IDENTIFICATION OF QUASI-ARMAX MODELS OF NONLINEAR STOCHASTIC SAMPLED-DATA SYSTEMS

Bernt M. Åkesson [1] **and Hannu T. Toivonen**

Department of Chemical Engineering
Åbo Akademi University
FI-20500 Turku, Finland

Abstract: State-dependent parameter representations of nonlinear stochastic sampled-data systems are studied. Velocity-based linearization is used characterize sampled-data systems using nominally linear models whose parameters can be represented as functions of past outputs and inputs. For stochastic systems the approach leads to state-dependent ARMAX (quasi-ARMAX) representations. The models and their parameters are identified from input-output data using feedforward neural networks to represent the model parameters as functions of past inputs and outputs.

Keywords: neural network models, nonlinear models, sampled-data systems, stochastic systems, parameter identification

1. INTRODUCTION

A widely used approach for the black-box modelling and identification of nonlinear dynamical systems is to apply various nonlinear function approximators, such as artificial neural networks or fuzzy models. A shortcoming of these models is that they do not provide much insight into the systems dynamics. For this reason various model structures, which provide such information, have been introduced. One general class of models of this type consists of models with a nominally linear structure, but with state-dependent parameters (Priestley, 1988; Hu *et al.*, 2001; Young *et al.*, 2001). An important class of state-dependent parameter models consists of ARX models, in which the model parameters are nonlinear functions of past system outputs and inputs. These models have been called quasi-ARX (Hu *et al.*, 1998; Hu *et al.*, 2001; Previdi and Lovera, 2001) or state-dependent ARX models (Priestley, 1988; Young *et al.*, 2001). State-dependent parameter representations have the useful property that explicit information about the local dynamics is provided by the locally valid linear model, and in a number of situations they can be treated as linear systems whose parameters are taken as functions of scheduling variables.

For discrete-time systems, state-dependent parameter representations are usually approximative descriptions introduced for convenience. In contrast, continuous-time systems can be represented exactly by state-space models with state-dependent parameters constructed using velocity-based linearization (Leith and Leithead, 1998*b*; Leith and Leithead, 1998*a*). This fact can be applied to construct exact discrete-time state-dependent parameter representations for sampled-data systems (Toivonen, 2003). Quasi-ARX models of sampled-data systems are obtained by reconstructing the state of the state-dependent parameter representation in terms of past inputs and outputs (Toivonen, 2003).

In practice it is important to be able to deal with systems which are subject to stochastic noise. It is shown that a nonlinear sampled-data system subject

[1] Supported by the Finnish Graduate School in Chemical Engineering (GSCE)

to an additive drifting disturbance and measurement noise can be represented by a quasi-ARMAX model. By using a feedforward neural network to describe the model parameters as functions of past inputs and outputs (Hu and Hirasawa, 2002), the quasi-ARMAX model is represented with a type of recurrent network. Identification of neural network quasi-ARMAX models from input-output data is studied and illustrated with numerical examples.

2. STATE-DEPENDENT PARAMETER MODELS OF STOCHASTIC SYSTEMS

In a previous study (Toivonen, 2003), state-dependent parameter representations were derived for deterministic nonlinear sampled-data systems. In this paper, a generalization to stochastic systems is studied. Consider a nonlinear system

$$\begin{aligned} \dot{x}_P(t) &= f_P(x_P(t), u(t)) \\ y(t) &= h_P(x_P(t)) + w(t) \end{aligned} \tag{1}$$

where $x_P(t)$ denotes the state vector, $u(t)$ is the control input and $y(t)$ denotes the output. It is assumed that the nonlinear functions $f_P(\cdot,\cdot)$ and $h_P(\cdot)$ are continuous with Lipschitz continuous first derivatives. The system is subject to an additive drifting disturbance $w(t)$, which is described by a Wiener process with the incremental variance r_w.

The continuous-time input $u(t)$ to the nonlinear system is generated from a discrete-time input $u_d(k)$ by a zero-order hold and a strictly proper low-pass filter with the state-space representation (A_H, B_H, C_H). The system input is thus generated according to

$$\begin{aligned} \dot{x}_H(t) &= A_H x_H(t) + B_H u_d(k),\ t \in (kh, kh+h] \\ u(t) &= C_H x_H(t) \end{aligned} \tag{2}$$

where h denotes the sampling interval. The filter (2) generates a continuous input $u(t)$ to (1), which is differentiable for all t, except possibly at the sampling instants kh.

