# ADAPTIVE NONLINEAR DIGITAL FILTER WITH SEQUENTIAL REGRESSION ALGORITHM

D.H. YOUN, K.K. YU and V.J. MATHEWS
Department of Electrical and Computer Engineering
The University of Iowa
Iowa City, Iowa 52242

## ABSTRACT

The purpose of this paper is to introduce an adaptive nonlinear digital filtering algorithm which use the sequential regression (SER) method to update the second order Volterra filter coefficients in a recursive way. Conventionally, the SER method has been used to invert large matrices which result from direct application of the Wiener filter theory to the Volterra filter. However, the algorithm proposed in this paper adopts the simplified least squares solution that results when the input signals are Gaussian, and hence the size of the matrix to be inverted is smaller than that of the conventional approaches. Simulation results are also included to demonstrate the performance of the proposed algorithm.

## 1. INTRODUCTION

The Volterra series representation of nonlinear systems is an extension of the linear system theory. The output y(k) of a (causal) discrete-time, nonlinear system can be represented as a function of the input x(k) using the Volterra series as

$$
\begin{aligned}
y(k) = h_0 &+ \sum_{m_1=0}^{\infty} h_1(m_1)x(k-m_1) \\
&+ \sum_{m_1=0}^{\infty} \sum_{m_2=0}^{\infty} h_2(m_1,m_2)\, x(k-m_1)x(k-m_2) + \dots \\
&+ \sum_{m_1=0}^{\infty} \sum_{m_2=0}^{\infty} \cdots \sum_{m_p=0}^{\infty} h_p(m_1,m_2,\dots,m_p)x(k-m_1)x(k-m_2)\dots x(k-m_p) \\
&+ \dots
\end{aligned}
\tag{1}
$$

where $h_p(m_1,m_2,\dots,m_p)$ is the p-th order Volterra kernal [1] of the nonlinear system. This paper is concerned with adaptive identification of nonlinear systems that can be represented using a second order Volterra series. A least square (LS) solution is obtained for the system identification problem assuming Gaussian reference inputs and an adaptive filtering algorithm which computes the LS solution in a recursive manner using the sequential regression (SER) method is derived.

Conventionally, the SER method has been used to invert a large matrix which results from direct application of the Wiener filter theory to nonlinear systems [6]. However, the algorithm proposed in this paper adopts the simplified solution [5] that results when the input signals are Gaussian, and therefore the size of the matrix to be inverted is smaller than that of the conventional approach.

The rest of the paper is organized as follows: In Section 2, a formal statement of the problem is presented. A LS solution to the problem is derived in this section. The adaptive Volterra filter with SER algorithm is derived in Section 3. The effectiveness of the proposed algorithm is demonstrated using a simulation example in Section 4. Finally, we make the concluding remarks in Section 5.

## 2. PROBLEM STATEMENT AND OPTIMUM SOLUTION

Let H in Fig. 1 represent an unknown nonlinear system, that can be represented as a second order Volterra filter. Then, the output $y(k)$ can be obtained using the matrix equation in terms of the input $x(k)$ as

$$y(k) = A^T X(k) + \mathrm{tr}\{B[X(k)X^T(k) - R_{xx}]\} \tag{2}$$

where

$$X(k) = [x(k), x(k-1), \ldots, x(k-N+1)]^T,$$

$$A = [a_0, a_1, \ldots, a_{N-1}]^T,$$

$$B = \begin{bmatrix} b_{0,0} & b_{0,1} & \cdots & b_{0,N-1} \\ b_{1,0} & b_{1,1} & \cdots & b_{1,N-1} \\ \vdots & & & \\ b_{N-1,0} & b_{N-1,1} & \cdots & b_{N-1,N-1} \end{bmatrix}$$

is a symmetric matrix so that $b_{i,j} = b_{j,i}$, $R_{xx} = E\{X(k)X^T(k)\}$ is the autocorrelation matrix of $X(k)$, $\mathrm{tr}\{\cdot\}$ denotes the trace of the matrix $\{\cdot\}$ and $(\cdot)^T$ denotes the transpose of the vector $(\cdot)$. We will assume that the input $x(k)$ is Gaussian. Also, it will be assumed without loss of generality that $x(k)$ and $y(k)$ are zero mean.

