Neural network system identification for improved noise rejection

DAVID C. HYLAND†, EMMANUELG. COLLINS, JR‡, WASSIMM. HADDAD§ and DOUGLASL. HUNTER¶

Neural networks are able to approximate a large class of input–output maps and are also attractive due to their parallel structure which can lead to numerically inexpensive weight update laws. These properties make neural networks a viable paradigm for adaptive system identification and control, and as a consequence the use of neural networks for identification and control has become an active area of research. This paper contributes to this research thrust by developing adaptive neural identification algorithms that are able to minimize the influences of extrinsic noise on the quality of the identified model. The development relies on the use of a batch ARMarkov model, a generalization of an ARMA model whose parameters include some of the Markov parameters of the system and whose output contains the system outputs at previous sample instants. Through both theoretical analyses and simulation results, this paper demonstrates the ability of the neural network predicated on a batch ARMarkov model to improve on the noise rejection properties of identification, based on either an ARMA model or a CARMA model developed by Watanabe et al. Although the focus here is on linear system identification, the paper lays a foundation for adaptive, nonlinear identification and control.

1. Introduction

A neural network consists of a weighted interconnection of fundamental elements called neurons, which are basically functions consisting of a summing junction and a nonlinear operation or activation function \( g(\cdot) \). An overview of the field of neural networks, including its history, various architectures and applications, was compiled by Hecht-Nielsen (1990). One of the primary reasons for the large interest in neural networks is their capacity to approximate a large class of input–output maps. In addition, neural networks have attracted attention due to their inherently parallel structure that makes it possible to develop parallel weight update laws. This parallelism makes it possible to effectively update a neural network online, that is, adaptively.

The above properties make neural networks a viable paradigm for adaptive system identification and control as demonstrated in the work of Narendra and Parthasarathy (1990, 1991), Nguyen and Widrow (1991), Hunt et al. (1992), and the work presented in the IEEE Control Systems Magazine (1989, 1990). This paper
focuses on the system identification of linear systems with output signals corrupted by sensor noise, and it develops adaptive neural algorithms that are able to minimize the influences of the extrinsic sensor corruptions. These noise rejection properties are based on developing neural networks derived from the batch ARMarkov model of the plant developed by Juang et al. (1991), Hyland (1991, 1993), and Phan et al. (1993). A batch ARMarkov model is an infinite impulse response (IIR) model, characterized by two positive integer parameters $L$ and $R$ and has the following properties: $L$ of its parameters are the first $L$ Markov parameters of the system; and its output vector includes the system measurement vector at the current sample time (denoted by $k$) and also the measurement vectors at the previous time samples, $k - 1, \ldots, k - (R - 1)$. For $L = 0$ and $R = 1$ the batch ARMarkov model is reduced to the well known ARMA (auto-regressive moving average) model. However, the choice of $L > 0$ and/or $R > 1$ improves the noise rejection properties that are obtained by the use of a simple ARMA model. In fact, as $L \to \infty$ and/or $R \to \infty$, it is possible to show (subject to certain assumptions) that the adaptive scheme is capable of identifying the system with arbitrary accuracy.

Causality in a batch ARMarkov model is enforced by forcing the weighting matrices containing the synapses to have a Toeplitz structure. Here, we develop algorithms with and without this constraint. The enforcement of the Toeplitz structure leads to adaptive laws with reduced parallelism but with faster convergence properties due to the decrease in the number of parameters that must be updated.

As this paper focuses on linear system identification, the final adaptive laws are based on choosing the activation functions in the neural network to be (linear) identity operators. However, these results are developed using two general update theorems (Theorems 1 and 2) that allow the incorporation of nonlinear activation functions. Hence, this paper lays the theoretical foundation for extending the main results to nonlinear system identification.

Backpropagation laws are usually developed for feedforward networks. Theorem 1 is essentially a generalization of standard backpropagation gradient computations to general interconnections of neurons that can include feedback terms. Theorem 2 adds the additional restriction of a Toeplitz constraint on the weighting matrix. The inclusion of feedback interconnections allows the neural network to represent a feedback system. Thus, this paper also lays the theoretical foundation for adaptive neural control systems.

It should also be noted that, in contrast to most neural network results, Theorems 1 and 2 are developed using an array of neurons, a dynamic ganglia, as the basic building block. This viewpoint allows the proofs of the theorems to be based on matrix algebra, which we believe to be more efficient than standard backpropagation proofs.

Section 2 of this paper describes the dynamic ganglion, the general interconnection of $N$ ganglia, and Toeplitz structure restrictions on the weighting matrices. Gradients are derived for the adaptation of the weighting matrices with and without the Toeplitz constraints. Section 3 develops the Batch ARMarkov replicator network based on the information presented in §2.

The noise rejection properties of the replicator network are discussed in §4. Definition of the mean and fluctuation model errors, and the capacity of decreasing them in the presence of sensor noise, is presented in theorems for networks with and without Toeplitz constraints. Demonstration of the noise rejection properties in the presence of both white and coloured sensor noise is presented in §5. The batch
ARMarkov network is compared with the ARMA model and the CARMA model developed by Watanabe et al. (1992). Changes in the parametrization of the batch ARMarkov model demonstrate the capacity to decrease model error with an increase in the proper network parameter.

2. Network architecture

The dynamic ganglion is an array of neurons processing input signals into an output signal with both forward (signal processing) and backward (error propagation) paths. A collection of $N$ dynamic ganglia may be generally interconnected, with the input signal to any ganglion being a combination of the output signals of any or all other ganglia, to form the network necessary for system modelling and control, although the focus of this paper is limited to system modelling. In addition, note that due to the feedback paths in the interconnections of the neurons in one ganglia with another ganglia, this type of network is a special case of a recurrent network (Fausett 1994). Its relationship with other recurrent networks remains to be explored.

Figure 1 illustrates the $i$th ganglion in a general interconnected network. The inputs to the forward path are the training input vectors, $\bar{\xi}_i \in \mathbb{R}^{p_i}$, and the forward outputs of the $N$ ganglia, $\bar{y}_j \in \mathbb{R}^{p_j}$, $j = 1, \ldots, N$. The output of the forward path, $\bar{y}_i \in \mathbb{R}^{p_i}$, is given by

$$\bar{y}_i = \mathcal{G}(\bar{u}_i)$$

where

$$\bar{u}_i = \sum_{j=1}^{N} W(i,j) \bar{y}_j + \Theta_i \bar{\xi}_j$$

$\mathcal{G}(\cdot)$ is the neural activation function (such as the sigmoid function), $W(i,j) \in \mathbb{R}^{p_i \times p_j}$ are (possibly Toeplitz) weighting matrices, and

$$\Theta_i = \begin{cases} 1, & \text{if the } i \text{th ganglion is an input ganglion} \\ 0, & \text{otherwise} \end{cases}$$

The inputs to the backward path of the $i$th ganglion are the training error vectors $\bar{e}_i$

![Figure 1. The $i$th ganglion in a general interconnection.](image-url)
and the backward output vectors of the $N$ ganglia, $\vec{y}^*_j, j = 1, \ldots, N$. The backward path output of the $i$th ganglion $\vec{y}_i$ is given by

$$
\vec{y}^*_i = \tilde{g}(\vec{u}_i)\left(\sum_{j=1}^{N} W^{(j,i)^T} \vec{y}^*_j + \Omega_i \vec{e}_i\right)
$$

(4)

where

$$
\tilde{g}(\vec{u}_i) = \frac{d\tilde{g}}{d\vec{u}_i}
$$

and

$$
\Omega_i = \begin{cases} 
1, & \text{if the } i\text{th ganglion is an output ganglion} \\
0, & \text{otherwise}
\end{cases}
$$

(5)

Keeping the internal structure of a dynamic ganglion in mind, Fig. 2 shows the block diagram convention for the $i$th ganglion of Fig. 1.

The total mean-squared error of the general interconnection of $N$ ganglia is given by

$$
\varepsilon^2 = \sum_{i=1}^{N} \Omega_i \|\vec{e}_i\|^2
$$

(6)

where

$$
\vec{e}_i = \overline{\eta}_i - \vec{y}_i
$$

(7)

and where $\overline{\eta}$ is the desired output of the $i$th ganglion. In the general interconnection case, each ganglion may send and receive forward and backward path signals from each of the $N$ ganglia as well as with outside sources. Adaptive updates of the weight matrices connecting the ganglia may be accomplished using a gradient search technique requiring the gradient of the total mean-squared error with respect to the weight matrices, as presented in the following theorem.

