CSTR CONTROL USING MULTIPLE MODELS

J. Novák, V. Bobál

Univerzita Tomáše Bati, Fakulta aplikované informatiky

Mostní 5139, Zlín

INTRODUCTION

Almost every real process exhibits nonlinear behavior in a full operating range. Local Model Networks are networks which are composed of locally accurate models, where output is interpolated by smooth locally active validity functions. This divide-and-conquer strategy is a general way of coping with complex systems. The architecture of LMN benefits from being able to incorporate a priori knowledge and conventional system identification methodology. The LMN structure also gives transparent and simple representation of the nonlinear system. Contrary to the black box representation of the nonlinear process by the neural networks, the conventional design methods can be utilized for nonlinear controller design. The idea of the LMN approach is to split the whole operating region into several sub-regions where in each region sub-region the process has close to linear behavior. For each region a local linear model is developed to approximate the non-linear dynamics. The global model of the process is a linear combination of the local models. In an initial off-line identification phase the local models and the validity function parameters have to be identified. Several methods can be used to obtain LMN parameters. The Expectation Maximization (McLachlan and Krishnan, 1997) algorithm is usually used for the Gaussian process models although it requires a priori knowledge of complexity of the system or more precisely the number of local models. Another development is the local linear model tree LOLIMOT (Nelles, 1997). It is based on the idea to approximate a nonlinear map with piece-wise linear local models. The algorithm systematically bisects partitions of input space. Local models that do not fit sufficiently well are replaced by two or more smaller models in the expectation that they will fit the nonlinear target function better in their region of validity. Another training algorithm discussed in (Johansen and Foss, 1995) uses two loops for structure optimization and parameter estimation to iteratively increase the number of models and thus preventing from overparametrization. If linear local models are employed and the parameters of the validity function are fixed, the parameters of local models can be obtained using the standard least-squares method.

LOCAL MODEL NETWORKS

Local model network (LMN) is a generalization of the radial basis function network, in which individual neurons are replaced by local sub-models with basis functions defining the regions of validity of individual sub-models, according to the expected operating regions of the plant (Murray-Smith and Johansen, 1997).

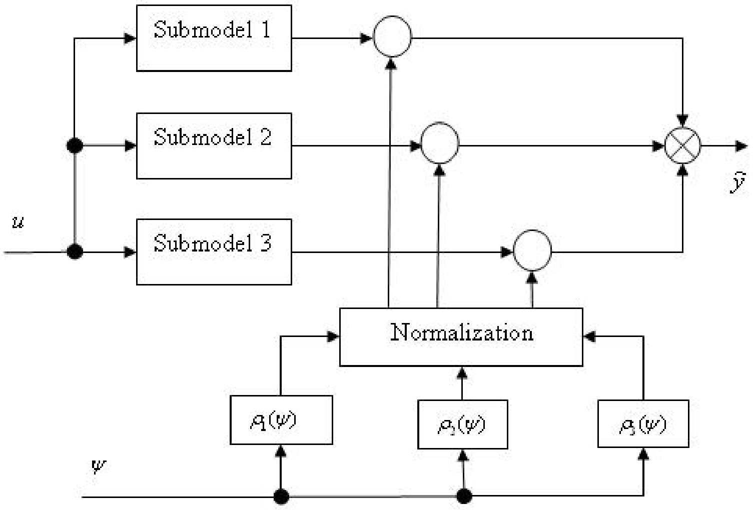

Figure 1 Local Model Network scheme

Controllers are designed for each of the local models and the basis functions of the local model networks are used to interpolate between them to produce a nonlinear controller. The output of the system is given by

$$\hat{y}(k) = \sum_{j=1}^{M} r_j \left[y\,(k)\right] \hat{y}_j(k)\,, \tag{1}$$

where $y\,(k)$ is a vector of scheduling variables, r_j is a basis function and $\hat{y}_j(k)$ is the output of the *j*-th model. The basis functions should form a partition of unity for the input space, i.e. at any point in the input space the sum of all basis function should equal 1. The network's basis function are normalized to achieve the partition of unity, i.e.

$$r_j(y\,) = \frac{\tilde{r}_j(y\,)}{\sum_{j=1}^{M} \tilde{r}_j(y\,)}\,, \tag{2}$$

where is the general unnormalized basis function and normalized basis functions   sum to unity. Though normalization is often desirable, it also results in several side-effects, for example change of shape or loss of local support (Shorten and Murray-Smith, 1998).

