Electron. J. Math. Phys. Sci., **2002**, *1*, *1*, 92-103

Electronic Journal of
Mathematical
and Physical Sciences
EJMAPS
ISSN: 1538-263X
www.ejmaps.org

A FAST ALGORITHM FOR BLIND CHANNEL IDENTIFICATION

Michael K. Ng

Department of Mathematics, The University of Hong Kong, Pokfulam Road, Hong Kong.
E-mail: mng@maths.hku.hk.

Received: 26 June 2002/ Accepted: 15 July 2002/ Published: 22 August 20002

Abstract: In this paper, we address the problem of restoring a signal from its noisy convolutions with two unknown channels. When the transfer functions of these two channels have no common factors, the blind channel identification problem can be solved by finding the minimum eigenvalue of the Toeplitz-block matrix and its corresponding eigenvector. We present a fast algorithm to solve the numerical solution of the eigenvalue problem for Toeplitz-block matrices. Experimental results are given to illustrate the effectiveness of the method.

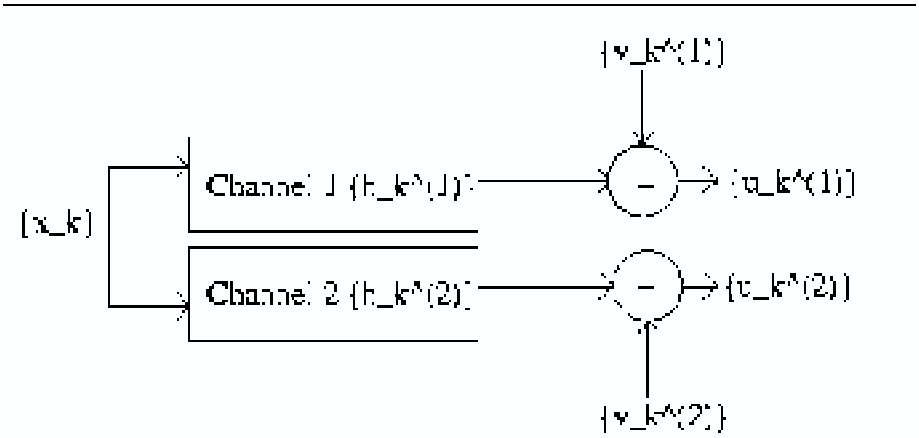
Keywords: Blind channel identification, Toeplitz-block matrix, eigenvalues
AMS Mathematical Subject Classification: 65F10, 65Y20

© 2002 by EJMAPS (http://www.ejmaps.org). Reproduction for noncommercial purposes permitted

1 Introduction

The problem of restoring a signal from a noisy version of its convolution with an unknown linear time-invariant nonminimum-phase system is a well-known problem in signal processing [5]. Traditionally, the channel identification can be done by using both known input and output signals. Once the channel coefficients are available, several linear or nonlinear methods can be used to estimate the input sequence, see Proakis [9] and the references therein. However, problems arise in

multi-point networks and multi-path fading channels, the receiver has to perform the channel



identification or equalization without the knowledge of input sequence.

Figure 1: 2-Channel Network.

Higher-order statistics based algorithms (see Haykin [5] for references therein) have been developed for solving blind channel identification and equalization problems. Since the time-average estimation of higher-order statistics requires a much larger sample size than that of the second-order statistics, the convergence rate of this algorithm is usually slow. Recently, Tong et al [11] and Moulines et al [8] have proposed blind channel identification methods using only the second-order statistics. Their method is to use signal and noise subspaces to extract the channel coefficients.

The outline of this paper is as follows. In §2, we formulate the blind identification method based on the second-order statistics. The blind identification problem can be reduced to compute the minimum eigenvalue of the Toeplitz-block matrix and its corresponding eigenvector. In §3, we propose a fast algorithm to solve the numerical solution of the eigenvalue problem for Toeplitz-block matrices. Finally, experimental results are given in §4 to illustrate the effectiveness of our method.

