Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Process Control 16 (2006) 877–886

JOURNAL OF PROCESS CONTROL

www.elsevier.com/locate/jprocont

State-dependent parameter modelling and identification of stochastic non-linear sampled-data systems

Bernt M. Åkesson, Hannu T. Toivonen *

Department of Chemical Engineering, Åbo Akademi University, FIN-20500 Åbo, Finland

Received 15 December 2004; received in revised form 8 September 2005; accepted 1 February 2006

Abstract

State-dependent parameter representations of stochastic non-linear sampled-data systems are studied. Velocity-based linearization is used to construct state-dependent parameter models which have a nominally linear structure but whose parameters can be characterized as functions of past outputs and inputs. For stochastic systems state-dependent parameter ARMAX (quasi-ARMAX) representations are obtained. The models are identified from input–output data using feedforward neural networks to represent the model parameters as functions of past inputs and outputs. Simulated examples are presented to illustrate the usefulness of the proposed approach for the modelling and identification of non-linear stochastic sampled-data systems.

Keywords: Sampled-data systems; Neural network models; Stochastic systems

1. Introduction

A widely used approach in black-box modelling and identification of non-linear dynamical systems is to apply various non-linear function approximators, such as artificial neural networks or fuzzy models, to describe the system output as a function of past inputs and outputs. This approach is based on the fact that under mild conditions, the output of a dynamical system is a function of a fixed number of past inputs and outputs, cf., the Embedding Theorem of Takens [17], stated originally for autonomous systems and generalized to forced and stochastic systems by Stark et al. [15,16]. In the control literature, Levin and Narendra [9] have given observability conditions under which the output of a non-linear discrete-time system is a function of past inputs and outputs. Leontaritis and Billings [8] generalized autoregressive moving average models with exogenous inputs (ARMAX models) to non-linear ARMAX models, where the output of the non-linear system is taken as a function of past inputs and outputs as well as past prediction errors.

A shortcoming of black-box models based on general function approximators is that they do not provide much insight into the system 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 [14,5,21,22]. An important class of models of this form consists of ARX models, in which the model parameters are non-linear functions of past system outputs and inputs. These models have been called quasi-ARX [4,5,13] or state-dependent ARX models [14,22]. 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. It is straightforward to adapt state-dependent models to the stochastic case by extending the quasi-ARX model structure with a moving average

* Corresponding author. Tel.: +358 2 2154451; fax: +358 2 2154479.
E-mail addresses: bakesson@abo.fi (B.M. Åkesson), htoivone@abo.fi (H.T. Toivonen).

0959-1524/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jprocont.2006.02.002

noise term. The quasi-ARMAX model structure obtained in this way has been found useful in the modelling of stochastic systems [4,5].

For discrete-time systems, state-dependent parameter representations are usually approximative descriptions introduced for the sake of convenience. In contrast, continuous-time systems can be represented exactly by state-space models with state-dependent parameters constructed using velocity-form linearization [6,7]. This fact can be applied to represent sampled-data systems exactly by discrete-time state-space models with state-dependent parameters [18]. 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 [18].

In this paper, the velocity-form linearization approach is applied to construct state-dependent parameter representations for stochastic non-linear sampled-data systems. It is shown that a finite-dimensional sampled-data system subject to an additive drifting disturbance and measurement noise can be represented by a quasi-ARMAX model. However, in contrast to the deterministic case, the model parameters cannot be described exactly as functions of past inputs and outputs only, as they are also functions of the unknown disturbances.

We will also consider the identification of state-dependent parameter ARX and ARMAX models from input–output data for both deterministic and stochastic systems. A feedforward neural network approximator is used to describe the model parameters as functions of past inputs and outputs, cf., [3]. The neural network is trained on input–output data, without knowledge of the true parameter values. For stochastic systems, two identification approaches are studied. By describing the parameters of the quasi-ARMAX representation as functions of past inputs and outputs a recurrent network structure is obtained, in which the output depends on past prediction errors via the moving average terms. In this approach the achievable accuracy of the parameter approximation is limited due to the fact that the outputs are corrupted by measurement noise. In order to obtain more accurate parameter estimates, we also study an approach in which the parameters are represented as functions of noise-free system outputs, which are estimated using extended Kalman filter techniques.

The paper is organized as follows. In Section 2, state-dependent parameter and quasi-ARMAX models of a class of stochastic sampled-data systems are derived. The modelling and identification of the models using neural network approximators is studied in Section 3. In Section 4, the model structures and identification methods are illustrated by numerical examples.

2. State-dependent parameter models of stochastic sampled-data systems

2.1. State-space representations

In this section the state-dependent parameter representation of deterministic sampled-data systems [18] is generalized to systems which are subject to stochastic disturbances. We consider the stochastic sampled-data system depicted in Fig. 1. The continuous-time control input $u(t)$ to the non-linear system $\mathscr{P}$ is generated from the discrete-time control signal $u_d(k)$ by a zero-order hold mechanism followed by a linear low-pass filter $\mathscr{H}$. The discrete-time output $y(kh)$ is obtained by sampling the system output $y(t)$ using the sampling time h. The system is subject to a process disturbance $w(t)$ and a disturbance $v(t)$ affecting the output. There is also a discrete-time measurement noise $e_m(k)$. The generalized system consisting of the filter $\mathscr{H}$ and the non-linear system $\mathscr{P}$ is described by

$$\begin{aligned} \dot{x}(t) &= f(x(t)) + Bu_d(k) + Ew(t), \quad t \in (kh, kh+h] \\ y(t) &= h(x(t)) + v(t) \\ y_m(kh) &= y(kh) + e_m(k) \end{aligned} \tag{1}$$

Notice that as the filter $\mathscr{H}$ is included in the system equation, the input $u_d(k)$ enters linearly if $\mathscr{H}$ is strictly proper. In a similar way, the assumption that the disturbance $w(t)$ enters linearly is not very restrictive as the system can be assumed to include the noise dynamics.

State-dependent parameter models of the stochastic system (1) can be constructed using velocity-based linearization, cf., [6,7,18]. However, the velocity-form linearization procedure is applicable only if all input signals are differentiable with respect to time. This implies in particular that the continuous-time disturbances cannot be modelled as white noise. Here it is assumed that the disturbances are drifting processes. The signal $w(t)$ is taken as a vector-valued Wiener process with unit incremental covariance matrix, $v(t)$ is a Wiener process with incremental variance r_v, and $\{e_m(k)\}$ is zero-mean discrete-time white noise with the variance σ_m^2. It is assumed that any additional disturbance dynamics are captured in $f(\cdot)$ and the state

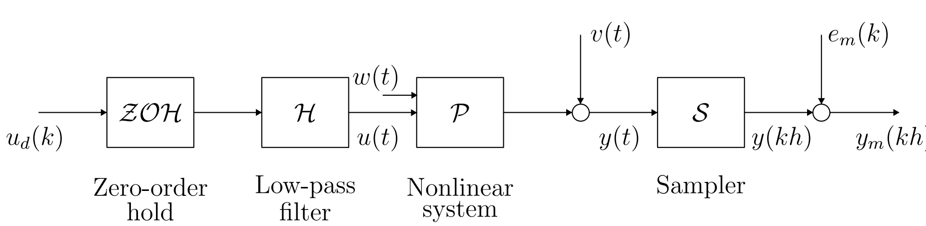

Fig. 1. Stochastic sampled-data system.

vector x. The modelling of the noise as a drifting disturbance is relevant in many control problems where the system is subject to slowly varying random disturbances or unknown offsets. It is also consistent with the linearized model representations studied here, which describe the relations between the input and output increments, rather than their absolute values.

