Data-Recursive Algorithms for Blind Channel Identification in Direct-Sequence Systems

Dennis L. Goeckel[1], Alfred O. Hero III[2], and Wayne E. Stark[3]

November 30, 1995

Abstract

Algorithms for performing blind channel identification for a binary phase-shift keyed (BPSK) direct-sequence spread-spectrum (DS/SS) system operating over a fading channel are presented. These algorithms are derived by identifying the DS/SS system as a discrete oversampled system with intersymbol interference. In this setting the spreading code can be viewed as a transmit filter, the knowledge of which can be used to aid in channel identification. An off-line solution to the channel identification problem involves the determination of the eigenvector corresponding to the minimum eigenvalue of a matrix that depends on the correlation statistics of the samples of the received signal. In this paper, the online solution is derived for the case that the transmit filter and propagation channel are unknown and jointly identified. Novel low complexity stochastic gradient algorithms and conjugate gradient algorithms are derived and mean convergence conditions given. Then it is shown how the knowledge of the spreading code can be incorporated to aid in identification. An algorithm is then derived that utilizes trellis searching for joint data and channel identification for oversampled systems. Finally, numerical results in the form of channel estimation error are presented.

Keywords: *Blind Channel Identification, Direct-Sequence Spread-Spectrum Systems, Oversampled ISI Systems*

Corresponding Author

Dennis Goeckel
c/o Prof. Wayne Stark
Univ. of Michigan - EECS
1301 Beal
Ann Arbor, MI 48109-2122
Tel:(313) 764-4110
Fax:(313) 763-8041
Email: goeckel@eecs.umich.edu

Alfred O. Hero III
Univ. of Michigan - EECS
1301 Beal
Ann Arbor, MI 48109-2122
Tel:(313) 764-0564
Fax:(313) 763-8041
Email: hero@eecs.umich.edu

Wayne E. Stark
Univ. of Michigan - EECS
1301 Beal
Ann Arbor, MI 48109 -2122
Tel:(313) 763-0390
Fax:(313) 763-8041
Email: stark@eecs.umich.edu

[1]This work was supported in part by a National Science Foundation Graduate Fellowship and a Rackham Pre-Doctoral Fellowship.

[2]This work was supported in part by the National Science Foundation under grant BCS-90243370.

[3]This work was supported in part by the National Science Foundation under grant NCR-9115869 and the Army Research Office under contract DAAH04-95-I-0246.

1 Introduction

Direct-sequence spread spectrum (DS/SS) systems find wide application in modern digital communications systems, partially due to their ability to suppress multipath interference. In DS/SS systems, the information-bearing waveform is purposely expanded in frequency by multiplying the waveform during each symbol period by a spreading waveform with much higher bandwidth than the data rate. The spreading waveform is constructed by dividing the symbol period into many "chips" with each chip containing the product of a fixed chip waveform (i.e. the same for all chips) with either positive or negative one (depending on the chip). In this paper, the focus is on blind identification for single-user DS/SS systems.

The optimal receiver for a DS/SS system on an additive white Gaussian noise (AWGN) channel with no intersymbol interference (ISI) matched filters the continuous time received signal with the full spreading waveform and samples the filter output only once per symbol period. However, in practice the optimal receiver is implemented with a matched filter to the chip waveform sampled at the chip rate of the system, which is much higher than the symbol rate. This receiver is simpler in design, only requiring a continuous time correlation over a single chip interval, which can be implemented as an integrate-and-dump device, allowing correlation with the spreading sequence to be done in the digital domain. Since there are many chips for a single symbol interval, this receiver effectively samples the received signal waveform many times per symbol interval.

When a DS/SS system is subject to *known* ISI, it is no longer optimal to make decisions on a symbol-by-symbol basis but instead one must do sequence estimation using a Viterbi algorithm. Furthermore, for an ISI system that is oversampled, it is no longer possible to obtain a sufficient statistic for calculation of the likelihood of a given path of the Viterbi algorithm by multiplying the samples within a symbol period by a *transmitter* spreading coefficient and then summing. The optimal combining of the samples in a given symbol period depends on the channel impulse response at each *sample* period, not just at the symbol period. This is the motivation for performing channel identification of DS/SS systems at the chip period.

Much work has appeared on channel identification/equalization via oversampling the output of the channel [2, 3, 4, 5, 6, 7, 8, 9]. The common base of this work is that the oversampled process is cyclostationary, thus allowing for blind channel identification based on only second order statistics. Because estimation of second order statistics requires fewer samples than that of higher order statistics for a given level of estimation accuracy, one expects algorithms based on second order statistics to exhibit faster convergence.

The special feature of the DS/SS systems considered in this paper is that they are naturally oversampled by a factor equal to the processing gain. The high chip rates in DS/SS communications require the use of efficient algorithms for channel identification. One option would be to combine chip periods to lower the oversampling rate and speed up identification. Besides being suboptimal, as information is being lost in such an operation, the known spreading code of the user cannot be incorporated to improve convergence and steady state error performance of the channel identification algorithms [10].

As in much of the previous work, it will be assumed that the channel has a finite impulse response (FIR) and that the time support of the channel response is known. Efficient adaptive algorithms for highly oversampled systems are considered for the mathematically identical algorithms described in [2, 4, 11]. The formulation that will be used throughout the paper is similar to the subchannel response matching (SRM) algorithm [4], a block-oriented channel identification scheme which minimizes a sum of pairwise differences between filtered subchannel outputs. The SRM algorithm requires obtaining the eigenvector corresponding to the minimum eigenvalue of a matrix of correlation statistics of the received signal. This eigenvector has dimension equal to the product of the oversampling factor and the number of symbol periods on which the channel response to be identified is nonzero; thus, exact calculation of this eigenvector requires a number of floating point operations (flops) which is cubic in the oversampling rate.

For real-time operation, data-recursive algorithms of much lower complexity are desirable.

The problem of data-recursively obtaining the eigenvector corresponding to the minimum eigenvalue of a matrix is well studied in the literature on Pisarenko's harmonic retrieval method [12, 13, 14, 15, 16, 17] and direction-of-arrival estimation [18, 19, 20, 21]. In each case, however, the matrix to be analyzed consisted of the expectation of an outer product of the received vector with itself, which allows a very simple rank 1 matrix update formula. Unfortunately, the matrix used in the SRM algorithm is not of this form and these data-recursive algorithms are not applicable.

In this work, two data-recursive algorithms for the DS/SS channel identification problem are presented. First, stochastic gradient and conjugate gradient algorithms are derived that operate on an error term defined by the Rayleigh quotient to update the minimizing eigenvector in the SRM algorithm. The computational load of these algorithms is only linear in the oversampling rate for the case of unknown transmit filter and channel identification. The number of computations becomes quadratic in the oversampling rate when the knowledge of the transmit filter is incorporated. Furthermore, these algorithms are capable of tracking changes in the channel unlike the batch or "off-line" algorithms of [2, 4, 11] on which the algorithms are based. Convergence analyses are presented to aid in selection of the gain factor in the stochastic gradient algorithm. The convergence issues that arise due to data bit transitions are also discussed.