2 Problem Formulation

Suppose that the communication channel can be described by moving average (MA) model, the structure of the network is shown as in Figure 1. Suppose $\{x_k\}$ is the common input signal sequence and $[h_0^{(1)}, h_1^{(1)}, \ldots, h_L^{(1)}]'$ and $[h_0^{(2)}, h_1^{(2)}, \ldots, h_L^{(2)}]'$ are channel coefficients of two communication channels respectively. We assume that the higher order of the two channels has

already been known to be L. Let $\{u_k^{(j)}\}(j = 1,2)$ be the output sequence of the jth channel. The relationship between $\{x_k\}$ and $\{u_k^{(j)}\}$ can be described as

$$u_k^{(1)} = \sum_{j=0}^{L} x_{k-j} h_j^{(1)} + v_k^{(1)} \quad \text{and} \quad u_k^{(2)} = \sum_{j=0}^{L} x_{k-j} h_j^{(2)} + v_k^{(2)} \tag{1}$$

respectively where $v_k^{(j)}$ is the jth channel noise. In matrix form, we have

$$\begin{bmatrix} u_k^{(1)} \\ \vdots \\ \vdots \\ u_{k-L}^{(1)} \\ u_k^{(2)} \\ \vdots \\ \vdots \\ u_{k-L}^{(2)} \end{bmatrix} = \begin{bmatrix} h_0^{(1)} & h_1^{(1)} & \cdots & h_L^{(1)} & 0 & \cdots & 0 \\ 0 & h_0^{(1)} & h_1^{(1)} & \cdots & h_L^{(1)} & & \vdots \\ \vdots & & \ddots & \ddots & & \ddots & 0 \\ 0 & \cdots & 0 & h_0^{(1)} & h_1^{(1)} & \cdots & h_L^{(1)} \\ h_0^{(2)} & h_1^{(2)} & \cdots & h_L^{(2)} & 0 & \cdots & 0 \\ 0 & h_0^{(2)} & h_1^{(2)} & \cdots & h_L^{(2)} & & \vdots \\ \vdots & & \ddots & \ddots & & \ddots & 0 \\ 0 & \cdots & 0 & h_0^{(2)} & h_1^{(2)} & \cdots & h_L^{(2)} \end{bmatrix} \begin{bmatrix} x_k \\ x_{k-1} \\ \vdots \\ \vdots \\ \vdots \\ x_{k-2L+1} \\ x_{k-2L} \end{bmatrix} + \begin{bmatrix} v_k^{(1)} \\ \vdots \\ \vdots \\ v_{k-L}^{(1)} \\ v_k^{(2)} \\ \vdots \\ \vdots \\ v_{k-L}^{(2)} \end{bmatrix}, \tag{2}$$

or $\mathbf{u}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k$. To proceed with a statistical characterization of the channel, we make the following assumptions [11, 8]. (i) The transmitted signal $x(k)$ and channel noise $v_j(k)$ originate from wide-sense stationary processes that are statistically independent. (ii) The $(2L + 1)$-by-1 transmitted signal vector $\mathbf{x}_k$ has zero mean and correlation matrix $\mathbf{R}_x = \mathcal{E}(\mathbf{x}_k \mathbf{x}_k')$. The $(2L + 2)$-by-1 noise vector $\mathbf{v}_k$ has zero mean and correlation matrix $\mathbf{R}_v = \mathcal{E}(\mathbf{v}_k \mathbf{v}_k') = \sigma^2 I$. It follows from (1) that

$$\mathbf{u}_k' \mathbf{h} = \sum_{j=0}^{L} u_{k-j}^{(2)} h_j^{(1)} - \sum_{j=0}^{L} u_{k-j}^{(1)} h_j^{(2)} = \sum_{j=0}^{L} v_{k-j}^{(2)} h_j^{(1)} - \sum_{j=0}^{L} v_{k-j}^{(1)} h_j^{(2)} = \mathbf{v}_k' \mathbf{h} \tag{3}$$

where $\mathbf{h} = [h_0^{(2)}, h_1^{(2)}, \cdots, h_L^{(2)}, h_0^{(1)}, h_1^{(1)}, \cdots, h_L^{(1)}]'$. If the correlation matrix of the vector $\mathbf{u}_k$ is denoted by $\mathbf{R}_u$, a direct of conclusion of (2) and (3) will be

