ON THE USE OF KERNEL STRUCTURE FOR BLIND EQUALIZATION

Jacob H. Gunther and A. Lee Swindlehurst

Brigham Young University
Dept. of Electrical and Computer Engineering
Provo, UT 84602
e-mail: gunther@ee.byu.edu

ABSTRACT

The mathematical theory of kernel (null space) structure of Hankel and Hankel-like matrices is applied to the problem of blind equalization of co-channel signals. This work builds on recently introduced ideas in blind equalization where the symbols are treated as deterministic parameters and estimated directly without estimating the channel first. The main contribution of the new approach is that it allows the simultaneous exploitation of shift structure in the data model and the finite alphabet property of the signals.

1. INTRODUCTION

The problem of blind equalization of multiple co-channel signals using an antenna array has received considerable attention in the literature. A large number of methods have been developed to solve this problem. These methods exploit structure in the data model or statistical or temporal properties of the signals. Techniques have been developed based on a range of different optimality criteria and can be found in adaptive or batch processing forms for single and multiple source cases. A few of these techniques are derived from subspace relationships that exist between the oversampled array outputs and the symbol sequences and the shift structure inherent in the data model [1, 2, 3]. Building upon these ideas, in this paper we propose a novel technique for blind equalization that is based on the mathematical theory of kernel (null space) structure of Hankel matrices. This method estimates the symbol sequences directly without an intermediate channel estimate.

Exploiting the finite alphabet property of digital communication signals is recognized as important for blind equalization. However, the subspace based blind equalizers do not exploit the FA property directly. Rather, they compute unconstrained least-squares estimates of the co-channel signals. The finite alphabet property is enforced in a second step by projecting the least-squares estimates onto the alphabet or by an exhaustive multiple symbol enumeration. In contrast, in this paper, we propose a recursive equalizer which exploits the FA property in a more direct way.

Many structured matrices (e.g., Hankel, Toeplitz, Vandermonde) have kernels which are also structured. Kernel structure theory plays an implicit role in a surprisingly wide variety of problems in mathematics, signal processing, control, and coding. We are interested in applying this theory to the blind equalization problem in an effort to develop a computationally efficient recursive equalizer.

2. DATA MODEL

Suppose a collection of d digital communication signals arrive at an M element antenna array whose outputs are sampled P times per symbol period. We assume that the channel has a finite impulse response spanning L symbol periods. Arranging all spatio-temporal samples of the array collected during the n^{th} symbol period into an MP vector, the following baseband model of the array outputs is obtained:

$$\begin{aligned}\mathbf{x}[n] &= \sum_{i=1}^{d}\sum_{j=0}^{L-1}\mathbf{h}_i[j]s_i[n-j] \\ &= \underbrace{\begin{bmatrix}\mathbf{H}_{L-1} & \cdots & \mathbf{H}_0\end{bmatrix}}_{\mathbf{H}}\begin{bmatrix}\mathbf{s}_{n-L+1}\\ \vdots \\ \mathbf{s}_n\end{bmatrix}\end{aligned}$$

where

$$\begin{aligned}\mathbf{H}_j &= \begin{bmatrix}\mathbf{h}_1[j] & \cdots & \mathbf{h}_d[j]\end{bmatrix} \quad MP \times d \\ \mathbf{s}_n &= \begin{bmatrix}s_1[n] & \cdots & s_d[n]\end{bmatrix}^T \quad d \times 1\end{aligned}$$

