SYSTEM IDENTIFICATION METHODS FOR PLASMA ENHANCED CHEMICAL VAPOR DEPOSITION [1]

Michael Gibson * **Enrique Ferreira** * **Xu Cheng** * **Thomas Knight** *
David Greve * **Bruce Krogh** *

* *ECE Department, Carnegie Mellon University*
email: edf@ece.cmu.edu

Abstract. The introduction of new in situ sensing creates the possibility of directly controlling critical process variables in plasma enhanced chemical vapor deposition systems (PECVD). The complexity of this process makes it necessary to develop empirical models of the system dynamics. This paper describes the experimental and numerical procedures for identifying both transfer function and recurrent neural net multi-input multi-output models of PECVD dynamics. The models are compared in terms of the resulting mean-squared-prediction-error for verification data sets. The neural net models are shown to be mildly superior in cases where the input signals vary widely, stimulating the nonlinear response of the system.

Keywords. system identification, neural networks, microelectronics manufacturing processes, plasma deposition

1. INTRODUCTION

Plasma enhanced chemical vapor deposition (PECVD) is an important process in the manufacture of integrated circuits (ICS) and flat panel displays. In current practice, the PECVD process is controlled by independent regulators for variables such as temperature, pressure and gas flow rates. Setpoint values for these variables are determined by off-line studies and applied without adjustment during the processing of a given batch of wafers. Although setpoint tuning may be performed from batch-to-batch using techniques such as statistical process control (Sachs *et al.*, 1995), the use of feedback to adjust the process setpoints in real time has not been possible until recently because of the absence of in situ measurements of process variables directly related to the actual process performance variables.

This paper reports on the development and evaluation of control-oriented models of PECVD dynamics for an experimental system at Carnegie Mellon University. A differentially-pumped quadrupole mass spectrometer, partial pressure transducer (PPT), has been integrated into the process equipment to sample species directly incident on the wafer platform during processing. The goal of this project is to control plasma parameters directly related to the quality of the resulting deposition. Several multivariable control experiments using the real-time mass spectrometer sensor have been performed. The controllers for these experiments have been based on simple first-order models of the process dynamics. The objective of the research reported in this paper is to investigate and evaluate techniques for obtaining more sophisticated models to be used in model predictive control algorithms. The experimental apparatus and preliminary control results are described elsewhere (Knight *et al.*, 1995).

[1] Partially supported by NSF, Grant ECS 93-12406, MKS Inc. and CONICYT–BID

The following section describes the experiments performed to obtain data for the system identification experiments. Sections 3 and 4 describe the numerical methods used to obtain transfer function and neural net models of the system dynamics from the experimental data sets. The performance of these models is evaluated and compared in section 5. The concluding section summarizes the results of this research and the potential for using these models for control design.

2. EXPERIMENTAL PROCEDURE

The regulated variables in the PECVD system are the two gas flows (silane and ammonia), RF power, temperature, and the chamber pressure. Three measured process variables were considered in this study: the DC bias voltage and the concentrations of mass 16 and mass 30 from the PPT. The DC bias voltage is directly related to the deposition mechanism of the plasma, and is attractive for control purposes in that it tends to have a fast ($\ll$ 1 s) response to input changes. Mass 16 is the radical NH_2^+, which is indicative of the amount of unreacted ammonia (Greve *et al.*, 1996). Mass 30 is the radical SiH_2^+, which is indicative of the amount of silane remaining in the chamber.

Factorial experiments were run to determine an operating point. Analysis of the resulting films led to the choice of the nominal operating point: 3 sccm silane, 60 sccm ammonia, 20 W RF power, 250 C and 0.400 Torr. The silane flow was limited to the region 0.5–5.5 sccm: the lower bound is due to a slow deposition rate and the upper bound reflects the goal of PECVD when used to deposit insulator layers in ICs, namely to deposit a nitrogen rich film — higher flows of silane would produce a nitrogen poor film. The ammonia flow was limited to the region 20–72 sccm. Initial experimentation showed that the pressure controller had significant trouble keeping a constant pressure at flows lower than 20 sccm, hence the lower bound. This presents no real problem, as the goal is to produce nitrogen rich films. The upper bound of 72 was chosen because the ammonia mass flow controller has a maximum setting of 73 sccm. A lower bound of 5 W was set for the RF power, due to resolution of the RF matching network, tuning and plasma extinction. An upper bound of 35 W was set for RF power, because powdering becomes a problem at higher powers.