**Theorem 1:** Consider the general interconnection of $N$ ganglia with the $i$th ganglion as shown in Fig. 1. Then the negative gradient of the total mean-squared error with respect to the weight matrix $W^{(i,j)}$ is given by

$$
- \frac{\partial}{\partial W^{(i,j)}} \left(\frac{1}{2} \varepsilon^2\right) = \vec{y}^*_i \vec{y}^T_j
$$

(8)
Proof: For the proof, see the Appendix.

When dealing with time sequences and tapped delay lines, it is customary only to use information from the ‘past’ when performing calculations in the ‘present’. Restricting the weighting matrices to Toeplitz structures will enforce this causality in the modelling network of the next section. Specifically, we assume that weight matrix \( W^{(i,j)} \) has a Toeplitz structure given by

\[
W^{(i,j)} = \begin{bmatrix}
    w^{(i,j)}_{1} & w^{(i,j)}_{2} & \cdots & w^{(i,j)}_{p} & 0 & \cdots & 0 & 0 \\
    0 & w^{(i,j)}_{1} & \cdots & w^{(i,j)}_{p-1} & w^{(i,j)}_{p} & \cdots & 0 & 0 \\
    \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\
    0 & 0 & \cdots & \cdots & \cdots & \cdots & w^{(i,j)}_{p} & w^{(i,j)}_{p} \\
\end{bmatrix}
\]

and hence \( W^{(i,j)} \) is completely represented by

\[
W^{(i,j)}_{T} = \begin{bmatrix}
    W^{(i,j)}_{1} & W^{(i,j)}_{2} & \cdots & W^{(i,j)}_{p} \\
\end{bmatrix}
\]

The next theorem provides the gradient of the total mean-squared error subject to this Toeplitz constraint.

**Theorem 2:** Consider the general interconnection of \( N \) ganglia with the \( i \)th ganglion, as shown in Fig. 1. If the weighting matrices are restricted to a Toeplitz structure, then the negative gradient of the total mean-squared error with respect to the weight matrix \( W^{(i,j)}_{T} \) is given by

\[
- \frac{\partial}{\partial W^{(i,j)}_{T}} \left( \frac{1}{2} \varepsilon^2 \right) = \begin{bmatrix}
    y^{*}_{i,1} & \cdots & y^{*}_{i,p} \\
    \vdots & \ddots & \vdots \\
    0 & \cdots & y^{*}_{i,p} \\
\end{bmatrix}
\]

Proof: For the proof, see the Appendix.

3. **Batch ARMarkov predictor network**

In this section we consider a class of linear, discrete-time, single-input/single-output (SISO) systems and consider a sequence of model forms, denoted here as: IR (impulse response), ARMA (auto-regressive moving average), ARMarkov (i.e. ARMA + Markov), and batch ARMarkov. The ARMarkov and batch ARMarkov models were originated by Juang et al. (1991) and Hyland (1993), and as described below the batch ARMarkov model is represented in terms of Toeplitz matrices. The first three forms are special cases of the batch ARMarkov model.

An IR model has the form

\[
y(k) = \sum_{i=1}^{M} H_{i} u(k - i)
\]

where \( H_{i}, i = 1, \ldots, M, \) are the Markov parameters from \( u(k) \) to \( y(k) \). \( M < \infty \) corresponds to a finite impulse response (FIR) model and \( M = \infty \) corresponds to an infinite impulse response (IIR) model.
An ARMA model has the form
\[ y(k) = \sum_{i=1}^{M} A_i y(k - i) + \sum_{i=1}^{M} B_i u(k - i) \] (13)
where \( A_i, B_i, i = 1, \ldots, M \), are the model coefficients.

An ARMarkov model has the form
\[ y(k) = \sum_{i=1}^{M} G_i y(k - L - i) + \sum_{i=1}^{M+L} J_i u(k - i), \quad L = 0, 1, \ldots \] (14)
where the first \( LJ_i \)'s are the Markov parameters \( H_1, \ldots, H_L \), and \( G_i, i = 1, \ldots, M \), and \( J_i, i = L + 1, \ldots, M + L \), are the model coefficients.

A batch ARMarkov model has the form
\[ \bar{y}(k) = P \bar{y}(k - 1) + Q \bar{u}(k - 1) \] (15)
where
\[ \bar{y}(k) = \begin{bmatrix} y(k) \\ \vdots \\ y(k - R) \end{bmatrix}, \quad R = 1, 2, \ldots \] (16)
\[ \bar{y}(k - 1) = \begin{bmatrix} y(k - L - 1) \\ \vdots \\ y(k - L - M - R + 1) \end{bmatrix} \] (17)
\[ \bar{u}(k - 1) = \begin{bmatrix} u(k - 1) \\ \vdots \\ u(k - L - M - R + 1) \end{bmatrix} \] (18)
\[ P \in \mathbb{R}^{(R) \times (M + R - 1)} \] (19)
\[ Q \in \mathbb{R}^{(R) \times (L + M + R - 1)} \] (20)

For (15) to correspond to a causal system, \( P \) and \( Q \) must have Toeplitz structures. Specifically, \( P = \text{Toep}(P) \) and \( Q = \text{Toep}(Q) \), where for \( S \in \mathbb{R}^{p \times p} \) \( \text{Toep}(S) \) is given by
\[ \text{Toep}(S) = \begin{bmatrix} s_{1,1} & s_{1,2} & \cdots & s_{1,p} & 0 & \cdots & 0 \\ 0 & s_{1,1} & \cdots & s_{1,p-1} & s_{1,p} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & s_{1,1} & s_{1,2} & \cdots & s_{1,p} \end{bmatrix} \in \mathbb{R}^{p \times p} \] (21)

The batch ARMarkov predictor network is a neural network architecture based on the batch ARMarkov model. A block diagram of the SISO representation of the batch ARMarkov predictor network is shown in Fig. 3. \( W^{(3,1)} \) and \( W^{(3,2)} \) correspond, respectively, to \( P \) and \( Q \) in (15). This network makes use of tapped delay lines shown as
\[ TD L \]
\[ (L) \] (22)
which map the scalar input $\xi(k)$ to the vector output

$$\bar{\xi}(k) = \begin{bmatrix} \xi(k) \\ \vdots \\ \xi(k-L+1) \end{bmatrix} \in \mathbb{R}^L$$

Hence, the tapped delay line is a memory unit which keeps past signals for use with present data. Furthermore, time delay units shown as

$$\Delta^L$$

allow inputs $\bar{\xi}(k)$ to be stored and release outputs $\bar{\xi}(k-L)$ to make use of distant past signals without dealing with the recent past.

The replicator network shown in Fig. 3 works in series–parallel with the plant to be identified. The training input signal $\bar{\xi}(k-1)$ is shared by the plant and the network alike. The output of the plant $\bar{y}_p(k)$ is compared with the output of the replicator network $\bar{y}(k)$ to generate an error signal for adaptation of the weighting matrices connecting the dynamic ganglia. The synapses connecting the three dynamic ganglia with block diagram conventions as in Fig. 2, if required to be Toeplitz, enforce causality in the system as well as reducing the number of parameters to be adapted. Furthermore, note that linear activation functions are used in each of the dynamic ganglia.

The structure of the replicator network is homologous to the structure of an associated form model. It is possible to generate FIR, ARMA and ARMarkov in addition to batch ARMarkov models using the basic building blocks of the network. FIR models are generated by constraining $W^{(3,1)} = 0$ and choosing $R = 1$. ARMA models correspond to $R = 1$ and $L = 0$, and ARMarkov models correspond to $R = 1$. 