In velocity-based linearization [6,7,18], the differential of (1) is formed, resulting in the non-linear stochastic system with jumps,

$$\begin{aligned} \mathrm{d}\dot{x}(t) &= A(x)\dot{x}(t)\,\mathrm{d}t + E\,\mathrm{d}w(t), \quad t \neq kh \\ \dot{x}(kh^+) &= \dot{x}(kh) + B\Delta u_d(k) \\ \mathrm{d}y(t) &= C(x)\dot{x}(t)\,\mathrm{d}t + \mathrm{d}v(t) \end{aligned} \tag{2}$$

where $x(kh^+) = \lim_{\epsilon\downarrow 0} x(kh+\epsilon)$,

$$\Delta u_d(k) = u_d(k) - u_d(k-1) \tag{3}$$

and

$$A(x) = \frac{\partial f(x)}{\partial x}, \quad C(x) = \frac{\partial h(x)}{\partial x} \tag{4}$$

Introducing the matrix functions $\Phi(t;w)$ and $\Phi_y(t;w)$ defined by

$$\frac{\partial \Phi(t;w)}{\partial t} = A(x(t))\Phi(t;w), \quad \Phi(kh^+;w) = I \tag{5}$$

$$\frac{\partial \Phi_y(t;w)}{\partial t} = C(x(t))\Phi(t;w), \quad \Phi_y(kh^+;w) = 0 \tag{6}$$

and integrating (2) from $t = kh^+$ to $t = s$ gives

$$\begin{aligned} \dot{x}(s) &= \Phi(s;w)\dot{x}(kh^+) + \Phi(s;w)\int_{kh^+}^{s} \Phi(\tau;w)^{-1}E\,\mathrm{d}w(\tau) \\ y(s) &= y(kh^+) + \Phi_y(s;w)\dot{x}(kh^+) + \int_{kh^+}^{s} \left[\Phi_y(s;w) - \Phi_y(\tau;w)\right] \\ &\quad \times \Phi(\tau;w)^{-1}E\,\mathrm{d}w(\tau) + v(s) - v(kh) \end{aligned} \tag{7}$$

It follows that the sampled-data system can be described by the discrete-time stochastic model:

$$\begin{aligned} \dot{x}(kh+h) &= F(\theta(k))\dot{x}(kh) + G(\theta(k))\Delta u_d(k) + w_d(k) \\ \Delta y(kh+h) &= H(\theta(k))\dot{x}(kh) + J(\theta(k))\Delta u_d(k) + v_d(k) \\ y_m(kh) &= y(kh) + e_m(k) \end{aligned} \tag{8}$$

where $\Delta y(kh+h) = y(kh+h) - y(kh)$,

$$F(\theta(k)) = \Phi(kh+h;w), \quad G(\theta(k)) = F(\theta_e(k))B \tag{9}$$

$$H(\theta(k)) = \Phi_y(kh+h;w), \quad J(\theta(k)) = \Phi_y(kh+h;w)B \tag{10}$$

and $\theta(k)$ denotes the information required to determine the propagation of the system in the interval $[kh, kh+h)$, i.e., $\theta(k) = (x(kh), u_d(k), w(t), t \in [kh, kh+h))$. The signals w_d and v_d are discrete-time white-noise disturbances given by

$$w_d(k) = \Phi(kh+h;w)\int_{kh^+}^{kh+h} \Phi(\tau;w)^{-1}E\,\mathrm{d}w(\tau) \tag{11}$$

$$\begin{aligned} v_d(k) &= \int_{kh^+}^{kh+h} \left[\Phi_y(kh+h;w) - \Phi_y(\tau;w)\right]\Phi(\tau;w)^{-1}E\,\mathrm{d}w(\tau) \\ &\quad + v(kh+h) - v(kh) \end{aligned} \tag{12}$$

The model (8) gives a state-dependent parameter state-space representation of the stochastic sampled-data system (1), and it can be regarded as a generalization of the deterministic case studied in [18].

2.2. State-dependent parameter ARMAX representations

In analogy with the quasi-ARX representation obtained in the deterministic case [18], stochastic sampled-data systems can be described by state-dependent parameter ARMAX models. However, in contrast to the deterministic case, the parameters of the input–output model cannot be represented as functions of the control input and the measured output only, but they are also functions of the stochastic disturbances.

A state-dependent parameter ARMAX representation of (8) can be constructed as follows. The state of (8) can be estimated by the extended Kalman filter

$$\begin{aligned} \begin{bmatrix} \hat{\dot{x}}(kh+h) \\ \hat{e}_m(k+1) \end{bmatrix} &= \begin{bmatrix} F(\hat{\theta}(k)) & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{\dot{x}}(kh) \\ \hat{e}_m(k) \end{bmatrix} \\ &\quad + \begin{bmatrix} G(\hat{\theta}(k)) \\ 0 \end{bmatrix}\Delta u_d(k) + \begin{bmatrix} K_1(k) \\ K_2(k) \end{bmatrix} e(k+1) \\ \Delta\hat{y}(kh+h) &= \begin{bmatrix} H(\hat{\theta}(k)) & -I \end{bmatrix} \begin{bmatrix} \hat{\dot{x}}(kh) \\ \hat{e}_m(k) \end{bmatrix} + J(\hat{\theta}(k))\Delta u_d(k) \\ \Delta y_m(kh+h) &= \Delta\hat{y}(kh+h) + e(k+1) \end{aligned} \tag{13}$$

where $\hat{\theta}(k) = \theta(k)$ with $x(kh) = \hat{x}(kh)$ and $w(t) = 0$, $t \in [kh, kh+h)$, or equivalently, $\hat{\theta}(k) = (\hat{x}(kh), u_d(k))$, and $K_1(k)$ and $K_2(k)$ are the extended Kalman filter gains. In analogy with the deterministic case [18], reconstruction of the state in terms of past inputs and outputs gives the quasi-ARMAX representation

$$\begin{aligned} \text{‘}\Delta y_m(kh+h) &= \sum_{i=1}^{l} A_i(k)\Delta y_m((k-i+1)h) \\ &\quad + \sum_{i=1}^{l+1} B_i(k)\Delta u_d(k-i+1) + e(k+1) \\ &\quad + \sum_{i=1}^{l+1} C_i(k)e(k-i+l) \end{aligned} \tag{14}$$

where $e(k) = \Delta y_m(kh) - \Delta\hat{y}(kh)$ is the minimum one-step prediction error.

