MULTIPLE MODEL MODELLING AND PREDICTIVE CONTROL

J. Novak, V. Bobal

Faculty of Applied Informatics
Tomas Bata University in Zlin
Nad Stranemi 4511, Zlin 760 05, Czech Republic
fax : +420 576035279 and e-mail :jnovak@fai.utb.cz

Abstract: This paper proposes multiple model modelling and predictive control based on Local Model Networks (LMN). An LMN modelling method uses genetic algorithms for optimization of the parameters of local models and validity functions. Since each model is designed for different operating conditions, different predictive controllers are designed for each local model. The presented modelling and control design procedures are demonstrated on the model of pH neutralization process.

Keywords: Predictive control, Local model networks

INTRODUCTION

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 *et al*. 1989), Hammerstein models (Billings *et al*. 1982), or Multiple Layer Perceptron (MLP) neural network (Narendra *et al*. 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 modelling and control domain. An appealing approach is to decompose a complex nonlinear control problem in a number of simpler linear problems each associated with restricted operating region. Local Model Network employs this divide-and–conquer strategy of dividing a complex problem into several simple sub-problems, whose individual solution yields the solution to the complex problem. Local model network (Murray-Smith 1994) belongs to the class of multiple model approaches with interpolation, wherein a small number of relatively simple models are blended together. Typically each local linear or affine model is associated with a corresponding weighting function that defines the validity of the model. The role of the blending is to provide smooth interpolation between the outputs of local models to achieve accurate representation only with a small number of models. The LMN framework provides transparency and enables incorporation of a priori knowledge which is important for practical applications. The resulting nonlinear representation is either a discrete or continuous-time one depending on the form of the local models. Although much more attention to date has been given to the discrete-time models, several works concerning the continuous-time case can be found (Ali 2003), (McLoone *et al*. 1998).

Two different multiple model controller design methods can be employed to maintain the performance. In one case, a controller is designed for each local model and the control action is then weighted based on the prediction error or validity functions (Schott et al. 1994). Dougherty and Cooper developed a multiple model control strategy for dynamic matrix control (DMC), where outputs of multiple linear DMC controllers are weighted to obtain an adaptive DMC controller (Dougherty et al. 2003). Multiple Model Predictive Control (MMPC) was proposed in (Li et al. 2004) which used different predictive controllers for different fuzzy rules of the Takagi-Sugeno model of the process. The use of local model networks is not limited only to predictive control. In (Brown et al. 1999) local GMV controllers were used to form a nonlinear controller network. Model-based predictive controller that is based on a local linear approximation of the fuzzy-based process model around the current operating point has been developed in (Abonyi et al. 2001).

LOCAL MODEL NETWORKS

Local Model networks (LMN), first introduced by Johansen and Foss, describe a set of sub-models, each valid for a specific regime in the operating space, weighted by some activation function. LMN is a generalization of the radial basis function network (RBF), 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 1997).

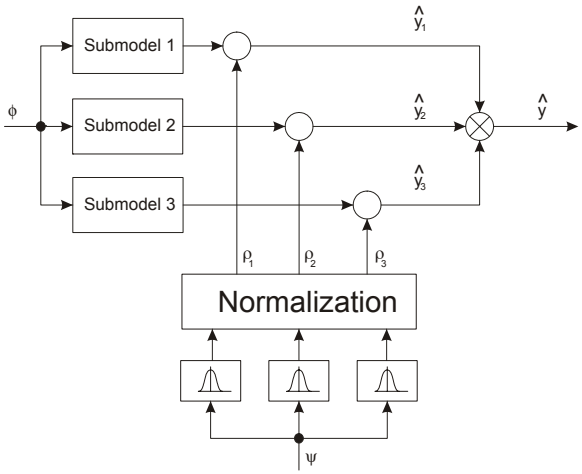

Fig. 1. Local Model Network

Here, the same input, *u*, is fed to all the models and the outputs are weighted according to some scheduling variable or variables, ψ . The underlying local models can be either linear or nonlinear. The LM network output is given by:

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

where $\boldsymbol{\psi}$ is a vector of scheduling variables, ρ_i is a validity function and $\hat{y}_i(k)$ is the output of the *i*-th model.

