presented at:
The 29th Midwest Symposium on Circuits and Systems, Lincoln, Nebraska, August 1986

ON THE NUMERICAL AND STATISTICAL PROPERTIES OF INSTRUMENTAL VARIABLE LATTICE ALGORITHMS

S. D. Roach, R. L. Moses, and S. Yurkovich

Department of Electrical Engineering
The Ohio State University
2015 Neil Ave
Columbus, OH 43210

Abstract

The numerical properties of three instrumental variable ARMA identification algorithms are compared. Two recursive lattice algorithms (one normalized) and a block processing algorithm are found to be essentially equivalent in numerical error and in response to ill-conditioned problems. Simulations demonstrate the effectiveness of the normalized lattice algorithm at maintaining near unity magnitudes in the lattice parameters. While in some cases the parameter magnitudes suddenly diverge, they recover in a few time samples; this behavior is associated with division by a parameter which approaches zero and is easily avoided.

I. Introduction

In many engineering applications it is necessary to obtain a model from a given time series. Examples include on-line identification of structural dynamics, speech processing, adaptive equalization, and geophysical signal processing. For these applications an often-used time series model is the autoregressive moving average (ARMA) model given by

$$x(k) = -\sum_{i=1}^{p} a_i x(k-i) + \sum_{j=0}^{q} b_j E(k-j) \quad , \tag{1}$$

where $E(k)$ is zero mean, unit variance white noise.

There are numerous algorithms for estimating the a_i and b_i coefficients in an ARMA model. Recently, there has been growing interest in so-called fast lattice algorithms for on-line estimation. Lattice algorithms are popular because they combine the advantages of low computational burden and relatively high estimation accuracy [1].

In this paper we are interested in an instrumental variable-based method for estimating the AR coefficients in the ARMA model [2]. There are three different implementations of this method: a block processing method based on the extended Yule-Walker equations, an unnormalized lattice implementation, and a normalized lattice implementation. An important question surrounding these three algorithms is whether one has significantly better numerical properties than the others. This paper addresses that question by considering two measures of numerical quality of the algorithms: the effects of finite precision arithmetic, and the ability of the normalized algorithm to keep its internal parameters "small".

II. The Three Estimation Algorithms

The following sections briefly outline the three algorithms used in this study. Further details appear in [3,4].

A. *The Prewindow Block Processing (PBP) Algorithm*

Assume the given data points $x(k)$, $k = 1,\ldots,N$, are samples of the ARMA(p,q) process defined in (1). The so-called prewindow block processing algorithm solves the following linear system of p extended Yule-Walker equations [4]:

$$R_N \underline{a}_N = -r_N \quad , \tag{2}$$

where

$$\underline{a}_N = [a(1),\ldots,a(p)]^T \quad , \tag{3}$$

$$r_N = [r_{10}(N),\ldots,r_{p0}(N)]^T \quad , \tag{4}$$

$$R_N = [r_{ij}(N)] \quad , \tag{5}$$

$$r_{ij}(N) = \sum_{k=q+1}^{N} x(k-q-i)x(k-j)\lambda^{N-k} \quad , \tag{6}$$

where $r_{ij}(N)$, $i = 1,\ldots,p$, $j = 1,\ldots,p$, is an estimate of the autocorrelation function of $x(k)$ at time lag $q+i-j$. The exponential forgetting factor, λ, (incorporated in all three algorithms used in this study) allows the algorithm to track time varying parameters.

Since the autoregressive order p is usually small, the solution of (2) is easily found numerically. However, the formation of R_N^{-1} requires $O(p^3)$ computations and does not permit efficient real-time identification.

