NEURAL NETWORK MODELLING OF A BOILER COMBUSTION SYSTEM

B. Molloy and J.V. Ringwood

Control Systems Group
School of Electronic Engineering
Dublin City University
Glasnevin, Dublin 9

ABSTRACT

Boiler combustion systems represent highly nonlinear systems with associated lags and delays which depend on operating point. Such systems represent a significant challenge to the control engineer. In this paper, we present a model typical of a medium-size industrial boiler, which highlights the difficulties associated with such systems. As a starting point for nonlinear control design, we demonstrate how a neural network may be used to obtain a concise functional description of the system, initially for fixed operating points and then for variation over the full range of operation.

1. INTRODUCTION

Boilers generate steam through the interaction of two processes:- combustion and evaporation. It is a complex, volatile operation which can present significant control problems. These problems are intensified by the high standards of safety and efficiency which are demanded of modern industrial boilers. Producing a comprehensive model for this process is often the first obstacle encountered in developing a reliable control strategy. This paper concentrates on one aspect of the combustion process - determination of the percentage O_2 emitted from the furnace stack from fuel and air valve signals. The combustion process is non-linear with varying dynamics (including pure delay) which depends on the system operating point. The plant is simulated using *SIMULINK*, a software package for simulating dynamical systems [1].

Neural networks offer an interesting alternative to conventional modelling and control methods. There are no restrictions on what can be modelled or controlled as it has been shown that neural networks are capable of approximating arbitrary nonlinear functions [2]. Process models can be quickly generated, as neural networks "learn" to model the system using past data. It is not necessary to study the system in order to construct a model from first principles and no programming is necessary. Neural networks have a parallel structure, which promises speed and fault tolerance. They can be adapted on-line to cater for time-variant processes. Neural networks are MIMO systems. They can accept a variety of inputs simultaneously e.g. qualitative and quantitative inputs. Finally, they have ability to generalise and cope with situations not presented in the training data [3].

2. BOILER COMBUSTION MODEL

The combustion section of a boiler may be represented schematically by Fig.1. This diagram describes the relationship between the $\%O_2$ produced in the stack gasses and the air and fuel inputs to the system.

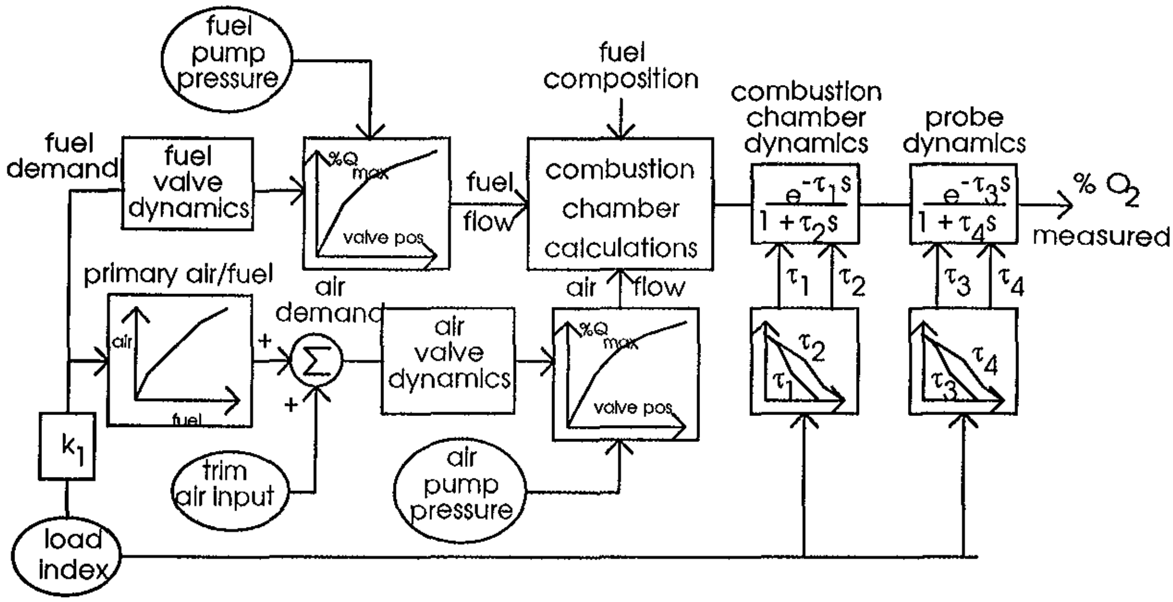

Fig.1 : Boiler model structure

The air and fuel supply subsystems contain valves, the inputs to which are determined by servo. loops as shown in Fig.2. These subsystems contain the dynamics described as "air valve dynamics" and "fuel valve dynamics" in Fig.1, consisting of servomotors which accept "raise" or "lower" type signals enclosed within a feedback loop for positional accuracy.