The sampled output is corrupted by measurement noise,

$$y_m(kh) = y(kh) + e_m(k) \tag{3}$$

which is described by a zero-mean white noise disturbance $\{e_m(k)\}$ with the variance σ_m^2.

Although a more realistic and complex disturbance model could be used, one reason for focusing on the model (3) is that it allows the construction of an exact quasi-ARMAX representation. It is therefore possible to compare identified models with the theoretically correct system description. It is also believed that the combination of a drifting disturbance and measurement noise provides a good approximation of more complex disturbances as well.

The generalized system consisting of the filter (2), the nonlinear system (1) and the disturbance model (3) can be described by a nonlinear system with a piecewise constant input,

$$\begin{aligned} \dot{x}(t) &= f(x(t)) + Bu_d(k),\ t \in (kh, kh+h] \\ y(t) &= h(x(t)) + w(t) \\ y_m(kh) &= y(kh) + e_m(k) \end{aligned} \tag{4}$$

where $x = [x_P^T, x_H^T]^T$ is the state of the generalized system, and

$$\begin{aligned} f(x) &= \begin{bmatrix} f_P(x_P, C_H x_H) \\ A_H x_H \end{bmatrix},\ B = \begin{bmatrix} 0 \\ B_H \end{bmatrix} \\ h(x) &= h_P(x_P) \end{aligned} \tag{5}$$

Differentiation of (4) with respect to time gives a nonlinear system with jumps,

$$\begin{aligned} \ddot{x}(t) &= A(x(t))\dot{x}(t),\ t \neq kh \\ x(kh^+) &= \dot{x}(kh) + B\Delta u_d(k) \\ dy(t) &= C(x(t))dx(t) + dw(t) \end{aligned} \tag{6}$$

where the notation $x(kh^+) = \lim_{\varepsilon\downarrow 0} x(kh+\varepsilon)$ has been used, $\Delta u_d(k) = u_d(k) - u_d(k-1)$ and

$$A(x) = \frac{\partial f(x)}{\partial x},\ C(x) = \frac{\partial h(x)}{\partial x} \tag{7}$$

Integration of (6) over the sampling intervals gives (Toivonen, 2003)

$$\begin{aligned} \dot{x}(kh+h) &= F(x(kh), u_d(k))\dot{x}(kh) \\ &\quad + G(x(kh), u_d(k))\Delta u_d(k) \\ \Delta y(kh+h) &= H(x(kh), u_d(k))\dot{x}(kh) \\ &\quad + J(x(kh), u_d(k))\Delta u_d(k) + e_w(k+1) \end{aligned} \tag{8}$$

where $\Delta y(kh+h) = y(kh+h) - y(kh)$ and $e_w(k+1) = w(kh+h) - w(kh)$. Hence $\{e_w(k)\}$ is a discrete-time white noise sequence with variance $\sigma_w^2 = r_w h$. The matrices $F(\cdot,\cdot)$, $G(\cdot,\cdot)$, $H(\cdot,\cdot)$ and $J(\cdot,\cdot)$ are smooth functions given by

$$F(x(kh), u_d(k)) = \Phi(kh+h) \tag{9}$$

$$H(x(kh), u_d(k)) = \Phi_y(kh+h) \tag{10}$$

and $G(\cdot,\cdot) = F(\cdot,\cdot)B$, $J(\cdot,\cdot) = H(\cdot,\cdot)B$, where $\Phi(\cdot)$ and $\Phi_y(\cdot)$ are defined by the differential equations

$$\frac{d\Phi(t)}{dt} = A(x(t))\Phi(t),\ \Phi(kh) = I \tag{11}$$

$$\frac{d\Phi_y(t)}{dt} = C(x(t))\Phi(t),\ \Phi_y(kh) = 0 \tag{12}$$

where $x(t)$ is given by (4).