$$\mathbf{R}_u \mathbf{h} = \mathcal{E}(\mathbf{u}_k \mathbf{u}_k')\mathbf{h} = \mathcal{E}(\mathbf{u}_k \mathbf{v}_k')\mathbf{h} = \mathcal{E}(\mathbf{v}_k \mathbf{v}_k')\mathbf{h} = \mathbf{R}_v \mathbf{h} = \sigma^2 \mathbf{h},$$

where $\mathcal{E}$ is the expection operator. We note that $\mathbf{h}$ is the eigenvector of the correlation matrix $\mathbf{R}_u$ and σ^2 is the corresponding eigenvalue of $\mathbf{R}_u$. The correlation matrix $\mathbf{R}_u$ is then given by $\mathbf{R}_u = \mathcal{E}(\mathbf{u}_k \mathbf{u}_k') = \mathbf{H}\mathbf{R}_x\mathbf{H}' + \mathbf{R}_v$. If $\mathbf{H}$ has full column rank then $\mathbf{R}_u$ will have the distinctive smallest

eigenvalue σ^2 and the corresponding eigenvector will be the normalized vector of **h**. In [11], Tong et al have proved that if the two channel transfer functions $H^{(1)}(z) = \sum_{k=0}^{L} h_k^{(1)} z^{-k}$ and $H^{(2)}(z) = \sum_{k=0}^{L} h_k^{(2)} z^{-k}$ have no common zeros, then the matrix **H** has full column rank. Hence the channel coefficients can be given by finding the eigenvector of the minimum eigenvalue of the matrix $\mathbf{R}_u$:

$$\min_{\|\mathbf{y}\|=1} \mathbf{y}' \mathbf{R}_u \mathbf{y}.$$

In practice, we have to work the estimate of the correlation matrix $\mathbf{R}_u$. In this case, the correlation matrix are estimated from finite number of data samples $\{u_1^{(1)}, u_2^{(1)}, \ldots, u_m^{(1)}\}$ and $\{u_1^{(2)}, u_2^{(2)}, \ldots, u_m^{(2)}\}$ taken from the channel 1 and the channel 2 respectively. The usual approach is to estimate $\mathbf{R}_u$ by $\mathbf{R}_u \approx \frac{1}{m} \sum_{k=1}^{m} \mathbf{u}_k \mathbf{u}_k' = \frac{1}{m} \mathbf{U}'\mathbf{U}$. In this paper, we consider the correlation windowing method to estimate $\mathbf{R}_u$ by assuming that the data prior to $k = 1$ and after $k = m$ are zero, i.e., $u_k^{(1)} = u_k^{(2)} = 0$ for $k < 1$ and $k > m$, see for instance [5].

We remark that the data matrix **U** has a special structure. Each row of $\mathbf{U}'$ is a right-shifted version of the previous row. By utilizing this special displacement structure of the data matrix, the normal equations matrix $\mathbf{U}'\mathbf{U}$ is a Toeplitz-block matrix and can be written in the form

$$\mathbf{U}'\mathbf{U} = \begin{bmatrix} \mathbf{T}_{11} & -\mathbf{T}_{12} \\ -\mathbf{T}_{21} & \mathbf{T}_{22} \end{bmatrix} \tag{4}$$

where $\mathbf{T}_{ij}$ are $(L+1)$-by-$(L+1)$ Toeplitz matrices and $T_{12} = T_{21}'$. The first column and the first row of $\mathbf{T}_{ij}$ is given by $[\gamma_0^{(ij)}, \gamma_1^{(ij)}, \ldots, \gamma_L^{(ij)}]'$ and $[\gamma_0^{(ij)}, \gamma_1^{(ij)}, \ldots, \gamma_L^{(ij)}]$ respectively where $\gamma_\ell^{(ij)} = \sum_{k=1}^{m-\ell} u_k^{(i)} u_{k+\ell}^{(j)}$ for $0 \le \ell \le L$.

