Fast Direct Methods for Toeplitz Least Squares Problems

Michael K. Ng
Computer Sciences Laboratory
Research School of Information Sciences & Engineering
The Australian National University, Australia
E-mail: mng@cslab.anu.edu.au

ABSTRACT: *Least squares estimations have been used extensively in many applications, e.g. system identification and signal prediction. In these applications, the least squares estimators can usually be found by solving Toeplitz least squares problems. In this paper, we present fast algorithms for solving the Toeplitz least squares problems. The algorithm is derived by using the displacement representation of the normal equations matrix. Numerical experiments show that these algorithms are efficient.*

1. INTRODUCTION

In signal processing, system identification and image processing applications, one encounters various forms of structured matrices. An m-by-n matrix $T_{m,n}$ is said to be *Toeplitz* if

$$T_{m,n} = \begin{bmatrix} t_0 & t_{-1} & \cdots & t_{2-n} & t_{1-n} \\ t_1 & t_0 & t_{-1} & & t_{2-n} \\ \vdots & t_1 & t_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ t_{m-2} & & \ddots & \ddots & t_{m-n-1} \\ t_{m-1} & t_{m-2} & \cdots & t_{m-n+1} & t_{m-n} \end{bmatrix}, \quad (1)$$

i.e., $T_{m,n}$ is constant along its diagonals. In this paper, we are interested in solving Toeplitz least squares problems.

Toeplitz least squares problems arise in a variety of applications in signal processing. In these applications, one usually uses filters to estimate the transmitted signal from a sequence of received signal samples or to model an unknown system. It has been studied in [9] that given $m+n-1$ data samples and desired response vector d with length m $(m \geq n)$, the filter w with n filter coefficients can be found by solving the Toeplitz least squares problem:

$$\min_w \|d - T_{m,n}w\|_2. \quad (2)$$

Here $\|\cdot\|$ denotes the usual Euclidean norm.

Fast algorithms for solving Toeplitz least squares problems have been developed by Bojanczyk rm et al. [2], Chun et al. [5] and Sweet [13] that solve (2) in $O(mn)$ operations as opposed to $O(mn^2)$ operations required for general dense least squares problems. The main aim of this paper is to present a new fast algorithm for solving the Toeplitz least squares problems where the rectangular Toeplitz matrix has full column rank. Our procedure is first to obtain the displacement representation of the n-by-n normal equations matrix of the Toeplitz least squares problems. Then we transform the normal equation matrix into a special structured matrix (Cauchy-like matrix) via fast Fourier transforms (FFTs) and solves the resulting n-by-n Cauchy-like linear system using Gaussian Elimination with pivoting technique. The Cauchy-like linear system can be solved in $O(n^2)$ operations via forward and back-substitution. Hence the solution of the Toeplitz least squares problems can be solved in $O(n^2 + m\log n)$ operations. The paper is organized as follows. In §2, we review some definitions and results on displacement representations of Toeplitz matrices. In §3, we consider the fast Gaussian Elimination solver for Cauchy-like linear system and then present our fast algorithm. Finally, some numerical results are given in §4.

2. DISPLACEMENT STRUCTURE

In this section we briefly review relevant definitions and results on displacement structure representation of a matrix. The *Stein type displacement* equation for a matrix $A_n \in \mathcal{C}^{n\times n}$ is

$$A_n - \Omega_n A_n \Lambda_n = B_{n,\alpha} C_{\alpha,n}, \quad (3)$$

where $\Omega_n, \Lambda_n \in \mathcal{C}^{n\times n}$, $B_{n,\alpha} \in \mathcal{C}^{n\times\alpha}$ and $C_{\alpha,n} \in \mathcal{C}^{\alpha\times n}$. The pair of matrices $B_{n,\alpha}$ and $C_{\alpha,n}$ is called the *generator* of A_n with respect to Ω_n and

0-7803-3679-8/96/$5.00 © 1996 IEEE

Λ_n. The matrix A_n is considered to possess a *displacement structure* with respect to Ω_n and Λ_n if $n \gg \alpha$. The scalar α is called *displacement rank* of A_n with respect to Ω_n and Λ_n. The advantage of using displacement representation of is that all the information about n^2 entries of the matrix A_n is efficiently stored in $2\alpha n$ entries of $B_{n,\alpha}$ and $C_{\alpha,n}$. The concept of displacement structure was first introduced in Kailath et al.[10].