The LMN are based on the *divide-and-conquer* strategy where the operating space of the nonlinear system is divided into sub-regions and the nonlinear model is achieved using blending between the local models associated with each sub-region. The assumption for the local modelling approach is that the modelled plant has to undergo significant changes in operating conditions as it moves in the operating space. Introducing the simpler models can reduce the complexity of the nonlinear system.

The blending of local models is calculated using the weighting or validity functions. Although any function with a locally limited activation might be applied as a validity function, a common choice for this function takes the form of Gaussian. Other popular validity functions as B-splines or multiquadratic functions have been proposed.

Gaussian basis functions are the most common choices for weighting the outputs of local models. The Gauss function for *i*-th model is given by

$$\rho_i(\psi) = \exp\left(-\frac{1}{2}(\psi - c_i)^T \sigma_i^{-2} (\psi - c_i)\right) \qquad (2)$$

where parameters c_i and σ_i^2, define the Gaussian centre and width, respectively and the scheduling variable ψ can be a system state or any system variable. The centre point defines where the model is most active. The number of centres and widths for each model depends on the number of scheduling variables. For one scheduling variable the weighting function is a typical bell-shaped curve.

Normalization of the basis function is often motivated by the desire to achieve a partition of unity. By partition of unity, it is meant that, at any point in the input space sum of all normalized basis functions equals unity, i.e.

$$\sum_{i=1}^{M} \rho_i(\boldsymbol{\psi}) = 1 \qquad (3)$$

Validity function represents the partition of the input space in the local model network structure. The normalized form of the validity function is denoted by $\tilde{\rho}_i(\boldsymbol{\psi})$. For the basis functions associated with local model *i* the normalized validity function takes form

$$\rho_i(\psi) = \frac{\tilde{\rho}_i(\psi)}{\sum_{i=1}^{M} \tilde{\rho}_i(\psi)} \qquad (4)$$

However, normalization also leads to a number of other important side-effects that have consequences for the resulting network.

MODELLING USING THE GENETIC ALGORITHMS

Training is a key issue in the application of the Local Model Networks since, in addition to parameter identification, there is the added complexity of having to determine the number and the structure of sub-models as well as the parameters of the validity functions. There are several methods for modelling a nonlinear system and the choice of a particular modelling method depends on the aim of modelling. If the aim of modelling is control design then the identification technique should lead to simple, transparent and mathematically tractable models. In many applications it is necessary to combine the information obtained from the numerical data with heuristic knowledge. Another major requirement for nonlinear system modelling algorithm is the universalness in the sense that a wide class of structurally different systems can be described. The

described architecture of local model networks is capable of fulfilling these requirements and can be applied to tasks where high degree of flexibility is required.

A major advantage of the local model networks is that they are not only useful architectures for general learning tasks, but that it is relatively easy to introduce the *a priori* knowledge about a particular problem. A priori information is initial knowledge about the system or a problem in question. *A priori* knowledge includes goals of the problem, characteristics of the system, its parameters and effect of the environment (disturbances and noise). The most general form of information is the expected dynamic order of the models, the form of the models (e.g. ARX models) and the sampling period. In many cases, there will not be sufficient data to train the model throughout the input space, especially outside the areas of normal operation. This can be overcome by fixing *a priori* models to the areas where system is well understood and applying the learning techniques only where data is available and reliable. A further option, for cases where a complex and too complicated model exists and is valid for certain operating regimes, is to pre-set the fixed local models obtained through the linearization of the nonlinear model. The linearization of the nonlinear model can also be used for comparison with the model obtained from the experimental data.

