Multiple model modeling and predictive control of the pH neutralization process

J. Novák, P. Chalupa and V. Bobál

***Abstract*—**The requirement for improved efficiency and safety induce the need for sophisticated control systems. Model predictive control represents such control method which makes explicit use of a model of the process to obtain the control signal. The performance of control algorithm depends on the quality of the derived model. A possible approach is to decompose the nonlinear dynamics into multiple linear models and switch or interpolate them based on the current operating conditions. Multiple models structure for modeling and control allow the transfer of many methods from the linear control theory to the nonlinear systems. The process operations are partitioned into several operating regions and within each region, a local linear model is developed to approximate the process. To save on computational load, a linear model is obtained by interpolating these linear models at each sample point and then obtained model is used in a Generalized Predictive Control (GPC) framework. The manipulated variable adjustments are computed through optimization at each sampling interval. The proposed identification and control method is illustrated by the simulation study on a nonlinear process.

***Keywords*—**clustering**,** fuzzy modeling, multiple models, nonlinear control, optimization , predictive control.

I. Introduction

MODEL predictive control (MPC) [1], [2] has been a major research topic for the last 30 years. The reason for this is the ability of MPC to optimally control multivariable system under various constraints. The main idea of the MPC is to calculate the actual and the subsequent control signals by minimizing the quadratic deviation of a reference signal and an output signal in a given future horizon. The solution to this optimization problem is the optimal input signal to the system at that particular time. According to the receding horizon control strategy, only the first control signal is used at the process input, and in the next sampling point the procedure is repeated. Conventional MPC techniques are based on the use of linear models. Linear MPCs can yield a satisfactory performance if the process is reasonably linear, or is operated close to the nominal steady state. However, a linear model is not sufficient to capture the properties of chemical engineering processes.

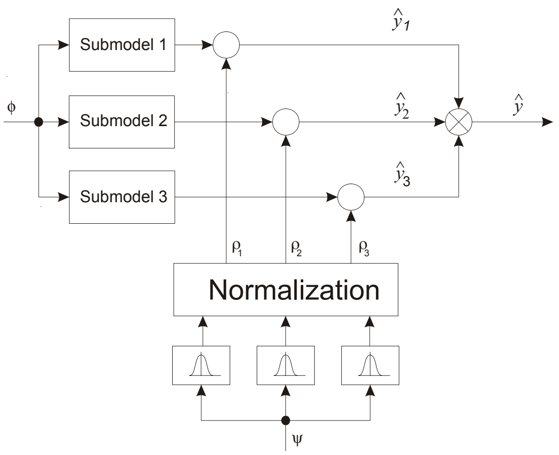
The poor performance of linear MPCs for processes with a strong degree of nonlinearity (for example pH control or batch reactors) has motivated the development of nonlinear model predictive control (NMPC), where a more accurate (nonlinear) model of a plant is used for prediction and optimization. Qin and Badgwell [3] presented a survey of nonlinear model predictive control applications in industry. In NMPC, the importance of having an accurate process model is crucial, and several nonlinear models that have been utilized for NMPC can be found in the literature. In Ref. [4], a Wiener-type nonlinear black box model was developed for capturing the dynamics of open loop stable Multiple Input Multiple Output (MIMO) nonlinear systems with deterministic inputs. The last decade has shown an increase in the use of local model representations of non-linear dynamic systems [5]. A multimodel approach has advantages in controlling industrial processes, especially those with inherent nonlinearity, a wide operating range, or load disturbances. Based on a divide-and-conquer strategy, multimodel approaches can be used to develop local linear models or controllers corresponding to typical operating regimes. The comparison of fuzzy model, Wiener model and nonlinear model predictive control can be found in [6]. Different nonlinear model-based predictive control algorithms Galan et al. [7] reported the real-time implementation of a multilinear model based control strategies for a bench top–scale pH neutralization reactor. In [8] the Takagi-Sugeno-Kang fuzzy-based predictive control of the heat exchanger is presented for control of the heat-exchanger. However, the model is obtained by solving the partial derivative equations with parameters that generally does not have to accurately reflect the properties of the controlled system. The approach developed in [9] uses a bank of local models developed in different operating conditions and switching based on linear differential inclusion. The neural network trained from the input-output data was used in the predictive control scheme to control a continuous stirred tank reactor in [10]. Multiple model predictive control strategy in [11] relies on the use of a bank of linear models to describe the dynamic behavior over a wide operating range. A recursive Bayesian scheme assigns weights to each model. The combined-weighted model is then used for the design of

Manuscript received June 29, 2011. This work was supported by the Czech Science Foundation under the grant 102/09/P243

Jakub Novák is with Tomas Bata University in Zlin, Faculty of Applied Informatics, nam. T.G. Masaryka 5555, 76001 Zlin, Czech Republic (email: jnovak@fai.utb.cz)

Petr Chalupa is with Tomas Bata University in Zlin, Faculty of Applied Informatics, nam. T.G. Masaryka 5555, 76001 Zlin, Czech Republic (email: chalupa@fai.utb.cz)

Vladimír Bobál is with Tomas Bata University in Zlin, Faculty of Applied Informatics, nam. T.G. Masaryka 5555, 76001 Zlin, Czech Republic (email: bobal@fai.utb.cz)



the

Fig. 1 A local model network scheme

controller. Robust Multiple-Model Adaptive Control strategy proposed by Athans and Pekri [12] uses a bank of Kalman Filters and bank of local robust compensators and relies on stochastic processes that provide sufficient excitation for identification. All these approaches used local linear models for controller design. The paper is aimed at the development of such model that combines local models and also optimization of its structure.

The pH neutralization process was chosen as a benchmark for control algorithms in several studies as it exhibits significant nonlinear behavior.

II. Multiple linear models

Modeling nonlinear dynamic systems from observed data and a priori engineering knowledge is a major area of science and engineering. In recent years a great deal of work has appeared in new areas, such as fuzzy modeling and neural networks. Local Model Networks were first introduced by Johansen and Foss [13] to describe a set of submodels, each of which was valid for a specific regime in an operating space, weighted by activation function. A LMN is a generalization of a radial basis function (RBF) network, in which individual neurons are replaced by local submodels with basic functions defining the regions of validity of individual submodels, according to the expected operating regions of a plant.