Figure 3. Batch ARMarkov replicator network.
For notational convenience, let the ganglion with input $\xi$ be referred to as ganglion 1, let the ganglion with input $\zeta$ be referred to as ganglion 2, and let the ganglion with an input from each of the other ganglia be referred to as ganglion 3. The three dynamic ganglia have forward pass output signals given by

$$y_1 = \xi = \begin{bmatrix} y_p(k - L - 1) \\ \vdots \\ y_p(k - L - M - R + 1) \end{bmatrix}$$

$$y_2 = \zeta = \begin{bmatrix} \xi(k - 1) \\ \vdots \\ \xi(k - L - M - R + 1) \end{bmatrix}$$

$$y_3 = W^{(3,1)} y_1 + W^{(3,2)} y_2 + N_d(k)$$

(25)  

(26)  

(27)

where $\xi = N_d(k)$ represents a sensor noise signal. Combining the forward pass output signals yields

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ W^{(3,1)} & W^{(3,2)} & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

(28)

The output signal of the network $y_3$ is compared with the output signal of the plant $y_p$ to form the output error signal

$$e_3 = y_p - y_3$$

(29)

which is backpropagated through the network to adapt the weight matrices. The backward pass outputs of the dynamic ganglia are

$$y_1^* = W^{(3,1)^T} y_3^*$$

$$y_2^* = W^{(3,2)^T} y_3^*$$

$$y_3^* = e_3$$

(30)  

(31)  

(32)

or equivalently

$$\begin{bmatrix} y_1^* \\ y_2^* \\ y_3^* \end{bmatrix} = \begin{bmatrix} 0 & W^{(3,1)^T} & 0 \\ 0 & W^{(3,2)^T} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_1^* \\ y_2^* \\ y_3^* \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ e_3 \end{bmatrix}$$

(33)

Using the forward and backward pass information for the three dynamic ganglia, it is possible to develop weight updates for the adaptation of the weight matrices $W^{(3,1)}$ and $W^{(3,2)}$. Theorem 3 gives the gradients needed for this adaptation.

**Theorem 3:** Consider the ARMarkov replicator network shown in Fig. 3 with the forward and backward pass signals given by (28) and (33). Then

$$-\frac{\partial}{\partial W^{(3,1)}} \left( \frac{1}{2} e^2 \right) = y_3^* y_1^T$$

$$-\frac{\partial}{\partial W^{(3,2)}} \left( \frac{1}{2} e^2 \right) = y_3^* y_2^T$$

(34)  

(35)
Proof: The result follows as a direct consequence of Theorem 1.

The next theorem provides the gradient expressions in the case that $W^{(3,1)}$ and $W^{(3,2)}$ are constrained to Toeplitz structures, and it requires the definition of the two Hankel matrices

$$H = \begin{bmatrix} y_{1d}(k-L) & \cdots & y_{1d}(k) \\ y_{1d}(k+1) & \cdots & y_{1d}(k+M) \\ \vdots & \ddots & \vdots \\ y_{1d}(L) & \cdots & y_{1d}(L+M) \\ y_{1d}(L+1) & \cdots & y_{1d}(L+M+1) \\ \vdots & \ddots & \vdots \\ y_{1d}(R-1) & \cdots & y_{1d}(R+M-1) \\ y_{1d}(R) & \cdots & y_{1d}(R+M) \\ y_{1d}(R+1) & \cdots & y_{1d}(R+M+1) \end{bmatrix}$$

$$H = \begin{bmatrix} y_{2d}(k) & \cdots & y_{2d}(k-L-1) \\ y_{2d}(k-L) & \cdots & y_{2d}(k-L-M) \\ \vdots & \ddots & \vdots \\ y_{2d}(k-L-M-1) & \cdots & y_{2d}(k-L-M-R) \\ y_{2d}(k-L-M-R+1) & \cdots & y_{2d}(k-L-M-R+2) \\ \vdots & \ddots & \vdots \\ y_{2d}(L+R) & \cdots & y_{2d}(L+R+M-1) \\ y_{2d}(L+R+1) & \cdots & y_{2d}(L+R+M) \\ y_{2d}(L+R+2) & \cdots & y_{2d}(L+R+M+1) \end{bmatrix}$$

where the vectors $y_1$ and $y_2$ are defined in (25) and (26).

Theorem 4: Consider the ARMarkov replicator network shown in Fig. 3 with the forward and backward pass signals given by (28) and (33) where $W^{(3,1)}$ and $W^{(3,2)}$ are Toeplitz. Then

$$- \frac{\partial}{\partial W^{(3,1)}} \left( \frac{1}{2} \epsilon^2 \right) = H^T \begin{bmatrix} y_{1d}(k-L) \\ \vdots \\ y_{1d}(k+M) \\ \vdots \\ y_{1d}(L+M) \\ \vdots \\ y_{1d}(R+M) \end{bmatrix} \quad (38)$$

$$- \frac{\partial}{\partial W^{(3,2)}} \left( \frac{1}{2} \epsilon^2 \right) = H^T \begin{bmatrix} y_{2d}(k) \\ \vdots \\ y_{2d}(k-L-1) \\ \vdots \\ y_{2d}(k-L-M-1) \\ \vdots \\ y_{2d}(k-L-M-R) \\ \vdots \\ y_{2d}(k-L-M-R+1) \end{bmatrix} \quad (39)$$

Proof: The result follows from Theorem 2. For details, see the Appendix.

To perform gradient descent adaptation, a step length parameter is needed along with the above gradients. An adaptive step parameter $\hat{\mu}_p(k)$ was chosen to ensure convergence of the adaptation algorithm. The adaptive step parameter is defined as

$$\hat{\mu}_p(k) = \frac{\alpha}{\|\tilde{\xi}(k)\|^2 + \|\tilde{\xi}(k)\|^2 + \|\tilde{\xi}(k)\|^2}$$

(40)

where $\alpha \in [0, 1]$ is a learning rate constant. We also define the recent and distant past input vectors as

$$\tilde{\xi}(k) = \begin{bmatrix} \xi(k) \\ \vdots \\ \xi(k-L) \end{bmatrix} \quad \text{and} \quad \tilde{\xi}(k) = \begin{bmatrix} \xi(k-L-1) \\ \vdots \\ \xi(k-L-M-R+1) \end{bmatrix} \quad (41)$$

4. Noise rejection properties of the network

A consequence of the adaptation of weighting matrices in the batch ARMarkov replicator network is the identification of dynamic systems with arbitrarily small steady-state errors in the presence of sensor measurements corrupted by noise. The output error is caused by the model weight error, that is the difference between the current weighting matrices and the ideal weighting matrices. The closer that the model weight error is to zero, the more closely the network replicates the plant at any
given time. Deviations from zero are caused by bias and fluctuation weight errors which will be discussed in the following subsections.

4.1. Noise rejection with unconstrained weighting structures

Consider the batch ARMarkov network with unconstrained synapses defined in (15)–(20). By corrupting the output signal \( \tilde{y}_p(k) \) with the noise signal \( \tilde{N}(k) \) the plant output can be expressed as

\[
\tilde{y}_p(k) = \tilde{N}(k) + \begin{bmatrix} 0_{R \times L} \end{bmatrix} W^{(3,1)*} \begin{bmatrix} \tilde{o}_L \end{bmatrix} + W^{(3,2)*} \begin{bmatrix} \tilde{y}_p(k - 1) \end{bmatrix}
\]

where \( W^{(3,1)*} \) and \( W^{(3,2)*} \) are the ideal weighting matrices. The corresponding network output can also be expressed as

\[
\tilde{y}(k) = \begin{bmatrix} 0_{R \times L} \end{bmatrix} W^{(3,1)*} \begin{bmatrix} \tilde{o}_L \end{bmatrix} + W^{(3,2)*} \begin{bmatrix} \tilde{y}_p(k - 1) \end{bmatrix}
\]

By incorporating the output equations (42) and (43) into the output error equation, applying the weight update law, and defining the model weight error as

\[
\omega(k) = \left[ W^{(3,2)*}(k) - W^{(3,2)*}, 0_{R \times L}, W^{(3,1)*}(k) - W^{(3,1)*} \right]
\]

results in the weight error evolution equation

\[
\omega(k + 1) = \omega(k) \left[ I - \hat{\mu}_p(k) \hat{\Delta}(k) \right] + \hat{\mu}_p(k) \hat{C}(k)
\]

where

\[
\hat{\Delta}(k) = \begin{bmatrix} \tilde{N}(k) \begin{bmatrix} \tilde{y}_p(k - 1) \end{bmatrix} \quad \tilde{N}(k) \begin{bmatrix} \tilde{y}_p(k - 1) \end{bmatrix} \quad \tilde{N}(k) \begin{bmatrix} \tilde{y}_p(k - 1) \end{bmatrix} \end{bmatrix}
\]

The derivation of (45) is given in the Appendix.

The closer \( \omega(k) \) is to zero, the more closely the network replicates the plant at any given time. Deviations from zero are caused by bias and fluctuation errors. The mean weight error is defined as the estimated value of the weight matrix, i.e. \( \tilde{\omega}(k) = \mathbb{E}[\omega(k)] \) where \( \mathbb{E} \) denotes expectation. The bias error \( \tilde{\omega}_B \) is the steady-state mean error, i.e.

\[
\tilde{\omega}_B = \lim_{k \to \infty} \left[ \tilde{\omega}(k) \right]
\]

Using (45), it is possible to determine the steady-state mean weight error. Furthermore, it is possible to conclude that an increase in the ‘distant time’ parameter \( L \) of the network causes this bias error to be minimized.

