Proceedings of the Institute of Acoustics 'Spectral Analysis and its Use in Underwater Acoustics': Underwater Acoustics Group Conference, Imperial College, London, 29–30 April 1982

EFFICIENT COMPUTATION OF AN ARMA SPECTRAL ESTIMATE

Benjamin Friedlander

Systems Control Technology, Inc.

Abstract

An ARMA spectral estimation technique based on the modified Yule-Walker equations is presented. Two recursive lattice algorithms are proposed for estimating the AR and MA spectral parameters. These computationally efficient algorithms provide spectral estimates of different orders.

1. Introduction

High resolution spectral estimation techniques based on autoregressive (AR) and autoregressive moving-average (ARMA) models have become increasingly popular in recent years. Efficient techniques for estimating the parameters of AR models have been developed by many authors [1],[2]. The estimation of ARMA parameters is considerably more difficult. Exact maximum likelihood methods provide, perhaps, the best estimates. However, the computational complexity of these techniques is quite high [3]. Practical ARMA modeling programs are based on suboptimal versions of the maximum likelihood method [4],[5] or on the modified Yule-Walker method [6],[7]. In this paper we present an efficient algorithm for estimating ARMA parameters using the latter method.

We start by a brief description of an ARMA technique based on the modified Yule-Walker equation. This algorithm is a modification of some previously proposed ARMA estimation techniques [6],[7]. The motivation for this approach and an analysis of its performance are given in [8]. The objective in this paper is to present a particular way for carrying out the computations involved in this technique.

Let y_t be an ARMA process of order (M,N), i.e.,

$$y_t = -\sum_{i=1}^{N} A_i y_{t-i} + \sum_{i=1}^{M} B_i v_{t-i} + v_t \qquad (1)$$

where v_t is a white noise process. It is straightforward to show that the AR parameters obey the recursion

$$R_k + \sum_{i=1}^{N} A_i R_{k-i} = 0 \quad k > N \qquad (2)$$

where $\{R_i\}$ are the true correlation coefficients of the process. Writing this in matrix form for k = N+1,...,K we get

This work was supported by the Office of Naval Research under Contract Number N00014-81-C-0300.

$$\begin{bmatrix} R_N & \cdots & R_1 \\ R_{N+1} & \cdots & R_2 \\ \vdots & & \vdots \\ R_{K-1} & \cdots & R_{K-N} \end{bmatrix} \begin{bmatrix} A_1 \\ \vdots \\ A_N \end{bmatrix} = - \begin{bmatrix} R_{N+1} \\ \vdots \\ R_K \end{bmatrix} \tag{3}$$

This equation often appears in the literature on stochastic realization (sometimes called the modified Yule-Walker equation), usually with $K = 2N$. When the correlation coefficients $\{R_i\}$ are replaced by their estimates $\hat{R}_i = 1/T \sum_{t=i} y_t y'_{t-i}$ (autocorrelation method), Eq. (3) holds only approximately and has to be solved in a least-squares sense. It has been noted that the (statistical) efficiency of the resulting estimates can be increased by choosing $K > 2N$. Furthermore, it has been shown that overestimating the model order (i.e. choosing N larger than the true order of the AR model) is essential for obtaining good results at low signal-to-noise ratios [7],[8]. This technique is closely related to the Instrumental Variable method of parameter estimation [9]. Various least-squares techniques such as Singular Value Decomposition (SVD) can be used to solve for the AR parameters. In Section 2 we present a recursive lattice algorithm for performing these computations.

The MA part of the spectrum is estimated next. Let $C(z)$ be defined as the numerator polynomial related to the causal part of the covariance sequence

$$R_+(z) = A^{-1}(z)\, C(z) \quad , \quad R_+(z) \triangleq \sum_{i=1}^{\infty} R_i z^{-i}$$

$$A(z) = I + A_1 z^{-1} + \ldots + A_N z^{-N} \tag{4}$$

$$C(z) = C_1 z^{-1} + \ldots + C_M z^{-M}$$

Denote the impulse response of the inverse AR filter by $H(z)$,

$$H(z) = A^{-1}(z) \tag{5}$$

Writing the relationship $R_+(z) = H(z)\, C(z)$ in matrix form we get

$$\begin{bmatrix} h_0 & & 0 \\ \vdots & \ddots & \\ h_{M-1} & \cdots & h_0 \\ \vdots & & \vdots \\ h_{L-1} & \cdots & h_{L-M} \end{bmatrix} \begin{bmatrix} C_1 \\ \vdots \\ C_M \end{bmatrix} = \begin{bmatrix} R_1 \\ \vdots \\ R_L \end{bmatrix} \tag{6}$$