Let us introduce the n-by-n the lower shift circulant matrices Z_n

$$Z_n = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & & & 0 \\ 0 & 1 & & & \vdots \\ \vdots & & \vdots & & \vdots \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix},$$

and consider n-by-n Hermitian Toeplitz matrix T_n as an example, i.e., $m = n$ and $t_{-j} = \overline{t_j}$ in (1). It is easy to check that

$$T_n - Z_n T_n Z_n^* =$$

$$\begin{bmatrix} \frac{1}{2} & \frac{-1}{2} \\ t_1 - \overline{t_{n-1}} & t_1 - \overline{t_{n-1}} \\ \vdots & \vdots \\ t_{n-1} - \overline{t_1} & t_{n-1} - \overline{t_1} \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \cdot$$

$$\begin{bmatrix} \frac{1}{2} & \overline{t_1} - t_{n-1} & \cdots & \overline{t_{n-1}} - t_1 \\ \frac{-1}{2} & \overline{t_1} - t_{n-1} & \cdots & \overline{t_{n-1}} - t_1 \end{bmatrix}^T. \quad (4)$$

We see that the displacement rank of T_n is 2 in general. We remark that because of the choice of the displacement operator Z_n, only part of the information on T_n is contained in its generator on the right-hand side of (4), see [11] for details. In order to have all the information about the n^2 entries of the Toeplitz matrx T_n, we need to use the optimal Frobenius-norm circulant approximation for T_n to generate the entries of T_n. More precisely,

$$T_n = C_n + \tilde{C}_n,$$

where C_n is the minimizer of $||T_n - Q_n||_F$ over all n-by-n circulant matrices Q_n and the entries of the matrix $\tilde{C}_n$ can be generated from the generator of T_n on the right-hand size of (4). Since any circulant matrix can be characterized by its first column, we only need to construct the first column of C_n. It has been shown in [4] that the first column of C_n can be constructed in $O(n)$ operations.

In this paper, we are interested in solving the Toeplitz least squares problems stated in (2). We note that $T_{m,n}^* T_{m,n}$ is in general not a Toeplitz matrix. However, using the displacement structure of $T_{m,n}$, we can write down the displacement equation of $T_{m,n}^* T_{m,n}$ with respect to the displacement operator Z_n:

$$T_{m,n}^* T_{m,n} - Z_n T_{m,n}^* T_{m,n} Z_n^* = E_{n,6} J_6 E_{n,6}^*, \quad (5)$$

where the matrix $E_{n,6}$ is given by

$$E_{n,6} = \left[\begin{matrix} 0 & 0 & \sqrt{u_0} & \sqrt{u_1} \\ \overline{t_{-1}} & \overline{t_{m-1}} & 0 & 0 \\ \overline{t_{-2}} & \overline{t_{m-2}} & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \overline{t_{2-n}} & \overline{t_{m-n}} & 0 & 0 \\ \overline{t_{1-n}} & \overline{t_{m-n+1}} & 0 & 0 \end{matrix} \right.$$

$$\left. \begin{matrix} \frac{1}{2} & -\frac{1}{2} \\ \overline{\ell_1} - \ell_{n-1} & \overline{\ell_1} - \ell_{n-1} \\ \overline{\ell_2} - \ell_{n-2} & \overline{\ell_2} - \ell_{n-2} \\ \vdots & \vdots \\ \overline{\ell_{n-2}} - \ell_2 & \overline{\ell_{n-2}} - \ell_2 \\ \overline{\ell_{n-1}} - \ell_1 & \overline{\ell_{n-1}} - \ell_1 \end{matrix} \right],$$

and the matrix J_6 is given by

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}.$$

Here the numbers u_0, u_1 and ℓ_j are given as follows:

$$u_0 = \sum_{s=m-n+1}^{m-1} t_s \overline{t_s}, \quad (6)$$

$$u_1 = \sum_{s=-1}^{n-1} t_s \overline{t_s} \quad (7)$$

and

$$\ell_j = \sum_{s=0}^{m-1} t_{s-j} \overline{t_s}, \quad 1 \leq j \leq n-1. \quad (8)$$

We note from (6), (7) and (8) that the numbers u_0, u_1 and ℓ_i are just the convolutions of $\{t_i\}$ and therefore they can be computed efficiently using FFTs.

