Proceedings of the 36th
Conference on Decision & Control
San Diego, California USA • December 1997

Time-Domain Identification Using ARMARKOV/Toeplitz Models With Quasi-Newton Update

James C. Akers and Dennis S. Bernstein
Department of Aerospace Engineering
The University of Michigan
Ann Arbor, MI 48109-2140

1 Introduction

Recursive identification methods using time-domain data have been developed in [1, 2] utilizing a gradient-based identification technique for estimating the Markov parameters of a system. This identification technique utilizes the ARMARKOV representation of a time-invariant finite-dimensional system which relates the current output of a system to past outputs as well as current and past inputs. While the ARMARKOV representation has the same form as an ARMA representation, the ARMARKOV representation explicitly contains Markov parameters of the system.

Appropriate "stacking" of time-delayed ARMARKOV representations yields a block-Toeplitz weight matrix which contains Markov parameters and which maps a vector of past outputs and inputs to a vector of current and past outputs. The recursive update law given in [1] is based upon a gradient that preserves the block-zero structure of the block-Toeplitz weight matrix. In the presence of a persistent input sequence, this gradient method guarantees that the estimated weight matrix converges to the actual weight matrix.

In this paper, we introduce a quasi-Newton method that utilizes a more efficient quasi-Newton update direction to estimate the Markov parameters recursively from time-domain input-output data. The step size is given by an explicit expression analogous to the optimal step size derived for the gradient method.

2 ARMARKOV Representations

Consider the discrete-time finite-dimensional linear time-invariant system

$$x(k+1) = Ax(k) + Bu(k), \qquad (2.1)$$

$$y(k) = Cx(k) + Du(k), \qquad (2.2)$$

where $A \in \mathcal{R}^{n\times n}$, $B \in \mathcal{R}^{n\times m}$, $C \in \mathcal{R}^{l\times n}$, and $D \in \mathcal{R}^{l\times m}$. The *Markov parameters* H_j are defined by

$$\begin{aligned} H_j &\triangleq D, \qquad j=-1, \\ &\triangleq CA^jB, \qquad j\geq 0, \end{aligned} \qquad (2.3)$$

and satisfy

$$G(z) \triangleq C(zI-A)^{-1}B + D = \sum_{j=-1}^{\infty} H_j z^{-(j+1)}. \qquad (2.4)$$

The ARMA transfer function representation of $G(z)$ is given by

$$G(z) = \frac{1}{z^n + a_1 z^{n-1} + \cdots + a_n}\left(B_0 z^n + B_1 z^{n-1} + \cdots + B_n\right), \qquad (2.5)$$

where $\det(zI - A) = z^n + a_1 z^{n-1} + \cdots + a_n$ and $B_i \in \mathcal{R}^{l\times m}$, $i = 0,\ldots,n$. Although (2.5) provides a rational representation of $G(z)$, it contains only the first Markov parameter $B_0 = H_{-1}$.

The ARMA time-domain representation of $G(z)$ corresponding to (2.5) is given by

$$y(k) = -a_1 y(k-1) - \cdots - a_n y(k-n) + B_0 u(k) + \cdots + B_n u(k-n). \qquad (2.6)$$

Replacing k by $k-1$ in (2.6) and substituting $\mu - 1$ times yields the *ARMARKOV time-domain representation* of $G(z)$

$$\begin{aligned} y(k) = & \sum_{j=1}^{n} -\alpha_{\mu,j} y(k-\mu-j+1) + \\ & \sum_{j=1}^{\mu} H_{j-2} u(k-j+1) + \\ & \sum_{j=1}^{n} \mathcal{B}_{\mu,j} u(k-\mu-j+1), \end{aligned} \qquad (2.7)$$

where $\alpha_{\mu,1},\ldots,\alpha_{\mu,n} \in \mathcal{R}$ and $\mathcal{B}_{\mu,1},\ldots,\mathcal{B}_{\mu,n} \in \mathcal{R}^{l\times m}$. Note that (2.7) involves the first μ Markov parameters $H_{-1},\ldots,H_{\mu-2}$. Furthermore, note that (2.6) is a specia case of (2.7) with $\mu = 1$.