The problem here is to derive an adaptive filtering algorithm that uses the SER method to track the (possibly time-varying) parameters A and B of the Volterra filter in (2) so as to minimize the cost functional

$$C(\hat{A}(k),\hat{B}(k)) = \sum_{i=0}^{k} q(i,k)\{y(i) - [\hat{A}(k)]^T X(i) - \mathrm{tr}[\hat{B}(k)(X(i)X^T(i) - \hat{R}_{xx}(k))]\}^2 \tag{3}$$

where $\hat{}$ denotes an estimated quantity and $q(i,k)$ is a weighting function for the squared estimation error $[y(i) - \hat{y}_k(i)]^2$ at time i, when the estimates $\hat{A}(k)$, $\hat{B}(k)$ and $\hat{R}_{xx}(k)$ at time k are used to obtain the estimate $\hat{y}_k(i)$ of $y(i)$. For analytical tractability, we will assume that the autocorrelation matrix $R_{xx}$ is estimated at time k as

$$\hat{R}_{xx}(k) = \sum_{i=0}^{k} q(i,k) X(k) X^T(k) \ . \tag{4}$$

Then, the cost functional that is to be minimized becomes

$$C(\hat{A}(k),\hat{B}(k)) = \sum_{i=0}^{k} \{q(i,k)\{y(i) - \hat{A}^T(k)X(i) - tr[\hat{B}(k)(X(i)X^T(i) - \sum_{j=0}^{k} q(j,k)X(j)X^T(j))]\}^2\} . \tag{5}$$

We can minimize the above cost functional with respect to $\hat{A}(k)$ by setting the gradient of $C(\hat{A}(k),\hat{B}(k))$ with respect to $\hat{A}(k)$ to zero, i.e., by setting $\nabla_{\hat{A}(k)} C(\hat{A}(k),\hat{B}(k)) = 0$. After some straightforward computations, we obtain

$$\hat{R}_{yx}(k) = \hat{R}_{xx}(k)\ \overset{0}{A}(k) + \sum_{m,n=0}^{N-1} \hat{b}_{m,n}(k)\ [\sum_{i=0}^{k} q(i,k)x(i-m)x(i-n)X(i)] + [\sum_{j=0}^{k} q(j,k)X^T(j)\hat{B}(k)X(j)]\ [\sum_{i=0}^{k} q(i,k)X(i)], \tag{6a}$$

where

$$\hat{R}_{yx}(k) = \sum_{i=0}^{k} q(i,k)y(i)X(i) \tag{6b}$$

and $\overset{0}{A}(k)$ yields the minimum $C(\hat{A}(k),\hat{B}(k))$.

If the weighting function $q(i,k)$ for $0 \leq i \leq k$ represents the impulse response function of a lowpass filter with unit gain at zero frequency, i.e., $\sum_{i=0}^{k} q(i,k)=1$, we can see that $\sum_{i=0}^{k} q(i,k)x(i-j)$ and $\sum_{i=0}^{k} q(i,k)x(i-m)x(i-n)x(i-j)$ approximate the mean and third order moment of $x(k)$, respectively, which means that second and third terms of (6a) are approximately zero. Substitution of this in (6b) results in the simplified expression for $\overset{0}{A}(k)$, which is given by

$$\overset{0}{A}(k) = \hat{R}_{xx}^{-1}(k)\ \hat{R}_{yx}(k) . \tag{7}$$

To obtain the LS solution for $\hat{B}(k)$, we once again set the gradient of $C(\hat{A}(k),\hat{B}(k))$ with respect to $\hat{B}(k)$ to zero and obtain

