EFFICIENT SOLUTION OF COVARIANCE EQUATIONS FOR PITCH-SYNCHRONOUS LINEAR PREDICTION

James Moorer

2021

ABSTRACT:

Morf, *et al* [**MDKV77**] described an efficient solution of covariance equations for a subset of a sequence of sampled-data input points from time-index S to time-index T. In this paper, we give an algorithm to "walk" the time-interval forward by two independent recursions – one to advance T by one, and another to advance S by one. Since these two recursions are independent and can be applied separately, in any order, and possibly multiple times in a row, this constitutes a complete iteration for a window that can change not only its location in time with every step but also its length. This makes it possible to make the process pitch-synchronous as long as the period is known and does not change too quickly.

INTRODUCTION:

One of the problems with using linear prediction for time-base correction and pitch shifting of speech is the interaction of the analysis window with the pitch period of voiced speech. For example, the RLSN method **[LMF1981]** uses an exponential window which will not go to zero over a single pitch period. The resulting analysis at high orders when used for synthesis will exhibit audible characteristics of the original pitch period, regardless of the pitch of the driving function used for the synthesis. These artifacts can include some signal at the original pitch, and sometimes a "blip" as the original glottal pulse enters the analysis window. One way of eliminating these artifacts is to perform a pitch-synchronous analysis using only the number of points in one period of the original speech.

FORMULATION OF THE PROBLEM:

In this paper, we will restrict the domain to *real, scaler* signals. This formulation can be extended to complex, matrix formulations, but then care must be taken with a number of operations, such as order in matrix multiplication and what kind of transpose and what kind of inverse is used. I leave these extensions to the ambitious reader.

We start with a sequence of real, scaler input points. We will look to begin with just at samples from time (sample number) *S* to time *T*: $[y_S, y_{S+1}, \dots, y_{T-1,} y_T]$. We arrange these in rows of $p+1$ samples at a time, with each row offset from the previous row by one sample. Each sample will show up in the matrix between 1 and $p+1$ times.

$$(1)\ \ \boldsymbol{Y}_{p,S,T} = \begin{bmatrix} y_{S+p} & \cdots & y_S \\ \vdots & \ddots & \vdots \\ y_T & \cdots & y_{T-p} \end{bmatrix}$$

This matrix has $(T\text{-}S\text{-}p+1)$ rows and $(p+1)$ columns. We now define the covariance matrix:

$$(2)\ \ \boldsymbol{R}_{p,S,T} = \boldsymbol{Y}'_{p,S,T}\boldsymbol{Y}_{p,S,T} = \left(\sigma_{p,S,T}(i,j)\right) = \left(\sum_{t=S+p}^{T} y_{t-i}y_{t-j}\right),\ \ 0 \le \mathrm{i} \le \mathrm{p}, 0 \le \mathrm{j} \le \mathrm{p}$$

$\boldsymbol{R}_{p,S,T}$ is a symmetric, square matrix, not Töplitz but product of two Töplitz matrices, and is of dimension (p+1)x(p+1). Note that the inverse of a real, symmetric matrix is also symmetric.

There is some aliasing/redundancy inherent in this definition. For instance:

$$\sigma_{p,S,T}(1,1) = \sigma_{p-1,S,T-1}(0,0) = y_{S+p-1}^2 + \cdots + y_{T-1}^2$$

This aliasing leads to the following basic identities:

$$(3)\ \ \boldsymbol{R}_{p,S,T} = \begin{pmatrix} \sigma_{p,S,T}(0,0) & \sigma_{p,S,T}(0,1) & \cdots & \sigma_{p,S,T}(0,p) \\ \sigma_{p,S,T}(1,0) & & & \\ \vdots & & \boldsymbol{R}_{p-1,S,T-1} & \\ \sigma_{p,S,T}(p,0) & & & \end{pmatrix}$$

$$(4)\ \ \boldsymbol{R}_{p,S,T} = \begin{pmatrix} & & & \sigma_{p,S,T}(0,p) \\ & \boldsymbol{R}_{p-1,S+1,T} & & \vdots \\ & & & \sigma_{p,S,T}(p-1,p) \\ \sigma_{p,S,T}(p,0) & \cdots & \sigma_{p,S,T}(p,p-1) & \sigma_{p,S,T}(p,p) \end{pmatrix}$$

These are combined order and time updates. Note that number of terms in elements of submatrix are the same as any other element, which is *(T-S-p+1)* terms. The submatrices are of dimension pxp. The matrices $\boldsymbol{R}_{x,S,T-1}$ and $\boldsymbol{R}_{x,S+1,T}$ will be termed "time-shifted" quantities. That is, any quantity with one less time point at the beginning or end of the range will be termed a "time-shifted" quantity.

$$(5)\ \ \boldsymbol{R}_{p,S,T} = \boldsymbol{R}_{p,S,T-1} + [y_T, y_{T-1}, \cdots, y_{T-p}]'[y_T, y_{T-1}, \cdots, y_{T-p}]$$

$$(6)\ \ \boldsymbol{R}_{p,S,T} = \boldsymbol{R}_{p,S+1,T} + [y_{S+p}, y_{S+p-1}, \cdots, y_S]'[y_{S+p}, y_{S+p-1}, \cdots, y_S]$$

These last two are rank-1 (outer-product) updates to either the beginning or the ending of the time interval. These identities differ in that (3) and (4) involve both unit order and time changes whereas (5) and (6) involve only unit (single-point) changes to the time interval.

As best I can tell, these invariants were first noticed by Morf [***ref].

In general, the following equations show calculations that do not require explicit knowledge of all the elements of $\boldsymbol{R}_{p,S,T}$ but only of the last (or first) column of $\boldsymbol{R}_{p,S,T}$**.** Appendix A shows a simplified way to calculate the last column of $\boldsymbol{R}_{p,S,T}$ in a recursive manner.

Define column vectors of length *(p+1)* $\boldsymbol{A}_{p,S,T}$, $\boldsymbol{B}_{p,S,T}$, $\boldsymbol{C}_{p,S,T}$, $\boldsymbol{D}_{p,S,T}$.

Furthermore, define $\boldsymbol{A}_{p,S,T}(0) \equiv 1$ and $\boldsymbol{B}_{p,S,T}(p) \equiv 1$.

Define the relation of the vectors to the data y_t as follows:

$$(7)\ \boldsymbol{R}_{p,S,T}\,[\boldsymbol{A}_{p,S,T}\quad \boldsymbol{B}_{p,S,T}\quad \boldsymbol{C}_{p,S,T}\quad \boldsymbol{D}_{p,S,T}] = \begin{bmatrix} R^e_{p,S,T} & 0 & y_T & y_{S+p} \\ 0 & 0 & y_{T-1} & y_{S+p-1} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & y_{T-p+1} & y_{S+1} \\ 0 & R^r_{p,S,T} & y_{T-p} & y_S \end{bmatrix}$$

$R^e_{p,S,T}$ and $R^r_{p,S,T}$ are scalar "forward" and "backward" residuals. Vector $\boldsymbol{A}_{p,S,T}$ is sometimes called the "forward" predictor, and $\boldsymbol{B}_{p,S,T}$ the "backward" predictor. Vectors $\boldsymbol{C}_{p,S,T}$ and $\boldsymbol{D}_{p,S,T}$ have something to do with the beginning and ending of the time span of the analysis window. The first two columns comprise the "normal equations".

