Transactions Letters

An Extended Berlekamp–Massey Algorithm for the Inversion of Toeplitz Matrices

Werner Henkel

Abstract— **Utilizing a new explanation [1], the Berlekamp–Massey algorithm (BMA) which solves special Toeplitz systems of linear equations is extended to an algorithm for inverting Toeplitz matrices. The usual BMA itself already leads to one row of the inverse of the corresponding Toeplitz matrix. The other rows are derived by additionally using the same operations that are central to the original BMA, too. Two alternatives for the extended BMA are presented where the first includes the usual BMA without any changes and the second simplifies the structure by some modifications also in the original BMA part. Both versions follow a tree-like structure. If the branches of the tree are implemented in parallel, the time demand would be nearly the same as for the usual BMA. In contrast to other Toeplitz algorithms, like Levinson's and Trench's, only slight modifications are to be incorporated to handle singular submatrices.**

I. Introduction

Toeplitz systems of equations arise in the manifold applications, such as the decoding of Reed–Solomon codes, linear prediction and parameter estimation. Several algorithms have been developed and refined to solve such systems of linear equation or to invert Toeplitz matrices. The most important are the ones by Levinson–Durbin [2]–[4] Berlekamp–Massey [5]–[7], and Trench [8], [9] where the second has mostly been used in decoding of Reed–Solomon, BCH and similar codes. Indeed there seems to be no application outside coding. Similarly, the other two methods were never used there. This may be reasoned by the fact, that the original algorithms by Levinson and Trench fail when singular submatrices arise. This typically occurs in the decoding of RS codes, but not only there. For this reason, especially Levinson's algorithm has been modified in various ways to bridge singularities, not realizing that Berlekamp's algorithm tolerates singularities without any changes. Some of that refinements of the Levinson algorithm should be mentioned shortly. Delsarte, Genin, and Kamp [10] developed a method for Hermitian Toeplitz systems that leads to a triangular Toeplitz system to be solved when coming across a singularity. Ciliz and Krishna [11] further specialized this method for real-symmetric Toeplitz systems applying the so called "split Levinson algorithm," a version with reduced number of multiplications. Pombra, Lev-Ari, and Kailath [12] published a modification based on a 'three term recursion' leading to some polynomial equations to be solved when a singularity is reached. Toeplitz matrix inversion from Levinsons algorithm is described by, e.g., the Gohberg–Semencul formula mentioned in [10] or the extension given in [13].

Paper approved by the Editor for Coding and Communications Theory of the IEEE Communications Society. Manuscript received December 15, 1989; revised August 10, 1990 and July 14, 1991.

The author is with the Research Institute of Deutsche Bundespost Telekom, D-6100 Darmstadt, Germany.

IEEE Log Number 9203747.

As has been stated, in the case of Berlekamp's algorithm no additional measures have to be taken to handle singular submatrices. Furthermore, its use is not restricted to problems over finite fields (Galois fields) like coding (see [14], [15]).

In this contribution an extension for the inversion of Toeplitz matrices is derived following a new description by the author published in [1]. In some respect, this explanation of the BMA discloses the structure more clearly than Massey's illustration as shift register synthesis [6]. Basically, it consists of the finding which two operations on the intermediately determined vectors are central to the method. These operations are also used to extend the algorithm to accomplish the inversion of Toeplitz matrices.

The contents of this paper may be summarized as follows. First of all, the new matrix description of the BMA has to be outlined. For further details the reader is referred to [1]. A treatment of the nonsingular case follows. The extension of the BMA for Toeplitz matrices with nonsingular minors is somewhat more regular. This is the reason for explaining this case beforehand. Two versions of the extended BMA are derived. Then the last section will show that with a minor modification of the extended BMA, singularities can be handled as well. But first, as indicated above, the main operations of the original BMA are pointed out.

