

```

integer addr(0:1,10000),loc(0:1,10000)
integer*1 s(5100)
integer size,sj_snext
data addr/20000*0/
...
c array "s" is string of binary digits: s(i) = 0 or 1
c length of "s" has to be somewhat larger than imax
...
ksum=0
size=1
inextword=2
nwords=1
do 1 i=1,imax
  k=1
  n1=1
  100 sj=s(i+k-1)
  nn=addr{sj,n1}
  ksum=ksum+1
  if(nn.ne.0) then
    k=k+1
    if(nn.lt.0) then
      size=size+1
      addr{sj,n1}=size
      nn=size
      jnext=loc(sj,n1)+1
      snext=s(jnext)
      addr{snext,size}=-k
      loc{snext,size}=jnext
      ksum=ksum+1
    endif
    n1=nn
    goto 100
  endif
  addc{sj,n1}=-k
  loc{sj,n1}=i+k-1
  if(i.eq.inextword) then
    inextword=i+k
    nwords=nwords+1
  endif
1 continue
c printout:
xlog4=alog(imax*1.)/alog(2.)
eta=imax*xlog4/ksum
ziv=xlog4/imax*nwords
print *, 'Length of bit chain (=imax): ',imax
print *, 'Tree size (nr. of nodes & leaves): ',size
print *, 'Entropy estimate eta(imax): ',eta
print *, 'Number of words in Lempel-Ziv parsing: ',nwords
print *, 'Lempel-Ziv entropy estimate: ',ziv
...

```

Fig. 6. Part of Fortran routine.

## REFERENCES

- [1] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*. Urbana, IL: Univ. of Illinois Press, 1949.
- [2] A. N. Komogorov, *Dokl. Akad. Nauk. SSSR*, vol. 124, p. 754, 1959.
- [3] Ya. G. Sinai, *Dokl. Akad. Nauk. SSSR*, vol. 124, p. 768, 1959.
- [4] C. E. Shannon, "Prediction and entropy of printed English," *Bell Syst. Tech. J.*, vol. 30, p. 50, 1951.
- [5] N. G. Burton and J. C. R. Licklider, *Amer. J. Psychol.*, vol. 68, p. 652, 1955.
- [6] E. B. Newman and N. C. Waugh, *Inform. Contr.*, vol. 3, p. 141, 1960.
- [7] D. H. Carson, *Kybernetik*, vol. 1, p. 46, 1961.
- [8] L. J. Gerstman, *J. Experimental Psychol.*, vol. 44, p. 2, 1952.
- [9] G. P. Bogulawskaja, "Ein neues Experiment zur Bestimmung der Entropie der englischen Sprache," in *Sprachstatistik*, P. M. Alexejew et al., Eds. Munich: W. Fink, 1973.
- [10] T. M. Cover and R. C. King, *IEEE Trans. Inform. Theory*, vol. IT-24, p. 413, 1978.
- [11] A. Lempel and J. Ziv, *IEEE Trans. Inform. Theory*, vol. IT-22, p. 75, 1976.
- [12] ———, *IEEE Trans. Inform. Theory*, vol. IT-23, p. 337, 1977.
- [13] A. N. Kolmogorov, "Three approaches to the quantitative definition of information," *Probl. Inform. Transmiss.*, vol. 1, p. 1, 1965.
- [14] R. J. Solomonoff, *Inform. Contr.*, vol. 7, p. 224, 1964.
- [15] G. J. Chaitin, *J. Assoc. Comput. Mach.*, vol. 13, p. 547, 1966.
- [16] A. Lempel and J. Ziv, *IEEE Trans. Inform. Theory*, vol. IT-24, p. 530, 1978.
- [17] F. Kaspar and H. G. Schuster, Univ. Frankfurt preprint, 1986.
- [18] M. Rodeh, V. R. Pratt, and S. Even, *J. Assoc. Comput. Mach.*, vol. 28, p. 16, 1981.
- [19] J. Rissanen, *IEEE Trans. Inform. Theory*, vol. IT-29, p. 656, 1983.
- [20] J. Ziv and A. Lempel, *IEEE Trans. Inform. Theory*, vol. IT-24, p. 530, 1978.
- [21] R. Badii and A. Politi, *Phys. Rev. Lett.*, vol. 52, p. 1661, 1984.
- [22] J. Guckenheimer, *Contemp. Math.*, vol. 28, p. 357, 1984.
- [23] A. Brandstaeter et al., *Phys. Rev. Lett.*, vol. 51, p. 1442, 1983.
- [24] J. D. Farmer, "Order within chaos," thesis, Univ. of California, Santa Cruz, 1981.
- [25] G. G. Langdon, *IEEE Trans. Inform. Theory*, vol. IT-29, p. 284, 1983.
- [26] J. Rissanen and G. G. Langdon, *IEEE Trans. Inform. Theory*, vol. IT-27, p. 12, 1981.
- [27] R. W. Hamming, *Coding and Information Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1980.
- [28] P. Grassberger, *J. Stat. Phys.*, vol. 45, p. 27, 1986.

## Impedance-Domain Levinson Algorithms

Y. BISTRITZ, SENIOR MEMBER, IEEE, H. LEV-ARI, MEMBER, IEEE,  
AND T. KAILATH, FELLOW, IEEE

**Abstract**—Several computationally efficient versions of the Levinson algorithm for solving linear equations with Toeplitz and quasi-Toeplitz matrices are presented, motivated by a new stability test due to Bistritz (1983). The new versions require half the number of multiplications and the same number of additions as the conventional form of the Levinson algorithm. The saving is achieved by using three-term (rather than two-term) recursions and propagating them in an *impedance/admittance* (or *impedance*) domain rather than the conventional *scattering* domain. One of our recursions coincides with the recent results of Delsarte and Genin on "split-Levinson" algorithms for symmetric Toeplitz matrices, where the efficiency is gained by using the symmetric and skew-symmetric versions of the usual polynomials. This special structure is lost in the quasi-Toeplitz case, but one still can obtain similar computational reductions by suitably using three-term recursions in the impedance domain.

## I. INTRODUCTION

In [1]–[4] Bistritz presented several tests for the root distribution of polynomials with respect to the unit circle that take only half the number of multiplications (and the same number of additions) as the well-known Schur–Cohn (or, equivalently, the Jury–Marden table) test [5]. Estimation theorists have long known that the Schur–Cohn test is essentially a reverse (degree-reducing) form of the Levinson algorithm for the prediction of stationary time series, which have Toeplitz covariance matrices [6]. It is, therefore, reasonable to expect that similar reductions in computational complexity could also be obtained for the Levinson algorithm. In fact, such improved algorithms were obtained by Bube and Burrige [17], by Delsarte and Genin [8], by us [13], and by Krishna and Morgera [14]. Our contribution is a different approach that, among other things, serves to delimit the *whole set of efficient Levinson algorithms*, which includes, among others, the "split Levinson" algorithm of Delsarte and Genin [8] and the recursion of Bistritz's root-location procedure [1]. We find that there are three classes of reduced complexity Levinson-type algorithms and demonstrate the relationships between them. Moreover, our approach enables us to obtain similar results for a class of nonstationary processes with what we have called quasi-Toeplitz covariance matrices. Before making more specific comparisons, it will be useful to introduce some notation and background results.