As was mentioned in case of Toeplitz matrix, similarly, we cannot recover the entries of $T_{m,n}^* T_{m,n}$ just using its generator on the right-hand size of (5). In this case, the normal equation matrix $T_{m,n}^* T_{m,n}$ can be decomposed into two parts, i.e.,

$$T_{m,n}^* T_{m,n} = X_n + \tilde{X}_n, \quad (9)$$

where X_n is the minimizer of $||T_{m,n}^* T_{m,n} - Q_n||_F$ over all n-by-n circulant matrices Q_n and the entries of the matrix $\tilde{X}_n$ can be generated from the generator of $T_{m,n}^* T_{m,n}$. In [3], Chan et al. proved that the circulant matrix X_n can be generated in $O(n \log n)$ operations. In the next section, we make use of (5) and (9) to derive a fast algorithm to solve the Toeplitz least squares problems.

3. FAST ALGORITHMS

Our method is based on a fast algorithm for solving a special structured matrix system proposed by Gohberg et al. [6] and Kailath et al. [11]. We first consider a special class of structured Hermitian matrices R_n. We choose the displacement operators

$$\Omega = \Lambda = D_n = \text{ diag } (d_1, d_2, \cdots, d_n)$$

in (3). Since R_n is Hermitian, we have

$$R_n - D_n R_n D_n^* = B_{n,\alpha} \tilde{J}_\alpha B_{n,\alpha}^*, \qquad (10)$$

where $B_{n,\alpha}$ is an n-by-α matrix and $\tilde{J}_\alpha$ is an α-by-α matrix. A matrix with low displacement rank ($n \gg \alpha$) is called a Cauchy-like matrix. It has been shown in [11] that the diagonal entries of the Cauchy-like matrix cannot be recovered from its generator $B_{n,\alpha}$. More precisely, a Cauchy-like matrix R_n, that satisfies (10), is decomposed as

$$R_n = S_n + \tilde{S}_n,$$

where S_n is a diagonal matrix and $\tilde{S}_n$ can be generated from its generator $B_{n,\alpha}$.

In [11], Kailath and Olshevsky developed an explicit fast block Gaussian Elimination with diagonal pivoting procedure for factorizing Cauchy-like matrix R_n. The step of block Gaussian Elimination is based on recursive applying of the well-known Schur complementation formula:

$$R_n = \begin{bmatrix} R_{11} & R_{21}^* \\ R_{21} & R_{22} \end{bmatrix} =$$

$$\begin{bmatrix} I & 0 \\ R_{21}R_{11}^{-1} & I \end{bmatrix} \cdot \begin{bmatrix} R_{11} & 0 \\ 0 & R_{22} - R_{21}R_{11}^{-1}R_{21}^* \end{bmatrix} \cdot \begin{bmatrix} I & R_{11}^{-1}R_{21}^* \\ 0 & I \end{bmatrix}. \qquad (11)$$

The displacement structure of the Cauchy-like matrix, that satisfies (3) with displacement operators D_n, is inherited by its Schur complement. This facts allows one to avoid expensive computing $(n-1)^2$ entries of the Schur complement in (11), and to compute instead only of its generator $B_{n,\alpha}$. In particular, Kailath and Olshevsky [11] proved the following result.

Theorem 1 Let R_n satisfy (10). Let the matrices D_n and $B_{n,\alpha}$ in (10) be partitioned as

$$D_n = \begin{bmatrix} D_{11} & 0 \\ 0 & D_{22} \end{bmatrix} \quad \text{and} \quad B_{n,\alpha} = \begin{bmatrix} B_{11} \\ B_{21} \end{bmatrix}.$$

If the upper left block R_{11} is invertible, then the Schur complement $R^{(2)} = R_{22} - R_{21}R_{11}^{-1}R_{21}^*$ satisfies

$$R^{(2)} - D_{22}R^{(2)}D_{22}^* = B^{(2)}\tilde{J}_\alpha(B^{(2)})^*,$$

with

$$B^{(2)} = B_{21} - ((\tau I - D_{22})R_{21}R_{11}^{-1})(\tau I - D_{11})^{-1}B_{11}, \qquad (12)$$

where τ is any number on the unit circle, which is not an eigenvalue of D_{11}.