Part of the calculation is to solve the normal equations – *i.e.*, find $\boldsymbol{A}_{p,S,T}$ and $\boldsymbol{B}_{p,S,T}$ that projects the covariance matrix onto a vector with a single non-zero element (first or last). If $\boldsymbol{R}_{p,S,T}$ is not singular, this can always be done. *In this paper, we assume that* $\boldsymbol{R}_{p,S,T}$ *is always full-rank.*

One might ask why we need the ***C*** and ***D*** matrices. The point is that (a) it gives us two more simultaneous constraints that help obtain a solution, (b) they say something about the boundary conditions of the data, and (c) they are a compact way to express the vectors involved in the outer products of equations (5) and (6). In one way of looking at it, they give another view into the structure of the covariance matrix. Yes, the definitions are somewhat contrived, but they will prove a useful shorthand for the recursions we seek.

With the above definitions, without explicitly inverting $\boldsymbol{R}_{p,S,T}$, we can say a few things about the inverse, $\boldsymbol{R}^{-1}_{p,S,T}$. The first row of $\boldsymbol{R}^{-1}_{p,S,T}$ is $\boldsymbol{A}'_{p,S,T}(R^e_{p,S,T})^{-1}$ and the last row of $\boldsymbol{R}^{-1}_{p,S,T}$ is $\boldsymbol{B}'_{p,S,T}(R^r_{p,S,T})^{-1}$. Note that this is a statement about how ***A*** and ***B*** are calculated rather than anything intrinsic to the inverse covariance matrix.

We define a set of scalars as follows:

$$(8)\ \begin{bmatrix} e_{p,S,T} \\ r_{p,S,T} \\ g_{p,S,T} \\ h_{p,S,T} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}'_{p,S,T} \\ \boldsymbol{B}'_{p,S,T} \\ \boldsymbol{C}'_{p,S,T} \\ \boldsymbol{D}'_{p,S,T} \end{bmatrix} \begin{bmatrix} y_T \\ y_{T-1} \\ \vdots \\ y_{T-p+1} \\ y_{T-p} \end{bmatrix}$$

And one more as follows:

$$(9)\ \begin{bmatrix} h_{p,S,T} \\ f_{p,S,T} \end{bmatrix} = \begin{bmatrix} \boldsymbol{C}'_{p,S,T} \\ \boldsymbol{D}'_{p,S,T} \end{bmatrix} \begin{bmatrix} y_{S+p} \\ y_{S+p-1} \\ \vdots \\ y_{S+1} \\ y_S \end{bmatrix}$$

Note – these are just dot products between the column vectors (***A***, ***B***, ***C***, or ***D***) and a vector of *p+1* input points either at the beginning ([S:S+p]) or end ([T-p:T]) of the sequence of input points.

It is useful to make a more general definition of time-variant residuals:

(10) $$\begin{bmatrix} e_{p,S,T}(n) \\ r_{p,S,T}(n) \\ g_{p,S,T}(n) \\ f_{p,S,T}(n) \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}'_{p,S,T} \\ \boldsymbol{B}'_{p,S,T} \\ \boldsymbol{C}'_{p,S,T} \\ \boldsymbol{D}'_{p,S,T} \end{bmatrix} \begin{bmatrix} y_n \\ y_{n-1} \\ \vdots \\ y_{n-p+1} \\ y_{n-p} \end{bmatrix}$$

We might ask if there is a time-variant definition $h_{p,S,T}(n)$. There is not, since $h_{p,S,T}(n)$ is only defined for $n=T$ (equation (8)) and $n=S+p$ (equation (9)).

Using the above designations and our knowledge of the first (last) rows of $\boldsymbol{R}_{p,S,T}^{-1}$, we may calculate the first and last elements of $\boldsymbol{C}$ and $\boldsymbol{D}$ as follows:

(11) $$\boldsymbol{C}_{p,S,T} = \begin{bmatrix} e_{p,S,T}(T)\left(R_{p,S,T}^{e}\right)^{-1} \\ \vdots \\ r_{p,S,T}(T)\left(R_{p,S,T}^{r}\right)^{-1} \end{bmatrix} = \begin{bmatrix} e_{p,S,T}\left(R_{p,S,T}^{e}\right)^{-1} \\ \vdots \\ r_{p,S,T}\left(R_{p,S,T}^{r}\right)^{-1} \end{bmatrix}$$

(12) $$\boldsymbol{D}_{p,S,T} = \begin{bmatrix} e_{p,S,T}(S+p)\left(R_{p,S,T}^{e}\right)^{-1} \\ \vdots \\ r_{p,S,T}(S+p)\left(R_{p,S,T}^{r}\right)^{-1} \end{bmatrix}$$

STATEMENT OF THE PROBLEM

Morf, *et al* [**MDKV77**] showed a method of calculating $\boldsymbol{A}_{p,S,T}$, $\boldsymbol{B}_{p,S,T}$, $\boldsymbol{C}_{p,S,T}$, and $\boldsymbol{D}_{p,S,T}$. We will assume that this calculation has been done already for one combination of p, S, and T. In fact, we require that the iteration be run to one higher order to produce $\boldsymbol{A}_{p+1,S,T}$, $\boldsymbol{B}_{p+1,S,T}$, $\boldsymbol{C}_{p+1,S,T}$, and $\boldsymbol{D}_{p+1,S,T}$. We also assume that all the intermediate data from the calculation has been preserved, so we have $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, and $\boldsymbol{D}$ for all (lower) values of p, and certain time-shifted values as well. Although the values of the scalars can easily be produced by dot products of the vectors and selected input points, it is reasonable to carry along previous values of the scalars as well.

We will produce a set of equations for calculating the vectors for the time intervals [S:T+1] and [S+1:T]. This gives us a way to independently advance the start time or the end time of the analysis window. This is sufficient to match the window length to an approximation of the pitch period at each point in time.