The Levinson algorithm is a fast method of solving the set of linear equations

$$\begin{bmatrix} a_{n,n} & \cdots & a_{n,1} & 1 \end{bmatrix} \mathbf{R}_{0:n} = \mathbf{R}_n^e \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix} \quad (1a)$$

for the unknowns  $\{a_{n,i}, R_n^e\}$ , where  $\mathbf{R}_{0:n}$  is the square Hermitian Toeplitz matrix

$$\mathbf{R}_{0:n} := \{c_{i-j}; 0 \leq i, j \leq n\}. \quad (1b)$$

Manuscript received April 15, 1986; revised August 18, 1988. This work was supported in part by the Air Force Office of Scientific Research, Air Force Systems Command, under Contract AF-83-0228, by the Department of the Navy, Office of Naval Research, under Contract N00014-85-K-0612, the US Army Research Office under Contract DAAL03-86-K-0045, and (for the first author also) by a Chaim Weizmann postdoctoral fellowship award. This correspondence was presented in part at the IEEE International Conference on Acoustics, Speech, and Signal Processing, Tokyo, Japan, April 1986.

Y. Bistritz was with the Information Systems Laboratory, Stanford University, Stanford, CA. He is now with the Department of Electrical Engineering, Tel-Aviv University, Tel-Aviv 69978, Israel.

H. Lev-Ari and T. Kailath are with the Information Systems Laboratory, Stanford University, Stanford, CA 94305, USA.

IEEE Log Number 8927895.

When the matrix  $\mathbf{R}_{0:n}$  is real (i.e.,  $c_{j-i} = c_{i-j}$ ), (1a) can be solved via the (Levinson) recursions

$$\begin{pmatrix} a_n(z) \\ a_n^\#(z) \end{pmatrix} = L_n(z) \begin{pmatrix} a_{n-1}(z) \\ a_{n-1}^\#(z) \end{pmatrix} \\ L_n(z) = \begin{pmatrix} 1 & -k_n \\ -k_n & 1 \end{pmatrix} \begin{pmatrix} z & 0 \\ 0 & 1 \end{pmatrix} \quad (2a)$$

where

$$a_n(z) := \sum_{i=0}^n a_{n,i} z^{n-i} \quad (2b)$$

$a_0(z) = 1$  and  $a_n^\#(z) := z^n a_n(z^{-1})$ , the reverse polynomial of  $a_n(z)$ . The coefficients  $k_n$  are computed by the inner-product formula

$$k_n = [R_{n-1}^c]^{-1} \sum_{i=0}^{n-1} a_{n-1,i} c_{n-i} \quad (2c)$$

and  $R_n^c$  are obtained from the recursion

$$R_n^c = (1 - k_n^2) R_{n-1}^c, \quad R_0^c = c_0. \quad (2d)$$

A real polynomial  $f(z) := \sum_{i=0}^n f_i z^i$  is called *symmetric* if  $f^\#(z) = f(z)$ , i.e., if  $f_i = f_{n-i}$ ; such a polynomial is fully specified by half of its coefficients (or, more precisely,  $[(n+1)/2]$  coefficients, where  $[x]$  denotes integer part of  $x$ ). Since  $a_n(z) + a_n^\#(z)$  is symmetric, one way of trying to reduce complexity is to try to propagate  $a_n(z) + a_n^\#(z)$  rather than  $a_n(z)$ . This turns out to work, and interestingly, propagating  $a_n(z) + b_n(z)$  instead of  $a_n(z)$  results in a reduction in computation even in cases where  $b_n(z)$  is *not* the reciprocal polynomial  $a_n^\#(z)$ .

This so-called *quasi-Toeplitz* [7] case occurs when the matrix  $\mathbf{R}_{0:n}$  in (1a) has displacement inertia  $\{1,1\}$ , i.e.,  $\mathbf{R}_{0:n}$  is such that the displacement  $\mathbf{R}_{0:n} - \mathbf{Z}\mathbf{R}_{0:n}\mathbf{Z}^*$  is rank 2 and has one positive and one negative eigenvalue, where  $\mathbf{Z}$  is the (shift) matrix with 1's on the first subdiagonal and zeros elsewhere. Quasi-Toeplitz matrices are completely characterized in terms of the column vectors  $\mathbf{u}_0, \mathbf{v}_0$  in the equation

$$\mathbf{R}_{0:n} - \mathbf{Z}\mathbf{R}_{0:n}\mathbf{Z}^* = \mathbf{u}_0 \mathbf{u}_0^* - \mathbf{v}_0 \mathbf{v}_0^*.$$

We shall restrict our discussion here to so-called *admissible* quasi-Toeplitz matrices (see, e.g., [7]), which satisfy the constraint

$$\mathbf{u}_0 - \rho \mathbf{v}_0 = [1 \quad 0 \quad \dots]^T$$

where  $\rho$  is a suitable (real) scalar. Such matrices are encountered, for example, in seismic deconvolution problems, where the reflection coefficient at the topmost layer is  $\rho$  rather than  $-1$  (see, e.g., [15]). For notational simplicity we also assume that  $\mathbf{R}_{0:n}$  has been scaled so that  $r_{0,0} = 1$ .

The set of equations (1a) is still solvable for admissible quasi-Toeplitz  $\mathbf{R}_{0:n}$  by a Levinson algorithm that has the form [7]

$$\begin{pmatrix} a_n(z) \\ b_n(z) \end{pmatrix} = L_n(z) \begin{pmatrix} a_{n-1}(z) \\ b_{n-1}(z) \end{pmatrix}, \quad a_0(z) = 1, \quad b_0(z) = \rho \quad (3)$$

where  $b_n(z) \neq a_n^\#(z)$  and  $\rho$  is the scalar associated with the admissibility property ( $\rho = 1$  for Toeplitz matrices). A reduction in computational complexity is possible also for such quasi-Toeplitz matrices because replacing  $\{a_n(z), b_n(z)\}$  by the pair  $\{f_n(z), g_n(z)\}$  given by the reversible transformation

$$\begin{pmatrix} f_n(z) \\ g_n(z) \end{pmatrix} := \psi_n \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} a_n(z) \\ b_n(z) \end{pmatrix}, \quad \psi_n \neq 0 \quad (4a)$$

results in the modified Levinson recursion

$$\begin{pmatrix} f_n(z) \\ g_n(z) \end{pmatrix} = \frac{\psi_n}{2\psi_{n-1}} \begin{pmatrix} 1 - k_n & 0 \\ 0 & 1 + k_n \end{pmatrix} \begin{pmatrix} z+1 & z-1 \\ z-1 & z+1 \end{pmatrix} \begin{pmatrix} f_{n-1}(z) \\ g_{n-1}(z) \end{pmatrix}, \\ f_0(z) = \psi_0(1 + \rho), \quad g_0(z) = \psi_0(1 - \rho), \quad (4b)$$

which for a suitable choice of the scaling factors  $\psi_n$  requires only one multiplication (rather than two, as in (3)) per recursion step.

A comparison of (4b) with (3) shows that while the number of multiplications has been reduced by the transformation (4a), the number of additions has, in fact, increased. It turns out that the increase in the number of additions can be avoided by converting the *two-term* recursion (4b), which involves both  $f_n(z)$  and  $g_n(z)$  into a *three-term* recursion involving only one of these polynomials. The three-term recursion for  $f_n(z)$  has the form

$$f_{n+1}(z) = \frac{\psi_{n+1}(1 - k_{n+1})}{\psi_n(1 - k_n)} \left\{ (z+1)f_n(z) - \frac{\psi_n}{\psi_{n-1}} (1 - k_n^2) z f_{n-1}(z) \right\} \quad (5)$$

with a similar recursion for  $g_n(z)$ .