Finally, a trellis-searching blind channel identification algorithm is described by extending the methods of [22] to an oversampled system. In [22], blind channel identification is done by extending the optimal *known* ISI decoder, the Viterbi Algorithm, to blind identification by allowing each state to retain multiple paths. Each path operates with its own channel estimate, which is updated at each time instant. The extension here to the oversampled case results in a data-recursive algorithm with a computational load that is only linear in the oversampling rate.

An outline of the paper is the following. In Section 2, the system and channel model are defined, and the continuous DS/SS communication system mapped to an oversampled discrete system. The equivalence of the oversampled system to a multichannel system is then demonstrated, which leads naturally to the subchannel response matching algorithm. Section 3 develops the new adaptive algorithm for second order identification of oversampled systems, while Section 4 develops adaptive algorithms for a known transmit filter. Section 5 contains the extension of the trellis-searching algorithm of [22] to oversampled systems. Finally, numerical results for the presented algorithms are shown in Section 6.

2 Motivation and Problem Formulation

2.1 Mapping of a DS/SS System to a Discrete Oversampled System

Assume that binary phase-shift keyed (BPSK) modulation is employed with a data symbol duration of T_s. The BPSK signal is spread by multiplication with a spreading sequence having chip duration $T_c \ll T_s$, thus achieving a processing gain of $N = T_s/T_c$ in terms of receiver SNR. The transmitted data bits, denoted by the sequence (s_k), are assumed to be independent and identically distributed (IID). Since binary phase-shift keying is being employed, the output of the transmitter can be expressed as (using complex baseband notation)

$$x(t) = \sqrt{\frac{E_b}{NT_c}} s(t) e^{j\theta}$$

where E_b is the energy per data bit, θ is the transmitter phase (which will be absorbed into the channel response and suppressed from here forward), and $s(t)$ is the spread spectrum waveform given by

$$s(t) = \sum_{l=-\infty}^{\infty} s_l \sum_{m=0}^{N-1} a_m p(t - lT_s - mT_c)$$

where $p(t)$ is the normalized chip waveform, and a_m is the value of the spreading code for the m^{th} chip.

If the continuous channel response is denoted by $g(t)$, the received waveform is given by $r(t) = g(t) * x(t) + w(t)$ where $w(t)$ is a complex white Gaussian random process. If $g(t)$ was known, the optimal front end for the receiver would be a filter matched to the single symbol response $g_m * \sum_{m=0}^{N-1} a_m p(t - mT_c)$. With $g(t)$ unknown, this is not possible and the front end of the receiver matches to the ideal chip response, i.e. it simply integrates the received signal over a T_c interval. The sampled output of the receiver front end corresponding to symbol k and chip n is given by

$$y_n(k) = y(kT_s + nT_c) = \int_{kT_s+nT_c}^{kT_s+(n+1)T_c} r(t)dt.$$

It is desired to fit the system to an oversampled discrete system so that results from the literature can be applied. The following sequence of simple algebraic manipulations establishes the desired relation:

$$\begin{aligned} y_n(k) &= \sqrt{\frac{E_b}{NT_c}} \int_{kT_s+nT_c}^{kT_s+(n+1)T_c} dt \int_{-\infty}^{\infty} d\tau\ g(\tau) \sum_{l=-\infty}^{\infty} s_l \sum_{m=0}^{N-1} a_m p(t - \tau - lT_s - mT_c) + \eta_n(k) \\ &= \sqrt{\frac{E_b}{NT_c}} \sum_{l=-\infty}^{\infty} s_l \sum_{m=0}^{N-1} a_m \int_{-\infty}^{\infty} d\tau\ g(\tau) \int_0^{T_c} dt\ p(t - \tau + (kN + n - lN - m)T_c) + \eta_n(k) \\ &= \sum_{l=-\infty}^{\infty} s_l h_{(k-l)N+n} + \eta_n(k) = \sum_{l=-\infty}^{\infty} h_n(l) s_{k-l} + \eta_n(k) \qquad (1) \end{aligned}$$

where $\eta_n(k)$ is a sequence of IID gaussian random variables with variance $\frac{N_0}{2}T_c$, $\{h_n(l)\}_{l=0}^{L-1}$ is the n^{th} subchannel response, which is FIR with L coefficients,

$$h_{lN+n} = \sqrt{\frac{E_b}{N}} \sum_{m=0}^{N-1} a_m g_{lN+n-m} \qquad (2)$$

and

$$g_m = \sqrt{\frac{1}{T_c}} \int_{-\infty}^{\infty} d\tau \; g(\tau) \int_0^{T_c} dt \; p(t + mT_c - \tau).$$

The transmitter energy has been purposely worked into the channel so that the average received signal energy for a given chip can be normalized to unity and the variance of the noise adjusted to obtain an overall *received* signal-to-noise ratio (SNR) per symbol period E_s/N_0.

Relation (1) establishes that for each symbol period (fix k in $y_n(k)$) the chip samples $\{y_n(k)\}_{n=0}^{N-1}$ can be viewed as the outputs of N different subchannels, $h_n(l)$, $n = 0, \ldots, N-1$, excited by the same symbol sequence $\{s_k\}$ and operating at the symbol rate T_s. The channel identification algorithms described here accomplish identification of the entire channel by identification of each of the N subchannels. Thus, the DS/SS system fortuitously falls into the setting of channel identification via oversampling as presented in [2, 4, 11]. The distinction is that in DS/SS systems the spreading waveform is naturally oversampled and no extra sampling equipment needs to be designed into the receiver. Observe also that when the spreading code $\{a_0, \ldots, a_{N-1}\}$ is known, as occurs in the majority of applications, the channel h_{lN+n} is partially known. This falls into the category of channel identification problems with known transmit filter; in this case, (2) indicates that the spreading code is the transmit filter.

2.2 Subchannel Response Matching

Next the subchannel identification criterion used here and in [2, 4, 11] is described. Consider any two subchannels with impulse responses $h_{n_0}(k)$ and $h_{n_1}(k)$, and outputs $y_{n_0}(k)$ and $y_{n_1}(k)$:

$$y_{n_0}(k) = h_{n_0}(k) * s_k$$

$$y_{n_1}(k) = h_{n_1}(k) * s_k$$

Since the channels have the same inputs, if one of the channels is attached to the output of the other and vice versa, in the noiseless case the outputs should be equivalent. In the absence of noise, it is obvious that $h_{n_1}(k) * y_{n_0}(k) = h_{n_0}(k) * y_{n_1}(k)$, since $h_{n_1}(k) * h_{n_0}(k) = h_{n_0}(k) * h_{n_1}(k)$. This suggests that to identify the channels $h_{n_0}(k)$ and $h_{n_1}(k)$, an option is to consider subchannel estimates $\hat{h}_{n_0}(k)$ and $\hat{h}_{n_1}(k)$ that minimize the mean-squared difference $\mathcal{E}_2(\hat{h}_{n_0}, \hat{h}_{n_1}) = E|\hat{h}_{n_1}(k) * y_{n_0}(k) - \hat{h}_{n_0}(k) * y_{n_1}(k)|^2$. Clearly $\mathcal{E}_2 = 0$ when the channels are identified correctly. Furthermore, if the channels have no common zeroes (which is the well-known identifiability criterion for second order identification [3]), $\mathcal{E}_2 = 0$ implies $\hat{h}_{n_0}(k) = h_{n_0}(k), \hat{h}_{n_1}(k) = h_{n_1}(k)$ if the trivial solution $\hat{h}_{n_0}(k) = 0, \hat{h}_{n_1}(k) = 0, \forall k$, is excluded.