Defining the *ARMARKOV regressor vector* $\Phi(k) \in \mathcal{R}^{(p+n-1)(l+m)+\mu m}$ by

$$\Phi(k) \triangleq \begin{bmatrix} y(k-\mu) \\ \vdots \\ y(k-\mu-p-n+2) \\ u(k) \\ \vdots \\ u(k-\mu-p-n+2) \end{bmatrix}, \qquad (2.8)$$

it follows that

$$Y(k) = W\Phi(k), \tag{2.9}$$

where the *ARMARKOV/Toeplitz weight matrix* W is the block-Toeplitz matrix defined by

$$W \triangleq \begin{bmatrix} -\mathcal{A}_\mu & 0_l & \cdots & 0_l & H_{-1} & \cdots & H_{\mu-2} & \mathcal{B}_\mu & 0_{l\times m} & \cdots & 0_{l\times m} \\ 0_l & \ddots & \ddots & \vdots & 0_{l\times m} & \ddots & & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0_l & \vdots & \ddots & \ddots & & \ddots & \ddots & 0_{l\times m} \\ 0_l & \cdots & 0_l & -\mathcal{A}_\mu & 0_{l\times m} & \cdots & 0_{l\times m} & H_{-1} & \cdots & H_{\mu-2} & \mathcal{B}_\mu \end{bmatrix}, \tag{2.10}$$

where

$$\mathcal{A}_\mu \triangleq [\alpha_{\mu,1}I_l \quad \cdots \quad \alpha_{\mu,n}I_l], \mathcal{B}_\mu \triangleq [\mathcal{B}_{\mu,1} \quad \cdots \quad \mathcal{B}_{\mu,n}].$$

Note that p determines the window of input-output data that appears in (2.8).

It follows from the ARMARKOV time-domain representation (2.7) that an *ARMARKOV transfer function representation* of $G(z)$ with μ Markov parameters is given by

$$G(z) = \frac{1}{z^{\mu+n-1} + \alpha_{\mu,1}z^{n-1} + \cdots + \alpha_{\mu,n}} \times$$
$$(H_{-1}z^{\mu+n-1} + \cdots + H_{\mu-2}z^n + \mathcal{B}_{\mu,1}z^{n-1} + \cdots + \mathcal{B}_{\mu,n}) \tag{2.11}$$

This representation of $G(z)$ can be viewed as a blending of the Markov parameter representation (2.4) and the ARMA transfer function representation (2.5), which correspond to $\mu = \infty$ and $\mu = 1$, respectively. The ARMA transfer function representation and the ARMARKOV transfer function representation are different representations of $G(z)$. However, the ARMARKOV transfer function representation, which is nonminimal when $\mu > 1$, allows direct estimation of the Markov parameters. Note, however, that the ARMARKOV transfer function representation is not equivalent to an arbitrary nonminimal ARMA representation since the coefficients of $z^{\mu+n-2}, \ldots, z^n$ in the denominator are constrained to be zero.

Henceforth, for convenience we omit the subscript μ and write $\mathcal{A}$, $\mathcal{B}$, α_i, and β_i for $\mathcal{A}_\mu$, $\mathcal{B}_\mu$, $\alpha_{\mu,i}$, and $\beta_{\mu,i}$, respectively.

3 Gradient Method

In this section we summarize several results given in [1]. Let $\widehat{W}(k)$ denote an estimate of the ARMARKOV/Toeplitz weight matrix W at time k, where $\widehat{W}(k)$ has the same block-zero structure as W. Let $\widehat{Y}(k)$ denote the *estimated output vector* defined by

$$\widehat{Y}(k) \triangleq \widehat{W}(k)\Phi(k) \in \mathcal{R}^{pl}. \tag{3.1}$$

Furthermore, define the *output error* $\varepsilon(k) \in \mathcal{R}^{pl}$ by

$$\varepsilon(k) \triangleq Y(k) - \widehat{Y}(k), \tag{3.2}$$

and the *output error cost function* $J(k)$ by

$$J(k) \triangleq \frac{1}{2}\varepsilon^{\mathrm{T}}(k)\varepsilon(k). \tag{3.3}$$

Lemma 3.1 The gradient of $J(k)$ with respect to the estimated weight matrix $\widehat{W}(k)$ is given by

$$\frac{\partial J(k)}{\partial \widehat{W}(k)} = -U \circ \left[\varepsilon(k)\Phi^{\mathrm{T}}(k)\right], \tag{3.4}$$

where $U \in \mathcal{R}^{pl\times[(p+n-1)(l+m)+\mu m]}$ is a one-zero matrix defined in [1].