**Theorem 5:** Assume that the plant to be identified is both observable and controllable from the input signal \( \tilde{z}(k) \) and noise signal \( \tilde{N}(k) \). Furthermore, assume that the evolution of the weight error of this plant is described by (45). Then, the mean weight error evolves as

\[
\tilde{\omega}(k + 1) = \tilde{\omega}(k) \left[ I - \hat{\mu}_p \overline{\Delta} \right] + \hat{\mu}_p \overline{C}
\]
where $\mathcal{C} = \mathbb{E}\left[\hat{\mathcal{C}}(k)\right]$ and $\mathcal{R} = \mathbb{E}\left[\hat{\mathcal{R}}(k)\right]$ and the steady-state mean weight error converges to the constant bias error

$$
\lim_{k \to \infty} \mathcal{W}(k) = \mathcal{W}_B = \mathcal{C} \mathcal{R}^{-1}
$$

In addition

$$
\lim_{k \to \infty} \mathcal{W}_B = 0
$$

**Proof:** For the proof, see the Appendix.

Along with the mean weight error, the noise signal induces a fluctuation weight error. The fluctuation weight error, defined as

$$
\tilde{w}(k) = w(k) - \bar{w}(k)
$$

is the variation of the instantaneous weighting matrix from the mean. Using (45), it is possible to determine the steady-state fluctuation error. Furthermore, it can be shown that an increase in the parameter $R$, the parameter dictating the time period over which output signals are predicted and compared on each iteration, reduces this fluctuation error.

**Theorem 6:** Assume that the plant to be identified is both observable and controllable from the input signal $\tilde{\mathcal{X}}(k)$ and the noise signal $\tilde{N}(k)$. Furthermore, assume that the evolution of the weight error is described by (45). Then the fluctuation error evolves as

$$
\tilde{w}(k + 1) = \tilde{w}(k)\left[1 - \tilde{\mu}_p \tilde{\mathcal{R}}\right] - \tilde{\mu}_p \bar{w}(k) \tilde{\mathcal{R}}(k) + \tilde{\mu}_p \tilde{\mathcal{C}}(k)
$$

where

$$
\tilde{\mathcal{R}}(k) = \hat{\mathcal{R}}(k) - \bar{\mathcal{R}}, \quad \tilde{\mathcal{C}}(k) = \hat{\mathcal{C}}(k) - \bar{\mathcal{C}}
$$

and the homogeneous solution of (52) converges exponentially to zero. In addition

$$
\lim_{k \to \infty} \left[\lim_{k \to \infty} \tilde{w}(k)\right] = 0
$$

**Proof:** For the proof, see the Appendix.

**4.2. Noise rejection with Toeplitz weighting structures**

Consider the batch ARMarkov network described by (42) and (43) with Toeplitz-constrained synapses. Utilizing the Hankel matrices defined in (36) and (37) as well as the Toeplitz weighting vector defined in (10), (42) and (43) can be rewritten as

$$
\tilde{y}_p(k) = \tilde{N}(k) + H\left[y(k - L - 1)\right]\mathcal{W}_{T(3,1)}^T + H\left[\mathcal{Z}(k - 1)\right]\mathcal{W}_{T(3,2)}^T
$$

$$
\tilde{y}(k) = H\left[y(k - L - 1)\right]\mathcal{W}_{T(3,1)}^T + H\left[\mathcal{Z}(k - 1)\right]\mathcal{W}_{T(3,2)}^T
$$

To evaluate the noise rejection properties of the ARMarkov replicator network, one need only evaluate the model weight error. Specifically, defining the model weight error as

$$
W = \begin{bmatrix} \mathcal{W}_{T(3,2)}^T & \mathcal{W}_{T(3,1)}^T \end{bmatrix}
$$

results in the weight error evolution function

$$
W(k + 1) = \left[I - \tilde{\mu}_p(k) \hat{\mathcal{R}}(k)\right]W(k) + \tilde{\mu}_p(k) \hat{\mathcal{C}}(k)
$$
where
\[
\hat{R}(k) = \frac{1}{R} \left[ H^T [\hat{\gamma}(k - 1)] \right] \left[ H \hat{\gamma}(k - 1) \right] H [\hat{\gamma}(k - L - 1)] \tag{58}
\]
\[
\hat{C}(k) = \frac{1}{R} \left[ H^T [\hat{\gamma}(k - 1)] \right] \bar{N}(k) \tag{59}
\]

Using (57), it is possible to determine the steady-state mean weight error for the network with Toeplitz-structured synapses. It can also be shown that an increase in the parameter \( L \), the ‘distant time’ parameter of the network, minimizes this bias error.

**Theorem 7:** Assume that the plant to be identified is both observable and controllable from the input signal \( \bar{\gamma}(k) \) and the noise signal \( \bar{N}(k) \). Furthermore, assume that (57) describes the evolution of the model weight error of the identification network with Toeplitz-structured synapses. Then the mean weight error evolves as
\[
\tilde{W}(k + 1) = \left[ I - \hat{\mu}_p(k) \hat{R} \right] \tilde{W}(k) + \hat{\mu}_p(k) \hat{C}(k) \tag{60}
\]
where \( \hat{R}(k) \equiv E[\hat{R}(k)] \), \( \hat{C}(k) \equiv E[\hat{C}(k)] \) and \( \tilde{W}(k) \equiv E[W(k)] \) and the steady-state weight error converges to the constant bias error
\[
\lim_{k \to \infty} \tilde{W}(k) = \tilde{W}_B = \hat{R}^{-1} \hat{C} \tag{61}
\]
In addition
\[
\lim_{L \to \infty} \tilde{W}_B = 0 \tag{62}
\]
**Proof:** For the proof, see the Appendix.

The fluctuation weight error is defined as
\[
\tilde{W}(k) = W(k) - \tilde{W}(k) \tag{63}
\]
as in (51). Using (57), it is possible to determine the steady-state fluctuation weight error for the network with Toeplitz-structured synapses. It can also be concluded that an increase in the parameter \( R \) the parameter dictating the time period over which output signals are predicted and compared on each iteration, minimizes this fluctuation weight error.

**Theorem 8:** Assume that the plant to be identified is both observable and controllable from the input signal \( \bar{\gamma}(k) \) and the noise signal \( \bar{N}(k) \). Furthermore, assume that (57) describes the evolution of the model weight error of the identification network with Toeplitz-structured synapses. Then the fluctuation error evolves as
\[
\tilde{W}(k + 1) = \left[ I - \hat{\mu}_p \hat{R} \right] \tilde{W}(k) - \hat{\mu}_p \hat{R}(k) \tilde{W}(k) + \hat{\mu}_p \hat{C} \tag{64}
\]
where
\[
\hat{R}(k) = \hat{R}(k) - \hat{R}, \quad \hat{C}(k) = \hat{C}(k) - \bar{C} \tag{65}
\]
and the homogeneous solution of (64) converges exponentially to zero. In addition
\[
\lim_{R \to \infty} \left[ \lim_{k \to \infty} \tilde{W}(k) \right] = 0 \tag{66}
\]
**Proof:** For the proof, see the Appendix.
5. Simulations

In this section we provide illustrations of the noise rejection properties of the batch ARMarkov network as a function of the design parameters $L$ and $R$. Specifically, we compare the relative performance of ARMA and CARMA models with that of batch ARMarkov models.

The ARMA and batch ARMarkov models were described in (13) and (15)–(21), respectively. The CARMA model, developed by Watanabe et al. (1992) has the form

$$y_m(k) = \sum_{i=1}^{M} A_i y_d(k - i) + \sum_{i=1}^{M} B_i u(k - i) = \sum_{j=1}^{n_c} C_j e(k - j)$$  \hspace{1cm} (66)

where $A_i, B_i, i = 1, \ldots, M$, and $C_j, j = 1, \ldots, n_c$, are model coefficients

$$e(k - j) = y_d(k - j) - y_m(k - j)$$

$y_d$ is plant output signal, and $y_m$ is model output signal. The CARMA model improves on the ARMA algorithm with its additional terms to compensate for the model’s error at previous estimates.