The representation (14) associated with the extended Kalman filter (13) is an incremental form of the quasi-ARMAX models studied in [4,5], and it provides a theoretical justification of the quasi-ARMAX model structure for non-linear sampled-data systems. The system representation can be considered as a state-dependent parameter ARMAX version of the general non-linear ARMAX representation of non-linear stochastic systems [8,11].

The parameters of (14) are functions of the estimated state. It is, however, very hard to determine the system parameters even for known systems. Therefore, we will also study a special case, where it is feasible to calculate the model parameters theoretically. The analysis of the state-dependent ARMAX model can be simplified significantly if (1) is affected by an additive disturbance at the output only, i.e., $w = 0$. The representation (8) then simplifies to

$$\begin{aligned} \dot{x}(kh+h) &= F(x(kh), u_d(k))\dot{x}(kh) + G(x(kh), u_d(k))\Delta u_d(k) \\ \Delta y(kh+h) &= H(x(kh), u_d(k))\dot{x}(kh) \\ &\quad + J(x(kh), u_d(k))\Delta u_d(k) + e_v(k+1) \\ y_m(kh) &= y(kh) + e_m(k) \end{aligned} \tag{15}$$

where $e_v(k+1) = v(kh+h) - v(kh)$ is discrete-time white noise with variance $\sigma_v^2 = r_v h$. As the system variable $y(t)$ is affected by the unmeasured drifting disturbance $v(t)$, an accurate prediction of y (or y_m) is not possible without making use of the measured output y_m. It is therefore natural to describe the system by a state-dependent parameter prediction error model. Such a model can be obtained by expressing the state of (15) in terms of the inputs and outputs, giving an input–output model of the form

$$\begin{aligned} \Delta y_m(kh+h) &= A_1(k)\Delta y_m(kh) + \cdots + A_l(k)\Delta y_m((k-l+1)h) \\ &\quad + B_1(k)\Delta u_d(k) + \cdots + B_{l+1}(k)\Delta u_d(k-l) \\ &\quad + n(k+1) \end{aligned} \tag{16}$$

where

$$n(k+1) = -\sum_{i=1}^{l} A_i(k) n_e(k+1-i) + n_e(k+1) \tag{17}$$

where

$$n_e(k) = e_m(k) - e_m(k-1) + e_v(k) \tag{18}$$

By (15), the parameters of (16) are functions of the system state. However, in contrast to the deterministic case, perfect reconstruction of the state from a finite number of past inputs and measured outputs is not possible, since the system is corrupted by noise.

In order to see what is possible, observe that with $w = 0$, the propagation of the system defined by the differential Eq. (1) at the sampling instants is described by a discrete-time system,

$$\begin{aligned} x(kh+h) &= f_d(x(kh), u_d(k)) \\ y(kh) &= h(x(kh)) + v(kh) \end{aligned} \tag{19}$$

Introducing the dynamics of the discrete-time drifting process $\{v(kh)\}$ gives

$$\begin{aligned} \begin{bmatrix} x(kh+h) \\ v(kh+h) \end{bmatrix} &= \begin{bmatrix} f_d(x(kh), u_d(k)) \\ v(kh) \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} e_v(k) \\ y(kh) &= h(x(kh)) + v(kh) \end{aligned} \tag{20}$$

Assuming generic observability [1,9] of (20), the state $x(kh)$ and $v(kh)$ can be reconstructed for almost every input sequence from a finite number of inputs and outputs,

$$\begin{aligned} \varphi_l(k) = [&y(kh), \ldots, y(kh-lh+h), u_d(k), \ldots, u_d(k-l), \\ &e_v(k), \ldots, e_v(k-l)] \end{aligned} \tag{21}$$

Following the deterministic case, we can now state the following result.

Theorem 2.1. *Consider the system (19). Assume that the system is generically observable. Let $u_d(k) \in \mathscr{U} \subset R$ and $x(kh) \in \mathscr{X} \subset R^n$, where $\mathscr{U}$ and $\mathscr{X}$ are open sets. Assume that the set*

$$\mathscr{X}_f(y, u_d) = \{\dot{x} \in R^n | \dot{x} = f(x) + Bu_d, h(x) = y, x \in \mathscr{X}\} \tag{22}$$

is such that $\operatorname{span}\{\mathscr{X}_f(y, u_d)\} = R^n$ for all $y \in h(\mathscr{X})$ holds for almost every $u_d \in \mathscr{U}$. Then the associated stochastic system (15) has the representation (16) where the parameters are functions of $\varphi_l(k)$. Moreover, if (16) is stable, the system has the state-dependent parameter ARMAX representation

$$\begin{aligned} \Delta y_m(kh+h) &= A_1(\varphi_l(k))\Delta y_m(kh) + \cdots \\ &\quad + A_l(\varphi_l(k))\Delta y_m((k-l+1)h) \\ &\quad + B_1(\varphi_l(k))\Delta u_d(k) + \cdots \\ &\quad + B_{l+1}(\varphi_l(k))\Delta u_d(k-l) + e(k+1) \\ &\quad + C_1(\varphi_l(k))e(k) + \cdots \\ &\quad + C_{l+1}(\varphi_l(k))e(k-l) \end{aligned} \tag{23}$$

where $e(k)$ is the minimum one-step prediction error. The parameters $C_i(\varphi_l(k))$ are given by

$$C_i(\varphi_l(k)) = \begin{cases} c - A_1(\varphi_l(k)), & i = 1 \\ -cA_{i-1}(\varphi_l(k)) - A_i(\varphi_l(k)), & i = 2, \ldots, l \\ -cA_i(\varphi_l(k)), & i = l+1 \end{cases} \tag{24}$$

where

$$c = -\frac{\sigma_v^2 + 2\sigma_m^2}{2\sigma_m^2} + \sqrt{\left(\frac{\sigma_v^2 + 2\sigma_m^2}{2\sigma_m^2}\right)^2 - 1} \tag{25}$$

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

$$Ee(k)^2 = -\frac{\sigma_m^2}{c} \tag{26}$$

Proof. The representation (16) follows from (15), the observability assumption and the assumption on the set (22) [18]. By observability, the parameters of (16) are functions of $\varphi_l(k)$. By (16), the minimum one-step prediction error $e(k+1) = \Delta y_m(kh+h) - \Delta\hat{y}_m(kh+h|kh)$ is also the minimum one-step prediction error of the disturbance $n(k)$, i.e., $e(k+1) = n(k+1) - \hat{n}(k+1|k)$. By (17) we also have $e(k+1) = n_e(k+1) - \hat{n}_e(k+1|k)$, where $n_e(k)$ is the moving average stochastic process defined by (18). By constructing a Kalman filter for the signal $n_e(k)$, it can be represented in terms of the prediction error $e(k)$ as

$$n_e(k) = e(k) + ce(k-1) \tag{27}$$

where c is given by (25), and the minimum prediction error $e(k)$ has the variance (26). Introducing (27) into (17) and

(16) gives (23) and (24). The stability of (16) ensures that the minimum prediction error $e(k)$ can be causally calculated from the system Eq. (23). □

Theorem 2.1 implies that the model (15) allows an exact quasi-ARMAX representation, similar to the quasi-ARX model obtained for deterministic systems. It is therefore possible to compare identified model parameters with the theoretically correct system description in Theorem 2.1. It is also believed that the combination of an output additive drifting disturbance and measurement noise provides a good approximation of more complex disturbances as well.