$$\overset{0}{B}(k) = \frac{1}{2}\ \hat{R}_{xx}^{-1}(k)T_{yx}(k)\hat{R}_{xx}^{-1}(k) \tag{8}$$

where

$$T_{yx}(k) = \sum_{i=0}^{k} q(i,k)y(i)X(i)X^T(i) . \tag{9}$$

The derivation is straightforward and can be found in [3]. It makes use of the fact that for a Gaussian signal $x(k)$,

$$E\{x(k-i)x(k-j)x(k-m)x(k-n)\} = r_{xx}(i-j)r_{xx}(m-n) + r_{xx}(i-m)r_{xx}(j-n) + r_{xx}(i-n)r_{xx}(j-m) \tag{10}$$

and the approximate relationship

$$\sum_{i=0}^{k} q(i,k)x(i-m)x(i-n) \ldots x(i-j) \approx E\{x(k-m)x(k-n) \ldots x(k-j)\}. \quad (11)$$

In (10), $r_{xx}(i-j) = E\{x(k-i)x(k-j)\}$.

It may be pointed out that the LS solutions given by (7) and (8) have the same form as those derived for the optimal minimum mean squared estimates for A(k) and B(k) in [4,5].

Also, for the general case, when the relevant signals are stationary, applying the Wiener filter theory yields the optimum solution [6]

$$\tilde{H} = \tilde{R}_{xx}^{-1}\tilde{R}_{yx} \quad (12)$$

where

$$\tilde{H} = [a_0,a_1,a_2,\ldots,a_{N-1}, b_{0,0},b_{0,1},\ldots,b_{N-1,N-1}]^T$$

$$\tilde{R}_{xx} = E\{\tilde{X}(k)\tilde{X}^T(k)\}$$

$$\tilde{R}_{yx} = E\{y(k)\tilde{X}(k)\}$$

and

$$\tilde{X}(k) = [X(k),x(k-1),\ldots,x(k-N+1), x^2(k)-r_{xx}(0),X(k)X(k-1)-r_{xx}(1), \ldots, x^2(k-N+1) - r_{xx}(0)]^T .$$

From (12) it can be seen that computing the optimal solution requires inverting an $(N+N^2)x(N+N^2)$ matrix, which may be computationally very difficult for large values of N. In comparison, the solutions derived in (7) and (8) require inverting only an NxN matrix. The savings in computations involved is quite evident from the above discussion.

In the next section, we will develop an exponentially weighted SER algorithm for recursively computing $\hat{A}(k)$ and $\hat{B}(k)$.

## 3. THE EXPONENTIALLY WEIGHTED SER ALGORITHM

The SER algorithm has been used to update the optimum linear [2] and nonlinear [6] filter coefficients in a recursive manner. In this section, we adopt the LS solution in (7) and (8) and apply the SER algorithm to update the Volterra filter weights that minimize the cost functional given by (5) with the weighting function q(i,k) selected as

$$q(i,k) = (1-\beta)\beta^{k-i} \quad (13)$$

where $0 < \beta < 1$. This weighting function penalizes the current estimation errors more than the past ones. It may be noted here that for large values of k

$$\sum_{i=0}^{k} q(i,k) = 1-\beta^{k+1} \approx 1 , \quad (14)$$

and therefore the results in Section 2 are applicable here. Then, the LS solutions work out to

$$\overset{0}{A}(k) = \hat{R}_{xx}^{-1}(k)\ \hat{R}_{yx}(k) \tag{15}$$

and

$$\overset{0}{B}(k) = \frac{1}{2}\ \hat{R}_{xx}^{-1}(k)\ T_{yx}(k)\ \hat{R}_{xx}^{-1}(k) \tag{16}$$