The parameters of (8) are functions of the system state. In order to construct a model using input-output data only, it will be assumed that the state can be reconstructed from the inputs and outputs. For this purpose, some facts about the observability of nonlinear systems will be summarized (Levin and Narendra, 1995). The differential equation (4) defines a discrete-time system, which describes the propagation of the system state and output at the discrete sampling instants kh,

$$\begin{bmatrix} x(kh+h) \\ w(kh+h) \end{bmatrix} = \begin{bmatrix} f_d(x(kh), u_d(k)) \\ w(kh) \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} e_w(k+1) \qquad (13)$$
$$y(kh) = h(x(kh)) + w(kh)$$

where $f_d(\cdot,\cdot)$ is a smooth function. The discrete-time system (13) is state-invertible, i.e. $x(kh)$ is uniquely defined by $x(kh+h)$ and $u_d(k)$.

Observability makes it possible to determine the state from a finite number of system inputs and outputs. Unlike the linear case, observability of nonlinear systems cannot in general be guaranteed for all input sequences. Instead, a useful concept for nonlinear systems is generic observability (Aeyels, 1981; Levin and Narendra, 1995), meaning that the system is observable for almost every input sequence. Levin and Narendra (1995) have shown that if $h(\cdot)$ has isolated critical points, then the set of $f_d \in \mathscr{C}^\infty$ for which the system (13) is generically observable is open and dense in $\mathscr{C}^\infty$.

Assuming observability, there exist an integer l and continuous functions $g_m(\cdot)$, such that

$$x((k-l+m)h) = g_m(\varphi_l(k)),\ m = 1,\ldots,l \qquad (14)$$

for almost every input sequence, where

$$\varphi_l(k) = [y(kh),\ldots,y((k-l+1)h), u_d(k-1),\ldots u_d(k-l), e_w(k),\ldots,e_w(k-l)] \qquad (15)$$

For a system with a scalar output, observability ensures that the state can be reconstructed from $l = 2n_a+1$ past inputs and outputs, where n_a is the dimension of the augmented system (13) (Levin and Narendra, 1995). As n_a independent observations are sufficient to reconstruct the state, a smaller value of l can be expected to work well in many cases, however.

In order to obtain a state-dependent ARMAX representation of (8), the following assumptions are needed. Let $u_d \in \mathscr{U} \subset R^p$ and $x(kh) \in \mathscr{X} \subset R^n$, where $\mathscr{U}$ and $\mathscr{X}$ are open sets, and define the set

$$\mathscr{X}_f(y,u_d) = \{\dot{x} \in R^n \mid \dot{x} = f(x) + Bu_d, h(x) = y, x \in \mathscr{X}\} \qquad (16)$$

Assumption 1. The discrete-time system (13) is generically observable, such that (14) hold for almost every input sequence.

Assumption 2. The set defined by (16) is such that $\text{span}\{\mathscr{X}_f(y,u_d)\} = R^n$ for all $y \in h(\mathscr{X})$ holds for almost every $u_d \in \mathscr{U}$.

Using (14) to reconstruct the state in (8) gives the following result (cf. Toivonen (2003)).

Theorem 1. Suppose Assumptions 1 and 2 hold. Then the system with jumps described by (6) has the discrete-time representation

$$\Delta y_m(kh+h) = \sum_{i=1}^{l} A_i(\varphi_l(k))\Delta y_m((k-i+1)h) + \sum_{i=1}^{l+1} B_i(\varphi_l(k))\Delta u_d(k-i+1) + e(k+1) + \sum_{i=1}^{l+1} C_i(\varphi_l(k))e(k-i+1) \qquad (17)$$

where $e(k) = \Delta y_m(kh) - \Delta\hat{y}_m(kh|kh-h)$ is the minimum one-step prediction error. The moving average parameters $C_i(\varphi_l(k))$ are given by

$$C_i(\varphi_l(k)) = \begin{cases} c - A_1(\varphi_l(k)) \\ -cA_{i-1}(\varphi_l(k)) - A_i(\varphi_l(k)) \\ -cA_{l+1}(\varphi_l(k)) \end{cases} \qquad (18)$$

where

$$c = -\frac{\sigma_w^2 + 2\sigma_m^2}{2\sigma_m^2} + \sqrt{\left(\frac{\sigma_w^2 + 2\sigma_m^2}{2\sigma_m^2}\right)^2 - 1} \qquad (19)$$

where $\sigma_w^2 = Ee_w(k)^2$ and $\sigma_m^2 = Ee_m(k)^2$. Moreover, $\{e(k)\}$ is a zero-mean white noise process with the variance $Ee(k)^2 = -\sigma_m^2/c$.