A suitable choice of the scaling factors  $\psi_n$  leaves only one nontrivial coefficient in this recursion. Since this choice can be made only in three ways, there are three computationally efficient forms of the recursion (5), namely,

$$f_{n+1}(z) = \delta_n(z+1)f_n(z) - z f_{n-1}(z) \quad (6a)$$

$$f_{n+1}(z) = (z+1)f_n(z) - \lambda_n z f_{n-1}(z) \quad (6b)$$

$$\lambda_{n+1} f_{n+1}(z) = (z+1)f_n(z) - z f_{n-1}(z). \quad (6c)$$

The form (6a) is exactly the Bistritz recursion described in [1], while (6b) is the recursion in an algorithm recently presented by Delsarte and Genin [8] for the pure Toeplitz case and dubbed by them a "split Levinson" algorithm. The same recursion has already been obtained by Bube and Burrige [17] in the context of solving the (discrete) Gopinath-Sondhi equation, which is the same as (1) but with a right side of the form  $[1 \quad \dots \quad 1]$ . Even though Delsarte and Genin were also motivated by the work of Bistritz, they do not establish the relationship between their algorithm and that of Bistritz. (The relations between these recursions and stability tests are discussed in more detail in [12].) The name "split Levinson" that they use arises from the fact that, following Bistritz, Delsarte and Genin work with the symmetric (or antisymmetric) parts of the predictor polynomials arising in the Levinson algorithm for symmetric Toeplitz matrices. This symmetry structure is not available for quasi-Toeplitz matrices, but the computational reductions can still be achieved via the transformation (4a).

The conventional formulation of the Levinson recursion (3) can be related to transmission-line models (see, e.g., [7]). In particular, the ratio  $b_n(z)/a_n(z)$  can be interpreted as the *scattering function* of a transmission line consisting of a cascade of (uniform) sections with different characteristic impedances. On the other hand, the ratio  $g_n(z)/f_n(z)$  can be interpreted as the *impedance* (or *admittance*) *function* of the same transmission line. For this reason we shall say that the original recursion (3) is expressed in the *scattering domain*, whereas the transformed recursions (4b) or (6) are expressed in the *immittance domain*.<sup>1</sup>

Finally, we note that to propagate the three-term immittance-domain recursions (6) beginning with the given covariance  $\mathbf{R}_{0:n}$ , we must also have a procedure for computing the coefficients  $\lambda_n, \delta_n$ , similar to the inner product formula (2c,d) used in the scattering-domain formulation to compute the reflection coefficients  $k_n$ . These calculations, which also require only half the

<sup>1</sup>Bode coined the term *immittance* to denote both impedance and admittance [9].

number of multiplications of those in (2c,d), are derived in Section III. We also present in Section III the relations required to reconstruct  $\{g_n(z)\}$  from the three-term recursion for  $\{f_n(z)\}$ . This makes it possible to reconstruct the predictor polynomial  $a_n(z)$  when necessary.

II. IMMITTANCE-DOMAIN RECURSIONS

It has been known for some time that the Levinson algorithm can be implemented with a single multiplier (and three adders) per iteration (see, e.g. [10]) by a suitable scaling of the polynomials  $a_n(z), b_n(z)$ . Here we consider the more general possibility of a linear transformation, viz.

$$\begin{pmatrix} f_n(z) \\ g_n(z) \end{pmatrix} := T_n \begin{pmatrix} a_n(z) \\ b_n(z) \end{pmatrix} \quad (7)$$

where  $T_n$  is any constant nonsingular  $2 \times 2$  matrix.

The choice of  $T_n$  determines the structure of a digital network that propagates  $f_n(z)$  or  $g_n(z)$ . Since we would like to keep the structure of such networks uniform, we shall restrict ourselves to transformations that differ from section to section only by a scaling factor, viz.,

$$T_n = \psi_n T, \quad \psi_n \neq 0, \det T \neq 0. \quad (8)$$

We observe that the choice of the matrix  $T$  affects both the structure (i.e., the configuration of delay elements, adders, and multipliers) and the computational complexity of the recursions, whereas the choice of  $\psi_n$  only affects the computational requirements. For instance, the choice

$$T_n = \psi_n \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \quad (9)$$

leads to the immittance-domain recursions (4b). These reduce, for a suitable choice of  $\psi_n$ , to a single-multiplier recursion, viz.,

$$\begin{pmatrix} f_n(z) \\ g_n(z) \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & \gamma_n \end{pmatrix} \begin{pmatrix} z+1 & z-1 \\ z-1 & z+1 \end{pmatrix} \begin{pmatrix} f_{n-1}(z) \\ g_{n-1}(z) \end{pmatrix} \quad (10a)$$

which requires only one multiplication and six additions per recursion step, per polynomial coefficient. Here  $\gamma_n$  is a constant defined as

$$\gamma_n := \frac{1+k_n}{1-k_n}. \quad (10b)$$

In contrast, the scattering-domain recursion (2) involves two multiplications and two additions per recursion step per polynomial coefficient.

We show in Appendix I that (9) is essentially the only transformation (7) that achieves this reduction in the number of multiplications. Every choice of  $T$  that reduces by half the number of multiplications in the Levinson recursion (3) is related to (9) by either scaling or permutations of rows and/or columns and further reduction in the number of multiplications is impossible.

Any two-term recursion for a pair of polynomials can be converted into a three-term recursion involving only one of the two polynomials in the pair. Indeed, let  $\{f_n(z), g_n(z)\}$  be a sequence of polynomial pairs satisfying the two-term recursion

$$f_n(z) = \alpha_n(z) f_{n-1}(z) + \beta_n(z) g_{n-1}(z) \quad (11a)$$

$$g_n(z) = \gamma_n(z) f_{n-1}(z) + \delta_n(z) g_{n-1}(z). \quad (11b)$$

Some simple algebra shows that

$$f_{n+1}(z) = \left\{ \alpha_{n+1}(z) + \frac{\beta_{n+1}(z) \delta_n(z)}{\beta_n(z)} \right\} f_n(z) - \frac{\beta_{n+1}(z) \Delta_n(z)}{\beta_n(z)} f_{n-1}(z) \quad (12a)$$

where

$$\Delta_n(z) := \alpha_n(z) \delta_n(z) - \gamma_n(z) \beta_n(z). \quad (12b)$$

This result is obtained by eliminating  $g_{n-1}(z)$  from (11) which leads to the auxiliary equation

$$g_n(z) = \frac{\delta_n(z) f_n(z) - \Delta_n(z) f_{n-1}(z)}{\beta_n(z)} \quad (12c)$$

which in conjunction with (11a) produces the three-term recursion (12a). The auxiliary equation is required in certain applications, e.g., when one wants to recover via (7) the polynomials  $\{a_n(z), b_n(z)\}$  from the linearly equivalent pair  $\{f_n(z), g_n(z)\}$ .

The previously introduced two-term recursions (4b) are characterized by the choices

$$\alpha_n(z) = \frac{1}{2} \frac{\psi_n}{\psi_{n-1}} (1-k_n)(z+1)$$

$$\beta_n(z) = \frac{1}{2} \frac{\psi_n}{\psi_{n-1}} (1-k_n)(z-1)$$

$$\gamma_n(z) = \frac{1}{2} \frac{\psi_n}{\psi_{n-1}} (1+k_n)(z-1)$$

$$\delta_n(z) = \frac{1}{2} \frac{\psi_n}{\psi_{n-1}} (1+k_n)(z+1).$$

Consequently,

$$\Delta_n(z) = \left( \frac{\psi_n}{\psi_{n-1}} \right)^2 (1-k_n^2) z,$$

and (12a) reduces to

$$f_{n+1}(z) = \frac{\psi_{n+1}(1-k_{n+1})}{\psi_n(1-k_n)} \left\{ (z+1) f_n(z) - \frac{\psi_n(1-k_n^2)}{\psi_{n-1}} z f_{n-1}(z) \right\} \quad (13a)$$

together with the auxiliary equation

$$(z-1) g_n(z) = \gamma_n \left\{ (z+1) f_n(z) - \frac{2\psi_n(1-k_n)}{\psi_{n-1}} z f_{n-1}(z) \right\} \quad (13b)$$

where  $\gamma_n$  is given by (10b). The three-term recursion (13a) is propagated independently of (13b), which need not be used until an explicit expression for  $g_n(z)$  is required, e.g. when the full order has been reached.