B. *The UPIV Algorithm*

The Unnormalized Prewindow Instrumental Variable

(UPIV) algorithm is a fast recursive implementation of the PBP algorithm [4]. The calculation involves updating the four prediction error scalars in (7)–(10) below and the five auxiliary scalars, or "lattice parameters", in (11)–(15) given below. The lattice filter multiplier coefficients are related to the five auxiliary scalars in a straightforward way, and time and order recursive formulas are available for obtaining the autoregressive coefficients [4].

$$f^z_{m+1,n} = f^z_{m,n} - [b^z_{m,n-1}][\sigma_{m,n}]/[\omega_{m,n-1}] \quad , \tag{7}$$

$$f^v_{m+1,n} = f^v_{m,n} - [b^v_{m,n-1}][\tau_{m,n}]/[\omega_{m,n-1}] \quad , \tag{8}$$

$$b^z_{m+1,n} = b^z_{m,n-1} - [f^z_{m,n}][\tau_{m,n}]/[\mu_{m,n}] \quad , \tag{9}$$

$$b^v_{m+1,n} = b^v_{m,n-1} - [f^v_{m,n}][\sigma_{m,n}]/[\mu_{m,n}] \quad , \tag{10}$$

$$\mu_{m+1,n} = \mu_{m,n} - [\sigma_{m,n}][\tau_{m,n}]/[\omega_{m,n-1}] \quad , \tag{11}$$

$$\omega_{m+1,n} = \omega_{m,n-1} - [\sigma_{m,n}][\tau_{m,n}]/[\mu_{m,n}] \quad , \tag{12}$$

$$\sigma_{m,n} = \lambda\sigma_{m,n-1} + [b^v_{m,n-1}][f^z_{m,n}]/[\gamma_{m,n}] \quad , \tag{13}$$

$$\tau_{m,n} = \lambda\tau_{m,n-1} + [f^v_{m,n}][b^z_{m,n-1}]/[\gamma_{m,n}] \quad , \tag{14}$$

$$\gamma_{m+1,n} = \gamma_{m,n} - [b^v_{m,n-1}][b^z_{m,n-1}]/[\omega_{m,n-1}] \quad . \tag{15}$$

C. *The NPIV Algorithm*

A shortcoming of the UPIV algorithm is that the four lattice parameters σ, τ, μ, and ω, tend to become large as the calculation progresses. For long data sets this growth prohibits implementation of the algorithm in fixed point arithmetic.

The Normalized Prewindow Instrumental Variable (NPIV) algorithm eliminates the problem of lattice parameter growth by normalizing lattice parameters with the square roots of certain factors in the following update equations [3]:

$$r^{(1)}_{n,p} = |1 - f^z_{n,p} f^v_{n,p} s^{(1)}_{n,p} s^{(3)}_{n,p}|^{\frac{1}{2}} \quad , \tag{16}$$

$$r^{(2)}_{n,p} = |1 - b^z_{n-1,p} b^v_{n-1,p} s^{(2)}_{n-1,p} s^{(3)}_{n,p}|^{\frac{1}{2}} \quad , \tag{17}$$

$$r^{(3)}_{n,p} = |1 - \sigma_{n,p}\tau_{n,p} s^{(1)}_{n,p} s^{(2)}_{n-1,p}|^{\frac{1}{2}} \quad , \tag{18}$$

$$\sigma_{n,p} = \sigma_{n-1,p} r^{(1)}_{n,p} r^{(2)}_{n,p} + f^z_{n,p} b^v_{n-1,p} s^{(3)}_{n,p} \quad , \tag{19}$$

$$\tau_{n,p} = \tau_{n-1,p} r^{(1)}_{n,p} r^{(2)}_{n,p} + b^z_{n-1,p} f^v_{n,p} s^{(3)}_{n,p} \quad , \tag{20}$$

$$f^z_{n,p+1} = [f^z_{n,p} - b^z_{n-1,p}\sigma_{n,p} s^{(2)}_{n-1,p}]/[r^{(2)}_{n,p} r^{(3)}_{n,p}] \quad , \tag{21}$$

$$f^v_{n,p+1} = [f^v_{n,p} - b^v_{n-1,p}\tau_{n,p} s^{(2)}_{n-1,p}]/[r^{(2)}_{n,p} r^{(3)}_{n,p}] \quad , \tag{22}$$

$$b^z_{n,p+1} = [b^z_{n-1,p} - f^z_{n,p}\tau_{n,p} s^{(1)}_{n,p}]/[r^{(1)}_{n,p} r^{(3)}_{n,p}] \quad , \tag{23}$$

$$b^v_{n,p+1} = [b^v_{n-1,p} - f^v_{n,p}\sigma_{n,p} s^{(1)}_{n,p}]/[r^{(1)}_{n,p} r^{(3)}_{n,p}] \quad . \tag{24}$$

As is shown in the sequel, the NPIV algorithm keeps the lattice parameters near unity in magnitude except when unfortunate data sets create ill-conditioned problems.