CONSEQUENCES OF IDENTITIES (5) AND (6)

We can immediately write the consequences of multiplying the vectors $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ and $\boldsymbol{D}$ by a covariance matrix with either the first or last data point dropped:

(13) $$R_{p,S+1,T}[\boldsymbol{A}_{p,S,T} \quad \boldsymbol{B}_{p,S,T}] = \begin{bmatrix} R_{p,S,T}^{e} - y_{S+p}e_{p,S,T}(S+p) & -y_{S+p}r_{p,S,T}(S+p) \\ -y_{S+p-1}e_{p,S,T}(S+p) & -y_{S+p-1}r_{p,S,T}(S+p) \\ \vdots & \vdots \\ -y_{S}e_{p,S,T}(S+p) & R_{p,S,T}^{r} - y_{S}r_{p,S,T}(S+p) \end{bmatrix}$$

$$(14)\qquad R_{p,S+1,T}[\boldsymbol{C}_{p,S,T}\quad \boldsymbol{D}_{p,S,T}] = \begin{bmatrix} y_T - y_{S+p}h_{p,S,T} & y_{S+p}(1-f_{p,S,T}) \\ y_{T-1} - y_{S+p-1}h_{p,S,T} & y_{S+p-1}(1-f_{p,S,T}) \\ \vdots & \vdots \\ y_{T-p} - y_S h_{p,S,T} & y_S(1-f_{p,S,T}) \end{bmatrix}$$

$$(15)\qquad R_{p,S,T-1}[\boldsymbol{A}_{p,S,T}\quad \boldsymbol{B}_{p,S,T}] = \begin{bmatrix} R^e_{p,S,T} - y_T e_{p,S,T} & -y_T r_{p,S,T} \\ -y_{T-1}e_{p,S,T} & -y_{T-1}r_{p,S,T} \\ \vdots & \vdots \\ -y_{T-p}e_{p,S,T} & R^r_{p,S,T} - y_{T-p}r_{p,S,T} \end{bmatrix}$$

$$(16)\qquad R_{p,S,T-1}[\boldsymbol{C}_{p,S,T}\quad \boldsymbol{D}_{p,S,T}] = \begin{bmatrix} y_T(1-g_{p,S,T}) & y_T - y_{S+p}h_{p,S,T} \\ y_{T-1}(1-g_{p,S,T}) & y_{T-1} - y_{S+p-1}h_{p,S,T} \\ \vdots & \vdots \\ y_{T-p}(1-g_{p,S,T}) & y_{T-p} - y_S h_{p,S,T} \end{bmatrix}$$

Note that the RHS of these identities involve vectors of the input data $y[T-p{:}T]$ and $y[S{:}S+p]$, which happen to correspond exactly to columns 3 and 4 of the RHS of equation (7), thus revealing part of the motivation for our use of the vectors $\boldsymbol{C}$ and $\boldsymbol{D}$.

We can use these to generate time-shifted versions of the vectors with the following general formulation:

$$(17)\qquad \boldsymbol{X}_\omega = \alpha\boldsymbol{X}_\mu + \beta\boldsymbol{Y}_\tau$$

Where $\boldsymbol{X}$ and $\boldsymbol{Y}$ are one of $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, or $\boldsymbol{D}$. ω, μ, and τ are order/range specifiers. α and β are factors to be determined. Sometimes there is a mismatch of order. This is handled by augmenting $\boldsymbol{Y}_\tau$ with a zero element in either the first or last position. If we guess the correct order/range specifiers, then often α and β can be computed directly from the givens (normal equations and identities involving the covariance matrix). For instance, let us compute some time-shifted quantities. We start by "guessing" that the use of the vector $\boldsymbol{D}$ would be useful for shifting the beginning point of the interval:

$$(18)\qquad \boldsymbol{A}_{p,S+1,T} = \alpha\boldsymbol{A}_{p,S,T} + \beta\boldsymbol{D}_{p,S,T}$$

This "guess" was produced by taking the values we have (presumably) already calculated and compute one we would like to have. We can proceed by multiplying by $\boldsymbol{R}_{p,S+1,T}$, to wit:

$$(19)\qquad \boldsymbol{R}_{p,S+1,T}\boldsymbol{A}_{p,S+1,T} = \alpha\boldsymbol{R}_{p,S+1,T}\boldsymbol{A}_{p,S,T} + \beta\boldsymbol{R}_{p,S+1,T}\boldsymbol{D}_{p,S,T}$$

$$(20)\qquad \begin{bmatrix} R^e_{p,S+1,T} \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} = \alpha\begin{bmatrix} R^e_{p,S,T} - y_{S+p}e_{p,S,T}(S+p) \\ -y_{S+p-1}e_{p,S,T}(S+p) \\ \vdots \\ -y_{S+p}e_{p,S,T}(S+p) \end{bmatrix} + \beta\begin{bmatrix} y_{S+p}(1-f_{p,S,T}) \\ y_{S+p-1}(1-f_{p,S,T}) \\ \vdots \\ y_S(1-f_{p,S,T}) \end{bmatrix}$$

We can choose $\alpha = R^e_{p,S+1,T}/R^e_{p,S,T}$ and $\beta = \alpha\, e_{p,S,T}(S+p)/(1-f_{p,S,T}(S+p))$ to make this an identity. We can do exactly the same thing a few more times to get the following time-shifted values of ***A*** and ***B***:

$$\boldsymbol{A}_{p,S+1,T} = \frac{R^e_{p,S+1,T}}{R^e_{p,S,T}}\left(\boldsymbol{A}_{p,S,T} + \frac{e_{p,S,T}(S+p)}{(1-f_{p,S,T})}\boldsymbol{D}_{p,S,T}\right) \tag{21}$$

$$\boldsymbol{B}_{p,S+1,T} = \frac{R^r_{p,S+1,T}}{R^r_{p,S,T}}\left(\boldsymbol{B}_{p,S,T} + \frac{r_{p,S,T}(S+p)}{(1-f_{p,S,T})}\boldsymbol{D}_{p,S,T}\right) \tag{22}$$

$$\boldsymbol{A}_{p,S,T-1} = \frac{R^e_{p,S,T-1}}{R^e_{p,S,T}}\left(\boldsymbol{A}_{p,S,T} + \frac{e_{p,S,T}}{(1-g_{p,S,T})}\boldsymbol{C}_{p,S,T}\right) \tag{23}$$

$$\boldsymbol{B}_{p,S,T-1} = \frac{R^r_{p,S,T-1}}{R^r_{p,S,T}}\left(\boldsymbol{B}_{p,S,T} + \frac{r_{p,S,T}}{(1-g_{p,S,T})}\boldsymbol{C}_{p,S,T}\right) \tag{24}$$

Note that this appears to require knowledge of the time-shifted versions of $R^e_{p,S,T}$ and $R^r_{p,S,T}$. These are easily calculated from the constraint that the first term of ***A*** and the last term of ***B*** must always both be exactly one. That is, after the term in the outer parentheses is calculated, it can be divided by the first (last) element in the vector to normalize the first (last) term to 1.0. We have implicitly calculated the value of α this way. Note that this observation makes these formulations somewhat simpler than equivalent ones in the literature, at the expense of not having a closed-form expression for the time-shifted quantities but rather an algorithm for finding the scaling. Additionally there are formulas that I haven't seen in the literature.