For LMN optimization with local ARX models an individual that represents possible solution of the optimization problem have the following structure:

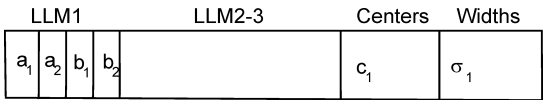

Fig. 2. Individual representing possible solution of the optimization problem

Usage of all the model parameters for optimization leads to the same problems as in the global learning technique since each local model influences all the other models. Possible solution is combination of SOMA optimization algorithm for optimization of the validity function parameters combined with the weighted least-squared method for local model parameters estimation.

MODEL PREDICTIVE CONTROL

Predictive control theory is a powerful tool to deal with process control problems due to its ability to deal with constraints and MIMO systems. Unfortunately, two major issues limit its application to nonlinear systems. Firstly, the assumption for the predictive control is a quite accurate model of the system. Unfortunately, complex systems are often connected with nonlinearities, wide operating range or uncertainty. The second is that a nonlinear non-convex optimization problem must be solved for each sampling period with very computationally demanding algorithms.

Kuipers and Astrom in (Kuipers et al. 1994) proposed a heterogeneous control law where several controllers were built for different process operating regions. The global controller output is obtained by combining the local controller outputs. The local models contained in the LMN can be used to develop a local predictive controller such as Generalised Predictive Controller (GPC). The advantage of this approach is that analytical solution can be used to obtain a controller output of each controller. Hence, the time consuming numerical search procedures and uncertainty in converge to the global optimum can be avoided.

Consider the following local linear model

$$A(z^{-1})y(k) = B(z^{-1})u(k-1) \tag{5}$$

where A and B polynomial in the backward shift operator z^{-1} are given as

$$\begin{aligned} A(z^{-1}) &= 1 + a_1 z^{-1} + \ldots + a_{na} z^{-na} \\ B(z^{-1}) &= 1 + b_1 z^{-1} + \ldots + b_{nb} z^{-nb} \end{aligned} \tag{6}$$

To derive a j-step ahead predictor of *y(k+j)*, the following Diophantine equation has to be solved

$$1 = E_j(z^{-1})A(z^{-1})\Delta + z^{-1}F_j(z^{-1}) \tag{7}$$

where $E_j(z^{-1})$ and $F_j(z^{-1})$ are polynomial uniquely defined given the $A(z^{-1})$ and the prediction horizon *j*. Δ is a difference operator $1 - z^{-1}$. The *j*-step ahead prediction of y, given the measured data up to time *k* is given

$$\hat{y}(k+j|k) = G_j(z^{-1})\Delta u(k+j-1) + F_j(z^{-1})y(k) \tag{8}$$

where

$$G_j(z^{-1}) = E_j(z^{-1})B(z^{-1}) \tag{9}$$

The Equation 8 can be rewritten for the prediction horizon *N* to the vector form

$$\hat{\boldsymbol{Y}} = \boldsymbol{G}\boldsymbol{U} + \boldsymbol{F} \tag{10}$$

where

$$\begin{aligned} \hat{\boldsymbol{Y}} &= \left[\hat{y}(k+1)\ \hat{y}(k+2) \ldots y(k+N)\right]^T \\ \boldsymbol{U} &= \left[\Delta u(k)\ \Delta u(k+1) \ldots \Delta u(k+N-1)\right]^T \\ \boldsymbol{F} &= \left[f(k+1)\ f(k+2) \ldots f(k+N)\right]^T \end{aligned} \tag{11}$$

The matrix $\boldsymbol{G}$ is lower-triangular of dimension *NxN*

$$G = \begin{bmatrix} g_0 & 0 & 0 & 0 \\ g_1 & g_0 & \vdots & 0 \\ \vdots & \vdots & \ddots & \\ g_{N-1} & g_{N-2} & \cdots & g_0 \end{bmatrix} \quad (12)$$

The unconstrained minimization of the objective function

$$J = (\boldsymbol{W} - \boldsymbol{Y})^T (\boldsymbol{W} - \boldsymbol{Y}) + \lambda \boldsymbol{U}^T \boldsymbol{U} \quad (13)$$

leads to the following optimal control strategy:

$$\boldsymbol{U} = \left(\boldsymbol{G}^T \boldsymbol{G} + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{G}^T (\boldsymbol{W} - \boldsymbol{F}) \quad (14)$$

The control action contains the integral action which provides zero static control offset.