II. The Main Operations of the BMA

The BMA recursively solves a Toeplitz system of the form

$$(1, C_1, C_2, \cdots, C_e) \cdot \begin{pmatrix} S_e & S_{e+1} & \cdot & \cdot & S_{2e} & \cdot & S_{M-1} \\ S_{e-1} & S_e & \ddots & & \cdot & & \cdot \\ \cdot & S_{e-1} & \ddots & \ddots & \cdot & & \cdot \\ \cdot & & \ddots & \ddots & S_{e+1} & & \cdot \\ S_0 & \cdot & \cdot & S_{e-1} & S_e & \cdot & S_{M-1-e} \end{pmatrix} = (0, \ldots, 0) \quad (1)$$

where the length of the vector $\vec{C}$ should be as small as possible.

The main operations of the method are formed simply by the adding of zeros to the left or right of intermediately resulting vectors, together with an enlargement of the corresponding Toeplitz matrix. Furthermore, the updated intermediate result is derived by combining the current vector, extended by zeros to the right, and the one before "the last length change," extended by zeros to the left.

To simplify the explanation, operators for the extensions with zeros are introduced.

$\mathcal{R}$ stands for right-hand zero extension:

$$\mathcal{R}(1, C_1, C_2, \cdots, C_l) = (1, C_1, C_2, \cdots, C_l, 0),$$

$\mathcal{L}$ stands for left-hand zero extension:

$$\mathcal{L}(1, C_1, C_2, \cdots, C_l) = (0, 1, C_1, C_2, \cdots, C_l).$$

Assuming an intermediate result to be given as

$$(1, C_1, C_2, \cdots, C_l) \cdot \begin{pmatrix} S_l & & S_{2l} \\ & \ddots & \\ S_0 & & S_l \end{pmatrix} = (\rho_0, \rho_1, \rho_2, \cdots, \rho_l),$$

the two operations combined with an enlargement of the corresponding sub-Toeplitz matrix have the following effects on the right side of the system of linear equations.

$\mathcal{R}$: The right-hand side vector is shifted to the left by one step and two new components are added at the rightmost location.

$$\mathcal{R}(1, C_1, C_2, \cdots, C_l) \cdot \begin{pmatrix} S_{l+1} & & S_{2(l+1)} \\ & \ddots & \\ S_0 & & S_{l+1} \end{pmatrix}$$
$$= (1, C_1, C_2, \cdots, C_l, 0) \cdot \begin{pmatrix} S_{l+1} & & S_{2(l+1)} \\ & \ddots & \\ S_0 & & S_{l+1} \end{pmatrix}$$
$$= (\rho_1, \rho_2, \cdots, \rho_l, \boxed{\rho_{l+1}, \rho_{l+2}}).$$

$\mathcal{L}$: The first $l + 1$ components of the right-hand side vector are left unchanged and one new component is added at the rightmost location.

$$\mathcal{L}(1, C_1, C_2, \cdots, C_l) \cdot \begin{pmatrix} S_{l+1} & & S_{2(l+1)} \\ & \ddots & \\ S_0 & & S_{l+1} \end{pmatrix}$$
$$= (0, 1, C_1, C_2, \cdots, C_l) \cdot \begin{pmatrix} S_{l+1} & & S_{2(l+1)} \\ & \ddots & \\ S_0 & & S_{l+1} \end{pmatrix}$$
$$= (\rho_0, \rho_1, \rho_2, \cdots, \rho_l, \boxed{\rho_{l+1}}).$$

These two operations together with linear combinations constitute the original BMA. For further details of the original algorithm the reader is referred to [1]. The way the operations are used will also become clear from the examples below.

For the first version of the extended algorithm in addition to the original BMA only the $\mathcal{L}$-operation together with linear combinations are needed.