This equation can be solved using any least-squares technique (e.g. SVD). In Section 3 we present a recursive lattice algorithm for computing the $\{C_i\}$ parameters. The spectrum of the ARMA process can finally be evaluated by

$$S(\omega) = A^{-1}(e^{j\omega})\, C(e^{j\omega}) + R_0 + C'(e^{-j\omega})\, A^{-T}(e^{-j\omega}) =$$

$$= A^{-1}(e^{j\omega}) \left[C(e^{j\omega})\, A'(e^{-j\omega}) + A(e^{j\omega})\, R_0 A'(e^{-j\omega}) + \right. \tag{7}$$

$$\left. + A(e^{j\omega})\, C'(c^{-j\omega}) \right] A^{-T}(e^{-j\omega})$$

The MA part $B(z)$ can be computed by performing spectral factorization

$$B(z)\, B(z^{-1}) = C(z)\, A'(z^{-1}) + A(z)\, R_0 A'(z^{-1}) + A(z)\, C'(z^{-1}) \tag{8}$$

Note, however, that $B(z)$ is not needed for computing the ARMA spectrum, since only the product $B(z)\,B(z^{-1})$ is required. One potential difficulty with the MA estimation procedure discussed above is that the right hand side of Eq. (8) is not guaranteed to be positive real, leading to possible negative values for the spectrum $S(\omega)$. Increasing the zero-lag correlation coefficient R_0 and windowing the data or the correlation coefficients are practical methods of ensuring positive realness [13]. See [8] for a more detailed discussion of this point.

2. AR Estimation Algorithm

Consider the problem of estimating the parameters of the optimal predictor for a time-series z_t:

$$z_t = -\sum_{i=1}^{N} a_i z_{t-i} + \varepsilon_t \qquad (9)$$

where ε_t is the prediction error. The predictor coefficients that minimize the sum of squared prediction errors $\sum \varepsilon_t \varepsilon_t$ (summed from t=N+1 to K) are given as the solution of the so-called normal equations

$$\begin{bmatrix} z_N & \cdots & z_1 \\ z_{N+1} & \cdots & z_2 \\ \vdots & & \\ z_{K-1} & \cdots & z_{K-N} \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_N \end{bmatrix} = - \begin{bmatrix} z_{N+1} \\ \vdots \\ z_K \end{bmatrix} \qquad (10)$$

This particular problem formulation appears in speech processing applications under the name of the covariance method. A recursive lattice algorithm was developed in [10] for computing the $\{a_i\}$ coefficients in (10). This algorithm is recursive both in predictor order N and in time t, and is computationally efficient. Note that Eqs. (3) and (10) are identical (with R_i replaced by z_i and A_i by a_i). Thus the lattice algorithm developed for solving the linear prediction problem can be used to solve the modified Yule-Walker equations.

The computations proceed in two steps: A set of lattice parameters are computed from the given covariance sequence $\{R_i\}$, as summarized in Table 1. These parameters are then used to form the lattice prediction filter $A(z)$. By computing the impulse response of this filter we obtain the AR coefficients A_i, as summarized in Table 2. For a more detailed explanation of this algorithm see [10],[11].

The recursive algorithm generates a sequence of solutions of all orders and times up to K,N. This makes it possible to choose the optimal order and to pick up the corresponding solution without additional computations. Using other least-squares methods will generally require re-solving Eq. (3) for different orders and different sizes of the covariance matrix. We have used the Akaike Information Criterion to determine the "optimal" order N [8]. This criterion can be computed in a trivial way from the lattice parameters. The maximum time lag is determined by comparing the correlation coefficient R_K to σ^4/K where σ^2 is the measurement noise variance.