where

$$\hat{R}_{yx}(k) = (1-\beta) \sum_{i=0}^{k} \beta^{k-i} y(i)X(i). \tag{17}$$

$$\hat{R}_{xx}(k) = (1-\beta) \sum_{i=0}^{k} \beta^{k-i} X(i)X^T(i) \tag{18}$$

and

$$T_{yx}(k) = (1-\beta) \sum_{i=0}^{k} \beta^{k-i} y(i)X(i)X^T(i)\ . \tag{19}$$

To obtain the recursive relationships, we proceed as follows: Substituting (17) in (15) and simplifying, we get

$$\hat{R}_{xx}(k)\hat{A}(k) = (1-\beta) \sum_{i=0}^{k-1} \beta^{k-i} y(i)X(i) + (1-\beta)y(k)X(k) \tag{20}$$

and

$$\left[(1-\beta) \sum_{i=0}^{k-1} \beta^{k-i} X(i)X^T(i)\right] \hat{A}(k-1) = (1-\beta) \sum_{i=1}^{k-1} \beta^{k-i} y(i)X(i)\ , \tag{21}$$

where we have used $\hat{A}(k)$ and $\hat{B}(k)$ instead of $\overset{0}{A}(k)$ and $\overset{0}{B}(k)$. Substituting (21) in (20) and manipulating the resultant equation, we obtain the recursive relationship for $\hat{A}(k)$ as

$$\hat{A}(k) = \hat{A}(k-1) + (1-\beta)\ \hat{R}_{xx}^{-1}(k)X(k)\ \left[y(k) - X^T(k)\hat{A}(k-1)\right]\ . \tag{22}$$

The derivation of the recursive relationship for the quadratic weights is more involved, even though straightforward. We will give only the final result. For details, the reader is referred to [3]. $\hat{B}(k)$ is related to $\hat{B}(k-1)$ as

$$\begin{aligned}\hat{B}(k) &= \frac{1}{\beta}\ \left[I-(1-\beta)\hat{R}_{xx}^{-1}(k)X(k)X^T(k)\right]\hat{B}(k-1)\left[I-(1-\beta)X(k)X^T(k)\hat{R}_{xx}^{-1}(k)\right] \\ &\quad + \frac{1}{2}\ (1-\beta)y(k)\hat{R}_{xx}^{-1}(k)X(k)X^T(k)\hat{R}_{xx}^{-1}(k)\ ,\end{aligned} \tag{23}$$

where I denotes the NxN identity matrix. Also, $\hat{R}_{xx}^{-1}(k)$ is computed using the recursive relationship [3]

$$\hat{R}_{xx}^{-1}(k) = \frac{1}{\beta}\ R_{xx}^{-1}(k-1) - \frac{1}{d(k)}\ \left[\frac{1-\beta}{\beta^2}\ \hat{R}_{xx}^{-1}(k-1)X(k)X^T(k)\hat{R}_{xx}^{-1}(k-1)\right] \tag{24}$$

where

$$d(k) = 1 + \frac{1-\beta}{\beta} X^T(k)\hat{R}_{xx}^{-1}(k-1)X(k) \tag{25}$$

and d(k) is assumed to be nonzero. In deriving (24) we made use of the fact that

$$\hat{R}_{xx}(k) = \beta\hat{R}_{xx}(k-1) + (1-\beta)X(k)X^T(k) \tag{26}$$

and then used the matrix inversion lemma [7] which says that if $A = B + CD$,

$$A^{-1} = B^{-1} - \frac{1}{1 + DB^{-1}C} B^{-1}CDB^{-1}$$

with $1 + DB^{-1}C \neq 0$.

In the next section, we will demonstrate the usefulness of the proposed algorithm with a simulation example.