3. System identification

As discussed in Section 2, it is in practice untractable to evaluate the mappings which define the parameters of the state-dependent ARX and ARMAX models as functions of past inputs and outputs. Therefore it is necessary to represent the model parameters using a function approximator. In this study, a feedforward neural network is used to identify the state-dependent parameter models. Networks with one hidden layer with hyperbolic tangent activation functions will be considered. It is well known that a network of this type is able to approximate any continuous non-linear function to arbitrary accuracy [2].

In the deterministic case, neural networks are used to identify quasi-ARX models obtained when the disturbance is zero. For stochastic systems, we consider both the quasi-ARMAX model (23) and a simplified form of the state-dependent model structure (8), which allows the estimation of the process output y using an extended Kalman filter.

3.1. Identification of state-dependent ARX and ARMAX models

In this section, we consider the identification of the quasi-ARMAX model (23) from input–output data. The model parameters are represented as functions of past inputs and outputs using feedfoward neural networks, cf., [3]. The representation of the model parameters is not a standard neural network approximation problem, because the approximated functions $A_i(\cdot)$, $B_i(\cdot)$, $C_i(\cdot)$ are observed only indirectly via the system output y_m. However, by taking the model Eq. (23) as an additional output layer with time-varying weights $\Delta y(kh - ih)$, $\Delta u_d(k-i)$, $e(k-i)$ as shown in Fig. 2, it is straightforward to use input–output data to train a neural network which approximates the quasi-ARMAX model parameters. The neural network output is given by

$$\begin{aligned}\Delta y_{NN}(kh+h) = {} & A_1(k)\Delta y_m(kh) + \cdots + A_{n_A}(k) \\ & \times \Delta y_m(kh-(n_A-1)h) + B_1(k)\Delta u_d(k) + \cdots \\ & + B_{n_B}(k)\Delta u_d(k-n_B+1) + C_1(k)\epsilon(k) + \cdots \\ & + C_{n_C}(k)\epsilon(k-n_C+1) \qquad (28)\end{aligned}$$

where $\epsilon(k) = \Delta y_m(kh) - \Delta y_{NN}(kh)$. The system output can be predicted using the quasi-ARMAX neural network model according to $\hat{y}(kh+h) = y_m(kh) + \Delta y_{NN}(kh+h)$.

The derivatives of the network output with respect to the weights W are given by

$$\begin{aligned}\frac{\partial \Delta y_{NN}(kh+h)}{\partial W} = {} & \sum_{i=0}^{n_A-1} \Delta y_m(kh-ih)\frac{\partial A_{i+1}(k)}{\partial W} \\ & + \sum_{i=0}^{n_B-1} \Delta u_d(k-i)\frac{\partial B_{i+1}(k)}{\partial W} \\ & + \sum_{i=0}^{n_C-1}\Bigg(\epsilon(k-i)\frac{\partial C_{i+1}(k)}{\partial W} \\ & - C_{i+1}(k)\frac{\partial \Delta y_{NN}(kh-ih)}{\partial W}\Bigg)\end{aligned}$$

where the derivatives $\partial A_{i+1}(k)/\partial W$, $\partial B_{i+1}(k)/\partial W$ and $\partial C_{i+1}(k)/\partial W$ of the hidden layer outputs are given by standard formulae [2].

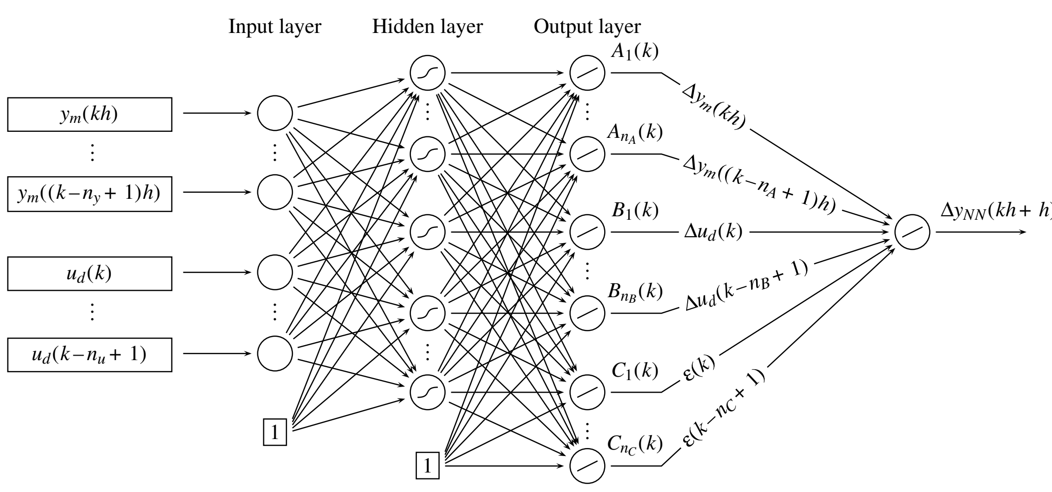

Fig. 2. Feedforward neural network for quasi-ARMAX model approximation.

Observe that in the stochastic case the network in Fig. 2 is a kind of recurrent network, as the output $\Delta y_{NN}(kh+h)$ depends of past outputs via the output layer weights $\epsilon(i)$ 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.

3.2. An extended Kalman filter approach

In the quasi-ARMAX neural network approach discussed above the model parameters are taken as functions of past inputs $u_d(k-i)$ and measured outputs $y_m(kh-ih)$. In the presence of measurement noise, more accurate parameter estimates would, however, be obtained by using the measurement noise-free system output y, cf., (1). In order to recover the system output y from the measured output, observe that in analogy with Section 2, the state of the state-dependent parameter representation (8) can be reconstructed in terms of past system outputs y and inputs as

$$\begin{aligned}\Delta y(kh+h) &= \sum_{i=1}^{l} A_i(k)\Delta y((k-i+1)h) \\ &\quad + \sum_{i=1}^{l+1} B_i(k)\Delta u_d(k-i+1) + e_{\rm w}(k+1) \\ y_m(kh) &= y(kh) + e_m(k)\end{aligned} \tag{29}$$

where the disturbance $e_{\rm w}(k+1)$ can be expressed in the form

$$\begin{aligned}e_{\rm w}(k+1) &= v_d(k+1) + \sum_{i=1}^{l} D_i(k)v_d(k-i+1) \\ &\quad + \sum_{i=1}^{l+1} E_i(k)w_d(k-i+l)\end{aligned} \tag{30}$$

and the parameters are functions of past inputs $u_d(k),\ldots,u_d(k-l)$, outputs $y(kh),\ldots,y((k-l+1)h)$, and disturbances, $w(t)$, $v(t)$, $t\in[(k-l)h,(k+1)h)$.