To determine the basic form of the frequency responses for the different inputs, a waveform was generated whose energy spectrum was non-zero at points logarithmically spaced between discrete frequencies $\theta = 0$ and $\theta = \pi$. Using this waveform, single input multiple output (SIMO) experiments were run, in which a scaled version of the

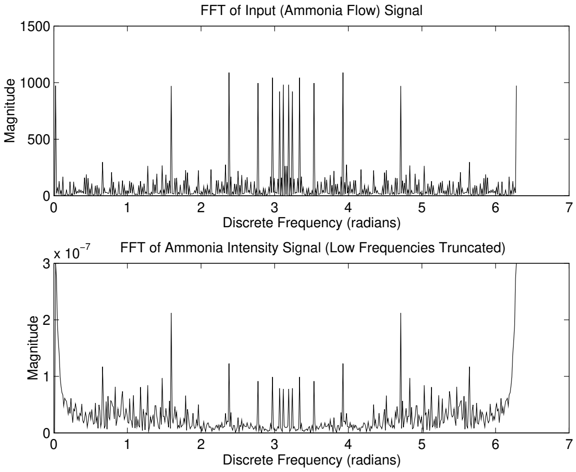

Fig. 1. FFTs for representative input and output signals used in determining the frequency response of the system.

(discretized) waveform was applied to one control input, while the other inputs were held at their nominal operating points. The SIMO silane experiment consisted of flows ranging from 2.5–3.5 sccm, the SIMO ammonia experiment consisted of flows ranging from 50–70 sccm and the SIMO RF power experiment consisted of powers ranging from 15–25 W. Frequency spectra of typical input and output signals are shown in Figure 1. The data from these SIMO experiments were used to estimate the frequency response at those frequencies contained in the input signal. The response has a low magnitude at $\pi \pm \frac{\pi}{16}$, so that frequency was used as a cutoff between the high frequency regime (hereafter referred to as the *fast regime*), and the low frequency regime. This paper focuses on the fast regime which is the primary regime for real-time multivariable control action. A waveform containing most of its energy in the fast regime was created. A set of experiments was run in which this waveform was applied to a single input while the others were held constant. The amplitudes of the waveform were evenly spaced over the operating region, and the order of the experiments was randomized. Three complete sets of experiments were used for identification purposes, and a fourth complete set was used for validation. The duration of the waveform was 10 minutes, which is approximately the time it takes to for a complete deposition. The data for each data set consisted of roughly 600 data points (600 seconds $\times$ 1 Hz).

3. IDENTIFICATION - TRANSFER FUNCTIONS

Two multiple input multiple output (MIMO) transfer function matrices were computed based on the experimental data sets. First order ARX (AutoRegressive with eXogenous inputs) (Ljung, 1987) models with time delay were used as a baseline for subsequent comparisons. These models are computationally simple and widely used in industry. Initial control designs have been based on these models (Knight , 1995).

The second class of models considered was Box-Jenkins ARMAX (AutoRegressive Moving Average with eXogenous input) models (Ljung, 1987) of the form

$$y_i(t) = \sum_{j=1}^{3} \frac{B_{ij}(q)}{F_{ij}(q)}\, u_j(t) + \frac{C_i(q)}{D_i(q)}\, e_i(t), \qquad (1)$$

where the inputs u_1, u_2 and u_3 are ammonia flow, silane flow and RF power, respectively, and the outputs y_1, y_2 and y_3 are DC bias voltage, mass 30 and mass 16, respectively. This model structure was chosen because it allows characterization of the error independent of the inputs to the system, and because for suitable orders of B, C, D and F it reduces to most other commonly used model structures, including the ARX structure.