We now consider the *estimated weight matrix update law*

$$\widehat{W}(k+1) = \widehat{W}(k) - \eta(k)\frac{\partial J(k)}{\partial \widehat{W}(k)}, \tag{3.5}$$

where $\eta(k) \geq 0$ is the *adaptive step size.* Furthermore, define the *estimated weight matrix error* by

$$E(k) \triangleq W - \widehat{W}(k), \tag{3.6}$$

and the *estimated weight matrix error cost function*

$$\mathcal{J}(k, \eta(k)) \triangleq \|E(k+1)\|_{\mathrm{F}}^2 - \|E(k)\|_{\mathrm{F}}^2. \tag{3.7}$$

Then it follows from the estimated weight matrix update law (3.5) that

$$E(k+1) = E(k) + \eta(k)\frac{\partial J(k)}{\partial \widehat{W}(k)} \tag{3.8}$$

and

$$\varepsilon(k) = E(k)\Phi(k). \tag{3.9}$$

Let the *optimal adaptive step size* $\eta_{\mathrm{opt}}(k)$ be defined by

$$\eta_{\mathrm{opt}}(k) \triangleq \frac{\|\varepsilon(k)\|_2^2}{\left\|\frac{\partial J(k)}{\partial \widehat{W}(k)}\right\|_{\mathrm{F}}^2}. \tag{3.10}$$

The following result shows that $\eta_{\mathrm{opt}}(k)$ minimizes $\mathcal{J}(k, \eta(k))$.

Theorem 3.1 Let $\widehat{W}(0)$ have the same block-zero structure as W and consider the estimated weight matrix update law (3.5). Assume that $\frac{\partial J(k)}{\partial \widehat{W}(k)} \neq 0$, $k \geq 0$, and assume that the adaptive step size $\eta(k)$ satisfies

$$0 < \eta(k) < 2\eta_{\mathrm{opt}}(k), \quad k \geq 0. \tag{3.11}$$

Then $\{\|E(k)\|_F\}_{k=0}^{\infty}$ is decreasing, and thus $\mathcal{J}(k,\eta(k)) < 0$, $k \geq 0$. Furthermore, for all $k \geq 0$, $\eta(k) = \eta_{\rm opt}(k)$ minimizes $\mathcal{J}(k,\eta(k))$, and

$$\mathcal{J}(k,\eta_{\rm opt}(k)) = -\|\varepsilon(k)\|_2^2\eta_{\rm opt}(k). \quad (3.12)$$

If, in addition,

$$\sup_{k\geq 0}\left|\frac{\eta(k)}{\eta_{\rm opt}(k)} - 1\right| < 1 \quad (3.13)$$

and

$$\sup_{k\geq 0}\|\Phi(k)\|_2 < \infty, \quad (3.14)$$

then

$$\sum_{k=0}^{\infty}\|\varepsilon(k)\|_2^2 < \infty. \quad (3.15)$$

Consequently,

$$\lim_{k\to\infty}\varepsilon(k) = 0, \quad (3.16)$$

and

$$\lim_{k\to\infty}\frac{\partial J(k)}{\partial \widehat{W}(k)} = 0. \quad (3.17)$$

Remark 3.1 Note that (3.13) implies (3.11), whereas the converse is not true. By choosing the adaptive step size $\eta(k)$ to be $\eta(k) = \alpha\eta_{\rm opt}(k)$, $\alpha \in (0,2)$, it follows that (3.13) is satisfied and thus the conclusions of Theorem 3.1 hold.

Next, assuming $\Phi(k) \neq 0$, define the *computationally efficient step size* $\eta_{\rm eff}(k)$ by

$$\eta_{\rm eff}(k) \triangleq \frac{1}{\|\Phi(k)\|_2^2}. \quad (3.18)$$

Note that $\eta_{\rm eff}(k)$ does not involve either $\frac{\partial J(k)}{\partial \widehat{W}(k)}$ or $\varepsilon(k)$ both of which are needed to compute $\eta_{\rm opt}(k)$. It is showed in [1] that

$$\eta_{\rm eff}(k) \leq \eta_{\rm opt}(k). \quad (3.19)$$

A corollary of Theorem 3.1 with $\eta_{\rm opt}(k)$ is replaced by $\eta_{\rm eff}(k)$ is given in [1].