This was so much fun that we might now try to do the same thing with ***C*** and ***D***:

$$\boldsymbol{C}_{p,S+1,T} = \alpha\boldsymbol{C}_{p,S,T} + \beta\boldsymbol{D}_{p,S,T} \tag{25}$$

$$\boldsymbol{R}_{p,S+1,T}\boldsymbol{C}_{p,S+1,T} = \alpha\boldsymbol{R}_{p,S+1,T}\boldsymbol{C}_{p,S,T} + \beta\boldsymbol{R}_{p,S+1,T}\boldsymbol{D}_{p,S,T} \tag{26}$$

$$\begin{bmatrix} y_T \\ y_{T-1} \\ \vdots \\ y_{T-p+1} \\ y_{T-p} \end{bmatrix} = \alpha\begin{bmatrix} y_T - y_{S+p}h_{p,S,T} \\ y_{T-1} - y_{S+p-1}h_{p,S,T} \\ \vdots \\ y_{T-p} - y_S h_{p,S,T} \end{bmatrix} + \beta\begin{bmatrix} y_{S+p}(1-f_{p,S,T}) \\ y_{S+p-1}(1-f_{p,S,T}) \\ \vdots \\ y_S(1-f_{p,S,T}) \end{bmatrix} \tag{27}$$

Clearly values of $\alpha = 1$ and $\beta = h_{p,S,T}/(1-f_{p,S,T}(S+p))$ gives an identity. The following equation follows:

$$\boldsymbol{C}_{p,S+1,T} = \boldsymbol{C}_{p,S,T} + \frac{h_{p,S,T}}{(1-f_{p,S,T})}\boldsymbol{D}_{p,S,T} \tag{28}$$

Note that these formulas are not unique. There can be more than one way to calculate time-shifted quantities. By analogy, we can write the update for ***D*** as follows:

$$\boldsymbol{D}_{p,S,T-1} = \frac{h_{p,S,T}}{(1-g_{p,S,T}(S+p))}\boldsymbol{C}_{p,S,T} + \boldsymbol{D}_{p,S,T} \tag{29}$$

We get into trouble trying to do $\boldsymbol{C}_{p,S,T-1}$ and $\boldsymbol{D}_{p,S+1,T}$ this way. One approach is to assume that we have all the vectors and scalers at coordinates $(p+1,S,T)$. From these we can compute $\boldsymbol{D}_{p,S+1,T}$ as follows:

(30) $$\boldsymbol{R}_{p+1,S,T}\,\boldsymbol{D}_{p+1,S,T} = \begin{bmatrix} y_{S+p+1} \\ y_{S+p} \\ \vdots \\ y_{S+1} \\ y_S \end{bmatrix} = \boldsymbol{R}_{p+1,S,T}\left\{\begin{bmatrix} \boldsymbol{D}_{p,S+1,T} \\ 0 \end{bmatrix} + \omega \boldsymbol{B}_{p+1,S,T}\right\}$$

Given that we know the value of the last element of $\boldsymbol{D}_{p+1,S,T}$, we can write the update rule as follows:

(31) $$\boldsymbol{D}_{p+1,S,T} = \begin{bmatrix} \boldsymbol{D}_{p,S+1,T} \\ 0 \end{bmatrix} + r_{p+1,S,T}\left(R^r_{p+1,S,T}\right)^{-1} \boldsymbol{B}_{p+1,S,T}$$

Rearrange to solve for the quantity we want:

(32) $$\begin{bmatrix} \boldsymbol{D}_{p,S+1,T} \\ 0 \end{bmatrix} = \boldsymbol{D}_{p+1,S,T} - r_{p+1,S,T}\left(R^r_{p+1,S,T}\right)^{-1} \boldsymbol{B}_{p+1,S,T}$$

Similar reasoning yields the following identity:

(33) $$\begin{bmatrix} 0 \\ \boldsymbol{C}_{p,S,T-1} \end{bmatrix} = \boldsymbol{C}_{p+1,S,T} - e_{p+1,S,T}\left(R^e_{p+1,S,T}\right)^{-1} \boldsymbol{A}_{p+1,S,T}$$

Note that these last two are a bit unconventional: they make use of the vectors of a *higher* order. I have not seen these relations in the literature. There is no reason that previous calculations of a *higher* order shouldn't be used. It is easy enough to execute the order-increasing iteration one extra time to produce this value.

Note also that we do not have to calculate the coefficients of $\boldsymbol{A}$ *and* $\boldsymbol{B}$ *in (32) and (33). It is sufficient to just scale* $\boldsymbol{A}$ *and* $\boldsymbol{B}$ *to exactly cancel the last (first) term of* $\boldsymbol{D}_{p+1,S,T}$ ($\boldsymbol{C}_{p+1,S,T}$).

CONSEQUENCES OF IDENTITIES (3) AND (4)

Identities (5) and (6) are "order preserving". That is, it involves combinations of vectors of the same order (length). Identities (3) and (4) allow for increasing or decreasing the order by one. One straightforward way to do this is augment the vector by annexing a zero term, producing a vector with *p+1* elements as follows:

(34) $$\boldsymbol{X}_\omega = \alpha \boldsymbol{X}_\mu + \beta \begin{bmatrix} 0 \\ \boldsymbol{Y}_\tau \end{bmatrix}$$

(35) $$\boldsymbol{X}_\omega = \alpha \boldsymbol{X}_\mu + \beta \begin{bmatrix} \boldsymbol{Y}_\tau \\ 0 \end{bmatrix}$$

(36) $$\boldsymbol{X}_\omega = \alpha \begin{bmatrix} 0 \\ \boldsymbol{X}_\mu \end{bmatrix} + \beta \begin{bmatrix} \boldsymbol{Y}_\tau \\ 0 \end{bmatrix}$$

For instance, $\boldsymbol{R}_{p,S,T}$ times $\begin{bmatrix} 0 \\ \boldsymbol{C}_{p-1,S,T-1} \end{bmatrix}$ results in the following:

$$(37) \qquad \boldsymbol{R}_{p,S,T} \begin{bmatrix} 0 \\ \boldsymbol{C}_{p-1,S,T-1} \end{bmatrix} = \begin{bmatrix} \xi \\ y_{T-1} \\ \vdots \\ y_{T-p+1} \\ y_{T-p} \end{bmatrix}$$

Where ξ is an unknown quantity at this point, but can often be determined by exploiting other symmetries and known relations. One common way would be the following simple decomposition:

$$(38) \qquad \boldsymbol{R}_{p,S,T} \begin{bmatrix} 0 \\ \boldsymbol{C}_{p-1,S,T-1} \end{bmatrix} = \begin{bmatrix} y_T \\ y_{T-1} \\ \vdots \\ y_{T-p+1} \\ y_{T-p} \end{bmatrix} + \begin{bmatrix} \xi - y_T \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} = \boldsymbol{R}_{p,S,T}\boldsymbol{C}_{p,S,T} + \begin{bmatrix} \xi - y_T \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

We know that the first row of $\boldsymbol{R}_{p,S,T}^{-1}$ is $\boldsymbol{A}'_{p,S,T}(R^e_{p,S,T})^{-1}$. We also know that the first element of $\boldsymbol{C}_{p,S,T}$ is $e_{p,S,T}\left(R^e_{p,S,T}\right)^{-1}$**.** We can now write a relation between $\boldsymbol{C}_{p,S,T}$ and $\boldsymbol{C}_{p-1,S,T-1}$ as follows:

$$(39) \qquad \boldsymbol{C}_{p,S,T} = \begin{bmatrix} 0 \\ \boldsymbol{C}_{p-1,S,T-1} \end{bmatrix} + e_{p,S,T}\left(R^e_{p,S,T}\right)^{-1}\boldsymbol{A}_{p,S,T}$$