The result can be proven in analogy with the deterministic case (Toivonen, 2003), and by using an innovations representation to model the stochastic noise.

The system representation (17) provides a theoretical justification for using a state-dependent parameter ARMAX (quasi-ARMAX) model structure to model nonlinear sampled-data systems. It is in general untractable to evaluate the functions in (14), which reconstruct the state, and hence the parameters $A_i(\varphi_l(k))$ and $B_i(\varphi_l(k))$ of (17), even for simple systems with known dynamics. In practical situations it is usually also unrealistic to assume that the whole state is known. These limitations lead to the problem of estimating the parameters of (17) directly from input-output data. For this purpose various function approximators, such as neural networks, can be applied.

3. SYSTEM IDENTIFICATION

In this section, identification of the quasi-ARMAX model (17) is considered. The model parameters are represented as functions of past inputs and outputs using feedforward neural networks (cf. Hu and Hirasawa, 2002). The representation of the model parameters is not a standard neural network approximation problem, because the approximated functions $A_i(\cdot)$, $B_i(\cdot)$ and $C_i(\cdot)$ are observed only indirectly via the system output y_m. However, by taking the model equation (17) as an additional output layer with time-varying weights $\Delta y(kh-ih)$, $\Delta u_d(k-i)$, $e(k-i)$, it is straightforward to use input-output data to train neural networks which approximate the quasi-ARMAX model parameters. A feedforward neural network representation of the quasi-ARMAX model is shown in Figure 1. The neural network output is given by

$$\begin{aligned}\Delta y_{NN}(kh+h) &= \sum_{i=1}^{n_A} A_i(k)\Delta y_m((k-i+1)h) \\ &+ \sum_{i=1}^{n_B} B_i(k)\Delta u_d(k-i+1) \\ &+ \sum_{i=1}^{n_C} C_i(k)\varepsilon(k-i+1) \qquad (20)\end{aligned}$$

where $\varepsilon(k) = \Delta y_m(kh) - \Delta y_{NN}(kh)$. The partial derivatives of (20) with respect to the network weights W can be evaluated according to

$$\begin{aligned}\frac{\partial \Delta y_{NN}(kh+h)}{\partial W} &= \sum_{i=1}^{n_A} \Delta y_m((k-i+1)h)\frac{\partial A_i(k)}{\partial W} \\ &+ \sum_{i=1}^{n_B} \Delta u_d(k-i+1)\frac{\partial B_i(k)}{\partial W} \\ &+ \sum_{i=1}^{n_C} \left(\varepsilon(k-i+1)\frac{\partial C_i(k)}{\partial W}\right. \\ &\left. - C_i(k)\frac{\partial \Delta y_{NN}((k-i+1)h)}{\partial W}\right) \qquad (21)\end{aligned}$$

where the derivatives $\partial A_i(k)/\partial W$, $\partial B_i(k)/\partial W$ and $\partial C_i(k)/\partial W$ of the outputs are given by standard formulae.

Observe that in the stochastic case the network in Figure 1 is a kind of a recurrent network, as the output $\Delta y_{NN}(kh+h)$ depends on past outputs via the output layer weights $\varepsilon(i)$ associated with the C-parameters. However, the training problem is simplified by the fact that the dependence on past outputs is linear. By taking the C-parameters as constants the complexity of the training problem can be reduced further.

4. EXPERIMENTAL RESULTS

In this section the modelling and system identification methods presented in this study are tested on a simulated bioreactor benchmark process (Ungar, 1990). The bioreactor consists of a continuous stirred tank reactor with a constant volume, containing cells and nutrients. The process dynamics are described by a nonlinear second-order system of the form (1), where

$$\begin{aligned}f_P(x_P,u) &= \begin{bmatrix} -x_{P,1}u+\xi \\ -x_{P,2}u+\xi\beta/(\beta-x_{P,2}) \end{bmatrix} \\ h_P(x_P) &= x_{P,1} \qquad (22)\end{aligned}$$

where $\xi = x_{P,1}(1-x_{P,2})\exp(x_{P,2}/\gamma)$. Here $x_{P,1}$ and $x_{P,2}$ are dimensionless cell mass and substrate conversion, respectively, and u is the flow rate through the reactor. The variables are chosen to lie in the interval $[0,1]$. The parameter values $\beta = 1.02$ and $\gamma = 0.48$ are used.