For illustrative purposes, consider the discrete-time representation of the simply-supported Euler–Bernoulli beam example described by Bernstein et al. (1986). Models were generated for a beam with a colocated sensor/actuator pair at 55/172 of the length of the beam. A diagram of this beam appears in Fig. 4. A ten-mode model was generated with a damping coefficient of 0.1 for each mode and a sampling period of 0.02 s. As seen in Fig. 5, the transfer function from actuator force to beam displacement, the modal frequencies are approximately 1.0, 4.0, 8.8, 15.9, 24.8, 35.9, 50.8, 65.6, 83.5 and 103.9 Hz.

Identification of the system was performed using the ARMA, CARMA and batch ARMarkov models. The batch ARMarkov model used Toeplitz constraints on its synapses. ARMA and CARMA models were of dimension $M + L$, where signal vectors consist of $L$ recent time signal values and $M$ distant time signal values, to maintain the same dimensionality as the batch ARMarkov model.

Comparison of the models is presented in the form of graphs of instantaneous real model error, $y - v - y_m$, versus the number of adaptation iterations. The instantaneous real model error is defined as the difference between the measured plant output less the sensor noise, $y - v$, and the model output, $y_m$.

The computational burden comparison was based on the coloured noise simulation in § 5.2. Flop counts in MATLAB were measured for each of the methods and are presented only as a means of comparison of the magnitudes. The flop counts were ARMA $28 \times 10^6$, CARMA $13 \times 10^6$, batch ARMarkov ($L = 50$) $106 \times 10^6$, and batch ARMarkov ($L = 100$) $91 \times 10^6$. With a constant parameter $R$ and sum $M + L$, note that the computational burden for the batch ARMarkov method

![Beam example diagram](image.png)
decreases with a decrease in parameter $M$ due to the use of the Hankel matrices defined in (36) and (37).

5.1. White sensor noise

Corrupting the sensor measurements with white noise was the first test of the noise rejection properties of the identification schemes. Sensor noise uncorrelated with the input signal and with a standard deviation of 10% of the standard deviation of the training input signal was generated. The instantaneous model error versus the adaptation iteration number can be seen in Fig. 6.

Parametrization of these models involved 0 recent time signal values ($L = 0$) and 200 distant time signal values ($M = 200$), thus with the distant time beginning at time step $k - 1$ as defined in (17). The CARMA model used 20 previous estimates ($n_c = 20$). The parameter $R$, which determines the number of past output signals generated and compared during each time interval, was set to $R = 1$ for the ARMA and CARMA models, but was increased to $R = 20$ and $R = 50$ in the batch ARMackov models for comparison. Note that the batch ARMakov model is reduced to the ARMA model when $R = 1$.

Theorem 8 predicts that an increase in parameter $R$ will reduce the fluctuation weight error. The fluctuation error caused by white sensor noise is demonstrated to be reduced in Fig. 6 with an increasing parameter $R$.

5.2. Coloured sensor noise

Next, the white noise was coloured before it corrupted the sensor measurements. Specifically, the white noise was filtered using

$$v_c(k) = 0.6v_c(k - 1) + v(k)$$

where $v_c(k)$ is the coloured noise with $v_c(0) = 0$, and $v(k)$ is the white noise signal. The instantaneous model error versus the adaptation iteration number is shown in Fig. 7.

Figure 5. Beam example transfer function.
Parametrization of these models involved 50 recent time signal values \((L = 50)\) and 150 distant time signal values \((M = 150)\). The second batch ARMarkov model used parameters \(L = 100\) and \(M = 100\). The CARMA model used 20 previous estimates \((n_c = 20)\). The parameter \(R\), which determines the number of past output signals generated and compared during each time interval, was set to \(R = 1\) for the ARMA and CARMA models, but increased to \(R = 20\) in the batch ARMarkov models for comparison.

Theorem 7 predicts that an increase in the parameter \(L\) will decrease the mean weight error. The mean error caused by coloured sensor noise is demonstrated to be reduced in Fig. 6 with increasing parameter \(L\).

### 6. Conclusion

This paper has demonstrated a system identification technique using neural networks capable of discerning the dynamics of a plant despite sensor signals corrupted by noise. The capacity to reduce fluctuation error, characterized by white sensor noise, was accomplished by the time averages in the structure of the network with sufficient parametrization. Likewise, an increase in the distant time of the network reduced the mean output error, characterized by coloured sensor noise. The basis for these noise rejection properties was developed in theorems for networks with and without Toeplitz constraints, and it was demonstrated with Toeplitz constraints in simulations.

This paper lays the theoretical foundation for extension of the main results into nonlinear system identification. Furthermore, general interconnection backpropagation laws allow for the development of adaptive neural control systems.
ACKNOWLEDGMENTS

This research was supported in part by the Air Force Office of Scientific Research under contracts F49620-92-C-0019 and F49620-95-1-0244, and the National Science Foundation under grant ECS-9496249.

Appendix

Proof of Theorem 1

Let $W^{A(i,j)}_{l,m}(\frac{1}{2}e^2)$ be the $lm$-element of $W^{A(i,j)}$. Then

$$- \frac{\partial}{\partial W_{l,m}^{A(i,j)}}(\frac{1}{2}e^2) = - \sum_{q} \varepsilon^{T} \frac{\partial e_{q}}{\partial W_{l,m}^{A(i,j)}}$$

$$= - \sum_{q} \varepsilon^{T} \frac{\partial}{\partial W_{l,m}^{A(i,j)}}(\Omega_{q}(\bar{y}_{q} - y_{q}))$$

$$= \sum_{q} \Omega_{q} \frac{\partial}{\partial W_{l,m}^{A(i,j)}}(\varepsilon_{q})$$

because $\Omega_{q}$ is a scalar and $\bar{y}_{q}$ is independent of $W^{A(i,j)}$.

For $q \neq i$, it follows that

$$\frac{\partial \bar{y}_{q}}{\partial W_{l,m}^{A(i,j)}} = \mathcal{G}(\bar{u}_{q}) \left( \sum_{k} \frac{\partial \bar{y}_{k}}{\partial W_{l,m}^{A(i,j)}} \right)$$

(A1)

Figure 7. Coloured sensor noise (std = standard deviation).
whereas for \( q = i \)

\[
\frac{\partial \bar{y}_i}{\partial W_{lm}^{(i,j)}} = \hat{g}(\bar{u}_i) \left( \sum_{k=1}^{N} W_{lk}^{(i,k)} \frac{\partial \bar{y}_k}{\partial W_{lm}^{(i,j)}} + \frac{\partial W_{lm}^{(i,j)}}{\partial W_{lm}^{(i,j)}} \bar{y}_j \right)
\]

\[
= \hat{g}(\bar{u}_i) \left( \sum_{k=1}^{N} W_{lk}^{(i,k)} \frac{\partial \bar{y}_k}{\partial W_{lm}^{(i,j)}} + E_{lm} \bar{y}_j \right)
\]

(A 3)

where

\[
E_{lm} = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix} = e_le_m^T
\]

Thus

\[
\frac{\partial}{\partial W_{lm}^{(i,j)}} \begin{bmatrix} \bar{y}_1 \\ \vdots \\ \bar{y}_i \\ \vdots \\ \bar{y}_N \end{bmatrix} = \begin{bmatrix} \hat{g}(\bar{u}_1) \\ \vdots \\ \hat{g}(\bar{u}_i) \\ \vdots \\ \hat{g}(\bar{u}_N) \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} + \hat{g}(\bar{u}_i) E_{lm} \bar{y}_j
\]

(A 5)

or, equivalently

\[
\frac{\partial}{\partial W_{lm}^{(i,j)}} \begin{bmatrix} \bar{y}_1 \\ \vdots \\ \bar{y}_i \\ \vdots \\ \bar{y}_N \end{bmatrix} = \sum_{q=1}^{N} \text{diag} \hat{g}(\bar{u}_q) [W_{r,s}^{(r,s)}] \frac{\partial}{\partial W_{lm}^{(i,j)}} \begin{bmatrix} \bar{y}_1 \\ \vdots \\ \bar{y}_i \\ \vdots \\ \bar{y}_N \end{bmatrix} + \sum_{q=1}^{N} \text{diag} \hat{g}(\bar{u}_q) E_{lm} \bar{y}_j
\]

(A 6)

Combining terms yields

\[
\frac{\partial}{\partial W_{lm}^{(i,j)}} \begin{bmatrix} \bar{y}_1 \\ \vdots \\ \bar{y}_i \\ \vdots \\ \bar{y}_N \end{bmatrix} = \sum_{q=1}^{N} \text{diag} \hat{g}(\bar{u}_q) \left( I - [W_{r,s}^{(r,s)}] \sum_{q=1}^{N} \text{diag} \hat{g}(\bar{u}_q) \right)^{-1} E_{lm} \bar{y}_j
\]