To initialize the three-term recursion for  $\{f_n(z)\}$ , one needs to know both  $f_0(z)$  and  $f_1(z)$ . The two-term recursion (4) implies that

$$f_0(z) = \psi_0(1+\rho) \quad f_1(z) = \psi_1(1-k_1)(z+\rho) \quad (14)$$

but it will be more convenient to replace the initial condition on  $f_1(z)$  by an equivalent initial condition on the (previously) undefined quantities  $f_{-1}(z), \psi_{-1}$  and  $k_0$ , which occur in (13a,b) for  $n=0$ . We will show that one consistent choice is

$$\psi_0 = (1-k_0)^{-1} \quad \psi_{-1} = 1+k_0 \quad (15a)$$

for all  $k_0 \neq 1$ . In particular, we find it convenient to choose  $k_0 = -1$  yielding

$$\psi_0 = \frac{1}{2} \quad \psi_{-1} = 0 \quad (15b)$$

and

$$z f_{-1}(z) := \frac{1}{2} (1-z)(1-\rho) \quad f_0(z) = \frac{1}{2} (1+\rho). \quad (15c)$$

Notice that for a Toeplitz system  $\rho=1$ , so that  $f_{-1}(z)=0$  and  $f_0(z)=1$ .

The computational advantage of converting the two-term recursion (4b) into the three-term form (13) becomes apparent when we consider specific choices for the scaling factors  $\psi_n$ . Suitable choices of  $\{\psi_n\}$  will result in minimum-complexity recursions with only one multiplication and two additions per recursion step, and per polynomial coefficient, in contrast to one multiplication and six additions involved in the two-term recursion (4b), and the two multiplications and two additions involved in the scattering-domain recursions (3). It is interesting to note that the *three-term scattering-domain* recursion (which is known in the Toeplitz case, see Geronimus [11, p. 132]), has the same computational requirements as the two-term scattering-domain recursion (3). Thus conversion of a two-term form into a three-term form does not by itself necessarily reduce computational complexity. We obtain a reduction by the combination of a change of variables and a three-term recursion. To obtain the minimum-complexity form of the recursions we note that there are only three choices of the scaling factors  $\psi_n$  which leave a single nontrivial coefficient in (13a).

1) *Balanced recursions*, obtained by choosing  $\psi_n$  to satisfy the constraint

$$\frac{\psi_{n+1}}{\psi_{n-1}}(1-k_{n+1})(1+k_n)=1, \quad n \geq 0 \quad (16)$$

result in

$$f_{n+1}^B(z) = \delta_n(z+1)f_n^B(z) - zf_{n-1}^B(z). \quad (17a)$$

The constraint (16) leaves  $\psi_{-1}$ ,  $\psi_0$ , and  $k_0$  unspecified. In particular, we may adopt (15b), resulting in the initial conditions (15c) and

$$\delta_0 = 1. \quad (17b)$$

The remaining  $\{\delta_n\}$  are related to each other by a recursion derived from the constraint of  $\psi_n$ , viz.,

$$\delta_n \delta_{n-1} \lambda_n = 1, \quad n \geq 1 \quad (18a)$$

where

$$\lambda_n := (1+k_n)(1-k_{n-1}), \quad n \geq 1. \quad (18b)$$

The reason for the name ‘‘balanced’’ for (15a) is that the recursions for ascending and descending indices are, essentially, identical.

2) *Monic recursions*, obtained by choosing  $\psi_n$  to satisfy the constraint

$$\psi_n(1-k_n)=1, \quad n \geq 0 \quad (19)$$

result in

$$f_{n+1}^M(z) = (z+1)f_n^M(z) - \lambda_n z f_{n-1}^M(z) \quad (20a)$$

where  $\lambda_n$  is defined by (18b). The constraint (19) (with  $n=0$ ) is consistent with (15a) and leaves only  $\psi_{-1}$  and  $k_0$  unspecified. Thus we may adopt (15b,c), which results in

$$\lambda_0 = 1. \quad (20b)$$

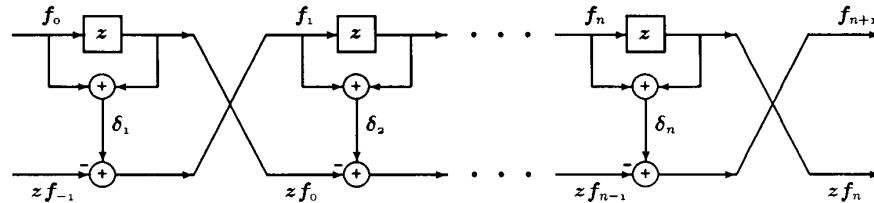


Fig. 1. Balanced recursion lattice.

3) *Dual recursions*, obtained by choosing  $\psi_n$  to satisfy the constraint

$$\frac{\psi_n}{\psi_{n-1}}(1-k_n^2)=1, \quad n \geq 0 \quad (21)$$

result in

$$\lambda_{n+1} f_{n+1}^D(z) = (z+1)f_n^D(z) - z f_{n-1}^D(z) \quad (22)$$

where  $\lambda_n$  is given by (18b). As in the monic case, the constraint (21) (with  $n=0$ ) is consistent with (15a) and leaves only  $\psi_{-1}$  and  $k_0$  unspecified. Thus we may adopt (15b,c). Here there is no need to specify  $\lambda_0$ .

Since the three polynomials  $f_n^B(z)$ ,  $f_n^M(z)$ , and  $f_n^D(z)$  are all proportional to  $a_n(z) + b_n(z)$  via different  $\psi_n$  they differ only by a scaling factor (i.e., they have the same zeros). In fact, we have

$$f_n^B(z) = \left( \prod_{i=0}^{n-1} \delta_i \right) f_n^M(z) \quad f_n^D(z) = \left( \prod_{i=1}^n \lambda_i \right) f_n^M(z). \quad (23)$$

The three recursions differ not only in the location of their nonunity coefficients, but also in the remaining flexibility available for scaling each polynomial of the sequence, or equivalently, by the freedom to choose the initial conditions. The sequence  $\{f_n^M(z)\}$  is completely determined by the requirement of being monic. Indeed, any choice of  $f_{-1}^M(z)$ ,  $f_0^M(z)$ , and  $\lambda_0$  that is admissible by the constraints (13) and (14) yields the same sequence of polynomials  $f_n^M(z)$ ,  $n \geq 1$ . The dual recursions have  $\psi_{-1}$  unspecified. Therefore, it is permissible (by (13) and (19)) to multiply the pairs  $f_{-1}^D(z)$  and  $f_0^D(z)$  in (20b) by any constant  $\epsilon \neq 0$  and, consequently, scale all  $f_n^D(z)$  by the same constant, viz.  $f_n^D(z) \rightarrow \epsilon f_n^D(z)$ . The balanced recursions have a useful extra degree of freedom. A pair of admissible initial conditions  $\{f_{-1}^B(z), f_0^B(z)\}$ , say (15b), can be replaced by another admissible pair  $\{\epsilon_1 f_{-1}^B(z), \epsilon_2 f_0^B(z)\}$  with any  $\epsilon_1, \epsilon_2 \neq 0$ , with the effect of alternatingly scaling the odd and the even degree polynomials of the sequence by  $\epsilon_1$  and  $\epsilon_2$ , respectively, viz.

$$f_{2i-1}^B(z) \rightarrow \epsilon_1 f_{2i-1}^B(z) \quad \text{and} \quad f_{2i}^B(z) \rightarrow \epsilon_2 f_{2i}^B(z), \quad i \geq 0.$$

The two degrees of freedom of the balanced recursions and their self-dual structure (the same recursion for ascending and descending degrees) make them the most useful in further applications. This is illustrated in Fig. 1 which presents a lattice realization of the balanced Levinson recursion. If the inputs  $f_0=1$  and  $z f_{-1}=0$  are replaced by  $f_{n-1}$  and  $f_n$ , the lattice of Fig. 1 produces  $\{f_k\}$  in descending degrees. Thus for appropriate choices of inputs, the same lattice implements degree reducing schemes such as the stability test of [1] or an immittance version of the Schur algorithm, as discussed in [12]. Finally, we remark that we can rewrite (23) as

$$f_n^M(z) = \left( \prod_{i=1}^{n-1} \delta_i \right)^{-1} f_n^B(z) \quad f_n^D(z) = \left( \prod_{i=0}^n \delta_i \right) f_n^B(z) \quad (24)$$

a form that further justifies the terms *dual* and *balanced*.