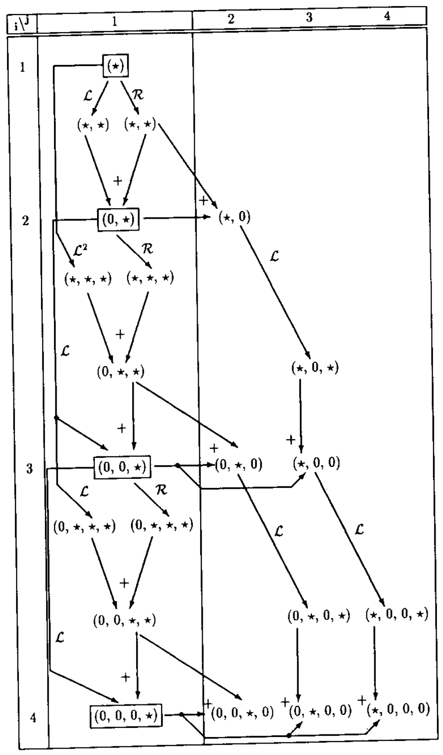

Fig. 1. The extended Berlekamp algorithm, version 1.

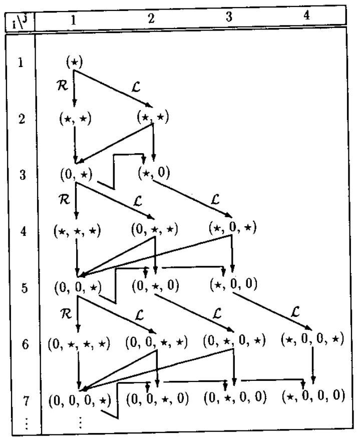

Fig. 2. The extended Berlekamp algorithm, version 2.

III. The Extended BMA for Toeplitz Inversion

For reasons of clarity, the case of *nonsingular submatrices* is treated first. Using the new matrix description given in [1], one realizes that, every second recursion, the BMA generates the last row of the inverse of the submatrix under "consideration" (see Fig. 1). To be precise, a right side is formed, with only the rightmost component being different from zero. After normalizing to that component, the aforementioned last row of the inverse follows.

The left of Fig. 1 shows the recursively appearing right sides in the usual BMA. The diagram is highly schematic and should only give a raw impression of the intermediate results of the original algorithm and the way they are used for the extension. The right side of Fig. 1—the extension—is constituted by linear combinations, characterized by arrows with a "+" and $\mathcal{L}$-operations. Stars symbolize components that may be unequal to zero. The linear combinations on the right are always used to force the rightmost component to zero, leading to unnormalized rows of the inverse of the corresponding sub-Toeplitz matrix as intermediate results. (Unnormalized rows of the identity matrix are given in the scheme). A short pseudoprogram for the extension may be given as follows (nomenclature partly based on that of [6]):

Pseudoprogram of the Extended BMA—Version 1

```
DO $i = 1, n$
    $C_1^i = C_b^i$
    $C_2^i = C_a^i - \frac{d_a^i}{d_b^i} C_b^i$
    DO $j = 3, i$
        $C_j^i = \mathcal{L}C_{j-1}^{i-1} - \frac{d_{\mathcal{L}C_{j-1}^{i-1}}}{d_b^i} C_1^i$
    END DO
END DO
```

n	Order of the whole system ($n \times n$).
i	Order of the subsystem ($i \times i$).
j	Counter for the columns in Fig.1.
C_j^i	One row vector of the $i \times i$-inverse—not normalized.
C_a^i	Intermediately resulting vector out of the BMA after the last length change.
C_b^i	Intermediately resulting vector out of the BMA before the next length change.
d_b^i	Discrepancy before the next length change.
d_a^i	Next discrepancy that would result from an unaltered C_a^i
$d_{\mathcal{L}C_{j-1}^{i-1}}$	Rightmost new component of the right side when applying $\mathcal{L}C_{j-1}^{i-1}$ to the $i \times i$-sub-Toeplitz matrix.

An example showing the inversion of a 4 × 4-matrix can be obtained from the author. For reasons of space limitations it cannot be included here.

A more regular algorithm is given in Fig. 2. There, only the two main operations of the usual BMA are used, too, but the original part of the algorithm has also been modified. Nevertheless, the structure of the left column is very similar. Only, to reduce complexity, the $\mathcal{L}$-operations of the extension have been utilized for the conventional BMA-part. Again, a pseudoprogram for the whole scheme is given:

(Note that the counter "i" has another meaning than before: it counts the rows of the scheme in Fig. 2, not directly the order of the subsystem. Again $C_{i,j}$ is the intermediate result vector and $r_{i,j}$ is the corresponding right side symbolized in Fig. 2.)