In this paper, we deal only with finite alphabet (FA) signals where $s_i[n] \in \Omega = \{\omega_1, \cdots, \omega_K\}$. Arranging consecutive samples into the columns of a matrix gives

$$\mathbf{X}_n^{(N-L+1)} = \mathbf{H}\mathbf{S}_n^{(N-L+1)} \tag{1}$$

where

$$\begin{aligned}\mathbf{X}_n^{(N-L+1)} &= \begin{bmatrix}\mathbf{x}_{n-N+L} & \cdots & \mathbf{x}_n\end{bmatrix} \\ \mathbf{S}_n^{(N-L+1)} &= \begin{bmatrix}\mathbf{s}_{n-N+1} & \cdots & \mathbf{s}_{n-L+1}\\ \vdots & & \vdots \\ \mathbf{s}_{n-N+L} & \cdots & \mathbf{s}_n\end{bmatrix}\end{aligned}$$

and the symbol matrix $\mathbf{S}_n^{(N-L+1)}$ is a block Hankel matrix with $d \times 1$ blocks $\mathbf{s}_n$. The superscript on the data and symbol matrices indicates the number of columns, while the subscript gives the index of the last block element in the matrix.

For now we neglect the influence of noise and assume that (1) is an exact mathematical description of the obser-

vations. In this paper, we assume that $\mathbf{H}$ has full column rank. When this is true ($MP > dL$ is necessary), then the row spans and the null spaces of the data and the symbol matrices coincide. It follows that

$$\mathbf{X}_n^{(N-L+1)}\mathbf{G}_n = 0 \quad \Longleftrightarrow \quad \mathbf{S}_n^{(N-L+1)}\mathbf{G}_n = 0. \qquad (2)$$

where the columns of $\mathbf{G}_n$ form a basis for the null space of $\mathbf{X}_n^{(N-L+1)}$. This relationship will be used in Section 4 to develop the blind equalizer.

3. KERNEL STRUCTURE OF HANKEL MATRICES

This section reviews the basic facts about kernel structure that will be needed to develop the blind equalizer. We focus on Hankel matrices but many of these facts can be adapted to hold for other types of structured matrices. We begin with some notational conventions.

Let $\mathcal{S}(n,N) = \{s_{n-N+1}, \cdots, s_n\}$ be an N point sequence terminating in s_n. Associated with the sequence $\mathcal{S}(n,N)$ is the family of Hankel matrices

$$\mathbf{S}_n^{(k)} = \begin{bmatrix} s_{n-N+1} & \cdots & s_{n-l+1} \\ \vdots & & \vdots \\ s_{n-k+1} & \cdots & s_n \end{bmatrix} \qquad (3)$$

where $k+l = N+1, k = 1, \cdots, N$, and as before the superscript indicates the number of columns. If the s_i are scalars then $\mathbf{S}_n^{(k)}$ is an $l \times k$ Hankel matrix. Note that $\mathbf{S}_n^{(1)}$ is a column and $\mathbf{S}_n^{(N)}$ is a row. If the s_i are matrices themselves, then the $\mathbf{S}_n^{(k)}$ are block Hankel matrices.

Many of the results which follow are more easily expressed in polynomial language rather than using matrix-vector notation. We will freely switch back and forth between these representations using the natural correspondence between spaces of length n vectors and the space of polynomials of degree less than n:

$$\mathbf{x} = \begin{bmatrix} x_0 \\ \vdots \\ x_{n-1} \end{bmatrix} \longleftrightarrow \mathbf{x}(\lambda) = x_0 + x_1\lambda + \cdots + x_{n-1}\lambda^{n-1}.$$

Denote the kernel (null space) of $\mathbf{S}_n^{(k)}$ by $\mathcal{N}_n^{(k)}$.

$$\mathcal{N}_n^{(k)} = \{\mathbf{u}(\lambda) : \mathbf{S}_n^{(k)}\mathbf{u} = 0\} \qquad (4)$$

Note that the coefficients in the vector polynomials $\mathbf{u}(\lambda)$ are $q \times 1$ vectors where q is the column dimension of s_i. The first theorem characterizes the null space of block Hankel matrices.