## 4. A SIMULATION EXAMPLE

To study the performance of the proposed algorithm, we consider the system identification problem for a second order Volterra filter whose coefficients A and B are given by

$$A = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} 0.6 \\ -0.2 \end{bmatrix} \tag{27}$$

and

$$B = \begin{bmatrix} b_{0,0} & b_{0,1} \\ b_{1,0} & b_{1,1} \end{bmatrix} = \begin{bmatrix} 0.3 & -0.1 \\ -0.1 & 0.15 \end{bmatrix} . \tag{28}$$

Note that B is symmetric. The system identification problem is schematically depicted in Fig. 1, where the reference input x(k) is white, Gaussian with zero mean and unit variance and y(k) is obtained from x(k) using (2) with A and B given by (27) and (28), respectively. The algorithm was initialized with

$$\hat{A}(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

$$\hat{B}(0) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \text{ and}$$

$$\hat{R}_{xx}(0) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} .$$

Thirty independent simulations were run using 2000 data samples each and the results presented are averaged over the thirty runs. Figure 2a-2e display plots of $\hat{a}_0(k)$, $\hat{a}_1(k)$, $\hat{b}_{00}(k)$, $\hat{b}_{01}(k) = \hat{b}_{10}(k)$ and $\hat{b}_{11}(k)$, respectively. In each case, we can see that the algorithm converges to the correct values. Also, the plot of the squared estimation error as a function of time in Fig. 2f shows that the extimation error decreases exponentially with time.

## 5. SUMMARY AND CONCLUSIONS

The adaptive second order Volterra filter with the SER algorithm was developed in this paper. The recursive algorithm tracks the LS solution that minimizes the given cost functional. The simulation results presented demonstrated the effectiveness of the algorithm in estimating the system parameters of a second order Volterra filter.

The Gaussian assumption for the reference input reduces the computational load considerably over the direct approach given by (12). The theoretical convergence properties of the adaptive filter are currently under investigation and will be the subject of a later paper.

## REFERENCES


1. M. Schetzen, The Volterra and Wiener Theories of the Nonlinear System, Wiley, New York, 1980.

2. N. Ahmed, et al., "A Short-term Sequential Regression Algorithm," IEEE Trans. Acoust. Speech, Signal Proc., Vol. ASSP-27, No. 5, October 1979.

3. K.K. Yu, "Nonlinear Adaptive Digital Filter with Sequential Regression (SER) Algorithm," Master's Thesis, University of Iowa, December 1984.

4. T. Koh and E.J. Powers, "An Adaptive Nonlinear Digital Filter with Lattice Orthogonalization," Proc. ICASSP 83, pp. 37-40, Boston, 1983.

5. T. Koh, E.J. Powers and R.W. Miksad, "An Approach to Time Domain Modelling of Nonlinear Drift Oscillations in Random Seas," Proc. Int. Symp. Offshore Engineering, Rio de Janeiro, Brazil, 1983.