As before, we approximate the state-dependent parameter representation (29) by a neural network model, in which the model parameters are taken as functions of past inputs u_d and noise-free process outputs y. The state-dependent parameter neural network model has the state-space representation

$$\begin{aligned}y(kh+h) &= y(kh) + f_{NN}(Y_{n_A}(k), U_{n_B}(k), W(k)) + \epsilon_{\rm w}(k+1) \\ y_m(kh) &= y(kh) + e_m(k)\end{aligned} \tag{31}$$

where $f_{NN}(\cdot,\cdot,\cdot)$ denotes the quasi-ARX feedforward neural network in Fig. 2, $Y_{n_A}(k)=[y(kh),\ldots,y((k-n_A+1)h)]$, $U_{n_B}(k)=[u_d(k),\ldots,u_d(k-n_B+1)]$, and W is a vector of neural network weights.

The system representation (31) is closely related to a general class of models studied in noisy time-series modelling. For these models, extended Kalman filter techniques have been developed for identification and state estimation, cf., Nelson and Wan [10,20]. In the system identification step the time-varying state vector $Y_{n_A}(k)$ and the network weights W are estimated simultaneously. In order to predict the output, the state of (31) is estimated using the network weights obtained in the identification phase. For the sake of simplicity, it is assumed that the process disturbance $\epsilon_{\rm w}(k+1)$ consists of discrete-time white noise. We refer to [10,20] for extended Kalman filter based techniques for the estimation of the state and the network weights in (31).

4. Simulation results

In this section the identification approaches described in Section 3 are applied to a non-linear bioreactor example process [19]. The process consists of a continuous stirred tank reactor (CSTR) with a constant volume, containing cells and nutrients. The control objective is to control the cell mass yield by manipulating the feed stream of nutrients into the reactor. The bioreactor is described by the differential equations

$$\begin{aligned}\frac{dc_1}{dt} &= -c_1u + c_1(1-c_2)e^{c_2/\gamma} \\ \frac{dc_2}{dt} &= -c_2u + c_1(1-c_2)e^{c_2/\gamma}\frac{\beta}{\beta-c_2}\end{aligned} \tag{32}$$

where c_1 and c_2 are dimensionless cell mass and substrate conversion, respectively, and u is the flow rate through the reactor. The parameter values $\beta=1.02$ and $\gamma=0.48$ are used.

The values of c_1 and c_2 lie in the interval $[0,1]$ and u is in $[0,2]$. When u exceeds a certain value the system begins to exhibit limit cycle behaviour and when the control is increased further the system becomes unstable. In this study only the stable region will be examined, and u is chosen to lie in the interval $[0,1]$.

The input u is obtained from a discrete-time input u_d by passing it through a zero-order hold followed by a low-pass filter $\mathscr{H}$. The filter $\mathscr{H}$ is taken as a first-order filter $\dot{u}(t)=A_Hu(t)+B_Hu_d(k), t\in[kh,kh+h)$, with $A_H=-100$ and $B_H=100$. The sampling time $h=0.5$ suggested in [19] is used.

It is assumed that only the cell mass c_1 is measured. Defining $y=x_{P,1}=c_1$, $x_{P,2}=c_2$, and $x=[x_{P,1},x_{P,2},u]^{\rm T}$ the generalized system is described by (1) with

$$f(x(t)) = \begin{bmatrix} f_P(x_P(t),u(t)) \\ A_Hu(t)\end{bmatrix},\quad B=\begin{bmatrix}0\\0\\B_H\end{bmatrix} \tag{33}$$
$$h(x(t)) = x_1(t)$$

where

$$f_P(x_P(t),u(t)) = \begin{bmatrix} -x_{P,1}(t)u(t)+\xi(t) \\ -x_{P,2}(t)u(t)+\xi(t)\frac{\beta}{\beta-x_{P,2}(t)}\end{bmatrix} \tag{34}$$

where $\xi(t) = x_{P,1}(t)(1 - x_{P,2}(t))e^{x_{P,2}(t)/\gamma}$. The differential Eqs. ((5) and (6)) take the form

$$\frac{d}{dt}\begin{bmatrix} \Phi_{11} & \Phi_{12} \\ 0 & \Phi_{22} \end{bmatrix} = \begin{bmatrix} \nabla_{x_P} f_P & \nabla_u f_P \\ 0 & A_H \end{bmatrix} \begin{bmatrix} \Phi_{11} & \Phi_{12} \\ 0 & \Phi_{22} \end{bmatrix},$$

$$\begin{bmatrix} \Phi_{11} & \Phi_{12} \\ 0 & \Phi_{1,22} \end{bmatrix}(kh) = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \quad (35)$$

$$\frac{d}{dt}[\,\Phi_{y,1}\Phi_{y,2}\,] \quad = [\,\Phi_{11} \quad \Phi_{12}\,], [\,\Phi_{y,1}(kh) \quad \Phi_{y,2}(kh)\,] = [\,0 \quad 0\,] \quad (36)$$

As the dynamic response of the low-pass filter $\mathscr{H}$ is much faster than the sampling rate, it follows that $u(kh) = u_d(k-1)$ and $\dot{u}(kh) = 0$ hold with high accuracy, and sequential application of the state-dependent state-space Eq. (8) gives

$$\begin{bmatrix} \Delta y(kh) \\ \Delta y(kh-h) \end{bmatrix} = T_1(k)\dot{x}_P(kh-2h) + T_2(k)\begin{bmatrix} \Delta u_d(k-1) \\ \Delta u_d(k-2) \end{bmatrix} \quad (37)$$

where

$$T_1(kh) = \begin{bmatrix} \Phi_{y,1}(kh)\Phi_{11}(kh-h) \\ \Phi_{y,1}(kh-h) \end{bmatrix} \quad (38)$$

$$T_2(kh) = \begin{bmatrix} \Phi_{y,2}(kh)B_H & \Phi_{y,1}(kh)\Phi_{12}(kh-h)B_H \\ 0 & \Phi_{y,2}(kh-h)B_H \end{bmatrix} \quad (39)$$

Solving (37) for the state $\dot{x}_P(kh-2\,h)$, introducing the reconstructed state into the system Eq. (8) and solving for $\Delta y(kh+h)$ gives the quasi-ARX representation

$$\begin{aligned} \Delta y(kh+h) = {} & A_1(k)\Delta y(kh) + A_2(k)\Delta y(kh-h) \\ & + B_1(k)\Delta u_d(k) + B_2(k)\Delta u_d(k-1) \\ & + B_3(k)\Delta u_d(k-2) \end{aligned} \quad (40)$$

where the parameters are given by

$$\begin{aligned} [\,A_1(k) \quad A_2(k)\,] &= \Phi_{y,1}(kh+h)\Phi_{11}(kh)\Phi_{11}(kh-h)T_1(kh)^{-1} \\ B_1(k) &= \Phi_{y,2}(kh+h)B_H \\ [\,B_2(k) \quad B_3(k)\,] &= \Phi_{y,1}(kh+h) \\ & \quad \times [\,\Phi_{12}(kh)B_H \quad \Phi_{11}(kh)\Phi_{12}(kh-h)B_H\,] \\ & \quad - [\,A_1(k) \quad A_2(k)\,]T_2(kh) \end{aligned} \quad (41)$$

Neural network based state-dependent parameter models of the form described in Section 3 were identified using input–output data. It turns out that the parameter B_3 in the state-dependent representation (40) is small, and $|B_3(k)| < 0.017|B_2(k)|$ holds in the whole operating region. Therefore, the parameter B_3 was ignored, and models with two A- and two B-parameters were identified. The model parameters were represented as functions of two past outputs and three past inputs. This agrees with the theoretical minimum number of inputs and outputs required to reconstruct the state-dependent parameters of a second-order system (cf., [18]), and it also resulted in the best models.