Theorem1 Given a sequence $\mathcal{S}(n,N) = \{\mathbf{s}_{n-N+1}, \cdots, \mathbf{s}_n\}$ of $p \times q$ matrices, there is a uniquely defined $(p+q)$-tuple of integers $(n_1, \cdots, n_{p+q})$, $0 \le n_1 \le \cdots \le n_{p+q} \le N+1$, and vector polynomials $\mathbf{u}_i(\lambda), i = 1, \cdots, p+q-\delta$ where δ is the defect of $\mathbf{S}_n^{(N)}$ such that $\deg(\mathbf{u}_i(\lambda)) \le n_i$ and the polynomials

$$\mathbf{u}_i(\lambda), \lambda\mathbf{u}_i(\lambda), \cdots, \lambda^{k-n_i-1}\mathbf{u}_i(\lambda) \qquad (5)$$

where i runs over all indices with $n_i < k$, form a basis of $\mathcal{N}_n^{(k)}$. Proof: See [4].

We call $\{\mathbf{u}_i(\lambda)\}_{i=1}^{p+q}$ the fundamental system (FS) and $\{n_i\}_{i=1}^{p+q}$ the characteristic degrees of $\mathcal{S}(n,N)$. The polynomials in (5) are referred to as a shift chain. The fundamental system is not unique but has the property that the $\mathbf{u}_i(\lambda)$ have no roots in common. In the following, we assume that the $(n_i, \mathbf{u}_i(\lambda))$ pairs are ordered so that the n_i are nondecreasing.

Theorem 1 characterizes the null space of the block Hankel symbol matrix $\mathbf{S}_n^{(N-L+1)}$ in (1) and all other block Hankel matrices $\mathbf{S}_n^{(k)}$ formed from the co-channel symbol sequences. A typical assumption in the blind equalization problem is that the co-channel signals (rows of $\mathbf{S}_n^{(N)}$) are linearly independent, in which case $\delta = 0$. Therefore, because the blocks in $\mathbf{S}_n^{(N-L+1)}$ are $d \times 1$ vectors, $d+1$ polynomials are needed to completely characterize the null space of $\mathbf{S}_n^{(k)}$ for all k, and the coefficients in these polynomials are just scalars since $q = 1$.

When there is only one source ($d = 1$) the theorem implies that $d+1 = 2$ polynomials (vectors) are needed to characterize the null space $\mathcal{N}_n^{(k)}$ of $\mathbf{S}_n^{(k)}$ for all k. This fact was overlooked in [5] where it was assumed that only one polynomial was needed to span $\mathcal{N}_n^{(k)}$ for all k. We summarize these results in the following corollary.

Corollary1 Let $\mathcal{S}(n,N) = \{\mathbf{s}_{n-N+1}, \cdots, \mathbf{s}_n\}$ consist of d linearly independent symbol sequences. Then, $d+1$ polynomials (vectors) are needed to completely specify $\mathcal{N}_n^{(k)}$ for all k. For a particular k, $\mathcal{N}_n^{(k)}$ is spanned by the columns of

$$\mathbf{U}_n^{(k)} = \left[\begin{array}{ccc|c|ccc} u_{1,0} & & & & u_{i,0} & & \\ \vdots & \ddots & & & \vdots & \ddots & \\ u_{1,n_1} & & u_{1,0} & \cdots & u_{i,n_i} & & u_{i,0} \\ & \ddots & \vdots & & & \ddots & \vdots \\ & & u_{1,n_1} & & & & u_{i,n_i} \end{array}\right]$$

$\underbrace{\qquad\qquad}_{k-n_1 \text{ columns}}$ $\qquad$ $\underbrace{\qquad\qquad}_{k-n_i \text{ columns}}$

where $u_{i,j}$ is the coefficient of λ^j in $\mathbf{u}_i(\lambda)$ and i runs over all indices with $k > n_i$.

The columns of $\mathbf{U}_n^{(k)}$ are the vector equivalent of the polynomial shift chain in (5).

