

Nonlinear Model Predictive Control Utilizing a Neuro-Fuzzy Predictor

Waller, Jonas B.

Process Control Lab, Abo Akademi University | Department of Electrical and Electronic Systems Engineering, Faculty of Information Science and Electrical Engineering, Kyushu University : Visiting Researcher

Hu, Jinglu

Department of Electrical and Electronic Systems Engineering, Faculty of Information Science and Electrical Engineering, Kyushu University

Hirasawa, Kotaro

Department of Electrical and Electronic Systems Engineering, Faculty of Information Science and Electrical Engineering, Kyushu University

<https://doi.org/10.15017/1515714>

出版情報 : 九州大学大学院システム情報科学紀要. 6 (1), pp.39-45, 2001-03-26. 九州大学大学院システム情報科学研究所

バージョン :

権利関係 :

Nonlinear Model Predictive Control Utilizing a Neuro-Fuzzy Predictor

Jonas B. Waller* , Jinglu HU** and Kotaro HIRASAWA**

(Received December 15, 2000)

Abstract: This paper applies a quasi-ARMAX modeling technique, recently presented in the literature, to a process control framework. The use of this quasi-ARMAX modeling technique in nonlinear model predictive control (NMPC) formulations applied to simple nonlinear process control examples is investigated. The quasi-ARMAX predictor can be interpreted as a neuro-fuzzy predictor, and this neuro-fuzzy predictor is computationally straightforward and has showed excellent prediction capabilities. The predictor is thus well suited for NMPC purposes. Furthermore, the parameters of the neuro-fuzzy model can be argued to have explicit meaning, thus making the procedure of tuning the NMPC system more transparent when using the neuro-fuzzy predictor.

Keywords: Model predictive control, Nonlinear control, Neuro fuzzy models, Nonlinear identification

1. Introduction

Model predictive control (MPC) has received a strong position when it comes to industrially implemented advanced control methodologies, especially in refinery and petrochemical fields¹⁾²⁾. One main reason for this is the intuitive way MPC incorporates the process model in the controller design, thus e.g. making it straightforward to take constraints into account at the design stage.

In many problems relevant in the process control field today, the plant under control shows a strongly non-linear behavior. As a means to handle this, non-linear MPC (NMPC) is an often used method. NMPC, simply put, is model predictive control, where a non-linear process model is used for prediction purposes, as opposed to a linear model for basic MPC. See e.g. Camacho and Bordons³⁾ for an overview of MPC, and Henson⁴⁾ and Rawlings et al.⁵⁾ for a focus on NMPC.

One main difficulty with non-linear MPC is that the non-linear models needed often are complex and give rise to computationally burdensome optimization problems. Since MPC requires the solution of an optimization problem on-line in every sampling instant, the computational simplicity is crucial.

This paper lists several non-linear process models used in recent literature for NMPC purposes, useful as an indicator of the research activity in the field. A new contribution of this paper is adapt-

ing a quasi-ARMAX modeling technique, recently presented in the literature⁶⁾⁷⁾, for non-linear MPC purposes. The idea of the quasi-ARMAX modeling scheme is to incorporate a group of certain non-linear non-parametric models (NNMs) into a linear ARMAX structure. The investigated predictor can thus be seen as a linear predictor network consisting of a global linear predictor and several local linear predictors with interpolation. The quasi-ARMAX predictor can in such a way be interpreted as a neuro-fuzzy predictor⁷⁾. The neuro-fuzzy predictor is computationally straightforward and has showed excellent prediction capabilities. In Hu et al.(1998)⁶⁾ and Hu et al. (1999)⁷⁾, the predictor is presented mainly from the perspective of system identification. In this paper, we apply the predictor in a nonlinear process control framework. More specifically, we illustrate how the predictor can be used in a NMPC concept, and illustrate preliminary control results.

Furthermore, non-linear MPC control strategies are often difficult to interpret and understand, thus making the choice of tuning-parameters a quite ad hoc procedure. The parameters of the neuro-fuzzy model can be argued to have explicit meaning⁷⁾, thus making the procedure of tuning the NMPC system more transparent when using the neuro-fuzzy predictor.