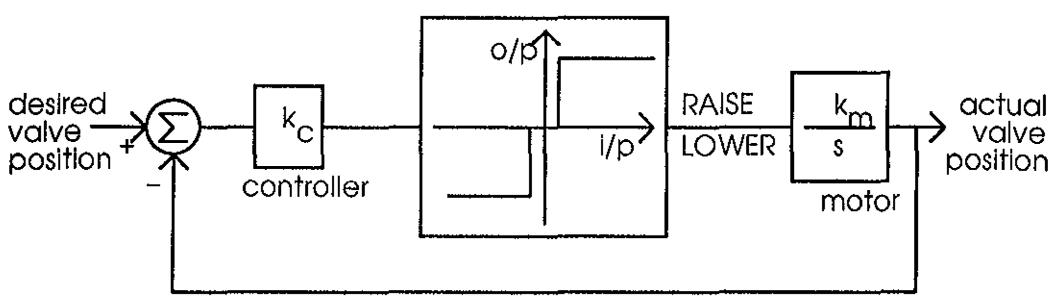

Fig.2 : Fuel and air supply servo. loops

The actual amount of fuel flow and air flow is determined by a combination of the valve positions and the pressure under which fuel and air are supplied. This is further compounded by the combustion chamber pressure, but this effect is not documented for brevity. Overall, the fuel demand is determined by load index, which is a measure of the required firing rate of the boiler, in turn determined by the required boiler heat output. The *primary air/fuel characteristic,* which is normally implemented using a series of mechanical cams, determines the desired air valve position in order to produce a given $\%O_2$ in the stack gasses. This value is determined from a combination of environmental and efficiency considerations. However, due to mechanical wear in the cam arrangement and inaccuracies in the initial setup, combined with a requirement for tighter dynamic control of $\%O_2$, an extra control input, termed "trim air input" in Fig.1, is included.

For the control engineer, therefore, the important input/output relationship is that between trim air input and measured $\%O_2$. This relationship is complex, due to a significant number of nonlinearities (valve characteristics, combustion equations, etc.) and the dynamics due to the combustion chamber and probe, which depend on operating point. The values for the lags and delays in the dynamical blocks generally decrease with load index (throughput). Note that the "combustion calculations" are non-dynamical [4], since instantaneous mixing of air and fuel is assumed. In summary, three factors present a significant challenge to the control engineer:

- The system contains a number of static nonlinearities
- Measurements are subject to pure time delay
- The dynamics of the system vary with operating point.

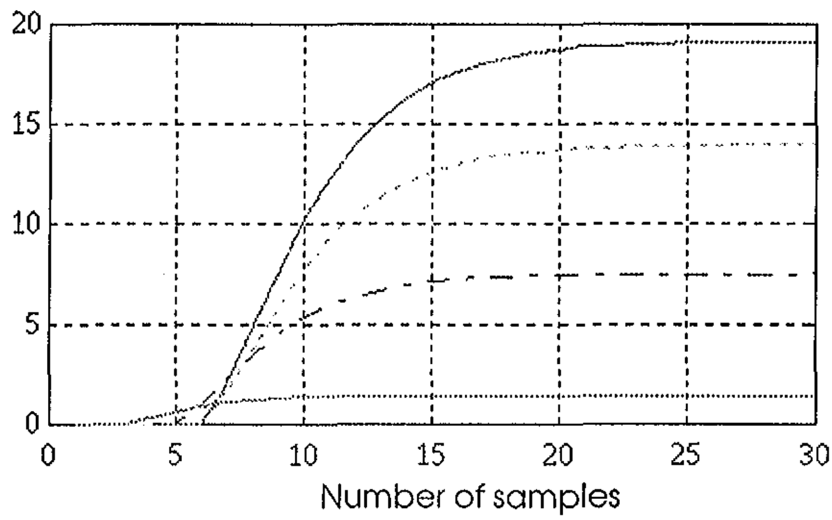

Fig.3 : Response of $\%O_2$ to a step in trim air input for different load indices

Fig.3 above shows the response in $\%O_2$ to a 3% step in trim air input for load indices of 1%, 10%, 30%. 60% and 90%, indicating the variation in dynamics and d.c. gain with operating point.