This parsimonious parameterization of the null space of $\mathbf{S}_n^{(k)}$ is easily modified as data are added to or removed from $\mathcal{S}(n,N)$. The process of adding a symbol to the end of $\mathcal{S}(n,N)$ is referred to as extension. Removing a symbol from the beginning of $\mathcal{S}(n,N)$ is referred to as reduction. Given an FS for $\mathcal{S}(n,N)$, Heinig has given algorithms for computing the FS for $\mathcal{S}(n+1,N+1)$ or $\mathcal{S}(n-1,N-1)$ for both the single source ($d = 1$) [6] and the multi-source ($d > 1$) [7] cases. These formulas can easily be manipulated to give fundamental systems for $\mathcal{S}(n,N-1)$ and $\mathcal{S}(n,N+1)$.

These modification formulas were originally proposed as algorithms for fast inversion of Hankel and block Hankel matrices and can be applied to Toeplitz matrices as well. There are Levinson-type formulas which require an inner product and Schur-type formulas that work with a "residual system" and are inner product free. Table 1 gives Heinig's Levinson-type extension and reduction algorithms for the single source case. The important thing to note is the simplicity of the update. The fundamental system is modified by multiplying $[\mathbf{u}_1(\lambda) \;\; \mathbf{u}_2(\lambda)]$ by a upper triangular matrix.

Heinig's Algorithm (Levinson-Type, $d = 1$)

Extension by s_{n+1}

Input: a FS of $\mathcal{S}(n, N)$ and s_{n+1}.
Output: a FS of $\mathcal{S}(n+1, N+1) = \{s_{n-N+1}, \cdots, s_n, s_{n+1}\}$.
Let $a = [s_{n-n_1+1}, \cdots, s_{n+1}]\mathbf{u}_1$, $b = [s_{n-n_2+1}, \cdots, s_{n+1}]\mathbf{u}_2$.

$$\begin{bmatrix}\mathbf{u}_1(\lambda) & \mathbf{u}_2(\lambda)\end{bmatrix} \longleftarrow \begin{bmatrix}\mathbf{u}_1(\lambda) & \mathbf{u}_2(\lambda)\end{bmatrix}\begin{bmatrix}1 & c(\lambda)\\ 0 & 1\end{bmatrix}$$

$$\begin{bmatrix}n_1 & n_2\end{bmatrix} \longleftarrow \begin{bmatrix}n_1 & n_2\end{bmatrix} + \boldsymbol{\eta}$$

If $a = 0$ then	$c(\lambda) = 0$	$\boldsymbol{\eta} = [0, 1]$
Else if $b = 0$ then	$c(\lambda) = 0$	$\boldsymbol{\eta} = [1, 0]$
Else	$c(\lambda) = -\frac{b}{a}\lambda^{n_2-n_1}$	$\boldsymbol{\eta} = [1, 0]$

Reduction by s_{n-N+1}

Input: a FS of $\mathcal{S}(n, N)$.
Output: a FS of $\mathcal{S}(n, N-1) = \{s_{n-N+2}, \cdots, s_n\}$. Let $a = u_{1,0}, b = u_{2,0}$ and let $\tilde{\mathbf{u}}_i(\lambda) = \lambda^{n_i}\mathbf{u}_i(\lambda^{-1})$ be the reciprocal polynomial of $\mathbf{u}_i(\lambda)$.

$$\begin{bmatrix}\tilde{\mathbf{u}}_1(\lambda) & \tilde{\mathbf{u}}_2(\lambda)\end{bmatrix} \longleftarrow \begin{bmatrix}\tilde{\mathbf{u}}_1(\lambda) & \tilde{\mathbf{u}}_2(\lambda)\end{bmatrix}\begin{bmatrix}1 & c(\lambda)\\ 0 & 1\end{bmatrix}$$

$$\begin{bmatrix}n_1 & n_2\end{bmatrix} \longleftarrow \begin{bmatrix}n_1 & n_2\end{bmatrix} + \boldsymbol{\eta}$$

If $a = 0$ then	$c(\lambda) = 0$	$\boldsymbol{\eta} = [-1, 0]$
Else if $b = 0$ then	$c(\lambda) = 0$	$\boldsymbol{\eta} = [0, -1]$
Else	$c(\lambda) = -\frac{b}{a}\lambda^{n_2-n_1}$	$\boldsymbol{\eta} = [0, -1]$