(A 7)
Next, substituting (A 7) into (A 1) yields

\[- \frac{\partial}{\partial W_{l,m}^{(i,j)}} \left( \frac{1}{2} e^2 \right) = \left[ I - \left[ \text{diag } \mathcal{G} \left( \bar{u}_q \right) \left[ W^{(r,s)} \right] \right]^{-1} \left[ \text{diag } \mathcal{G} \left( \bar{u}_q \right) \right] \right]^T \begin{bmatrix} \Omega_1 \bar{\epsilon}_1 \\ \vdots \\ \Omega_N \bar{\epsilon}_N \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \]

From (4) for \( \bar{y}_i^* \) it follows that

\[
\begin{bmatrix} \bar{y}_1^* \\ \vdots \\ \bar{y}_i^* \\ \vdots \\ \bar{y}_N^* \end{bmatrix} = \left[ \text{diag } \hat{\mathcal{G}} \left( \bar{u}_q \right) \right] \begin{bmatrix} W^{(1,1)}^T \\ \vdots \\ W^{(N,1)}^T \\ \vdots \\ W^{(N,N)}^T \end{bmatrix} \begin{bmatrix} \bar{y}_1 \\ \vdots \\ \bar{y}_i \\ \vdots \\ \bar{y}_N \end{bmatrix} + \left[ \text{diag } \hat{\mathcal{G}} \left( \bar{u}_q \right) \right] \begin{bmatrix} \Omega_1 \bar{\epsilon}_1 \\ \vdots \\ \Omega_i \bar{\epsilon}_i \\ \vdots \\ \Omega_N \bar{\epsilon}_N \end{bmatrix}
\]

Finally, substituting (A 9) into (A 8) yields

\[- \frac{\partial}{\partial W_{l,m}^{(i,j)}} \left( \frac{1}{2} e^2 \right) = \begin{bmatrix} \bar{y}_1^T \\ \vdots \\ \bar{y}_N^T \end{bmatrix} \begin{bmatrix} \Omega_1 \bar{\epsilon}_1 \\ \vdots \\ \Omega_i \bar{\epsilon}_i \\ \vdots \\ \Omega_N \bar{\epsilon}_N \end{bmatrix} = \bar{y}_i^T E_{l,m} \bar{y}_j = (\bar{y}_i^*) (\bar{y}_j)_m \]

and hence we arrive at the result

\[- \frac{\partial}{\partial W_{l,m}^{(i,j)}} \left( \frac{1}{2} e^2 \right) = \bar{y}_i^T \bar{y}_j \quad \square \]

**Proof of Theorem 2**

Let \( W_{i,j}^{(i,j)} \left( \frac{1}{2} e^2 \right) \) be the \( l \)th component of \( W_{l,m}^{(i,j)} \). Then

\[- \frac{\partial}{\partial W_{i,j}^{(i,j)}} \left( \frac{1}{2} e^2 \right) = \sum_{q=1}^{N} \Omega_q \bar{\epsilon}_q \frac{\partial \bar{y}_q}{\partial W_{i,j}^{(i,j)}} \]

\[(A 12)\]
Now for \( q \neq i \) it follows that
\[
\frac{\partial \bar{y}_q}{\partial W^{(i,j)}_l} = \mathcal{G}(\bar{u}_q) \left( \sum_{k=1}^N W^{(q,k)} A_{ijkl} \frac{\partial \bar{y}_k}{\partial W^{(i,j)}_l} \right)
\]  
(A 13)

whereas for \( q = i \)
\[
\frac{\partial \bar{y}_i}{\partial W^{(i,j)}_l} = \mathcal{G}(\bar{u}_i) \left( \sum_{k=1}^N W^{(i,k)} A_{ijkl} \frac{\partial \bar{y}_k}{\partial W^{(i,j)}_l} + \frac{\partial W^{(i,j)}}{\partial W^{(i,j)}_l} \bar{y}_j \right)
\]
\[
= \mathcal{G}(\bar{u}_i) \left( \sum_{k=1}^N W^{(i,k)} A_{ijkl} \frac{\partial \bar{y}_k}{\partial W^{(i,j)}_l} + T_l \bar{y}_j \right)
\]  
(A 14)

where
\[
T_l = \begin{bmatrix}
0 & 0 & \cdots & 0 & 1 & 0 \\
0 & 0 & 1 & \cdots & 0 & 0 \\
0 & 0 & 0 & \cdots & 0 & 1 \\
\end{bmatrix} \in \mathbb{R}^{p \times p_j}
\]  
(A 15)

consists of 1s along the \( l \)th super-diagonal. Thus
\[
\frac{\partial}{\partial W^{(i,j)}_l} \begin{bmatrix} \bar{y}_1 \\ \vdots \\ \bar{y}_i \\ \vdots \\ \bar{y}_N \end{bmatrix} = \begin{bmatrix} \mathcal{G}(\bar{u}_1) \\ \cdots \\ \mathcal{G}(\bar{u}_i) \\ \cdots \\ \mathcal{G}(\bar{u}_N) \end{bmatrix} \begin{bmatrix}
0 & & & \\
& \ddots & & \\
& & \ddots & \\
& & & 0 \\
\end{bmatrix} + \begin{bmatrix} \mathcal{G}(\bar{u}_1) \\ \cdots \\ \mathcal{G}(\bar{u}_i) \\ \cdots \\ \mathcal{G}(\bar{u}_N) \end{bmatrix} T_l \bar{y}_j
\]
\[
\times \frac{\partial}{\partial W^{(i,j)}_l} \begin{bmatrix} \bar{y}_1 \\ \vdots \\ \bar{y}_i \\ \vdots \\ \bar{y}_N \end{bmatrix}
= \begin{bmatrix} \mathcal{G}(\bar{u}_1) \\ \cdots \\ \mathcal{G}(\bar{u}_i) \\ \cdots \\ \mathcal{G}(\bar{u}_N) \end{bmatrix} \begin{bmatrix}
0 & & & \\
& \ddots & & \\
& & \ddots & \\
& & & 0 \\
\end{bmatrix} + \begin{bmatrix} \mathcal{G}(\bar{u}_1) \\ \cdots \\ \mathcal{G}(\bar{u}_i) \\ \cdots \\ \mathcal{G}(\bar{u}_N) \end{bmatrix} T_l \bar{y}_j
\]  
(A 16)

or equivalently
\[
\frac{\partial}{\partial W^{(i,j)}_l} \begin{bmatrix} \bar{y}_1 \\ \vdots \\ \bar{y}_i \\ \vdots \\ \bar{y}_N \end{bmatrix} = \begin{bmatrix} \mathcal{G}(\bar{u}_q) \\ \cdots \\ \mathcal{G}(\bar{u}_q) \end{bmatrix} \begin{bmatrix}
\begin{bmatrix} W^{(r,s)} & \cdots & W^{(r,N)} \\
\vdots & \ddots & \vdots \\
W^{(N,r)} & \cdots & W^{(N,N)} \\
\end{bmatrix} & \\
\mathcal{G}(\bar{u}_q) & \cdots & \mathcal{G}(\bar{u}_q) \end{bmatrix} \\
\begin{bmatrix} \bar{y}_1 \\ \vdots \\ \bar{y}_i \\ \vdots \\ \bar{y}_N \end{bmatrix}
\end{bmatrix} + \begin{bmatrix} \mathcal{G}(\bar{u}_q) \\ \cdots \\ \mathcal{G}(\bar{u}_q) \end{bmatrix} T_l \bar{y}_j
\]  
(A 17)
Collecting terms yields

\[
\frac{\partial}{\partial W_{ij}^{l}} \begin{bmatrix}
\hat{y}_1 \\
\vdots \\
\hat{y}_y \\
\vdots \\
\hat{y}_N
\end{bmatrix} = \left[ I - \left[ \frac{N}{q=1} \text{diag} \hat{G}(\bar{u}_q) \right] \left[ \frac{N}{q=1} \text{diag} \hat{G}(\bar{u}_q) \right]^T \right]^{-1} \begin{bmatrix}
0 \\
\vdots \\
T_i \hat{y}_j \\
\vdots \\
0
\end{bmatrix} \quad (A\ 18)
\]

