

# Autoregressive Modeling: Elementary Least-Squares Methods

Harvey Thornburg  
Center for Computer Research in Music and Acoustics (CCRMA)  
Department of Music, Stanford University  
Stanford, California 94305

February 19, 2006

# Introduction and review

---

- We consider the multichannel autoregressive model:

$$y_n + A_1 y_{n-1} + \dots + A_p y_{n-p} = e_n \quad (1)$$

where  $\{y_n\}$ ,  $n = 1 \dots N$  are  $(m \times 1)$  *vector-valued* observations, and  $\{A_k\}$ ,  $k = 1 \dots p$  are  $(m \times m)$  *matrix-valued* model parameters. Though most audio applications involve only scalar observations, some important extensions (e.g. time-variant models) can, under suitable transformation, be addressed within the multichannel framework. Everything proceeds as in the scalar case, except we must be careful with the order of matrix multiplications.

## Optimization criteria

- Let  $M^*$  denote the Hermitian (conjugate) transpose of a matrix  $M$ . We choose  $\{A_k\}$  such that the sum-of-squares error:

$$J(\{A_k\}) = \sum e_n e_n^* \quad (2)$$

is minimal, in the sense indicated by positive semidefinite (PSD) matrices: If  $J$  is the cost due to

an optimal set of model parameters and  $J'$  is the cost due to another set then  $J' - J$  is PSD; i.e. all quadratic forms  $x(J' - J)x^*$  are nonnegative. Practical consequences of this “PSD-minimal” criterion are:

1. We minimize the sum-of-squares of *any* linear combination of error components: Let  $w$  be a weight vector, and define the cost due to  $\{A_k\}$ :

$$\begin{aligned}
 j(w) &= \sum |w^* e_n|^2 \\
 &= \sum w^* e_n e_n^* w \\
 &= w^* \left( \sum e_n e_n^* \right) w \\
 &= w^* J w
 \end{aligned} \tag{3}$$

Similarly, let  $j'(w) = w^* J' w$  be the cost due to  $\{A'_k\}$ .

Since  $J' - J$  is PSD, it follows that

$w^*(J' - J)w \geq 0$ , and by (3) it follows:

$j'(w) \geq j(w)$ .

2. We minimize the sum of all weighted error norms, provided the weight matrix is Hermitian and PSD. Let  $W$  be that  $(m \times m)$  weight matrix, and define

the cost due to  $\{A_k\}$ :

$$\begin{aligned}
 j(W) &= \sum e_n^* W e_n \\
 &= \sum \|W^{1/2} e_n\|^2 \\
 &= \text{Tr} \left[ W^{1/2} \left( \sum e_n e_n^* \right) W^{*/2} \right] \\
 &= \text{Tr}(W^{1/2} J W^{*/2}) \tag{4}
 \end{aligned}$$

Similarly  $j'(W) = \text{Tr}(W^{1/2} J' W^{*/2})$  is the cost due to  $\{A'_k\}$ .

The existence of  $W^{1/2}$  such that  $W = W^{*/2} W^{1/2}$  is justified by the Hermitian/PSD properties of  $W$ . If  $J' - J$  is PSD,  $x^*(J' - J)x \geq 0$  for any vector  $x$ . Given a vector  $y$ , choose  $x = W^{1/2}y$ .

Substituting, we find

$y^*(W^{1/2} J' W^{*/2} - W^{1/2} J W^{*/2})y \geq 0$ ; because  $y$  is arbitrary,  $(W^{1/2} J' W^{*/2} - W^{1/2} J W^{*/2})$  is PSD. So we have shown if  $J$  is PSD-minimal then so is  $W^{1/2} J W^{*/2}$ .

Finally, a PSD matrix has a nonnegative trace. This follows because the trace of a matrix is the sum of its eigenvalues and all eigenvalues of a PSD matrix are nonnegative. Using this fact and (4) it follows that  $j'(W) - j(W) \geq 0$  for all  $W$ , as was to be shown.