For each local model, the local controller can be designed using the GPC algorithm. The multi-model predictive control strategy is shown in Fig. 3. Because each model is linear equation, it is easy to design a linear controller for it. The global nonlinear controller is the sum of local ones multiplied with corresponding validity function.

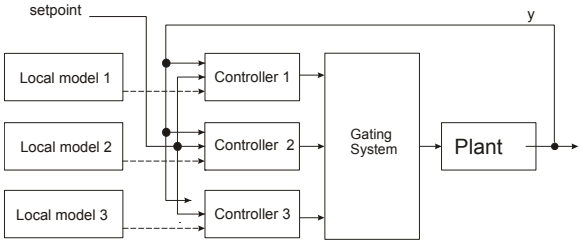

Fig. 3 Multi-model predictive control

APPLICATION TO CSTR PROCESS

Continuous stirred tank reactor (CSTR) is highly nonlinear process. A schematic of the CSTR system

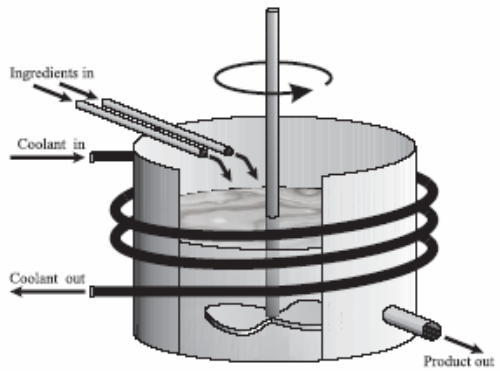

Fig. 4 CSTR scheme

Continuous stirred tank reactor (CSTR) is highly nonlinear process. A schematic of the CSTR system is shown in Fig. 4. The reaction is exothermic and produces heat which slows down the reaction. The dynamics of the system can be described by the two following differential equations:

$$\begin{aligned} \dot{C}_a(t) &= \frac{q_f}{V}\left(C_{a0} - C_a(t)\right) - k_0 C_a(t) e^{\frac{-E}{RT(t)}} \\ \dot{T}(t) &= \frac{Q}{V}\left(T_0 - T(t)\right) - \frac{\triangle H k_0}{\rho C_p} C_a(t) e^{\frac{-E}{RT(t)}} + \\ &+ \frac{\rho_c C_{pc}}{\rho C_p V} q_c(t) \left(1 - e^{\frac{h_a}{\rho_c C_{pc} q_c(t)}}\right)\left(T_{c0} - T(t)\right) \end{aligned} \quad (15)$$

The parameters of the model were taken from (Gao et al. 2002). The objective is to control the concentration *c(t)* by manipulating coolant flow rate $q_c(t)$. The open-loop step test shows that the responses vary from under-damped to over-dumped.

MODELLING THE PROCESS

The stable equilibrium regime lies in $c(t) \in (0, 1.3566)\, mol/l$. The global model with 5 local models was chosen as a trade-off between the complexity of the model and its performance. The parameters of validity functions were optimized using the genetic algorithm SOMA (Zelinka 2004) while the parameters of the local models were obtained by quadratic programming which enables the implementation of *a priori* information about the process such as: gain, stability etc. All the models were assumed to be stable with the positive gain. The training data were generated by perturbation of the input signal (Figure 5).