3. BOILER MODELLING USING NEURAL NETWORKS

This problem was approached in two stages. The first stage was to produce a neural network model of the plant, with load index maintained at a constant level. The second stage was to produce a neural network model of the plant with varying load index. For the first stage, the plant contains static nonlinearities and fixed dynamics. In the second stage, the plant has nonlinearities plus dynamics which are dependant on load index. The neural networks were built and trained using algorithms from the "*Neural Network Toolbox*" [5]. This is a neural network development tool which offers a variety of network structures and training algorithms.

A recurrent multilayer perceptron structure was used for both models. The first model required a two layer network with six tan-sigmoid neurons in the hidden layer and one linear neruon in the output layer. Using tan-sigmoid neurons in the hidden layer, gives the neural network it's required nonlinear characteristics. The linear neuron, used in the output layer, allows the network output to exceed the range [-1,+1]. The actual number of layers and neurons required was determined on a trial and error basis. Increasing the number of neurons beyond the required number, does not improve the network's accuracy significantly and may reduce the network's ability to generalise. Using a recurrent structure increases the network's power and reduces the number of neurons required to approximate a function.

Dynamics are incorporated in the network by passing the trim air demand signal and the previous network output through a tapped delay line. Each stage of the tapped delay line is then used as a network input. The size of each tapped delay line is determined by the order of the plant. The network structure is illustrated in Fig. 4.

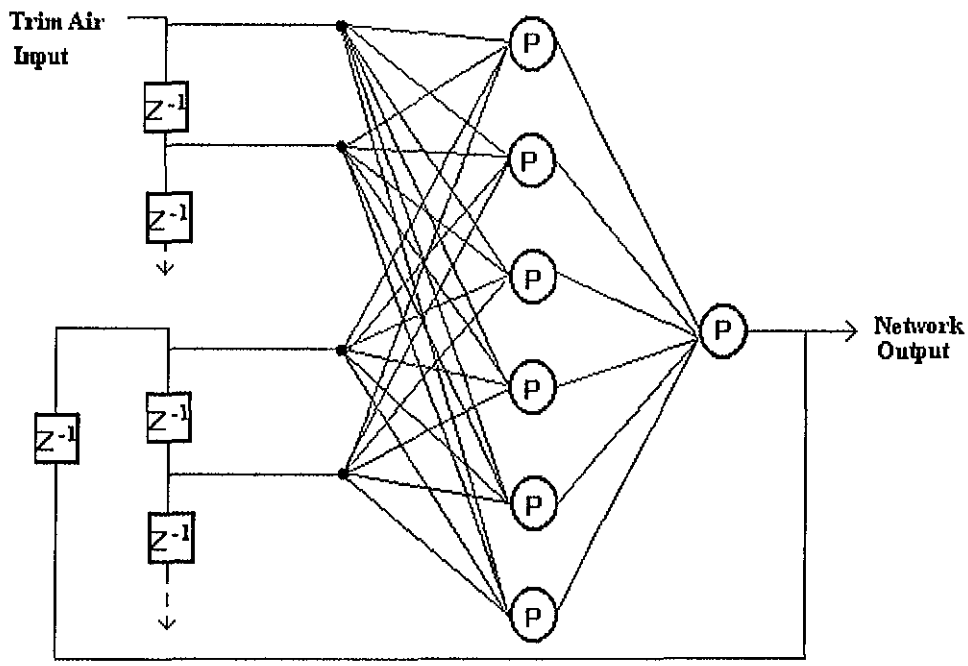

Fig.4 : <u>Structure of neural network for fixed load index</u>

An initial attempt was made to train the network using the basic backpropogation algorithm with a fixed adaptation rate [6]. This adaptation rate must be very carefully chosen. Too large an adaptation rate may cause instability in the weight update, while too small an adaptation rate results in extremely long training times. A more sophisticated version of the backpropogation algorithm, with an adaptive learning rate was then used very successfully. The adaptation gain is increased if weight adaptation has resulted in a decreased error. The adaptation gain is decreased if weight adaptation has resulted in an increased error and the new weights are discarded. In effect this produces a very small adaptation gain which ensures that the network consistently converges. Learning with momentum was also used to speed up training. Momentum allows the network to follow trends in the error surface and slide over small local minima. It is achieved by incorporating a percentage of the previous weight adaptation to the current weight adaptation.