- The first criterion (3) follows from the second (4) by setting  $W = ww^*$ , but it's easier to prove directly.

## Least-squares solution and data windowing

- Observations come in a finite interval, say  $n = 1 \dots N$ . So far we haven't considered the range of the error-summation (2). Writing the model equation (1) for every  $n = 1 \dots N$  and collecting the equations in a single matrix-vector equation gives:

$$AT = H \tag{5}$$

where

$$T = \begin{bmatrix} y_0 & y_1 & \dots & y_{N-1} \\ y_{-1} & y_0 & \dots & y_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ y_{1-p} & y_{2-p} & \dots & y_{N-p} \end{bmatrix}$$

$$A = [A_1 \dots A_p]$$

$$H = [y_1 \dots y_N]$$

- Assuming  $T$  has full row rank, there exists a unique solution for the PSD-minimal least-squares criterion 2, namely the standard one:

$$A = HT^*(R)^{-1}, \text{ where}$$

$$R = (TT^*) \quad (6)$$

Let  $R \triangleq TT^*$  for the remainder of these notes. The proof is deferred to an Appendix.  $R$  is singular iff  $T$  is rank deficient, in which case optimal  $A$  exist, but are no longer unique. To simplify we assume  $R$  is nonsingular.

- The problem is that  $y_{1-p} \dots y_0$  are outside the observation window. Usually, this problem is addressed by one of three windowing methods:

1. **Covariance method** Delete columns of  $T$  (and corresponding elements of  $H$ ) until no data is accessed outside  $n = 1 \dots N$ , to obtain:

$$T_C = \begin{bmatrix} y_p & y_{p+1} & \dots & y_{N-1} \\ y_{p-1} & y_p & \dots & y_{N-2} \\ \vdots & \vdots & \ddots & \vdots \\ y_1 & y_2 & \dots & y_{N-p} \end{bmatrix}$$

$$H_C = [y_{p+1} \dots y_N] \quad (7)$$

2. **Prewindowed method** Set the inaccessible data at the beginning of the observation window  $y_{1-p} \dots y_0$  to zero. Since the first column of  $T$  is identically zero, we delete it (and the corresponding element in  $H$ ) because it offers no useful information.

$$T_P = \begin{bmatrix} y_1 & y_2 & \dots & y_{N-1} \\ 0 & y_1 & \dots & y_{N-2} \\ \vdots & 0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & y_{N-p} \end{bmatrix}$$

$$H_P = [y_2 \dots y_N] \quad (8)$$

3. **Autocorrelation method** Window the data symmetrically at both ends, setting inaccessible elements  $y_{1-p} \dots y_0, y_{N+1} \dots y_{N+p}$  to zero:

$$T_A = \begin{bmatrix} y_1 & y_2 & \dots & 0 & 0 \\ 0 & y_1 & \ddots & \vdots & \vdots \\ \vdots & 0 & \ddots & 0 & \vdots \\ \vdots & \vdots & \ddots & y_N & 0 \\ 0 & 0 & \dots & y_{N-1} & y_N \end{bmatrix}$$

$$H_A = [y_2 \dots y_N \ 0 \dots 0] \quad (9)$$

• **Remarks**

1. The covariance method has the least bias and should be used whenever conditions allow. Any

windowing has the effect of smoothing spectral peaks, causing pole estimates to be over damped.