Next we consider applying an iterative method based on the displacement structure of the Toeplitz-block matrix for computing the minimum eigenvalue of $\mathbf{U}'\mathbf{U}$ and its corresponding eigenvector.

3 The Algorithm

An n-by-n matrix **A** has displacement structure in the sense that

$$\mathbf{AZ} - \mathbf{ZA} = \mathbf{BC}^t, \tag{5}$$

where **Z** is the n-by-n lower shift matrix, **B** and **C** are in $\mathcal{C}^{n,\alpha}$, and α is small compared to n. (For discussions of other types of displacement structure see Kailath and Sayed [7]). The smallest integer α for which (5) holds with some n-by-α matrices **B** and **C**, is called *the* $\{Z, Z\}$*-displacement rank of* **A**; we will call it simply the *displacement rank of* **A**. Henceforth we will say that a matrix which satisfies (5) with α small compared to n is a *Toeplitz-like* matrix.

An n-by-n matrix which has displacement structure indicated in (5) is determined by only $O(n)$ entries rather than n^2 entries, therefore there are many efficient direct and iterative methods that exploit displacement structure to solve Toeplitz-like systems, see for instance Kailath and Sayed [7], and Chan and Ng [1] respectively. Efficient algorithms for the numerical solution of the Toeplitz eigenvalue problem have also been developed, see [2, 4, 6, 10]. In [10], Trench has presented an iterative method based on displacement structure for computing eigenvalues and eigenvectors of a class of Toeplitz matrices. The method obtains a specific individual eigenvalue (i.e., the i-th smallest, where i is a specified integer in $[1, 2, \ldots, n]$) of an n-by-n matrix at a computational cost of $O(n^2)$ operations. An associated eigenvector is obtained as a byproduct. The method is more efficient than general purpose methods such as the QR algorithm for obtaining a small number (compared to n) of eigenvalues, see [10]. The main idea of the iterative method is given by the following theorem.

Theorem 1 *Let* $\mathbf{A} = [a_{ij}]_{i,j=1}^{n}$ *be an* n*-by-*n *symmetric matrix, and define* $\mathbf{A}_m = [a_{ij}]_{i,j=1}^{m}$ *for* $1 \le m \le n$. *Let* $q_m(\lambda) = p_m(\lambda)/p_{m-1}(\lambda)$ *for* $1 \le m \le n$, *where* $p_0(\lambda) = 1$ *and* $p_m(\lambda) = \det(A_m - \lambda I)$, *and* $\Sigma_m(\mathbf{A}_m)$ *be the spectrum of* $\mathbf{A}_m$. *For each* $\lambda \in \Sigma_n(\mathbf{A}_n) = \bigcup_{m=1}^{n-1} \Sigma_m(\mathbf{A}_m)$, *then* $\mathrm{Neg}_n(\lambda)$ *(the number of eigenvalues of* $\mathbf{A}_n$ *less than* λ*) equals the number of negative quantities in* $\{q_1(\lambda), q_2(\lambda), \ldots, q_n(\lambda)\}$, *where*

$$q_1(\lambda) = a_{11} - \lambda, \qquad q_m(\lambda) = a_{mm} - \lambda - \mathbf{r}'_{m-1}\mathbf{w}_{m-1}(\lambda) \qquad (m = 2,\ldots,n).$$

$\mathbf{w}_{m-1}(\lambda)$ is the solution of the linear system

$$(A_{m-1} - \lambda\mathbf{I})\mathbf{w}_{m-1}(\lambda) = \mathbf{r}_{m-1}, \tag{6}$$

and $\mathbf{r}_{m-1} = [a_{1,m}a_{2,m}\cdots a_{m-1,m}]'$, i.e., the mth column of A_m with the first $(m-1)$ entries.