We avoid operations on the n^2 entries of a Cauchy-like matrix and only manipulate on the entries of its generator. Since the matrix R_n is just Hermitian and is not positive definite, the numerical stability of the Gaussian Elimination may not be achieved using only scalar elimination steps. Sometimes, one has to perform the step with 2-by-2 block R_{11}. The details can be found in [7]. To enhance the accuracy of factorization, Kailath abd Olshevsky [11] also proposed to use the diagonal pivoting in the block Gaussian Elimination. By multiplying the displacement equation with a permutation P_n, one immediately sees that

$$P_n R_n P_n^* - P_n D_n P_n^* P_n R_n P_n^* P_n D_n^* P_n^* =$$

$$P_n B_{n,\alpha} \tilde{J}_\alpha B_{n,\alpha}^* P_n.$$

We note that $P_n R_n P_n$ and $P_n D_n P_n$ are Cauchy-like and diagonal matrices respectively. If the permutation P_n is known, then it is easy to incorporate into the algorithm of the factorization of the Cauchy-like matrix, see [11].

Cholesky Factorization Algorithm of Cauchy-like Matrix

Step 1: If $n = 0$, then stop.

Step 2: The size ν of the pivot R_{11} is chosen to be 1 or 2 to enhance the stability of the algorithm. Perform diagonal pivoting by choosing a suitable permutation matrix P_n.

Step 3: The nondiagonal entries of the first column of R_n are given by the formula

$$r_{ij} = \frac{b_i \tilde{J}_\alpha b_j^*}{1 - d_i d_j^*},$$

where b_i is the ith row of $B_{n,\alpha}$. The diagonal entries of R_n are stored in the diagonal of S_n.

Step 4: The matrix $B^{(2)}$ (the generator of the Schur complement $R^{(2)}$) is computed by the formula (12).

Step 5: The diagonal part of $R^{(2)}$ is obtained by computing only the diagonal entries in Schur complement.

Step 6: Set $n = n - \nu$ and repeat the steps 1–6 for the Schur complement $R^{(2)}$ given by its generator and diagonal entries.

As was mentioned above, Cauchy-like matrices allow the incorporation of pivoting techniques into fast algorithms, so they are attractive from the viewpoint of the stability. In particular, Gohberg et al. [6] and Kailath et al. [11] transformed Toeplitz matrix into Cauchy-like matrix and make use of the above procedure to develop a fast and stable Toeplitz system solver.

In this paper, we apply the previous fast algorithm to solve the Toeplitz least squares problems. Our idea is first to transform the normal equations matrix $T_{m,n}^* T_{m,n}$ into Cauchy-like matrix by the discrete Fourier transforms to the generator of $T_{m,n}^* T_{m,n}$ (cf. (5)).

Lemma 1 Given the rectangular Toeplitz matrix $T_{m,n}$ as in (1). Then $F_n T_{m,n}^* T_{m,n} F_n^*$ is a Cauchy-like matrix, satisfying

$$F_n T_{m,n}^* T_{m,n} F_n^* - D_n F_n T_{m,n}^* T_{m,n} F_n^* D_n^* = B_{n,6} J_6 B_{n,6}^*, \tag{13}$$

where F_n is the n-by-n discrete Fourier transform matrix

$$[F_n]_{jk} = \frac{1}{\sqrt{n}} e^{2\pi i (j-1)(k-1)/n},$$

$$D_n = F_n^* Z_n F_n = \operatorname{diag}(1, e^{2\pi i/n}, \cdots, e^{2\pi i(n-1)/n}),$$

and

$$B_{n,6} = F E_{n,6}. \tag{14}$$

Moreover, the diagonal entries of S_n is given by

$$[S_n]_{jj} = \sqrt{n} F_n x, \tag{15}$$

where x is the first column of the circulant matrix X_n as in (9).

Since the rectangular Toeplitz matrix has full rank, the corresponding normal equations matrix $T_{m,n}^* T_{m,n}$ is Hermitian positive definite and therefore the corresponding Cauchy-like matrix $F_n T_{m,n}^* T_{m,n} F_n^*$ is also Hermitian positive definite. It leads us to use the size of pivot to be 1 in the Cholesky factorization of the Cauchy-like matrix.