The LMN output is given by:

$$\hat{y}(k)=\sum_{i=1}^{M}\rho_i(\boldsymbol{\psi}(k))\hat{y}_i(k) \qquad (1)$$

where $\boldsymbol{\psi}(k)$ is a vector of scheduling variables, $\rho_i(k)$ is a normalized validity function, and $\hat{y}_i(k)$ is the output of the *i*-th model. The network that is described by Equation (1) is shown in Fig. 1. The blending of local models is calculated using weighting or validity functions. Although any function with a locally limited activation may be applied as a validity function, a common choice for this function takes the Gaussian form. The validity function for the i-th model is given by:

$$\tilde{\rho}_i(\boldsymbol{\psi}(k)) = \exp\left(-\frac{1}{2}\left(\boldsymbol{\psi}-c_i\right)^T \sigma_i^{-2}\left(\boldsymbol{\psi}-c_i\right)\right) \qquad (2)$$

where the parameters c_i, σ_i, define the Gaussian centre and width, respectively, and the scheduling variable $\psi(k)$ can be a system state or any system variable.

Basically, there are two ways to design controllers for local model structures: the linearization-based and local model-based approaches (Fig. 2). In the linearization-based approach, the local model network is linearized at the current operating point, and the linear controller is designed. The linearization of the LMN is very simple due to the structure of the model. A linear model is obtained by interpolating these linear models at each sample point. In the second approach a local controller is designed for each local model, and the control output is then calculated as an interpolation of the local controller outputs according to the current operating point.

III. pH neutralization process

The system chosen for this study is a pH neutralization process model developed in [14]. The model is used in many studies to test the nonlinear control strategies [15],[16]. The process was modeled using a nonlinear first-principles model, which was computationally too demanding for MPC computations. Therefore it is a good benchmark example for local modeling and control. Based on the time constants of the process the sampling period was chosen to be 15 s. The process consists of an acid (HNO3) stream, a buffer (NaHCO3) stream and a base (NaOH) stream being continually mixed in the tank (Fig. 3). The model is based on the assumptions that the streams are perfectly mixed, and the density is constant throughout the entire tank. The process is aimed at controlling the pH value of the outlet stream by varying the inlet base stream. The outlet flow rate is dependent on the fluid height in the tank as well as the position of the valve.

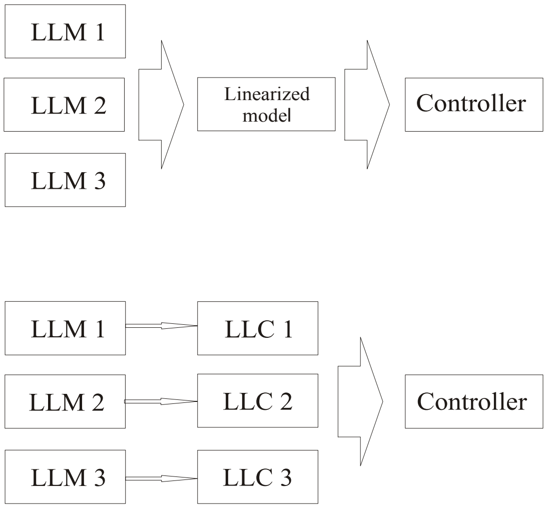


Fig. 2 Controller design using linearization and local models (LLM = local linear model, LLC = local linear controller)

A differential equation that describes the total mass balance of the tank is:

$$\frac{dh}{dt}=\frac{1}{A}\left(Q_1+Q_2+Q_3-c\sqrt{h}\right) \tag{3}$$

where *c* is a valve constant, *A* is the tank cross-sectional area, and *h* is the tank level. The differential equations for the effluent reaction invariants W_a and W_b can be derived as

$$\begin{aligned}\frac{dW_a}{dt}&=\frac{1}{Ah}\begin{pmatrix}Q_1(W_{a1}-W_a)+Q_2(W_{a2}-W_a)\\ +Q_3(W_{a3}-W_a)\end{pmatrix}\\ \frac{dW_b}{dt}&=\frac{1}{Ah}\begin{pmatrix}Q_1(W_{b1}-W_b)+Q_2(W_{b2}-W_b)\\ +Q_3(W_{b3}-W_b)\end{pmatrix}\end{aligned} \tag{4}$$

where W_{ai} and W_{bi} are the chemical reaction invariants of the *i*-th stream. The variables are defined in Table 1.

Table 1 Parameters of the pH neutralization plant

Symbol .	Variable	Nom. value
A	Tank area	*207 cm²*
h	Tank level	*14 cm*
Q_1	Acid flow rate	*16.6 ml/s*
Q_2	Buffer flow rate	*0.55 ml/s*
Q_3	Base flow rate	*15.6 ml/s*
c	Valve constant	$8\ ml/s\sqrt{cm}$
W_{a1}	$[HNO_3]_1$	*0.003 mol*
W_{a2}	$-[NaHCO_3]_2$	*-0.03 mol*
W_{a3}	$-[NaHCO_3]_3-$ $[NaOH]_3$	*-0.00305 mol*
W_{b1}	$[NaHCO_3]_1$	*0 mol*
W_{b2}	$[NaHCO_3]_2$	*0.03 mol*
W_{b3}	$[NaHCO_3]_3$	*0.00005 mol*
pK_{a1}	$-\log_{10}K_{a1}$	*6.35*
pK_{a2}	$-\log_{10}K_{a2}$	*10.33*
pK_w	$-\log_{10}K_w$	*14*

The pH can be determined from the values of W_a and W_b using an implicit equation:

$$W_a=\left[H^+\right]-\frac{K_w}{\left[H^+\right]}-W_b\frac{\frac{K_{a1}}{\left[H^+\right]}+\frac{2K_{a1}K_{a2}}{\left[H^+\right]^2}}{1+\frac{K_{a1}}{\left[H^+\right]}+\frac{K_{a1}K_{a2}}{\left[H^+\right]^2}} \tag{5}$$

Solving the equation for $\left[H^+\right]$, the pH can be computed from

$$pH=\log_{10}\left[H^+\right] \tag{6}$$

The implicit expression of the pH value in the Equation (5) disallows the possibility of application of the first principle model in the nonlinear predictive control scheme.

IV. Modeling the process using steady-state analysis

The steady state analysis (Fig. 4) shows the nonlinearity of the process. Five operating areas with almost linear behavior can be clearly identified on the steady-state plot. To obtain models relating to the pH, the base flow rate was perturbed about their nominal values.