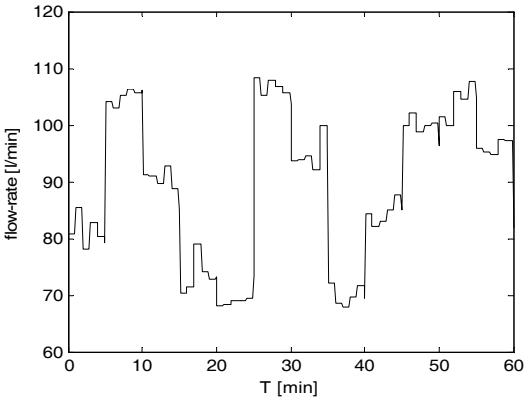

Fig. 5 Input training signal

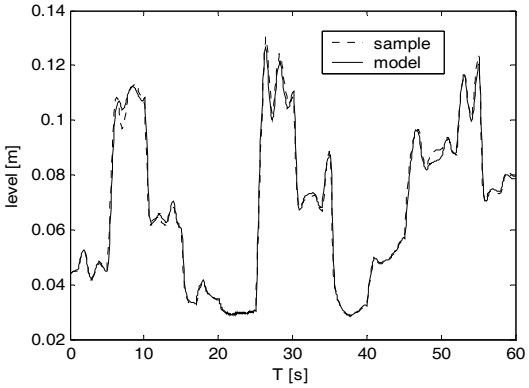

Fig. 6 LMN predictions on training data

Figure 6 shows the modelling results of SOMA algorithm. The predictions of the model are computed as infinity number of steps ahead predictions, which means that the new prediction are included in the regression vector.

GPC CONTROLLER FOR CSTR

The global nonlinear controller is formulated by blending the local controllers through the gating system resulted from the LMN structure. The global performance of the controller depends on the performance of the local controllers. The prediction horizon of all the controllers is set to *N=20 steps*. The controller output is penalized by penalization constant $\lambda = 0.0001$.

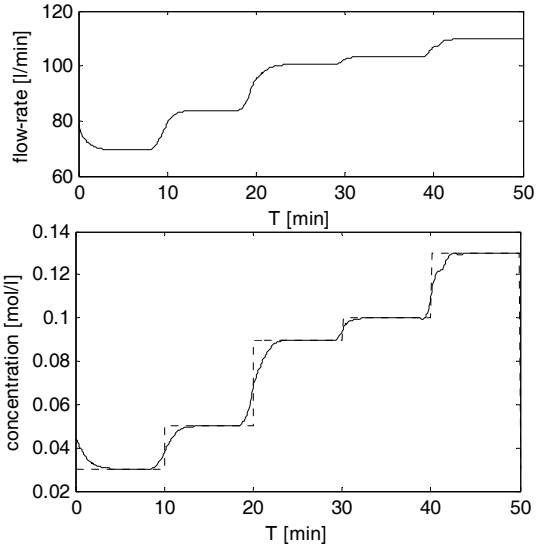

Fig. 7 Generalized predictive control of the CSTR process – multiple models

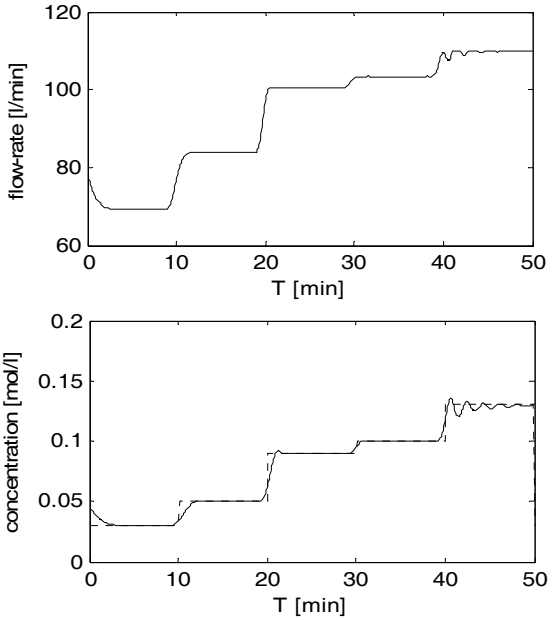

Fig. 8 Generalized predictive control of the CSTR process – single model

Figure 8 and 9 show the performance of the LMN based predictive controller and single model predictive controller. It can be seen that the response is oscillating in some operating regions. However, the response under the LMN controller is satisfactory in all operating regions. This can be contributed to the fact that the response in this region is determined mainly by the corresponding controller.

