II - C. TIME SERIES DERIVATIONS AND THE BURG ESTIMATION TECHNIQUE

Up to now, our study has started with the autocorrelation values, $R(n)$, $-N \leq n \leq N$, and we have not dealt directly with the time series to any appreciable extent. To get closer to the problems involved in analysis of data samples, we shall rederive some of our previous results by applying prediction error filters to the time series. The connection between the maximum entropy assumption and modeling the time series as an autoregressive process will be made clear. The problems in directly estimating the autocorrelation function from time series data will also be discussed. These problems are avoided by using the Burg technique, which directly estimates reflection coefficients instead of autocorrelation values.

1. The Least Mean Square Error Linear Predictor

Suppose we have a stationary time series, x_s , and we wish to predict the next value of the time series by using a linear combination of the N immediately previous samples. Let $-a_n$ be the weight on the nth previous sample. Then the predicted value of x_s will be

$$\sum_{n=1}^{N} (-a_n)\, x_{s-n}$$

and the error in the prediction will be

$$x_s - \sum_{n=1}^{N} (-a_n)\, x_{s-n} = \sum_{n=0}^{N} a_n\, x_{s-n} \quad ,$$

where we define $a_0 \equiv 1$. The mean square error is given by

$$\overline{\sum_{m=0}^{N} a_m^* x_{s-m}^* \sum_{n=0}^{N} a_n\, x_{s-n}} = \sum_{m=0}^{N}\sum_{n=0}^{N} a_m^*\, \overline{x_{s-m}^*\, x_{s-n}}\, a_n$$

$$= \sum_{m=0}^{N}\sum_{n=0}^{N} a_m^*\, R(m-n)\, a_n \quad , \qquad \text{(II-46)}$$

where $R(\tau) = \overline{x_n^*\, x_{n+\tau}}$ is the autocorrelation function of the stationary time series.

There are several ways of solving for the a_n (n=1 to N) that make the mean square error a minimum. We shall use a method similar to completing the square. For simplicity, we shall assume that the N th order autocorrelation matrix formed from $R(\tau)$ is positive definite.

Let b_n (n=0 to N) , $b_0 \equiv 1$, be the solution to the matrix equation

$$\begin{bmatrix} R(0) & R(-1) & \cdots & R(-N) \\ R(1) & R(0) & \cdots & R(-N+1) \\ \vdots & & & \\ R(N) & R(N-1) & \cdots & R(0) \end{bmatrix} \begin{Bmatrix} 1 \\ b_1 \\ \vdots \\ b_N \end{Bmatrix} = \begin{Bmatrix} P_N \\ 0 \\ \vdots \\ 0 \end{Bmatrix} . \quad \text{(II-47)}$$

Next, using (II-47), we note that

$$\sum_{m=0}^{N} \sum_{n=0}^{N} (a_m^* - b_m^*) R(m-n) (a_n - b_n) = \sum_{m=0}^{N} \sum_{n=0}^{N} \{a_m^* R(m-n) a_n$$

$$- a_m^* R(m-n) b_n - b_m^* R(m-n) a_n + b_m^* R(m-n) b_n \}$$

$$= \sum_{m=0}^{N} \sum_{n=0}^{N} a_m^* R(m-n) a_n - P_N .$$

Thus we see that

$$\sum_{m=0}^{N} \sum_{n=0}^{N} a_m^* R(m-n) a_n = P_N + \sum_{m=0}^{N} \sum_{n=0}^{N} (a_m^* - b_m^*) R(m-n)(a_n - b_n) . \quad \text{(II-48)}$$

Now since $P_N > 0$ and $R(m-n)$ is positive definite, the minimum value of the right hand side occurs when $a_m = b_m$. Thus, P_N is the least mean square prediction error and solving (II-47) gives us the optimum linear filter. If our filter a_n differs from the optimum filter b_n ,then (II-48) tells us how much additional mean square error the non-optimum filter will have over the optimum filter.