III. Comparative Studies

Simulations have been run to investigate the effects of finite precision and to observe the ability of the NPIV algorithm to keep its internal parameters "small". The simulations have been conducted for varying data bandwidths, forgetting factors, and model orders.

Numerical errors are evaluated through simulations coded in FORTRAN and run on a VAX 11/785 computer. The absolute error between single and double precision real arithmetic is then plotted as a function of time step. The single precision mantissa is 24-bits or 7.22 decimal digits, and the double precision mantissa is 56-bits or 16.9 decimal digits. The data used for numerical comparisons is single-pole, filtered white noise, generated according to

$$x(n) = \left[\frac{z}{(z-u)}\right] w(n) \quad , \quad n = 1, \ldots, 1000 \quad , \tag{25}$$

where $w(n)$ is unit variance, Gaussian white noise. Simulations are conducted with wide, medium, and narrow bandwidth data (u=0.3, 0.8, and 0.99, respectively.) The noise sequence $w(n)$ has been varied for several representative cases and was found to have no substantial effect on the identifications.

Figure 1 shows the convergence of the autoregressive coefficient for a first order, wide-band process with λ=0.95. The results are identical for all three algorithms to machine precision. The large "spikes" are associated with near singularity of the autocorrelation matrix R_N in the PBP algorithm and with division by a parameter approaching zero in the lattice algorithms. These spikes can be avoided by validating the parameters with a variety of simple heuristic tests. One method, successfully applied to lattice parameter convergence problems encountered in the identification of the modes of flexible structures, compares the absolute sum of ten successive changes in the estimated parameters against a threshold [5,6]. If the sum exceeds the threshold, past estimates of the parameters are held until the condition improves.

Figures 2–7 show the absolute numerical error in the autoregressive coefficient using wide-band data. For λ=1.0, Figures 2–4 show that for all three algorithms, this error is on the order of single precision machine epsilon. In Figures 5 and 6, with λ=0.95, the numerical error in the UPIV and NPIV algorithms is again on the order of machine epsilon, except at points where the ill-conditioning problem arises. Comparison of Figure 1 with 5 and 6 reveals that spikes in the autoregressive coefficient estimate correspond to spikes in the numerical error. Although not shown, the

numerical error in the PBP algorithm shows the same spike phenomenon.

Parameter magnitude growth is studied by running the UPIV and NPIV algorithms at double precision and plotting the lattice parameters. The data used for these comparisons is six-pole filtered white noise, generated according to

$$x(n) = \left[\frac{z^6}{(z-0.9)^6}\right] w(n) \quad , \quad n = 1,\ldots,1000 \quad . \quad (26)$$

Figures 7 and 8 show the variation of the lattice parameter σ for λ=1.0 and wide-band data. In the UPIV algorithm (Figure 7), σ reaches 700 at the 1000th data point. (In the most extreme case we have observed, with λ=1.0 and narrow-band data, σ reached 80,000 in 1000 time steps.) In the normalized algorithm (Figure 6), σ remains between zero and 0.8 for all time steps.

For λ=0.95, Figures 9 and 10 show how the NPIV algorithm greatly reduces the dynamic range of σ in comparison to the UPIV algorithm.

Finally, Figure 11 is a histogram of Figure 8, showing the distribution of the normalized parameter σ for the NPIV algorithm. The clustering of the magnitude of the parameter around 0.75 is evident.