As the system dynamics vary significantly in the operating region, quite long training sequences are required in order to collect a sufficient amount of data for system identification. This problem is well known in non-linear identification [11]. In the stochastic case, a test data sequence of sufficient length is required as well, in order to ensure reliable model validation results which are not sensitively dependent on the particular noise sequence. In this study, both deterministic and stochastic systems were identified. In the deterministic case, 1000 input–output training data pairs were used, and the stochastic identification experiments were based on 2500 data pairs. In both cases, 2500 data pairs were used for testing.

In the deterministic case, a quasi-ARX (Q-ARX) feedforward neural network model of the form shown in Fig. 2 with $n_C = 0$ was identified. The network input consisted of two past outputs and three past inputs. The best performance on the test data sequence was achieved using a network with five neurons in the hidden layer (corresponding to a total of 54 network weights), giving the root-mean-square prediction error 1.89×10^{-4} on the training data and 5.74×10^{-4} on the test data. The prediction results and the model parameters are shown in Figs. 3 and 4, respectively, for a part of the test set. It is seen that the identified neural network quasi-ARX representation (28) correctly reconstructs the theoretical parameters of the quasi-ARX system representation (40). For comparison, a standard feedforward neural network ARX (NNARX) model [11] was also identified. The inputs to the network were the same as for the quasi-ARX model. The best performance was obtained using nine hidden layer neurons, corresponding to 43 weights. The

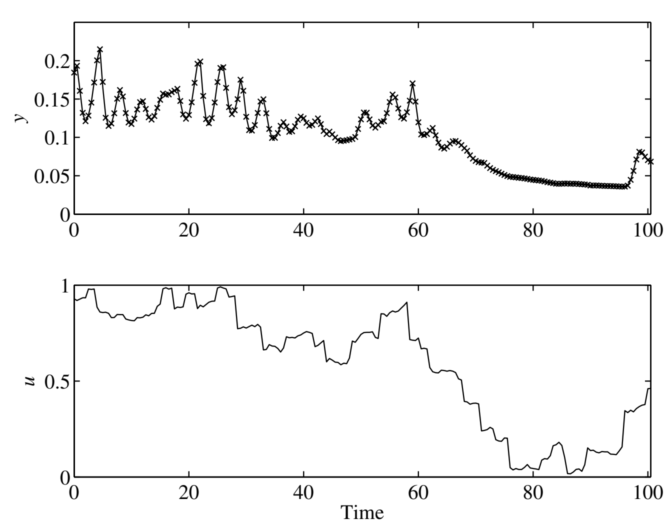

Fig. 3. Upper graph: system output (solid line) and one-step ahead predictions using identified quasi-ARX model (crosses). Lower graph: Input.

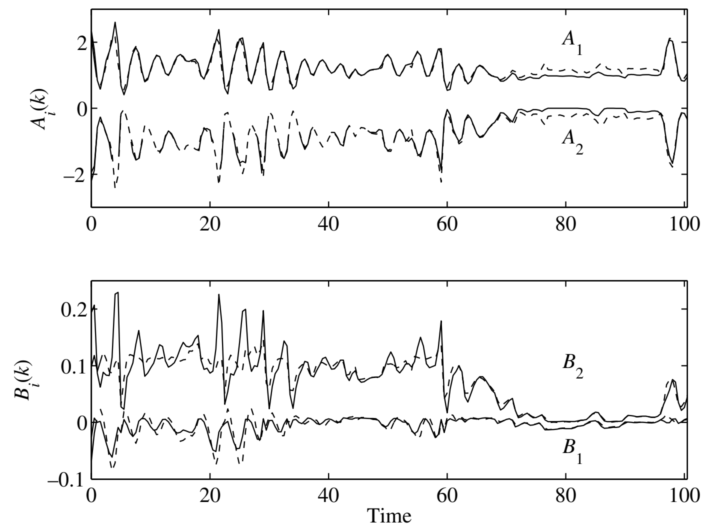

Fig. 4. Parameters of theoretical quasi-ARX system representation (solid lines) and identified model (dashed lines).

root-mean-square prediction error was 1.95×10^{-4} on the training data and 5.09×10^{-4} on the test data.

In order to study the identification of state-dependent parameter models in the stochastic case, the system (32) was augmented with a noise model, consisting of additive drifting process noise $v(t)$ with incremental variance $r_v = 2 \times 10^{-7}$ and measurement noise with variance $\sigma_m^2 = 10^{-4}$. By Theorem 2.1 the minimum prediction error variance is then 1.03×10^{-4}. The stochastic system can be described by the quasi-ARMAX model (23), where the A- and B-parameters are defined by (41) and the C-parameters are given by (24). Input–output data were generated by applying the same input sequence which was used in the deterministic case. The results obtained using various identification methods to identify the noise-corrupted system are summarized in Table 1. The table presents the results achieved with the optimal network complexities (number of hidden layer neurons), which give the smallest mean square prediction errors on the test data. The mean square one-step ahead prediction errors (MSE) are given both with respect to the measured output y_m and the noise-free system output y. The performance achieved with the optimal predictor based on the known system parameters and Theorem 2.1 is also given.

In order to study the effect of the number of C-parameters (n_C), neural network quasi-ARX and quasi-ARMAX (Q-ARMAX) models (28) with various numbers of parameters were identified. The relation (24) was not used in the identification experiments, but the C-parameters were identified independently. Models with both constant and state-dependent C-parameters were trained. In both cases, the smallest prediction error for the test data was achieved with $n_C = 3$, which corresponds to the theoretical number of parameters. Due to the incremental form of the quasi-ARMAX model, at least a second-order noise model is required for a satisfactory modelling of the noise dynamics. In particular, a noise model with one C-parameter tends to become unstable, as the parameter value is approximately equal to one.

Table 1
Mean square one-step ahead prediction errors (MSE) obtained with various quasi-ARMAX model structures

Model	n_C	n_h	n_w	MSE ($\times 10^4$)			
				$y_m - \hat{y}$		$y - \hat{y}$	
				Training	Test	Training	Test
Q-ARX	0	2	24	1.90	2.27	0.97	1.26
	0	3	34	1.79	2.17	0.87	1.18
	0	4	44	1.70	2.01	0.76	1.01
	0	5	54	1.60	1.87	0.69	0.90
	0	6	64	1.53	2.02	0.66	1.03
Q-ARMAX	1	2	25	1.45	1.72	0.46	0.73
	2	2	26	1.33	1.50	0.34	0.50
Constant	3	1	17	1.29	1.42	0.29	0.47
C-parameters	3	2	27	1.23	1.37	0.24	0.40
	3	3	37	1.21	1.42	0.22	0.45
	3	4	47	1.19	1.46	0.22	0.52
Q-ARMAX	3	1	20	1.26	1.37	0.25	0.42
State-dependent	3	2	33	1.19	1.35	0.18	0.38
C-parameters	3	3	46	1.18	1.71	0.19	0.74
Q-ARX EKF	0	2	24	1.19	1.28	0.18	0.32
	0	3	34	1.14	1.28	0.15	0.30
	0	4	44	1.13	1.35	0.13	0.35
Optimal	3	–	–	1.14	1.19	0.14	0.19

Here n_h denotes the number of hidden layer neurons and n_w is the total number of neural network weights. y_m is the measured output, y is the noise-free system output and $\hat{y}$ is the predicted output.