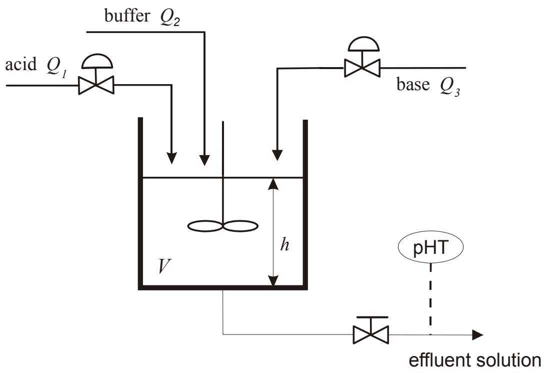


Fig 3 pH neutralization plant scheme

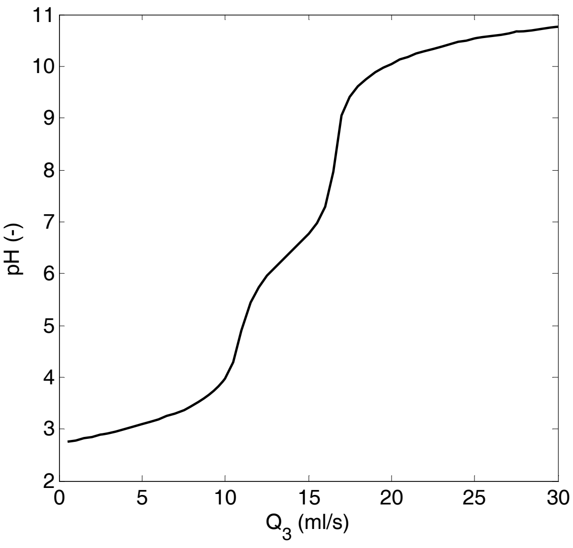


Fig. 4 Steady-state plot of the process

Due to the relationship of the pH value to the base flow-rate only the pH value at the time instant k-1 is used for scheduling the local models. A local model network that describes the nonlinear plant was constructed using local auto-regressive with exogenous input (ARX) models of the first order. The local models had the form of a first-order ARX model:

$$y(k+1) = ay(k) + bu(k) \tag{7}$$

This open loop data was used to construct 5 local models at the operating point at the centers of the linear parts. The local least squares cost function for the *i*-th model can be written as follows:

$$J(\theta_i) = \sum_{k=1}^{N} (y_i(k) - \hat{y}_i(k,\theta_i))^2 \tag{8}$$

The local cost function can be rewritten into a matrix form:

$$J(\theta_i) = \left(\boldsymbol{Y} - \boldsymbol{\Psi\Theta}_i\right)^T \left(\boldsymbol{Y} - \boldsymbol{\Psi\Theta}_i\right) \tag{9}$$

where $\boldsymbol{Y}$ is the output vector and regression matrix $\boldsymbol{\Psi}$ is given as:

$$\boldsymbol{\Psi} = \left[\boldsymbol{\Phi}^T(1)\boldsymbol{\Phi}^T(2)\ldots\boldsymbol{\Phi}^T(N)\right]^T \tag{10}$$

Knowledge about the process gain, stability and settling time could be translated into the form of inequality constraints. Thus optimization in the form of quadratic programming (QP) can be used to obtain model parameters, instead of conventional least-squared method. If the system to be identified is assumed stable there exist several limits on the parameters of local models. For example, for the system of the second order given by

$$G(z^{-1}) = \frac{b_1 z^{-1} + b_2 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}} \tag{11}$$

the following stability margins for the parameters can be introduced if the system is stable

$$\begin{aligned} a_2 &\le 1 \\ -a_2 + a_1 &\le 1 \\ -a_1 - a_2 &\le 1 \end{aligned} \tag{12}$$

which can be translated as the inequality conditions for QP as:

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ b_1 \\ b_2 \end{bmatrix} \le \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \tag{13}$$

The constrained optimization problem can then be formulated as a QP:

$$\min_{\theta} \left\{ \frac{1}{2} \boldsymbol{\Theta}^T \boldsymbol{H\Theta} + \boldsymbol{c}^T \boldsymbol{\Theta} \right\} \tag{14}$$

with matrix $\boldsymbol{H}$ and vector $\boldsymbol{c}$ given by

$$\boldsymbol{H} = 2\boldsymbol{\Psi}^T \boldsymbol{Q\Psi}, \quad \boldsymbol{c} = -2\boldsymbol{\Psi}^T \boldsymbol{Y} \tag{15}$$

and the constraints defined as:

$$\boldsymbol{A}_{inq}\boldsymbol{\Theta} \le \boldsymbol{b} \tag{16}$$

By using the constraints during the training process, more accurate model with improved interpretability can be identified using the input-output data. The parameters of the local models are shown in Table 2. The prediction of the local model network is given by:

$$\hat{y}(k+1) = \sum_{i=1}^{M} \mu_i(c_i, \sigma_i, \psi(t)) f_i(A, B, u(k), y(k)) \tag{17}$$

The centers of the validity functions were obtained from steady-state characteristic and parameters of local models by solving the quadratic problem with the data in the vicinity of the centre of the operating region. The remaining unknown parameter σ_i^2 from (17) is obtained by minimization the following criterion using the validation data:

$$MSE = \frac{1}{N-1} \sum_{k=1}^{N-1} \left(\hat{y}(k+1) - y(k+1)\right)^2 \tag{18}$$

The resulting distribution of the local models in the operating space of the system is shown in Fig. 5.

Table 2 Local Model parameters

Model	a_1	b_1	Gain	c_1	σ
1	0.75	0.020	0.08	3	0.13
2	0.81	0.246	1.29	4.9	0.16
3	0.89	0.035	0.32	6.4	0.31
4	0.85	0.319	2.08	8	0.20
5	0.99	0.003	0.42	10.5	0.22

