Proc. 8th IFAC/IFORS Symposium on Identification and System Parameter Estimation, Beijing, 1988

IDENTIFICATION OF A TAPPED LATTICE REPRESENTATION OF A LINEAR SYSTEM

C. Richard Johnson, Jr.[1]

School of Electrical Engineering, Phillips Hall, Cornell University, Ithaca, NY 14853 USA.

Brian D. O. Anderson

Department of Systems Engineering, Australian National University, GPO Box 4, Canberra, ACT 2601 AUSTRALIA.

Abstract. This paper provides a formulation of a tractable error system of an adaptive algorithm for output error identification of the parameters of a tapped lattice model of a linear, infinite-impulse-response (IIR) plant. This algorithm is an attractive solution to the underlying output error problem due to (i) the simplicity of its crucial stability check and maintenance procedure for the adapted IIR parameterization and (ii) the numerical insensitivity properties of the lattice structure. Our error system reformulation allows proof of this adaptive tapped lattice identifier's local convergence and robustness with the use of nonlinear stability theory that has been fashioned elsewhere for analysis of other adaptive systems. Reproducible simulation evidence is presented that supports such results.

Keywords. Identification; adaptive systems; sensitivity analysis; stability criteria; modelling

1. Introduction

"People will remember you better if you always wear the same outfit."
- David Byrne

Recursive identification of a recursive, pole-zero model is a classic adaptive parameter estimation problem. System identification specialists know this problem by its output error formulation label [1]. Signal processing types would likely call it an adaptive infinite impulse response (IIR) filter problem [2].

A simple description of this problem could be based on an assumed linear system description for generating a measurable output y from a measurable input u as in

$$y(k) = \sum_{i=1}^{n} [a_i y(k-i) + b_i u(k-i)] \qquad (1.1)$$

with the parameters a_i and b_i unknown. With the similarly structured, direct form, recursive, pole-zero model

$$\hat{y}(k) = \sum_{i=1}^{n} [\hat{a}_i(k)\hat{y}(k-i) + \hat{b}_i(k)u(k-i)], \qquad (1.2)$$

the parameters of which are to be adapted, the prediction error can be written as

$$e(k) = y(k) - \hat{y}(k) = X^T(k)\tilde{\theta}(k) + \sum_{i=1}^{n} a_i e(k-i)$$

$$= [1-A(q^{-1})]^{-1}[X^T(k)\tilde{\theta}(k)], \qquad (1.3)$$

where

$$X(k) = [\hat{y}(k-1) \cdots \hat{y}(k-n) \quad u(k-1) \cdots u(k-n)]^T \qquad (1.4)$$

and

$$\tilde{\theta}(k) = [a_1 - \hat{a}_1(k) \cdots a_n - \hat{a}_n(k) \; b_1 - \hat{b}_1(k) \cdots b_n - \hat{b}_n(k)]^T. \qquad (1.5)$$

This prediction error e, and the accompanying regressor X, contain the information usable to adjust the $\hat{a}_i$ and $\hat{b}_i$ so $\hat{y}$ best matches y. It is the use of past $\hat{y}$ in the identifier/predictor of (1.2) that earns it the label of output error identifier [1] or adaptive IIR filter [2]. This use of $\hat{y}$ in (1.2) also causes the prediction error e in (1.3) to be nonlinear in the parameters $\hat{\theta}$ (or $\tilde{\theta}$). This complicates the problem, as we will see.

The presence of $[1-A]^{-1}$ in (1.3) also complicates the problem. If the prediction error in (1.3) were simply $X^T\tilde{\theta}$ rather than a filtered version $H[X^T\tilde{\theta}]$, a candidate parameter adaptation algorithm would be easily developed. For example, if $A = 0$, and $\hat{A} = 0$ so that (1.1) and (1.2) reduces to a finite impulse response (FIR) filter and (1.3) reduces to $e = X^T\tilde{\theta}$, a standard algorithm for updating the $\hat{b}_i$ is LMS [3]

$$\hat{b}_i(k+1) = \hat{b}_i(k) + \mu \, u(k-i)e(k). \qquad (1.6)$$

Appropriately defining $\hat{\theta}$ as the vector of $\hat{b}_i$s converts the $i = 1, 2, ..., n$ versions of (1.6) to

$$\hat{\theta}(k+1) = \hat{\theta}(k) + \mu \, X(k)e(k), \qquad (1.7)$$

where X is as in (1.4) but with the $\hat{y}$s removed.

Derivation of the algorithm in (1.7) can proceed from an approximate gradient descent strategy

$$\hat{\theta}(k+1) = \hat{\theta}(k) - \mu \frac{\partial\{\frac{1}{2} e^2(k)\}}{\partial\hat{\theta}(k)} \qquad (1.8)$$

where $\partial\{\frac{1}{2} e^2(k)\}/\partial\hat{\theta}(k) = -e(k)\{\partial\hat{y}(k)/\partial\hat{\theta}(k)\} = -e(k)X(k)$ as long as $\hat{y}$ and X are not functionally dependent on $\hat{\theta}$. Analysis of the convergence and robustness properties of LMS can be based on stability analysis [4] of (1.7) rewritten as the error system

$$\tilde{\theta}(k+1) = [I - \mu X(k)X^T(k)]\tilde{\theta}(k). \qquad (1.9)$$

With μ sufficiently small relative to the maximum value of a bounded $X(k)X^T(k)$, (1.9) is Lyapunov stable.