Note that this solution is not unique. Here is a different, equivalent definition:

$$(40) \qquad \boldsymbol{C}_{p,S,T} = \begin{bmatrix} \boldsymbol{C}_{p-1,S+1,T} \\ 0 \end{bmatrix} + r_{p,S,T}\left(R^r_{p,S,T}\right)^{-1}\boldsymbol{B}_{p,S,T}$$

This kind of derivation finesses the direct calculation of ξ by knowledge of what the first element of $\boldsymbol{C}_{p,S,T}$ must be. Similar arguments can be used to derive analogous recursions for $\boldsymbol{D}_{p,S,T}$.

$$(41) \qquad \boldsymbol{D}_{p,S,T} = \begin{bmatrix} 0 \\ \boldsymbol{D}_{p-1,S,T-1} \end{bmatrix} + e_{p,S,T}(S+p)\left(R^e_{p,S,T}\right)^{-1}\boldsymbol{A}_{p,S,T}$$

$$(42) \qquad \boldsymbol{D}_{p,S,T} = \begin{bmatrix} \boldsymbol{D}_{p-1,S+1,T} \\ 0 \end{bmatrix} + r_{p,S,T}(S+p)\left(R^r_{p,S,T}\right)^{-1}\boldsymbol{B}_{p,S,T}$$

These are relations for increasing (or decreasing) the order of ***C*** or ***D*** through the use of time-shifted versions of these vectors of order $p-1$.

Relation (36) gives us a way of advancing the order of ***A*** and ***B***.

$$(43) \qquad \boldsymbol{A}_{p,S,T} = \alpha \begin{bmatrix} 0 \\ \boldsymbol{B}_{p-1,S,T-1} \end{bmatrix} + \beta \begin{bmatrix} \boldsymbol{A}_{p-1,S+1,T} \\ 0 \end{bmatrix}$$

There is some leap of faith here about where this form comes from. It is plausible because it is similar to the Levinson recursion **[LEVINSON1947]** and to Szegö polynomials **[SZEGO1939]** The changes in the time subscripts are plausible because relations (3) and (4) involve those same time spans. One would imagine that applying (3) and (4) would force the vectors to have the same subscripts as the submatrices of the RHS of (3) and (4). There is an analogous equation for $\boldsymbol{B}_{p,S,T}$. If we first assume that $\beta = 1$, again by analogy to the Levinson *et al* recursion, then we get two simultaneous equations in two unknowns:

(44) $$\boldsymbol{A}_{p,S,T} = \begin{bmatrix} \boldsymbol{A}_{p-1,S+1,T} \\ 0 \end{bmatrix} + \kappa_r \begin{bmatrix} 0 \\ \boldsymbol{B}_{p-1,S,T-1} \end{bmatrix}$$

(45) $$\boldsymbol{B}_{p,S,T} = \begin{bmatrix} 0 \\ \boldsymbol{B}_{p,-1S,T-1} \end{bmatrix} + \kappa_e \begin{bmatrix} \boldsymbol{A}_{p-1,S+1,T} \\ 0 \end{bmatrix}$$

We multiply each of these equations by $\boldsymbol{R}_{p,S,T}$. This will annihilate everything except the first and last terms of the resulting vectors. Of those the first (last) can be set equal to 1. The last term of the result from (44) and the first term of the result from (45) must be equal by symmetry. This is "Burg's Lemma" **[BURG1975].**

(46) $$\begin{bmatrix} R^e_{p,S,T} \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} R^e_{p-1,S+1,T} \\ 0 \\ \vdots \\ 0 \\ \Delta_{p,S,T} \end{bmatrix} + \kappa_r \begin{bmatrix} \Delta_{p,S,T} \\ 0 \\ \vdots \\ 0 \\ R^r_{p-1,S,T-1} \end{bmatrix}$$

(47) $$\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ R^r_{p,S,T} \end{bmatrix} = \begin{bmatrix} \Delta_{p,S,T} \\ 0 \\ \vdots \\ 0 \\ R^r_{p-1,S,T-1} \end{bmatrix} + \kappa_e \begin{bmatrix} R^e_{p-1,S+1,T} \\ 0 \\ \vdots \\ 0 \\ \Delta_{p,S,T} \end{bmatrix}$$

Note that $\Delta_{p,S,T}$ is a *calculated* quantity. It is the dot product of the last row of $\boldsymbol{R}_{p,S,T}$ with $\begin{bmatrix} \boldsymbol{A}_{p-1,S+1,T} \\ 0 \end{bmatrix}$. Or, equivalently, the product of the first row of $\boldsymbol{R}_{p,S,T}$ with $\begin{bmatrix} 0 \\ \boldsymbol{B}_{p-1,S,T-1} \end{bmatrix}$. Since $\Delta_{p,S,T}$ is easily calculated, we can write the multipliers down directly:

(48) $$\kappa_r = -\Delta_{p,S,T}/R^r_{p-1,S,T-1}$$
(49) $$\kappa_e = -\Delta_{p,S,T}/R^e_{p-1,S+1,T}$$

Technically, these multipliers should have a subscript of *p,S,T* as well.

Once ***A*** and ***B*** have been calculated, updated versions of ***C*** and ***D*** are easily available from (39) and (41) above. The updates for the error terms are easily extracted from (46) and (47) above:

(50) $$R^e_{p,S,T} = R^e_{p-1,S+1,T} + \kappa_r \Delta_{p,S,T}$$
(51) $$R^r_{p,S,T} = R^r_{p-1,S,T-1} + \kappa_e \Delta_{p,S,T}$$

Note that the literature does not entirely agree on the sign of κ_r and κ_e. Ultimately it does not matter, but care must be taken to be internally consistent with whatever convention is adopted.

THE TIME RECURSIONS

For pitch-synchronous, we will have to have a set of recursions to move the vectors from time T to time $T+1$, as well as a separate and independent recursion to move the starting point from time S to time $S+1$. Since we know that $\boldsymbol{C}$ has something to do with the end of the time interval, and $\boldsymbol{D}$ has something to do with the beginning of the time interval, we can hypothesize the form of the updates as follows:

(52) $$\boldsymbol{A}_{p,S,T+1} = \boldsymbol{A}_{\mu} + \beta \begin{bmatrix} 0 \\ \boldsymbol{C}_{\tau} \end{bmatrix}$$

(53) $$\boldsymbol{B}_{p,S,T+1} = \boldsymbol{B}_{\eta} + \gamma \begin{bmatrix} \boldsymbol{C}_{\upsilon} \\ 0 \end{bmatrix}$$

(54) $$\boldsymbol{A}_{p,S+1,T} = \boldsymbol{A}_{\mu} + \delta \begin{bmatrix} 0 \\ \boldsymbol{D}_{\tau} \end{bmatrix}$$