6. D. Graupt, Identification of Systems, Van Nostrand, 1972.

7. D.K. Faddeeva and V.N. Faddeeva, Computational Methods of Linear Algebra, W.H. Freeman and Co., San Francisco, 1963.

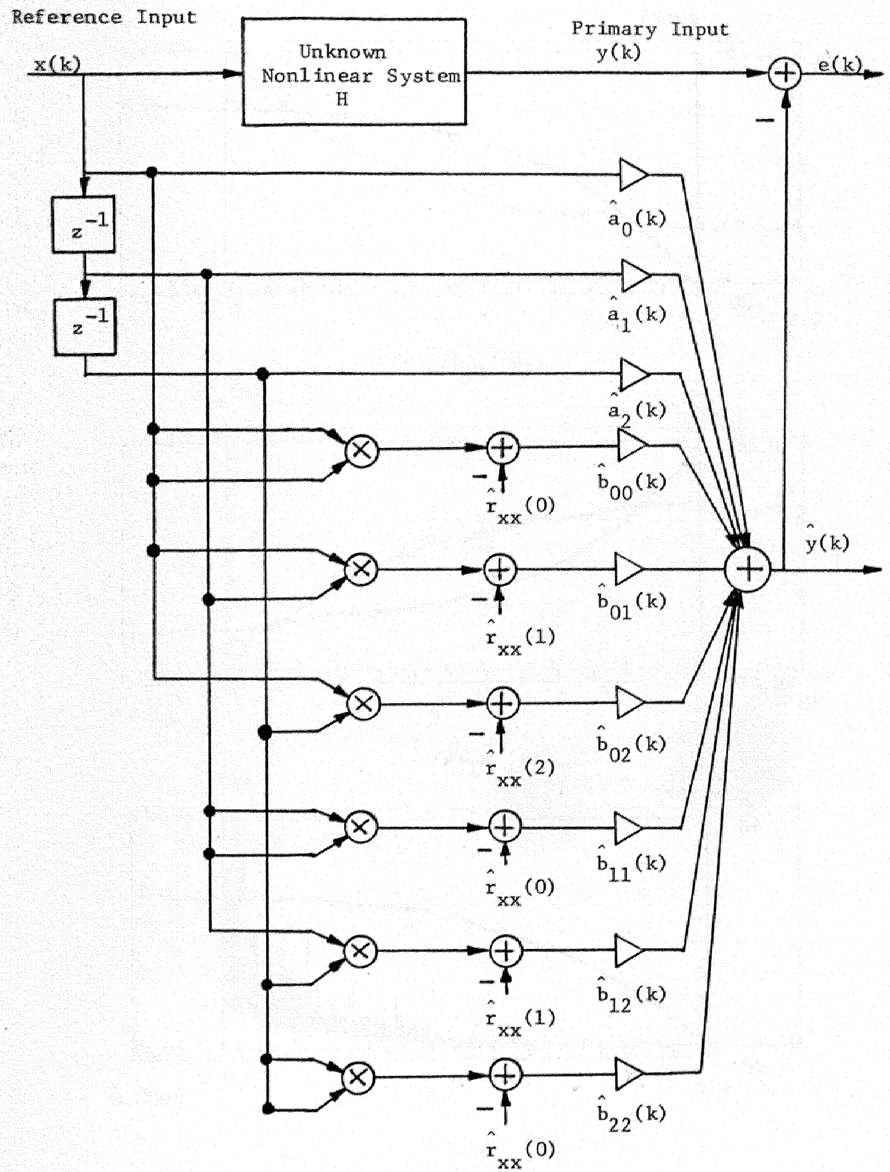



Figure 1. Tapped delay line model realization of a second order Volterra filter for the system identification problem.