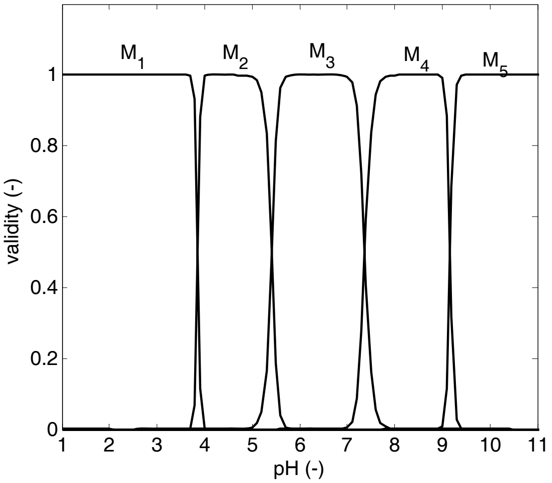


Fig. 5 Distribution of the local models

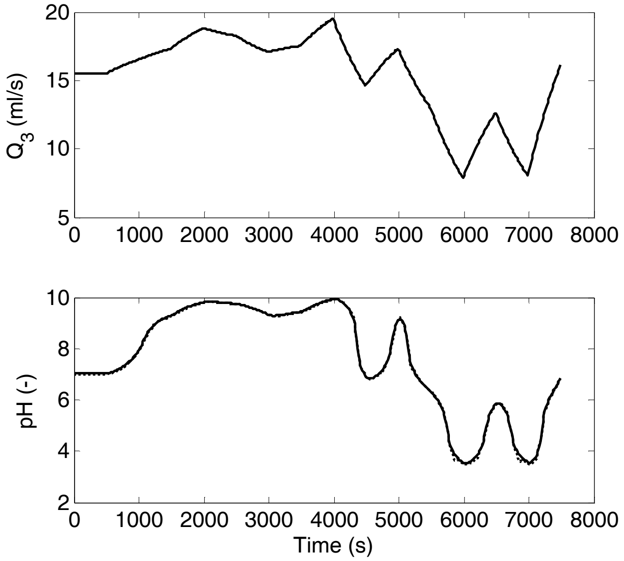


Fig. 6 Modeling performance of LMN with 5 models (LMN – solid, process output - dotted)

The performance of the model is shown in Fig. 6. The model is able to accurately imitate the behavior of the process for slow changes of the input variable i.e. the obtained model is accurate and valid only at regions closed to steady-state points. The off-equilibrium behavior of dynamic system is not included in the model. The problem of identifying of off-equilibrium linear models is caused by the fact that that is difficult to gather data of sufficient quality in these regions. To overcome the problem the authors in [17] suggested a heterogeneous solution with multiple-model around equilibrium points and Gaussian process submodels in transient areas. Here the global process model is obtained from input-output data that sufficiently cover the operating space of the process.

V. Gap Metric for Similarity Measure

To quantify the similarity between two systems a gap metric [18] is used. The gap metric is much more suitable to measure the distance between two linear systems than a metric based on norms. The gap metric between the local models associated with the clusters are computed. The gap metric for two SISO dynamic models M_1, M_2 is defined as:

$$\delta(M_1,M_2)=\sup_{\omega}\frac{\left|M_1(j\omega)-M_2(j\omega)\right|}{\sqrt{1+\left|M_1(j\omega)\right|^2}\sqrt{1+\left|M_2(j\omega)\right|^2}} \qquad (19)$$

where 0≤ δ ≤ 1, $M_1(j\omega)$ and $M_2(j\omega)$ represent the frequency responses of the systems M_1 and M_2, respectively. Two models have similar behavior in close-loop if the value of δ is close to 0 and behave differently for value of δ close to 1.

Table 3 Distances between the linear models measured using the gap metric

M	1	2	3	4	5
1	0	0.36	0.08	0.43	0.14
2	0.36	0	0.25	0.06	0.29
3	0.08	0.25	0	0.32	0.06
4	0.43	0.06	0.32	0	0.35
5	0.14	0.29	0.06	0.35	0

Table 3 shows the gap metric between the pairs of five linear models representing the whole operating range of the nonlinear process. The obtained values show the similarity between the models 1, 3 and 5 and models 2 and 4. The similarity between models 1, 3, and 5 can be explained physically by the fact that these models represent low-sensitivity regions.

VI. Modeling the Process from Experimental Data

Optimization of local model networks structure from the input-output data structure presents a challenging problem. The objective of the modeling process is to find the parameters of the validity function, parameters of the local models and also the number of local models at the same time. The modeling performance can be obtained by computation the following criterion:

$$J(M,\Theta_i,\mu_i)=\frac{1}{N}\sum_{k=1}^{N}\left(\sum_{i=1}^{M}\mu_i\Theta_i\Phi(k)-y(k+1)\right)^2 \qquad (20)$$

where N is the number of samples, M is the number of local models, μ_i are the validity functions, Θ are the local model parameters and regression vector $\Phi(k)$ contains past data.

Several approaches related to optimization of the local model structure can be found in literature. Methods developed by Johansen and Foss in [19] and Nelles [20] start with a single model and hierarchically partition operating space and iteratively increase the number of models and thus preventing from over-fitting. Division of the data to the local models is made via clustering which enables division of the complex nonlinear regions into simpler subspaces. Here the Gustafson and Kessel [21] algorithm is implemented which extends the standard fuzzy c-means algorithm by employing an adaptive distance norm, in order to detect clusters of different geometrical shapes in one data set. This algorithm has the advantage of looking for ellipsoids of variable size and orientation. Gustafson-Kessel algorithm finds clusters by minimizing the following function:

$$J_{X,m}\left(\boldsymbol{U},\boldsymbol{V}\right)=\sum_{j=1}^{N}\sum_{i=1}^{M}\mu_{i,j}^{m}D_{A_i}^{2}\left(x_j-c_i\right) \qquad (21)$$

where $\boldsymbol{U}$ is a set of membership degrees , $\boldsymbol{V}$ is a set of cluster centers c, m is fuzziness factor (usually a value close to 2), X is a set of N samples x, and is a norm induced by matrix $\boldsymbol{A}$. Fuzzy covariance matrix $\boldsymbol{F}$ is defined as:

$$\boldsymbol{F}_i = \frac{\sum_{j=1}^{N} \mu_{i,j}^m \left(x_j - c_i\right)\left(x_j - c_i\right)^T}{\sum_{j=1}^{M} \mu_{i,j}^m} \qquad (22)$$

The distance $D_{A_i}^2$ of a data point to the cluster centre is induced by matrix $\boldsymbol{A}_i$ as:

$$D_{A_i}^2(x) = \left(c_i - x\right)^T A_i \left(c_i - x\right) \qquad (23)$$

The centers of the clusters are calculated as the weighted mean value of all membership degrees:

$$c_i = \frac{\sum_{j=1}^{N} \mu_{i,j}^m x_j}{\sum_{j=1}^{N} \mu_{i,j}^m} \qquad (24)$$

Equation (21) represents a non-linear optimization problem, which is solved in an iterative manner. The cluster algorithm stops when a predetermined stopping criterion is fulfilled. The set of membership degrees is initialized randomly and then the following steps are repeated:

Step 1. Compute cluster prototypes

Step 2. Compute distance to cluster prototype

Step 3. Update the partition matrix *U*

The Steps 1- 3 are repeated until the following condition holds:

$$\left\| U(k) - U(k-1) \right\| < \delta \qquad (25)$$

If we fix the parameters of the validity function the only free parameters are those in the linear regression equations. The advantage of such a treatment is that the parameter identification problem is reduced to a simple linear optimization problem and, thus, can be solved using efficient linear learning algorithms. Simultaneous optimization of parameters of all local models yields the best results in the sense of the prediction error. Alternatively, local estimation approach, which is more computationally efficient, can be used. The parameters of each of local model are identified independently to guarantee individual local models to be local approximations of the underlying system. The local least squares cost function for the *i*-th model can be written as follows:

$$J(\theta_i) = \sum_{k=1}^{N} \mu_{i,k}(k)(y_i(k) - \hat{y}_i(k,\theta_i))^2 \qquad (26)$$

The local cost function can be rewritten into a matrix form:

$$J(\theta_i) = \left(\boldsymbol{Y} - \boldsymbol{\Psi}\theta\right)^T \boldsymbol{Q} \left(\boldsymbol{Y} - \boldsymbol{\Psi}\theta\right) \qquad (27)$$

where ***Y*** is the output vector and regression matrix is given as:

$$\boldsymbol{\Psi} = \left[\boldsymbol{\Phi}^T(1)\boldsymbol{\Phi}^T(2)\ldots\boldsymbol{\Phi}^T(N)\right]^T \qquad (28)$$

and diagonal matrix $\boldsymbol{Q}$ is given

$$\boldsymbol{Q} = \begin{bmatrix} \mu_{i,1} & 0 & \cdots & 0 \\ 0 & \mu_{i,2} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \mu_{i,N} \end{bmatrix} \qquad (29)$$

To reduce the amount of data in regression vector only the in the vicinity of the center of the local models are used i.e. with the value of validity function $\mu_{i,k} > 0.1$.

The clustering algorithm is started with a large number of clusters which results in a local network with imitates the behavior of the nonlinear system accurately. If the global model is build using larger number of local models, the quality of fitness of the model to input-output data increases. However, the idea of ‘parsimony principle’ should be introduced that says that among the models which explain the data well, the model with the smallest number of independent parameters should be chosen. Since the clustering of the dynamic data can result in clusters with similar or same local model parameters, the structure can be reduced in order to obtain simpler model. Therefore strategy for model reduction using prediction error and gap metric developed in Gugaliya and Gudi [22] is adopted. The distance between each pair of cluster centers is computed

$$D_{ij} = \left| M_i - M_j \right| \qquad (30)$$

and the pair *(i,j)* with minimal distance D_{ij} is found. The clusters M_i, M_j are merged together and new covariance matrix F_{new} and the centre of the new cluster c_{new} are defined by the following equations:

$$\begin{aligned} W_{new} &= W_i + W_j \\ c_{new} &= \frac{W_i}{W_{new}} c_i + \frac{W_j}{W_{new}} c_j \\ F_{new} &= \frac{W_i}{W_{new}} F_i + \frac{W_j}{W_{new}} F_j + \\ &+ \frac{W_i W_j}{W_{new}^2} \left[\left(c_i - c_j\right)\left(c_i - c_j\right)^T \right] \end{aligned} \qquad (31)$$

where W_i, W_j are the numbers of data points inside the cluster. Merging of two clusters is plotted in Fig. 7.

The modeling performance of the original set of models without merging MSE_{orig} and the performance of the reduced set $MSE_{reduced}$ is computed using the following prediction performance criterion:

$$MSE_{orig}, MSE_{reduced} = \frac{1}{N-1} \sum_{k=1}^{N-1} \left(\hat{y}(k+1) - y(k+1)\right)^2 \qquad (32)$$

If the following condition holds:

$$MSE_{red} - MSE_{orig} < \delta_{pred} AND \delta(M_1, M_2) < \delta_{gap} \qquad (33)$$

the model merging is accepted and distances between the centers in the reduced set are recomputed. If the condition (33) is false the original structure is used and new pair with

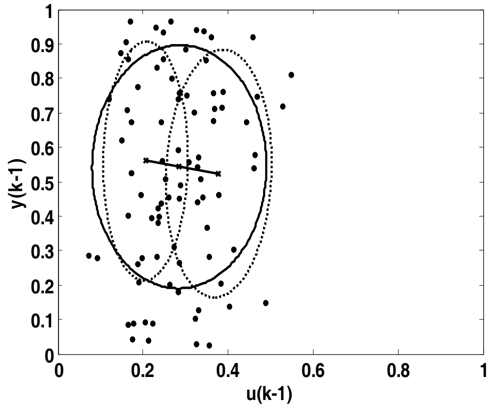


Fig. 7 Merging two clusters (dotted line – original clusters)

minimal distance is found and algorithm repeated. The algorithm ends when no clusters can be merged without violating the condition (33).

VII. Modeling pH neutralization from input-output data

The nonlinear analytical model of the process is used to generate input–output data for the identification of the position of clusters. Generalized multi level noise [23] is used as a test signal for the identification. The output of the system was corrupted with an additive Gaussian white noise with zero mean and standard deviation = 0.001 to simulate the measurement noise. The input–output data has 2000 samples with a sampling period of 15 s. The first 1000 samples are used for identification of the model and the rest of 1000 samples for validation purpose. The input–output data are plotted in Fig. 8. The initial number of local models was set to 30. The Gustafson-Kessel algorithm as described in Section VI was used for initial position and orientation of these clusters. All the local models were assumed to be of the 1-order and have the following structure:

$$y(k+1) = ay(k) + bu(k) \tag{34}$$

The modeling performance of the LMN with 30 local models is $MSE = 0.1$ and the response of the model is compared with the validation data in Fig. 9. The structure of the local model is further optimized using the strategy described in Section VI. The prediction error threshold and the gap metric threshold were set to 0.001 and 0.2, respectively. With the model reduction algorithm the number of models is reduced from 30 to 15 while the value of performance given by the MSE increased only from 0.10 to 0.11. The position and orientation of the clusters was extracted from the covariance matrix $\boldsymbol{F}$ and shown in Fig. 10.