2. Nonlinear Model Predictive Control

Nonlinear model predictive control is an open-loop optimal control sequence calculation where the main characteristics of the controller are³⁾

· Explicit use of a nonlinear model to predict the

* Process Control Lab, Abo Akademi University, 20500 Abo, Finland, currently as a Visiting Researcher at Department of Electrical and Electronic Systems Engineering

** Department of Electrical and Electronic Systems Engineering

process output at future time instants (a horizon).
 · Calculation of a control sequence minimizing an open-loop objective function.
 · Receding strategy so that at each instant the horizon is displaced towards the future, which involves the application of the first control signal of the sequence calculated at each step.

The problem formulation for non-linear MPC can briefly be summarized according to the following. See e.g. Henson⁴⁾ for details.

The process model is assumed to have the form of a non-linear discrete-time input-output mapping, e.g.

$$y(k) = g(\varphi(k)) \quad (1)$$

$$\varphi(k) = [y(k-1), \dots, y(k-n), \\ u(k-d), \dots, u(k-m-d+1)]^T$$

The open-loop optimization problem which is solved in every sampling instant can be formulated as

$$\min_{u(k|k), u(k+1|k), \dots, u(k+N_u-1|k)} J \\ \text{where } J = \phi[y(k+N_y|k)] + \\ \sum_{j=0}^{N_y-1} L[y(k+j|k), u(k+j|k), \Delta u(k+j|k)] \quad (2)$$

Where N_u is used to denote the control horizon and N_y denotes the prediction horizon.

We thus have a generally expressed loss function consisting of (quite arbitrary) functions of the predicted outputs and future control actions as well as the successive change in the inputs. Often the loss function is specified as a quadratic loss function such that

$$\phi = [y(k+N_y|k) - y_s(k)]^T Q \\ [y(k+N_y|k) - y_s(k)] \quad (3)$$

$$L = [y(k+j|k) - y_s(k)]^T Q [y(k+j|k) - y_s(k)] \\ + [u(k+j|k) - u_s(k)]^T R [u(k+j|k) - u_s(k)] \\ + [\Delta u(k+j|k)]^T S [\Delta u(k+j|k)] \quad (4)$$

Where $y_s(k)$ and $u_s(k)$ are steady-state targets for y and u , and Q , R and S are weighting matrices. The predicted outputs are obtained from the non-linear process model, equation (1).

The capability to handle constraints is considered one of the main motivations to use (nonlinear) MPC. The non-linear optimization problem formu-

lated in equation (2) is in the case of constraints present solved subject to the input inequality constraints

$$u_{\min} \leq u(k+j|k) \leq u_{\max}, \quad 0 \leq j \leq N_u - 1 \quad (5)$$

$$\Delta u_{\min} \leq \Delta u(k+j|k) \leq \Delta u_{\max}, \quad 0 \leq j \leq N_u - 1$$

and the output inequality constraints

$$y_{\min} \leq y(k+j|k) \leq y_{\max}, \quad 1 \leq j \leq N_y \quad (6)$$

For the case with a linear time-invariant process model, a quadratic loss function and in the absence of constraints, an analytical solution to the optimization problem can be found.

If there are constraints present, and the constraints are convex, the optimization problem is convex and can easily be solved⁸⁾. When the model is non-linear and non-convex constraints are present, a non-convex optimization problem must be solved iteratively at every sampling time. This is the most common case for non-linear MPC.

If no analytical solution can be found, a numerical optimization method has to be used. One could address the solution of nonlinear optimization problems for the purpose of model predictive control for example through successive linearization of model equations, sequential model solution and optimization, and simultaneous model solution and optimization.

Successive linearization of model equations corresponds to the use of the current operating point to linearize the (non-linear) model before each calculation of the control action of the NMPC controller. Different approaches can be taken in an attempt to reduce the error caused by this linearization. For example, in quadratic dynamic matrix control (QDMC) the process model is linearized once for the entire trajectory and in non-linear quadratic matrix control (NLQDMC), the process model is linearized iteratively in each control interval⁹⁾. Sequential model solution and optimization in turn use a standard NLP solver to compute the manipulated inputs, and an ODE solver is used to integrate the non-linear model equations. This is performed iteratively until sufficient accuracy is achieved. Simultaneous model solution and optimization requires that the process model equations are discretized as orthogonal collocation of finite elements¹⁰⁾, in order to solve the nonlinear program. Successive (or

sequential) quadratic programming (SQP) is considered the method of choice for solving non-linear programs. A good treatment of relevant optimization methods is offered in e.g. Gopal and Biegler¹¹).