For the general case of $N > 2$, define the vector (of length LN) of concatenated subchannels:

$$\underline{h} \triangleq [h_0(0)\ h_0(1) \ldots h_0(L-1)\ h_1(0)\ h_1(1) \ldots h_1(L-1) \ldots h_{N-1}(0)\ h_{N-1}(1) \ldots h_{N-1}(L-1)]^T$$

and the vector (of length LN) of concatenated subchannel observation vectors as:

$$\underline{y}(k) \triangleq [\underline{y}_0^T(k), \underline{y}_1^T(k), \ldots, \underline{y}_{N-1}^T(k)]^T$$

where

$$\underline{y}_n(k) \triangleq [y_n(k)\ y_n(k-1) \ldots y_n(k-L+1)]^T.$$

Let the notation $\underline{x}[n]$ denote the n^{th} element of the vector $\underline{x}$. The objective function is then defined as the sum of all pairwise mean-squared differences:

$$\mathcal{E}(\hat{\underline{h}}) = \sum_{m=0}^{N-2} \sum_{n=m+1}^{N-1} E[|\sum_{i=0}^{L-1} \hat{\underline{h}}[mL+i] y_n(k-i) - \sum_{j=0}^{L-1} \hat{\underline{h}}[nL+j] y_m(k-j)|^2], \tag{3}$$

which when minimized over $\hat{\underline{h}}$, the estimate of $\underline{h}$, gives a solution $\underline{h}^{opt}$. A constraint must be added to avoid the trivial solution $\underline{h}^{opt} = \underline{0}$; the constraint considered here is $||\hat{\underline{h}}||^2 = 1$.

Note that there is an implicit amplitude ambiguity in the problem; multiplying $\underline{h}$ by a complex constant does not affect the solution $\underline{h}^{opt}$.

The objective (3) can be rewritten in an alternate form. Define the L by L matrices: $R_{mn} \triangleq E[\underline{y}_m(k)\underline{y}_n^H(k)]$. Then

$$\mathcal{E}(\underline{\hat{h}}) = \underline{\hat{h}}^T S \underline{\hat{h}}^* \tag{4}$$

where $\underline{\hat{h}}^*$ is the complex conjugate (no transpose) of $\underline{\hat{h}}$,

$$S = [I_N \otimes \sum_{n=0}^{N-1} R_{nn}] - V, \tag{5}$$

I_N is an N by N identity matrix, $\otimes$ is the Kronecker product, and

$$V = \begin{bmatrix} R_{00} & R_{10} & \dots & R_{(N-1)0} \\ R_{01} & R_{11} & & \vdots \\ \vdots & & \ddots & \\ R_{0(N-1)} & \dots & & R_{(N-1)(N-1)} \end{bmatrix}.$$

The following properties of S are important:

1. S is Hermitian symmetric, i.e. $S^H = S$.
2. From (3) and (4), it can be shown that S is non-negative definite.
3. If the channels have no common zeroes, there will be a *unique* minimum eigenvalue of S and the conjugate of its eigenvector will be the desired channel $\underline{h}$ [3]. From (5), the minimum eigenvalue in a noiseless or noisy system is equal to 0 or $(N-1)\sigma^2$, respectively, where σ^2 is the variance of the observation noise per sample.
4. For $N > 2$, V is *not* an outer product of the form $E[\underline{y}_k \underline{y}_k^T]$.

3 Adaptive Algorithms for the General Problem

For the DS/SS application considered here, the oversampling rate will be much higher than the number of periods over which the impulse response of the channel has support (i.e. $N \gg L$). Thus, adaptive algorithms will be sought that have low complexity in N. Historically, algorithms for iteratively solving for the eigenvector corresponding to the minimum eigenvalue of a non-negative matrix have fallen into three classes: stochastic gradient [12, 14, 15, 20], conjugate gradient [13, 23], and Newton's method [16, 17]. It is well known for the harmonic retrieval problem that the latter two algorithms converge significantly faster than the stochastic gradient algorithm [13, 17]. However, in the DS/SS problem, Newton's method is impractical because it requires the computation of the inverse of the Hessian matrix. Thus, attention here is restricted to the stochastic gradient and conjugate gradient algorithms.

3.1 Stochastic Gradient Algorithm: Definition

Although finding the eigenvector corresponding to the minimum eigenvalue of a matrix corresponds closely to Pisarenko's harmonic retrieval method [12] and direction of arrival estimation [18], as noted previously S is not expressible as the expectation of an outer product of the observed vector unless $N = 2$; hence, the simple data updates of [12, 20] do not apply. As in early work on Pisarenko's harmonic retrieval [12], a simple stochastic gradient algorithm is considered first. The algorithm updates $\underline{\hat{h}}_{k-1}$, the estimate of $\underline{h}^{opt}$ at iteration $k-1$, based on an additional symbol period of data: $y_0(k), y_1(k), \ldots, y_{N-1}(k)$.

The standard stochastic gradient update equation for $\underline{\hat{h}}_k$ is given by

$$\underline{\hat{h}}_k = \underline{\hat{h}}_{k-1} - \mu \bigtriangledown e_k(\underline{\hat{h}}_{k-1}) \tag{6}$$

where $\bigtriangledown$ is the gradient operator and $e_k(\underline{\hat{h}}_{k-1})$ is the empirical estimate of the Rayleigh quotient of S^*:

$$e_k(\underline{\hat{h}}_{k-1}) = \frac{\mathcal{E}_k(\underline{\hat{h}}_{k-1})}{||\underline{\hat{h}}_{k-1}||^2} = \frac{\underline{\hat{h}}_{k-1}^T S_k \underline{\hat{h}}_{k-1}^*}{||\underline{\hat{h}}_{k-1}||^2}, \tag{7}$$

$\mathcal{E}_k(\underline{\hat{h}}_{k-1}) = \underline{\hat{h}}_{k-1}^T S_k \underline{\hat{h}}_{k-1}^*$ is the empirical estimate of $\mathcal{E}(\underline{\hat{h}}_{k-1})$, and S_k is defined by S in (5) except that the single sample empirical estimates $\hat{R}_{mn} = \underline{y}_m(k)\underline{y}_n^H(k)$ are used in place of $R_{mn} = E[\underline{y}_m(k)\underline{y}_n^H(k)]$. As contrasted with the algorithms of [12, 16], normalization of $\underline{\hat{h}}_k$ to unity is unnecessary since it is accounted for by the denominator of $e_k(\underline{\hat{h}}_{k-1})$.