Table 1:

Modifications for the $d > 1$ case also involve multiplication by an unit upper triangular matrix. After executing the algorithm we reorder the $\mathbf{u}_i(\lambda)$ if necessary so that the n_i are in nondecreasing order. By itself, Heinig's algorithm is numerically unstable. In practice, the FS must be scaled periodically. Also, we replace the conditions $a = 0$ and $b = 0$ with $|a| < \epsilon$ and $|b| < \epsilon$ for some small ϵ.

4. RECURSIVE BLIND EQUALIZATION USING STRUCTURED KERNELS

Now we combine the results from the previous two sections to develop an efficient recursive equalizer. For simplicity, we shall present the single source case. To begin, we assume that N is chosen large enough so that $\mathbf{S}_n^{(N-L+1)}$ is wide. We also assume that the input is rich in modes so that both $\mathbf{u}_1(\lambda)$ and $\mathbf{u}_2(\lambda)$ are required to span $\mathcal{N}_n^{(N-L+1)}$ (a necessary and sufficient condition for this is that $n_1, n_2 < N - L + 1$). In [5], modes are defined analogous to the number frequency components in an infinite sequence. The number of modes is equal to n_1, the characteristic degree of $\mathbf{u}_1(\lambda)$, whose roots $z_1, \cdots, z_{n_1}$ are the modes. One of the advantages of the scheme we propose is that it adapts naturally to the number of modes in the signals.

By Corollary 1, for $d = 1$ the columns of

$$\mathbf{U}_n^{(N-L+1)} = \left[\begin{array}{ccc|ccc} u_{1,0} & & & u_{2,0} & & \\ \vdots & \ddots & & \vdots & \ddots & \\ u_{1,n_1} & & u_{1,0} & u_{2,n_2} & & u_{2,0} \\ & \ddots & \vdots & & \ddots & \vdots \\ & & u_{1,n_1} & & & u_{2,n_2} \end{array}\right]$$

form a basis for $\mathcal{N}_n^{(N-L+1)}$. When no noise is present, it follows from (2) that $\operatorname{span}\{\mathbf{G}_n\} = \operatorname{span}\left\{\mathbf{U}_n^{(N-L+1)}\right\}$ and

$$\mathbf{X}_n^{(N-L+1)}\mathbf{U}_n^{(N-L+1)} = 0 \Longleftrightarrow \mathbf{S}_n^{(N-L+1)}\mathbf{U}_n^{(N-L+1)} = 0$$

Next we show how to exploit the finite alphabet property of the signal. For simplicity assume that the symbol alphabet is $\{+1, -1\}$. Given a FS for $\mathcal{S}(n, N)$ we compute, using Heinig's extension algorithm, an FS for a $+1$ extension $\mathcal{S}(n+1, N+1, +1)$ and a -1 extension $\mathcal{S}(n+1, N+1, -1)$. This leads to the shift chains $\mathbf{U}_{n+1}^{(N-L+2)}(+1)$ and $\mathbf{U}_{n+1}^{(N-L+1)}(-1)$, respectively. One of these extensions is correct (i.e., corresponds to the sequence actually transmitted). Define $r(s_{n+1})$ by

$$r(s_{n+1}) = \left\|\mathbf{X}_{n+1}^{(N-L+2)}\mathbf{U}_{n+1}^{(N-L+2)}(s_{n+1})\right\|_F^2 \qquad (6)$$

and suppose that the $+1$ extension is actually the correct extension (i.e., $s_{n+1} = +1$). Then we will have