3. Quasi-Armax Modeling

A key issue to address when designing a NMPC controller is the choice of process model, i.e. the structure of the (non-linear) process model to be used. Here we can roughly divide the area into two (or three) kinds of models. Firstly there are models based on fundamental relationships and secondly there are models based on empirical data. The third kind of model would be one combining fundamental and empirical modeling techniques (which usually is referred to as a hybrid model).

With main focus on empirical non-linear models, one decision to make is the selection of suitable model form. The types of discrete-time non-linear models utilized for NMPC in the recent literature include e.g.

- Hammerstein and Wiener models¹²)
- Volterra models¹³⁾¹⁴⁾
- Polynomial ARMAX¹⁵⁾
- Artificial neural networks¹⁶⁾⁹⁾
- Fuzzy logic models¹⁷⁾⁸⁾

A Hammerstein model consists of a static non-linearity followed by a linear dynamical system, whereas a Wiener model has the order reversed, i.e. the linear dynamics are followed by a static non-linearity. Concerning neural network or fuzzy logic models, several different structures can come in question. Arahall et al.¹⁶⁾ use a multilayer perceptron-based model, and a model based on RBF (radial basis function) networks whereas Rohani et al.⁹⁾ mainly focus on a recurrent neural network and on a feed-forward neural network based model. Concerning the fuzzy models, both Fisher et al.¹⁷⁾ and Sousa et al.⁸⁾ mainly focus on the Takagi-Sukeno fuzzy model.

Here, we shall look at a quasi-ARMAX (or neuro-fuzzy) modeling scheme as a model to be used for in the NMPC formulation. We formulate the problem as a general non-linear, time-invariant system

$$\begin{aligned} y(t) &= g(\varphi(t)) + v(t) \\ \varphi(t) &= [y(t-1), \dots, y(t-n), \\ &\quad u(t-d), \dots, u(t-m-d+1)]^T \\ v(t) &= e(t) + c_1 e(t-1) + \dots + c_p e(t-p) \end{aligned} \quad (7)$$

where $y(t)$ is the output at discrete time-intervals, $u(t)$ is the input, $v(t)$ is a disturbance, $d(t)$ is white

noise, d is the time delay, $\varphi(t)$ is the regression vector and g is a non-linear function.

A hybrid quasi-ARMAX model can be described according to the following. Perform a Taylor expansion of the nonlinear function g , and use two coefficient vectors according to

$$y(t) = g(0) + g'(0)\varphi(t) + \frac{1}{2}\varphi^T(t)g''(0)\varphi(t) + \dots$$

$$\theta = g'(0)^T = [a_1 \dots a_n \quad b_1 \dots b_m]^T$$

$$\begin{aligned} \Delta\theta_t &= \left(\frac{1}{2}\varphi^T(t)g''(0) + \dots\right)^T \\ &= [\Delta a_{1,t} \dots \Delta a_{n,t} \quad \Delta b_{1,t} \dots \Delta b_{m,t}]^T \end{aligned}$$

(where $g(0)$ is ignored for the sake of simplicity.) The system can thus be expressed as a combined structure

$$y(t) = \varphi^T(t)(\theta + \Delta\theta_t) + v(t) \quad (8)$$

The parameters ($\Delta\theta_t$) can be approximated⁶⁾ as

$$\begin{aligned} \Delta a_{i,t} &= f_i(\varphi(t)), \quad i = 1, \dots, n \\ \Delta b_{j,t} &= f_{j+n}(\varphi(t)), \quad j = 1, \dots, m \end{aligned}$$

where

$$f_i(\varphi(t)) = \sum_{j=1}^M \omega_{ij} N_f(p_j, \varphi(t)) \quad (9)$$

where $N_f(p_j, \varphi(t))$ are fuzzy basis functions, ω_{ij} are the coordinate parameters and p_j are the scale and position parameter vectors.

$$N_f(p_j, \varphi(t)) = \frac{\Lambda_{k=1}^r \mu_{A_k^j}(x_k(t))}{\sum_{j=1}^L (\Lambda_{k=1}^r \mu_{A_k^j}(x_k(t)))} \quad (10)$$