The Matlab System Identification Toolbox was used for all calculations. Where applicable, actual function names appear in small capitals. For each output, the following iterative approach was taken:

(1) Start with the minimum possible order of B, C, D and F.
(2) Use prediction error methods (Matlab command PEM) to create the best model of the chosen structure.
(3) Based on the correlation functions of the residuals (Matlab command RESID), do one of the following:
   - If the auto-correlation function (ACF) of the residuals and all of the cross-correlation functions (CCFs) between residuals and inputs are within the 99 % confidence interval, accept the model and go to step 6.
   - Consider the ACF and the three CCFs. If, of the four, the ACF exceeds its confidence bound by the most, increase the order of C, D or both.
   - Otherwise, increase the order of B_{ij}, F_{ij} or both for the input j whose CCF exceeds its confidence bound by the most.
(4) For each new model structure, calculate the best model parameters (with PEM) and compare its loss function and Akaike's final prediction–error criterion (FPE) to those values for previous models. If the new model is better (i.e. has lower values of loss function and FPE), keep the new model and go to step 2.
(5) Reject the model. Go to step 3, incrementing the order model in a different way.
(6) Manually check all estimated parameters and their covariances. If the covariance is greater than the magnitude of the parameter estimate, try a model where that coefficient is set to 0. If that model is not significantly worse, use it instead, in the name of parsimony.

Using the approach described above, the following first order ARX model (FO) with time delay was obtained:

$$\begin{bmatrix} DCBias(t) \\ Mass16(t) \\ Mass30(t) \end{bmatrix} = \mathbf{A} \begin{bmatrix} NH_3(t) \\ SiH_4(t) \\ RFPower(t) \end{bmatrix},$$

$$\mathbf{A} = \begin{bmatrix} \dfrac{0.0203}{1-0.675q^{-1}} & \dfrac{0.734q^{-2}}{1-0.805q^{-1}} & \dfrac{-0.448}{1-0.785q^{-1}} \\ \dfrac{0.504}{1-0.750q^{-1}} & \dfrac{-11.3q^{-2}}{1-0.467q^{-1}} & \dfrac{-1.12}{1-0.692q^{-1}} \\ \dfrac{-0.0339q^{-5}}{1+0.375q^{-1}} & \dfrac{36.8q^{-1}}{1-0.233q^{-1}} & \dfrac{-0.197}{1+0.346q^{-1}} \end{bmatrix}$$

The following higher order ARMAX model (HO) was obtained:

$$\begin{aligned} y_{DC}(t) = {} & \frac{0.0117+0.0660q^{-1}}{1-0.482q^{-1}+0.544q^{-2}}\, u_{NH_3}(t) \\ & + \frac{-1.82+0.140q^{-1}}{1-0.225q^{-1}}\, u_{RF}(t) \\ & + \frac{0+1.38q^{-1}-1.15q^{-2}}{1-1.35q^{-1}+0.434q^{-2}-0.0729q^{-3}} u_{SiH_4}(t) \\ & + \frac{1+0.0233q^{-1}+0.206q^{-2}-0.624q^{-3}}{1-0.272q^{-1}-0.0361q^{-2}-0.692q^{-3}} e_{DC}(t) \end{aligned}$$

$$\begin{aligned} y_{M30}(t) = {} & \frac{-0.128}{1}\, u_{NiH_3}(t) + \frac{-1.84}{1}\, u_{RF}(t) \\ & + \frac{11.7+17.3q^{-1}}{1-0.158q^{-1}}\, u_{SiH_4}(t) \\ & + \frac{1-0.212q^{-1}-0.512q^{-2}-0.120q^{-3}}{1-0.210q^{-1}-0.7811q^{-2}} e_{M30}(t) \end{aligned}$$

$$\begin{aligned} y_{M16}(t) = {} & \frac{0.220+1.85q^{-1}+0.202q^{-2}}{1-0.291q^{-1}+0.304q^{-2}}\, u_{NiH_3}(t) \\ & + \frac{-0.368-1.61q^{-1}}{1-0.494q^{-1}}\, u_{RF}(t) \\ & + \frac{0-2.74q^{-1}-2.42q^{-2}}{1-0.822q^{-1}+0.155q^{-2}}\, u_{SiH_4}(t) \\ & + \frac{1-0.411q^{-1}-0.440q^{-2}}{1-0.996q^{-1}}\, e_{M16}(t) \end{aligned}$$

All of the poles in this model lie within the unit circle, but several of the zeros lie outside the unit circle, indicating that the system is non-minimum phase. This could be problematic in controller design.