Theorem 1 provides a way to compute $p_n(\lambda)/p_{n-1}(\lambda)$ and the inertia of $\mathbf{A} - \lambda\mathbf{I}$. We note that if λ is an eigenvalue of $\mathbf{A}$, then $[\mathbf{w}_{n-1}(\lambda), -1]'$ is an associated eigenvector. Therefore, it can be used in conjunction with a root-finding procedure to determine a given eigenvalue of $\mathbf{A}$. Using Theorem 1, we can derive an efficient algorithm for computing the minimum eigenvalue and associated eigenvector of the Toeplitz-block matrix $\mathbf{U}'\mathbf{U}$ in (4). We first note that

$$\mathbf{U}'\mathbf{U}\mathbf{Z} - \mathbf{Z}\mathbf{U}'\mathbf{U} = \mathbf{B}\mathbf{C}'$$

$$= \begin{bmatrix} 1 & 0 & -\gamma_0^{(12)} & 0 \\ 0 & 0 & -\gamma_0^{(12)} - \gamma_L^{(11)} & \gamma_L^{(21)} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & -\gamma_L^{(12)} - \gamma_1^{(11)} & \gamma_1^{(21)} \\ 0 & 1 & \gamma_0^{(22)} & 0 \\ 0 & 0 & \gamma_1^{(22)} + \gamma_L^{(12)} & -\gamma_L^{(22)} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \gamma_L^{(22)} + \gamma_1^{(12)} & -\gamma_1^{(22)} \end{bmatrix} \begin{bmatrix} \gamma_1^{(11)} & -\gamma_1^{(21)} - \gamma_L^{(11)} & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \gamma_L^{(11)} & -\gamma_1^{(12)} - \gamma_1^{(11)} & 0 & 0 \\ 0 & -\gamma_0^{(11)} & 1 & 0 \\ \gamma_1^{(21)} & -\gamma_1^{(22)} + \gamma_1^{(12)} & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ -\gamma_L^{(21)} & -\gamma_L^{(22)} + \gamma_1^{(12)} & 0 & 0 \\ 0 & \gamma_0^{(12)} & 0 & 1 \end{bmatrix}' \tag{7}$$

Here $\mathbf{B}$ and $\mathbf{C}$ are n-by-4 matrices with $n = 2L + 2$.

Henceforth $\mathbf{B}_m$ and $\mathbf{C}_m$ $(1 \le m \le n)$ are the $m \times 4$ matrices obtained by dropping rows $m+1,\ldots,n$ from $\mathbf{B}$ and $\mathbf{C}$ in (5); thus $\mathbf{B}_m = \mathbf{V}_m\mathbf{B}$ and $\mathbf{C}_m = \mathbf{V}_m\mathbf{C}$, where $\mathbf{V}_m$ is the $m \times n$ matrix obtained by dropping the same rows from $\mathbf{I}$. We denote the jth column of $\mathbf{B}_m$ by $\mathbf{b}_j^{(m)} = [b_{1j}\cdots b_{mj}]'$; and therefore

$\mathbf{B}_m = [\mathbf{b}_1^{(m)}\mathbf{b}_2^{(m)}\mathbf{b}_3^{(m)}\mathbf{b}_4^{(m)}]$. The following algorithm provides an $O(n^2)$ method for solving the linear systems (6) if $\mathbf{U}'\mathbf{U}$ satisfies (7).

ALGORITHM: *If* $\lambda \in \Sigma_n(\mathbf{U}'\mathbf{U})$ *then* $q_1(\lambda),\ldots,q_n(\lambda)$ *can be computed as follows:*

$$q_1(\lambda) = \gamma_0^{(22)} - \lambda, \quad w_1(\lambda) = \frac{\gamma_1^{(22)}}{q_1(\lambda)}, \quad f_j^{(0)}(\lambda) = \frac{b_{1j}}{q_1(\lambda)}, \quad 1 \le j \le 4,$$

$$\left(f_1^{(0)} = \frac{1}{q_1(\lambda)}; \quad f_2^{(0)} = 0; \quad f_3^{(0)} = -\frac{\gamma_0^{(21)}}{q_1(\lambda)}; \quad f_4^{(0)} = 0 \right),$$

and $q_m(\lambda) = a_{mm} - \lambda - \mathbf{r}_m' \mathbf{w}_{m-1}(\lambda)$ $(m = 2,\ldots,n)$ where