Suppose now that we wish to find the linear, least mean square error estimate of x_s, not from the N previous values of the time series, but from the next N values. Let $-d_n$ be the optimum weight for the nth later sample. Then the Nth order backward predicted value of x_s will be

$$\sum_{n=1}^{N} (-d_n)\, x_{s+n}$$

and the error in the estimate will be

$$x_s - \sum_{n=1}^{N} (-d_n)\, x_{s+n} = \sum_{n=0}^{N} d_n\, x_{s+n} ,$$

where $d_0 \equiv 1$. If we continue on and develop the equations corresponding to (II-46) through (II-48), it will be obvious that we have just reversed the direction of time. Thus, according to (II-47) the d_n will satisfy the matrix equation

$$\begin{bmatrix} R(0) & R(1) & R(N) \\ R(-1) & R(0) & R(N-1) \\ & & \\ R(-N) & R(1-N) & R(0) \end{bmatrix} \begin{Bmatrix} 1 \\ d_1 \\ \\ d_N \end{Bmatrix} = \begin{Bmatrix} P_N \\ 0 \\ \\ 0 \end{Bmatrix} \qquad \text{(II-49)}$$

or, by taking complex conjugates,

$$\begin{bmatrix} R(0) & R(-1) & R(-N) \\ R(1) & R(0) & R(1-N) \\ & & \\ R(N) & R(N-1) & R(0) \end{bmatrix} \begin{Bmatrix} 1 \\ d_1^* \\ \\ d_N^* \end{Bmatrix} = \begin{Bmatrix} P_N \\ 0 \\ \\ 0 \end{Bmatrix} \qquad \text{(II-50)}$$

Thus, the optimum backward prediction error filter is just the complex conjugate, time reverse of the optimum forward prediction error filter. Thus, we can write the backward prediction error, h_s, in terms of the forward prediction error filter coefficients as

$$h_s = \sum_{n=0}^{N} a_n^* x_{s+n} \quad .$$

We should note that the backward least mean square prediction error is also equal to P_N. However, the fact that the forward and backward prediction error filters have the same output power is not a unique property of prediction error filters. Actually this property is a simple consequence of the complex conjugate, time reverse property. This is clear from the fact that for an arbitrary filter, $a_0 + a_1 z + \ldots + a_N z^N$, we have

$$\left| \sum_{n=0}^{N} a_n z^n \right|^2 = \sum_{n=0}^{N} a_n z^n \sum_{m=0}^{N} a_m^* z^{-m} = \left| \sum_{m=0}^{N} a_m^* z^{-m} \right|^2 .$$

Since the power responses of these two filters are the same, their output powers are the same for any input time series.

Finally, we should note that with $a_n = d_n^*$, we can write (II-49) in the form of the by now familiar equation,

$$\begin{bmatrix} R(0) & R(-1) & R(-N) \\ R(1) & R(0) & R(1-N) \\ R(N) & R(N-1) & R(0) \end{bmatrix} \begin{Bmatrix} a_N^* \\ a_{N-1}^* \\ 1 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ P_N \end{Bmatrix} \qquad \text{(II-51)}$$

2. The Levinson Algorithm Revisited

It is instructive to develop the modern Levinson algorithm by means of prediction error time series. To begin, suppose we filter the x_s time series with its optimum, Nth order, forward prediction error filter to get the forward prediction error time series, e_s, where

$$e_s = \sum_{n=0}^{N} a_n x_{s-n} \quad . \tag{II-52}$$

The corresponding backward prediction error time series, h_s, is given by

$$h_s = \sum_{n=0}^{N} a_n^* x_{s+n} \quad . \tag{II-53}$$

These time series and filters can be pictured for N = 2 as

$$\begin{array}{ccccccccc}
\cdots & e_{-3} & e_{-2} & e_{-1} & e_0 & e_1 & e_2 & e_3 & \cdots \\
 & & & a_2 & a_1 & 1 & & & \\
\cdots & x_{-3} & x_{-2} & x_{-1} & x_0 & x_1 & x_2 & x_3 & \cdots \\
 & & 1 & a_1^* & a_2^* & & & & \\
\cdots & h_{-3} & h_{-2} & h_{-1} & h_0 & h_1 & h_2 & h_3 & \cdots
\end{array} \tag{II-54}$$

Here, the filters are to be thought of as sliding along the x time series, with the e and h time series being output relative to the unity weight coefficient of their respective filters.