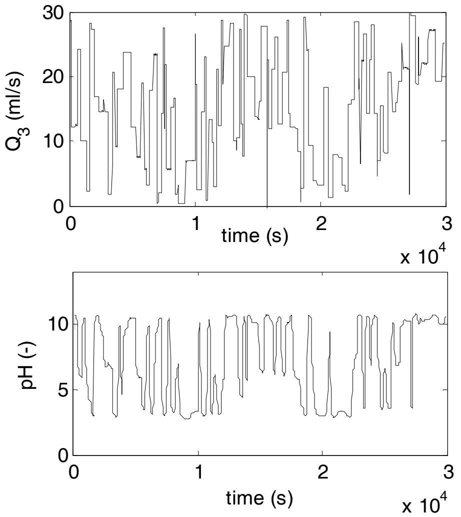


Fig. 8 Input-output data for identification and validation

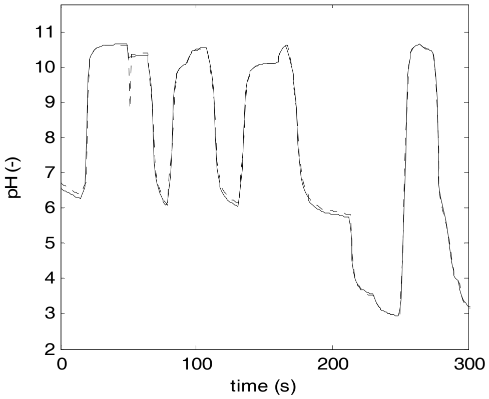


Fig. 9 Modeling performance of LMN with 30 local models (dotted – plant, solid - model)

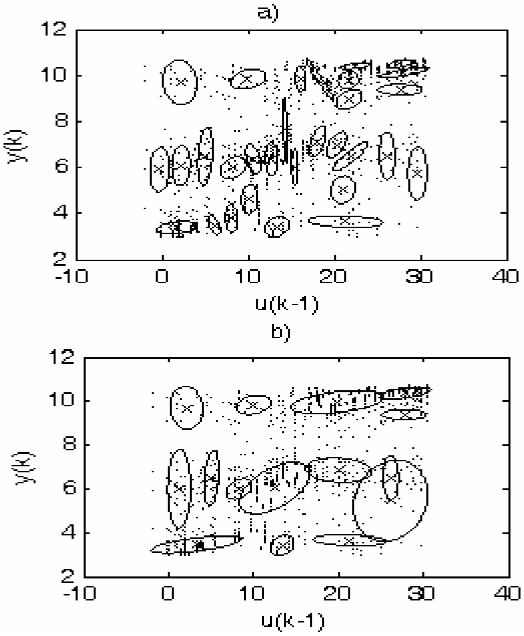


Fig. 10 Position and orientation of clusters (a –LMN with 30 models b – reduced LMN with 15 models)

VIII. GPC Control with local model network

A multivariable controlled auto-regressive integrated moving average (CARIMA) model for a SISO system can be described by

$$\boldsymbol{A}(z^{-1})\boldsymbol{y}(k) = \boldsymbol{B}(z^{-1})\boldsymbol{u}(k-1) + \frac{\xi(k)}{\Delta} \tag{35}$$

where $\Delta = 1 - z^{-1}, z^{-1}$ is the difference operator, whose function is to guarantee integral action in the controller to eliminate any offset. The terms $\boldsymbol{y}(k), \boldsymbol{u}(k), \xi(k)$ are the output, input, and noise vectors, respectively. The terms $\boldsymbol{A}(z^{-1}), \boldsymbol{B}(z^{-1})$ are the matrix polynomials of z^{-1} ,

$$\begin{aligned} \boldsymbol{A} &= \boldsymbol{I} + \boldsymbol{A}_1 z^{-1} + \ldots + \boldsymbol{A}_{na} z^{-na} \\ \boldsymbol{B} &= \boldsymbol{B}_0 + \boldsymbol{B}_1 z^{-1} + \ldots + \boldsymbol{B}_{nb} z^{-nb} \end{aligned} \tag{36}$$

where *na* and *nb* are the orders of the model output and input, respectively. To design a GPC controller it is necessary to derive predictions k-step ahead:

$$\begin{aligned} \boldsymbol{Y}(t+k) &= \boldsymbol{G}\Delta\boldsymbol{U}(t) + \boldsymbol{S}(t+k) \\ \boldsymbol{S}(t+k) &= \boldsymbol{Y}_{k-1}\boldsymbol{Y}(t) + \boldsymbol{U}_{k-1}\Delta\boldsymbol{u}(t-1) \end{aligned} \tag{37}$$

where $\boldsymbol{S}$ represents the free response of the system and the other terms are given:

$$\begin{aligned} &\boldsymbol{Y}(t+k) = \left[\boldsymbol{y}(t+1), \boldsymbol{y}(t+2), \ldots, \boldsymbol{y}(t+k)\right]^T \\ &\boldsymbol{y}(t+1) = [y_1(t+1), \ldots, y_R(t+1)]^T \\ &\Delta\boldsymbol{U}(t) = [\Delta\boldsymbol{u}(t), \ldots, \Delta\boldsymbol{u}(t+k-1)]^T \\ &\Delta\boldsymbol{u}(t) = \left[\Delta u_1(t), \ldots, \Delta u_S(t)\right] \\ &\boldsymbol{Y}(t) = [\boldsymbol{y}(t), \ldots, \boldsymbol{y}(t-na)]^T \\ &\boldsymbol{y}(t) = [y_1(t), \ldots y_R(t)]^T \end{aligned} \tag{38}$$

The cost function used in the GPC algorithms is defined as:

$$\begin{aligned} J = &\sum_{k=1}^{N_p} \left\| \boldsymbol{Y}(t+k|t) - \boldsymbol{W}(t+k) \right\|_{\boldsymbol{Q}}^2 + \\ &\sum_{k=1}^{N_u} \left\| \Delta\boldsymbol{U}(t+k-1) \right\|_{\boldsymbol{R}}^2 \end{aligned} \tag{39}$$

where $\boldsymbol{W}(t+k)$ is the reference trajectory at a future time point *k*, N_p is the output prediction horizon, and N_u is the control increment horizon.

This criterion can be rewritten in a matrix form

$$J = \frac{1}{2}\Delta\boldsymbol{u}^T \boldsymbol{H}\Delta\boldsymbol{u} + \boldsymbol{b}^T\Delta\boldsymbol{u} + \boldsymbol{f}_0 \tag{40}$$

where $\boldsymbol{H}, \boldsymbol{b}, \boldsymbol{f}_0$ are defined as

$$\begin{aligned} \boldsymbol{H} &= 2(\boldsymbol{G}^T\boldsymbol{Q}\boldsymbol{G} + R) \\ \boldsymbol{b} &= 2(\boldsymbol{S} - \boldsymbol{W})^T \boldsymbol{Q}\boldsymbol{G} \\ f_0 &= (\boldsymbol{S} - \boldsymbol{W})^T(\boldsymbol{S} - \boldsymbol{W}) \end{aligned} \tag{41}$$