Pseudoprogram of the Extended BMA—Version 2

```
DO $i = 1, 2n - 1$
    IF $i$ mod $2 = 0$ THEN
        $C_{i,1} = \mathcal{R}C_{i-1,1}$
        DO $j = 2, i/2 + 1$
            $C_{i,j} = \mathcal{L}C_{i-1,j-1}$
        END DO
    ELSE
        $C_{i,1} = C_{i-1,1} - \frac{r_{i-1,1}(\lfloor i/2 \rfloor - 1)}{r_{i-1,3}(\lfloor i/2 \rfloor - 1)} C_{i-1,3} - \frac{r_{i-1,1}(\lfloor i/2 \rfloor)}{r_{i-1,2}(\lfloor i/2 \rfloor)} C_{i-1,2}$
        DO $j = 2, \lfloor i/2 \rfloor + 1$
            $C_{i,j} = C_{i-1,j} - \frac{r_{i-1,j}(\lfloor i/2 \rfloor + 1)}{r_{i,1}(\lfloor i/2 \rfloor + 1)} C_{i,1}$
        END DO
    END IF
END DO
```

Example 1 shows the inversion of a 4 × 4-matrix.

The structure especially of the second version of the extended BMA is seen to be comparatively simple. In the next section the occurrence of singularities will also turn out to be only slightly more complicated.

IV. A Possibility of Handling Singular Submatrices

Most Toeplitz algorithms, except the BMA and some modifications of the Levinson–Durbin algorithm, lead to difficulties when singular submatrices occur. In the sequel, it will be shown that the extended BMA in the form of Fig. 2 with only slight changes can overcome singularities.

The necessary modifications will be pointed out by examining Example 2. There, two submatrices are chosen to be singular, the 2 × 2-matrix and the 3 × 3-matrix. The first variation to be noticed is that, of course, linear combinations with C_{i1} are suppressed, if the corresponding right side equals zero (see $\star_1$) in Example 2). Anyhow, such combinations would not have altered the other right sides in that row. At the position marked with $\star_2$ other linear combinations have been performed than proposed in Fig. 2. These are determined by the structure of the appearing right sides. With the operations under $\star_2$ the singularity is overcome. Compared to the nonsingular case the rows of the 4 × 4-inverse are given in a permuted order. At $\star_3$ the arrangement is restored and from that position on the extended BMA could again proceed according to Fig. 2 if the following submatrices (of a greater matrix) are nonsingular.

Example 1:

GF(11), primitive element: 6

$$\underline{S} = \begin{pmatrix} 2 & 1 & 7 & 4 \\ 2 & 2 & 1 & 7 \\ 1 & 2 & 2 & 1 \\ 1 & 1 & 2 & 2 \end{pmatrix}, \quad \underline{S}_1 = 1, \quad \underline{S}_2 = \begin{pmatrix} 1 & 2 \\ 1 & 1 \end{pmatrix}, \quad \underline{S}_3 = \begin{pmatrix} 2 & 2 & 1 \\ 1 & 2 & 2 \\ 1 & 1 & 2 \end{pmatrix}$$