IV. Conclusions

Because of space limitations, we have presented only a few representative plots. However, results from a large number of simulations with various data bandwidths, forgetting factors, and model orders have allowed us to make the following general observations:

- Smaller forgetting factors and wider data bandwidths tend to increase the variance of parameter estimations.
- All three instrumental variable algorithms are equally vulnerable to ill-conditioning that arises with unfortunate data sets. Ill-conditioning leads to sudden "explosions" of estimated parameters, but there are known methods of detecting and avoiding this phenomenon [5,6]. Moreover, recovery from these errors is very fast, typically occuring in a few time steps.
- Numerical error is consistently on the order of machine epsilon for all three algorithms. While the error may sometimes suddenly deviate from machine epsilon, this phenomenon is associated with the ill-conditioning problem. Over a period of 1000 data points, all three instrumental variable algorithms are stable with respect to numerical error.
- The NPIV algorithm keeps its internal parameters near one in magnitude, except when ill-conditioning arises. If a parameter test is used to detect and avoid ill-conditioning, "explosions" of the internal parameters are of no consequence.

In summary, no surprising numerical behavior has been found in any of the three instrumental variable algorithms. The NPIV algorithm effectively keeps its internal parameters small in a variety of test cases. Current studies along these lines involves an application to system identification and adaptive control schemes on flexible mechanical structures.

References

[1] P. Fabre and C. Gueguen, "Improvement of the fast recursive least-squares algorithms via normalization: A comparative study," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-34, no. 2, pp. 296–308, April 1986.

[2] R. L. Moses and A. A. Beex, "Instrumental variable adaptive array processing," *IEEE Transactions on Aerospace and Electronic Systems*, 1985. (submitted).

[3] R. L. Moses and A. A. Beex, "Normalized prewindow instrumental variable algorithm," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1986. (submitted).

[4] R. L. Moses, J. A. Cadzow, and A. A. Beex, "A recursive procedure for ARMA modeling," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-33, no. 4, pp. 1188–1196, October 1985.

[5] N. Sundararajan and R. C. Montgomery, "Identification of structural dynamics systems using least-square lattice filters," *Journal of Guidance, Control, and Dynamics*, vol. 6, no. 5, pp. 374–381, September 1983.

[6] J. P. Williams and R. C. Montgomery, "Experimental implementation of parameter adaptive control on a free–free beam," In *Proceedings of the Fourth VPI & AIAA Symposium on Dynamics and Control of Large Structures*, June 1983.