The predictions and the measurements when using an identified model with three constant C-parameters are shown in Fig. 5. Fig. 6 shows the approximated and the theoretical parameters. The constant C-parameters correspond well with the average values of the theoretical ones. The results achieved when using models with state-dependent C-parameters were similar to the case with constant parameters (cf., Table 1). However, the neural network

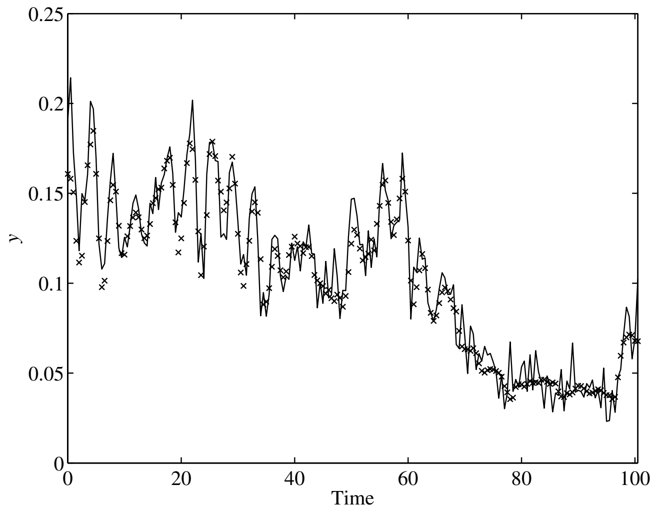

Fig. 5. System output y_m (solid line) and one-step ahead predictions using neural network quasi-ARMAX model with three C-parameters. The input sequence is the same as in Fig. 3.

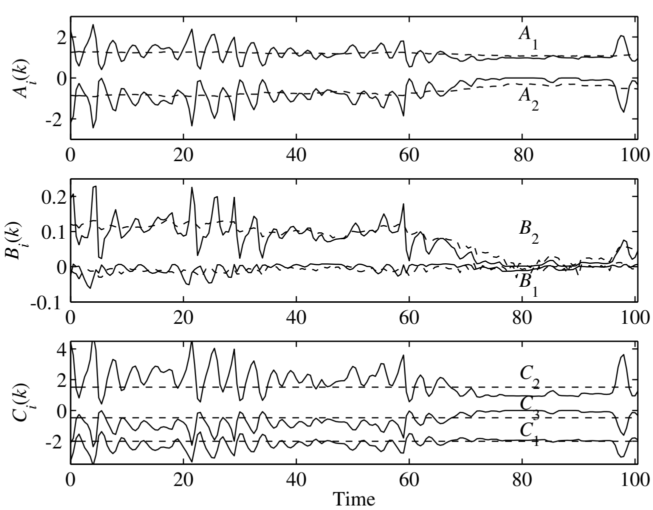

Fig. 6. Parameters of theoretical (solid lines) and neural network (dashed lines) quasi-ARMAX model parameters.

approximator is harder to train when the C-parameters are taken as functions of past inputs and outputs.

Simultaneous estimation of the noise-free system output and the neural network weights (Q-ARX EKF) according to Section 3.2 gives better performance than the quasi-ARMAX model (cf., Table 1). The procedure is, however, more complex to implement and requires knowledge about the noise variances. The results are shown in Figs. 7 and 8.

For comparison, feedforward neural network ARX (NNARX) and ARMAX (NNARMAX) models [11],

$$y_{NN}(kh+h) = g_{NN}(\varphi_{yu}(k)) + \sum_{i=1}^{n_C} C_i \epsilon_{NN}(k-i+1) \tag{42}$$

were also identified. Here $g_{NN}(\cdot)$ is a feedforward neural network with input vector

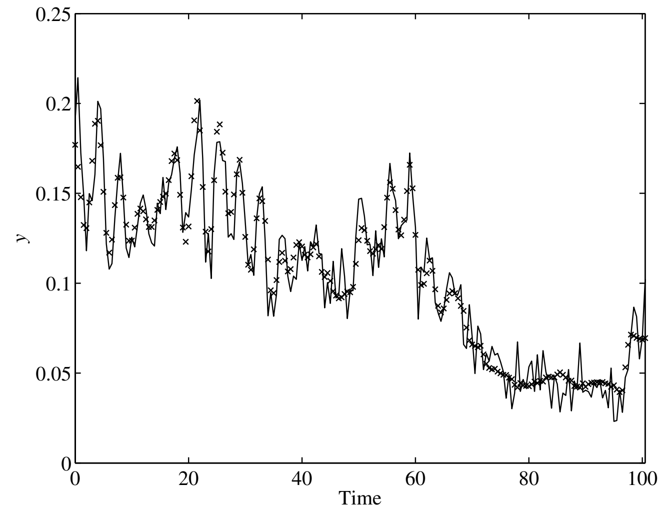

Fig. 7. System output y_m (solid line) and one-step ahead predictions using the quasi-ARX model (29) with extended Kalman filter estimation of the system output. The input sequence is the same as in Fig. 3.

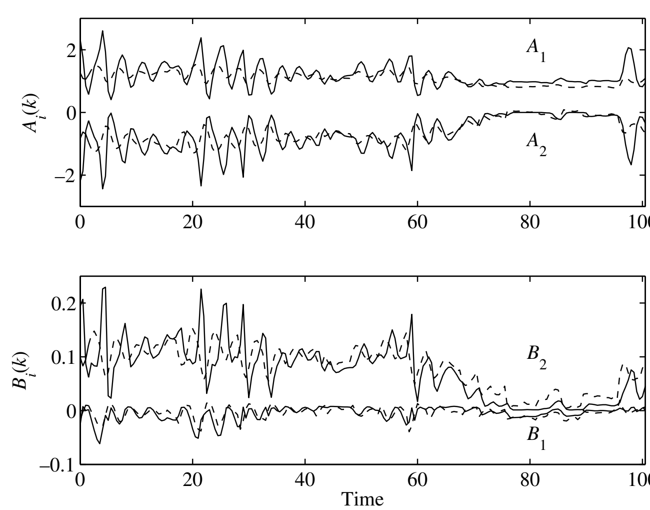

Fig. 8. Parameters of quasi-ARX model (29) with extended Kalman filter estimation of the system output.

$$\varphi_{yu}(k) = [y_m(kh), \ldots, y_m((k-n_y+1)h), u_d(k), \ldots, u_d(k-n_u+1)]$$

and $\epsilon_{NN}(k+1) = y_m(kh+h) - y_{NN}(kh+h)$. The networks were chosen to have the same inputs as the quasi-ARMAX models. The extended Kalman filter approach of [10,20] (cf., Section 3.2) was also applied to estimate the noise-free system output and the neural network weights of the NNARX models (NNARX EKF). The results are presented in Table 2.