i\j	1	2	3	4
1	$(1)\cdot\underline{S}_1 = (1)$			
2	$(1,0)\cdot\underline{S}_2 = (1,2)$	$(0,1)\cdot\underline{S}_2 = (1,1)$		
	$(1,0) - \frac{1}{1}(0,1) = (1,10)$	$(0,1) - \frac{1}{1}(1,10) = (10,2)$		
3	$(1,10)\cdot\underline{S}_2 = (0,1)$	$(10,2)\cdot\underline{S}_2 = (1,0)$		
4	$(1,10,0)\cdot\underline{S}_3 = (1,0,10)$	$(0,1,10)\cdot\underline{S}_3 = (0,1,0)$	$(0,10,2)\cdot\underline{S}_3 = (1,0,2)$	
	$(1,10,0) - \frac{1}{1}(0,10,2)$ $-\frac{0}{1}(0,1,10) = (1,0,9)$	$(0,1,10) - \frac{0}{8}(1,0,9) =$ $= (0,1,10)$	$(0,10,2) - \frac{2}{8}(1,0,9) =$ $= (8,10,8)$	
5	$(1,0,9)\cdot\underline{S}_3 = (0,0,8)$	$(0,1,10)\cdot\underline{S}_3 = (0,1,0)$	$(8,10,8)\cdot\underline{S}_3 = (1,0,0)$	
6	$(1,0,9,0)\cdot\underline{S} = (0,8,3,2)$	$(0,1,0,9)\cdot\underline{S} = (0,0,8,3)$	$(0,0,1,10)\cdot\underline{S} = (0,1,0,10)$	$(0,8,10,8)\cdot\underline{S} = (1,0,0,5)$
	$(1,0,9,0) - \frac{8}{1}(0,0,1,10)$ $-\frac{3}{8}(0,1,0,9) = (1,1,1,6)$	$(0,1,0,9) - \frac{3}{2}(1,1,1,6) =$ $= (4,5,4,0)$	$(0,0,1,10) - \frac{10}{2}(1,1,1,6) =$ $= (6,6,7,2)$	$(0,8,10,8) - \frac{5}{2}(1,1,1,6) =$ $= (3,0,2,4)$
7	$(1,1,1,6)\cdot\underline{S} = (0,0,0,2)$	$(4,5,4,0)\cdot\underline{S} = (0,0,8,0)$	$(6,6,7,2)\cdot\underline{S} = (0,1,0,0)$	$(3,0,2,4)\cdot\underline{S} = (1,0,0,0)$

$$\Longrightarrow \underline{S}^{-1} = \begin{pmatrix} 3 & 0 & 2 & 4 \\ 6 & 6 & 7 & 2 \\ 4/8 & 5/8 & 4/8 & 0/8 \\ 1/2 & 1/2 & 1/2 & 6/2 \end{pmatrix}$$

Example 1.

Example 2:

GF(11), primitive element: 6

$$\underline{S} = \underline{S}_5 = \begin{pmatrix} 2 & 1 & 8 & 1 & 1 \\ 7 & 2 & 1 & 8 & 1 \\ 4 & 7 & 2 & 1 & 8 \\ 7 & 4 & 7 & 2 & 1 \\ 4 & 7 & 4 & 7 & 2 \end{pmatrix}, \quad \underline{S}_1 = 4, \quad \underline{S}_2 = \begin{pmatrix} 7 & 4 \\ 4 & 7 \end{pmatrix}, \quad \underline{S}_3 = \begin{pmatrix} 4 & 7 & 2 \\ 7 & 4 & 7 \\ 4 & 7 & 4 \end{pmatrix}, \quad \underline{S}_4 = \begin{pmatrix} 7 & 2 & 1 & 8 \\ 4 & 7 & 2 & 1 \\ 7 & 4 & 7 & 2 \\ 4 & 7 & 4 & 7 \end{pmatrix}$$