As shown in [1], $\{\widehat{W}(k)\}_{k=0}^{\infty}$ given by the estimated weight matrix update law (3.5) is guaranteed to converge to W if $\{u(k)\}_{k=0}^{\infty}$ satisfies a persistent excitation condition. Next we consider a variation of the estimated weight matrix update law (3.5).

4 Quasi-Newton Method

The gradient method is a descent method since the estimated weight matrix update law (3.5) utilizes the update direction $-\frac{\partial J(k)}{\partial \widehat{W}(k)}$. In this section we introduce a quasi-Newton method that uses an AR-MARKOV/Toeplitz representation with a more efficient quasi-Newton update direction. For convenience, we consider SISO systems. Let

$$\widehat{\mathcal{A}}(k) \triangleq [\widehat{\alpha}_1(k) \ \cdots \ \widehat{\alpha}_n(k)], \widehat{\mathcal{B}}(k) \triangleq [\widehat{\beta}_1(k) \ \cdots \ \widehat{\beta}_n(k)],$$

and $\widehat{\mathcal{H}}_j$ denote parameter estimates and let $P(k) \in \mathcal{R}^{2n+\mu}$ denote the *parameter vector* defined by

$$P(k) \triangleq \Big[\widehat{\alpha}_1(k) \ \cdots \ \widehat{\alpha}_n(k) \ \widehat{H}_{-1}(k) \ \cdots \ \widehat{H}_{\mu-2}(k) \ \widehat{\beta}_1(k) \ \cdots \ \widehat{\beta}_n(k)\Big]^{\rm T}. \quad (4.1)$$

Then the gradient $\nabla J(k) \in \mathcal{R}^{(2n+\mu)}$ of the output error cost function $J(k)$ with respect to $P(k)$ is given by

$$\nabla J(k) = \left[\frac{\partial J(k)}{\partial\widehat{\alpha}_1(k)} \ \cdots \ \frac{\partial J(k)}{\partial\widehat{\alpha}_n(k)} \ \frac{\partial J(k)}{\partial\widehat{H}_{-1}(k)} \ \cdots \ \frac{\partial J(k)}{\partial\widehat{H}_{\mu-2}(k)} \ \frac{\partial J(k)}{\partial\widehat{\beta}_1(k)} \ \cdots \ \frac{\partial J(k)}{\partial\widehat{\beta}_n(k)}\right]^{\rm T}. \quad (4.2)$$

To compute $\nabla J(k)$ it is convenient to define

$$\begin{aligned} L_j &\triangleq [I_p \ \ 0_{p\times(n-1)} \ \ 0_{p\times(p+n+\mu-1)}], \ j=1,\\ &\triangleq [0_{p\times(j-1)} \ \ I_p \ \ 0_{p\times(n-j)} \ \ 0_{p\times(p+n+\mu-1)}], \ j\geq 2,\\ R_j &\triangleq [0_{p\times(p+n+j-2)} \ \ I_p \ \ 0_{p\times(n+\mu-j)}], \ j=1,\ldots,\mu,\\ T_j &\triangleq [0_{p\times(p+n+\mu+j-2)} \ \ I_p \ \ 0_{p\times(n-j)}], \ j=1,\ldots,n-1,\\ &\triangleq [0_{p\times(p+2n+\mu-2)} \ \ I_p], \ j=n. \end{aligned}$$

Then $\widehat{W}(k)$ can be written as

$$\widehat{W}(k) = \sum_{j=1}^{n}\widehat{\alpha}_j(k)L_j + +\sum_{j=1}^{\mu}\widehat{H}_{j-2}(k)R_j + \sum_{j=1}^{n}\widehat{\beta}_j(k)T_j. \quad (4.3)$$

Hence, it follows that

$$\frac{\partial J(k)}{\partial\widehat{\alpha}_j(k)} = -\varepsilon^{\rm T}(k)L_j\Phi(k), \ j=1,\ldots,n, \quad (4.4)$$

$$\frac{\partial J(k)}{\partial\widehat{H}_{j-2}(k)} = -\varepsilon^{\rm T}(k)R_j\Phi(k), \ j=1,\ldots,\mu, \quad (4.5)$$

$$\frac{\partial J(k)}{\partial\widehat{\beta}_j(k)} = -\varepsilon^{\rm T}(k)T_j\Phi(k), \ j=1,\ldots,n. \quad (4.6)$$