Fortunately, when $A \neq 0$, and $\hat{A} \neq 0$, the gradient approach algorithm derivation and error system stability analysis can be extended. Added complexities do arise. Repeating (1.8), X is no longer functionally independent of $\hat{\theta}$ because X in (1.4) includes $\hat{y}$ which is composed using $\hat{\theta}$. This nonlinearity complicates the evaluation of the gradient of e^2 with respect to $\hat{\theta}$, unlike its rather straightforward description in the linear-in-the-parameters case of (1.6)-(1.8). Under the pragmatic assumption/requirement that the stepsize μ is small enough, a recursive approximation to the gradient exists, such that (1.8) becomes [2]

$$\hat{\theta}(k+1) = \hat{\theta}(k) + \mu[1 - \hat{A}(q^{-1}, k)]^{-1}[X(k)]e(k); \qquad (1.10)$$

Compare (1.10) to (1.7). Recall that for (1.7) $e = X^T\tilde{\theta}$, while for (1.10) $e = [1-A]^{-1}X^T\tilde{\theta}$. The error system underlying (1.10) can be closely approximated as

$$\tilde{\theta}(k+1) = (I - \mu\{[1-\hat{A}(q^{-1}, k)]^{-1}[X(k)]\}\{[1-A(q^{-1})]^{-1}[X^T(k)]\})\tilde{\theta}(k) \qquad (1.11)$$

by again utilizing the fact that μ is sufficiently small. Note that, if $\hat{A} = A$, the form of (1.11) matches that of (1.9) with X in (1.9) replaced by $[1-A]^{-1}[X]$. This suggests that (1.11) should, at least locally about $\hat{A} = A$, have stability properties similar to those of (1.9).

But, if $[1 - \hat{A}]^{-1}$ were, even temporarily, unstable, $[1 - \hat{A}]^{-1}[X]$ in (1.10) could become undesirably large which would wreak havoc. In fact, for algorithms (such as (1.10)), which utilize time-varying regressor filtering (or time-varying prediction error filtering) to compensate for the filtering of $X^T\tilde{\theta}$ in e, as in (1.3), typical convergence proofs rely on maintenance at each iteration of the stability of the added time-varying filter [2] [5]. The necessity, in certain cases, of such stability maintenance, sometimes referred to as a projection procedure, can be readily verified via simulation.

[1] Supported by NSF Grant MIP-8608787.

For (1.10), checking the stability at each iteration of $[1-\hat{A}(q^{-1},k)]^{-1}$ is a computationally costly task. However, for alternative pole-zero models that realize the same transfer function, such a stability check could be quite simple. For example, for the IIR tapped lattice model

$$y(k)=\sum_{i=0}^{N} v_i B_i(k) \tag{1.12}$$

$$B_i(k)=B_{i-1}(k-1)+w_i F_{i-1}(k),\quad i=1,2,3,\ldots,N \tag{1.13}$$

$$F_i(k)=F_{i+1}(k)-w_{i+1}B_i(k-1),\quad i=0,1,2,\ldots,N-1 \tag{1.14}$$

$$F_N(k)=u(k) \tag{1.15}$$

$$B_0(k)=F_0(k), \tag{1.16}$$

the third-order case of which is depicted in Figure 1, stability is assured if and only if the w_i all have magnitudes less than one. Another potential benefit of the lattice form is its reduced sensitivity of pole and zero locations to parameter variation in comparison to the sensitivity of the direct form. Note that the parameters v_i and w_i of (1.12) - (1.16) can be related uniquely to those of (1.1) for $n=N$. See, e.g., [6]. This one-to-one relationship between direct form and tapped lattice model parameters suggests that excitation in $\{u\}$, which is sufficient to solve exactly, given the associated $\{y\}$, for the unknown a_i and b_i in (1.1), will prove sufficient to solve uniquely for the unknown v_i and w_i in (1.12)-(1.16).

So why not develop a gradient descent-based algorithm for (1.12) - (1.16) as an alternative to (1.10) for (1.1)? In fact, it's been done [7]. Indeed, simulations indicate that stability maintenance is needed and that it is easily done. Missing, however, is an examination of the error system of this adaptive parameter estimator for a tapped lattice that dovetails nicely with those of (1.9) and (1.11). This paper provides this interconnection. Two consequences accrue:

(i) The adaptive IIR tapped lattice of [7] is reformulated to permit its first formal convergence analysis.

(ii) The error system of the adaptive IIR tapped lattice is a curious extension of (1.9) and (1.11) that offers a new class of challenges to adaptive parameter estimation theorists.

2. *Pole-Zero Model Identification via Adaptive IIR Tapped Lattice Filter Algorithm of Parikh, Ahmed and Stearns*

An adaptive version of the IIR tapped lattice filter in (1.12) - (1.16) would replace the unknown parameters by estimated quantities. But the signals $F_i(k)$ and $B_i(k)$ within the lattice structure of Figure 1 are also "unknown" to the adaptive filter and must be estimated as well. Thus, (1.12) - (1.16) are replaced by

$$\hat{y}(k)=\sum_{i=0}^{N}\hat{v}_i(k)\hat{B}_i(k) \tag{2.1}$$

$$\hat{B}_i(k)=\hat{B}_{i-1}(k-1)+\hat{w}_i(k)\hat{F}_{i-1}(k),\quad i=1,2,\ldots,N \tag{2.2}$$

$$\hat{F}_i(k)=\hat{F}_{i+1}(k)-\hat{w}_{i+1}(k)\hat{B}_i(k-1),\quad i=0,1,2,\ldots,N-1 \tag{2.3}$$

$$\hat{F}_N(k)=u(k) \tag{2.4}$$

$$\hat{B}_0(k)=\hat{F}_0(k). \tag{2.5}$$

The tapped lattice structure of (2.1) - (2.5), diagrammed in Figure 2 clearly replicates Figure 1 with "^"s and time indices added in the right places.

The update algorithm from [7] for the $\hat{v}_i$ and $\hat{w}_i$ in (2.1) - (2.3) is based on an approximate gradient descent procedure.

$$\hat{v}_i(k+1)=\hat{v}_i(k)-\mu_i\frac{\partial\frac{1}{2}\{[y(k)-\hat{y}(k)]^2\}}{\partial\hat{v}_i(k)} \tag{2.6}$$

$$\hat{w}_i(k+1)=\hat{w}_i(k)-\rho_i\frac{\partial\frac{1}{2}\{[y(k)-\hat{y}(k)]^2\}}{\partial\hat{w}_i(k)} \tag{2.7}$$

Since $y(k)$ is not a function of $\hat{v}_i(k)$ or $\hat{w}_i(k)$, (2.6) and (2.7) become

$$\hat{v}_i(k+1)=\hat{v}_i(k)+\mu_i[y(k)-\hat{y}(k)]\frac{\partial\hat{y}(k)}{\partial\hat{v}_i(k)} \tag{2.8}$$

$$\hat{w}_i(k+1)=\hat{w}_i(k)+\rho_i[y(k)-\hat{y}(k)]\frac{\partial\hat{y}(k)}{\partial\hat{w}_i(k)} \tag{2.9}$$