The results of Tables 1 and 2 show that the state-dependent parameter models give better prediction performance than the NNARX and NNARMAX models. The best performance was achieved with the neural network quasi-ARX model structure (31) with estimation of the noise-free system output. Notably, the number of hidden layer neurons

Table 2
Mean square one-step ahead prediction errors (MSE) obtained with various NNARMAX model structures

Model	n_C	n_h	n_w	MSE ($\times 10^4$)			
				$y_m - \hat{y}$		$y - \hat{y}$	
				Training	Test	Training	Test
NNARX	0	4	29	1.48	1.80	0.54	0.82
	0	5	36	1.46	1.79	0.52	0.82
	0	6	43	1.45	1.80	0.52	0.82
NNARMAX	3	4	32	1.44	1.74	0.50	0.81
	3	5	39	1.43	1.75	0.51	0.82
NNARX EKF	0	4	29	1.33	1.66	0.36	0.66
	0	5	36	1.33	1.66	0.36	0.67
Optimal	3	–	–	1.14	1.19	0.14	0.19

Here n_h denotes the number of hidden layer neurons and n_w is the total number of neural network weights. y_m is the measured output, y is the noise-free system output and $\hat{y}$ is the predicted output.

required to obtain the best approximation accuracy on the test data is quite small ($n_h = 2 - 3$) for the state-dependent parameter models.

5. Conclusion

State-dependent parameter representations of stochastic sampled-data systems have been studied. The analysis shows that the class of systems under considerations can be described by ARMAX models with state-dependent parameters. The model parameters can be characterized exactly as functions of past outputs and inputs, including the disturbances.

In this work, feedforward neural networks have been used to describe the model parameters as functions of past inputs and outputs. Two approaches have been studied. The first method uses a quasi-ARMAX model structure, and the parameters are modelled as functions of past inputs and the measured outputs, which are corrupted by measurement noise. In the second approach, the parameters are taken as functions of past inputs and the noise-free process outputs, which are estimated using an extended Kalman filter technique. Experimental results show that both methods can be used to train the neural networks from input–output data to give good approximations of the model parameters and the system dynamics.

The results were also compared to other approaches. The prediction errors achieved with the neural network state-dependent parameter models were uniformly smaller than when using neural network based NNARMAX models having a corresponding complexity. This result is in accordance with previous studies on state-dependent parameter ARX models, cf., [12].

Acknowledgement

Bernt Åkesson was supported by the Finnish Graduate School in Chemical Engineering (GSCE).

References

[1] D. Aeyels, Generic observability of differentiable systems, SIAM Journal on Control and Optimization 19 (5) (1981) 595–603.
[2] S. Haykin, Neural Networks. A Comprehensive Foundation, MacMillan College Publishing Company, New York, 1994.
[3] J. Hu, K. Hirasawa, A method for applying multilayer perceptrons to control of nonlinear systems, in: Proceedings of the 9th International Conference on Neural Information Processing, Singapore, 2002, pp. 1267–1271.
[4] J. Hu, K. Kumamaru, K. Inoue, K. Hirasawa, A hybrid quasi-ARMAX modeling scheme for identification of nonlinear systems, Transactions of the Society of Instrument and Control Engineers 34 (8) (1998) 977–985.
[5] J. Hu, K. Kumamaru, K. Hirasawa, A quasi-ARMAX approach to modeling of nonlinear systems, International Journal of Control 74 (18) (2001) 1754–1766.
[6] D.J. Leith, W.E. Leithead, Gain-scheduled controller design: an analytic framework directly incorporating non-equilibrium plant dynamics, International Journal of Control 70 (2) (1998) 249–269.
[7] D.J. Leith, W.E. Leithead, Gain-scheduled and nonlinear systems: dynamic analysis by velocity-based linearization families, International Journal of Control 70 (2) (1998) 289–317.
[8] I.J. Leontaritis, S.A. Billings, Input–output parametric models for nonlinear systems. Part II: Stochastic nonlinear systems, International Journal of Control 41 (2) (1985) 329–344.
[9] A.U. Levin, K.S. Narendra, Recursive identification using feedforward neural networks, International Journal of Control 61 (3) (1995) 533–547.
[10] A.T. Nelson, E.A. Wan, A two-observer Kalman framework for maximum-likelihood modeling of noisy time series, in: Proceedings of the International Joint Conference on Neural Networks, vol. 3, 1998, pp. 2489–2494.
[11] M. Nørgaard, O. Ravn, N.K. Poulsen, L.K. Hansen, Neural Networks for Modelling and Control of Dynamic Systems, Springer-Verlag, London, 2000.
[12] H.T. Peng, T. Ozaki, V. Haggan-Ozaki, Y. Toyoda, A parameter optimization method for radial basis function type models, IEEE Transactions on Neural Networks 14 (2) (2003) 432–438.
[13] F. Previdi, M. Lovera, Identification of a class of nonlinear parametrically varying models, in: Proceedings of the European Control Conference, Porto, Portugal, 2001, pp. 3086–3091.
[14] M.B. Priestley, Non-linear and Non-stationary Time Series Analysis, Academic Press, London, 1988.
[15] J. Stark, Delay embeddings for forced systems. I. Deterministic forcing, Journal of Nonlinear Science 9 (1999) 255–332.
[16] J. Stark, D.S. Broomhead, M.E. Davies, J. Huke, Delay embeddings for forced systems. II. Stochastic forcing, Journal of Nonlinear Science 13 (2003) 519–577.
[17] F. Takens, Detecting strange attractors in turbulence, in: D.A. Rand, L.S. Young (Eds.), Dynamical Systems and Turbulence, Springer-Verlag, 1980.
[18] H.T. Toivonen, State-dependent parameter models of non-linear sampled-data systems: a velocity-based linearization approach, International Journal of Control 76 (18) (2003) 1823–1832.
[19] L.H. Ungar, A bioreactor benchmark for adaptive network-based process control, in: W.T. MillerIII, R.S. Sutton, P.J. Werbos (Eds.), Neural Networks for Control, The MIT Press, Cambridge, MA, 1990, Chapter 16, Appendix A, pp. 387–402, 476–480.
[20] E.A. Wan, A.T. Nelson, Dual extended Kalman filter methods, in: S. Haykin (Ed.), Kalman Filtering and Neural Networks, John Wiley & Sons, New York, NY, 2001, Chapter 5, pp. 123–173.
[21] P.C. Young, Applying parameter estimation to dynamic systems. Part I: Theory, Control Engineering 16 (10) (1969) 119–124.
[22] P.C. Young, P. McKenna, J. Bruun, Identification of non-linear stochastic systems by state dependent parameter estimation, International Journal of Control 74 (18) (2001) 1837–1857.