$$r(+1) = 0 \qquad \text{and} \qquad r(-1) > 0.$$

So, the decision, is easy, at least when there is no noise since the subspace relation (2) holds exactly. In practice, we compute $r(s_{n+1})$ and decide $s_{n+1} = +1$ if $r(+1) < r(-1)$ and $s_{n+1} = -1$ otherwise. Additionally, we should point out that Heinig's algorithm gives an FS which is not scaled appropriately for comparison of $r(+1)$ with $r(-1)$. For this reason, we replace $\mathbf{U}_{n+1}^{(N-L+2)}(s_{n+1})$ with the $\mathbf{Q}$ factor from its QR factorization which can be computed very efficiently by exploiting displacement structure [8]. Before moving on to $n+2$ we execute a reduction step via Heinig's reduction algorithm to remove s_{n-N+1}. Hence, we track the FS for the most recent N samples of the symbol sequence.

The shift structure of the data has been fully exploited by using the shift chain basis of the null space. Furthermore, the FA property of the signal has been exploited by considering only feasible extensions. For arbitrary K element alphabets Ω, a total of K extensions must be computed (K^d for d signals). However, because the fundamental system is a parsimonious parameterization of the null space and can be extended and reduced very easily using Heinig's algorithm, this results in a very efficient algorithm. The largest expense in this algorithm is the matrix product in the computation of $r(s_{n+1})$.

We place no lower limit on the number of modes in the signals. Because we track the fundamental system of the exact symbol matrix, our algorithm can accommodate any number of signal modes from the minimum of 1 up to the maximum $\lfloor (N+1)/2 \rfloor$. This makes the algorithm useful as a blind start up procedure which could be used to equalize at the very beginning of transmission without

a channel estimate or training signals. Consider the $d = 1$ case. We can assume without loss of generality that the very first symbol transmitted is $s_0 = +1$ and $s_i = 0$ for $i < 0$. Initialize Heinig's algorithm with $[n_1, n_2] = [1, L]$ and $[\mathbf{u}_1(\lambda), \mathbf{u}_2(\lambda)] = [1, \lambda^L]$. Now apply Heinig's extension algorithm to the data as it arrives $\mathbf{X}_i^{(i+1)}$ for $i = 1, 2, \cdots$ until the desired window size is reached and then interleave reductions with the extensions. At each step we compare $r(+1)$ and $r(-1)$ to make a decision.

So far we have assumed that the number of sources d and the channel length L were known. The algorithm, however, is not very sensitive to an incorrectly chosen L. If L is underestimated, then fewer shifts are used which does not significantly effect the performance. Overestimating L however does impact the performance since more shifts are assumed in the data than are actually present.

A fundamental limitation of this algorithm occurs due to noise amplification. Computing $r(s_{n+1})$ amplifies the noise in the space spanned by the columns of $\mathbf{U}_{n+1}^{(N-L+2)}(s_{n+1})$. Denote this space by $\mathcal{N}(s_{n+1})$. Even if $s_{n+1} = +1$ is correct, the noise power in $\mathcal{N}(+1)$ might be greater than that in $\mathcal{N}(-1)$ leading to $r(+1) > r(-1)$ and an incorrect decision. As with any equalizer with decision feedback, a propagating error condition can occur. We suggest several remedies for this situation.

Remedy1 If the noise is white, we would expect its projection onto $\mathcal{N}(+1)$ and $\mathcal{N}(-1)$ to be similar in "size" so we would expect a large difference between $r(+1)$ and $r(-1)$. If the difference is small relative to some threshold T,

$$|r(+1) - r(-1)| < T,$$

then we cannot with certainty make a decision about s_{n+1}. We could then delay our decision until the next symbol period by extending the fundamental systems and computing $r(+1,+1), r(+1,-1), r(-1,+1), r(-1,-1)$. Then a decision about the pair $[s_{n+1}, s_{n+2}]$ could be made by choosing the smallest r.