4. IDENTIFICATION - NEURAL NETWORKS.

The use of neural networks in system identification and control has been growing in recent years. Several authors (Narendra and Parthasarathy, 1991; Nerrand *et al.*, 1993) have shown many architectures and learning algorithms to deal with different types of system models. Many applications have been developed taking advantage of the properties of these models, e.g. in forecasting (Ydstie, 1990). Neural networks are useful for systems that have a very complex nonlinear dynamic behavior and, when incorporated into appropriate control algorithms, offer the possibility to improve the performance over methods based on standard identification techniques. The PECVD system presented in this paper can be classified within this group of complex nonlinear systems. Other authors have already applied neural networks techniques to model a similar system (Han *et al.*, 1994; Kim and May, 1994) with successful results. However, they have focused on modelling the response surface, that is, the steady-state input-output relationships. In this project, the main interest was modeling the dynamic behavior of the PECVD system.

We considered three types of non-linear models: *output error* (NOE), $NARX$ (Nonlinear AutoRegressive with eXogenous inputs), and $NARMAX$ (Nonlinear AutoRegressive Moving Average with eXogenous inputs). The equations for the models are:
NOE model:

$$x(k) = f(X(k-1), U(k-1)),$$
$$y(k) = x(k) + w(k)$$

NARX model:

$$x(k) = f(X(k-1), U(k-1)) + w(k),$$
$$y(k) = x(k)$$

NARMAX model:

$$x(k) = f(X(k-1), U(k-1), W(k-1)) + w(k),$$
$$y(k) = x(k)$$

where

$$\begin{aligned} X(k-1) &= [x(k-1), x(k-2), \cdots, x(k-N)], \\ U(k-1) &= [u(k-1), u(k-2), \cdots, u(k-M)], \\ W(k-1) &= [w(k-1), w(k-2), \cdots, w(k-P)], \end{aligned}$$

where the vectors $X(k)$, $U(k)$ and $W(k)$ stand for the states, inputs and noise, respectively, and $f(\cdot)$ is a nonlinear function of its arguments. It has been shown that

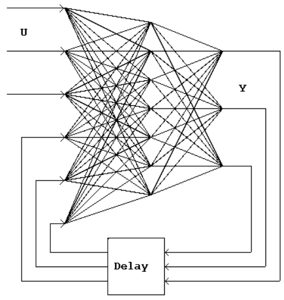

Fig. 2. Time lagged Recurrent Neural Network.

any feedback network can be transformed to a time-lagged recurrent network with feedback connections only from the outputs to the inputs (Nerrand , 1993). Figure 2 shows a diagram of such a neural net architecture. Several methods have been developed for training a recurrent network, (Nerrand *et al.*, 1994) including the use of extended Kalman filtering (EKF) (Williams, 1992), and robust estimation (Connor *et al.*, 1994). Truncated backpropagation through time is the most economical method in terms of the computational complexity of the algorithm, but it takes longer to train. EKF is the fastest, and it can be implemented on-line, but it requires more memory.

For the PECVD the external inputs are: RF power, SiH_4 and NH_3 flows, and the outputs are: DC bias and the partial pressures of the species of masses 16 (NH_2) and 30 (SiH_2). The following steps are taken:

- Use experimental data to train the neural networks of the types chosen to predict the behavior of the system in the next time step.
- Optimize the size of the network, i.e. the number of inputs and hidden nodes, to get an acceptable performance.
- To prevent overtraining: observe the performance in the validation sets of data during training, stopping the process when the performance begins to degrade.

Semidirected algorithms using EKF procedures with robust estimation were used to train the networks for the NOE and $NARMAX$ models. In the case of the $NARX$ model, standard backpropagation is sufficient (Nerrand , 1994). Stuttgart Neural Network Simulator, SNNS version 3.3, and Matlab were used for the computations.

A pruning algorithm was run after some training to eliminate those hidden nodes with small correlation with the output of the network. Standard analysis including mean square error on the estimated and validation sets of data and correlation computations were used to compare the different neural network models against the first and higher order linear models developed. The optimized neural networks for each model have the characteristics shown in table 1. The exogenous inputs are

Table 1. **Neural Network Models.**

Prop.	**NOE**	**NARX**	**NARMAX**
no. exogenous Inputs	10	10	19
no. feedback Inputs	9	9	9
no. Hidden units	22	23	19
no. Output units	3	3	3
MSE training set			
DCbias	0.70	1.05	0.33
Mass 16 ($\times 10^{19}$)	1.05	1.25	0.39
Mass 30 ($\times 10^{21}$)	2.12	2.51	1.03
MSE validation set			
DCbias	2.20	3.49	1.32
Mass 16 ($\times 10^{19}$)	5.61	6.55	3.56
Mass 30 ($\times 10^{21}$)	6.06	5.45	3.91

composed of the three previous inputs, the three previous outputs and the time, in secs., from the beginning of the experiment. The feedback inputs are the two previous output values for the NOE model or errors for the $NARMAX$ model. Tables 2, 3 and 4 and figure 3 show the performance of the models for step changes in the inputs. Figures and tables show that the $NARMAX$

Table 2. MSE for DC Bias Voltage.