We use Λ as the minimum operator, $r = \dim(\phi(t))$, L is the number of fuzzy rules, $x_k(t)$ are the elements of $\phi(t)$, and $\mu_{A_k^j}$ is the membership function of fuzzy set A_k^j . $\mu_{A_k^j}$ may be a Gaussian function defined by

$$\mu_{A_k^j}(x_k(t)) = \exp\left[-\frac{1}{2}\left(\frac{x_k(t) - \bar{x}_k^j}{\sigma^j}\right)^2\right] \quad (11)$$

In this case, the parameter vector is defined by

$$p_j = [\bar{x}_1^j \ \bar{x}_2^j \ \dots \ \bar{x}_r^j \ \sigma^j]^T, \quad j = 1, \dots, L \quad (12)$$

Our model can then be reformulated as a combined structure

$$y(t) = \varphi^T(t)\theta + v(t) + \sum_{j=1}^M \varphi^T(t)\Omega_j N_f(p_j, \varphi(t))$$

where $\Omega_j = [\omega_{1j} \ \dots \ \omega_{rj}]$.

Or, alternatively, the system can be expressed as a linear regression structure;

$$y(t) = \varphi_{NL}^T(t)\Theta + e(t) \quad (13)$$

where

$$\Theta = [\theta^T \ \omega_{11} \ \dots \ \omega_{rL} \ c_1 \ \dots \ c_p]^T$$

$$\varphi_{NL}(t) = [\varphi^T(t)\varphi^T(t) \otimes \varphi_{N_f}^T(t) \\ e(t-1) \ \dots \ e(t-p)]^T$$

$$\varphi_{N_f}^T(t) = [N_f(p_j, \varphi(t)), \quad j = 2, \dots, L]$$

$$\theta = [a_1 \ \dots \ a_n \ b_1 \ \dots \ b_m]$$

Kronecker production is indicated by \otimes . This gives us an implementable algorithm, where the parameters Θ can be evaluated using e.g. a recursive least squares algorithm. This is the form of the algorithm implemented in the simulations below. This form is also usable for adaptive control.

The investigated predictor can be seen as a learning network based predictor interpreted as a linear predictor network consisting of a global linear predictor and several local linear predictors with interpolation. This predictor should thus be well suited for control purposes in e.g. process control, when the plant in question has to run at several different operating points. Furthermore, the predictor is computationally efficient, thus making it exceptionally well suited for use in NMPC, where the computational simplicity is crucial, since a non-linear optimization problem has to be solved in every sampling instant.

4. Quasi-Armax NMPC

The predictor presented above has, for control purposes, only been illustrated using a one-step ahead predictive controller⁷⁾, often referred to as a

myopic controller.

The myopic controller can be seen as the most simple case of model predictive control, i.e. the case with a prediction horizon and the control horizon = 1. It can quickly be illustrated according to the following. Choose a controller which minimizes the cost function

$$J = \frac{1}{2}(y(k+1) - y_r(k+1))^2 + \frac{\lambda}{2}u(k)^2$$

where λ is a tuning factor. Substitute $y(k+1)$ with its prediction $\hat{y}(k+1)$, where the prediction is given by the model used.

We look at the example used in⁷⁾, where the system under control is given by

$$y(k) = G(z^{-1})z(k), \quad (14) \\ G(z^{-1}) = \frac{0.7z^{-1} - 0.68z^{-2}}{1 - 1.72z^{-1} + 0.74z^{-2}}$$

where $z(k)$ is a nonlinear function of $u(k)$ given by

$$z(k) = \begin{cases} u(k) - 1.75, & u(k) > 2 \\ 0.0625\text{sign}(u(k))u(k)^2, & |u(k)| \leq 2 \\ u(k) + 1.75, & u(k) < -2 \end{cases} \quad (15)$$

which behaves like a dead zone. Assume that a linear ARX-model is used to model the system

$$y(k+1) = -a_1y(k) - a_2y(k-1) \\ + b_1u(k) + b_2u(k-1) \\ = 1.72y(k) - 0.74y(k-1) \\ + 0.7u(k) - 0.68u(k-1)$$

Taking the derivative of the loss function J with respect to $u(k)$ and equaling it to zero gives an analytical expression for the control action

$$u(k) = \frac{b_1}{(b_1^2 + \lambda)} [a_1y(k) + a_2y(k-1) \\ - b_2u(k-1) + y_r(k+1)] \quad (16)$$

The myopic controller, although useful for illustrative reasons due to its simplicity, is not practically very useful for process control applications. A myopic controller e.g. requires the system to be minimum-phase and its I/O delay exactly known. This is a severely limiting assumption, especially in process control.