III. RECURSION COEFFICIENTS

To complete our immittance-domain algorithms, we have to provide formulas to compute the coefficients  $\delta_n, \lambda_n$  for the three-term recursions (6) (and  $\gamma_n$  for the two-term recursions (10)). For some applications (e.g., a solution to (1)) we also have to show how  $g_n(z)$  and  $a_n(z)$  can be recovered.

The derivation of recursive expressions for  $k_n$  in the scattering domain Levinson recursions for non-Toeplitz systems involves the concept of a generalized Schur algorithm [7], which we briefly summarize in Appendix II. The same comment also applies to computing  $\delta_n, \lambda_n$ .

We show in Appendix II that for  $n \geq 1$

$$\tau_n := [f_{n,0} \ f_{n,1} \ \cdots \ f_{n,n}] [1 \ u_{0,1} \ \cdots \ u_{0,n}]^T = \rho \psi_n R_n^e, \quad n \geq 1 \tag{25}$$

where  $\{f_{n,i}\}$  are the coefficients of  $f_n(z)$  and  $\{u_{0,i}\}$  are the elements of the column vector  $u_0$ . We also define in agreement with (15b) and the right side of (25),

$$\tau_0 := \rho \psi_0 R_0^e = \frac{\rho}{2}. \tag{26}$$

Thus the ratio of two successive  $\tau_n$  is

$$\frac{\tau_n}{\tau_{n-1}} = \frac{\psi_n}{\psi_{n-1}} (1 - k_n^2). \tag{27}$$

Substituting in (27) the constraints (16) for the balanced recursions, and (19) for the monic recursions, leads to expressions for the coefficients of these recursions given, respectively, by

$$\delta_n = \frac{\tau_{n-1}^B}{\tau_n^B} \quad \lambda_n = \frac{\tau_n^M}{\tau_{n-1}^M}. \tag{28}$$

Proceeding similarly for the dual recursions, substitution of (21) into (27) results in  $\tau_n^D = \tau_0^D$  for all  $n$ . This result expresses the fact that the determination of the right side expression in (6c),

$$h_n(z) := (z+1)f_{n-1}^D(z) - zf_{n-2}^D(z) = \sum_{i=0}^n h_{n,i} z^i \tag{29a}$$

does not require any new coefficients. If instead one takes the inner product with  $[1 \ u_{0,1} \ \cdots \ u_{0,n}]$  of both sides of (6c), the required inner product for the computation of  $\lambda_{n+1}$  in this case is found to be

$$\lambda_{n+1} = \frac{2}{\rho} \sum_{i=0}^n h_{n,i} u_{0,i}. \tag{29b}$$

The two-term recursion (10a) can also be completed to a Levinson algorithm; the choice of  $\psi_n$  that yields (10a) implies, in conjunction with (27), that  $\tau_n/\tau_{n-1} = 2(1 + k_n)$ . Consequently,

$$\gamma_n = \frac{\tau_n}{4\tau_{n-1} - \tau_n} \tag{30}$$

with the inner product  $\tau_n$  given by (25).

For the Toeplitz case  $\rho=1$  and  $b_n(z) = a_n^\#(z)$ . Thus the scattering-domain computation is reduced by half because it is only necessary to propagate the  $a_n(z)$  polynomials. In the immittance domain the fact that  $b_n(z) = a_n^\#(z)$  means that  $f_n(z) = f_n^\#(z)$  and  $g_n(z) = -g_n^\#(z)$ . Therefore, the computation (both in the two-term and the three-term recursions) is reduced by half because only half of the coefficients need to be calculated. Since the Toeplitz case, both in the scattering and the immittance domains, involves half the computation, the ratios of the computational complexities are maintained. In particular, each of the three three-term recursions requires half the number of multiplications and the same number of additions as the Levinson recursions (2). Moreover, the computation of  $\lambda_n, \delta_n$ , and  $\gamma_n$  via the formulas (25)–(30) involves inner products with *symmetric* vectors. Such a

symmetric inner product can be carried out in (approximately) half the number of multiplications needed to compute  $k_n$  via the formula (2c,d); viz. we can compute (25) as follows:

$$\tau_n = \left[ f_{n,0}, f_{n,1}, \dots, f_{n, \lfloor \frac{n}{2} \rfloor} \right] \cdot \left[ c_0 + c_n, c_1 + c_{n-1}, \dots, c_{\lfloor \frac{n}{2} \rfloor} + c_{\lfloor \frac{n+1}{2} \rfloor} \right]^T \tag{31}$$

where  $[x]$  denotes the integer part of  $x$ . Thus a complete Levinson algorithm for Toeplitz matrices based on one of the minimum-complexity recursions (6) in conjunction with the appropriate inner-product formula for the coefficients of the recursion ((28) or (29)) requires half the number of multiplications (and the same number of additions) as the conventional Levinson algorithm (2) for Toeplitz matrices. The two-term immittance-domain Levinson algorithm consisting of (10a) and (30) also requires half the number of multiplications but twice the number of additions.

After producing  $\{f_i(z); 0 \leq i \leq n\}$  in  $n$  steps of any of the three-term recursions (6) with coefficients calculated as before, it is possible to reconstruct  $g_n(z)$  for the respective last two terms  $f_n(z)$ , and  $f_{n-1}(z)$  using (13b). The reconstruction requires two coefficients. The first is found by setting  $z=1$  into (13b),

$$\mu_n := \frac{2\psi_n(1 - k_n)}{\psi_{n-1}} = \frac{2f_n(1)}{f_{n-1}(1)}. \tag{32}$$

The second coefficient is  $\gamma_n$ , which was defined in (10b) and is obtained after computing the coefficient

$$\xi_n := \frac{1 + \gamma_n}{\gamma_n} = \frac{2}{1 + k_n} \tag{33a}$$

by one of the following three formulas

$$\xi_n = \delta_n \mu_n^B = \frac{\mu_n^M}{\lambda_n} = \mu_n^D. \tag{33b}$$

Finally, the monic polynomial  $a_n(z)$  can be reconstructed using the expression

$$a_n(z) = \frac{f_n(z) + g_n(z)}{f_n(\infty) + g_n(\infty)} \tag{34}$$

where  $f_n(\infty)$  indicates the leading coefficient of the polynomial  $f_n(z)$ . The complexity of reconstructing  $a_n(z)$  is approximately  $2n$  multiplications and additions. Since the computation is carried out only once (for the highest desirable  $n$ ), this complexity is negligible in comparison to the rest of the algorithm, which involves  $O(\alpha n^2)$  computations (see Table II for specific values of  $\alpha$ ).

IV. SUMMARY AND CONCLUSIONS

We have derived some new Levinson-type algorithms that are computationally more efficient than the conventional Levinson algorithm. The immittance-domain Levinson algorithms described in this paper are summarized for convenience in Table I. The computational requirements of the new algorithms are summarized and compared to the scattering-domain algorithms in Table II.