Now, substituting (A\ 18) into (A\ 12)

\[- \frac{\partial}{\partial W_{ij}^{l}} (\frac{1}{2} \epsilon^2) = \left( I - \left[ \frac{N}{q=1} \text{diag} \hat{G}(\bar{u}_q) \right] \left[ \frac{N}{q=1} \text{diag} \hat{G}(\bar{u}_q) \right]^T \right]^{-1} \begin{bmatrix}
\frac{N}{q=1} \text{diag} \hat{G}(\bar{u}_q) \\
\vdots \\
\frac{N}{q=1} \text{diag} \hat{G}(\bar{u}_q)
\end{bmatrix} \begin{bmatrix}
\Omega_{1 \bar{e}_1} \\
\vdots \\
\Omega_{N \bar{e}_N}
\end{bmatrix} + 1 \begin{bmatrix}
\frac{N}{q=1} \text{diag} \hat{G}(\bar{u}_q) \\
\vdots \\
\frac{N}{q=1} \text{diag} \hat{G}(\bar{u}_q)
\end{bmatrix} \begin{bmatrix}
\Omega_{1 \bar{e}_1} \\
\vdots \\
\Omega_{N \bar{e}_N}
\end{bmatrix}
\]

From (4) for \( \tilde{y}_i^* \) it follows that

\[
\begin{bmatrix}
\tilde{y}_1^* \\
\vdots \\
\tilde{y}_y^* \\
\vdots \\
\tilde{y}_N^*
\end{bmatrix} = \begin{bmatrix}
\frac{N}{q=1} \text{diag} \hat{G}(\bar{u}_q) \\
\vdots \\
\frac{N}{q=1} \text{diag} \hat{G}(\bar{u}_q)
\end{bmatrix} \begin{bmatrix}
W^{(1,1)} \\
\vdots \\
W^{(N,1)}
\end{bmatrix} \begin{bmatrix}
\tilde{y}_1 \\
\vdots \\
\tilde{y}_y \\
\vdots \\
\tilde{y}_N
\end{bmatrix} + \begin{bmatrix}
\frac{N}{q=1} \text{diag} \hat{G}(\bar{u}_q) \\
\vdots \\
\frac{N}{q=1} \text{diag} \hat{G}(\bar{u}_q)
\end{bmatrix} \begin{bmatrix}
\Omega_{1 \bar{e}_1} \\
\vdots \\
\Omega_{N \bar{e}_N}
\end{bmatrix}
\]

Finally, substituting (A\ 20) into (A\ 19) yields

\[- \frac{\partial}{\partial W_{ij}^{l}} (\frac{1}{2} \epsilon^2) = \begin{bmatrix}
\tilde{y}_1^T \\
\vdots \\
\tilde{y}_y^T
\end{bmatrix} + \begin{bmatrix}
\tilde{y}_j \\
\vdots \\
\tilde{y}_j
\end{bmatrix} \begin{bmatrix}
\Omega_{1 \bar{e}_1} \\
\vdots \\
\Omega_{N \bar{e}_N}
\end{bmatrix}
\]

\[
= \tilde{y}_i^T T_i \tilde{y}_j
\]

\[
= \begin{bmatrix}
y_{i,1}^* \\
\vdots \\
y_{i,p_i}^*
\end{bmatrix}
\begin{bmatrix}
y_{j,1} \\
\vdots \\
y_{j,p_i+1-1}
\end{bmatrix}
\]

\[
= \sum_{k} y_{i,k}^* y_{j,k+i-1} \quad (A\ 21)
\]
and hence

\[- \frac{\partial}{\partial W_T^{(i,j)}} \left( \frac{1}{2} \epsilon^2 \right) = \begin{bmatrix} y_{j,1} & \cdots & y_{j,p_j} \\ \vdots & \ddots & \vdots \\ y_{i,p_i} & \cdots & y_{i,1} \\ 0 & \cdots & y_{i,1} \end{bmatrix} \]

(A 22)

**Proof of Theorem 4**

First, recall that the weighting matrix update is given by

\[- \frac{\partial}{\partial W_T^{(3,1)}} \left( \frac{1}{2} \epsilon^2 \right) = \begin{bmatrix} y_{3,1} & \cdots & y_{3,1} \\ \vdots & \ddots & \vdots \\ y_{3,R} & \cdots & y_{3,R} \\ 0 & \cdots & y_{3,R} \end{bmatrix} \]

(A 23)

Thus

\[- \frac{\partial}{\partial W_T^{(3,1)}} \left( \frac{1}{2} \epsilon^2 \right) = \begin{bmatrix} y_{1,1} & \cdots & y_{1,1} \\ \vdots & \ddots & \vdots \\ y_{1,R} & \cdots & y_{1,R} \\ y_{1,M+R-1} & \cdots & y_{1,M+R-1} \end{bmatrix} \]

\[= H^T \begin{bmatrix} y_d(k-L-1) \end{bmatrix}^* \]

(A 24)
and hence

\[- \frac{\partial}{\partial W^{(3,2)}_T}(\frac{1}{2} \varepsilon^2) = \begin{bmatrix}
  y_{2,1} & \cdots & y_{2,L+M+R-1}^*
\end{bmatrix}
\]

\[= H^T \bar{\xi}_0(k-1) \bar{\xi}_3^* \tag{A 25} \]

**Evolution of the weight error**

Incorporating (42) and (43) into the output error equation, it follows that

\[\hat{\varepsilon} = \begin{bmatrix} 0_{R \times L} \quad W^{(3,1)}^* - W^{(3,1)} \end{bmatrix} \xi - \begin{bmatrix} 0_{R \times L} \quad W^{(3,1)} - W^{(3,1)}^* \end{bmatrix} \xi + \bar{\eta}(k) \tag{A 26} \]

The weight update laws yield

\[\begin{bmatrix} 0_{R \times L} \quad W^{(3,1)}(k+1) \end{bmatrix} = \begin{bmatrix} 0_{R \times L} \quad W^{(3,1)}(k) \end{bmatrix} - \begin{bmatrix} \hat{\mu}_p \xi \end{bmatrix}^T \begin{bmatrix} \bar{y}^T_p(k-1) \end{bmatrix} \tag{A 27} \]

\[W^{(3,2)}(k+1) = W^{(3,2)}(k) - \hat{\mu}_p \xi \tag{A 28} \]

By substituting (A 26) into (A 27), it follows after some algebraic manipulation that

\[\begin{bmatrix} 0_{R \times L} \quad W^{(3,1)}(k+1) - W^{(3,1)^*} \end{bmatrix} = \begin{bmatrix} 0_{R \times L} \quad W^{(3,1)}(k) - W^{(3,1)^*} \end{bmatrix} \xi + \bar{\eta}(k) \tag{A 29} \]

Similarly, substituting (A 26) into (A 28) yields

\[\begin{bmatrix} W^{(3,2)}(k+1) - W^{(3,2)^*} \end{bmatrix} = \begin{bmatrix} W^{(3,2)}(k) - W^{(3,2)^*} \end{bmatrix} \xi + \bar{\eta}(k) \tag{A 30} \]
Thus, substituting (A.29) and (A.30) into (44) results in the weight error evolution function

\[ w(k + 1) = w(k)[I - \hat{\mu}_p(k)\hat{R}(k)] + \hat{\mu}_p(k)\hat{C}(k) \]  

(31)

where

\[ \hat{R}(k) = \begin{bmatrix} \tilde{\xi}(k - 1) \\ \tilde{\xi}(k - 1) \\ 0_L \end{bmatrix} \begin{bmatrix} \tilde{\xi}^T(k - 1) & \tilde{\xi}^T(k - 1) & 0_L & \tilde{y}_p^T(k - 1) \end{bmatrix} \]  

(A.32)

and

\[ \hat{C}(k) = \overline{N}(k)\begin{bmatrix} \tilde{\xi}_2^T(k - 1) & \tilde{\xi}_2^T(k - 1) & 0_L & \tilde{y}_p^T(k - 1) \end{bmatrix} \]  

(A.33)

Proof of Theorem 5

If the input vector \( \tilde{\xi}(k) \) and the noise signal \( \overline{N}(k) \) are ergodic processes, then so is the network output \( \overline{y}(k) \) because the network is linear and strictly proper and driven solely by \( \tilde{\xi}(k) \) and \( \overline{N}(k) \). Next, note that

\[ \text{tr}[\hat{R}(k)] = \sum_{l=1}^{M+L} \overline{\xi}^2(k - l) + \sum_{l=1}^{M} y^2(k - L - l) \]  

(A.34)