The weights a_1 and a_2 are optimum weights for making $\overline{e_s^* e_s}$ a minimum. Thus we have $\overline{x_{s-1}^* e_s} = \overline{x_{s-2}^* e_s} = 0$, since

otherwise e_s would still be partially predictable from x_{s-1} and x_{s-2}. Furthermore, since $e_s = x_s + a_1 x_{s-1} + a_2 x_{s-2}$, we have $\overline{e_s^* e_s} = \overline{e_s^* x_s} = P_N$. A similar set of equations holds for the backward prediction case. The problem we shall now look at is how to include x_{s-3} in an optimum way to improve our prediction of x_s.

If we only had x_{s-3} available for estimating x_s, then the least mean square prediction would be gx_{s-3}, where

$$g = \frac{\overline{x_{s-3}^* x_s}}{\overline{x_{s-3}^* x_{s-3}}} .$$

However, x_{s-3} is correlated with x_{s-1} and x_{s-2} and thus we cannot simply sum the prediction from x_{s-3} with the prediction from x_{s-1} and x_{s-2}. Such a procedure would end up predicting part of x_s twice. However, if we had first removed the part of x_{s-3} that is correlated with x_{s-1} and x_{s-2}, then summing the two predictions would give us the correct answer. Looking at the backward prediction error term, $h_{s-3} = x_{s-3} + a_1^* x_{s-2} + a_2^* x_{s-1}$, we see that $\overline{x_{s-1}^* h_{s-3}} = \overline{x_{s-2}^* h_{s-3}} = 0$ and thus h_{s-3} is that part of x_{s-3} that is uncorrelated with x_{s-1} and x_{s-2}.

Noticing that $\overline{h_{s-3}^* e_s} = \overline{h_{s-3}^* x_s} + a_1 \overline{h_{s-3}^* x_{s-1}} + a_2 \overline{h_{s-3}^* x_{s-2}} = \overline{h_{s-3}^* x_s}$, the optimum weight $(-c_3)$ to apply to h_{s-3} to predict x_s is

$$-c_3 = \frac{\overline{h_{s-3}^* x_s}}{\overline{h_{s-3}^* h_{s-3}}} = \frac{\overline{h_{s-3}^* e_s}}{P_2} \qquad \text{(II-55)}$$

where $P_2 = \overline{h_s^* h_s} = \overline{e_s^* e_s}$. Summing the two predictions, the third order prediction error series becomes, $e_s' = e_s + c_3 h_{s-3} = x_s + (a_1 + c_3 a_2^*)\, x_{s-1} + (a_2 + c_3 a_1^*)\, x_{s-2} + c_3\, x_{s-3}$ (see II-19) and $\overline{e_s'^* e_s'} = \overline{(e_s^* + c_3^* h_{s-3}^*)(e_s + c_3 h_{s-3})} = \overline{e_s^* e_s}\,(1 - c_3^* c_3)$ (see II-20) by use of (II-55).

Finally, we complete this derivation of the modern Levinson algorithm by noting that

$$\overline{h_{s-3}^* e_s} = \{0 \quad a_2^* \quad a_1^* \quad 1\} \begin{bmatrix} R(0) & R(-1) & R(-2) & R(-3) \\ R(1) & R(0) & R(-1) & R(-2) \\ R(2) & R(1) & R(0) & R(-1) \\ R(3) & R(2) & R(1) & R(0) \end{bmatrix} \begin{Bmatrix} 1 \\ a_1 \\ a_2 \\ 0 \end{Bmatrix} = \Delta_2 \qquad \text{(II-56)}$$

and thus

$$c_3 = -\frac{\Delta_2}{P_2}$$

corresponding to (II-17) and (II-18).

