, H.O., K. Tanaka and M.F. Griffin (1996). An approach to fuzzy control of nonlinear systems: Stability and design issues. *IEEE Transactions on Fuzzy Systems* **4**(1), 14–23.

Xie, W.F. and A.B. Rad (2000). Fuzzy adaptive internal model control. *IEEE Transactions on Industrial Electronics* **47**(1), 193–202.