The weights are adapted as follows [7]:

$$w_{ij}(t+1) = w_{ij}(t) + \eta \delta_j x_j' + \alpha(w_{ij}(t) - w_{ij}(t) - w_{ij}(t) - w_{ij}(t-1)$$

where $w_{ij}(t)$ is the weight from hidden neuron i or from an input to neuron j. x_j' is the output of neuron j or an input. η is the adaptation gain, α is the momentum term $(0 < \alpha < 1)$ and δ_j is an error term for neuron j. If neuron j is an output neuron, then

$$\delta_j = y_j(1 - y_j)(d_j - y_j),$$

where d_j is the desired neuron output and y_j is the actual neuron output. If neuron j is an internal hidden neuron, then

$$\delta_j = x_j'(1 - x_j') \sum_k \delta_k w_{jk},$$

where k is over all neurons in the layers above neuron j. All inputs were scaled to between [-1,+1]. This improves numerical conditioning and speeds up training. The training set was presented in batch to the network. Approximately 100 data points were included in the training set per hidden layer neuron, based on a heuristic guideline [8]. The sum squared error over the complete training set dropped below the target of 0.01 after approximately 600 training iterations.

A square-wave-type signal of varying amplitude and frequency was applied to the trim air demand input to generate the training set. A variable switching interval is employed to excite the plant over the complete frequency range. Varying the signal amplitude provides the neural network with information about the plant static nonlinearities. A typical trim air input sequence is shown in Fig.5.

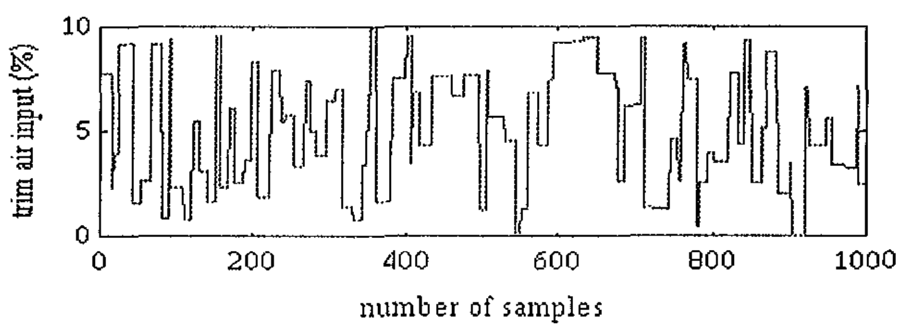

Fig.5 Trim air input sequence

A second neural network was then used to model the plant with varying load index. This time a three layer neural network was utilised. The first hidden layer contains 20 tan-sigmoid neurons.

The second hidden layer contains 80 tan-sigmoid neurons. The output layer consists of one linear neuron. Load index is included as a network input. Load index is a static input, so it is not necessary to provide the network with previous values of this variable. A square-wave-like signal of varying amplitude and frequency was applied as load index. The minimum switching intervals must always be longer than the maximum plant delay. This enables the network to observe plant dynamics. A sample input sequence is shown in Fig. 6.

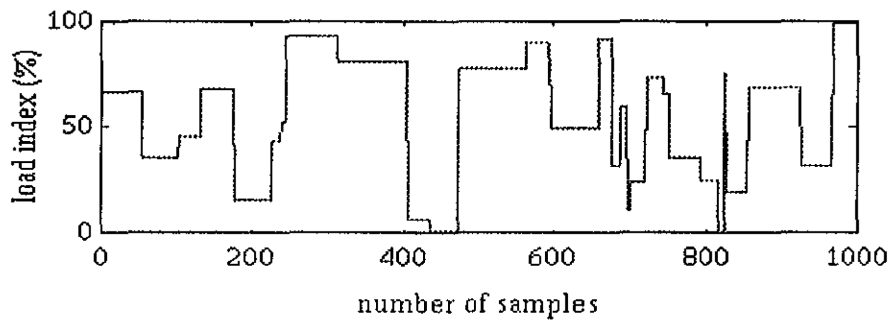

Fig.6 :Load index input sequences

A larger training set was used, for this larger network. Fig. 7 shows how the sum squared error reduces during training and the variation in the adaptation gain.