3. Whitening the Time Series

The error time series produced by an infinite length prediction error filter has a white power spectrum. We easily prove this by noting that 1) $\overline{e_s^* x_{s-n}} = 0$ for $n \geq 1$ and 2) e_{s-m} is a linear combination of the x_{s-n}, $n \geq m$. Therefore $\overline{e_s^* e_{s-m}} = 0$ for all $m > 0$ and we have the z tranform of the autocorrelation of e_s to be simply $\overline{e_s^* e_s} = P_\infty$. The power spectrum of the x_s time series can easily be found from the relation that the output spectrum is the product

of the input spectrum times the power response of the filter. In this case, the output spectrum is $P_\infty/2W = P_\infty \Delta t$, and if the z transform of the prediction error filter is $\sum_{n=0}^{\infty} a_n z^n$, the output spectrum is

$$P_{out}(z) = \frac{P_\infty \Delta t}{[\sum_{n=0}^{\infty} a_n z^n][\sum_{n=0}^{\infty} a_n^* z^{-n}]} \quad .$$

Let us look at (II-54) again and assume that $\overline{e_s^* x_{s-3}} = 0$. Then $\overline{e_s^* h_{s-3}} = 0$ and $c_3 = 0$. In this case, the optimum 3rd order prediction error filter is the same as the optimum 2nd order prediction error filter. Continuing on, let us also assume that $\overline{e_s^* x_{s-n}} = 0$ for $n > 3$. If this is true, we can see that the 2nd order prediction error filter will also be the optimum infinite length filter. It then follows that the true spectrum of the x_s time series is given by

$$\frac{P_2 \Delta t}{[\sum_{n=0}^{2} a_n z^n][\sum_{n=0}^{2} a_n^* z^{-n}]}$$

which is the second order maximum entropy spectrum.

4. The Autoregressive Process

Let us take the prediction error equation

$$e_s = \sum_{n=0}^{N} a_n x_{s-n} \quad ,$$

and rewrite it as

$$x_s = e_x - \sum_{n=1}^{N} a_n x_{s-n} \quad . \qquad \text{(II-60)}$$

If we assume that the e_s are linearly independent variables, i.e., if the e_s time series has a white spectrum, then the x_s time series generated by (II-60) is an Nth order autoregressive process. In order for the feedback operation of (II-60) to be stable, the filter $\sum_{n=0}^{N} a_n z^n$ must be minimum phase. We have already shown this to be true since it is an optimum prediction error filter. The spectrum of the x_s time series is clearly given by the Nth order maximum entropy spectrum.

5. Problems with Direct Estimation of Autocorrelation Functions

So far, in this section we have discussed prediction error filters in terms of time series but have assumed that the autocorrelation function is available for their calculation. Of course, in practice, the statistics of the stationary time series must be estimated from the time series itself. We shall discuss here certain problems which arise when one attempts to directly estimate the autocorrelation function from a finite sample of a stationary time series.

Suppose our sample consists of N consecutive values $(x_1, x_2, \ldots, x_N)$ and that we wish to estimate the autocorrelation function, $R(\tau) = x_n^* x_{n+\tau}$. One common method of doing this is by calculating the sum of lag products, i.e.,

$$\sum_{n=1}^{N-\tau} x_n^* x_{n+\tau} \quad .$$

We should note that this function of τ is the exact "energy" autocorrelation function of the infinite time series $(\ldots, 0, 0, x_1, x_2, \ldots, x_{N-1}, x_N, 0, 0, \ldots)$. If one divides this "energy" function by N to get an estimate of the autocorrela-

tion function of the stationary time series, then such an estimate is guaranteed to be a possible autocorrelation function. Our autocorrelation matrices will then be positive definite and the solutions for the prediction error filters will make sense.