Since the vector is a constant vector and does not have an effect on the quadratic programming result, the constrained optimization problem can be defined as:

$$\begin{aligned} &J = \frac{1}{2}\Delta\boldsymbol{u}^T \boldsymbol{H}\Delta\boldsymbol{u} + \boldsymbol{b}^T\Delta\boldsymbol{u} \\ &\boldsymbol{A}\Delta\boldsymbol{u} \le \omega \end{aligned} \tag{42}$$

where the constraints for a control action can be rewritten as

$$\boldsymbol{A} = \begin{bmatrix} I_u \\ -I_u \\ I_{\Delta u} \\ -I_{\Delta u} \end{bmatrix} \quad \omega = \begin{bmatrix} u^{\max} - Iu(k-1) \\ -u^{\min} + Iu(k-1) \\ \Delta u^{\max} \\ -\Delta u^{\min} \end{bmatrix} \tag{43}$$

Parameters $u^{\max}$ and $u^{\min}$ are constraints for the control signals $\Delta u^{\max}$ and $\Delta u^{\min}$ and are constraints for the control signal increments. Matrices $I_{\Delta u}$ and I_u are defined as:

$$\begin{aligned} I_{\Delta u} &= \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix}, \\ I_u &= \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 1 & 0 & \vdots \\ \vdots & 1 & \ddots & 0 \\ 1 & \cdots & 1 & 1 \end{bmatrix} \end{aligned} \tag{44}$$

In the single-step linearization, a single linear model *M(k)* is used over the entire prediction horizon at the time k. The optimal control signal can be found by means of quadratic programming. For multiple-step-ahead control, however, the linear model may significantly deteriorate from the nonlinear process and therefore negatively influence the controller performance. To reduce the computational complexity of the nonlinear optimization linearization along the trajectory [24] or MPC algorithm with Nonlinear Prediction and Linearization (MPC-NPL) described in [25] can be used. Both algorithms use a numerically reliable quadratic programming procedure thus the necessity of repeating full nonlinear optimization at each sampling instant is avoided.

The procedure of multi-step linearization along nominal trajectory is executed as follows:

Step 1. Using the current scheduling vector $\psi(k)$ validity functions for each of the local models are computed:

$$\rho_i = f\left(c_i, \sigma_i, \psi(k)\right) \tag{45}$$

Step 2. The local model network is linearized around the current operating point to give linear model $M(k)$ of the plant

$$A = \sum_{i=1}^{M} \rho_i A_i, B = \sum_{i=1}^{M} \rho_i B_i \tag{46}$$

Step 3. The obtained linear model $M(k)$ is used to compute the control signal increments Δu over the entire prediction horizon H_p .

Step 4. Using $\Delta u(k)$ compute $M(k+1)$, compute the response and linearize the LMN at the new point.

Step 5. Using $M(k)$ and $M(k+1)$, compute new control sequence $\boldsymbol{u}$ over the entire prediction horizon.

Steps 4 and 5 are repeated for $1..H_p$. The set of linear models $M(k).....M(k+H_p-1)$ is then used in the quadratic programming problem and the first element of control increment $\Delta u(k)$ is applied to the system.

IX. Control of the pH Neutralization Process

Saturation constraints in the manipulated variables are imposed to take into account the minimum/maximum aperture of the valve regulating the base flow rate. A lower limit of 0ml/s and an upper limit of 30 ml/s are chosen for this variable. The prediction horizon was set to 8 samples as a result of using different values and comparing control performances. A control horizon of 4 samples was selected since further increase did not add significant improvement in terms of performance. The weighting matrix $\boldsymbol{Q}$ associated with the error from set point was set two times greater than matrix $\boldsymbol{R}$ associated control signal changes.

$$\boldsymbol{Q} = 2I, \boldsymbol{R} = I \tag{47}$$

The model predictive control algorithm described in Section VII was implemented using the "quadprog" function in MATLAB's Optimization Toolbox to minimize the cost function. To reduce the on-line computational load, the control sequence computed at the step $k - 1$ was shifted backwards and used as an initial guess for the computation of the future controller output at time k. The resulting control courses for stepwise set-point changes are shown in Fig. 11.

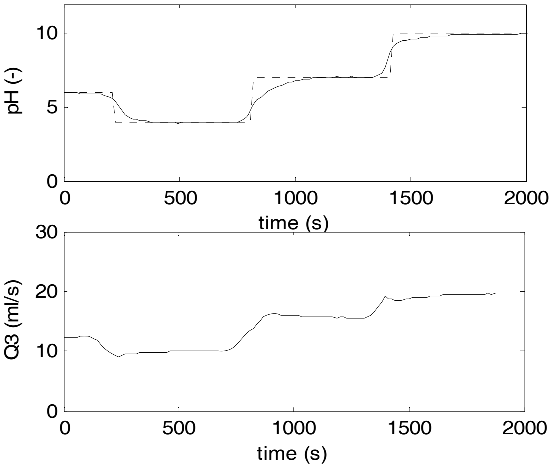


Fig. 11 Controller performance

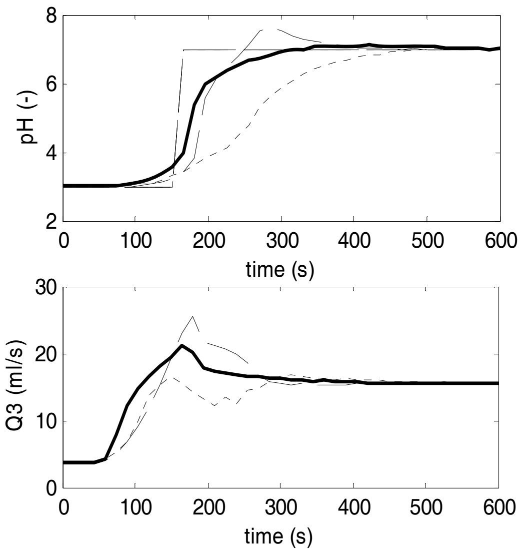


Fig. 12. Comparison of the LMN MPC controller with linearization along trajectory with other MPC strategies (solid – LMN MPC with linearization, dotted – LMN without linearization, dashed - linear MPC with a single model)

The results shown in Fig. 12 show that, for the considered application, the LMN MPC with linearization along the future trajectory performs slightly better than the MPC based on linear model or MPC where linearized model is used for the whole prediction horizon.

X. Conclusion

The multiple model modeling and control strategies represent an efficient tool for nonlinear processes. During the development of nonlinear model the apriori information about the process can be used. If accurate model of the process in off-equilibrium regions of the steady-state characteristic is needed the model can be identified from the input-output data. The structure of the model can be optimized using gap metric without loosing the accuracy and stability of the control loop. The control problem is solved using a generalized predictive controller that used parameters obtained from linearization of the local model network at each sampling interval. The control tests executed on the simulation model gave satisfactory results. It was proved that the examined method can be implemented and used successfully to control such nonlinear processes.