From (2.1)

$$\frac{\partial\hat{y}(k)}{\partial\hat{v}_i(k)}=\hat{B}_i(k)+\sum_{j=0}^{N}\hat{v}_j(k)\frac{\partial\hat{B}_j(k)}{\partial\hat{v}_i(k)}=\hat{B}_i(k), \tag{2.10}$$

since the $\hat{B}_j$ do not depend explicitly on the $\hat{v}_i$. Also, from (2.1)

$$\frac{\partial\hat{y}(k)}{\partial\hat{w}_i(k)}=\sum_{j=0}^{N}\hat{v}_j(k)\frac{\partial\hat{B}_j(k)}{\partial\hat{w}_i(k)}. \tag{2.11}$$

From (2.2)

$$\frac{\partial\hat{B}_j(k)}{\partial\hat{w}_i(k)}=\frac{\partial\hat{B}_{j-1}(k-1)}{\partial\hat{w}_i(k)}+\frac{\partial\hat{w}_j(k)}{\partial\hat{w}_i(k)}\hat{F}_{j-1}(k)+\hat{w}_j(k)\frac{\partial\hat{F}_{j-1}(k)}{\partial\hat{w}_i(k)} \tag{2.12}$$

Under the assumption that μ_i in (2.6) is sufficiently small so that

$$\frac{\partial\hat{B}_{j-1}(k-1)}{\partial\hat{w}_i(k)}\approx\frac{\partial\hat{B}_{j-1}(k-1)}{\partial\hat{w}_i(k-1)} \tag{2.13}$$

we can approximate (2.12) by

$$\frac{\partial\hat{B}_j(k)}{\partial\hat{w}_i(k)}\approx\begin{cases}\dfrac{\partial\hat{B}_{j-1}(k-1)}{\partial\hat{w}_i(k-1)}+\hat{w}_j(k)\dfrac{\partial\hat{F}_{j-1}(k)}{\partial\hat{w}_i(k)}, & i\neq j\\[2ex] \dfrac{\partial\hat{B}_{j-1}(k-1)}{\partial\hat{w}_i(k-1)}+\hat{F}_{j-1}(k)+\hat{w}_j(k)\dfrac{\partial\hat{F}_{j-1}(k)}{\partial\hat{w}_i(k)}, & i=j\end{cases} \tag{2.14}$$

for $j=1,2,\ldots,N$. Note the (time) recursive form of (2.14). From (2.3)

$$\frac{\partial\hat{F}_j(k)}{\partial\hat{w}_i(k)}=\frac{\partial\hat{F}_{j+1}(k)}{\partial\hat{w}_i(k)}-\frac{\partial\hat{w}_{j+1}(k)}{\partial\hat{w}_i(k)}\hat{B}_j(k-1)-\hat{w}_{j+1}(k)\frac{\partial\hat{B}_j(k-1)}{\partial\hat{w}_i(k)}$$

$$\approx\begin{cases}\dfrac{\partial\hat{F}_{j+1}(k)}{\partial\hat{w}_i(k)}-\hat{w}_{j+1}(k)\dfrac{\partial\hat{B}_j(k-1)}{\partial\hat{w}_i(k-1)}, & i\neq j+1\\[2ex] \dfrac{\partial\hat{F}_{j+1}(k)}{\partial\hat{w}_{i}(k)}-\hat{B}_j(k-1)-\hat{w}_{j+1}(k)\dfrac{\partial\hat{B}_j(k-1)}{\partial\hat{w}_i(k-1)}, & i=j+1\end{cases} \tag{2.15}$$

for $j=0,1,\ldots,N-1$. From (2.4),

$$\frac{\partial\hat{F}_N(k)}{\partial\hat{w}_i(k)}=0. \tag{2.16}$$

From (2.5),

$$\frac{\partial\hat{B}_0(k)}{\partial\hat{w}_i(k)}=\frac{\partial\hat{F}_0(k)}{\partial\hat{w}_i(k)}. \tag{2.17}$$

Define

$$c_{i,j}(k)=\begin{cases}c_{i,j-1}(k-1)+\hat{w}_j(k)d_{i,j-1}(k), & i\neq j,\ j=1,\ldots,N\\ c_{i,j-1}(k-1)+\hat{F}_{j-1}(k)+\hat{w}_j(k)d_{i,j-1}(k), & i=j,\ j=1,\ldots,N\\ d_{i,0}(k), & j=0.\end{cases} \tag{2.18}$$

for $i=1,\ldots,N$ and

$$d_{i,j}(k)=\begin{cases}d_{i,j+1}(k)-\hat{w}_{j+1}(k)c_{i,j}(k-1), & i\neq j+1,\ j=0,\ldots,N-1\\ d_{i,j+1}(k)-\hat{B}_j(k-1)-\hat{w}_{j+1}(k)c_{i,j}(k-1), & i=j+1,\ j=0,\ldots,N-1\\ 0 & j=N\end{cases} \tag{2.19}$$

for $i=1,\ldots,N$. Compare (2.18) and (2.19) to (2.14) - (2.17). Note that $c_{i,j}(k)$ is an approximation of $\partial\hat{B}_j(k)/\partial\hat{w}_i(k)$ and $d_{i,j}(k)$ is an approximation of $\partial\hat{F}_j(k)/\partial\hat{w}_i(k)$. The adaptive algorithm of [7] combines (2.18) and (2.19) with (2.8) - (2.11) rewritten as

$$\hat{v}_i(k+1)=\hat{v}_i(k)+\mu_i[y(k)-\hat{y}(k)]\hat{B}_i(k),\quad i=0,1,\ldots,N \tag{2.20}$$

$$\hat{w}_i(k+1)=\hat{w}_i(k)+\rho_i[y(k)-\hat{y}(k)][\sum_{j=0}^{N}\hat{v}_j(k)c_{i,j}(k)],\quad i=1,2,\ldots,N. \tag{2.21}$$

Figure 3 illustrates the generation of

$$S_i(k)=\sum_{j=0}^{N}\hat{v}_j(k)c_{i,j}(k) \tag{2.22}$$

for $i=1$, 2, and 3 where $N=3$. Note that the sensitivity function approximation of (2.11) is a particular filtering of the states $\hat{B}_i$ and $\hat{F}_i$ of Figure 2 through a replica of the adapted lattice in Figure 2. This structural "coincidence" is examined in a more general setting in [8].