Remedy2 We could continue delaying our decision about s_{n+1} as in Remedy 1 until time $n+M$ by computing the FS for all 2^M possible $r(\pm 1, \cdots, \pm 1)$. Then a decision about s_{n+1} could be made based on choosing the smallest r or comparing the sets $r(+1, \pm 1, \cdots, \pm 1)$ with $r(-1, \pm 1, \cdots, \pm 1)$ in which the sequences in each set are constant on s_{n+1}. Either alternative could be implemented efficiently in a recursive manner using the Viterbi algorithm with $r(\cdot)$ replacing the branch metric.

5. SIMULATIONS

Previously we suggested that the proposed algorithm could be used to equalize at the beginning of transmission without a channel estimate and without a training sequence. This blind "start-up" feature distinguishes our algorithm from other recursive algorithms. Batch processing methods, which can also be used for blind start-up, require at least 20 to 30 samples and can be very computationally demanding. In contrast, our efficient recursive method begins equalizing from the very start.

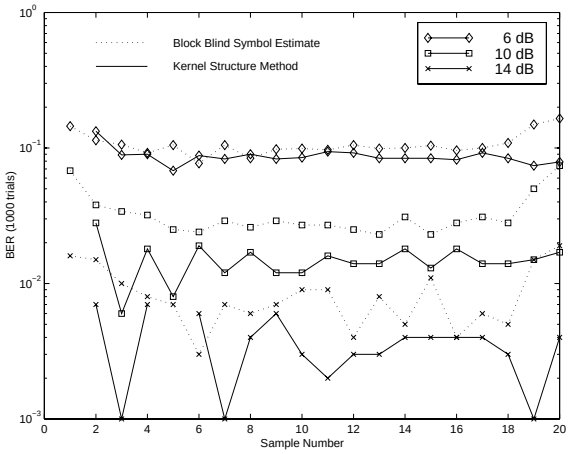

Figure 1: Kernel structure recursive blind equalizer in "start-up" mode.

We simulated reception of a single source over a length $L = 4$ channel. Assuming $MP = 6$. The fundamental system was initialized as $[n_1, n_2] = [1, L]$ and $[\mathbf{u}_1(\lambda), \mathbf{u}_2(\lambda)] = [1, \lambda^L]$. Possible signal kernels were tracked as suggested in Remedy 2 using the Viterbi algorithm and a 2^L state trellis. The results are displayed in Figure 1. For comparison, we also give simulation results obtained using a block method for blind symbol estimation proposed by van der Veen et al.[1] and Liu et al.[2].

6. REFERENCES

[1] A. van der Veen, S. Talwar, and A. Paulraj, "A subspace approach to blind space-time signal processing for wireless communication systems," IEEE Trans. SP, vol. 45, pp. 173–190, Jan. 1997.

[2] H. Liu and G. Xu, "Closed-form blind symbol estimation in digital communications," IEEE Trans. Sig. Proc., vol. 43, pp. 2714–2723, Nov. 1995.

[3] J. Gunther and A. Swindlehurst, "Blind sequential symbol estimation of co-channe finite alphabet signals," in 30^{th} Asilomar Conf., vol. 2, pp. 823–827, IEEE, Nov. 1996.

[4] G. Heinig and P. Jankowski, "Kernel structure of block hankel and toeplitz matricies and partial realization," Lin. Alg. Appl., vol. 175, pp. 1–30, 1992.

[5] G. Xu, H. Liu, L. Tong, and T. Kailath, "A least-squares approach to blind channel identification," IEEE Trans. SP, vol. 43, pp. 2982–2993, Dec. 1995.

[6] G. Heinig and P. Jankowski, "Parallel and superfast algorithms for hankel systems of equations," Numer. Math., vol. 58, pp. 109–127, 1990.

[7] G. Heinig, "Formulas and algorithms for block hankel matrix inversion and partial realization," in Progress in Systems and Control Theory, pp. 79–90, Boston: Birkhauser, 1989.

[8] T. Kailath and J. Chun, "Generalized displacement structure for block-toeplitz, toeplitz-block, and toeplitz-derived matricies," SIAM J. Matrix Anal. Appl., vol. 15, pp. 114–128, Jan. 1994.