CONCLUSION

Local Model Networks are very effective in modelling nonlinear processes which have different dynamic properties in different operating regions. LM networks are also easier to interpret than conventional MLP networks. Network provides a basis for creating a controller network. The global controller is obtained by blending the local controller's outputs. The disadvantage of the nonlinear model predictive control is the time consuming numerical optimisation algorithms and uncertainty in the global optimum. If the global controller is combined from multiple linear controllers the solution of the optimization problem can be found analytically. Moreover the control actions produced from the local controllers contain the integral action which removes the steady-state error.

ACKNOWLEDGMENT

The work has been supported by Ministry of Education of the Czech Republic under the grant 1M360567023. This support is very gratefully acknowledged.

REFERENCES

Abonyi, J., L. Nagy and F. Szeifert. (2001). Fuzzy model-based predictive control by instantaneous linearization. *Fuzzy Sets and Systems*, Vol. 120. pp. 109-122.

Ali, A. 2003. Application of Local Model Networks to Nonlinear System Identification and Control. Ph. D. Thesis. Bochum, Germany.

Billings, S. and S.Y. Fakhouri. (1987). Identification of systems containing linear dynamic and static nonlinear elements. *Automatica*, Vol. 18. pp 15-26.

Brown, M.D. and G.W. Irwin. (1999). Nonlinear Identification and Control of Turbogenerators Using Local Model Networks, In: *Proceeedings of the American Control Conference.* San Diego. California.

Chen, S. and S.A.Billings. (1989). Representation of Nonlinear Systems: the NARMAX model. *International Journal of Control*. Vol. 49. pp 1013-1032.

Dougherty, D. and D. Cooper. (2003). A practical multiple model adaptive strategy for single-loop MPC. *Control Engineering Practice*, Vol. 11. pp 141-159.

Gao, R., O'dywer, A. and E. Coyle. (2002). A Non-linear PID Controller for CSTR Using Local Model Networks. *In: Proceedings of the 4th World Congress on Intelligent Control and Automation*. Shanghai. China.

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.

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

Kuipers, B. and K. Astrom. (1994). The composition and validation of heterogeneous control laws. Automatica. Vol. 30. pp 233-249.

Li, N S.Y. Li and Y.G. Xi. (2004). Multi-model predictive control based on the Takagi-Sugeno fuzzy models: A case study. *Information Science*, Vol. 3 pp. 247–263.

McLoone, S.C. and G.W. Irwin. (1998). Continuous-time multiple model networks. *In: Proceedings of the Irish Signals and Systems Conference*. Dublin. Ireland.

Murray-Smith, R. (1994). A local model network approach to nonlinear modelling. Ph. D. Thesis. University of Strathclyde, UK.

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. pp 4-27.

Shorten, R. and R. Murray-Smith. (1996). Side effects of Normalising Radial Basis Function networks. *International Journal of Neural Systems*. Vol. 2.

Schott, K.D. and B.W. Bequette. (1994). Control of Nonlinear Chemical Processes using Multiple-model Adaptive Control. AICHE Annual Meeting. San Francisco. California.

Zelinka, I. (2004) SOMA – Self Organizing Migrating Algorithm, Chapter 7 in Babu, B.V., Onwubolu, G.(eds), *New Optimization Techniques in Engineering*. Springer-Verlag. Heidelberg, Germany.