$$\mathbf{f}_j^{(m)}(\lambda) = \begin{bmatrix} \mathbf{f}_{j-1}^{(m-1)}(\lambda) \\ 0 \end{bmatrix} - \frac{(b_{mj} - \mathbf{r}_m' \mathbf{f}_{j-1}^{(m-1)}(\lambda))}{q_m(\lambda)} \begin{bmatrix} \mathbf{w}_{m-1}(\lambda) \\ -1 \end{bmatrix}, \quad 1 \le j \le 4, \tag{8}$$

and

$$\mathbf{w}_m(\lambda) = \begin{bmatrix} 0 \\ \mathbf{w}_{m-1}(\lambda) \end{bmatrix} - [\mathbf{f}_1^{(m)}(\lambda)\mathbf{f}_2^{(m)}(\lambda)\mathbf{f}_3^{(m)}(\lambda)\mathbf{f}_4^{(m)}(\lambda)]\mathbf{C}_m' \begin{bmatrix} \mathbf{w}_{m-1}(\lambda) \\ -1 \end{bmatrix}.$$

Proof. For simplicity, we write $\mathbf{A} = \mathbf{U}'\mathbf{U}$. If $\mathbf{A} = \mathbf{U}'\mathbf{U}$ satisfies (7) then

$$\mathbf{A}_m\mathbf{Z} - \mathbf{Z}\mathbf{A}_m = \mathbf{B}_m\mathbf{C}_m' - \mathbf{r}_m\mathbf{e}_m', \quad 2 \le m \le n, \tag{9}$$

where $\mathbf{e}_m$ is the last unit m-vector. Adding and subtracting $\lambda\mathbf{Z}$ on the left side of (9) yields

$$(\mathbf{A}_m - \lambda\mathbf{I})\mathbf{Z} - \mathbf{Z}(\mathbf{A}_m - \lambda\mathbf{I}) = \mathbf{B}_m\mathbf{C}_m' - \mathbf{r}_m\mathbf{e}_m', \quad 2 \le m \le n. \tag{10}$$

Since

$$\mathbf{e}_m' \begin{bmatrix} \mathbf{w}_{m-1}(\lambda) \\ -1 \end{bmatrix} = -1,$$

$$\mathbf{Z}(\mathbf{A}_m - \lambda\mathbf{I}) \begin{bmatrix} \mathbf{w}_{m-1}(\lambda) \\ -1 \end{bmatrix} = 0$$

and

$$\mathbf{Z} \begin{bmatrix} \mathbf{w}_{m-1}(\lambda) \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{w}_{m-1}(\lambda) \end{bmatrix},$$

we multiply (10) on the right by $[\mathbf{w}_{m-1}(\lambda), -1]'$ and then one the left by $(\mathbf{A}_m - \lambda\mathbf{I})^{-1}$ yields

$$\mathbf{w}_m(\lambda) = \begin{bmatrix} 0 \\ \mathbf{w}_{m-1}(\lambda) \end{bmatrix} - (\mathbf{A}_m - \lambda\mathbf{I})^{-1}\mathbf{B}_m\mathbf{C}_m' \begin{bmatrix} \mathbf{w}_{m-1}(\lambda) \\ -1 \end{bmatrix}.$$

We write in terms of the columns of $(\mathbf{A}_m - \lambda\mathbf{I})^{-1}\mathbf{B}_m$ as

$$\left[\mathbf{r}_1^{(m)}(\lambda)\mathbf{r}_2^{(m)}(\lambda)\mathbf{r}_3^{(m)}(\lambda)\mathbf{r}_4^{(m)}(\lambda)\right].$$

These columns are the solutions of

$$(\mathbf{A}_m - \lambda I_m)\mathbf{r}_j^{(m)}(\lambda) = \mathbf{b}_j^{(m)} = \begin{bmatrix} \mathbf{b}_j^{(m-1)} \\ b_{mj} \end{bmatrix}, \quad 1 \le j \le 4. \tag{11}$$

Since

$$(\mathbf{A}_{m-1} - \lambda \mathbf{I})\mathbf{r}_j^{(m-1)}(\lambda) = \mathbf{b}_j^{(m-1)}$$

and

$$(\mathbf{A}_m - \lambda \mathbf{I})\begin{bmatrix} \mathbf{w}_{m-1}(\lambda) \\ -1 \end{bmatrix} = -q_m(\lambda)\mathbf{e}_m,$$

it follows that the solutions of (11) are given by (8).