(55) $$\boldsymbol{B}_{p,S+1,T} = \boldsymbol{B}_{\eta} + \zeta \begin{bmatrix} \boldsymbol{D}_{\upsilon} \\ 0 \end{bmatrix}$$

Let us make some reasonable guesses as to what the order/range specifiers should be, as follows:

(56) $$\boldsymbol{A}_{p,S,T+1} = \boldsymbol{A}_{p,S,T} + \beta \begin{bmatrix} 0 \\ \boldsymbol{C}_{p-1,S,T} \end{bmatrix}$$

(57) $$\boldsymbol{B}_{p,S,T+1} = \boldsymbol{B}_{p,S,T} + \gamma \begin{bmatrix} \boldsymbol{C}_{p-1,S+1,T+1} \\ 0 \end{bmatrix}$$

(58) $$\boldsymbol{A}_{p,S+1,T} = \boldsymbol{A}_{p,S,T} + \delta \begin{bmatrix} 0 \\ \boldsymbol{D}_{p-1,S,T} \end{bmatrix}$$

(59) $$\boldsymbol{B}_{p,S+1,T} = \boldsymbol{B}_{p,S,T} + \zeta \begin{bmatrix} \boldsymbol{D}_{p-1,S+1,T} \\ 0 \end{bmatrix}$$

These are reasonable choices since the first term relates to equations (5) and (6), and the second term relates to equations (3) and (4). Now, can we find the appropriate multipliers? We can proceed by multiplying (56) and (57) by $\boldsymbol{R}_{p,S,T+1}$ .We can expand $\boldsymbol{R}_{p,S,T+1}$ in the first terms using (5) and the second term by (3) to produce the following:

(60) $$\begin{bmatrix} R^e_{p,S,T+1} \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} R^e_{p,S,,T} \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} y_{T+1} \\ y_T \\ \vdots \\ y_{T-p+2} \\ y_{T-p+1} \end{bmatrix} e_{p,S,T}(T+1) + \beta \begin{bmatrix} \xi \\ y_T \\ \vdots \\ y_{T-p+2} \\ y_{T-p+1} \end{bmatrix}$$

To perfectly cancel out the lower p-1 elements of the last two vectors, β must be $-e_{p,S,T}(T+1)$. We do not need to know what ξ is at this point. All we need to know is the value of β.

Similarly, we can expand (57) using equations (3) and (5) as follows:

$$(61)\qquad \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ R^r_{p,S,T+1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ R^r_{p,S,,T} \end{bmatrix} + \begin{bmatrix} y_{T+1} \\ y_T \\ \vdots \\ y_{T-p+2} \\ y_{T-p+1} \end{bmatrix} r_{p,S,T}(T+1) + \gamma \begin{bmatrix} y_{T+1} \\ y_T \\ \vdots \\ y_{T-p+2} \\ \xi \end{bmatrix}$$

Clearly γ must be $-r_{p,S,T}(T+1)$ to yield the following:

$$(62)\qquad \boldsymbol{B}_{p,S,T+1} = \boldsymbol{B}_{p,S,T} - r_{p,S,T}(T+1)\begin{bmatrix} \boldsymbol{C}_{p-1,S+1,T+1} \\ 0 \end{bmatrix}$$

Note that at this point, there is some question as to how to calculate $\boldsymbol{C}_{p-1,S+1,T+1}$.

Equation (58) can be expanded as follows:

$$(63)\qquad \begin{bmatrix} R^e_{p,S+1,T} \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} R^e_{p,S,,T} \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} y_{S+p} \\ y_{S+p-1} \\ \vdots \\ y_{S+1} \\ y_S \end{bmatrix} e_{p,S,T}(S+p) + \delta \begin{bmatrix} \xi \\ y_{S+p-1} \\ \vdots \\ y_{S+1} \\ y_S \end{bmatrix}$$

As before, we can identify δ as $e_{p,S,T}(S+p)$ to yield the following:

$$(64)\qquad \boldsymbol{A}_{p,S+1,T} = \boldsymbol{A}_{p,S,T} + e_{p,S,T}(S+p)\begin{bmatrix} 0 \\ \boldsymbol{D}_{p-1,S,T} \end{bmatrix}$$

Similarly equation (59):

$$(65)\qquad \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ R^r_{p,S+1,T} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ R^r_{p,S,,T} \end{bmatrix} - \begin{bmatrix} y_{S+p} \\ y_{S+p-1} \\ \vdots \\ y_{S+1} \\ y_S \end{bmatrix} r_{p,S,T} + \delta \begin{bmatrix} y_{S+p} \\ y_{S+p-1} \\ \vdots \\ y_{S+1} \\ \xi \end{bmatrix}$$

Which yields the following:

$$(66)\qquad \boldsymbol{B}_{p,S+1,T} = \boldsymbol{B}_{p,S,T} + r_{p,S,T}\begin{bmatrix} \boldsymbol{D}_{p-1,S+1,T} \\ 0 \end{bmatrix}$$

There is some question about how to calculate $\boldsymbol{D}_{p-1,S+1,T}$. Since (64) and (66) are calculating the same quantities as (21) and (22), we can use whichever is easier to calculate. Using equation (22) means we don't have to calculate $\boldsymbol{D}_{p-1,S+1,T}$. This is another demonstration that these calculations are *not* unique. In fact, it is good to develop as many as we can, since some will be easier to evaluate than others.

We might hypothesize updates for $\boldsymbol{C}$ and $\boldsymbol{D}$ as follows:

(67) $$\boldsymbol{C}_{p,S,T+1} = \begin{bmatrix} 0 \\ \boldsymbol{C}_{p-1,S,T} \end{bmatrix} + \alpha \boldsymbol{A}_{p,S,T+1}$$

(68) $$\boldsymbol{D}_{p,S,T+1} = \begin{bmatrix} 0 \\ \boldsymbol{D}_{p-1,S,T} \end{bmatrix} + \beta \boldsymbol{A}_{p,S,T+1}$$

These are suggested by the form of (3) and that we know what the first element of $\boldsymbol{C}_{p,S,T+1}$ and $\boldsymbol{D}_{p,S,T+1}$ must be. We must have already computed $\boldsymbol{A}_{p,S,T+1}$ from (56). This leads immediately to the following update formulas:

(69) $$\boldsymbol{C}_{p,S,T+1} = \begin{bmatrix} 0 \\ \boldsymbol{C}_{p-1,S,T} \end{bmatrix} + e_{p,S,T+1}\left(R^{e}_{p,S,T+1}\right)^{-1} \boldsymbol{A}_{p,S,T+1}$$

(70) $$\boldsymbol{D}_{p,S,T+1} = \begin{bmatrix} 0 \\ \boldsymbol{D}_{p-1,S,T} \end{bmatrix} + e_{p,S,T+1}(S+p)\left(R^{e}_{p,S,T+1}\right)^{-1} \boldsymbol{A}_{p,S,T+1}$$