The gradient of $e_k(\underline{\hat{h}}_{k-1})$ is simply computed to obtain the update formula:

$$\underline{\hat{h}}_k = \underline{\hat{h}}_{k-1} - \mu \bigtriangledown e_k(\underline{\hat{h}}_{k-1}) = \underline{\hat{h}}_{k-1} - \mu \frac{(||\underline{\hat{h}}_{k-1}||^2 S_k^* \underline{\hat{h}}_{k-1} - (\underline{\hat{h}}_{k-1}^T S_k \underline{\hat{h}}_{k-1}^*)\underline{\hat{h}}_{k-1})}{||\underline{\hat{h}}_{k-1}||^4}. \tag{8}$$

In (8), the constant factor of 2 has been absorbed into μ. Brute force computation of this update requires $O(L^2N^2)$ flops because of the matrix-vector multiplication $S_k\underline{\hat{h}}_{k-1}^*$. However, a more careful consideration of the product $S_k\underline{\hat{h}}_{k-1}^*$ reveals that only $O(L^2N)$ flops are necessary. Noting that $\bigtriangledown\mathcal{E}_k(\underline{\hat{h}}_{k-1}) = 2S_k^*\underline{\hat{h}}_{k-1}$, element $mL+l$ of $S_k\underline{\hat{h}}_{k-1}^*$ can be obtained as

$$\begin{aligned}
\frac{\partial \mathcal{E}_k(\underline{\hat{h}}_{k-1})}{\partial \underline{\hat{h}}_{k-1}[mL+l]} &= 2\sum_{n=0,n\neq m}^{N-1}(\sum_{i=0}^{L-1}\underline{\hat{h}}_{k-1}[mL+i]y_n(k-i)y_n^*(k-l) \\
&\quad -\sum_{j=0}^{L-1}\underline{\hat{h}}_{k-1}[nL+j]y_m(k-j)y_n^*(k-l)) \\
&= 2(\sum_{i=0}^{L-1}\underline{\hat{h}}_{k-1}[mL+i]\sum_{n=0}^{N-1}y_n(k-i)y_n^*(k-l) \\
&\quad -\sum_{j=0}^{L-1}y_m(k-j)\sum_{n=0}^{N-1}\underline{\hat{h}}_{k-1}[nL+j]y_n^*(k-l)).
\end{aligned} \tag{9}$$

The computation of the right side of (9) proceeds as follows: the sums over n must be done for each l and i (or j), thus leading to $O(L^2N)$ operations; given these sums, $S_k\underline{\hat{h}}_{k-1}^* = \bigtriangledown\mathcal{E}_k(\underline{\hat{h}}_{k-1})$ can be found in $O(LN)$ inner products of vectors of length L. Thus, the update (8) can be done in $O(L^2N)$ flops. Since $N \gg L$ for DS/SS systems, this amounts to a large savings in computational effort for the algorithm. By reversing the summations in (9), the update can be performed in $O(LN^2)$ flops, which may be useful in other applications. Finally, note that there are alternate ways to obtain efficient updates; one option is to use the structure of S to write the length L subvectors of $S_k\underline{\hat{h}}^*_{k-1}$ as the sum of outer products of subchannel observations.

3.2 Stochastic Gradient Algorithm: Convergence

In this section, the mean convergence behavior is evaluated. This is used to select the gain factor μ of the adaptive algorithm.

3.2.1 Convergence Analysis Assuming Perfect Estimation of S

Global convergence is difficult to demonstrate due to the complexity of the error surface. However, it is possible to show that for a noisy system all of the eigenspaces except the one corresponding to the minimum eigenvector are unstable; thus, the algorithm will eventually leave these spaces [14]. A less ambitious but desirable goal is to choose μ such that the algorithm will converge to the correct solution when it is near the minimizing eigenvector. Here a first order convergence analysis is performed, which is equivalent to assuming S can be estimated without error. In this case, the error after iteration k of the algorithm is given by the Rayleigh quotient:

$$e(\underline{\hat{h}}_k) = \frac{\underline{\hat{h}}_k^T S \underline{\hat{h}}_k^*}{||\underline{\hat{h}}_k||^2} = \frac{\underline{\hat{h}}_k^T S_Z \underline{\hat{h}}_k^*}{||\underline{\hat{h}}_k||^2} + \lambda_0 \tag{10}$$

where S_Z is the matrix S for a noiseless system $(w(t) = 0)$ and $\lambda_0 = (N-1)\sigma^2$ is the unique smallest eigenvalue of S. It is appropriate to study only the convergence of the noiseless system as noise only contributes a constant noise floor, λ_0, to the objective function $\mathcal{E}(\underline{\hat{h}}_k)$. Since S_Z is Hermitian symmetric, it has an eigendecomposition $S_Z = \sum_{i=0}^{LN-1} \lambda_i \underline{v}_i \underline{v}_i^H$ where $\underline{v}_i$ is the i^{th}

eigenvector and λ_i the i^{th} eigenvalue of S_Z. Then, using the shorthand notation $\epsilon_k = e(\underline{\hat{h}}_k)$, the objective function becomes

$$\epsilon_k = \sum_{i=0}^{LN-1} \lambda_i \frac{||\underline{\hat{h}}_k^T \underline{v}_i||^2}{||\underline{\hat{h}}_k||^2} = \sum_{i=0}^{LN-1} \epsilon_k^i.$$

$\underline{\hat{h}}_k$ will converge to $\underline{h}^{opt}$ only if $\epsilon_k^i = \lambda_i \frac{||\underline{\hat{h}}_k^T \underline{v}_i||^2}{||\underline{\hat{h}}_k||^2}$ converges to zero, $i = 1, 2, \ldots, NL-1$. Substituting $\underline{\hat{h}}_k = \underline{\hat{h}}_{k-1} - \mu \bigtriangledown e_k(\underline{\hat{h}}_{k-1})$, and retaining only terms that are linear or constant in ϵ_k (i.e. assuming $\epsilon_k \ll 1$ near the solution), one obtains

$$\epsilon_k^i = \left(\frac{\lambda_i ||\underline{\hat{h}}_{k-1}^T \underline{v}_i||^2}{||\underline{\hat{h}}_{k-1}||^2}\right)\left(\frac{||\underline{\hat{h}}_{k-1}||^2}{||\underline{\hat{h}}_k||^2}\right)\left(1 - 2\frac{\mu\lambda_i}{||\underline{\hat{h}}_{k-1}||^2} + \frac{\mu^2\lambda_i^2}{||\underline{\hat{h}}_{k-1}||^4}\right)$$