The computational requirements of these lattice algorithms are proportional to NK (Table 1) and N^2 (Table 2) operations. For comparison consider a least-squares method in which Eq. (3) is solved by computing a pseudo-inverse, which requires in the order of $N^2K + N^3$ operations. For high orders N the lattice method will require less computations.

The prediction filter A(z) can be rewired to form the inverse filter $H(z) = A^{-1}(z)$ which is needed in the MA estimation part. The algorithm in Table 2 will be modified as follows:

Initialize: $\bar{B}_{0,i} = \bar{C}_{0,i} = \bar{D}_{0,i} = 0$

Set $\bar{A}_{N,i} = \bar{A}_{N,0}$ for $i=0$ and $\bar{A}_{N,i} = 0$ for $i > 0$.

Do for $p=N-1,\ldots,0$

$$\hat{A}_{p+1,i} = G^{-1}(\bar{A}_{p+1,i}, \bar{B}_{p,i-1}, K_{p+1})$$

$$\tilde{A}_{p+1,i} = G^{-1}(\hat{A}_{p+1,i}, \bar{D}_{p,i-1}, d_{p+1})$$

$$\bar{A}_{p,i} = G(\tilde{A}_{p+1,i}, \bar{C}_{p,i-1}, \varepsilon_{p+1})$$

Update $\bar{B}_{p,i}$, $\bar{C}_{p,i}$, $\bar{D}_{p,i}$ as in Table 2, for $p=0,\ldots,N-1$.

Set $h_i = S^{1/2}\bar{A}_{0,i}$

In other words, the sequence $\{h_i\}$ is computed as the impulse response of the inverse of the prediction filter.

3. MA Estimation Algorithm

Consider the problem of estimating the parameters of the optimal predictor for a time-series x_t from measurements of a related time series z_t:

$$x_t = \sum_{i=1}^{M} b_i z_{t-i} + \varepsilon_t^x \qquad (11)$$

where ε_t^x is the prediction error. The predictor coefficients that minimize the sum of squared prediction errors are given as the solution of the normal equations

$$\begin{bmatrix} z_0 & & 0 \\ \vdots & \ddots & \\ z_{M-1} & \cdots & z_0 \\ \vdots & & \vdots \\ z_{L-1} & \cdots & z_{L-M} \end{bmatrix} \begin{bmatrix} b_1 \\ \vdots \\ b_M \end{bmatrix} = \begin{bmatrix} x_1 \\ \vdots \\ \vdots \\ x_L \end{bmatrix} \qquad (12)$$

The assumption that $0 = z_{-1} = z_{-2} = \ldots$ is implicit in this equation, i.e., the data are "pre-windowed". A recursive lattice algorithm for solving Eq. (12) for the predictor parameters $\{b_i\}$ was derived in [12] under the name of the joint-process lattice form. Comparison of Eqs. (12) and (6) reveals the similarity of the MA estimation problem and the problem of "predicting" $\{R_i\}$ from past values of $\{h_i\}$. The lattice algorithm is summarized in Tables 3 and 4. For a derivation and more detailed description of this algorithm see [11],[12]. Note that the lattice algorithm provide estimates of the MA spectral parameters for all orders up to M. This makes it possible to choose the "best" order without repeating the computations.

4. Simulation Results

The computational experience with the lattice algorithms presented in this paper is fairly limited. In this section we present some preliminary simulation

results to illustrate the behavior of the algorithm. Four test cases are presented, involving two sinusoids in noise.

Case 1

A time series was generated by

$$y_t = \sqrt{6.3} \sin(0.4\pi t) + \sqrt{20} \sin(0.6\pi t) + n_t \qquad (13)$$

with n_t being a unit variance white noise process. The signal-to-noise ratio was 5 dB and 10 dB respectively for the two sinusoids. An ARMA (10,10) model was estimated based on 100 data points. The resulting spectrum in depicted on Fig. 1; see [13] for comparison.

Case 2

$$y_t = \sqrt{2} \sin(0.4\pi t) + \sqrt{2} \sin(0.6\pi t) + n_t \qquad (14)$$

The signal-to-noise ratio of both sinusoids was 0 dB. An ARMA (10,10) model was estimated, based on 100 data points. The spectral estimate is depicted in Fig. 2; see [13] for comparison.