Fast Algorithm of Solving Toeplitz Least Squares Problems $\|d - T_{m,n} w\|_2^2$

Step	Sequential	Parallel
1	$O(kn \log n)$	$O(k \log n)$
2	$O(n \log n)$	$O(\log n)$
3	$On \log n)$	$O(\log n)$
4	$O(n^2)$	$O(n)$
5	$O(n^2)$	$O(n)$

Table 1: Computation cost of the proposed algorithm with $m = kn$.

Step 1: Compute the first column of the circulant matrix X_n using the algorithm stated in [3].

Step 2: Compute the generator $B_{n,6}$ of the Cauchy-like matrix $R_n = F_n T_{m,n}^* T_{m,n} F_n$ (cf. (14)) and the diagonal matrix S_n (cf. (15)) using FFTs.

Step 3: Perform the Cholesky factorization algorithm of the Cauchy-like matrix R_n and obtain the Cholesky factorization

$$R_n = F_n T_{m,n}^* T_{m,n} F_n^* = P_n^* L_n L_n^* P_n,$$

where L_n is a lower triangular matrix.

Step 4: The solution w is given by

$$w = F_n P_n (L_n^*)^{-1} (L_n)^{-1} P_n^* F_n^* \tilde{d}.$$

Table 1 below lists the computation cost in each step of the above algorithm. The basic tool of our algorithm is the FFT. Since the FFT is highly parallelizable and has been implemented on multiprocessors efficiently [1]. Our algorithm can be expected to perform efficiently in a parallel environment for large-scale applications. In Table 1, we assume that we have n processors for doing FFTs on an n-vector in $O(\log n)$ operations.

We remark that Gu [8] recently proposed using fast Gaussian Elimination algorithm for Cauchy-like linear system to solve the Toeplitz least squares problems with full column rank. However, our approach is different from that proposed by Gu. Gu first transforms the rectangular Toeplitz matrix to the rectangular Cauchy-like matrix. Then the Cauchy-like least squares problem is reduced into two Cauchy-like linear systems and these Cauchy-like linear systems are solved by the fast Gaussian Elimination algorithm. In our case, we consider the displacement structure of the normal equations matrix $T_{m,n}^* T_{m,n}$ and transform it to the Cauchy-like matrix. The motivation behind our procedure is that we only solve one Cauchy-like linear system and the transformed Cauchy-like matrix is Hermitian positive definite as the rectangular Toeplitz matrix $T_{m,n}$ has full rank.

4. NUMERICAL RESULTS

We performed a computer experiment with the algorithm designed in the present paper to investigate its performance. We illustrate the performance of the method by using finite impulse response (FIR) system identification as an example. FIR system identification has wide applications in engineering [12]. Figure 1 is a block diagram of an FIR system identification model. The input signal x_k drives the unknown system to produce the output sequence y_k. We model the unknown system as an FIR filter. If the unknown system is actually an FIR system, then the model is exact. In the tests, we formulate a well-defined least squares prediction problem by estimating the autocovariances from the data samples. By solving the normal equations, the FIR system coefficients can be found. The rectangular Toeplitz matrices $T_{m,n}$ in (1) we used are

1. t_k is randomly chosen from the normal distribution with zero mean and variance 1;
2. t_k is generated from the second order autoregressive process given by
$$x(t) - 1.4x(t-1) + 0.5x(t-2) = v(t);$$
3. t_k is generated from the autoregressive moving average process given by
$$x(t) - 1.8x(t-1) + 0.9x(t-2) =$$
$$v(t) + 0.3v(t-1) - 0.5v(t-2);$$
4. t_k is generated from the mixed process given by
$$x(t) + 1.3x(t-1) - 0.7x(t-2) =$$
$$0.7\cos(0.2t + \phi_1) + 0.9\cos(0.8t + \phi_2),$$

where $\{v(t)\}$ is a Gaussian process with zero mean and variance 1, the phases ϕ_i are random variables uniformly distributed on the interval $-\pi \le \phi_i \le \pi$. We remark that autoregressive, autoregressive moving average and mixed processes are commonly used in signal processing applications, see [9]. We employ these input processes to generate the Toeplitz data matrices. In the test, we choose the solution w in (2) to be a random vector and the right-hand side vector d is computed by $T_{m,n}w$ correspondingly. The computations were done by Matlab on a Sparc workstation in double precision. We compare our method and