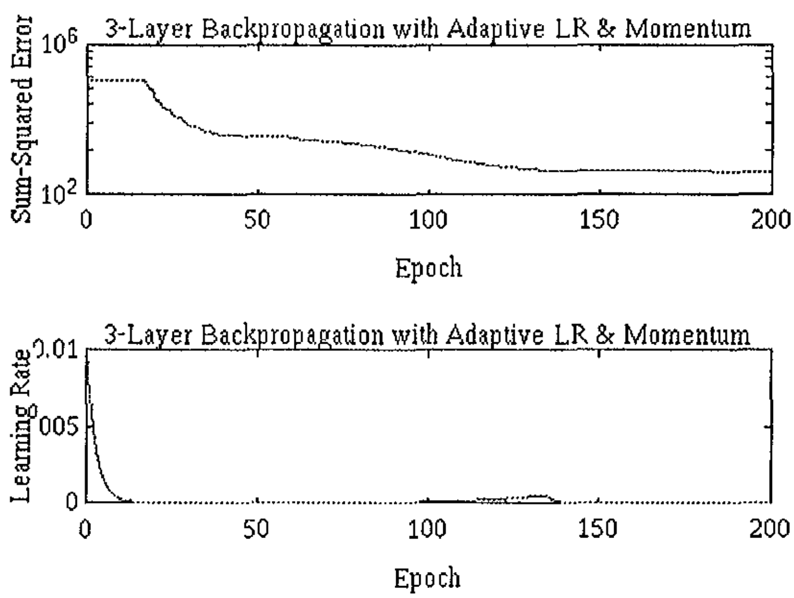

Fig.7 :Variation in sum squared error and adaptive learning rate during training

Plots of network output versus plant output for both fixed and varying load index are given in Figs. 8 and 9 respectively. In both cases, the network output closely follows the actual plant output.