which implies

$$\epsilon_k^i \le \epsilon_{k-1}^i \left(1 - \frac{\mu\lambda_i}{||\underline{\hat{h}}_{k-1}||^2}\right)^2 \tag{11}$$

where the last inequality comes from the fact that $||\underline{\hat{h}}_k||^2$ is nondecreasing as a function of k. This nondecreasing property can be derived as follows: note from the definition of $\bigtriangledown e_k(\underline{\hat{h}}_{k-1})$ that $\underline{\hat{h}}_{k-1}^H(\underline{\hat{h}}_k - \underline{\hat{h}}_{k-1}) = \underline{\hat{h}}_{k-1}^H \bigtriangledown e_k(\underline{\hat{h}}_{k-1}) = 0$, which implies

$$\begin{aligned} ||\underline{\hat{h}}_k||^2 &= ||\underline{\hat{h}}_k - \underline{\hat{h}}_{k-1} + \underline{\hat{h}}_{k-1}||^2 \\ &= ||\underline{\hat{h}}_k - \underline{\hat{h}}_{k-1}||^2 + ||\underline{\hat{h}}_{k-1}||^2 \ge ||\underline{\hat{h}}_{k-1}||^2. \end{aligned}$$

If the initial guess is chosen on the unit circle (i.e. $||\underline{\hat{h}}_0^2|| = 1$), the nondecreasing property and (11) can be used to show that convergence occurs in all of the modes if $\mu < \frac{2}{\lambda_{\max}}$. Since $\lambda_{\max}$ is difficult to obtain, the conservative estimate $\mu \le \frac{2}{tr(S_Z)}$ is used instead, where

$tr(S_Z) = L(N-1)\sum_{i=0}^{N-1} E|y_i(k)|^2 - LN(N-1)\sigma^2$, which can be readily estimated if the signal-to-noise ratio is known.

Note that the inequality (11) implies that the speed of convergence depends on $||\underline{\hat{h}}_{k-1}||^2$ in general. This suggests defining a normalized algorithm to eliminate the $||\underline{\hat{h}}_{k-1}||^2$ in the convergence condition. The normalized algorithm is obtained by using a variable gain factor $\mu||\underline{\hat{h}}_{k-1}||^2$ at the k^{th} step. Rewriting (6) with this "variable" gain factor yields

$$\underline{\hat{h}}_k = \underline{\hat{h}}_{k-1} - \mu\frac{(||\underline{\hat{h}}_{k-1}||^2 S_k^* \underline{\hat{h}}_{k-1} - (\underline{\hat{h}}_{k-1}^T S_k \underline{\hat{h}}_{k-1}^*)\underline{\hat{h}}_{k-1})}{||\underline{\hat{h}}_{k-1}||^2}. \tag{12}$$

and the convergence analysis follows as before to give

$$\epsilon_k^i \quad \leq \quad \epsilon_{k-1}^i (1-\mu\lambda_i)^2 \tag{13}$$

which implies convergence if $\mu < \frac{2}{\lambda_{\max}}$. Note that this normalized algorithm is similar to the algorithms of [12, 16], but here it has been motivated directly from the first order convergence analysis of the unnormalized stochastic gradient algorithm.

3.2.2 Rank of S_k and Convergence

Even in a noiseless system, when the empirical estimate S_k is used the algorithm will see a loss in rank relative to the rank $NL-1$ of the noiseless ensemble average S_Z.

To see that the rank of S_{k_Z} can be somewhat less than $NL-1$, consider the case where the channel input bits are all equal to $+1$ from time k down to time $k-2L+2$ (i.e. all of the channel input bits that affect S_k). By observing equation (5), one notes that the first L rows are equal, the second L rows are equal, etc., which guarantees a rank of no greater than N. In fact, the solution $\underline{h}^{opt}$ is now only constrained to satisfy the following set of equations:

$$\left(\sum_{l=0}^{L-1} h_i(l)\right)\left(\sum_{l=1}^{L-1} \underline{h}^{opt}[jL+l]\right) = \left(\sum_{l=0}^{L-1} h_j(l)\right)\left(\sum_{l=1}^{L-1} \underline{h}^{opt}[iL+l]\right) \qquad \forall i \neq j$$

which has a solution space of dimension $L+(N-1)(L-1)$, implying that the rank of S_k is only $N-1$. Similar results hold for other sequences of input bits. This loss of rank of S_k points out the requirement for the data bits to change in such a way that each of the modes is persistently excited to avoid convergence stagnation in an eigendirection.

3.3 Conjugate Gradient Algorithm

In this section, the application of conjugate gradient (CG) algorithms to the minimization of the Rayleigh quotient is reviewed. It is illustrated how data-adaptive versions of these algorithms can be implemented in $O(L^2N)$ flops for the oversampled blind channel identification problem. The two algorithms considered here are presented in the notation of [23] - for a more detailed discussion of the application of these algorithms to harmonic retrieval, see [13, 23].

The data-adaptive CG algorithms implemented here use the update $\underline{\tilde{h}}_{k+1} = \underline{\hat{h}}_k + \mu_k \underline{p}_k$ followed by the normalization $\underline{\hat{h}}_{k+1} = \frac{\underline{\tilde{h}}_{k+1}}{\|\underline{\tilde{h}}_{k+1}\|}$, where μ_k is chosen to minimize the Rayleigh quotient in direction $\underline{p}_k$ and is given by

$$\mu_k = \frac{-B + \sqrt{B^2 - 4CD}}{2D}$$

where

$$D \triangleq P_b(k)P_c(k) - P_a(k)P_d(k) \qquad B \triangleq P_b(k) - e_{k+1}(\underline{\hat{h}}_k)P_d(k) \qquad C \triangleq P_a(k) - e_{k+1}(\underline{\hat{h}}_k)P_c(k)$$

$$P_a(k) \triangleq \frac{\underline{p}_k^H S_{k+1}^* \underline{\hat{h}}_k}{\|\underline{\hat{h}}_k\|^2} \qquad P_b(k) \triangleq \frac{\underline{p}_k^H S_{k+1}^* \underline{p}_k}{\|\underline{\hat{h}}_k\|^2} \qquad P_c(k) \triangleq \frac{\underline{p}_k^H \underline{\hat{h}}_k}{\|\underline{\hat{h}}_k\|^2} \qquad P_d(k) \triangleq \frac{\underline{p}_k^H \underline{p}_k}{\|\underline{\hat{h}}_k\|^2}.$$

The new search direction $\underline{p}_k$ is selected by $\underline{p}_k = -\bigtriangledown e_{k+1}(\underline{\hat{h}}_k) + b_k \underline{p}_{k-1}$. In Chen et. al. [13], b_k is chosen such that $\underline{p}_k^H S^* \underline{p}_{k-1} = 0$, which yields

$$b_k = \frac{\nabla^H e_{k+1}(\underline{\hat{h}}_k) S_k^* \underline{p}_{k-1}}{\underline{p}_{k-1}^H S_k^* \underline{p}_{k-1}}.$$