i\j	1	2	3	4	5
1	$(1)\cdot\underline{S}_1 = (4)$				
2	$(1,0)\cdot\underline{S}_2 = (7,4)$	$(0,1)\cdot\underline{S}_2 = (4,7)$			
	$(1,0) - \frac{7}{4}(0,1) = (1,1)$	$- - \star_1 - -$			
3	$(1,1)\cdot\underline{S}_2 = (0,0)$				
4	$(1,1,0)\cdot\underline{S}_3 = (0,0,9)$	$(0,1,1)\cdot\underline{S}_3 = (0,0,0)$	$(0,0,1)\cdot\underline{S}_3 = (4,7,4)$		
			$(0,0,1) - \frac{4}{9}(1,1,0)$ $= (2,2,1)$		
5			$(2,2,1)\cdot\underline{S}_3 = (4,7,0)$		
6	$(1,1,0,0)\cdot\underline{S}_4 = (0,9,3,9)$	$(0,1,1,0)\cdot\underline{S}_4 = (0,0,9,3)$	$(0,0,1,1)\cdot\underline{S}_4 = (0,0,0,9)$	$(0,2,2,1)\cdot\underline{S}_4 = (4,7,0,2)$	
$\star_2$	$(1,1,0,0) - \frac{3}{9}(0,1,8,7)$ $-\frac{9}{9}(0,0,1,1)$ $= 1,8,0,4)$	$(0,1,1,0) - \frac{3}{9}(0,0,1,1)$ $= (0,1,8,7)$		$(0,2,2,1) - \frac{7}{9}(1,8,0,4)$ $-\frac{2}{9}(0,0,1,1)$ $= (9,8,3,5)$	
7	$(1,8,0,4)\cdot\underline{S}_4 = (0,9,0,0)$	$(0,1,8,7)\cdot\underline{S}_4 = (0,0,9,0)$	$(0,0,1,1)\cdot\underline{S}_4 = (0,0,0,9)$	$(9,8,3,5)\cdot\underline{S}_4 = (4,0,0,0)$	
8	$(1,8,0,4,0)\cdot\underline{S}_5 = (9,0,0,7,2)$	$(0,1,8,0,4)\cdot\underline{S}_5 = (0,9,0,0,7)$	$(0,0,1,8,7)\cdot\underline{S}_5 = (0,0,9,0,8)$	$(0,0,0,1,1)\cdot\underline{S}_5 = (0,0,0,9,3)$	$(0,9,8,3,5)\cdot\underline{S}_5 = (4,0,0,0,9)$
	$(1,8,0,4,0) - \frac{7}{9}(0,0,0,1,1)$ $-\frac{9}{4}(0,9,8,3,5)$ $= (1,7,4,9,6)$	$(0,1,8,0,4) - \frac{7}{6}(1,7,4,9,6)$ $= (8,2,7,6,8)$	$(0,0,1,8,7) - \frac{8}{6}(1,7,4,9,6)$ $= (6,9,3,7,10)$	$(0,0,0,1,1) - \frac{3}{6}(1,7,4,9,6)$ $= (5,2,9,2,9)$	$(0,9,8,3,5) - \frac{9}{6}(1,7,4,9,6)$ $= (4,4,2,6,7)$
9	$(1,7,4,9,6)\cdot\underline{S}_5 = (0,0,0,0,6)$	$(8,2,7,6,8)\cdot\underline{S}_5 = (0,9,0,0,0)$	$(6,9,3,7,10)\cdot\underline{S}_5 = (0,0,9,0,0)$	$(5,2,9,2,9)\cdot\underline{S}_5 = (0,0,0,9,0)$	$(4,4,2,6,7)\cdot\underline{S}_5 = (4,0,0,0,0)$
$\star_3$					
10	$(1,7,4,9,6)\cdot\underline{S}_5 = (0,0,0,0,6)$	$(5,2,9,2,9)\cdot\underline{S}_5 = (0,0,0,9,0)$	$(6,9,3,7,10)\cdot\underline{S}_5 = (0,0,9,0,0)$	$(8,2,7,6,8)\cdot\underline{S}_5 = (0,9,0,0,0)$	$(4,4,2,6,7)\cdot\underline{S}_5 = (4,0,0,0,0)$

$$\boxed{\Longrightarrow \underline{S}^{-1} = \begin{pmatrix} 4/4 & 4/4 & 2/4 & 6/4 & 7/4 \\ 8/4 & 2/4 & 7/4 & 6/4 & 8/4 \\ 6/9 & 9/9 & 3/9 & 7/9 & 10/9 \\ 5/9 & 2/9 & 9/9 & 2/9 & 9/9 \\ 1/6 & 7/6 & 4/6 & 9/6 & 6/6 \end{pmatrix}}$$

Example 2.