The key to reduction in computation is really the proper use of the following two additional degrees of freedom:

- 1) the possibility of linear transformations of the variables propagated in the Levinson and Schur algorithms, and especially the transformations (well-known in circuit theory) between wave variables and immittance (voltage, current) variables;
- 2) the use of three-term recursions

TABLE I  
IMMITTANCE-DOMAIN THREE-TERM RECURSIONS FOR QUASI-TOEPLITZ COVARIANCES

Balanced	Monic	Dual
	$zf_{-1}(z) = \frac{1}{2}(1-z)(1-\rho)$	
	$f_0(z) = \frac{1}{2}(1+\rho)$	
	$\tau_0 = \frac{1}{2}\rho, \quad \delta_0 = 1 = \lambda_0$	
	<b>for <math>i = 0, 1, 2, \dots, n-1</math> do</b>	
$f_{i+1}^B(z) =$	$f_{i+1}^M(z) =$	$h_{i+1}^D(z) :=$
$\delta_i(z+1)f_i^B(z) - zf_{i-1}^B(z)$	$(z+1)f_i^M(z) - \lambda_i zf_{i-1}^M(z)$	$(z+1)f_i^D(z) - zf_{i-1}^D(z)$
$\sum_j f_{i,j}^B z^j := f_i^B(z)$	$\sum_j f_{i,j}^M z^j := f_i^M(z)$	$\sum_j h_{i,j}^D z^j := h_i^D(z)$
$\tau_i^B = \sum_j f_{i,j}^B u_{0,j}^a$	$\tau_i^M = \sum_j f_{i,j}^M u_{0,j}^a$	$\lambda_{i+1} = 2\rho^{-1} \sum_j h_{i+1,j}^D u_{0,j}^a$
$\delta_i = \tau_{i-1}^B / \tau_i^B$	$\lambda_i = \tau_i^M / \tau_{i-1}^M$	$f_{i+1}^D(z) = \lambda_{i+1}^{-1} h_{i+1}^D(z)$
	$\mu_n = 2f_n(1)/f_{n-1}(1)$	
$\xi_n^B = \delta_n \mu_n$	$\xi_n^M = \mu_n / \lambda_n$	$\xi_n^D = \mu_j n$
	$g_n(z) = \{(z+1)f_n(z) - \mu_n zf_{n-1}(z)\} / \{(z-1)(\xi_n - 1)\}$	
	$a_n(z) = \{f_n(z) + g_n(z)\} / \{f_n(\infty) + g_n(\infty)\}$	

<sup>a</sup>For Toeplitz systems we compute this inner product via the symmetric formula (31).

TABLE II  
COMPARISON OF COMPUTATION COUNTS FOR VARIOUS LEVINSON ALGORITHMS

	Quasi-Toeplitz			Toeplitz		
	Recursion	Inner Product	Total	Recursion	Inner Product	Total
<b>Scattering</b>						
Multiplications	$O(n^2)$	$O(0.5n^2)$	$O(1.5n^2)$	$O(0.5n^2)$	$O(0.5n^2)$	$O(n^2)$
Additions	$O(n^2)$	$O(0.5n^2)$	$O(1.5n^2)$	$O(0.5n^2)$	$O(0.5n^2)$	$O(n^2)$
<b>Immittance two-term</b>						
Multiplications	$O(0.5n^2)$	$O(0.5n^2)$	$O(n^2)$	$O(0.25n^2)$	$O(0.25n^2)$	$O(0.5n^2)$
Additions	$O(3n^2)$	$O(0.5n^2)$	$O(3.5n^2)$	$O(1.5n^2)$	$O(0.5n^2)$	$O(2n^2)$
<b>Immittance three-term</b>						
Multiplications	$O(0.5n^2)$	$O(0.5n^2)$	$O(n^2)$	$O(0.25n^2)$	$O(0.25n^2)$	$O(0.5n^2)$
Additions	$O(n^2)$	$O(0.5n^2)$	$O(1.5n^2)$	$O(0.5n^2)$	$O(0.5n^2)$	$O(n^2)$

This is the approach developed here and in [13]. We may note that besides enabling a simple extension from Toeplitz to quasi-Toeplitz systems, our approach has also served to delimit the whole set of efficient Levinson algorithms.

Closely related to the Levinson algorithm in the scattering domain are the Schur algorithm and the Schur-Cohn test, having in common recursion forms and lattice structures. Analogous relationships also hold in the immittance domain. For example, the lattice in Fig. 1, which realizes the balanced Levinson recursion (17), can also be viewed for appropriate inputs as the implementation of an immittance-domain Schur-Cohn test (essentially [1]), or an immittance-domain Schur algorithm. Efficient immittance-domain versions of the Schur algorithm and the Levinson algorithm for the "near-Toeplitz" cases are discussed in [12].

We should note that we have not exploited the whole family of possible fast Levinson algorithms as there are many compatible alternatives. For example, one can choose  $g_n(z)$  rather than  $f_n(z)$  as the primary polynomial and obtain three additional three-term recursions that propagate antisymmetric polynomials and require antisymmetric scalar products to compute their coefficients. One can further obtain hybrid three-term recursions to propagate polynomials that are symmetric for even  $n$  and antisymmetric for odd  $n$  (cf. [3], [4], [8]). The immittance-domain two-term recursions (10a) also have an immediate dual form in which  $\text{diag}[1, \gamma_n]$  is replaced by  $\text{diag}[\gamma_n^{-1}, 1]$  and  $\gamma_n^{-1}$  is computed as an antisymmetric scalar product. Our choice of the four fast algorithms presented here reflects the conceptual simplicity (in contrast to computational complexity) of dealing with a sequence of poly-

nomials of the same symmetry class and of using symmetric, rather than antisymmetric, polynomials and scalar products to exploit the reduced amount of computation.

Morgera and Krishna have recently derived the balanced version of the three-term immittance-domain Levinson algorithm for Toeplitz matrices with *complex* entries [14]. The same recursion has been previously derived in the context of stability testing for polynomials with complex coefficients by Delsarte and Genin [16] and by Bistritz [2]. We show in [12] how the results of our work extend to the complex case for both Toeplitz and quasi-Toeplitz matrices. The key to this extension is the selection of the linear transformation  $T_n$  in (7) as

$$T_n = \psi_n \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \eta_n \end{pmatrix} \quad (35)$$

where  $\eta_n$  is an arbitrary complex coefficient. The analysis of Section II can be carried out with this choice of  $T_n$  instead of the one given by (9), resulting in a new three-term recursion for  $f_n(z)$ . We show in [12] that the coefficients of  $f_n(z)$  and of  $f_{n-1}(z)$  in this recursion are *rational functions* of  $z$ . We also show that this additional complication can be avoided if and only if the  $\eta_n$  in (35) satisfy the recursive relation

$$\eta_n = \frac{\eta_{n-1} + k_n}{1 + k_n^* \eta_{n-1}} \quad (36)$$

where  $k_n$  are the reflection coefficients associated with the matrix  $\mathbf{R}$ . This means that the  $\{\eta_n\}$  are uniquely determined by the reflection coefficients and cannot be chosen at will. In the real

case it follows from (36) that  $\eta_n = 1$ , and (35) reduces to the linear transformation (9), which is the one used here.

#### APPENDIX I

We show in this appendix that (9) is essentially the only transformation (7) that achieves the factor two reduction in the number of multiplications. The memoryless part of  $L_n(z)$  in (2) and (3) can be transformed into

$$M_n := T \begin{pmatrix} 1 & -k_n \\ -k_n & 1 \end{pmatrix} T^{-1} = \Delta \begin{pmatrix} \Delta + k_n \Gamma & k_n(\beta^2 - \alpha^2) \\ k_n(\gamma^2 - \delta^2) & \Delta - k_n \Gamma \end{pmatrix} \quad (I.1a)$$

where

$$T = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}, \quad \Delta := \alpha\delta - \beta\gamma, \quad \Gamma := \alpha\gamma - \beta\delta. \quad (I.1b)$$

Our objective is to select  $T$  to reduce to a minimum the number of elements of  $M_n$  that depend upon  $n$  since each such element implies a multiplication of a polynomial of degree  $n-1$  by a scalar.