For the full NMPC case, we will minimize a cost function given by

$$J(N_1, N_y, N_u) = \sum_{j=N_1}^{N_y} \sigma(j) [\hat{y}(t+j|t) - y_r(t+j)]^2 + \sum_{j=1}^{N_u} \lambda(j) [u(t+j-1)]^2 \quad (17)$$

In our example the weights are chosen as $\sigma = 1$ and $\lambda = \text{constant}$. Further, in our example we choose $N_1 = 1$, $N_y = 3$ (the prediction horizon) and $N_u = 3$ (the control horizon). Also we use the change of the control input instead of the control input itself, in order to smoothen the control input. In process control, the smoothness of the control action is often of significance, in order to e.g. reduce wear on actuators.

$$J(k) = \sum_{j=1}^3 (\hat{y}(k+j) - y_r(k+j))^2 + \sum_{j=1}^3 \lambda \Delta u(k+j-1)^2 \quad (18)$$

For the example above (given by equation 14 and 15) we can identify a quasi-ARMAX model with e.g. 5 parameters in the global model, and 18 local models, active in different regions of the input space. The identification is performed with a recursive least squares algorithm, and the identified model output compares to the actual process output and to the output of an identified linear model as shown in **Fig.1**.

The NMPC controller can then be implemented according to minimizing equation 18, where

$$\hat{y}(k+1) = \varphi_{NL}^T(k+1)\Theta$$

$$\begin{aligned} \varphi_{NL}(k+1) &= [\varphi^T(k+1) \varphi^T(k+1) \otimes \varphi_{N_f}(k+1)]^T \\ \varphi_{N_f}^T(k+1) &= [N_f(p_j, \varphi(k+1), j=1, \dots, L) \\ \varphi(k+1) &= [y(k), \dots, y(k-n+1), \\ & \quad u(k-d+1), \dots, u(k-m-d+2)]^T \end{aligned}$$

and $\hat{y}(k+j)$, $j = 2, 3, \dots$ are calculated recursively. The minimization is performed at every sample instant, and only the first calculated control move ($u(k)$) is implemented. The parameter vector p_j is kept fixed. Control results for the deterministic case are illustrated in **Fig.2**.

Another example, useful for demonstrating the principles of the proposed controller consists of a process working in a few distinct operating regions. In each region the process can be described by a

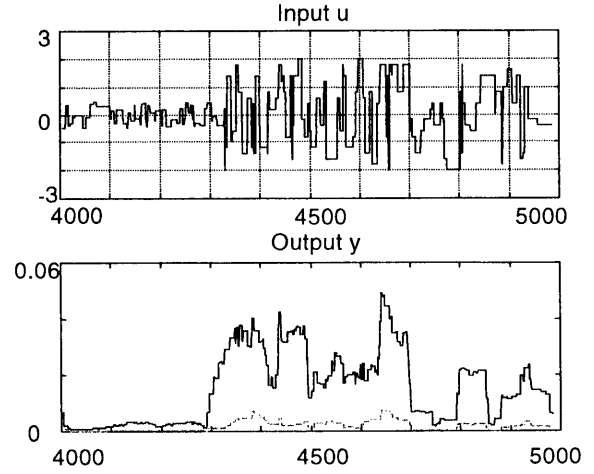


Fig.1 A comparison of identification results on system in equation 14 - 15. Upper graph shows the current input sequence, lower graph shows filtered errors of the linear model (solid, blue line) and of the neuro-fuzzy model (dashed, green line). Models have been identified through experiments with a PRBS sequence, similar to the upper graph.

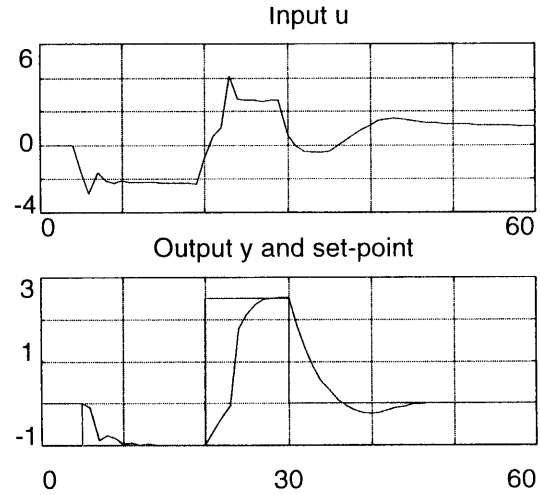


Fig.2 Set-point changes for NMPC control of process given in equations 14 and 15. The model used for control is identified according to **Fig.1**. The tuning factor $\lambda = 0.2$, and control and prediction horizons are equal to 3.