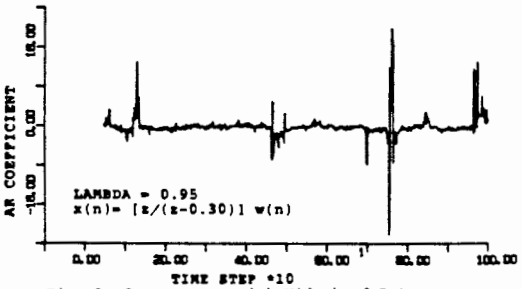

Fig. 1 Convergence with Wide-band Data

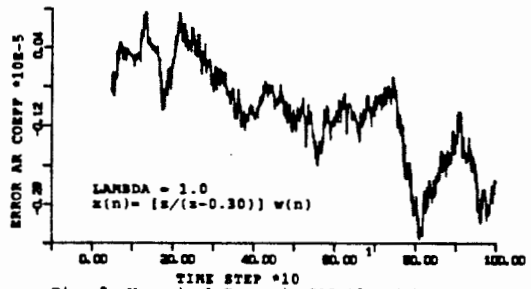

Fig. 2 Numerical Error in PBP Algorithm

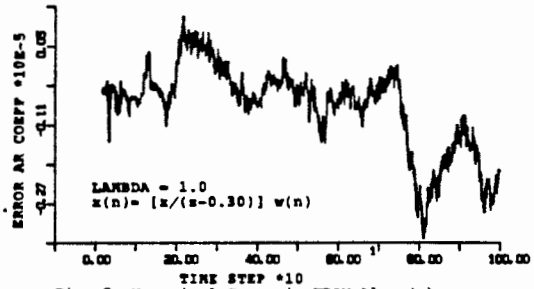

Fig. 3 Numerical Error in UPIV Algorithm

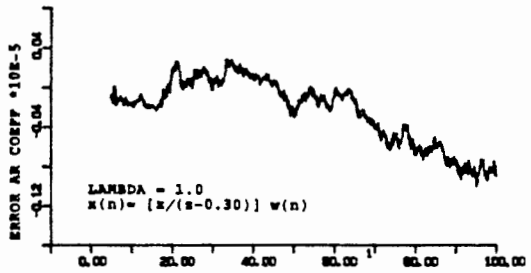

Fig. 4 Numerical Error in NPIV Algorithm

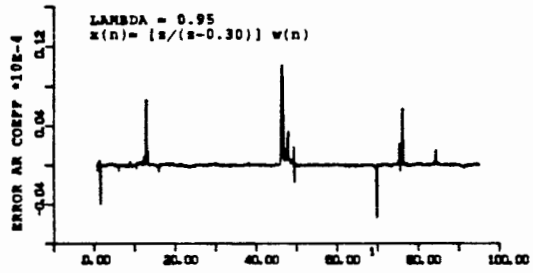

Fig. 5 Numerical Error in UPIV Algorithm

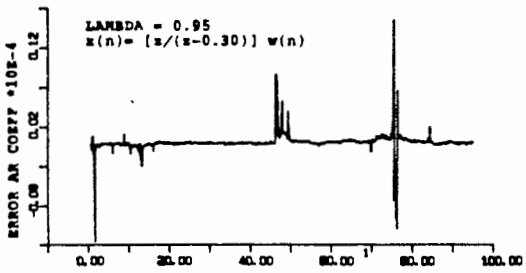

Fig. 6 Numerical Error in NPIV Algorithm

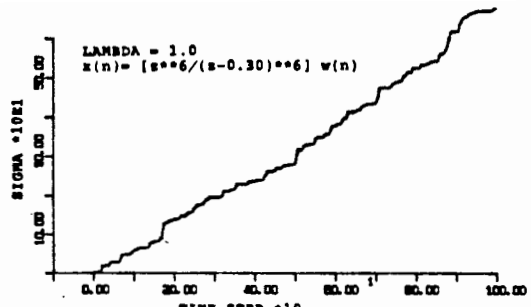

Fig. 7 Parameter Magnitude in UPIV Algorithm

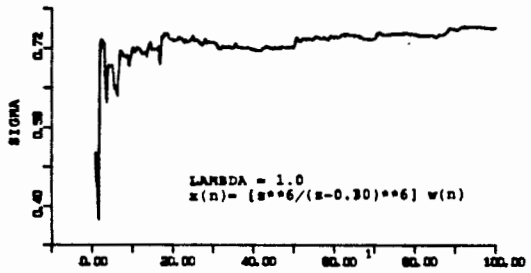

Fig. 8 Parameter Magnitude in NPIV Algorithm

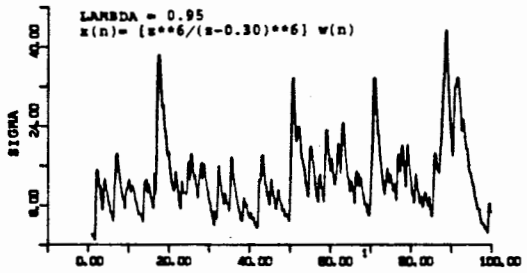

Fig. 9 Parameter Magnitude in UPIV Algorithm

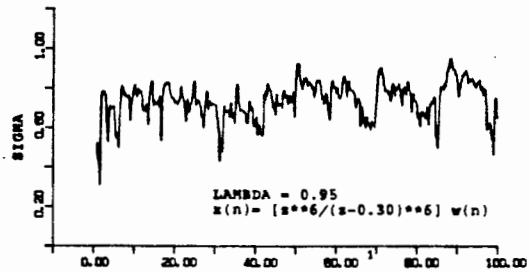

Fig. 10 Parameter Magnitude in NPIV Algorithm

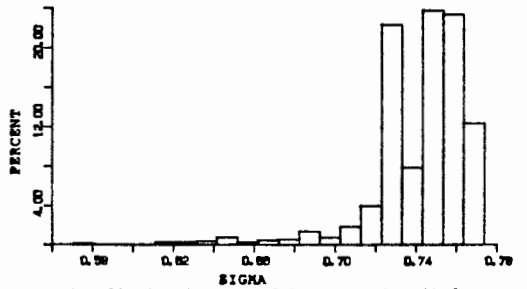

Fig. 11 Distribution of Parameter Magnitudes In NPIV Algorithm