The continuous-time input u is obtained from the discrete-time input u_d by passing it through a zero-order hold followed by a low-pass filter. The low-pass filter (2) had the parameters $A_H = -100$, $B_H = 100$ and $C_H = 1$. The sampling time $h = 0.5$ suggested by Ungar (1990) was used.

The system is augmented with a disturbance model according to (1) and (3), consisting of an additive drifting process noise $w(t)$ with incremental variance $r_w = 2\times 10^{-7}$ and measurement noise with variance $\sigma_m^2 = 10^{-4}$.

By Theorem 1, the system can be represented by a quasi-ARMAX model (17). The theoretical values of the model parameters can be calculated (cf. Toivonen, 2003) and used for comparison with the identified models. Using the approach in (Toivonen, 2003), the bioreactor can be represented by a quasi-ARMAX model with two A-parameters, three B-parameters and three C-parameters, which are functions of two past outputs and three past inputs. As the parameter $B_3(k)$ is very close to zero ($|B_3(k)| < 0.0017|B_2(k)|$) it is not estimated in the model. Thus, quasi-ARMAX models with two A-parameters and two B-parameters were identified.

Two sets of 2500 input-output pairs each were generated; one was used for training and the other for testing. The minimum one-step prediction error variance of the theoretical model is 0.000103. The optimal predictor gives the prediction error variances 0.000114 and 0.000104 on training data and test data, respectively.

In order to study the effect of the number of C-parameters, models with various numbers of constant C-parameters were identified. The results are shown in Table 1. Using four hidden layer neurons, the total number of network weights ranges from 44 to 47. The results can be compared to the optimal prediction error variances, cf. above. It is seen that the prediction error is the smallest with three C-parameters, although two parameters give quite similar results, in terms of the test error. Due to the incremental form of the quasi-ARMAX model structure, at least a second-order noise model is required for a satisfactory modelling of the noise dynamics. In particular, using a model with one C-parameter the noise model tends to become unstable, as the parameter value is approximately equal to one.

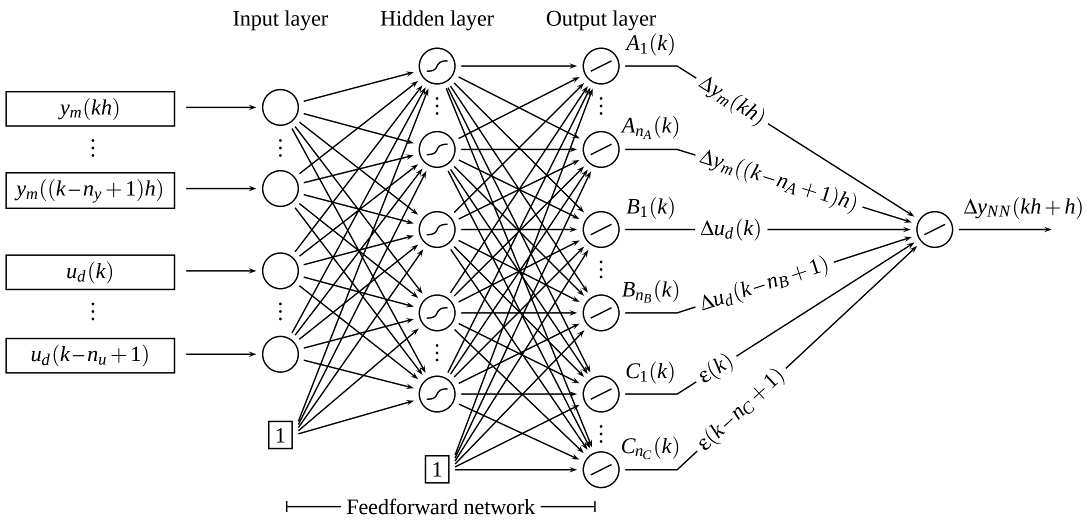

Fig. 1. Structure of the quasi-ARMAX model (20), whose parameters are represented by a feedforward neural network.