Given $\boldsymbol{C}_{p,S,T+1}$ and $\boldsymbol{D}_{p,S,T+1}$, we can produce $\boldsymbol{C}_{p,S+1,T+1}$ using (28) as follows:

(71) $$\boldsymbol{C}_{p,S+1,T+1} = \boldsymbol{C}_{p,S,T+1} + \frac{h_{p,S,T+1}}{(1-f_{p,S,T+1})} \boldsymbol{D}_{p,S,T+1}$$

$\boldsymbol{A}_{p,S,T+1} = \boldsymbol{A}_{p,S,T} - e_{p,S,T}(T+1) \begin{bmatrix} 0 \\ \boldsymbol{C}_{p-1,S,T} \end{bmatrix}$
$\boldsymbol{C}_{p,S,T+1} = \begin{bmatrix} 0 \\ \boldsymbol{C}_{p-1,S,T} \end{bmatrix} + e_{p,S,T+1}\left(R^{e}_{p,S,T+1}\right)^{\boldsymbol{-1}} \boldsymbol{A}_{p,S,T+1}$
$\boldsymbol{B}_{p,S,T+1} = \boldsymbol{B}_{p,S,T} - r_{p,S,T}(T+1) \begin{bmatrix} \boldsymbol{C}_{p-1,S+1,T+1} \\ 0 \end{bmatrix}$
$\boldsymbol{D}_{p,S,T+1} = \begin{bmatrix} 0 \\ \boldsymbol{D}_{p-1,S,T} \end{bmatrix} + e_{p,S,T+1}(S+p)\left(R^{e}_{p,S,T+1}\right)^{\boldsymbol{-1}} \boldsymbol{A}_{p,S,T+1}$

Note there is a bit of slight-of-hand in this table. Presumably $\boldsymbol{C}_{p-1,S+1,T+1}$ would have to have been computed at the previous value of p and saved. Also, we must either compute the scalers e and r directly or derive recursions for them that are not listed here.

$\boldsymbol{A}_{p,S+1,T} = \chi \left(\boldsymbol{A}_{p,S,T} + \dfrac{e_{p,S,T}(S+p)}{(1-f_{p,S,T})} \boldsymbol{D}_{p,S,T} \right)$
$\boldsymbol{B}_{p,S+1,T} = \chi \left(\boldsymbol{B}_{p,S,T} + \dfrac{r_{p,S,T}(S+p)}{(1-f_{p,S,T})} \boldsymbol{D}_{p,S,T} \right)$
$\boldsymbol{C}_{p,S+1,T} = \boldsymbol{C}_{p,S,T} + \dfrac{h_{p,S,T}}{(1-f_{p,S,T})} \boldsymbol{D}_{p,S,T}$

$$\begin{bmatrix} \boldsymbol{D}_{p,S+1,T} \\ 0 \end{bmatrix} = \boldsymbol{D}_{p+1,S,T} - r_{p+1,S,T}\left(R^{r}_{p,+1S,T}\right)^{-\mathbf{1}} \boldsymbol{B}_{p+1,S,T}$$

Note also that, as mentioned above, the calculation of $\boldsymbol{D}_{p,S+1,T}$ requires values of $\boldsymbol{D}_{p+1,S,T}$ and $\boldsymbol{B}_{p+1,S,T}$ which are of order one higher than the current step.

This constitutes a complete set of time updates for advancing either the starting sample or the ending sample of the analysis interval. Each update is an order-*p* calculation. With these two sets of updates, we can always make the analysis window one period in length (if the period doesn't change too quickly). This is sufficient for doing pitch-synchronous analysis. The solution after each update satisfies the normal equations.

NOW WHAT?

In this investigation, the goal is to be able to resynthesize the original speech in modified form, changing pitch, timing, and timbre in sometimes extreme manners. The most stable way to perform high-order synthesis is to use the normalized lattice form. This requires a set of reflection coefficients that correspond to the polynomial representing the forward filter (**A**) or the backward filter (**B**). We might think that we can just use the results of equations (48) and (49). Without further study, it is not clear that we can use this formulation of the reflection coefficients, since they are based on the time-shifted quantities $\boldsymbol{A}_{p,S+1,T}$ and $\boldsymbol{B}_{p,S,T-1}$. Rather than $\boldsymbol{A}_{p,S,T}$ and $\boldsymbol{B}_{p,S,T}$ This can be done by a modified form of the lattice filter that includes a calculation of the time-shifted polynomials, but there are a number of unanswered questions about stability and relation to the underlying distribution that are not trivial to analyze. For these reasons, I would like to suggest that we take $\boldsymbol{A}_{p,S,T}$ or $\boldsymbol{B}_{p,S,T}$ and recalculate a set of reflection coefficients that can be used for synthesis by a normalized lattice filter. This can be done by the use of the "step-down" procedure in Markelo&Gray **[MG1976]**This is a restatement of the Schur-Cohn algorithm for determining stability of a polynomia [***ref] and again the Szegö polynomials **[SZEGO1939]**. It is possible that the reflection coefficients from equations (46) or (47) are "close enough" to be used for synthesis, but it is not *a priori* clear without further study. Recall that there is a 1-to-1 correspondence between every stable polynomial and a sequence of reflection coefficients all of magnitude less than 1, and the "step-down" procedure will produce that unique set of coefficients. The problem is that the Schur-Cohn algorithm is numerically unstable and will not always produce a stable set of reflection coefficients [***refs], especially at high orders. Next, the recursion described in the literature **[MDKV77]** at high orders will sometimes produce unstable filters. Without a way to stabilize the Schur-Cohn recursion, *there is no way to produce a set of reflection coefficients that can be used for speech synthesis*. This has been noted in the literature [***refs]. There is some amount of research on increasing the stability of the Schur-Cohn recursion [***refs] that I will not try to evaluate here.

Another non-trivial problem is that after taking one step in time, from, say, T to T+1, we will not be able to take the next step without somehow calculating $\boldsymbol{C}_{p-1,S,T+1}$ and $\boldsymbol{D}_{p-1,S,T+1}$. There doesn't seem to be any closed-form solution to this problem, thus the recursions above allow us to advance by one step in time but no further. One must restart the recursion entirely to take another step. Note that restarting the recursion allows us to assure that every window is any given length, again allowing pitch-synchronous analysis. This is discussed further in Appendix B.