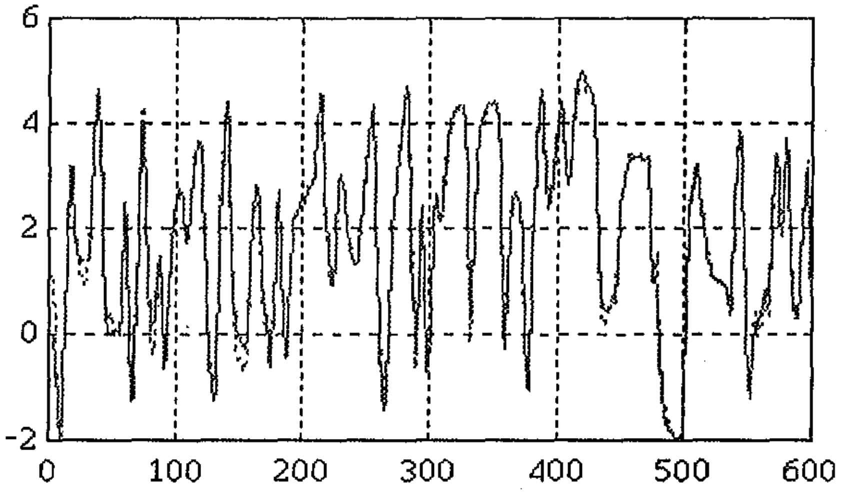

Fig.8 : Plant versus network output for fixed load index

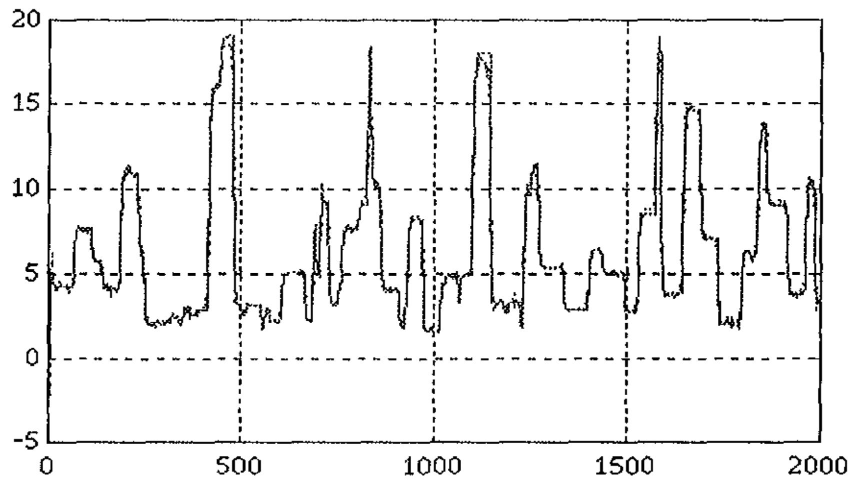

Fig.9 : Plant versus network output for varying load index

4. CONCLUSIONS

The results show that neural networks are capable of producing an excellent model of a nonlinear process with varying dynamics. In theory these models can be generated quickly (within a matter of hours) and do not require detailed knowledge of the system processes. However a certain amount of system knowledge is necessary to construct a neural network; e.g. which inputs are dynamical, an approximate estimate of the order of the system and the maximum system delay. A good understanding of the system is also vital in order to generate a training set which adequately covers the complete operating range of the plant. The negative aspect of not requiring extensive plant knowledge is that neural networks provide very little information about the underlying plant processes. It is a black-box technique with the associated disadvantages.

Neural networks do not always train consistently. There are a large number of factors affecting network convergence e.g. initial weights used, size of adaptation gain, scaling of inputs, etc.

Training neural networks as controllers presents a further difficulty. Unlike boiler modelling where the actual plant output is available as a desired reference output for network training, no such desired reference output is available at the controller output. A number of methods have been proposed to overcome this problem. Most of these methods employ a model reference adaptive control strategy. One method involves backpropagating the error (between the output of the reference model and the plant output) through a neural network model of the plant [9]. This backpropogated error is then used as the error at the controller output. A second method uses a neural network inverse plant model to determine the error at the controller output [10]. A third method involves placing a model of the plant before the controller during training and using the output of the reference model as the desired controller output [11]. Using these methods, attempts have been made to train a neural-network controller in the current application, so far with limited success.

Despite these problems, neural networks have much potential in the area of nonlinear modelling and control.

REFERENCES

1. *SIMULINK User's Guide,* The MATHWORKS Inc., March 1992.
2. Narenda, K.S. and Parthasarathy, K., *Identification and Control of Dynamical Systems using Neural Networks,* IEEE Trans. on Neural Networks, Vol. 1, No. 1 pp 4-27, March 1990.
3. Hunt, K.J., Sbarbaro, D., Zbikowski, R. and Gawthrop, P.J., *Neural Networks for Control Systems - A Survey,* Automatica, Vol 28, No. 6. pp 1083-112,1992.
4. Ringwood, J.V., *Boiler Model - Final Report,* Report No. NIHEDWES3.1286, Dublin City University, Dec. 1986.
5. Demuth, H. and Beale, M., *Neural Network Toolbox User's Guide,* The MATHWORKS Inc., June 1992.
6. Rumelhart, D.E., Hinton, G.E. and Williams, R.J., *Learning internal representations by error propogation,* in Parallel Distributed Processing, Vol.1 (Eds. D.E. Rummelhart and J.L. McClelland), MIT Press, 1986.
7. Lippmann, R.P., *An Introduction to Computing with Neural Nets,* IEEE ASSP Magazine, pp 4-22, April 1987
8. Hall, C. and Smith R., *Pitfalls in the Application of Neural Networks for Process Control,* in Neural Networks for Control and Systems (Eds. K.Warwick, G.W. Irwin and K.J. Hunt), IEE Control Engineering Series 46, Peter Peregrinus Ltd., 1992.
9. Hunt, K.J. and Sbarbaro, D.J., *Studies in Neural Network Based Control,* in Neural Networks for Control and Systems (Eds. K.Warwick, G.W. Irwin and K.J. Hunt), IEE Control Engineering Series 46, Peter Peregrinus Ltd., 1992.
10. Hoskins, D.A., Hwang, J.N. and Vagners, J., *Iterative inversion of neural networks and its application to adaptive control,* IEEE Trans. Neural Nets., Vol.3, No.2, March 1992.
11. Widrow, B. and Stearns, S.D., *Adaptive Signal Processing,* Prentice-Hall, 1987.