Therefore, $\nabla J(k)$ is given by

$$\nabla J(k) = -\begin{bmatrix} \varepsilon^{\mathrm{T}}(k)L_1 \\ \vdots \\ \varepsilon^{\mathrm{T}}(k)L_n \\ \varepsilon^{\mathrm{T}}(k)R_1 \\ \vdots \\ \varepsilon^{\mathrm{T}}(k)R_\mu \\ \varepsilon^{\mathrm{T}}(k)T_1 \\ \vdots \\ \varepsilon^{\mathrm{T}}(k)T_n \end{bmatrix} \Phi(k). \tag{4.7}$$

Now we consider the quasi-Newton update direction which has the form $-F^{-1}(k)\nabla J(k)$, where $F^{-1}(k)$ is a symmetric positive-definite approximation to the inverse of the Hessian of $J(k)$. For convenience let $s(k)$ denote the difference between the current and past parameter vectors defined by

$$s(k) \triangleq P(k) - P(k-1), \tag{4.8}$$

and let $z(k)$ denote the difference between the current and past gradients defined by

$$z(k) \triangleq \nabla J(k) - \nabla J(k-1). \tag{4.9}$$

The BFGS inverse Hessian update is given by

$$F^{-1}(k+1) = F^{-1}(k) + \frac{s(k)s^{\mathrm{T}}(k)}{s^{\mathrm{T}}(k)z(k)} - \frac{F^{-1}(k)z(k)z^{\mathrm{T}}(k)F^{-1}(k)}{z^{\mathrm{T}}(k)F^{-1}(k)z(k)}. \tag{4.10}$$

The quasi-Newton method uses the BFGS update (4.10) with the *parameter vector update law*

$$P(k+1) = P(k) - \eta_{\mathrm{qn}}(k)F^{-1}(k)\nabla J(k), \tag{4.11}$$

where the step size $\eta_{\mathrm{qn}}(k)$ is defined by

$$\eta_{\mathrm{qn}}(k) = \frac{\|\varepsilon(k)\|_2^2}{\|\nabla J(k)\|_{\mathrm{F}}^2}. \tag{4.12}$$

The expression (4.12) is analogous to the optimal adaptive step size $\eta_{\mathrm{opt}}(k)$ given by (3.10). In the case $p = 1$ it follows that $\frac{\partial J(k)}{\partial \widehat{W}(k)} = \nabla J(k)$ and thus $\eta_{\mathrm{qn}}(k) = \eta_{\mathrm{opt}}(k)$. Furthermore, if $p > 1$ it can be shown that $\|\nabla J(k)\|_{\mathrm{F}} \leq \left\|\frac{\partial J(k)}{\partial \widehat{W}(k)}\right\|_{\mathrm{F}}$ and thus $\eta_{\mathrm{qn}}(k) \leq \eta_{\mathrm{opt}}(k)$. Since the cost function $J(k)$ changes with each update the quasi-Newton search direction may become inappropriate after a large number of updates. Therefore, we reset the Hessian to the identity whenever $\|\varepsilon(k+1)\|_2 > \|\varepsilon(k)\|_2$.

5 Numerical Example

In this section Markov parameters of a second-order asymptotically stable SISO system are estimated using both the gradient method and the quasi-Newton method. The gradient method uses an adaptive step size $\eta(k) = \eta_{\mathrm{opt}}(k)$, and the estimated Markov parameters are obtained after each update of the estimated ARMARKOV/Toeplitz weight matrix $\widehat{W}(k)$ by averaging over the corresponding entries of $\widehat{W}(k)$. Since a second-order system requires a minimum of five Markov parameters, we choose $\mu = 6$.

$$\widehat{G}(s) = \frac{\omega_{\mathrm{n}}^2}{s^2 + 2\zeta\omega_{\mathrm{n}}s + \omega_{\mathrm{n}}^2} \tag{5.1}$$

with $\omega_{\mathrm{n}} = 6.28$ rad/sec and $\zeta = 1\%$. We use a balanced realization of the zero-order-hold discretization of (5.1) at a sampling frequency of 100 Hz. The input $u(k)$ is chosen to be zero-mean white noise uniformly distributed on $[-1, 1]$. The input was applied to (5.1) for 10 seconds ($-10 \leq k \leq 990$) with zero initial conditions so that data for $\Phi(0)$ is available. As shown in [1] $u(k)$ is persistent with respect to $G(z)$ for the available data.