3. *The Structure of the Prediction Error*

As discussed in the introduction, in certain cases, the structure of the prediction error can be used to immediately state the squared prediction error gradient descent algorithm. Recall (1.3) and (1.10) and [2]. More generally, the basic adaptive parameter estimation algorithm update term kernel is a product of the (possibly filtered) regressor and the (possibly filtered) prediction error. When the prediction error is a filtered version of the inner product of the regressor and parameter error vector, an error system, such as (1.11), can be formed for stability/convergence analysis [4] [5] [9]. In this section we will follow adaptive parameter estimator error system concepts as an alternative method of algorithm derivation to that of the gradient-descent approach of the preceding section and consider the local stability of the resulting "new" error system.

The first step is to designate the regressor. Mimicking the definition used in (1.4) for (1.2), we consider those signals that the adapted parameters multiply in Figure 2, i.e. the $\hat{F}_i$ and $\hat{B}_i$. Note that $\hat{v}_i(k)$ is multiplied by $\hat{B}_i(k)$ while $\hat{w}_i(k)$ is multiplied by both $-\hat{B}_{i-1}(k-1)$ and $\hat{F}_{i-1}(k)$. Thus, given (1.3) and a small stepsize assumption, we are hoping for something like

$$e(k) = \left\{ M(q^{-1}) \begin{bmatrix} \hat{B}_0(k) \\ \hat{B}_1(k) \\ \cdot \\ \cdot \\ \cdot \\ \hat{B}_N(k) \\ \hat{F}_0(k) - \hat{B}_0(k-1) \\ \cdot \\ \cdot \\ \cdot \\ \hat{F}_{N-1}(k) - \hat{B}_{N-1}(k-1) \end{bmatrix} \right\}^T \begin{bmatrix} v_0 - \hat{v}_0(k) \\ v_1 - \hat{v}_1(k) \\ \cdot \\ \cdot \\ \cdot \\ v_N - \hat{v}_N(k) \\ w_1 - \hat{w}_1(k) \\ \cdot \\ \cdot \\ \cdot \\ w_N - \hat{w}_N(k) \end{bmatrix} \tag{3.1}$$

where $e = y - \hat{y}$, y is from (1.12), $\hat{y}$ from (2.1), and M is a scalar transfer function. We almost get it. The needed "extension" is that M turns out to be a transfer function matrix rather than a scalar.

Actually,

$$e(k) = y(k) - \hat{y}(k) = \sum_{i=0}^{N} v_i B_i(k) - \sum_{i=0}^{N} \hat{v}_i(k)\, \hat{B}_i(k)$$

$$= \sum_{i=0}^{N} [v_i - \hat{v}_i(k)]\, \hat{B}_i(k) + \sum_{i=0}^{N} v_i [B_i(k) - \hat{B}_i(k)]. \tag{3.2}$$

Note that the left sum in the last line of (3.2) matches the first half of (3.1) with $M = 1$. But what about the second sum?

From subtraction of (2.2) from (1.13)

$$\tilde{B}_i(k) = B_i(k) - \hat{B}_i(k) = \tilde{B}_{i-1}(k-1) + w_i \tilde{F}_{i-1}(k) + \tilde{w}_i(k)\hat{F}_{i-1}(k) \tag{3.3}$$

for $i = 1, 2, 3, \ldots, N$, where, from subtraction of (2.3) from (1.14),

$$\tilde{F}_i(k) = F_i - \hat{F}_i(k) = \tilde{F}_{i+1}(k) - w_{i+1}\tilde{B}_i(k-1) - \tilde{w}_{i+1}(k)\hat{B}_i(k-1) \tag{3.4}$$

for $i = 0, 1, 2, \ldots, N-1$. And from subtraction of (2.4) from (1.15)

$$\tilde{F}_N(k) = 0 \tag{3.5}$$

and (2.5) from (1.16)

$$\tilde{B}_0(k) = \tilde{F}_0(k). \tag{3.6}$$

Figure 4 illustrates the third-order case of (3.3) - (3.6) used to form the right sum in the last line of (3.2).

Compare the form of (2.18) and (2.19) with that of (3.3) - (3.6). With the following replacements in the summed versions of (2.18) and (2.19)

$$\sum_{i=1}^{N} c_{i,j}(k) \to \tilde{B}_j(k) \tag{3.7}$$

$$\sum_{i=1}^{N} d_{i,j}(k) \to \tilde{F}_j(k) \tag{3.8}$$

$$\hat{w}_j(k) \to w_j \tag{3.9}$$

$$\hat{F}_{j-1}(k) \to \tilde{w}_j(k)\hat{F}_{j-1}(k), \quad j = 1, \ldots, N \tag{3.10}$$

$$\hat{B}_j(k-1) \to \tilde{w}_{j+1}(k)\,\hat{B}_j(k-1), \quad j = 0, \ldots, N-1 \tag{3.11}$$

(3.3) - (3.6) emerge (but with subscripts j rather than i). This replacement connects Figure 4 to the sum of the three Figures 3(a) - (c). This representation supports the interpretation of $\sum v_i \tilde{B}_i$ in (3.2) as the output of the unknown lattice in (1.12) - (1.16) shown in Figure 1 when forced, not by $u(k)$, but by $-\tilde{w}_i(k)\, \hat{B}_{i-1}(k-1)$ and $\tilde{w}_i(k)\, \hat{F}_{i-1}(k)$, which all enter the lattice at different summers. Due to these different entry points into a time-invariant lattice the contribution to $\sum v_i \tilde{B}_i$ by each $\tilde{w}_i(k)\, \hat{B}_{i-1}(k-1)$ and $\tilde{w}_i(k)\, \hat{F}_{i-1}(k)$ passes through a different transfer function.