The off-diagonal elements of  $M_n$  can be completely eliminated by choosing  $|\alpha| = |\beta|$  and  $|\gamma| = |\delta|$ . Since  $T$  is required to be nonsingular, it follows that it must have the form

$$T = \begin{pmatrix} \alpha & -s\alpha \\ s\delta & \delta \end{pmatrix}, \quad \Delta = 2\alpha\delta, \quad \Gamma = 2s\alpha\delta \quad (I.2a)$$

where  $s = \pm 1$ . Consequently,

$$M_n = (2\alpha\delta)^2 \begin{pmatrix} 1 + sk_n & 0 \\ 0 & 1 - sk_n \end{pmatrix}. \quad (I.2b)$$

Since  $T_n = \psi_n T$ , we can absorb scaling constants into  $\psi_n$ , resulting in a single multiplier section, viz.,

$$M_n \sim \begin{pmatrix} 1 + sk_n & 0 \\ 1 - sk_n & 0 \\ 0 & 1 \end{pmatrix}. \quad (I.3)$$

This is the only form of  $T$  that leaves a single nontrivial element in  $M_n$ . If we fail to set the off-diagonal elements of  $M_n$  to zero, then the best we can do is set  $\Gamma = 0$  which results in a two-multiplier form

$$M_n \sim \begin{pmatrix} 1 & k_n \frac{\beta^2 - \alpha^2}{\Delta} \\ k_n \frac{\gamma^2 - \delta^2}{\Delta} & 1 \end{pmatrix} \quad (I.4)$$

which has the same complexity as the original scattering-domain section.

The dynamic part of  $L_n(z)$  transforms in a similar way and, for  $T$  as in (I.2), we obtain

$$T \begin{pmatrix} z & 0 \\ 0 & 1 \end{pmatrix} T^{-1} = \frac{1}{2} \begin{pmatrix} z+1 & s \frac{\alpha}{\delta} (z-1) \\ \delta & z+1 \\ s \frac{\delta}{\alpha} (z-1) & \end{pmatrix}. \quad (I.5)$$

Thus the simplest choice is  $\delta = s\alpha$ , which results, essentially, in our choice (7)-(10), leading to immittance-domain recursions. Alternatively, we could choose  $\delta = 2s\alpha$ , which would require some additional scaling by powers of 2 (i.e., shifts in a binary representation).

#### APPENDIX II

The Schur algorithm for quasi-Toeplitz complex Hermitian matrices is, according to [7],

$$G_n(z) = G_{n-1}(z) \begin{pmatrix} z & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & k_n \\ k_n^* & 1 \end{pmatrix} (1 - |k_n|^2)^{-1/2} \quad (II.1)$$

where  $G_n(z)$  is a row vector consisting of two functions (power series) in  $z$ . To get a recursion similar in form to (4) we need to convert (II.1) into a column vector format and eliminate the scaling factor  $(1 - |k_n|^2)^{-1/2}$ . For this purpose let us define power series (in  $z$ )  $u_n(z)$ ,  $v_n(z)$  as follows:

$$\begin{bmatrix} u_n(z) & -v_n(z) \end{bmatrix} := \sqrt{R_n^c} G_n(z) \quad (II.2)$$

where

$$R_n^c := \prod_{i=1}^n (1 - |k_i|^2). \quad (II.3)$$

Notice that  $[u_0(z) \quad -v_0(z)] = G_0(z)$  and, therefore, that the quasi-Toeplitz covariance associated with  $G_0(z)$ , in the sense of [7], is

$$R_{0:N} = L_N(u_0) L_N^*(u_0) - L_N(v_0) L_N^*(v_0) \quad (II.4)$$

where  $L_N(f)$  denotes the lower triangular Toeplitz matrix associated with a power series  $f(z) = \sum_{i=0}^N f_i z^i$ , namely,

$$L_N(f) := \begin{pmatrix} f_0 & & & & \\ f_1 & f_0 & & & \\ f_2 & f_1 & f_0 & & \\ \vdots & \vdots & \vdots & \ddots & \\ f_N & & & & f_0 \end{pmatrix}. \quad (II.5)$$

With this definition of  $u_0(z)$ ,  $v_0(z)$  the admissibility constraint  $G_0(z)[1 \quad \rho]^T = 1$  of [7] results in  $u_0(z) - \rho v_0(z) = 1$ , as mentioned in the introduction. Since (II.4) remains unaltered when  $v_0(z)$  is multiplied by a unit modulus scalar, we can always make  $\rho$  real and nonnegative by a suitable scaling of  $v_0(z)$ .

The recursion (II.1) can, therefore, be rearranged in the form

$$\begin{pmatrix} \tilde{u}_n(z) \\ \tilde{v}_n(z) \end{pmatrix} = \begin{pmatrix} 1 & -k_n \\ -k_n^* & 1 \end{pmatrix} \begin{pmatrix} z & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \tilde{u}_{n-1}(z) \\ \tilde{v}_{n-1}(z) \end{pmatrix} \quad (II.6)$$

where  $\tilde{u}_n(z)$  denotes conjugation of coefficients alone in the power series  $u_n(z)$ . It can be rewritten as the following nonrecursive expression

$$\begin{pmatrix} \tilde{u}_n(z) \\ \tilde{v}_n(z) \end{pmatrix} = M_n(z) \begin{pmatrix} \tilde{u}_0(z) \\ \tilde{v}_0(z) \end{pmatrix} \quad (II.7a)$$

where

$$M_n(z) := L_n(z) L_{n-1}(z) \cdots L_1(z) \quad (II.7b)$$

and  $L_n(z)$  is the matrix defined in (2a). The internal symmetry of the matrix  $L_n(z)$  induces a corresponding symmetry in the matrix function  $M_n(z)$ , viz.,

$$M_n(z) = \begin{pmatrix} \lambda_n(z) & \mu_n(z) \\ \mu_n^{\#}(z) & \lambda_n^{\#}(z) \end{pmatrix} \quad (II.8a)$$

where the sharp (#) denotes conjugate reversal of coefficients of a polynomial, namely,

$$p^{\#}(z) := z^{\deg p(z)} [p(1/z^*)]^*. \quad (II.8b)$$

We now turn to expressing (II.7) directly in terms of the coefficients of the polynomials

$$\lambda_n(z) := \sum_{i=0}^n \lambda_{n,i} z^i \quad \mu_n(z) := \sum_{i=0}^n \mu_{n,i} z^i.$$

Introduce the row vectors of coefficients

$$\lambda_n := [\lambda_{n,0} \lambda_{n,1} \cdots \lambda_{n,n}] \quad (\text{II.9a})$$

$$\mu_n := [\mu_{n,0} \mu_{n,1} \cdots \mu_{n,n}] \quad (\text{II.9b})$$

and recall (from [7]) that the first  $n$  coefficients of  $u_n(z)$ ,  $v_n(z)$  are zero. Consequently, (II.7) and (II.8) imply that

$$\lambda_n L_n^*(u_0) + \mu_n L_n^*(v_0) = R_n^c [0 \quad \cdots \quad 0 \quad 1] \quad (\text{II.10a})$$

and

$$\mu_n L_n(u_0) + \lambda_n L_n(v_0) = 0. \quad (\text{II.10b})$$

This is a set of linear equations in the unknown vectors  $\lambda_n$ ,  $\mu_n$  and its (unique) solution is

$$\lambda_n = R_n^c [0 \quad \cdots \quad 0 \quad 1] R_{0:n}^{-1} L_n(u_0) \quad (\text{II.11a})$$

$$\mu_n = -R_n^c [0 \quad \cdots \quad 0 \quad 1] R_{0:n}^{-1} L_n(v_0) \quad (\text{II.11b})$$

where  $L_n(u_0)$ ,  $L_n(v_0)$  are truncated to size  $(n+1) \times (n+1)$ , and  $R_{0:n}$  is the corresponding truncation of the covariance matrix  $R_{0:N}$ .