Thus, linear combinations during a singularity are performed to generate as much zeros on the right side as possible, as long as they do not lead to linear dependencies. All-zero right sides are not used for combinations. After the singularity has been passed a permutation may be performed to restore the arrangement of appearing rows of the intermediate and final inverses.

Thus, it could be concluded that singularities lead to a certain irregularity in the algorithm, but they will not cause a failure.

REFERENCES

[1] W. Henkel, "Another description of the Berlekamp–Massey algorithm," *IEE Proc. Part I,* vol. 136, no. 3, June 1989.

[2] N. Levinson, "The Wiener RMS (Root Mean Square) error criterion in filter design and prediction," *J. Mathemat. Phys.,* vol. 25, no. 4, pp. 261–278, Jan. 1947. Reprinted in: N. Wiener, *Extrapolation, Interpolation and Smoothing of Stationary Time Series, with Engineering Applications.* New York: Technology Press and Wiley, 1949.

[3] J. Durbin, "The fitting of time-series models," *Rev. Int. Statist. Inst.,* vol. 28, pp. 233–244, 1960.

[4] T. Kailath, "A view of three decades of linear filtering theory," *IEEE Trans. Inform. Theory,* vol. IT-20, pp. 146–181, Mar. 1974.

[5] E. R. Berlekamp, *Algebraic Coding Theory.* New York: McGraw Hill, 1968.

[6] J. L. Massey, "Shift-register synthesis and BCH decoding," *IEEE Trans. Inform. Theory,* vol. IT-15, pp. 122–127, Jan. 1969.

[7] R. E. Blahut, *Theory and Practice of Error Control Codes.* Reading, MA: Addison-Wesley, 1984.

[8] W. F. Trench, "An algorithm for the inversion of finite Toeplitz matrices," *J. SIAM,* vol. 12, no. 3, pp. 512–522, 1964.

[9] R. E. Blahut, *Fast Algorithms for Digital Signal Processing. Reading, MA: Addison-Wesley, 1985.*

[10] P. Delsarte, Y. V. Genin, and Y. G. Kamp, "A generalization of the Levinson algorithm for Hermitian Toeplitz matrices with any rank profile," *IEEE Trans. Acoust. Speech, Signal Processing,* vol. ASSP-33, pp. 964–971, Aug. 1985.

[11] M. K. Ciliz and H. Krishna, "Split Levinson algorithm for Toeplitz matrices with singular sub-matrices," *IEEE Trans. Circuits Syst.*, vol. CAS-36, pp. 922–924, June 1989.

[12] S. Pombra, H. Lev-Ari, and T. Kailath, "Levinson and Schur algorithms for Toeplitz matrices with singular minors," *1988 Int. Conf. Acoust., Speech, Signal Processing (ICASSP 88),* New York, Apr. 11–14, 1988, pp. 1643–1646.

[13] W. Henkel, "An extension of the Levinson-Durbin algorithm for the inversion of Toeplitz matrices," *Archiv für Elektronik und Uebertragungstechnik,* vol. 44, no. 5, Sept./Oct. 1990.

[14] ——, "Multiple error correction with analog codes," in *6th Int. Conf. Appl. Algebra, Algebraic Algorithms Error Correcting Codes (AAECC-6),* Rome, July 4–8, 1988, published as Springer Lect. Notes Comput. Sci., vol. 357, pp. 239–249, 1989.

[15] ——, "Zur Decodierung algebraischer Blockcodes über komplexen Alphabeten," Ph.D. dissertation, VDI-Verlag, Fortschrittsberichte, Series 10, no. 109, Düsseldorf, 1989.

[16] K. Imamura and W. Yoshida, "A simple derivation of the Berlekamp–Massey algorithm and some applications," *IEEE Trans. Inform. Theory,* vol. IT-33, pp. 146–150, Jan. 1987.

[17] E. Jonckheere and C. Ma, "A simple Hankel interpretation of the Berlekamp–Massey algorithm," *Lin. Algorithm Appl.,* vol. 125, pp. 65–76, 1989.