Thus, (3.2) can be rewritten as

$$e(k) = \sum_{i=0}^{N} \tilde{v}_i(k)\hat{B}_i(k) + \sum_{j=0}^{N} v_j \Big\{ \sum_{i=1}^{N} G_{ij}(q^{-1})[\tilde{w}_i(k)\hat{F}_{i-1}(k)]$$

$$+ H_{ij}(q^{-1})[\tilde{w}_i(k)\hat{B}_{i-1}(k-1)]\Big\}. \tag{3.12}$$

Under the small step-size assumption, such that $\{\tilde{w}_i(k)\}$ has very low frequency bandwidth compared to that of G_{ij}, H_{ij}, $\{\hat{B}_i(k)\}$ and $\{\hat{F}_i(k)\}$, (3.12) can be approximated by

$$e(k) = \sum_{i=0}^{N} \tilde{v}_i(k)\, \hat{B}_i(k) + \sum_{i=1}^{N} \tilde{w}_i(k) \Big[\sum_{j=0}^{N} v_j \{G_{ij}(q^{-1})[\hat{F}_{i-1}(k)]$$

$$+ H_{ij}(q^{-1})[\hat{B}_{i-1}(k-1)]\}\Big]. \tag{3.13}$$

Replacing sums by matrix multiplication, rewrite (3.13) as

$$e(k) = \begin{bmatrix} \hat{B}_0(k) \\ \cdot \\ \cdot \\ \cdot \\ \hat{B}_N(k) \\ \sum_{j=0}^{N} v_j \{H_{1j}(q^{-1})[\hat{B}_0(k-1)] + G_{1j}(q^{-1})[\hat{F}_0(k)]\} \\ \cdot \\ \cdot \\ \cdot \\ \sum_{j=0}^{N} v_j \{H_{Nj}(q^{-1})[\hat{B}_{N-1}(k-1)] + G_{Nj}(q^{-1})[\hat{F}_{N-1}(k)]\} \end{bmatrix}^T \begin{bmatrix} \tilde{v}_0(k) \\ \cdot \\ \cdot \\ \cdot \\ \tilde{v}_N(k) \\ \tilde{w}_1(k) \\ \cdot \\ \cdot \\ \cdot \\ \tilde{w}_N(k) \end{bmatrix} \tag{3.14}$$

Comparison with (3.1) indicates that the "generic" form of (3.1) should be expanded to consider M a *matrix* transfer function with vector input and vector output. Indeed, the "new" form of M that equates (3.14) and (3.1) is

$$M(q^{-1}) = \begin{bmatrix} M_{11} & 0 \\ M_{21} & M_{22} \end{bmatrix} \tag{3.15}$$

where

$$M_{11} = I_{N+1} \tag{3.16}$$

$$M_{21} = diag\{q^{-1}\sum_{j=0}^{N} v_j \{H_{1j}(q^{-1},v,w) + G_{1j}(q^{-1},v,w)\}, \ldots,$$

$$q^{-1}\sum_{j=0}^{N} v_j \{H_{Nj}(q^{-1},v,w) + G_{Nj}(q^{-1},v,w)\}\} \tag{3.17}$$

and

$$M_{22} = diag[\sum_{j=0}^{N} v_j G_{1j}(q^{-1},v,w), \ldots, \sum_{j=0}^{N} v_j G_{Nj}(q^{-1},v,w)] \tag{3.18}$$

Based on the algorithm specification of (1.10) for (1.3), a logical parameter adaptation algorithm for (2.1) - (2.5) given (3.14) is

$$\hat{v}_i(k+1) = \hat{v}_i(k) + \mu_i e(k)\, \hat{B}_i(k), \quad i = 0, 1, \ldots, N \tag{3.19}$$

$$\hat{w}_i(k+1) = \hat{w}_i(k) + \rho_i e(k)\, \hat{S}_i(k), \quad i = 1, \ldots, N \tag{3.20}$$

where

$$\hat{S}_i(k) = \sum_{j=0}^{N} \hat{v}_j(k)\{\hat{H}_{ij}(q^{-1}, \hat{v}, \hat{w}, k)\, [\hat{B}_{i-1}(k-1)]$$

$$+ \hat{G}_{ij}(q^{-1}, \hat{v}, \hat{w}, k)\, [\hat{F}_{i-1}(k)]\} \tag{3.21}$$

where $\hat{H}_{ij}(q^{-1}, \hat{v}, \hat{w}, k)$ (and $\hat{G}_{ij}(\cdot,\cdot,\cdot,\cdot)$) are time-varying transfer function operators obtained by substituting $\hat{v}_i(k)$ and $\hat{w}_i(k)$ for v_i and w_i in the time-invariant operators $H_{ij}(q^{-1})$ and $G_{ij}(q^{-1})$ of (3.12)-(3.18). The choice of (3.19)-(3.21) for the adaptive algorithm mimics the use of time-varying regressor filtering in (1.10) relative to (1.7) for the form of the prediction error in (1.3) relative to simply $X^T\tilde{\theta}$. In (3.19)-(3.21), we are using a time-varying regressor filter that is intended to compensate for M in (3.15). To recognize that the desired $\hat{M}$ ($= M(q^{-1},\hat{v},\hat{w},k)$) regressor filtering is occuring in (3.19)-(3.21) recall the previously noted connection between Figures 3 and 4: They are "equivalent" with $\tilde{w}\hat{F}$ and $-\tilde{w}\hat{B}$ as inputs to Figure 4 and $\hat{F}$ and $-\hat{B}$ as inputs to Figure 3. The small stepsize approximation, in essence, permits approximation of the output of Figure 4 as the weighted sum of the outputs of Figures 3(a) - (c) multiplied by the corresponding $\tilde{w}_i$. Thus, the $\hat{S}_i$ of (2.22) equal the $\hat{S}_i$ in (3.21). This "equates" the algorithm (3.19) - (3.21), which is implied by extension of the relationship of (1.3) and (1.10), to the gradient descent version of (2.18) - (2.22) from [7].

The error system of (3.19) - (3.21) (or equivalently (2.18) - (2.22)), corresponding to (1.11) can be written as

$$\tilde{\theta}(k+1) = \{I - \mu\, [(M(q^{-1},\hat{\theta}(k))X(k)]\, [M(q^{-1},\theta)X(k)]]^T\}\tilde{\theta}(k) \tag{3.22}$$

where $\hat{\theta}$ is a $2N+1$ vector of $\hat{v}_i$s and $\hat{w}_i$s, $X(k)$ is a $2N+1$ vector of of $\hat{B}_i(k)$ for $i = 0$ to N and $\hat{F}_i(k) - \hat{B}_i(k-1)$ for $i = 0$ to $N-1$, and $M(q^{-1},\hat{\theta}(k))$ is from (3.16)-(3.18) with corresponding entries from $\hat{\theta}$ are used instead of those from θ as in $M(q^{-1},\theta)$. With S defined by $M[X]$, $\hat{S}$ recognized as $\hat{M}[X]$, and $M = \hat{M}$ such that $S = \hat{S}$, (3.22) matches the form of (1.9) with X replaced by S.