First we consider $\mu = 6$, $n = 2$, and $p = 4$. The values of $\|\varepsilon(k)\|_2$, $\left\|\frac{\partial J(k)}{\partial \widehat{W}(k)}\right\|_{\mathrm{F}}$, and $\eta(k)$ for the gradient method are shown in Figure 1 and the values of $\|\varepsilon(k)\|_2$ for both methods are shown in Figure 2. While the quasi-Newton method converges faster than the gradient method, it can be seen in Figure 3 that the quasi-Newton method uses approximately 5 times the number of floating point operations (flops) compared to the gradient method. The fluctuations in the flop count of the quasi-Newton method are due to the resetting of the Hessian to the identity every few time steps. The six estimated Markov parameters obtained over the first 1 second or 100 time steps are shown in Figure 4. Although not shown, the quasi-Newton method does not converge if the inverse Hessian update is not reset.

Next we consider $\mu = 6$, $n = 2$, and $p = 20$. The values of $\|\varepsilon(k)\|_2$ for both methods are shown in Figure 5, the number of flops is shown in Figure 6, and the six estimated Markov parameters are shown in Figure 7. The quasi-Newton method converges significantly faster than the gradient method while using approximately 3.5 times the number of flops compared to the gradient method.

Finally, we consider a third variation of the quasi-Newton approach. In this case we allow the algorithm to do a complete quasi-Newton optimization procedure at each time step k. The *optimal quasi-Newton method* thus takes full advantage of the data at time step k. We consider $\mu = 6$, $n = 2$, and $p = 50$. The values of $\|\varepsilon(k)\|_2$ for all three algorithms are shown in Figure 8, the number of flops is shown in Figure 9, and the six estimated Markov parameters are shown in Figure 10. Again it can be seen that the quasi-Newton method converges faster than the gradient method. The optimal quasi-Newton method has the fastest rate of convergence. After the fifth time step the Markov parameters obtained by the optimal quasi-Newton method have converged to within 1% of their respective true values. However, this high rate of convergence was achieved at the expense of using more than an order of magnitude more flops than the quasi-Newton method.

References

[1] J. C. Akers and D. S. Bernstein, "Time-Domain Identification Using ARMARKOV/Toeplitz Models," *Proc. Amer. Contr. Conf.*, Albuquerque, NM, June, 1997, pp. 186 - 190.

[2] J. C. Akers and D. S. Bernstein, "ARMARKOV Least-Squares Identification," *Proc. Amer. Contr. Conf.*, Albuquerque, NM, June, 1997, pp. 191 - 195.

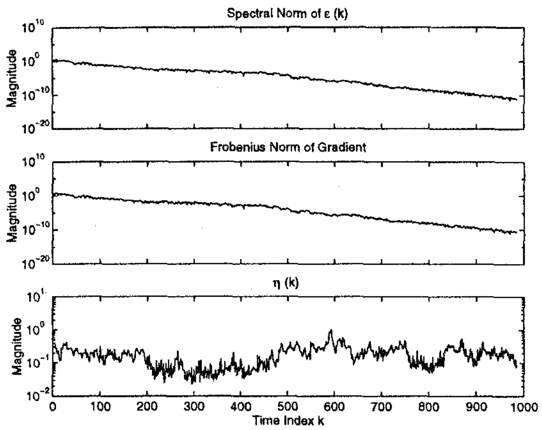

Figure 1: Error, gradient, and step size of the gradient method with $\mu = 6$, $n = 2$, and $p = 4$.