On the other hand, Yang et. al. [23] choose b_k such that $\underline{p}_k^H H \underline{p}_{k-1} = 0$ where H is the Hessian of the Rayleigh quotient. This yields

$$b_k = \frac{-[\nabla^H e_{k+1}(\underline{\hat{h}}_k) S_k^* \underline{p}_{k-1} + \| \nabla e_{k+1}(\underline{\hat{h}}_k)\|^2 \underline{\hat{h}}_k^H \underline{p}_{k-1}]}{\underline{p}_{k-1}^H S_k^* \underline{p}_{k-1} - e_{k+1}(\underline{\hat{h}}_k)\|\underline{p}_{k-1}\|^2}.$$

The algorithms above differ from the algorithms presented in [13, 23]; similarly to the stochastic gradient algorithms presented in Section 3.1, they use only the instantaneous data estimates of S_k^* to step once per symbol period. The complexity of the data-adaptive CG algorithms is primarily due to the matrix-vector multiplications $S_k^* \underline{p}_{k-1}$ and $S_k^* \underline{\hat{h}}_{k-1}$ which can be done in $O(L^2 N)$ operations as per (9). Note the nice feature that the conjugate gradient algorithms do not require specification of an additional gain factor.

4 Adaptive Algorithm for a Known Spreading Code

When NL is fairly large, the stochastic algorithm may take a long time to converge because of the high dimensionality (NL) of the search space. However, in [10], it is shown how knowledge of the transmit filter can reduce this dimensionality. In the case where the spreading code is the known transmit filter as in (2), the search space dimension is reduced to $NL - N + 1$. Because the DS/SS application considered here assumes large N and fairly small L, this saving of $(N-1)$ dimensions promises to greatly reduce the dimensionality of the problem with an associated improvement in performance. When the spreading sequence $\{a_k\}$ is fixed and known, the subchannel responses depend only on the unknown channel response $\underline{g} \triangleq [g_0 \; g_1 \ldots g_{NL-N}]$. As a function of $\underline{\hat{g}}$, the estimate of $\underline{g}$, the objective function (3) becomes:

$$\begin{aligned}\mathcal{E}(\underline{\hat{g}}) &= \sum_{m=0}^{N-2}\sum_{n=m+1}^{N-1} E[|\sum_{i=0}^{L-1}\sum_{u=0}^{N-1} a_u \hat{g}_{iN+m-u} y_n(k-i) \\ &- \sum_{j=0}^{L-1}\sum_{u=0}^{N-1} a_u \hat{g}_{jN+n-u} y_m(k-j)|^2]\end{aligned}$$

with the constraint $||\underline{\hat{g}}||^2 = 1$. In [10], it was shown that

$$\mathcal{E}(\underline{\hat{g}}) = \underline{\hat{g}}^T Q S Q^H \underline{\hat{g}}^*$$

where Q, an $(L-1)N+1$ by LN matrix of spreading coefficients, is defined as $Q \triangleq [Q_0\, Q_1 \ldots Q_{N-1}]$, where

$$Q_n \triangleq [\underline{q}_n(0)\ \underline{q}_n(1) \ldots \underline{q}_n(L-1)]$$

with length $LN - N + 1$ vectors

$$\underline{q}_n(k) \triangleq [a_{kN+n}\ a_{kN+n-1} \ldots a_{kN+n-LN+N}]^T$$

where a_l is defined to be zero for $l \notin [0, N-1]$. The stochastic gradient algorithm can be applied as before to yield the normalized update formula

$$\underline{\hat{g}}_k = \underline{\hat{g}}_{k-1} - \mu \frac{(||\underline{\hat{g}}_{k-1}||^2 Q^* S_k^* Q^T \underline{\hat{g}}_{k-1} - (\underline{\hat{g}}_{k-1}^T Q S_k Q^H \underline{\hat{g}}_{k-1}^*)\underline{\hat{g}}_{k-1})}{||\underline{\hat{g}}_{k-1}||^2} \tag{14}$$

but the complexity of this algorithm is $O(L^3N^3)$ if implemented with a brute force matrix multiplication, and still $O(L^2N^3)$ if one uses the property that each $\underline{q}_n(k)$ contains at most N nonzero entries. The key is to recognize that $Q^* S_k^* Q^T \underline{\hat{g}}_{k-1} = \triangledown \mathcal{E}_k(\underline{\hat{g}}_{k-1})$ where the gradient in this case is with respect to the elements of $\underline{\hat{g}}_{k-1}$. Applying the chain rule for gradients [24, pg. 134], element j of $Q^* S_k^* Q^T \underline{\hat{g}}_{k-1}$ is given by

$$
\begin{aligned}
\frac{\partial(\underline{\hat{g}}_{k-1}^T Q S_k Q^H \underline{\hat{g}}_{k-1}^*)}{\partial \underline{\hat{g}}_{k-1}[j]} &= \sum_{m=0}^{N-1}\sum_{l=0}^{L-1} \frac{\partial(\underline{\hat{g}}_{k-1}^T Q S_k Q^H \underline{\hat{g}}_{k-1}^*)}{\partial \underline{\hat{h}}_{k-1}[mL+l]} \frac{\partial \underline{\hat{h}}_{k-1}[mL+l]}{\partial \underline{\hat{g}}_{k-1}[j]} \\
&= \sum_{m=0}^{N-1}\sum_{l=0}^{L-1} \frac{\partial(\underline{\hat{h}}_{k-1} S_k \underline{\hat{h}}_{k-1}^*)}{\partial \underline{\hat{h}}_{k-1}[mL+l]} a_{lN+m-j} \qquad (15)
\end{aligned}
$$

The NL partial derivatives required on the right side of (15) are obtained from (9). Also, for a given j, only N of the terms on the right side of (15) are non-zero. Thus, the procedure at each step is:

1. Form the gradient with respect to $\underline{\hat{h}}_{k-1}$ using (9) in $\min\{O(L^2N), O(LN^2)\}$ operations.
2. Apply (15) to get the gradient with respect to $\underline{\hat{g}}_{k-1}$ in $O(LN^2)$ operations.
3. Use the result of step 2 in the numerator of (14) to update $\underline{\hat{g}}_{k-1}$.

The $O(LN^2)$ operations required for this algorithm is larger than for the stochastic gradient algorithm when the knowledge of the transmit filter is not used. Finally, note that the conjugate gradient algorithms can take advantage of the transmit filter knowledge in an analogous way to the stochastic gradient algorithm.

5 Adaptive Algorithm Using Trellis Searching

In this section, a joint data and channel estimation algorithm due to Seshadri [22] is extended to oversampled systems.