As with the connection of (1.9) and (1.11), the exponential stability of (1.9) for persistently exciting X establishes the local stability of (3.22), and the adaptive IIR tapped lattice algorithm of (1.12) - (1.16) from [7], about $\hat{\theta} = \theta$ and $\hat{S} = S$, when S is similarly persistently exciting. This persistent excitation condition on $\hat{S}$ is established if S is bounded and possesses a sustained adequate spectral richness. To help maintain the boundedness of $\hat{S}$, keep $|\hat{w}_i(k)| < 1$ for all k. It seems likely that spectral richness of $\{u\}$ should translate to the desired spectral richness of $\{S\}$ due to the uniqueness of the direct to tapped lattice model conversion when the estimated model order exactly matches that of the minimal description of the actual plant, i.e. the plant in (1.12)-(1.16) has no pole-zero cancellations in its transfer function. The use of the normalization suggested in (8) and (9) in [7] does not alter the existence of locally stable behavior, but may modify the impact of certain quantitative requirements for proof of its existence.

4. *Simulation Evidence*

This section provides readily reproducible simulated evidence of the local stability and robustness of this adaptive IIR tapped lattice identifier. Consider the first order lattice of (1.12), with $N = 1$, $v_0 = 0.3$, $v_1 = 0.2$, $w_1 = -0.7$, driven by $u(k) = 0.5 + 0.5\sin(0.23k) + 0.5\sin(0.57k) + 0.5\sin(0.73k)$, for $k > 0$, from zero initial conditions at $k = 0$. Starting the identifier of (2.1)-(2.5) and (2.20)-(2.21) with $\hat{v}_0(0) = \hat{v}_1(0) = \hat{w}_1(0) = \hat{B}_0(0) = \hat{B}_1(0) = \hat{F}_0(0) = \hat{F}_1(0) = 0$ and $\mu_0 = \mu_1 = \rho_1 = 0.02$, yields the decaying prediction error $y - \hat{y}$ in Figure 5(a) and the linearly decreasing logarithm of the summed squared parameter error $\log_{10}[\tilde{v}_0^2(k)+\tilde{v}_1^2(k)+\tilde{w}_1^2(k)]$ in Figure 5(b). Note that, as expected, the prediction error and summed squared parameter error are converging to zero, with the summed squared parameter error doing so exponentially fast.

The expected robustness to system output measurement noise can be tested by adding a low-variance, white, zero-mean, gaussian noise to the prediction error $y-\hat{y}$ before it is used in the algorithm in (2.20) and (2.21). Figures 6(a) plots the prediction error before this noise is added, which is the disturbance free prediction error we are unable to measure. Figure 6(b) plots the summed squared parameter error which shows an initial exponential decay that quickly reaches a floor that represents the unremovable parameter variation due to the unpredictable noise in the prediction error and a nonvanishing adaptive algorithm stepsize. Our understanding of (1.9) suggests that decreasing the stepsize in (2.20)-(2.21), say from .02 to .005, should result in a slower convergence rate during the exponential decay phase and a lower floor to the unremovable variance in the summed squared parameter error. This is verified by comparing Figures 6 and 7.

Higher-order plant and adaptive tapped lattice identifier simulations have illustrated these same characteristics: (i) local exponential convergence rate of the summed squared lattice parameter error in the noise-free, ideal case and (ii) a tradeoff with stepsize variation between convergence rate and minimum summed squared parameter error. Tests of higher-order situations also appeared to indicate a dependence of the convergence rate on the input spectrum, which is an acknowledged property of (1.9) or (1.11).

5. *Conclusion*

This paper has reinterpreted an adaptive IIR filter algorithm developed by Parikh, Ahmed, and Stearns [7] as an algorithm for output error identification that enjoys a computationally trivial stability check and maintenance procedure. This algorithm, as indicated in [7], was developed from a gradient descent strategy, as described in section 2.

The main contribution of the present paper is its development, in section 3, of an error system formulation in a form amenable to analytical tools, which have been extracted from nonlinear system stability theory to prove the local exponential stability and robustness of a variety of adaptive identifiers [4], [5], [8], [9]. Thus, the adaptive IIR tapped lattice output error identifier from [7] is provable to be locally robust under certain conditions, such as persistent excitation similar to that associated with other output error identifiers [2], [4], [9]. Simulations in section 4 verify the existence of such behavior.

Several issues remain to be addressed if this algorithm is to be promoted as a viable adaptive output error identifier. Theoretically and practically, description of the size of attraction of the local robustness region in the space of initial parameter estimates and other initial conditions about a satisfactory answer is a significant concern. Detailing the translation of the satisfaction of the persistent excitation condition on the regressor to a condition on the external input and establishment of an accurate approximation for the local convergence rate are also nontrivial tasks. All of these, and other, questions are rather standard accoutrements of the stability theory approach to adaptive system analysis that provide insight into the applicability and performance of adaptive algorithms in various problems. Such studies of applicability and performance of this provably stable adaptive algorithm are now in order. This adaptive tapped-lattice IIR filter also serves as an example of extension of adaptive parameter estimator error system analysis to situations where the prediction error can be approximated as the inner product of the parameter error vector and a regressor that has been passed through a matrix transfer function. Other examples of such composite error systems exist in quite different adaptive system applications.[9]

Acknowledgement: The authors thank Geoff Williamson of Cornell University for his assistance.

6. *References*

[1] I. D. Landau, "Unbiased recursive identification using model reference adaptive techniques," *IEEE Trans. on Auto. Control*, vol. AC-21, pp. 194-202, April 1976.

[2] C. R. Johnson, Jr., "Adaptive IIR filtering: Current results and open issues," *IEEE Trans. on Info. Thy.*, vol. IT-30, pp. 237-250, March 1984.