The problem with estimating the autocorrelation function by

$$\hat{R}(\tau) = \frac{1}{N} \sum_{n=1}^{N} x_n^* x_{n+\tau} \qquad \text{(II-57)}$$

is that $\hat{R}(\tau)$ is biased for $\tau \neq 0$. That is, the average value of $\hat{R}(\tau)$ is not equal to the true value of $R(\tau)$. Instead, we have

$$\overline{\hat{R}(\tau)} = \frac{N-\tau}{N} R(\tau) .$$

One can avoid this problem by changing (II-57) to

$$\hat{R}(\tau) = \frac{1}{N-\tau} \sum_{n=1}^{N-\tau} x_n^* x_{n+\tau} , \qquad \text{(II-58)}$$

so that

$$\overline{\hat{R}(\tau)} = R(\tau) .$$

However, if one uses (II-58), then the resulting function may not be a possible autocorrelation function and our matrix equations will not give sensible answers. A simple example of how (II-58) can fail is to consider the three point sample (1.0, 1.1, 1.0). Then, (II-58) gives $\hat{R}(0) = 1.07$ and $\hat{R}(1) = 1.1$.

A considerably different approach to estimating the autocorrelation function is made highly practical by the fast fourier transform. In this case, one takes the discrete fast fourier transform of the time sample, forms the frequency spectral function by taking the absolute square values and then inverse fourier transforming to get the autocorrelation function. This procedure is a computationally quick method of obtaining the function

$$\hat{R}(\tau) = \frac{1}{N} \sum_{n=1}^{N} x_n^* x_{n+\tau} , \qquad \text{(II-59)}$$

where the time series is $(\ldots, x_{N-1}, x_N, x_1, x_2, \ldots, x_N, x_1, x_2, \ldots)$, i.e., where we assume that the N point sample repeats itself periodically. It is clear that $R(\tau)$ will give us a possible autocorrelation function , but including the product $x_N^* x_1$ as part of our estimate of $\hat{R}(1)$ does not make sense.

The problems encountered by these different ways of estimating the autocorrelation function are due to assumptions imposed upon us about the ends of the data sample. If our data consists of one long data sample, so that there are only two ends to give us trouble, then these problems are not severe. However, if our data consists of many short samples, where the ratio of ends to data is high, the conventional methods discussed here have serious problems. We shall now discuss a method of estimating the statistics of a stationary time series which avoids these "end effect" problems.

6. The Burg Technique of Estimating Reflection Coefficients

As was shown in II-B.5, there is a one-to-one relationship between the sequences $(R(0), R(1), R(2), \ldots)$ and $(R(0), C_1, C_2, \ldots)$. Thus, the second order statistics of a stationary time series can be specified in terms of $R(0)$ and the reflection coefficients as well as in terms of the autocorrelation function (or the power spectrum). The Burg technique of estimating second order statistics from time series data is based on direct estimation of reflection coefficients instead of autocorrelation function values. By doing this, it will be seen that the end effect problem is avoided.

To describe the Burg technique in general terms, we shall assume that the reflection coefficients $C_1, C_2, \ldots, C_{N-1}$ have been obtained and that we now wish to estimate C_N. From $C_1, C_2, \ldots, C_{N-1}$, we know the N-1 th order prediction error filter, $1 + a_1 z + a_2 z^2 + \ldots + a_{N-1} z^{N-1}$. From this, we can write the Nth order prediction error filter in the form $1 + (a_1 + C_N a^*_{N-1})z + \ldots + (a_{N-1} + C_N a^*_1) z^{N-1} + C_N z^N$. The correct value for C_N is the one that makes the power output of this filter a minimum when it is applied to the stationary time series.