The blending of local models is calculated using the weighting functions. Gaussian basis functions are usually used for weighting the outputs of local models. The Gauss function for j-*th* model is given by

$$\tilde{r}_j(y\,) = \exp\left(-\frac{1}{2}\left(y\, - c_j\right)^T s_j^{-2}\left(y\, - c_j\right)\right), \tag{3}$$

where parameters c_j, s_j , define the Gaussian center and width, respectively and the scheduling variable y can be a system state or any system variable. For one scheduling variable the weighting function is a typical bell-shaped curve.

NONLINEAR MODELING USING SOMA ALGORITHM

In attempt to accurately model the nonlinear system, a wide variety of techniques have been developed such as nonlinear autoregressive moving average with exogenous inputs

(NARMAX) models (Chen and Billings, 1989), Hammerstein models (Billings and Fakhouri, 1982) or Multiple Layer Perceptron (MLP) neural network (Narendra and Parthasarathy, 1990). Even though, these methods offers improved accuracy over a single linear model, the black box representation of dynamics in these methods fails to exploit the theoretical results available in the conventional modeling and control domain. Besides MLP networks, Radial Basis Function (RBF) networks, which were initially introduced for multivariable interpolation, are other popular neural networks. The RBF network is a generalized version of LMN network where the output weights are substituted by the local models.

The Self-Organizing Migrating Algorithm - SOMA (Zelinka, 2002) is based on the competitive-cooperative behavior of intelligent creatures solving a common problem. Such behavior of intelligent creatures can be observed anywhere in the world. A group of wolves or other predators may be a good example. If they are looking for food, they usually cooperate and compete so that if one member of the group is more successful than the previous best one (e.g. has found more food) then all members change their trajectories towards the new most successful member. It is repeated until all members meet at one food source. In SOMA, wolves are replaced by individuals. They ‘live’ in the optimized model’s hyperspace, looking for the best solution. It can be said, that this kind of behavior of intelligent individuals allows SOMA to realize very successful searches. The identification of local operating regimes for unknown plant is difficult. The flexibility offered by SOMA provides opportunity to optimize both validity function and model parameters simultaneously.

INTERNAL MODEL CONTROL

When the model is available, the Internal Model Control (IMC) is one of the widely used approaches for control of the linear systems. The IMC scheme, first proposed in (Morari and Zafiriou, 1989), has found a number of successful applications. Figure 3 shows the standard IMC control structure where Gs represents the transfer function of the process, GM is the process model and GR is the asymptotically stable transfer functions. The feedback signal is the difference and the regulator contains a model of the process explicitly. In fact, the IMC is a generalization of the Smith predictor.

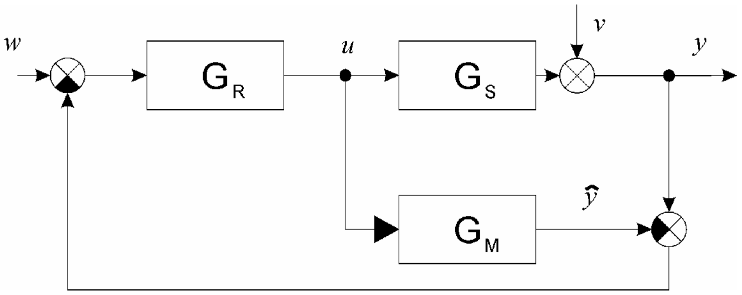

Figure 2 IMC block diagram

The IMC synthesis is a two step method. The Controller GR is divided into two parts:

$$G_R = G_Q.G_F$$

Firstly, the parameters of the controller G_R has to be determined. The best policy is to choose G_Q as an approximated inverse of the process model G_M, which will yield good tracking and disturbance rejection. In the second step the low-pass filter G_F is augmented to ensure robustness. The structure and parameters of G_F are chosen to achieve balance between robust stability and performance. The filter constant is the only parameter that has to be tuned, thus making the controller design simple.

The IMC approach can be extended to nonlinear models based on local model. In general, the inversion of nonlinear model is not simple and analytical solution may not exist. The LMN structure enables the following inverse control law for the 2nd order plant to be defined:

$$u(k)=\frac{1}{B_0}(v(k)-A_1y(k)-A_2y(k-1)-B_1u(k-1))$$

The v*(k)* replaces *y(k+1)* since it is not available at time *k*. The set-point *w(k)* is filtered using filter

$$G_F=\frac{1-c}{1-cz^{-1}}$$

so the *v(k)* becomes

$$y(k+1)\ \mathbf{;}\ v(k)=G_F(w(k)-d(k))$$

where *d(k)=y_p(k)-y_m(k)* is the difference between the outputs of the plant and model.