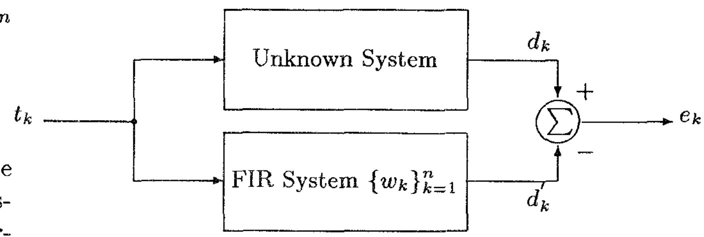

Figure 1: FIR System Identification Model.

	n			
k	32	64	128	256
8	0.17	0.47	1.31	4.54
16	0.21	0.57	1.62	5.05
32	0.31	0.77	2.06	5.94
64	0.50	1.17	2.94	7.75

Table 2: Number of mega-flops used by our method with $m = kn$.

QR method that solve a general dense linear least squares problem. The cost of the our method is of $O(m \log n + n^2)$ flops, and the cost for QR is of $O(mn^2)$ flops.

As for the comparison of times, Tables 2 and 3 show the number of mega-flops (counted by Matlab) used by our method and QR method respectively for the above examples. e see from the table that the number of mega-flops used for our method is significantly less than that of QR. For the above examples, we observe that the error of the computed solution $\tilde{w}$ by our method is at least $O(10^{-13})$. The error is computed as
$$\frac{\|\tilde{w} - w\|_2}{\|\tilde{w}\|_2}.$$

It is accurate as compared with QR method. These preliminary experiments suggest that the proposed method may be an efficient and effective method for the Toeplitz-least squares problems.

	n			
k	32	64	128	256
8	0.53	4.12	32.55	259.45
16	1.08	8.41	65.03	519.41
32	2.17	17.00	130.56	1039.12
64	4.37	34.17	261.89	2078.91

Table 3: Number of mega-flops used by QR method with $m = kn$.

5. REFERENCES

1. S. Akl, *The Design and Analysis of Parallel Algorithms*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

2. A. Bojanczyk, R. Brent and F. de Hoog, *QR Factorization of Toeplitz matrices*, Numer. Math., 49 (1986), pp. 81–94.

3. R. Chan, X. Jin, and M. Yeung, *The Circulant Operator in the Banach Algebra of Matrices*, Linear Algebra Appl., 149 (1991), pp. 41–53.

4. T. Chan, *An Optimal Circulant Preconditioner for Toeplitz Systems*, SIAM J. Sci. Stat. Comput., 9 (1988), pp. 766–771.

5. J. Chun, T. Kailath, and H. Lev-Ari, *Fast Parallel Algorithms for QR and Triangular Factorization*, SIAM J. Sci. Stat. Comput., 8 (1987), pp. 899–913.

6. I. Gohberg, T. Kailath and V. Olshevsky, *Fast Gaussian Elimination with Partial Pivoting for Matrices with Displacement Structure*, Math. Comp., 64 (1995), pp. 1557–1576.

7. G. Golub and Van Loan, *Matrix Computations*, Johns Hopkins Press, Baltimore, Second Edition (1989).

8. M. Gu, *New Fast Algorithms for Structured Least Squares Problems*, Lawrence Berkeley National Laboratory, preprint, 1995.

9. S. Haykin, *Adaptive Filter Theory*, 2nd ed., Prentice-Hall, Englewood Cliffs, NJ, 1991.

10. T. Kailath, S. Kung and M. Morf, *Displacement Ranks of Matrices and Linear Equations*, J. Math. Anal. and Appl., 68 (1979), pp. 395–407.

11. T. Kailath and V. Olshevsky, *Symmetric and Bunch-Kaufman Pivoting for Partiall Structured Cauchy-like Matrices with Applications to Toeplitz-like Linear Equations*, Information System Laboratory, Stanford University, preprint, 1995.

12. S. Marple, *Efficient Least Squares FIR System Identification*, IEEE TRans. Acoustics, Speech Signal Process., V29 (1981), pp. 62–73.

13. D. Sweet, *Fast Toeplitz Orthogonalization*, Numer. Math., 43 (1984), pp. 1–21.

5. ACKNOWLEDGMENTS: Research by M. Ng was supported by the Cooperative Research Centre for Advanced Computational Systems.