Table 1. Prediction errors for quasi-ARMAX models with different numbers of constant C-parameters.

n_C	Prediction error variance	
	Training data	Test data
0	0.000170	0.000166
1	0.000140	∞
2	0.000129	0.000130
3	0.000119	0.000124

Models with time-varying C-parameters were also identified. Using three C-parameters and a network with three hidden layer neurons corresponds to 46 weights. The result was similar to the case with constant C-parameters, giving the prediction error variance 0.000118 on the training data and 0.000124 on the test data.

The predictions and the measurements of the neural network quasi-ARMAX model with three constant C-parameters are shown in Figure 2. Figure 3 shows the approximated and the theoretical parameters. The constant C-parameters correspond well with the average values of the theoretical ones.

For comparison, a neural network ARMAX (NNARMAX) model (Nørgaard *et al.*, 2000) was identified. The model consists of a feedforward neural network and a linear noise filter,

$$y_{NN}(kh+h) = g_{NN}(\varphi_{yu}(k)) + \sum_{i=1}^{n_C} C_i \varepsilon_{NN}(k-i+1) \qquad (23)$$

where $g_{NN}(\cdot)$ is a feedforward network with input vector $\varphi_{yu}(k) = [y_m(kh), \ldots, y_m((k-n_y+1)h), u_d(k), \ldots, u_d(k-n_u+1)]$ and

$$\varepsilon_{NN}(k+1) = y_m(kh+h) - y_{NN}(kh+h)$$

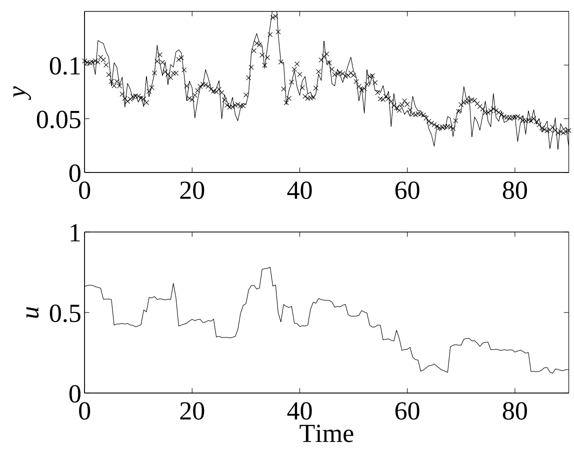

Fig. 2. Prediction results with a quasi-ARMAX model with three C-parameters. The upper graph shows the measurements (solid lines) and the one-step ahead predictions (crosses). The lower graph shows the system input.

The network is chosen to have the same inputs as the quasi-ARMAX model and six hidden layer neurons. An NNARX model and an NNARMAX model with three C-parameters were identified. The NNARX model had a total of 43 weights and the NNARMAX model had 46 weights. The results are summarized in Table 2. It is seen that quasi-ARMAX model gives smaller prediction error variances provided a sufficiently complex noise model is used.

Table 2. Comparison of different models.

Model	n_C	Prediction error variance	
		Training data	Test data
Quasi-ARX	0	0.000170	0.000166
Quasi-ARMAX	3	0.000119	0.000124
NNARX	0	0.000145	0.000156
NNARMAX	3	0.000142	0.000142

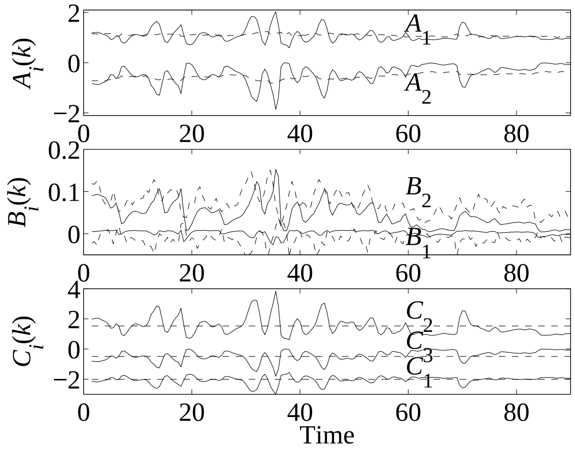

Fig. 3. Quasi-ARMAX model parameters corresponding to the simulation in Figure 2. Dashed lines represent the neural network approximations and solid lines the theoretically correct parameter values.