2. The autocorrelation method involves the most windowing, but the following properties hold:
- The estimated “modeling filter”; e.g.  

$$y(n) = (I + A_1z^{-1} + \dots + A_pz^{-p})^{-1}e(n),$$
 where  $z^{-1}$  is the delay operator, is guaranteed stable. This holds even in the matrix case.
  - Thanks to the symmetry of the windowing, the matrix  $R_A = (T_A T_A^*)$  is Hermitian and has the “Toeplitz” property that the  $(i, j)$  block of dimension  $(m \times m)$ ,  $1 < i < p$ ,  $1 < j < p$  depends only on  $i - j$ . Hence  $R_A$  is a valid autocorrelation matrix for  $y_n$  treated as a stationary vector process (hence the name) The structure of  $R_A$  leads to the simple (*Levinson-Durbin*) recursion for updating model parameter estimates recursively in the order  $(p \rightarrow p + 1)$ .
  - In practice, pole frequency estimates are unaffected by autocorrelation windowing. So if you want to estimate groups of formant frequencies as features for speech recognition, but don't care about the bandwidths (commonly assumed irrelevant), use autocorrelation windowing.

3. The pre-windowed method simplifies the development of adaptive *lattice algorithms*, which update the least squares solution recursively in time ( $N \rightarrow N + 1$ ) and in the order ( $p \rightarrow p + 1$ ). It is easiest to develop the covariance lattice method as an extension of the pre-windowed method.

## Likelihood interpretation

- The use of a deterministic least squares criterion has an important statistical interpretation when the observations are Gauss-Markov. A “Gauss-Markov” process results from a finite-length all-pole filter driven by white Gaussian noise, as in (1) where  $A_1 \dots A_p$  are “filter coefficients” and  $e_n$  is the “noise”. The “Markov” property refers to the finite memory of the filter, and means that given the  $p$ th-order past  $y_{n-p:n-1}$ ,  $y_n$  is independent of the further past  $y_{1:n-p-1}$ .
- **Asymptotic efficiency of maximum likelihood** From the statistical interpretation, (1) gives a parametric form for the observations’ joint density. We wish to estimate the unknown parameters,  $A_1 \dots A_p$ , which can be thought of as a block vector, say  $\theta$ . A good, or “efficient” estimate  $\hat{\theta}$  is a *function* of observations with the following properties:
  - **Unbiased:**  $E(\hat{\theta}) = \theta$
  - **Minimum variance:**  $E \left[ (\hat{\theta} - \theta)(\hat{\theta} - \theta)^* \right]$  is PSD-minimal

By the Cramer-Rao inequality, the maximum likelihood (ML) estimate:

$$\hat{\theta} = \arg \max \{ \theta : f_{\theta}(y_{1:N}) \}$$

( $f_{\theta}$  gives the joint density) is asymptotically efficient as  $N \rightarrow \infty$ .

- Computing the ML estimate in the Gauss-Markov case, we have:

$$\begin{aligned}
f_{\{A_k\}}(y_{1:N}) &= \prod_{n=1}^N f_{\{A_k\}}(y_n | y_{1:n}) \\
&\approx \prod_{n=1}^N f_{\{A_k\}}(y_n | y_{n-p:n-1}) \\
&= \prod_{n=1}^N f_{\{A_k\}}(e_n | y_{n-p:n-1}) \\
&= \prod_{n=1}^N f_{\{A_k\}}(e_n) \tag{10}
\end{aligned}$$

The second step is justified by the Markov property: conditional independence allows us to drop conditioning on the further past  $y_{1:n-p-1}$ . The  $\approx$  is due to the absence of data for  $n < p$ . These “edge effects” wash out for large  $N$ . The third step comes by the fact that conditional on  $y_{n-p:n-1}$ ,  $e_n$  and  $y_n$  differ by a constant; thus the Jacobian for the change-of-variables  $y_n \rightarrow e_n$  is identity. Finally, the last step results from independence of the  $e_n$  and the fact  $y_n$  depends only on present and past  $e_n$ .