Now, for sufficiently large \( L \) and \( M \), the summations in (A.34) are proportional to time averages, and hence

\[ \text{tr}[\hat{R}(k)] \approx (M + L) \mathbb{E}[\overline{\xi}^2(k)] + M \mathbb{E}[y^2(k)] \]

\[ = \text{tr}\mathbb{E}[\hat{R}(k)] \]

\[ = \text{tr}[\overline{R}(k)] \]  

(A.35)

Furthermore, consider the step length parameter introduced in (40) rewritten using (46) as

\[ \hat{\mu}_p(k) = \frac{\alpha}{\text{tr} \hat{R}(k)} \]  

(A.36)

Hence

\[ \hat{\mu}_p(k) \approx \frac{\alpha}{\overline{R}} \]

\[ \hat{\mu}_p \]  

(A.37)

Furthermore, because \( w \) typically varies slowly from \( \overline{w} \), it follows that

\[ \mathbb{E}[w(k)\hat{R}(k)] \approx \overline{w}(k) \overline{R} \]  

(A.38)

Using the approximations above and taking the expected values of both sides of (45) yields

\[ \overline{w}(k + 1) = \overline{w}(k)[I - \hat{\mu}_p \overline{R}] + \hat{\mu}_p \overline{C} \]  

(A.39)
with
\[
\mathcal{C} = \mathbb{E}\left[\hat{C}(k)\right] = \begin{bmatrix}
\mathbb{E}\left[N(k)\bar{x}^T(k - 1)\right] \\
\mathbb{E}\left[N(k)\bar{x}^T(k - 1)\right] \\
\tilde{0}_{K \times L}^T \\
\mathbb{E}\left[N(k)\tilde{v}_p^T(k - 1)\right]
\end{bmatrix}^T
\] (A 40)

If the plant is observable and controllable from \(N(k)\) and \(\bar{z}(k)\), the homogeneous solution of (A 39) converges to zero, and the non-homogeneous solution converges to
\[
\bar{w}_B = \mathcal{C}\mathcal{R}^{-1}
\] (A 41)

If \(\bar{z}(k)\) and \(\bar{N}(k)\) are independent processes, then \(\mathcal{C}(k)\) reduces to
\[
\mathcal{C} = \begin{bmatrix}
\tilde{0}_{K \times L}^T & \tilde{0}_{K \times (M + R)}^T & \tilde{0}_{K \times L}^T \\
\mathbb{E}\left[N(k)\bar{x}^T(k - 1)\right]
\end{bmatrix}^T
\] (A 42)

As the parameter \(L\) increases, the correlation between \(\bar{y}_p(k - 1)\) and \(\bar{N}(k)\) diminishes. Thus
\[
\lim_{L \to \infty} \mathbb{E}\left[N(k)\bar{y}_p^T(k - 1)\right] = 0
\] (A 43)

**Proof of Theorem 6**

From (45), (48) and (51), the assumption that \(\hat{\mu}_p(k) = \bar{\mu}_p\), and neglecting products of fluctuations, it follows that
\[
\bar{w}(k + 1) = \bar{w}(k)\left[I - \bar{\mu}_p\bar{R}\right] + \bar{\mu}_p \bar{w}(k)\bar{R}(k) + \bar{\mu}_p \mathcal{C}(k)
\] (A 44)

where
\[
\bar{R}(k) = \hat{R}(k) - \bar{\mathcal{R}}, \quad \mathcal{C}(k) = \hat{\mathcal{C}} - \bar{\mathcal{C}}
\]

The homogeneous solution of (A 44) converges to zero, but the non-homogeneous solution is dependent on the fluctuations \(\bar{R}(k)\) and \(\mathcal{C}(k)\). The structure of \(\bar{R}(k)\) and \(\mathcal{C}(k)\) is such that all terms are simply time averages over \(R\) time steps. For example, from the definition of \(\bar{R}(k)\)

\[
\frac{1}{R}H^T\left[\bar{z}_d(k - 1)\right]H\left[\bar{z}_d(k - 1)\right] = \\
\begin{bmatrix}
\frac{1}{R}\sum_{m=1}^{R} \bar{z}^2(k - m) & \frac{1}{R}\sum_{m=1}^{R} \bar{z}(k - m)\bar{z}(k - m - 1) & \cdots \\
\frac{1}{R}\sum_{m=1}^{R} \bar{z}(k - m - 1)\bar{z}(k - m) & \frac{1}{R}\sum_{m=1}^{R} \bar{z}^2(k - m - 1) & \cdots \\
\vdots & \vdots & \ddots
\end{bmatrix}
\] (A 45)
Thus

$$\lim_{k \to \infty} \hat{R}(k) = \bar{R}(k)$$

(A 46)

and

$$\lim_{k \to \infty} \hat{C}(k) = \bar{C}(k)$$

(A 47)

leading to the conclusion

$$\lim_{k \to \infty} \left[ \lim_{k \to \infty} \tilde{w}(k) \right] = 0$$

(A 48)

Proof of Theorem 8

The output error equation for the plant output described in (54) and network model described in (55) results in the output error equation

$$\bar{e}(k) = \bar{N}(k) + H \left[ y(k - L - 1) W_{1,1} - W_{1,1} \right] + H \left[ \hat{y}(k - 1) W_{1,1} - W_{1,1} \right]$$

(A 49)

Using the definitions \( \bar{R}(k) = \mathbb{E} \left[ \hat{R}(k) \right] \), \( \bar{C}(k) = \mathbb{E} \left[ \hat{C}(k) \right] \) and \( \tilde{W}(k) = \mathbb{E} \left[ \hat{W}(k) \right] \) and the approximations \( \hat{\mu}_p(k) = \bar{\mu}_p \) and \( \mathbb{E} \left[ \hat{R}(k)W(k) \right] = \bar{R} \tilde{W}(k) \), and taking the expected value of both sides of (57) yields

$$\tilde{W}(k + 1) = \left[ I - \bar{\mu}_p \bar{R} \right] \tilde{W}(k) + \bar{\mu}_p \bar{C}(k)$$

(A 50)

If the plant is observable and controllable from \( \bar{N}(k) \) and \( \bar{e}(k) \), the homogeneous solution of (A 50) converges to zero and the non-homogeneous solution converges to

$$\tilde{W}_B = \bar{R}^{-1} \bar{C}$$

(A 51)

If \( \bar{e}(k) \) and \( \bar{N}(k) \) are independent, then \( \bar{C}(k) \) reduces to

$$\bar{C} = \left[ \begin{array}{cc} 0 & \bar{N}(k) \\ \frac{1}{R} H^T \left[ y(k - L - 1) \right] \end{array} \right]$$

(A 52)

As parameter \( L \) increases, the correlation between \( \bar{y}_p(k - 1) \) and \( \bar{N}(k) \) diminishes. Thus

$$\lim_{L \to \infty} \tilde{W}_B = 0$$

(A 53)

Proof of Theorem 8

From (57), (60) and (63) and the assumption that \( \hat{\mu}_p(k) = \bar{\mu}_p \), it follows that

$$\tilde{W}(k + 1) = \tilde{W}(k) - \bar{\mu}_p \bar{R}(k) \tilde{W}(k) - \bar{\mu}_p \bar{R}(k) \tilde{W}(k) + \bar{\mu}_p \bar{R} \tilde{W}(k) + \bar{\mu}_p \bar{C}(k)$$

(A 54)

where

$$\bar{R}(k) = \bar{R}(k) - \bar{R}, \quad \bar{C}(k) = \bar{C}(k) - \bar{C}$$

Next, collecting terms and ignoring products of fluctuations yields

$$\tilde{W}(k + 1) = \left[ I - \bar{\mu}_p \bar{R} \right] \tilde{W}(k) - \bar{\mu}_p \bar{R}(k) \tilde{W}(k) + \bar{\mu}_p \bar{C}(k)$$

(A 55)

The homogeneous solution of (A 55) converges to zero, but, as in the proof of Theorem 6, the non-homogeneous solution is dependent on the fluctuations \( \bar{R}(k) \)
The structures of $\hat{R}(k)$ and $\hat{C}(k)$ have all terms as simply time averages over $R$ time values. Thus

$$\lim_{R \to \infty} \hat{R}(k) = \bar{R}(k) \quad \text{(A 56)}$$

$$\lim_{R \to \infty} \hat{C}(k) = \bar{C}(k) \quad \text{(A 57)}$$

leading to the conclusion

$$\lim_{R \to \infty} \left[ \lim_{k \to \infty} \tilde{w}(k) \right] = 0 \quad \text{(A 58)}$$

---

**REFERENCES**