5.1 Joint Data/Channel Data Estimation

It is well known that the maximum likelihood sequence estimator for a discrete channel with *known* impulse response of duration L symbol periods can be implemented with a Viterbi Algorithm with 2^{L-1} states, one state for each possible set of the previous $L-1$ data bits [25, pg. 610]. The Euclidean distance metric (equivalent to likelihood for the AWGN case) for a path in the trellis originating at state i at time k is updated as follows: the states between which the path travels determine an estimate of the input bit sequence $\underline{\hat{b}}_k = [\hat{b}_k \; \hat{b}_{k-1} \ldots \hat{b}_{k-L+1}]$. This estimate is used to evaluate the path metric by

$$\psi_k^i(\underline{\hat{b}}_k) = \beta_{k-1}^i - (y(k) - \sum_{l=0}^{L-1} \underline{\hat{b}}_k[l] h(l))^2$$

where β_{k-1}^i is the accumulated metric of state i at time $k-1$ (or, alternatively, the metric of the path that was retained at state i at time $k-1$), $y(k)$ is the data sample at time k, and $h(l)$ is the channel impulse response at delay l. Two paths enter each of the states at time k. At each state, the path with the highest metric is kept and the other discarded.

The algorithm considered in [22] extends the above idea to channels where the ISI is unknown. The algorithm is nearly identical to the algorithm described above except that multiple paths (the number denoted M) are kept for each state at each time. In this case, each path contains and updates its own estimate of the channel. Metrics are now associated with a path and are calculated for a given path based on its channel estimate instead of the actual channel. In equation form, the metric for the j^{th} path from state i with bit estimate $\underline{\hat{b}}_k$ is given by

$$\psi_k^{ij}(\underline{\hat{b}}_k) = \beta_{k-1}^{ij} - (y(k) - \sum_{l=0}^{L-1} \underline{\hat{b}}_k[l] \hat{h}^{ij}(l))^2$$

where β_{k-1}^{ij} is the metric of the j^{th} path of state i at time $k-1$ and $\hat{h}^{ij}(l)$ is the channel estimate of the j^{th} path from state i for delay l. Note that $2M$ paths enter each state at time k; the M paths with the best metrics are retained and the others discarded. For the M surviving paths in each state, the channel estimate is updated using an LMS algorithm.

5.2 Extension to Oversampled Systems

As discussed in Section 2.2, the oversampled channel can be viewed as N subchannels operating at rate T_s, which all need to be equalized. Since the method of [22] is able to equalize a channel that is not oversampled, each subchannel could be separately equalized with this algorithm and the results combined (although there would be a sign ambiguity for *each* subchannel). Of course, this would be suboptimal since it is known that the input to each of the subchannels is the same. Instead, a single Viterbi Algorithm is used with 2^{L-1} states and M paths kept per state, each with its estimate of all of the subchannels. The metric for the j^{th} path from state i corresponding to input $\hat{\underline{b}}_k$ is given by

$$\psi_k^{ij}(\hat{\underline{b}}_k) = \beta_{k-1}^{ij} - \sum_{m=0}^{N-1}(y_m(k) - \sum_{l=0}^{L-1}\hat{\underline{b}}_k[l]\hat{h}_m^{ij}(l))^2$$

where β_{k-1}^{ij} is the metric of the j^{th} path of state i at time $k-1$, and $\hat{h}_m^{ij}(l)$ is the channel estimate of the j^{th} path from state i for subchannel m at delay l.

A method of updating the channel estimate for a given path must now be considered. Although the LMS algorithm could be used just as it was in [22], a different method is proposed here for the following reason. Stochastic gradient algorithms (like LMS) are largely utilized to reduce complexity as has been done in the previous sections here. In the channel estimation portion of the trellis-searching algorithm, however, the minimal mean square error for ensemble averages occurs for the correct path at $h_n^{opt}(l) = R_n(l) = E[y_n(k)\hat{\underline{b}}_k[l]]$. which is easily approximated. Thus, instead of using a gradient search algorithm, for each path this ensemble average can be approximated by the sample mean

$$\hat{\underline{h}}_n^{ij}(l) = \hat{R}_n(l) = \frac{1}{T}\sum_{k=0}^{T-1} y_n(k)\hat{\underline{b}}_k[l].$$

For a channel that does not vary with time, this appears to be an attractive alternative to the LMS algorithm. If the considered channel does vary with time, a “forgetting factor” can be added to the distribution so that it can adapt to new statistics. Note the added benefit that there is no need for the choice of a gain factor.

Finally, the trellis-searching algorithm has one large advantage over the SRM-based algorithms. Not only can it be easily modified to take advantage of a known spreading code, but it can also perform blind channel identification when the (known) spreading code varies across symbol periods as in many practical systems. Since the focus here is on comparisons with the SRM-based algorithms, this will not be addressed further.

6 Numerical Results

6.1 Implementation Issues

It is important to point out the complexity and limitations of the algorithms before the numerical results are presented. Both the "standard" stochastic gradient algorithm and the "normalized" stochastic gradient algorithm along with both conjugate gradient algorithm algorithms based on subchannel response matching algorithm require only $O(L^2 N)$ operations at each iteration. As mentioned earlier, the algorithms have the drawback that they are only correct to within a complex constant; in other words, they have both an amplitude and phase ambiguity. As discussed in [26], this is not a severe limitation. Furthermore, the algorithms only identify the channel and do not decode the transmitted bits. One possible method of doing so is to run a Viterbi Algorithm concurrent with the channel estimator. At each symbol period, path metrics in the trellis are evaluated using the current channel estimate. Note that this adds $O(2^L N)$ operations for a total complexity that is $\max\{O(2^L N), O(L^2 N)\}$.

The trellis-searching algorithm requires $O(L2^L NM)$ operations per symbol, significantly more than the stochastic gradient algorithm, but also decodes the bits as it runs with no need for an additional sequence estimator. Note that the algorithm suffers from a sign ambiguity. It cannot distinguish between the true bit stream across the true channel and the negative of the true bit stream across the negative of the true channel because they both give the same outputs. Thus, some sort of differential encoding at the transmitter is required [25, pg. 266].

6.2 Simulation Results

The algorithms presented in the previous sections are compared in this section through simulation. As in [11], the normalized mean squared error at iteration k is defined by

$$\mathrm{NMSE}_k = \left\| \frac{\hat{\underline{h}}_k}{\|\hat{\underline{h}}_k\|} - \frac{\underline{h}}{\|\underline{h}\|} \right\|^2 .$$

The goal is to determine both how the algorithms compare to each other and also how these results compare to results reported on similar problems such as those from Pisarenko's harmonic retrieval. Each of the algorithms certainly can be tweaked to obtain better performance. In particular, the trellis-searching algorithms have higher complexity than the SRM-based algorithms. Since the SRM-based algorithms calculate $e_k(\hat{\underline{h}}_{k-1})$ at each step, it could be used to implement a multiple starting point iteration to close both the performance and complexity gap [27]. Slight extensions, however, will be ignored and only the base algorithms considered.