Input(Step)	NOE	NARX	NARMAX
RF	0.81	1.06	0.55
NH_3	1.03	2.45	0.85
SiH_4	2.27	5.01	1.09

Table 3. MSE for mass 16 ($\times 10^{19}$)

Input(Step)	NOE	NARX	NARMAX
RF	1.54	2.72	1.24
NH_3	1.55	3.15	1.33
SiH_4	2.12	3.54	1.42

Table 4. MSE for mass 30 ($\times 10^{21}$)

Input(Step)	NOE	NARX	NARMAX
RF	8.15	9.12	7.78
NH_3	7.94	7.86	7.61
SiH_4	8.42	8.45	8.13

model is the best model of the three to identify the dynamics of the PECVD system. Introduction of the time from the beginning of the experiment as an external input to the neural network was very useful in order to compensate for the drift of the system and other disturbance variables that are not considered here.

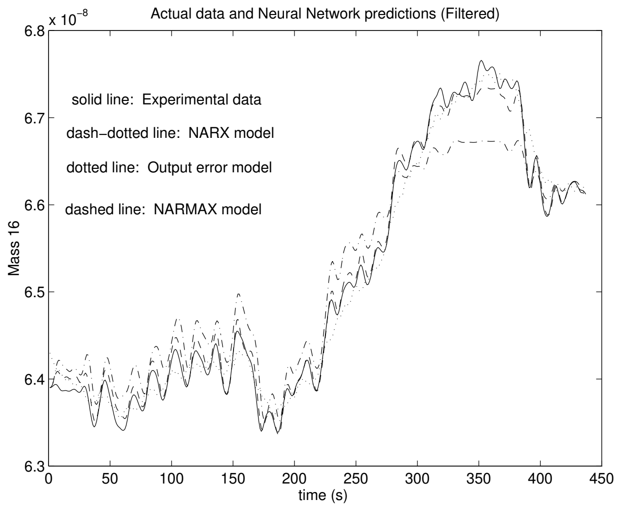

Fig. 3. Comparison of Neural Network predictions of Mass16 output for a step in RF Power.

5. COMPARISON OF MODELS

The objective of this modeling exercise is to create a model which accurately predicts the behavior of the system. To compare different models, the mean square error (MSE) of the residuals was computed for each model type, using both SISO step response data and the validation data. A comparison of the different models and the system responses can be found in the Figure 4 and in the following tables. In all cases, the higher order linear

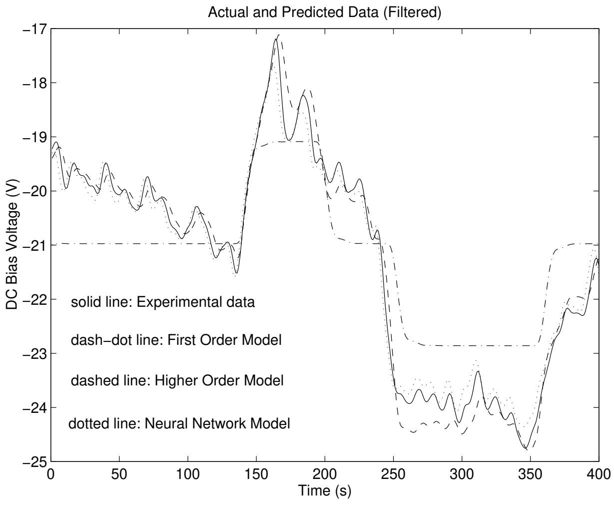

Fig. 4. Actual and predicted system response for a silane step change.

model outperforms the simple ARX model. Comparing the neural networks models with the higher order linear model it is observed that the performance is quite similar for the step changes in the case of the NARMAX model, showing that in these regions a linear approximation for the system is accurate. In the case of the validation set, where the system is going farther away from the operating point chosen, the NARMAX model is better than the HO model.