Let $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$ be the eigenvalues of the Toeplitz-block matrix U^*U, and suppose we wish to find its smallest eigenvalue λ_1. We assume that λ_1 is not an eigenvalue of any of the principal submatrices of U^*U. We first find an interval (β_1,β_n) containing λ_1 but not any other eigenvalues of U^*U, or any eigenvalues of the principal submatrices of U^*U. On such an interval the function $q_n(\lambda)$ in Theorem 1 is continuous. It can be shown that β_1 and β_n satisfy this requirement if and only if

$$\mathrm{Neg}_n(\beta_1) = 0, \quad \mathrm{Neg}_n(\beta_n) = 1, \quad q_n(\beta_1) > 0 \quad \text{and} \quad q_n(\beta_n) < 0,$$

and a strategy is given for obtaining (β_1,β_n) by means of bisection. After (β_1,β_n) is determined. We find λ_1 as a root of the function $q_n(\lambda)$ by root-finding algorithms. For an n-by-n Toeplitz-block matrix, the computations in Algorithm 1 require approximately $4Nn^2$ floating point operations (flops) for its smallest eigenvalue and an associated eigenvector, where N is the number of iterative steps (function evaluations) required in the root-finding algorithm. By comparison, standard QR method requires approximately $2n^3/3$ flops for the preliminary tridiagonalization of the matrix, after which all the eigenvalues can be computed with $O(n)$ flops [3]. On the basis of this count only, we see that the method presented here has a clear advantage over the QR method if only the smallest eigenvalue of

U^TU is computed, provided that N is small compared to $N < n$. In the next section, we will see that this constant N in our numerical examples will be smaller than n and almost a constant independent of n.

4 Simulation Results

In this section, simulation results are given. All computations were done with Matlab in double precision. The channel coefficients are given below:

$$h_k^{(1)} = 1.1 - \frac{|2k - L|}{L} \quad \text{and} \quad h_k^{(2)} = e^{-0.05|k-L|}, \qquad k = 0, 1, \ldots, L,$$

with $L = 64$, 128, 256. The shapes of the channel coefficients are triangular and Gaussian respectively. White Gaussian noise is added to the output. The output SNR is defined as $10\log_{10}\|\mathbf{u}\|_2 / \|\mathbf{v}\|_2$. In the test, the numbers of data samples used to estimate $\mathbf{R}_y$ are also varied. We use the Matlab M-file "fzero" to find the smallest eigenvalue as a root of the function $q_n(\lambda)$. This root-finding algorithm was originated by T. Dekker and further improved by R. Brent; see Matlab on-line documentation.) We stop the iteration of the root-finding method for the minimum eigenvalue when the difference between successive iterates μ_{k-1} and μ_k obtained by the root finder satisfies the inequality $|\mu_k - \mu_{k-1}| \le 5 \times 10^{-16} \times \max\{|\mu_k|, 1\}$. To check the accuracy of the minimum eigenvalue μ of U^TU and its corresponding eigenvector y, we compute the residual norm $res = \|U^T Uy - \mu y\|_2 / \|y\|_2$. We also compute the mean-square errors err of the estimated channel coefficients, which are defined as the average, over all the unknown parameters, $err = \|\mathbf{h} - \mathbf{h}_c\|_2^2/(2L + 2)$ where $\mathbf{h}$ is the original channel coefficients vector and $\mathbf{h}_c$ is the computed coefficients vector.