References

[1] E.F. Camacho, C. Bordons, *Model Predictive Control*. London, UK: Apringer, 1999.

[2] J.M. Maciejowski, *Predictive Control with Constraints*. Harlow, UK: Prentice Hall, 2002.

[3] S.J. Qin, T.A. Badgwell, An overview of industrial model predictive control technology, in *Chemical Process Control*, Tahoe City, 1996, pp. 232-256.

[4] P. Saha, S.H. Krishnan, V.S.R. Rao, S.C. Patwardhan, Modeling and predictive control of MIMO nonlinear systems using Wiener-Laguerre models, Chemical Engineering Communications, vol. 191, no. 8, 2004, pp. 1083-1119.

[5] R. Murray-Smith, T.A. Johansen, *Multiple Model Approaches to Modelling and Control*, London, UK: Taylor and Francis, 1997.

[6] I. Skrnjanc, M. Lepetic, J.L.Figueroa, S. Blazic, Fuzzy model-based predictive control for a CSTR with multiple steady state: A simulation study and a comparison with other nonlinear MBPC control algorithms, in *5th WSEAS Int. Conf. on Fuzzy Sets and Fuzzy Systems*, Italy, 2004.

[7] O. Galan, J.A. Romagnoli, A. Palazoglu, Real-time implementation of multi-linear model-based control strategies—an application to a bench-scale pH neutralization reactor, *Journal of Process Control*, vol. 15, no. 5, 2004, pp. 571–579.

[8] A.H. Mazinan, N. Sadati, Fuzzy Multiple Modeling and Fuzzy Predictive control of A Tubular Heat Exchanger System, in *7th WSEAS Interantional Conference on Application of Electrical Engineering*, Trondheim, Norway, 2008.

[9] A. Ichtev, T. Puleva, Multiple-Model Adaptive Control of Hydro Turbine Generator with Fuzzy TS Models, in *9th WSEAS International Conference on Fuzzy Systems*, Bulgaria, 2008.

[10] L. Bláhová, J. Dvoran, Neuro-Fuzzy Control of Chemical Reactor with Disturbances, in *14th WSEAS International Conference on Systems*, Greece, 2010.

[11] B. Aufderheide, V. Prasad, B.W. Bequette, A comparison of fundamental model-based and multiple model predictive control., in *Proc. the 40th Conference on Decision and Control*, vol. 5, 2001, pp. 4863 – 4868.

[12] M. Athans, S. Fekri, A. Pascoal. Issues on robust adaptive feedback control. In *Proc.16th IFAC World Congress*. Invited plenary paper, 2005, pp. 9-39.

[13] T.A. Johansen, B.A. Foss, Identification of nonlinear system structure and parameters using regime decomposition, *Automatica*, vol. 31, no. 2, pp. 321-326.

[14] Hall, R. C. and D. E. Seborg (1989). Modelling and self-tuning control of a multivariable pH neutralization process, part I: Modelling and multiloop control, in *Proc. of the American Control Conference*, Pittsburgh, PA, 1989, pp. 1822-1828.

[15] J.M. Boling, D.E. Seborg, J.P. Hespanha, Multi-model adaptive control of a simulated pH neutralization process, *Control Engineering practice*, vol. 15, no. 6, 2007, pp. 663-672.

[16] M. Pottmann, D.E. Seborg, A nonlinear predictive control strategy based on radial basis function models, *Computers and Chemical Engineering*, vol. 21, no. 9, 1997, pp. 965-980.

[17] R. Murray-Smith, T.A. Johansen, R. Shorten, On transient dynamics, off-equilibrium behaviour and identification in blended multiple model structures, in *Proc. 5th European Control Conference*, Karlsruhe, Germany, 1999.

[18] T.T Georgiou, M.C. Smith, Optimal robustness in the gap metric. *IEEE Transactions on Automatic Control*, vol. 35, 1990, pp. 673-686.

[19] T.A. Johansen, and B.A. Foss, Identification of non-linear system structure and parameters using regime decomposition, *Automatica*, vol. 31, no. 2, 1995, pp.321-326.

[20] O. Nelles, Orthonormal basis functions for nonlinear system identification with local linear model trees (LOLIMOT), in *Proc. of 11th IFAC Symposium on System Identification*, Kitakyushu, Japan, 1997.

[21] D.E. Gustafson, W.C. Kessel, Fuzzy clustering with a fuzzy covariance matrix, in *Proc. of IEEE Conference on Decision and Control*, 1978, pp.761–766.

[22] J.K. Gugaliya, R.D. Gudi, Multimodel decomposition of Nonlinear Dynamics Using Fuzzy Classification and Gap Metric Analysis, in *Proc. of the 9th International Symposium on Dynamics and Control of Process Systems*, Leuven, Belgium, 2010.

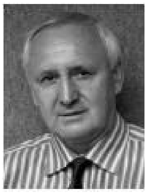
[23] Y. Zhu, *Multivariable System Identification for Process Control*,.New York, NY: Elsevier Science Ltd, 2001.

[24] S. Molov, R. Babuska, J.A. Roubos & H.B. Verbruggen, MIMO predictive control by multiple-step linearization of Takagi–Sugeno fuzzy models, in *Proc. the IFAC/IMACS International Conference on Artificial Intelligence in Real-Time Control*, Grand Canyon, AZ, 1998.

[25] P. Tatjewski, *Advanced control of industrial processes, Structures and algorithms*, London, UK: Springer, 2007.

Jakub Novák received the Ph.D. degree in chemical and process engineering from Tomas Bata University in Zlin in 2007.
He is a researcher at the Center for Applied Cybernetics at Tomas Bata University in Zlin. His research interests are modeling and predictive control of the nonlinear systems.

Petr Chalupa graduated in 1999 from Brno University of Technology and received the Ph.D. degree in Technical cybernetics from Tomas Bata University in Zlin in 2003.
He is a researcher at the Center for Applied Cybernetics at Tomas Bata University in Zlin. His research interests are adaptive and predictive control of real-time systems.



Vladimír Bobál graduated in 1966 from the Brno University of Technology. He received his Ph.D. degree in Technical Cybernetics at Institute of Technical Cybernetics, Slovak Academy of Sciences, Bratislava, Slovak Republic.
He is now Professor in the Department of Process Control, Faculty of Applied Informatics of the Tomas Bata University in Zlín. His research interests are adaptive control systems, system identification and CAD for self-tuning controllers.