- Now to maximize the likelihood, it is equivalent to minimize the negative log likelihood,  $-L(\{A_k\}) = -\log f_{\{A_k\}}(y_{1:N})$ . From (10) and the form of the Gaussian likelihood:

$$\begin{aligned}
-L(\{A_k\}) &= \sum_{n=1}^N -\log \left[ (2\pi |R_e|)^{1/2} \exp \left( -e_n^* R_e^{-1} e_n / 2 \right) \right] \\
&= \frac{1}{2} \log (2\pi |R_e|) + \frac{1}{2} \sum_{n=1}^N e_n^* R_e^{-1} e_n \quad (11)
\end{aligned}$$

The first term is constant w.r.t.  $\{A_k\}$  and may be dropped. The second term is equivalent to the weighted-norm criterion (4) with  $W = R_e^{-1}$ . Hence in the Gauss-Markov case, the deterministic least-squares criterion approaches the ML criterion for large  $N$ , giving an asymptotically efficient estimate of the  $\{A_k\}$ .

# Fixed order-recursive prediction

---

- Recall that the least-squares solution for  $A$  is obtained from the matrix equation:

$$AR = HT^* \quad (12)$$

where  $R = TT^*$ .

- Using autocorrelation windowing, we verify  $R_A$  (which is Hermitian by construction) has also the block Toeplitz property. Consider the  $(i, j)$  block entry of dimension  $m$ -by- $m$ , where  $1 < i < p$ ,  $1 < j < p$ :

$$[R_A]_{(i,j)} = \sum_n y_{n+1-i} y_{n+1-j}^*$$

With  $y_{1-p} \dots y_0 = 0$ ,  $y_{N+1} \dots y_{N+p} = 0$ , we restrict the summation index to satisfy  $n > \max(i, j) - 1$  and  $n < N + \min(i, j)$ . Defining  $n' = n + 1 - j$ , the sum is rewritten:

$$\begin{aligned} [R_A]_{(i,j)} &= \sum_{\max(i,j)-j+1}^{\min(i,j)-j+N} y_{n-(i-j)} y_n^* \\ &= \sum_{\max(i-j,0)+1}^{\min(i-j,0)+N} y_{n-(i-j)} y_n^* \end{aligned}$$

which depends only on  $i - j$ .

- Thus,  $R_A$  for the order  $p$  is parameterized by  $p$  block entries:

$$R_{A,p} = \begin{bmatrix} R_0 & R_1^* & \cdots & R_{p-1}^* \\ R_1 & R_0 & \cdots & R_{p-2}^* \\ \vdots & \vdots & \ddots & \vdots \\ R_{p-1} & R_{p-2} & \cdots & R_0 \end{bmatrix}$$

- Furthermore, it is easily verified:

$$\rho_p \stackrel{\Delta}{=} H_{A,p} T_{A,p}^* = [R_1 \cdots R_p]$$

- Therefore, we see there are two ways to partition  $R_{A,p}$ , such that  $R_{A,p-1}$  appears as a submatrix in the lower left or upper right corners. This motivates a recursive algorithm. The “hard work” in 6 is that of inverting  $R_{A,p}$ : if we know already the inverse of  $R_{A,p-1}$ , perhaps it is not so hard to get the inverse of  $R_{A,p}$ .
- We choose the following partition. Defining “#” as the operation which reverses the block elements of a vector, we verify:

$$R_{A,p} = \begin{bmatrix} R_{A,p-1} & \rho_{p-1}^{*\#} \\ \rho_{p-1}^{\#} & R(0) \end{bmatrix} \quad (13)$$

## Block matrix decompositions

- The main formula we need concerns the inverse of a block matrix:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = ??$$

We take the scenic route, obtaining several other results that will be useful later. In fact, mostly the entire theory of recursive algorithms for autoregressive modeling comes from these block matrix results. The development follows that of Appendix A in [KSH00]: We omit many extensions.

- Consider the matrix-vector equation:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} E \\ F \end{bmatrix} = \begin{bmatrix} G \\ H \end{bmatrix}$$

Interpreting as two equations, if we multiply the top equation by  $-CA^{-1}$  and add to the bottom we eliminate  $E$ , obtaining:

$$(D - CA^{-1}B)F = H$$

which is easily solved for  $F$ . The quantity  $(D - CA^{-1}B)$  is called the *Schur complement* of  $A$  and will be denoted  $S_A$ ). To solve for  $E$ , we take our solution for  $F$  and substitute in to the top equation, which has been left alone:  $AE + BF = G$ .