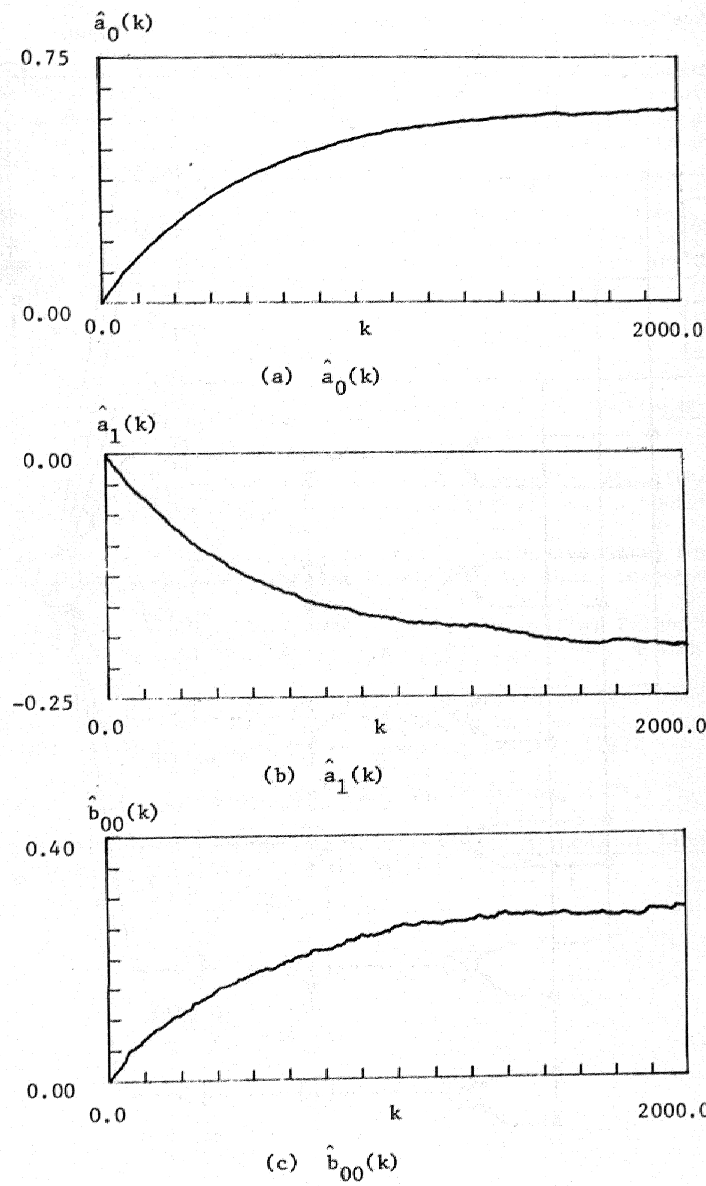
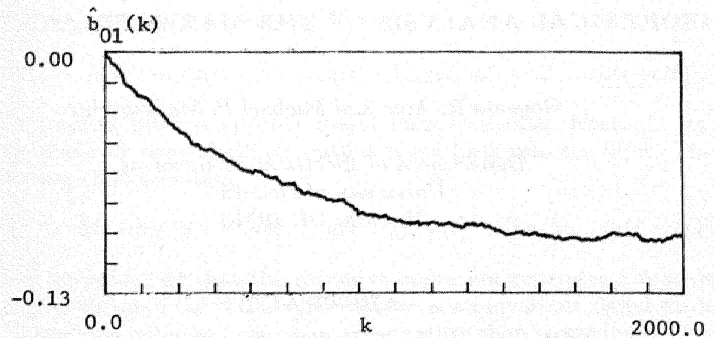
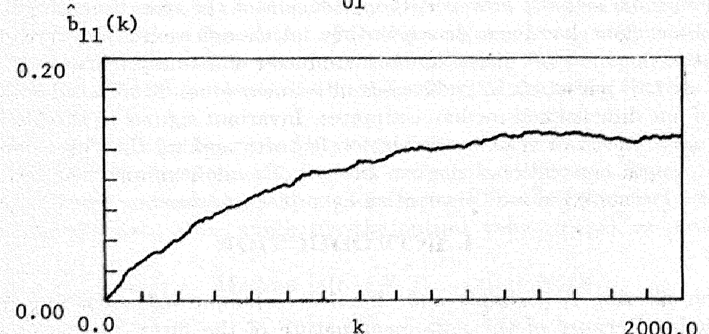



Figure 2. Ensemble averages of the system identification problem in the simulation example ($\beta = 0.998$).




(d) $\hat{b}_{01}(k)$




(e) $\hat{b}_{11}(k)$

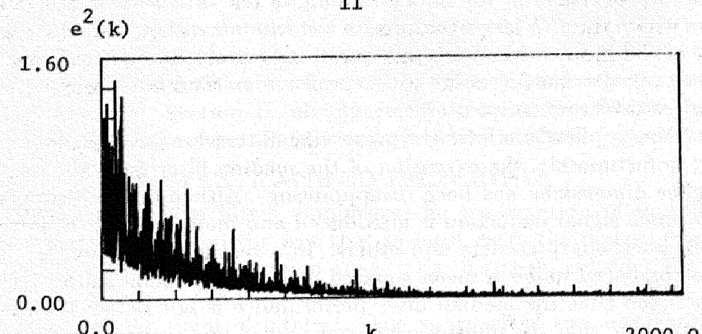



Figure 2. (cont.)