APPENDIX A: RECURSIVE CALCULATION OF FIRST ROW OF $\boldsymbol{R}_{p,S,T}$

In the order iteration described in **[MDKV77]**, the first or last row of $\boldsymbol{R}_{p,S,T}$ is used to calculate $\Delta_{p,S,T}$, from which the reflection coefficients κ_r and κ_e are formed. This operation must be performed $\boldsymbol{p}$ times. If this is done by direct application of (2), then this calculation becomes the most time-consuming part of the iteration. It is of interest to see if there is a recursive implementation that can reduce the overall computational burden. From (2), we can write the first row of $\boldsymbol{R}_{p,S,T}$ as follows:

(1) $\sigma_{p,S,T}(0,0{:}p) = \left[\sum_{t=S+p}^{T} y_t y_t \quad \sum_{t=S+p}^{T} y_t y_{t-1} \quad \cdots \quad \sum_{t=S+p}^{T} y_t y_{t-p}\right]$

On the next iteration, we will move from p to $p+1$ as follows:

(2) $\sigma_{p+1,S,T}(0,0{:}p+1) =$
$\left[\sum_{t=S+p+1}^{T} y_t y_t \quad \sum_{t=S+p+1}^{T} y_t y_{t-1} \quad \cdots \quad \sum_{t=S+p+1}^{T} y_t y_{t-p} \quad \sum_{t=S+p+1}^{T} y_t y_{t-p-1}\right]$

This vector is of length $p+1$. We extend (76) by annexing a zero at the end, then subtract (76) from (77) to get the following:

(3) $\sigma_{p+1,S,T}(0,0{:}p+1) - \left[\sigma_{p,S,T}(0,0{:}p) \;\; 0\right] =$
$\left[-y_{S+p} y_{S+p} \quad -y_{S+p} y_{S+p-1} \quad \cdots \quad -y_{S+p} y_S \quad \sum_{t=S+p+1}^{T} y_t y_{t-p-1}\right]$

The key is that the first p sums can be advanced to $p+1$ by subtracting off the first term. The vector must be extended by direct calculation of the new last term. Note that in **[MDKV77]**, we start with direct calculation for $p=1$, then advance p by one on each iteration. Note that a similar recursion can be derived for the last row of $\boldsymbol{R}_{p,S,T}$ as well.

(4) $\sigma_{p,S,T}(p,0{:}p) = \left[\sum_{t=S+p}^{T} y_{t-p} y_t \quad \sum_{t=S+p}^{T} y_{t-p} y_{t-1} \quad \cdots \quad \sum_{t=S+p}^{T} y_{t-p} y_{t-p}\right]$

On the next iteration, we will move from p to $p+1$ as follows:

(5) $\sigma_{p+1,S,T}(p+1,0{:}p+1) =$
$\left[\sum_{t=S+p+1}^{T} y_{t-p-1} y_t \quad \sum_{t=S+p+1}^{T} y_{t-p-1} y_{t-1} \quad \cdots \quad \sum_{t=S+p+1}^{T} y_{t-p-1} y_{t-p} \quad \sum_{t=S+p+1}^{T} y_{t-p-1} y_{t-p-1}\right]$

Subtract (79) from (80) to get the following update rule:

(6) $\sigma_{p+1,S,T}(p+1,0{:}p+1) - \left[0 \; \sigma_{p,S,T}(p,0{:}p)\right] =$
$\left[\sum_{t=S+p+1}^{T} y_{t-p-1} y_t \quad -y_{T-p-1} y_T \quad -y_{T-p-1} y_{T-1} \quad \cdots \quad -y_{T-p-1} y_{T-p-1}\right]$

This gives a more efficient way to update the first and/or last row of $\boldsymbol{R}_{p+1,S,T}$.

APPENDIX B: BACKING UP THE ORDER RECURSION

So the problem is this: Given $\boldsymbol{A}_{p,S,T}$, $\boldsymbol{B}_{p,S,T}$, $\boldsymbol{C}_{p,S,T}$, and $\boldsymbol{D}_{p,S,T}$, how do we find $\boldsymbol{A}_{p-1,S,T}$, $\boldsymbol{B}_{p-1,S,T}$, $\boldsymbol{C}_{p-1,S,T}$, and $\boldsymbol{D}_{p-1,S,T}$? We can start by identifying the last term of $\boldsymbol{A}_{p,S,T}$ as κ_r and the first term of $\boldsymbol{B}_{p,S,T}$ as κ_e. We can then invert the linear equations (44) and (45) as follows:

(7) $\begin{bmatrix} \boldsymbol{A}_{p-1,S+1,T} \\ 0 \end{bmatrix} = \frac{1}{1-\kappa_e \kappa_r}\left(\boldsymbol{A}_{p,S,T} - \kappa_r \boldsymbol{B}_{p,S,T}\right)$

(8) $\begin{bmatrix} 0 \\ \boldsymbol{B}_{p,-1S,T-1} \end{bmatrix} = \frac{1}{1-\kappa_e \kappa_r}\left(-\kappa_e \boldsymbol{A}_{p,S,T} + \boldsymbol{B}_{p,S,T}\right)$

Now we need to back out the time shift to encompass the entire interval from $\boldsymbol{S}$ to $\boldsymbol{T}$. We need to start with $\boldsymbol{C}$ and $\boldsymbol{D}$.

(9) $$\begin{bmatrix} \boldsymbol{C}_{p-1,S+1,T} \\ 0 \end{bmatrix} = \boldsymbol{C}_{p,S,T} - r_{p,S,T}\left(R^{r}_{p,S,T}\right)^{-1} \boldsymbol{B}_{p,S,T}$$

(10) $$\begin{bmatrix} 0 \\ \boldsymbol{D}_{p-1,S,T-1} \end{bmatrix} = D_{p,S,T} - e_{p,S,T}(S+p)\left(R^{e}_{p,S,T}\right)^{-\mathbf{1}} \boldsymbol{A}_{p,S,T}$$

We have an interesting problem of circularity in backing out the time shift. In the time-shift equations (21), (23), (28), and (29), there are a number of scalers of index $p-1, S, T$. But these are calculated from the non-time-shifted vectors, which are exactly what we wish to calculate.

REFERENCES:

[MORF74] – Martin Morf "Fast Algorithms for Multivariable Systems" Stanford University PhD Dissertation 1974

[LEE1980] – Daniel Tai Lik Lee "Canonical Ladder Form Realizations and Fast Estimation Algorithms" Stanford University PhD Dissertation 1980

[VIEIRA1977] – Augusto Cesar Gadelha Vieira, "Matrix Orthogonal Polynomials with Applications to Autoregressive Modeling and Ladder Forms", Stanford University PhD Dissertation 1977

[MDKV77] Martin Morf, Bradley Dickinson, Thomas Kailath, Augusto Vieira, "Efficient Solution of Covariance Equations for Linear Prediction" IEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-25, No. 5, October 1977, pp429-433

[SZEGO1939] – Gabor Szegö "Orthogonal Polynomials" 1939 (Reprinted by the American Mathematical Society in 2000)

[MG1976] – John D. Markel, Augustine H. Gray, Jr. "Linear Prediction of Speech" Springer-Verlag 1976

[LMF1981] – Daniel Lee, Martin Morf, Benjamin Friedlander "Redursive Least Squares Ladder Estimation Algorithms" IEEE Transactions on Circuits and Systems, Vol. CAS-28, No. 6, June, 1981 pp467-481

[LEVINSON1947] – N. Levinson "The Wiener RMS (Root Mean Square) Error Criterion in Filter Design and Prediction" J. Math. Phys. 25, 1947, pp261-278

[BURG75] – John Parker Burg, "Maximum Entropy Spectral Analysis", PhD Dissertation Stanford University 1975

Schur-Cohn

Stability of Schur-Cohn