MODELING AND CONTROL OF CSTR

The studied system is a pH neutralization tank. A schematic diagram of the pH neutralization process is depicted in Figure 2. The neutralization process represents a highly nonlinear process. The dynamic model used in this work has been developed by Hall and Seborg (1989) and has been used to test single loop strategies in (Hu et al, 2000, Townsend et al, 1997).

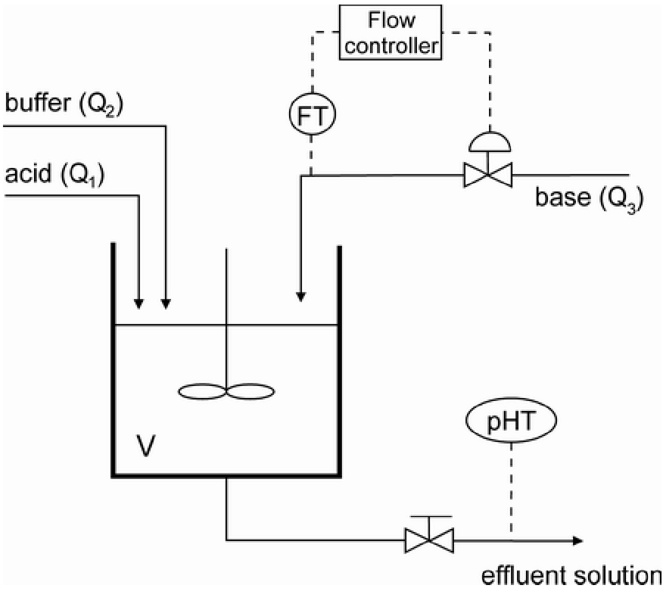

Figure 3 pH neutralization plant scheme

The process consists of an acid (HNO_3) stream, a buffer ($NaHCO_3$) stream and a base (NaOH) stream being continually mixed in a tank. The model is based on assumptions that the streams are perfectly mixed, the density is constant in the whole tank. The process aims at controlling the pH value (controlled variable) of the outlet stream by varying the inlet base stream Q_3 (control variable). The outlet flow-rate is dependent on the fluid height in the tank as well as the position of the valve. Titration curve (Figure 4) shows the nonlinearity of the neutralization process.