The Levinson algorithm propagates the polynomials  $a_n(z)$ ,  $b_n(z)$  which, by (3), are linear combinations of the elements of  $M_n(z)$ , viz.,

$$\begin{pmatrix} a_n(z) \\ b_n(z) \end{pmatrix} = M_n(z) \begin{pmatrix} 1 \\ \rho \end{pmatrix} = \begin{pmatrix} \lambda_n(z) + \rho \mu_n(z) \\ \mu_n^*(z) + \rho \lambda_n^*(z) \end{pmatrix}. \quad (\text{II.12})$$

Consequently, the corresponding row vectors of coefficients  $a_n$ ,  $b_n$  are given by the same combination of the coefficient vectors  $\lambda_n$ ,  $\mu_n$  of (II.11), namely,

$$a_n = R_n^c [1 \quad \cdots \quad 0 \quad 1] R_{0:n}^{-1} \{L_n(u_0) - \rho L_n(v_0)\} \quad (\text{II.13a})$$

$$b_n^* = R_n^c [0 \quad \cdots \quad 0 \quad 1] R_{0:n}^{-1} \{\rho L_n(u_0) - L_n(v_0)\} \quad (\text{II.13b})$$

where  $b_n^*$  denotes the row vector of coefficients of  $b_n^*(z)$ , i.e.,  $b_n^* := [b_{n,n}^* \cdots b_{n,0}^*]$ . Admissibility implies that  $L(u_0) - \rho L(v_0) = I$ , which simplifies (II.13a). In fact, we obtain the normal equation

$$a_n R_{0:n} = R_n^c [0 \quad \cdots \quad 0 \quad 1]. \quad (\text{II.14})$$

Since  $L_n(v_0)$  has zero diagonal elements, it follows from (II.14) that

$$\begin{aligned} R_{0:n} [1 \quad 0 \quad \cdots \quad 0]^T \\ = \{L_n(u_0) L_n^*(u_0) - L_n(v_0) L_n^*(v_0)\} [1 \quad 0 \quad \cdots \quad 0]^T \\ = L_n(u_0) [1 \quad 0 \quad \cdots \quad 0]^T = [1 \quad u_{0,1} \quad \cdots \quad u_{0,n}]^T \end{aligned}$$

which establishes the result

$$a_n [1 \quad u_{0,1} \quad \cdots \quad u_{0,n}]^T = 0. \quad (\text{II.15a})$$

Similarly, we observe that under admissibility (II.13b) transforms into

$$b_n^* L_n^*(u_0) = R_n^c [0 \quad \cdots \quad 0 \quad 1] R_{0:n}^{-1} \{\rho R_{0:n} - L_n(v_0)\}$$

and, consequently, that

$$b_n^* L_n^*(u_0) [0 \quad \cdots \quad 0 \quad 1]^T = R_n^c \rho.$$

Thus,

$$b_n [1 \quad u_{0,1} \quad \cdots \quad u_{0,n}]^T = R_n^c \rho \quad (\text{II.15b})$$

and a linear combination of (II.15a) and (II.15b) establishes (25).

## REFERENCES

- [1] Y. Bistritz, "Zero location with respect to the unit cycle of discrete-time linear system polynomials," *Proc. IEEE*, vol. 72, pp. 1131-1142, Sept. 1984.
- [2] Y. Bistritz, "A circular stability test for general polynomials," *Syst. Contr. Lett.*, vol. 7, pp. 89-97, Apr. 1986.
- [3] —, "Z-domain continued-fraction expansions for stable discrete systems polynomials," *IEEE Trans. Circ. Syst.*, vol. CAS-32, pp. 1162-1166, 1985.
- [4] —, "A new unit circle stability criterion," in *Proc. 1983 Int. Symp. Math. Theory Net. Syst.*, Beer-Sheva, Israel, 1983; published in *Lecture Notes in Contr. and Information Science*, vol. 58, pp. 69-87, Springer-Verlag, 1984.
- [5] E. I. Jury, *Theory and Applications of the Z-Transform Method*. New York: Wiley, 1964.
- [6] N. Levinson, "The Wiener RMS (root-mean-square) error criterion in filter design and prediction," *J. Math. Phys.*, vol. 25, no. 4, pp. 261-278, 1947.
- [7] H. Lev-Ari and T. Kailath, "Lattice-filter parameterization and modeling of nonstationary processes," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 2-16, Jan. 1984.
- [8] P. Delsarte and Y. Genin, "The split Levinson algorithm," *IEEE Trans. Acoust. Speech, Signal Proc.*, vol. ASSP-34, pp. 470-478, June 1986.
- [9] H. W. Bode, *Network Analysis and Feedback Amplifier Design*. New York: Van Nostrand, 1945.
- [10] J. Makhoul, "A class of all zero lattice digital filters; Properties and applications," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-26, pp. 304-314, 1978.
- [11] Ya. L. Geronimus, *Polynomials Orthogonal on a Circle and Interval* (translated from Russian). London: Pergamon, 1960.
- [12] Y. Bistritz, H. Lev-Ari, and T. Kailath, "Immittance domain three-term Schur and Levinson recursions for quasi-Toeplitz complex Hermitian matrices," to be published.
- [13] —, "Immittance-domain Levinson algorithms," in *Proc. 1986 IEEE Int. Conf. Acoust. Speech and Signal Processing*, Tokyo, Japan, Apr. 1986, pp. 253-256.
- [14] H. Krishna and S. D. Morgera, "The Levinson recurrence and fast algorithms for solving Toeplitz systems of Linear equations," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-35, pp. 839-848, June 1987.
- [15] E. A. Robinson and T. S. Durrani, *Geophysical Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1986.
- [16] P. Delsarte, Y. Genin, and Y. Kamp, "Application of the index theory of pseudo-lossless functions to the Bistritz stability test," *Philips J. Res.*, vol. 39, pp. 226-241, 1984.
- [17] K. P. Bube and R. Burridge, "The one-dimensional inverse problem of reflection seismology," *SIAM Rev.*, vol. 25, pp. 497-559, Oct. 1983.

## A New Method for Evaluating the Log-Likelihood Gradient, the Hessian, and the Fisher Information Matrix for Linear Dynamic Systems

MORDECHAI SEGAL AND EHUD WEINSTEIN, SENIOR MEMBER, IEEE

**Abstract**—A new method is presented for evaluating the log-likelihood gradient (score), the Hessian, and the Fisher information matrix (FIM) of the parameters of linear dynamic stochastic systems. The method incorporates the optimal smoothing equations and is, therefore, ideal for simultaneous state estimation and parameter identification.

Manuscript received May 21, 1987; revised June 13, 1988. This work was supported in part by the Office of Naval Research under Contract N00014-85-K-0272.

M. Segal is with the Department of Electronic Systems, Faculty of Engineering, Tel Aviv University, Tel Aviv, 69978, Israel.

E. Weinstein is with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 50 Vassar St., Room 36-615, Cambridge, MA 02139, on sabbatical from the Department of Electronic Systems, Faculty of Engineering, Tel Aviv University, Tel Aviv, Israel. He is also an adjunct scientist with the Department of Ocean Engineering, Woods Hole Oceanographic Institution, Woods Hole, MA 02543.

IEEE Log Number 8927887.