[3] B. Widrow, J. M. McCool, M. G. Larimore, and C. R. Johnson, Jr., "Stationary and nonstationary learning characteristics of the LMS adaptive filter," *Proc. IEEE*, vol. 64, pp. 1151-1162, August 1976.

[4] B. D. O. Anderson, R. R. Bitmead, C. R. Johnson, Jr., P. V. Kokotovic, R. L. Kosut, I. M. Y. Mareels, L. Praly, and B. D. Riedle, *Stability of Adaptive Systems: Passivity and Averaging Analysis*, MIT Press, 1986.

[5] W. A. Sethares, B. D. O. Anderson and C. R. Johnson, Jr., "Adaptive algorithms with filtered regressor and filtered error," *Mathematics of Control, Signals and Systems*, to appear.

[6] B. Widrow and S. Stearns, *Adaptive Signal Processing*, Prentice-Hall, pp. 164-182, 1985.

[7] D. Parikh, N. Ahmed, and S. Stearns, "An adaptive lattice algorithm for recursive filters," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. ASSP-28, pp. 110-111, February 1980.

[8] B. D. O. Anderson and C. R. Johnson, Jr., "Identification of linear systems containing unknown linear gain elements," *Proc. 8th IFAC Symp. on Ident. and Sys. Par. Est.*, Beijing, China, August 1988.

[9] C. R. Johnson, Jr., *Lectures on Adaptive Parameter Estimation*, Prentice-Hall, 1988.

[10] G. A. Williamson, C. R. Johnson, Jr., and S. Dasgupta, "On a composite error system arising from adaptive error filtering in noise cancelers," *Proc. 22nd Conf. on Info. Sciences and Sys.*, Princeton, NJ, March 1988.