linear model. Assume dynamics for three regions according to

$$\begin{aligned} \frac{y(z^{-1})}{u(z^{-1})} &= \frac{0.0137z^{-1} + 0.0092z^{-2}}{1 - 1.1866z^{-1} + 0.3012z^{-2}}, \quad y < -1 \\ \frac{y(z^{-1})}{u(z^{-1})} &= \frac{0.0274z^{-1} + 0.0184z^{-2}}{1 - 1.1866z^{-1} + 0.3012z^{-2}}, \quad |y| < 1 \\ \frac{y(z^{-1})}{u(z^{-1})} &= \frac{0.0548z^{-1} + 0.0368z^{-2}}{1 - 1.1866z^{-1} + 0.3012z^{-2}}, \quad y > 1 \end{aligned}$$

i.e. the process gain changes with the operating region. The usefulness of the physical interpretation

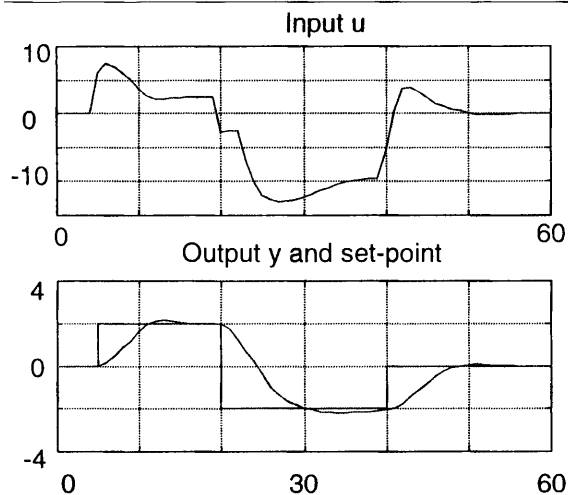


Fig.3 Control of piecewise linear process with NMPC controller. The controller works well in all regions.

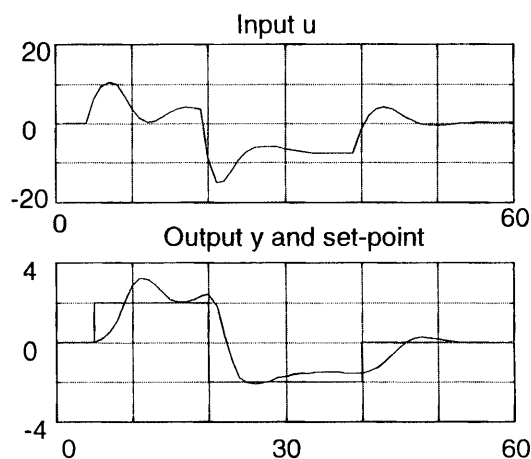


Fig.4 Control of piecewise linear process with linear MPC controller. The linear MPC cannot operate smoothly in all three regions.

of the parameters can be demonstrated through this (oversimplified) example. Let the global model correspond to the parameters in the region $-1 < y < 1$, and the neuro-fuzzy local models consist of the difference between the global model and the model at the current operating region. The fuzzy basis functions are defined so as to correspond to each of the distinct operating regions. Control results with this controller (prediction and control horizon = 3, and a tuning factor equal to 0.05) gives the simulated set-point changes in **Fig.3**, and as comparison a linear MPC controller is shown in **Fig.4**.

5. Conclusion

The purpose of this paper is to relate the quasi-ARMAX identification method to a thorough process control framework, thus this paper presents the derivation and application of a nonlinear quasi-

ARMAX predictor to nonlinear model predictive control purposes.

The proposed controller shows good results, being able to combine good prediction capability with computational efficiency. The controller proposed should be well suited for use in the NMPC formulation, since the identification algorithm is computationally more efficient than e.g. a neural network based model⁶⁾.

Furthermore, the specific meaning of the parameters makes the predictor well suited for NMPC control, since the model thus can be understood and analyzed based on fundamental properties of the process to be controlled.