To estimate the power output of this filter, we apply it to our time series data samples. In particular, we consider all possible sets of N+1 consecutive samples that can be formed from our time series data. For example, if N=3, and our data consists of three separate samples, $(x_1, x_2, x_3, x_4, x_5, x_6, x_7)$, $(x_{50}, x_{51}, x_{52}, x_{53}, x_{54})$ and (x_{90}, x_{91}, x_{92}), then we have six possible quadruplets, (x_1, x_2, x_3, x_4), (x_2, x_3, x_4, x_5), (x_3, x_4, x_5, x_6), (x_4, x_5, x_6, x_7), $(x_{50}, x_{51}, x_{52}, x_{53})$

and $(x_{51}, x_{52}, x_{53}, x_{54})$. Let us assume that there are M sets of N+1 tuplets, i.e., $(x_{1m}, x_{2m}, \ldots, x_{N+1,m})$, m=1 to M and, to be most general, that we use a positively weighted average in estimating the average square output in applying the prediction error filter to these M samples. The estimated power is then taken as

$$\sum_{m=1}^{M} W_m \left| C_N x_{1,m} + (a_{N-1} + C_N a_1^*) x_{2,m} + \ldots + (a_1 + C_N a_{N-1}^*) x_{N,m} + x_{N+1,m} \right|^2$$

with $\sum_{m=1}^{M} W_m = 1 \quad , \quad W_m \geq 0 \; .$

If we write

$$e_m = a_{N-1} x_{2,m} + \ldots + a_1 x_{N,m} + x_{N+1,m}$$

and

$$b_m = x_{1,m} + a_1^* x_{2,m} + \ldots + a_{N-1}^* x_{N,m} \quad ,$$

then (II-60) can be written as

$$\sum_{m=1}^{M} W_m \left| e_m + C_N b_m \right|^2 \quad . \tag{II-61}$$

We note that e_m and b_m are forward and backward prediction error values.

Since the average power out of the backward prediction error filter is the same as out of the forward filter, we also could have estimated this power equally well by

$$\sum_{m=1}^{M} W_m \left| b_m + C_N^* e_m \right|^2 \quad . \tag{II-62}$$

We note in particular that the average values of (II-61) and (II-62) are the same, independent of the value of C_N . However, for a general set of M N+1 tuplets, (II-61) and (II-62) do not have the same values. Since there is no reason to prefer (II-61) to (II-62) and vice versa, and since they are estimates of the same quantity, the average of the two should be a better estimator than either one alone. Thus we shall use

$$\frac{1}{2} \sum_{m=1}^{M} W_m \{ |e_m + C_N b_m|^2 + |b_m + C_N^* e_m|^2 \} \quad . \tag{II-63}$$

Note that the same weight, W_m , is used on the mth sample for both the forward and backward terms.

We now wish to find the value of C_N that makes (II-63) a minimum. Since (II-63) can be written as

$$\{1 \quad C_N^*\} \begin{bmatrix} \sum_{m=1}^{M} W_m (e_m^* e_m + b_m^* b_m) & 2 \sum_{m=1}^{M} W_m e_m^* b_m \\ 2 \sum_{m=1}^{M} W_m b_m^* e_m & \sum_{m=1}^{M} W_m (e_m^* e_m + b_m^* b_m) \end{bmatrix} \begin{bmatrix} 1 \\ C_N \end{bmatrix} ,$$

the minimizing value of C_N is

$$C_N = - \frac{2 \sum_{m=1}^{M} W_m b_m^* e_m}{\sum_{m=1}^{M} W_m (e_m^* e_m + b_m^* b_m)} \quad . \tag{II-64}$$

Furthermore, because $W_m \geq 0$, our two by two matrix is non-negative definite. Thus,

$$\left| 2 \sum_{m=1}^{M} W_m b_m^* e_m \right| \leq \sum_{m=1}^{M} W_m (e_m^* e_m + b_m^* b_m)$$

and we see that $|C_N| \leq 1$, no matter what values we have for our data samples. Thus, the estimated value for C_N is always a possible value for the N th reflection coefficient.

Depending on the maximum length of any particular sample of data, it is clear that one can use this procedure to estimate C_1, C_2, ... Then together with a reasonable estimate for R(0), one has the information to calculate the corresponding autocorrelation function and/or maximum entropy spectrum. It is to be noted that the Burg technique allows one to handle disjoint time series samples in a very general way and that there are no implied assumptions about the data off of the ends of the samples. Furthermore, since the magnitude of all estimated reflection coefficients must be less than or equal to unity, the procedure automatically generates a possible sequence of reflection coefficients.