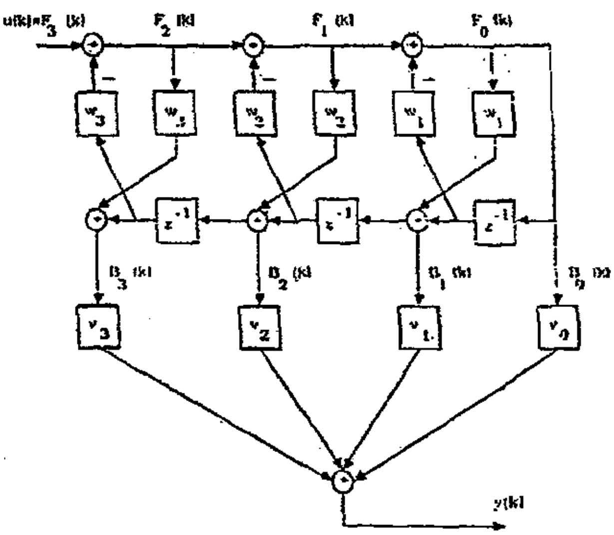

Figure 1: Third Order Tapped IIR Lattice

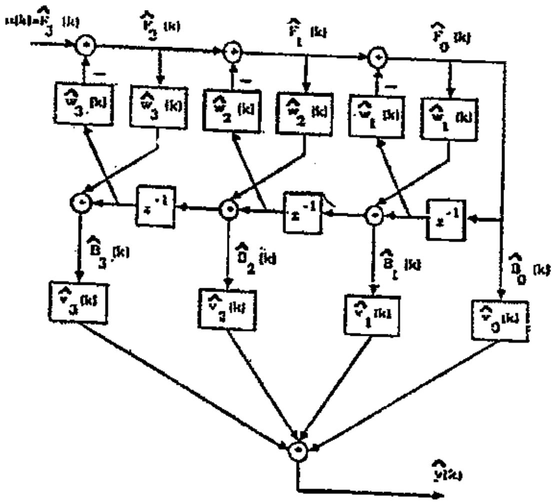

Figure 2- Adaptive Tapped IIR Lattice

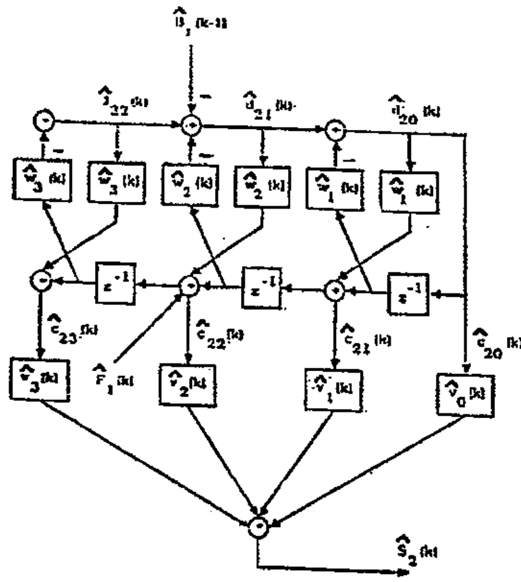

Figure 3b:- Approximate Sensitivity Function Generation

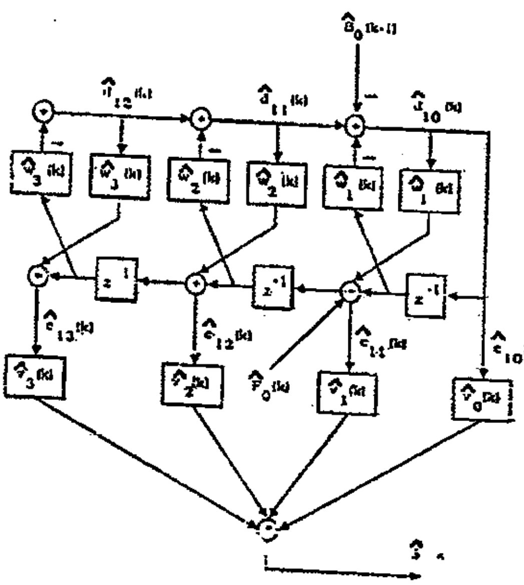

Figure 3a Approximate Sensitivity Function Generation

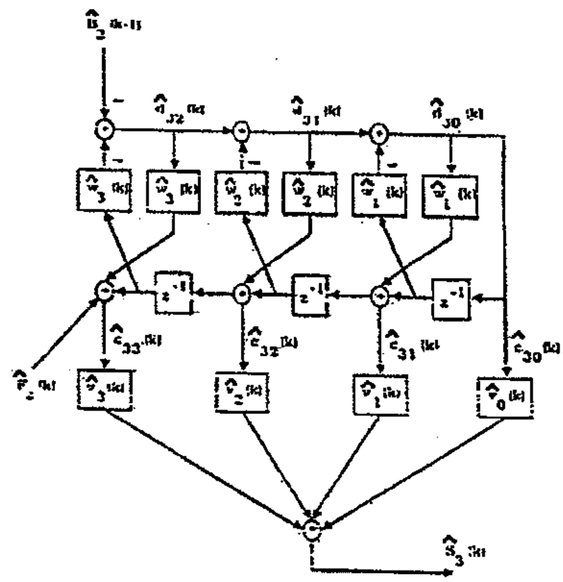

Figure 3c: Approximate Sensitivity Function Generation

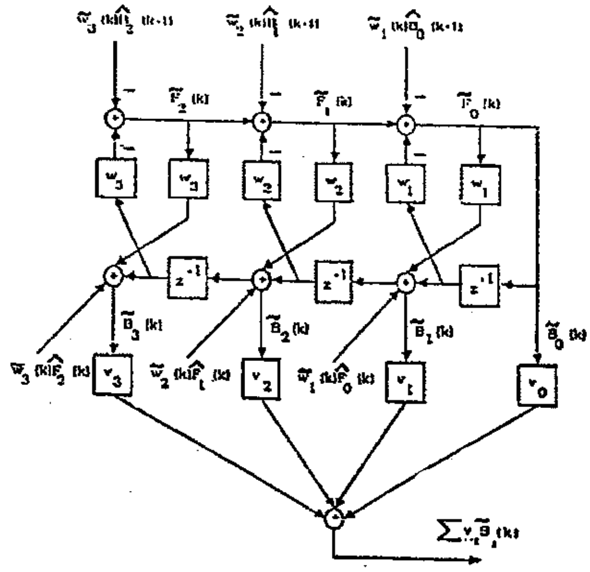

Figure 4: Prediction Error Component

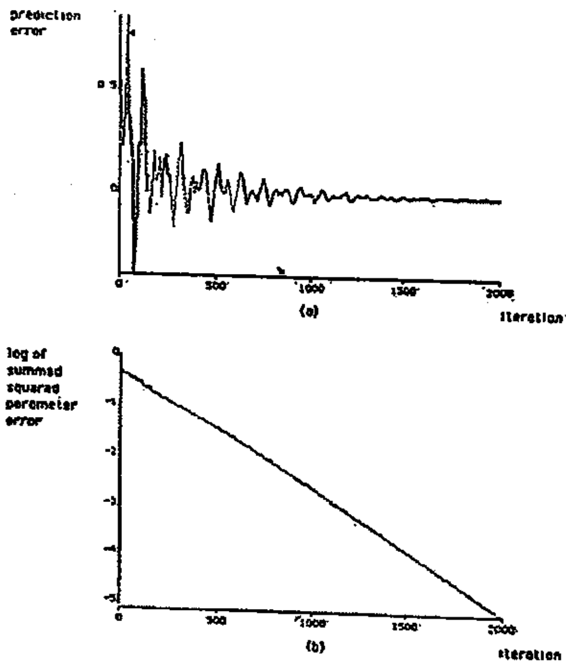

Figure 5 Local Asymptotically Convergent Behavior, Ideal Case: (a) Prediction Error (b) Summed Squared Parameter Error

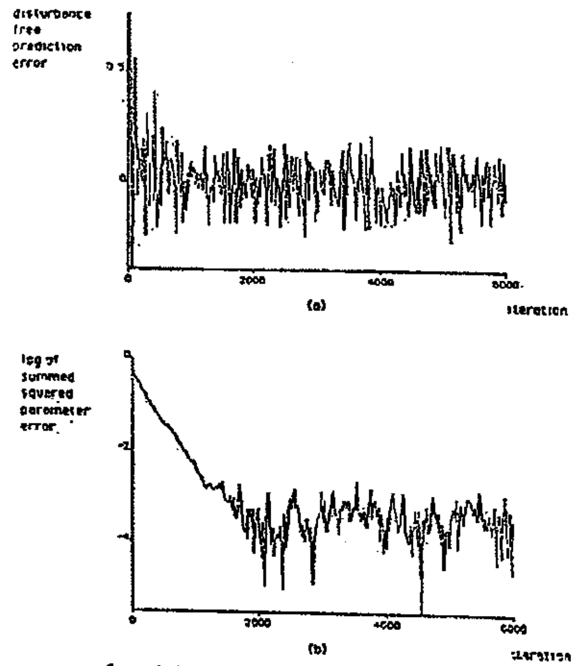

Figure 6 Robust Behavior Despite Output Measurement Noise (a) Prediction Error (b) Summed-Squared Parameter Error

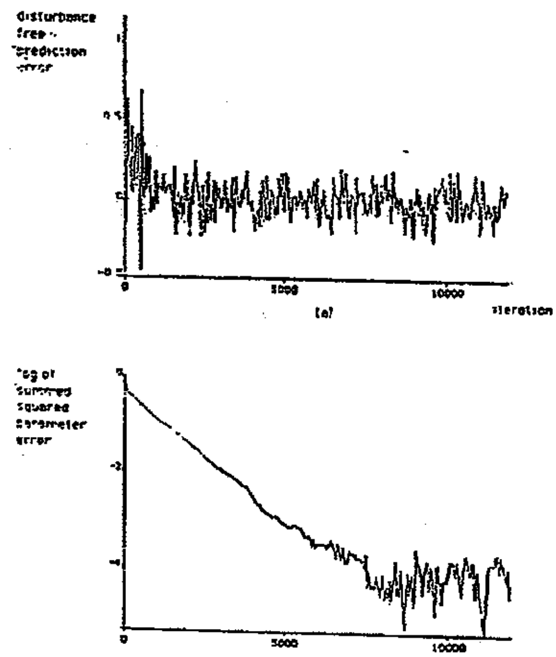

Figure 7 Robust Behavior Despite Output Measurement Noise (in the same situation as Figure 6 except the stepsize is smaller) (a) Prediction Error (b) Parameter Error