- The elimination step is equivalent to multiplying (both sides) on the left by a matrix  $L$ , and the substitution step (using  $(D - CA^{-1}B)F$  rather than  $F$ ) is equivalent to multiplying on the right by  $U$  where

$$L = \begin{bmatrix} I & 0 \\ -CA^{-1} & I \end{bmatrix}$$
$$U = \begin{bmatrix} I & -A^{-1}B \\ 0 & I \end{bmatrix}$$

- According to the elimination and substitution steps, these matrices *block-diagonalize* the original matrix:

$$\begin{bmatrix} I & 0 \\ -CA^{-1} & I \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} I & -A^{-1}B \\ 0 & I \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & S_A \end{bmatrix}$$

- Note that to reverse the action of adding a multiple of one equation to another, we subtract that multiple. Hence, we may write:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} I & 0 \\ CA^{-1} & I \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & S_A \end{bmatrix} \begin{bmatrix} I & A^{-1}B \\ 0 & I \end{bmatrix} \quad (14)$$

- Alternatively, we could have done the elimination step by adding a multiple ( $-BD^{-1}$ ) of the bottom equation to the top. This gives a second block decomposition:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} I & BD^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} S_D & 0 \\ 0 & D \end{bmatrix} \begin{bmatrix} I & 0 \\ D^{-1}C & I \end{bmatrix} \quad (15)$$

where  $S_D = A - BD^{-1}C$  is the Schur complement of  $D$ .

- From the block decompositions we get inversion formulas:

$$\begin{aligned} \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} &= \begin{bmatrix} I & -A^{-1}B \\ 0 & I \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & S_A^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -CA^{-1} & I \end{bmatrix} \\ &= \begin{bmatrix} A^{-1} + A^{-1}BS_A^{-1}CA^{-1} & -A^{-1}BS_A^{-1} \\ -S_A^{-1}CA^{-1} & S_A^{-1} \end{bmatrix} \quad (16) \end{aligned}$$

$$\begin{aligned}
\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} &= \begin{bmatrix} I & 0 \\ -D^{-1}C & I \end{bmatrix} \begin{bmatrix} S_D^{-1} & 0 \\ 0 & D^{-1} \end{bmatrix} \begin{bmatrix} I & -BD^{-1} \\ 0 & I \end{bmatrix} \\
&= \begin{bmatrix} S_D^{-1} & -S_D^{-1}BD^{-1} \\ -D^{-1}CS_D^{-1} & D^{-1} + D^{-1}CS_D^{-1}BD^{-1} \end{bmatrix} \quad (17)
\end{aligned}$$

- **Matrix inversion lemma** An important result comes by equating block elements of the two inversion formulas, e.g:

$$(A - BD^{-1}C)^{-1} = A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}C \quad (A8)$$

- The matrix inversion lemma arises naturally when considering any *time-update* step. Time updates to  $(TT^*)^{-1}$  involve adding a block column to  $T$ , e.g.

$$T_N \rightarrow T_{N+1} = [T_N | t_{N+1}]. \text{ Hence}$$

$$(T_{N+1}T_{N+1}^*)^{-1} = (T_N T_N^* + t_{N+1} t_{N+1}^*)^{-1}. \text{ Here (18)}$$

obtains an efficient way for inverting the sum

$$T_N T_N^* + t_{N+1} t_{N+1}^*$$

## Levinson-Durbin algorithm

- We pursue a block decomposition of  $R_{A,p}$ .  
Substituting into the inversion formula (16)

$$\begin{aligned} A &= R_{A,p-1} \\ B = C^* &= \rho_{p-1} \\ D &= R(0) \end{aligned}$$

obtains:

$$\begin{aligned} R_{A,p}^{-1} &= \begin{bmatrix} R_{A,p-1} & \rho_{p-1}^{*\#} \\ \rho_{p-1}^{\#} & R(0) \end{bmatrix}^{-1} \\ &= \begin{bmatrix} R_{A,p-1}^{-1} + R_{A,p-1}^{-1} \rho_{p-1}^{*\#} \lambda_{p-1}^{-1} \rho_{p-1}^{\#} R_{A,p-1}^{-1} & -R_{A,p-1}^{-1} \rho_{p-1}^{*\#} \lambda_{p-1}^{-1} \\ -\lambda_{p-1}^{-1} \rho_{p-1}^{\#} R_{A,p-1}^{-1} & \lambda_{p-1}^{-1} \end{bmatrix} \end{aligned}$$

- The Schur complement, denoted as  $\lambda_{p-1}$ , is nothing but the PSD-minimal squared error for the model of order  $p - 1$ :

$$\begin{aligned}
\lambda_{p-1} &= R(0) - \rho_{p-1}^{\#} R_{A,p-1}^{-1} \rho_{p-1}^{*\#} \\
&= R(0) + A_{p-1}^{\#} \rho_{p-1}^{*\#} \\
&= R(0) + A_{1,p-1} R(1) + \dots + A_{p-1,p-1} R(p-1) \\
&= \sum y_n \left[ y_n + A_{1,p-1} y_{n-1} + \dots + A_{p-1,p-1} y_{n-(p-1)} \right]^* \\
&= \sum y_n e_{n,p-1}^* \\
&= \sum e_{n,p-1} e_{n,p-1}^* \tag{19}
\end{aligned}$$

The last step follows by writing

$y_n = e_{n,p-1} - A_{1,p-1} y_{n-1} - \dots - A_{p-1,p-1} y_{n-(p-1)}$  and using *orthogonality* of the error  $e_{n,p-1}$  and past outputs  $y_{n-1} \dots y_{n-(p-1)}$  under optimal choice of model parameters, i.e.

$$\sum y_{n-k} e_{n,p-1}^* = \mathbf{0}, \quad k = 1 \dots p-1 \tag{20}$$

See Appendix ?? for proof. Note that  $\lambda_p/N$  gives a (biased) estimate of the prediction error covariance of order  $p$ .

- From the formula for  $R_{A,p}^{-1}$ , we find a recursion for  $A_p$ :

$$\begin{aligned}
A_p &= \rho_p R_{A,p}^{-1} \\
&= \left[ \rho_{p-1} \mid R(p) \right] \begin{bmatrix} R_{A,p-1}^{-1} + R_{A,p-1}^{-1} \rho_{p-1}^{*\#} \lambda_{p-1}^{-1} \rho_{p-1}^\# R_{A,p-1}^{-1} & -R_{A,p-1}^{-1} \rho_{p-1}^{*\#} \lambda_{p-1}^{-1} \\ -\lambda_{p-1}^{-1} \rho_{p-1}^\# R_{A,p-1}^{-1} & \lambda_{p-1}^{-1} \end{bmatrix} \\
&= \left[ \rho_{p-1} R_{A,p-1}^{-1} - \left( R(p) - \rho_{p-1} R_{A,p-1}^{-1} \rho_{p-1}^{*\#} \right) \lambda_{p-1}^{-1} \rho_{p-1}^\# R_{A,p-1}^{-1} \mid \left( R(p) - \rho_{p-1} R_{A,p-1}^{-1} \rho_{p-1}^{*\#} \right) \lambda_{p-1}^{-1} \right] \\
&= \left[ A_{p-1} - k_p A_{p-1}^\# \mid k_p \right] \tag{21}
\end{aligned}$$

In the final step, we define  $k_p = \Delta_{p-1} \lambda_{p-1}^{-1}$ , where  $\Delta_{p-1} = R(p) - \rho_{p-1}^{*\#} R_{A,p-1}^{-1} \rho_{p-1}^\#$ , like  $\lambda_{p-1}$ , is in the form of a Schur complement.