Case 3

$$y_t = \sqrt{20} \sin(0.4\pi t) + \sqrt{2} \sin(0.426\pi t) + n_t \qquad (15)$$

The signal-to-noise ratios were 10 dB and 0 dB. An ARMA (15,15) model was estimated based on 1024 data points. The spectrum is depicted in Fig. 3; see [7] for comparison.

Case 4

$$y_t = \sqrt{2} \sin(0.32812\pi t) + \sqrt{2} \sin(0.5\pi t) + n_t \qquad (16)$$

The signal-to-noise ratio of both sinusoids was 0 dB. An ARMA (5,5) model was estimated, based on 64 data points. The resulting spectrum is depicted in Fig. 4; see [7] for comparison.

5. Conclusions

Two types of lattice algorithms were proposed for solving the linear equations arising in the problem of estimating ARMA spectra. The modular structure of these lattice forms has some advantages when spectral estimation algorithms need to be implemented in special purpose hardware. When spectral estimation is performed using a general purpose computer it is possible to use any robust least-squares technique for solving these equations. The potential advantage of the lattice approach in this case is one of computational efficiency, especially when it is desired to look at spectra of different orders.

References

1. D.G. Childers, Modern Spectral Estimation, IEEE Press, 1978.
2. J.D. Markel and A.H. Gray, Jr., Linear Prediction of Speech, Springer-Verlag, 1976.
3. C. Gueguen and L.L. Scharf, "Exact Maximum Likelihood Identification of ARMA Models: A Signal Processing Perspective," in Signal Processing: Theories

and Applications, M. Kunt and F. deCoulon, eds., North Holland Publishing, Co., 1980.
4. G.E.P. Box and G.M. Jenkins, Time Series Analysis: Forecasting and Control, Holden-Day, San Francisco, California, 1970.
5. H. Akaike, "Maximum Likelihood Identification of Gaussian Autoregressive Moving Average Models," Biometrika, vol. 60, no. 2, pp. 255-265, 1973.
6. T.W. Anderson, The Statistical Analysis of Time Series, John Wiley & Sons, 1971.
7. J.A. Cadzow, "High Performance Spectral Estimation - A New ARMA Method," IEEE Trans. Acoustics, Speech, Signal Processing, vol. ASSP-28, no. 5, pp. 524-529, October 1980.
8. B. Friedlander and B. Porat, "An Improved Estimator for ARMA Spectra," submitted for publication.
9. B. Friedlander, "Instrumental Variable Methods for ARMA Spectral Estimation," IEEE Intl. Conf. Acoustics, Speech, Signal Processing, May 1982.
10. B. Porat, B. Friedlander, and M. Morf, "Square-Root Covariance Ladder Algorithms," Proc. Intl. Conf. Acoustics, Speech, Signal Processing, Atlanta, Georgia, March 1981. Also to appear in IEEE Trans. Automatic Control.
11. B. Friedlander, "Lattice Methods for Spectral Estimation," Proc. IEEE, to appear.
12. D. Lee, M. Morf, and B. Friedlander, "Square-Root Covariance Ladder Algorithms," IEEE. Acoustics, Speech, Signal Processing, vol. ASSP-29, no. 3, pp. 627-641, June 1981.
13. S.M. Kay, "A New ARMA Spectral Estimator," IEEE Trans. Acoustics, Speech, Signal Processing, vol. ASSP-28, no. 5, pp. 585-588, October 1980.