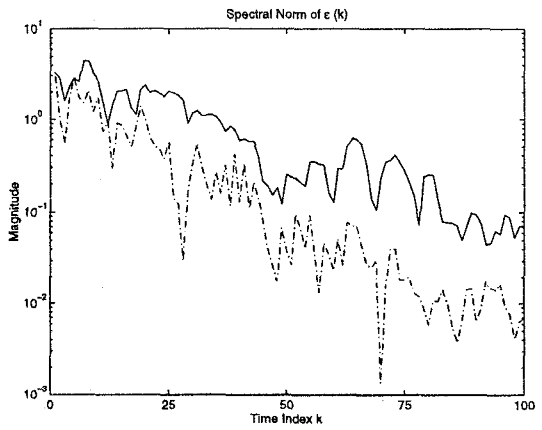

Figure 2: Values of $\|\varepsilon(k)\|_2$ of the gradient method (solid line) and the quasi-Newton method (dash-dot line) with $\mu = 6$, $n = 2$, and $p = 4$.

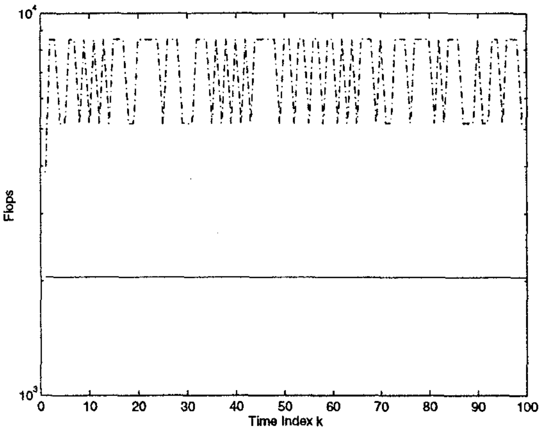

Figure 3: Floating point operations of the gradient method (solid line) and the quasi-Newton method (dash-dot line) with $\mu = 6$, $n = 2$, and $p = 4$.

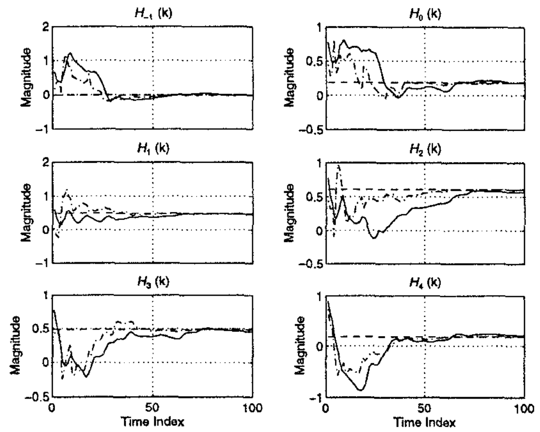

Figure 4: Markov parameter estimates obtained from the gradient method (solid line) and the quasi-Newton method (dash-dot line) with $\mu = 6$, $n = 2$, and $p = 4$.

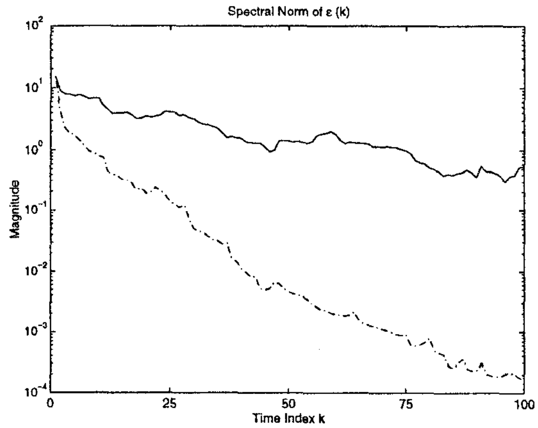

Figure 5: Values of $\|\varepsilon(k)\|_2$ of the the gradient method (solid line) and the quasi-Newton method (dash-dot line) with $\mu = 6$, $n = 2$, and $p = 20$.

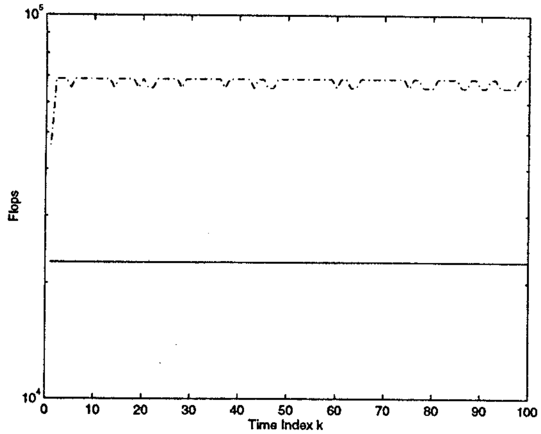

Figure 6: Floating point operations of the gradient method (solid line) and the quasi-Newton method (dash-dot line) with $\mu = 6$, $n = 2$, and $p = 20$.