5. DISCUSSION AND CONCLUSIONS

Identification of quasi-ARMAX models of nonlinear stochastic sampled-data systems was presented. The quasi-ARMAX structure can be be derived by velocity-based linearization and reconstruction of the state in terms of past outputs and inputs. The parameters were approximated by a feedforward neural network. The quasi-ARMAX model can be seen as a recurrent network, which can be trained on input-output data.

A bioreactor was used as a numerical example. The system was subject to an additive drifting disturbance and measurement noise. The correct model parameters were calculated for comparison with the approximations. Identification from input-output data gave an accurate model in terms of the one-step prediction error. The quasi-ARMAX model was also more accurate than an NNARMAX model with approximately the same structure and complexity. This result is in accordance with the results of Peng *et al.* (2003), who studied the deterministic case using radial basis functions for model approximation.

The quasi-ARMAX model structure provides a convenient way of describing nonlinear stochastic sampled-data systems. In particular, the nominally linear structure makes it suitable for prediction and control applications, and the state-dependent parameters provide information about the system dynamics.

REFERENCES

Aeyels, D. (1981). Generic observability of differentiable systems. *SIAM Journal on Control and Optimization* **19**(5), 595–603.

Hu, J. and K. Hirasawa (2002). A method for applying multilayer perceptrons to control of nonlinear systems. In: *Proceedings of the 9th International Conference on Neural Information Processing*. Singapore. pp. 1267–1271.

Hu, J., K. Kumamaru and K. Hirasawa (2001). A quasi-ARMAX approach to modelling of nonlinear systems. *International Journal of Control* **74**(18), 1754–1766.

Hu, J., K. Kumamaru, K. Inoue and K. Hirasawa (1998). A hybrid quasi-ARMAX modeling scheme for identification of nonlinear systems. *Transactions of the Society of Instrument and Control Engineering* **34**(8), 977–985.

Leith, D. J. and W. E. Leithead (1998*a*). Gain-scheduled and nonlinear systems: dynamic analysis by velocity-based linearization families. *International Journal of Control* **70**(2), 289–317.

Leith, D. J. and W. E. Leithead (1998*b*). Gain-scheduled controller design: an analytic framework directly incorporating non-equilibrium plant dynamics. *International Journal of Control* **70**(2), 249–269.

Levin, A. U. and K. S. Narendra (1995). Recursive identification using feedforward neural networks. *International Journal of Control* **61**(3), 533–547.

Nørgaard, M., O. Ravn, N. K. Poulsen and L. K. Hansen (2000). *Neural Networks for Modelling and Control of Dynamic Systems*. Springer-Verlag. London.

Peng, H., T. Ozaki, V. Haggan-Ozaki and Y. Toyoda (2003). A parameter optimization method for radial basis function type models. *IEEE Transactions on Neural Networks* **14**(2), 432–438.

Previdi, F. and M. Lovera (2001). Identification of a class of nonlinear parametrically varying models. In: *Proceedings of the European Control Conference*. Porto, Portugal. pp. 3086–3091.

Priestley, M. B. (1988). *Non-linear and Non-stationary Time Series Analysis*. Academic Press. London.

Toivonen, H. T. (2003). State-dependent parameter models of non-linear sampled-data systems: a velocity-based linearization approach. *International Journal of Control* **76**(18), 1823–1832.

Ungar, L. H. (1990). A bioreactor benchmark for adaptive network-based process control. In: *Neural Networks for Control* (W. T. Miller, III, R. S. Sutton and P. I. Werbos, Eds.). Chap. 16, Appendix A, pp. 387–402, 476–480. The MIT Press. Cambridge, MA.

Young, P. C., P. McKenna and J. Bruun (2001). Identification of non-linear stochastic systems by state dependent parameter estimation. *International Journal of Control* **74**(18), 1837–1857.