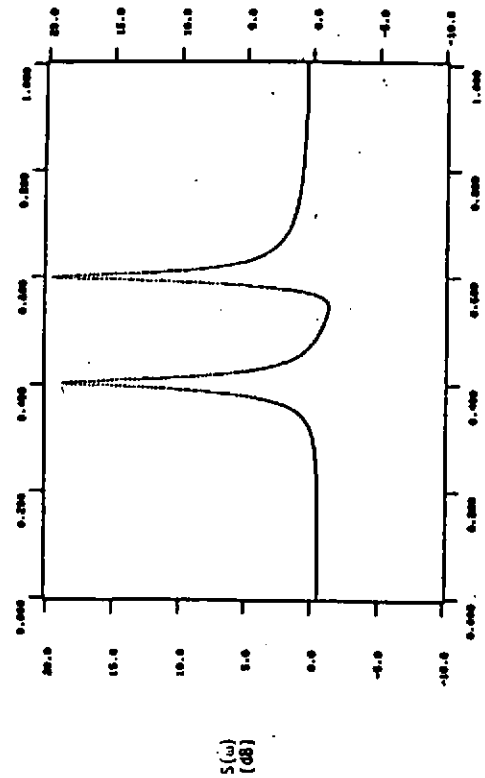

Figure 2: Spectral estimate for case 2
N=M=10, L=K=40, 100 data points

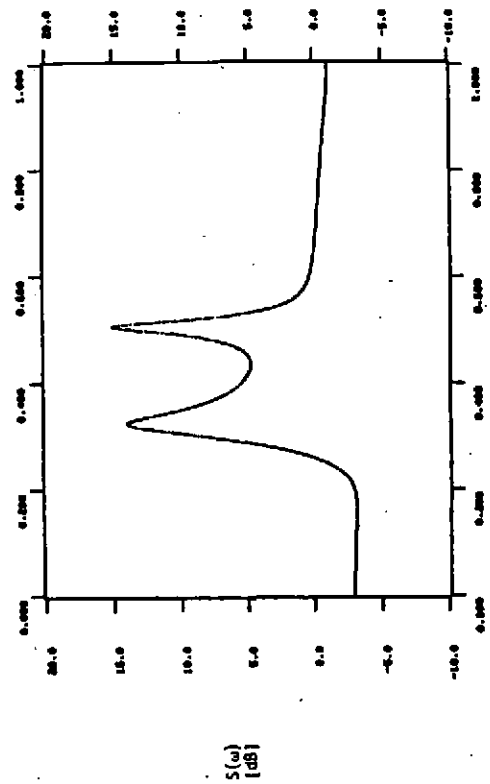

Figure 4: Spectral estimate for case 4
N=M=5, L=K=20, 64 data points

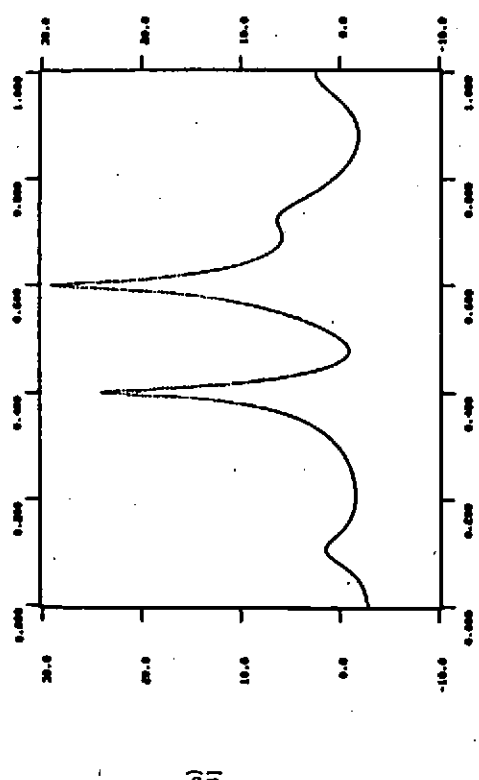

Figure 1: Spectral estimate for case 1
N=M=10, L=K=40, 100 data points

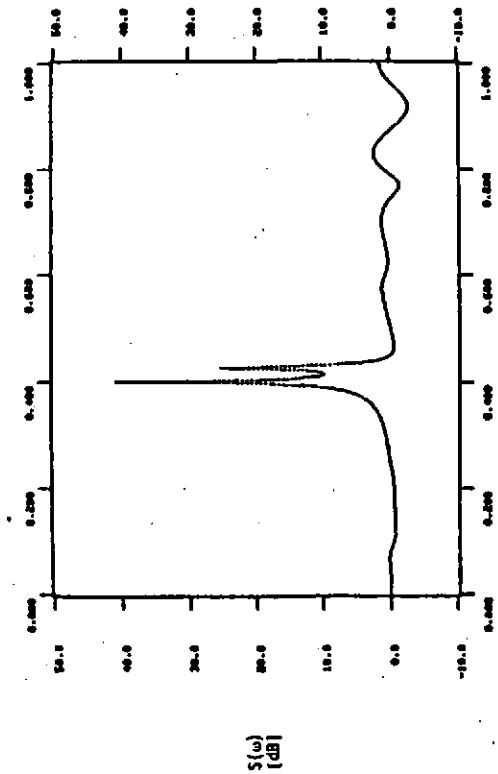

Figure 3: Spectral estimate for case 3
N=M=15, L=K=80, 1024 data points

Table 1: The (Growing Memory) Covariance Lattice Form for Computing Reflection Coefficients

Initialize:

$$d_{1,1} = (R_1 R_1')^{-1/2} R_1$$

For $t = 1,\ldots,K$ do:

$$S_t = S_{t-1} + R_t R_t'$$

$$\tilde{\beta}_{0,t} = 0$$

$$\tilde{\epsilon}_{0,t} = \tilde{r}_{0,t} = S_t^{-1/2} R_t$$

For $p=0,\ldots,N-1$ do:

$$d_{p+1,t} = F^{-1}(d_{p+1,t-1}, \tilde{\beta}'_{p,t-1}, \tilde{\epsilon}_{p,t})$$

$$K_{p+1,t} = F^{-1}(K_{p+1,t-1}, \tilde{r}'_{p,t-1}, \tilde{\epsilon}_{p,t})$$

$$\tilde{\epsilon}^1_{p,t} = F(\tilde{\epsilon}_{p,t}, \tilde{\beta}_{p,t-1}, d_{p+1,t})$$

$$\tilde{\beta}_{p+1,t} = F(\tilde{\beta}_{p,t-1}, \tilde{\epsilon}_{p,t}, d'_{p+1,t})$$

$$\tilde{\epsilon}_{p+1,t} = F(\tilde{\epsilon}^1_{p,t}, \tilde{r}_{p,t-1}, K_{p+1,t})$$

$$\tilde{r}_{p+1,t} = F(\tilde{r}_{p,t-1}, \tilde{\epsilon}^1_{p,t}, K'_{p+1,t})$$