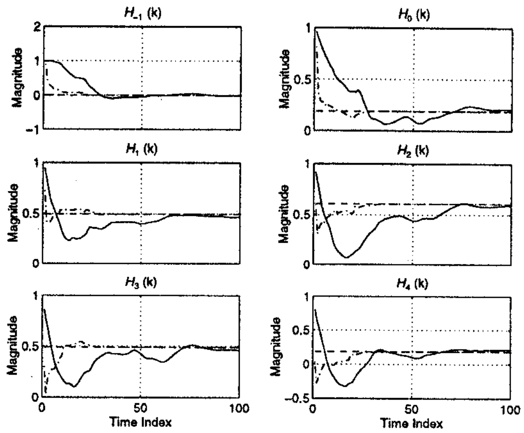

Figure 7: Markov parameter estimates obtained from the gradient method (solid line) and the quasi-Newton method (dash-dot line) with $\mu = 6$, $n = 2$, and $p = 20$.

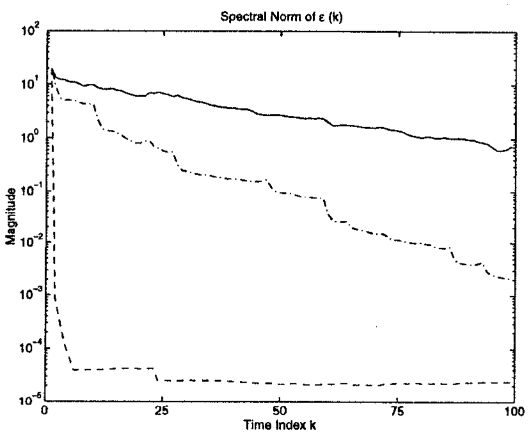

Figure 8: Values of $\|\varepsilon(k)\|_2$ of the gradient method (solid line) , the optimal quasi-Newton method (dashed line), and the quasi-Newton method (dash-dot line) with $\mu = 6$, $n = 2$, and $p = 50$.

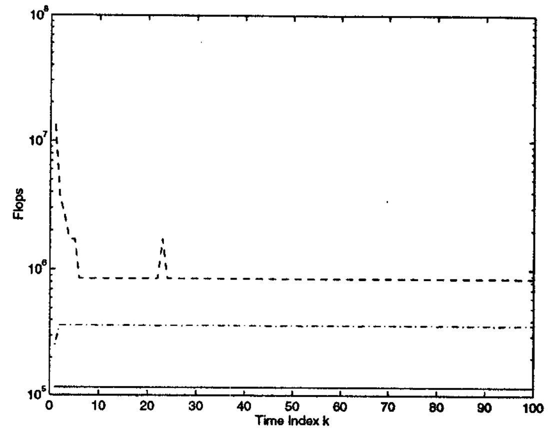

Figure 9: Floating point operations of the gradient method (solid line) , the optimal quasi-Newton method (dashed line), and the quasi-Newton method (dash-dot line) with $\mu = 6$, $n = 2$, and $p = 50$.

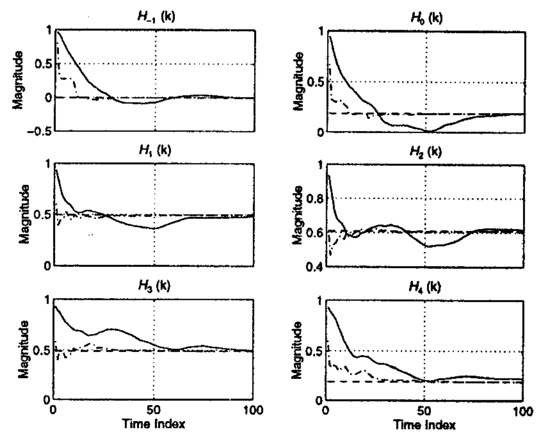

Figure 10: Markov parameter estimates obtained from the gradient method (solid line) , the optimal quasi-Newton method (dashed line), and the quasi-Newton method (dash-dot line) with $\mu = 6$, $n = 2$, and $p = 50$.