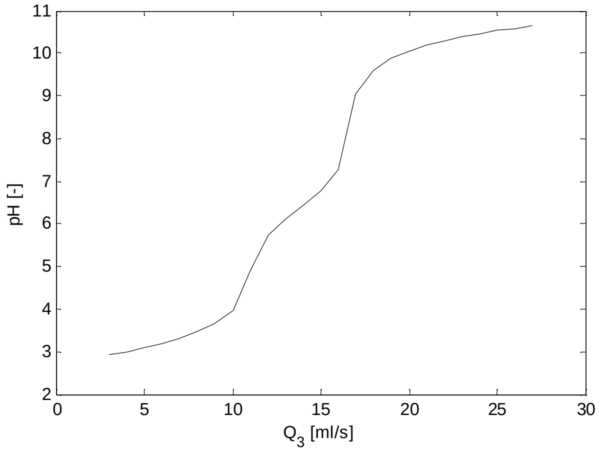

Figure 4 Titration curve

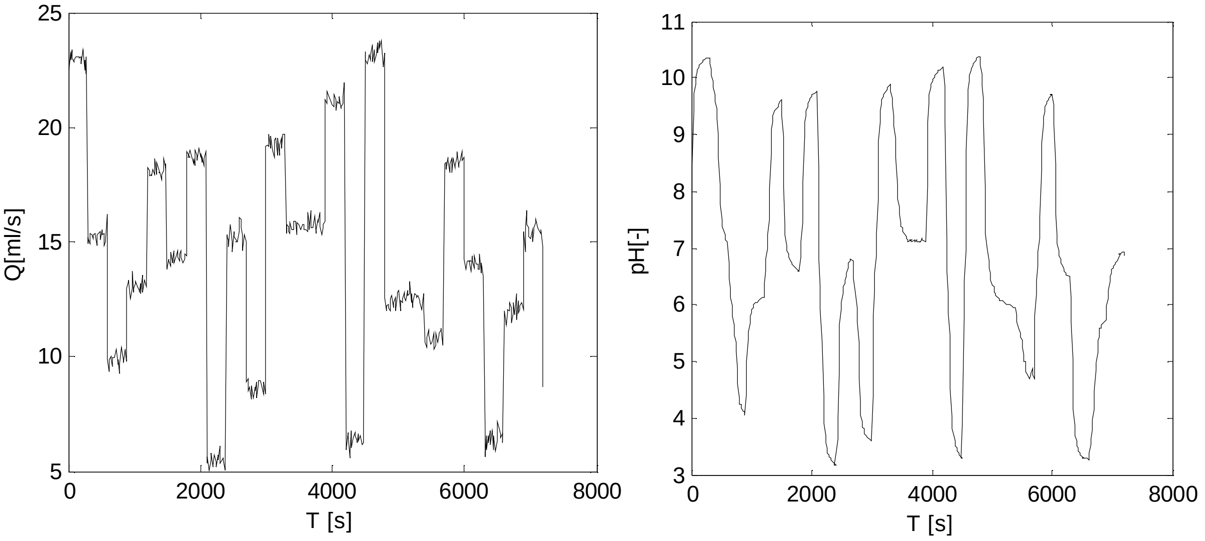

Figure 5 Training data

Training and validation data were obtained using perturbations on the base flow-rate, with sample period of 15s. The local models were chosen to have the form of the second-order ARX model such that the local model network had the form:

$$pH(k) = \sum_{i=1}^{M} r\left(\psi(k)\right) f_i(\varphi(k))$$

where $\psi(k)$ is a vector of scheduling variables $\psi(k) = pH(k-1)$

Five affine local model of the structure

$$f_i(\varphi(k)) = a_0 + a_{i1} pH(k-1) + a_{i2} pH(k-2) + b_{i1} Q_3(k-1) + b_{i2} Q_3(k-2)$$

were used to construct the local model network. For qualitative comparison, the mean sum squared error as in following equation is used as a modeling performance. Here, y is a vector of measured outputs, $\hat{y}$ is a vector of predicted outputs.

$$J = \frac{1}{N}(y-\hat{y})^T (y-\hat{y})$$

Low value of the criterion signifies good modeling performance. The parameters of local models and validity functions were optimized with SOMA algorithm. The output of the resulted model is depicted in Figure 5.

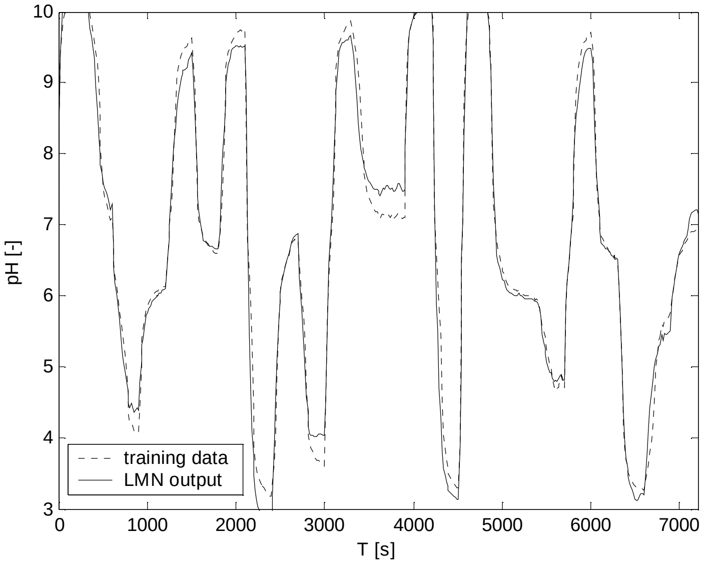

Figure 5 Comparison of the model data

To control pH in the reactor the control scheme from Figure 2 was used. The obtained LMN was used as an internal model. Since the all the local models are affine the controller output was computed using the equation

$$u(k) = \frac{1}{B_0}(v(k) - A_0 - A_1 y(k) - A_2 y(k-1) - B_1 u(k-1)).$$

The discrete filter with transfer function

$$G_F = \frac{0.1}{1 - 0.9z^{-1}}$$

was added to the inverse controller to achieve smooth controller output. The resulted set point tracking performance and controller outputs are shown in Figure 6.