$$F(U,V,W) \triangleq [I - WW']^{-1/2} [U - WV][I - VV']^{-T/2}$$

$$F^{-1}(U,V,W) \triangleq [I - WW']^{1/2} U[I - VV']^{T/2} + WV$$

The prediction filter parameters (Table 2) are given by $K_{p+1} = K_{p+1,K}$, $\epsilon_{p+1} = \tilde{\epsilon}_{p,K}$, $d_{p+1} = d_{p+1,K}$, $S = S_K$.

Table 2: The (Growing Memory) Covariance Lattice Prediction Filter

For $i=0,\ldots,N$ do:

$$\bar{A}_{0,0} = \bar{B}_{0,0} = S^{-1/2} \qquad \bar{C}_{0,0} = \bar{D}_{0,0} = 0 \qquad i = 0$$

$$\bar{A}_{0,i} = \bar{B}_{0,i} = \bar{C}_{0,i} = \bar{D}_{0,i} = 0 \qquad i > 0$$

For $p=0,\ldots,N-1$ do:

$$\bar{A}_{p+1,i} = G^{-1}(\bar{A}_{p,i}, \bar{C}_{p,i-1}, \epsilon_{p+1})$$

$$\bar{C}_{p+1,i} = \bar{G}^{-1}(\bar{C}_{p,i-1}, \bar{A}_{p,i}, -\epsilon'_{p+1})$$

$$\hat{A}_{p+1,i} = G(\bar{A}_{p+1,i}, \bar{D}_{p,i-1}, d_{p+1})$$

$$\bar{D}_{p+1,i} = G(\bar{D}_{p,i-1}, \bar{A}_{p+1,i}, d'_{p+1})$$

$$\bar{A}_{p+1,i} = G(\hat{A}_{p+1,i}, \bar{B}_{p,i-1}, K_{p+1})$$

$$\bar{B}_{p+1,i} = G(\bar{B}_{p,i-1}, \hat{A}_{p+1,i}, K'_{p+1})$$

To unnormalize: $A_N(z) = \bar{A}_{N,0}^{-1} \bar{A}_N(z)$

$$G(U,V,W) \triangleq [I - WW']^{-1/2} [U - WV]$$

$$G^{-1}(U,V,W) \triangleq [I - WW']^{1/2} U + WV$$

$$\bar{G}^{-1}(U,V,W) \triangleq [I - WW']^{T/2} U + [I - WW']^{-1/2} W[I - W'W]^{1/2} V$$