Table 1 shows the residual norms (res) in the order of magnitude 10^{-7}, the relative errors (err) in the order of magnitude 10^{-4}, and the number ($iter$) of iterations (function evaluations) required in the root-finding algorithm averaged over 100 runs of the algorithm. We find that the mean-square estimation error decreases when the number of data samples m increases since the estimate U^TU tends to the correlation matrix $\mathbf{R}_y$. From Table 1, we see that under various SNR, the average res was in the interval $[10^{-8}, 10^{-7})$ for the computed minimum eigenvalues of the U^TU of 100 generated matrices of

order 66, 130 and 258 ($2L+2$). We also note that the average number of iterations is about 14 and is independent of L (the order of the channels) and SNR. These experimental results reported here show that our algorithm is an efficient and effective method for computing the minimum eigenvalue of the Toeplitz-block matrix and hence for solving the blind channel identification problem.

In summary, we have presented a new algorithm for blind 2-channel identification problem based on the minimum eigenvalue of the Toeplitz-block matrix. Proof and simulations show its high convergence rate and accuracy.

Number of data samples m	SNR								
	∞			60			30		
	res	*err*	iter	*res*	*err*	iter	*res*	*err*	iter
30L	0.523	1.12	13.23	3.41	1.34	13.85	6.72	1.78	13.49
50L	2.81	0.95	13.56	6.19	1.02	13.81	4.56	1.24	13.79
100L	4.92	0.61	13.19	5.46	0.79	13.47	9.01	0.94	13.83
150L	0.834	0.29	13.34	2.23	0.40	13.52	4.51	0.56	13.50

Table 1: Average residual norms, relative errors and number of iterations required for the computation of the minimum eigenvalue of U^tU when $L = 32$

Number of data samples m	SNR								
	∞			60			30		
	res	*err*	iter	*res*	*err*	iter	*res*	*err*	iter
30L	0.785	1.45	14.54	3.47	1.51	14.87	8.64	1.89	14.56
50L	0.971	1.05	14.98	5.90	1.16	14.52	7.91	1.35	14.66
100L	6.57	0.89	14.76	4.12	0.90	14.38	5.60	0.99	14.45
150L	8.19	0.37	14.78	9.64	0.46	14.84	8.90	0.70	14.89

Table 2: Average residual norms, relative errors and number of iterations required for the computation of the minimum eigenvalue of U^tU when $L = 64$

Number of data samples m	SNR								
	∞			60			30		
	res	*err*	iter	*res*	*err*	iter	*res*	*err*	iter
30L	8.91	1.67	14.91	1.67	1.98	15.01	2.35	2.23	14.98
50L	9.10	1.23	14.26	3.34	1.58	14.90	1.48	1.79	14.97
100L	7.91	0.94	14.48	0.918	1.21	14.28	0.971	1.44	14.78
150L	9.56	0.45	14.33	0.991	0.78	14.45	1.05	1.01	14.65

Table 3: Average residual norms, relative errors and number of iterations required for the computation of the minimum eigenvalue of U^HU when $L = 128$.

Acknowledgement. M. Ng's research supported in part by RGC Grant Nos. 7132/00P, 7130/02P, and HKU CRCG Grant Nos. 10203501, 10203907 and 10203408.

References

1. R. Chan and M. Ng, *SIAM Review*, **1996**, *38*, 427.

2. G. Cybenko and C. Van Loan, *SIAM J. Sci. Stat. Comput.*, **1986**, *7*, 131.

3. G. Golub and C. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, **1996**, Chapter 8.

4. M. Hayes and M. Clements, *IEEE Trans. Acoustics, Speech, Sig. Proc.*, **1986**, *34*, 485.

5. S. Haykin, *Adaptive Filter Theory*, Prentice-Hall, 3rd Edition, **1996**, Chapter 18.

6. Y. Hu and S. Kung, *IEEE Trans. Acoustics, Speech, Sig. Proc.*, **1985**, *33*, 1264.

7. T. Kailath and A. Sayed, *SIAM Review*, **1995**, *37*, 297.

8. E. Moulines, P. Duhamel, J. Cardoso and S. Mayrargue, *IEEE Trans. Signal Processing*, **1995**, *43*, 516.

9. J. Proakis, *Digital Communications*, New York: McGraw-Hill, **1989**, Chapter 5.

10. W. Trench, *SIAM J. Matrix Th. Appl.*, **1988**, 9, 291.

11. L. Tong, G. Xu and T. Kailath, *Proc. 25th Asilomar Conf.*, **1991**, 856.