Several issues are still open concerning this modeling approach for use in NMPC applications. The robustness of the resulting control system seems promising, but a more rigorous evaluation should be performed. Furthermore, even though the parameters of the neuro-fuzzy model have explicit meanings, the tuning of the controller as a whole is still a quite ad hoc procedure. Also here, more thorough investigations are needed.

Acknowledgments

Jonas Waller gratefully acknowledges financial support from Svenska Tekniska Vetenskapsakademien i Finland.

References

- 1) S. Joe Qin and T.A. Badgwell, "An Overview of Nonlinear Model Predictive Control Applications", *Proc. Workshop on NMPC*, Ascona, Switzerland, 1998.
- 2) H. Takatsu, T. Itoh and M. Araki, "Future needs for the control theory in industries - report and topics of the control technology survey in Japanese industry", *J. Proc. Cont.*, vol. 8, no 5-6, pp. 369-374, 1998.
- 3) E.F. Camacho and C. Bordons, *Model Predictive Control*, Springer Verlag, 1999.
- 4) M.A. Henson, "Non-linear model predictive control: current status and future directions", *Computers chem. Engng.*, vol. 23, pp. 187-202, 1998.
- 5) J.B. Rawlings, E.S. Meadows and K.R. Muske, "Non-linear model predictive control: A tutorial and a survey", *Preprints IFAC Symposium ADCHEM '94*, Kyoto Research Park, Kyoto, Japan, 1994.
- 6) J. Hu, K. Kumamuru, K. Inoue and K. Hirasawa, "A Hybrid Quasi-ARMAX Modeling Scheme for Identification of Nonlinear Systems", *Trans. SICE*, vol. 34, pp. 977-986, 1998.
- 7) J. Hu, K. Hirasawa and K. Kumamuru, "Adaptive Predictor for Control of Nonlinear Systems Based on Neurofuzzy Models", *Proc. ECC*, Karlsruhe, 1999.
- 8) J.M. Sousa, R. Babuska and H.B. Verbruggen, "Fuzzy Predictive Control Applied to an Air-Conditioning Sys-

- tem”, *Control Eng. Practice*, vol. 5, no 10, pp. 1395-1406, 1997.
- 9) S. Rohani, M. Haeri and H.C. Wood, “Modeling and control of a continuous crystallization process. Part 1. Linear and non-linear modeling”, *Computers chem. Engng.*, vol. 23, pp. 263-277, 1999.
- 10) B. Wayne Bequette, “Nonlinear Control of Chemical Processes: A Review”, *Ind. Eng. Chem. Res.*, vol. 30, pp. 1391-1413, 1991.
- 11) Vipin Gopal and Lorenz Biegler, “Large Scale Inequality Constrained Optimisation and Control”, *IEEE Control Systems*, December, 1998.
- 12) K.P. Fruzzetti, A. Palazoglu and K.A. MacDonald, “Non-linear model predictive control using Hammerstein models”, *J.Proc. Cont.*, vol. 7, no 1, pp. 31-41, 1997.
- 13) Bryon R. Maner and Francis J. III Doyle, “Polymerization Reactor Control Using Autoregressive-Plus Volterra-Based MP”, *AICHE J.*, vol. 43, no 7, pp. 1763-1784, 1997.
- 14) B.R. Maner, Francis J. III Doyle, B.A. Ogunnaike and R.K. Pearson, “Nonlinear Model Predictive Control of a Simulated Multivariable Polymerization Reactor Using Second-order Volterra Models”, *Automatica*, vol. 32, no 9, pp. 1285-1301, 1996.
- 15) A. Banerjee and Y. Arkun, “Model predictive control of plant transitions using a new identification technique for interpolating non-linear models”, *J. Proc. Cont.*, vol. 8, no 5-6, pp. 441-457, 1998.
- 16) M.R. Arahal, M. Berenguel and E.F. Camacho, “Neural identification applied to predictive control of a solar plant”, *Control Eng. Practice*, vol. 6, pp. 333-344, 1998.
- 17) M. Fisher, Oliver Nelles and Rolf Isermann, “Adaptive predictive control of a heat exchanger based on a fuzzy model”, *Control Eng. Practice*, vol. 6, pp. 259-269, 1998.
- 18) R.K. Pearson, “Nonlinear Input/Output Modeling”, *Preprints IFAC Symposium ADCHEM '94*, Kyoto Research Park, Kyoto, Japan, 1994.
-