Table 5. MSE for step input responses.

Output	Input	FO	HO	NN
DC Bias	RF	1.80	0.557	0.55
	NH_3	2.15	0.749	0.85
	SiH_4	4.56	0.623	1.09
Mass 16	RF	10.8	1.27	1.24
	NH_3	3.46	1.25	1.33
	SiH_4	2.54	1.65	1.42
Mass 30	RF	15.9	7.94	7.78
	NH_3	8.50	7.63	7.61
	SiH_4	10.2	7.84	8.13

Table 6. MSE for Validation Data

Output	FO	HO	NN	Scale
DC Bias	16.7	7.63	1.32	1
Mass 16	26.2	5.71	3.56	10^{-19}
Mass 30	51.1	14.2	3.91	10^{-21}

6. DISCUSSION

Both classical linear and neural network system identification were used to model a PECVD system. Many possibilities were investigated, including design of experiments and model refinements, to give a better framework for a good comparison between the methods. The results show that the neural network constitutes an improvement over the use of linear models when the operation of the system covers a large region. The crucial issue is, however, how the models can be used for control design and implementation.

Transfer function models can be used as the basis for standard controller design methods. The primary method proposed for using neural nets in process control is as the system model in a model predictive control (MPC) algorithm, e.g. (Draeger and Engell, 1994). Given the extensive computation involved to train neural networks as compared to the linear identification algorithms, it remains to be seen whether the neural net models offer any real advantage even for MPC control of this PECVD system. Since it is the eventual closed-loop system performance that matters, neural net based MPC must be compared with linear control techniques. Currently, an evaluation of the use of higher order linear models over the simple first-order models for control design methods is being investigated.

7. REFERENCES

Connor, J.T., R.D. Martin and L.E. Atlas (1994). Recurrent neural networks and robust time series prediction. *IEEE Trans. on Neural Networks* **5**(2), 240–254.

Draeger, A. and S. Engell (1994). Nonlinear model predictive control using neural net plant models. Technical report. Dept. of Chem. Engineering, Univ. of Dortmund.

Greve, D.W., T.L. Knight, X. Cheng, B.H. Krogh, M.A. Gibson and J. LaBrosse (1996). Process control based on quadrupole mass spectrometry. *J. of Vacuum Science and Technology - B.*

Han, S.S., M. Ceiler, S.A. Bidstrup, P. Khol and G. May (1994). Modeling the properties of pecvd silicon dioxide films using optimized back-propagation neural networks. *IEEE Trans. Comp. Pack. and Man. Tech.* **17**, 174–182.

Kim, B. and G.S. May (1994). An optimal neural network process model for plasma etching. *IEEE Trans. on Semiconductor Manufacturing* **7**, 12–21.

Knight, T.J., X. Cheng, B.H. Krogh, D.W. Greve and M.A. Gibson (1995). Application of sampling quadrupole mass spectrometry to multivariable process control. In: *Proc. Symp. on Process Control, Diagnostics, and Modeling in Semiconductor Manufacturing (I)*. Electrochemical Society Spring Meeting. Reno, Nevada. to appear.

Ljung, L. (1987). *System Identification: Theory for the user.* Prentice Hall.

Narendra, K.S. and K. Parthasarathy (1991). Gradient methods for the optimization of dynamical systems containing neural networks. *IEEE Trans. on Neural Networks* pp. 252–262.

Nerrand, O., P. Rousell-Ragot, D. Urbani L. Personnaz and G. Dreyfus (1994). Training recurrent neural networks: Why and how? an illustration in dynamical process modeling. *IEEE Trans. on Neural Networks* **5**, 178–184.

Nerrand, O., P. Rousell-Ragot, L. Personnaz and G. Dreyfus (1993). Neural networks and nonlinear adaptive filtering: Unifying concepts and new algorithms. *Neural Computation* pp. 165–199.

Sachs, E., A. Hu and A. Ingolfsson (1995). Run by run process control: Combining spc and feedback control. *IEEE Trans. on Semiconductor Manufacturing* **8**(1), 26–43.

Williams, R.J. (1992). Training recurrent networks using the extended kalman filter. In: *Proc.IJCNN*. Vol. IV. pp. 241–246.

Ydstie, B.E. (1990). Forecasting and control using adaptive connectionist networks. *Computers Chem. Engng.* **14**, 583–599.