Note: in the scalar case $\bar{G}^{-1}(U,V,W) = G^{-1}(U,V,W)$

Table 3: The Joint-Process (Pre-Windowed) Lattice Form for Computing Reflection Coefficients

For $t=1,\ldots,L$ do:

$$S_t = S_{t-1} + h_{t-1} h'_{t-1}$$

$$S^x_t = S^x_{t-1} + R_t R_t'$$

$$\tilde{\epsilon}_{0,t} = \tilde{r}_{0,t} = S_t^{-1/2} h_{t-1}$$

$$\tilde{\epsilon}^x_{0,t} = (S^x_t)^{-1/2} R_t$$

For $p=0,\ldots,M-1$

$$K_{p+1,t} = F^{-1}(K_{p+1,t-1}, \tilde{r}'_{p,t-1}, \tilde{\epsilon}_{p,t})$$

$$\tilde{\epsilon}_{p+1,t} = F(\tilde{\epsilon}_{p,t}, \tilde{r}_{p,t-1}, K_{p+1,t})$$

$$\tilde{r}_{p+1,t} = F(\tilde{r}_{p,t-1}, \tilde{\epsilon}_{p,t}, K'_{p+1,t})$$

$$K^x_{p+1,t} = F^{-1}(K^x_{p+1,t-1}, \tilde{r}'_{p,t}, \tilde{\epsilon}^x_{p,t})$$

$$\tilde{\epsilon}^x_{p+1,t} = F(\tilde{\epsilon}^x_{p,t}, \tilde{r}_{p,t}, K^x_{p+1,t})$$

The prediction filter parameters (Table 4) are given by $K_p = K_{p,L}$, $K^x_p = K^x_{p+1,L}$, $\epsilon_{p+1} = \tilde{\epsilon}_{p,L}$, $\epsilon^x_p = \tilde{\epsilon}^x_{p,L}$, $S = S_L$

Table 4: The Joint-Process (Pre-Windowed) Lattice Prediction Filter

For $i=0,\ldots,M$ do:

$$\bar{A}_{0,0} = \bar{B}_{0,0} = S^{-1/2} \qquad \bar{C}_{0,0} = 0 \qquad \bar{A}^x_{0,0} = -[I - K^x_1 K^{x'}_1]^{-1/2} K^x_1 S^{-1/2}, \quad i=0$$

$$\bar{A}_{0,i} = \bar{B}_{0,i} = \bar{C}_{0,i} = \bar{A}^x_{0,i} = 0, \quad i > 0$$

For $p=0,\ldots,M-1$ do:

$$\bar{A}_{p+1,i} = G^{-1}(\bar{A}_{p,i}, \bar{C}_{p,i-1}, \epsilon_{p+1})$$

$$\bar{C}_{p+1,i} = \bar{G}^{-1}(\bar{C}_{p,i-1}, \bar{A}_{p,i}, -\epsilon'_{p+1})$$

$$\bar{A}_{p+1,i} = G(\bar{A}_{p+1,i}, \bar{B}_{p,i-1}, K_{p+1})$$

$$\bar{B}_{p+1,i} = G(\bar{B}_{p,i-1}, \bar{A}_{p+1,i}, K'_{p+1})$$

$$\bar{A}^x_{p+1,i} = G^{-1}(\bar{A}^x_{p,i}, \bar{C}_{p+1,i}, \epsilon^x_{p+1})$$

$$\bar{A}^x_{p+1,i} = G(\bar{A}^x_{p+1,i}, \bar{B}_{p+1,i}, K^x_{p+1})$$

Finally set $C_{i+1} = \bar{A}^x_{M,i}$ for $i=0,\ldots,M-1$