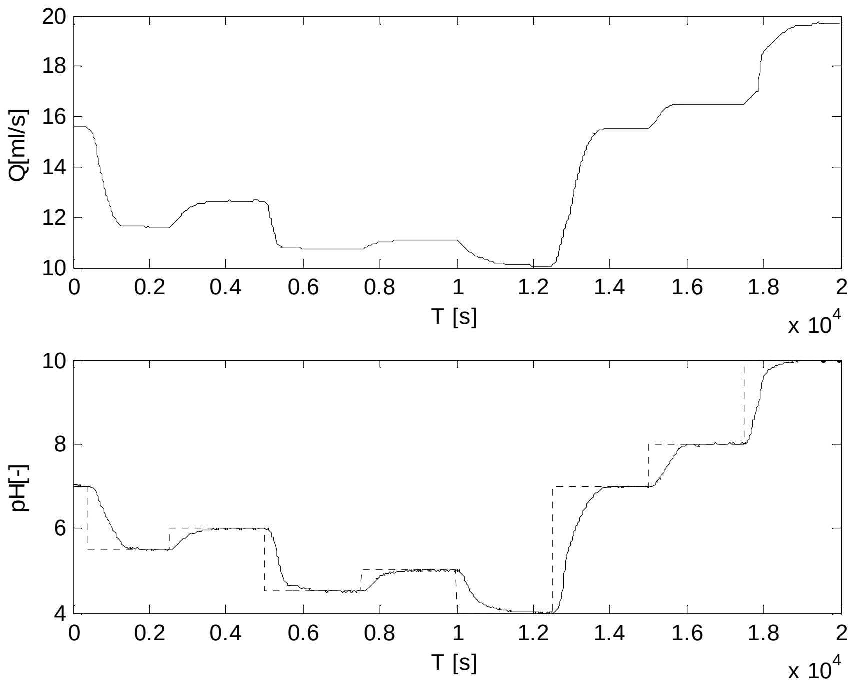

Figure 6 Controller output and set-point tracking

CONCLUSION

The nonlinear model has to represent the system sufficiently and provide easy controller design. Thus the local model network modeling can be viewed as a controller design oriented method. The main idea is to construct a set of local transfer-function models to represent the dynamic system around each operating point, and then to connect the set of local models with validity functions to form a global dynamic model. The SOMA algorithm for the optimization of local model network was proposed. The approach optimizes both the local model parameters and validity function. The SOMA algorithm enables easy integrations of constraints for the local model parameters. The conventional IMC structure is adapted for plant description using local models. Simulation results for pH neutralization plant illustrate the potentional of the genetic algorithms for nonlinear system identification and control.

REFERENCES

Billings, S. and S.Y. Fakhouri (1982). Identification of systems containing linear dynamic and static nonlinear elements. *Automatica*. Vol. 18, no. 1, pages 15-26.

Chen S. and S.A.Billings (1989). Representation of Nonlinear Systems: the NARMAX model. *International Journal of Control*. Vol. 49, no. 3, pages 1013-1032.

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: Proceeding of American Control Conference, Pitsburgh.

Hu, Q., P. Saha and G.P. Rangaiah (2000). Experimental evaluation of an augmented IMC for nonlinear systems. *Control Engineering Practice*. Vol. 8, pages 1167-1176.

Johansen, T.A. and B.A. Foss (1995). Identification of nonlinear system structure and parameters using regime decomposition, *Automatica*, Volume 31.

Morari, M. and E. Zafirou (1989). *Robust Process Control*, Prentice Hall, New Jersey.

Murray-Smith, R. and T.A. Johansen (1997). *Multiple Model Approaches to Modelling and Control*. Taylor and Francis. London, UK.

Narendra K.S. and K. Parthasarathy (1990). Identification and Control of Dynamic Systems using neural networks. *IEEE Transactions on Neural Networks*. Vol. 1, pages 4-27.

Nelles, O. (1997). Orthonormal basis functions for nonlinear system identification with local linear model trees (LOLIMOT), In: *Proceeding of 11th IFAC Symposium on System Identification.* Kitakyushu, Fukuoka, Japan.

Shorten R. and R. Murray-Smith (1996). Side effects of Normalising Radial Basis Function networks. *International Journal of Neural Systems*. Vol. 7, No. 2.

Townsend, S., G. Lightbody, M.D. Brown and Q.W. Irwin (1998). Nonlinear Dynamic Matrix Control Using Local Models. *Transactions of the Institute of Measurement and Control*. Vol. 20, No. 1, pages 47-56.

Zelinka, I. (2002). *Umělá Inteligence v Problémech Globální Optimalizace*. BEN – technická literatura, Praha..