- To interpret  $\Delta_{p-1}$  and  $k_p$ , we parallel the development for  $\lambda_{p-1}$  (19):

$$\begin{aligned}
\Delta_{p-1} &= R(p) - \rho_{p-1} R_{A,p-1}^{-1} \rho_{p-1}^{*\#} \\
&= R(p) + A_{1,p-1} R(p-1) + \dots + A_{p-1,p-1} R(1) \\
&= \sum y_n \left[ y_{n-p} + A_{1,p-1} y_{n-(p-1)} + \dots + A_{p-1,p-1} y_{n-1} \right] \tag{22}
\end{aligned}$$

The quantity

$y_{n-p} + A_{1,p-1} y_{n-(p-1)} + \dots + A_{p-1,p-1} y_{n-1}$  is interpreted as a *backwards* prediction error, and we denote as  $f_{n,p-1}$ . Proceeding:

$$\begin{aligned}
\Delta_{p-1} &= \sum y_n f_{n,p-1}^* \\
&= \sum e_{n,p-1} f_{n,p-1}^* \tag{23}
\end{aligned}$$

Again by orthogonality arguments, (see Appendix), we replace  $y_n$  by  $e_{n,p-1}$ . Note that  $\Delta_p/N$  gives a (biased) estimate of the cross-covariance between forward and backward prediction errors of order  $p$ .

- The recursion for  $A_p$  depends on the unknown quantities  $\lambda_{p-1}$  and  $\Delta_{p-1}$ . Because  $\lambda_{p-1} = R(0) - A_{p-1}\rho_{p-1}^*$  and  $\Delta_{p-1} = R(p) - A_{p-1}\rho_{p-1}^{\#\#}$ , we see everything is available for the computation. However, forming  $A_{p-1}\rho_{p-1}^*$  takes  $p - 1$   $m$ -by- $m$  matrix multiplications. To eliminate all but one of these multiplications, we develop a recursion for  $\lambda_p$ :

$$\begin{aligned}
 \lambda_p &= R(0) - A_p \rho_p^* \\
 &= R(0) - \left[ A_{p-1} - k_p A_{p-1}^{\#\#} \mid k_p \right] \begin{bmatrix} \rho_{p-1} \\ R(p) \end{bmatrix} \\
 &= (R(0) - A_{p-1} \rho_{p-1}^*) - k_p (R(0) - A_{p-1} \rho_{p-1}^*) \\
 &= \lambda_{p-1} - k_p \Delta_{p-1}^* \tag{24}
 \end{aligned}$$

This gives the standard form of the Levinson-Durbin algorithm.

- For the initialization, consider  $p = 0$ . Since no prediction is done, the forward error is nothing but the observation:  $e_{n,0} = y_n$  and the backward error is nothing but the past observation:  $f_{n,0} = y_{n-1}$ , given

the way it was defined (23). It follows that  $\lambda_0 = R(0)$  and  $\Delta_0 = R(1)$ ; consequently,  $k_1 = R(1) [R(0)]^{-1}$ , which equals  $A_1$  thanks to the original least squares solution (12). The Levinson-Durbin algorithm may be summarized:

**Recursion:**

$$\begin{aligned}\lambda_p &= \lambda_{p-1} - k_p \Delta_{p-1}^* \\ \Delta_p &= R(p) - A_{p-1} \rho_{p-1}^{*\#} \\ k_{p+1} &= \Delta_p \lambda_p^{-1} \\ A_{p+1} &= [ A_p - k_{p+1} A_p^\# \mid k_{p+1} ]\end{aligned}$$

**Initialization:**

$$\begin{aligned}\lambda_0 &= R(0) \\ \Delta_0 &= R(1) \\ k_1 &= \Delta_0 \lambda_0^{-1} \\ A_1 &= k_1\end{aligned}$$