In the first set of numerical results, two channels are considered with zeroes as tabulated below. For both channels, $N = 4, L = 5$, which yields $\mu = 0.14$ from the convergence analysis in Section 3.2.

Subchannel	*Good*	*Bad*
0	0.353 ± 0.353 j 0.259 ± 0.150 j	0.636 ±0.636j 0.692 ± 0.400j
1	-0.104 ± 0.590 j 1.691 ± 0.615 j	-0.121 ± 0.689j 1.122 ± 0.410j
2	0.069 ± 0.393 j -1.221 ± 1.455 j	0.148 ± 0.837j 1.034 ± 0.376j
3	0.752 ± 2.067 j -0.800 ± 1.385 j	0.393 ± 1.081j -0.525 ± 0.909j

Note in Figure 1 how the zeroes in the "Bad" channel are closer to the unit circle than the zeroes for the "Good" channel. Also note that on the "Bad" channel, subchannels one and two

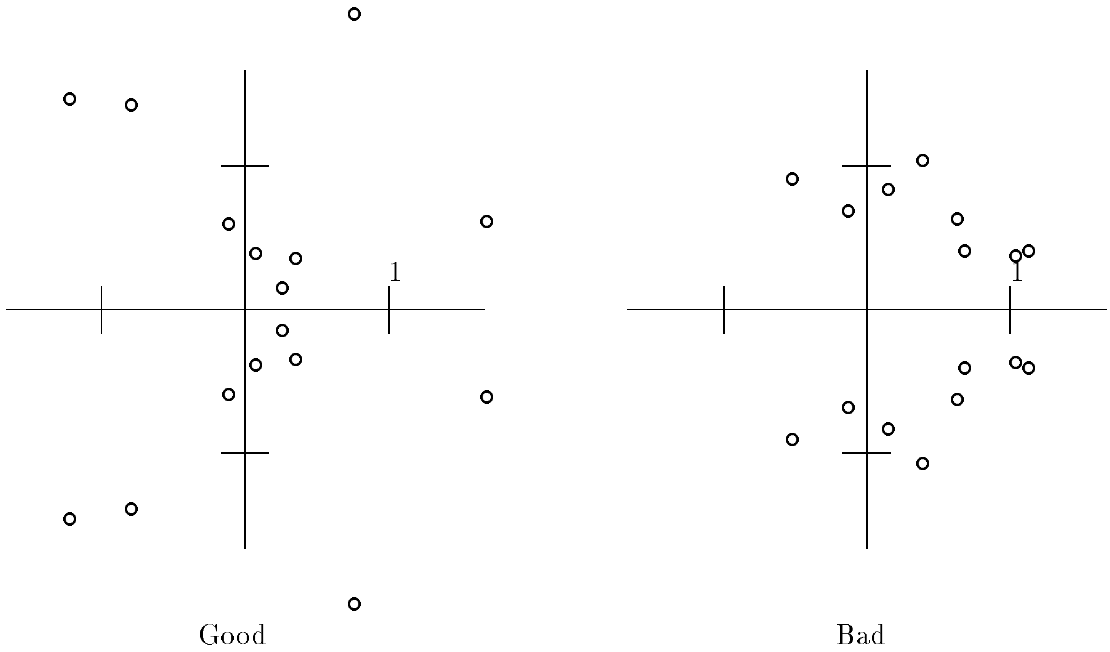

Figure 1: Zeroes of the “Good” and “Bad” Channels

have a set of zeroes that fall nearly on top of each other, making it plausible that identification is more difficult.

The “standard” stochastic gradient algorithms and “normalized” stochastic gradient algorithms were run on channels “Good” and “Bad” at signal-to-noise ratios of 20 dB and 10 dB. The best gain factors from among $\mu = 0.06, \mu = 0.14, \mu = 0.30$ (the gain factor the convergence analysis suggests and one on either side) were selected and the results are shown in Figure 2 through 5 along with the other SRM-based algorithms. The results are averaged over 200 sample trials. Only the data-adaptive version of the conjugate gradient algorithm of Chen et. al. [13] is shown as the data-adaptive version of the conjugate gradient algorithm of Yang et. al. [23] was found to have nearly identical performance. Comparisons with the trellis-searching algorithms are made in Figures 6 and 7.

Key to the Figure abbreviations:

1. "SG": The standard stochastic gradient algorithm.
2. "Norm": The normalized stochastic gradient algorithm.
3. "SRM": The off-line $O(L^3N^3)$ algorithm of [4] where S_k is obtained at time k by averaging all of the data through time k to estimate ensemble averages.
4. "Trellis LMS": The adaptive trellis-searching algorithm where channel updating is done with an LMS algorithm with (empirically chosen) gain factor $\mu = 0.02$.
5. "Trellis Mean": The adaptive trellis-searching algorithm where channel updating is done via sample mean.
6. "Conj Grad": The data-adaptive conjugate gradient channel identification algorithm.

A few things are immediately apparent from Figures 2 through 5. First and foremost is that all of the SRM-based data-adaptive algorithms (SG, Norm, CG) perform *much* worse than the "off-line" SRM algorithm. The conjugate gradient algorithm shows faster initial convergence for both "Good" and "Bad" channels at both signal-to-noise ratios, but for SNR=10 dB, its performance levels off very quickly and is surpassed slightly by the stochastic gradient algorithms. Note in particular how the normalized stochastic gradient algorithm with properly chosen gain factor comes very close in performance to the conjugate gradient algorithm. In comparison with the trellis-searching algorithms, one notes that for $M = 2$ the trellis-searching algorithms perform quite poorly. For $M = 8$, a complexity much greater than the SRM-based algorithms, the trellis-searching algorithms perform markedly better, especially for the "Bad" channel as they suffer no performance impairment based on the placement of the zeroes of the channels. Initial convergence is especially impressive. Although the trellis-searching algorithm that uses sample means for updating the channel estimates has the advantage of not requiring the choice of a gain factor as in the trellis-searching algorithm with the LMS-based channel updating, the performance advantage that was expected is not in evidence. In the case displayed in Figure 6, it actually performed slightly worse. Since the LMS channel updating strategy will also track better for channels that vary with time, it is concluded that it is superior to the sample mean channel updating strategy.

There are some effects not evident from the plots - whereas the average values displayed are representative of all of the sample runs for the SRM-based stochastic gradient algorithms, the performance curves for the other methods show wider variation. For the conjugate gradient algorithm, the error on a given sample run tends to bounce around a lot in the steady state with the smooth displayed curves being a byproduct of averaging over many sample runs. The only moderately good performance of the trellis-searching algorithm with either channel updating strategy is caused by outstanding performance on certain sample paths and a complete lack of convergence in others. This was an unexpected phenomenon as it was not noted in [22].

Finally, the benefit of knowledge of the transmit filter is considered. In this case, a channel with $N = 8, L = 3$ is formed by convolving a length 17 randomly generated complex channel with exponentially decreasing average power (drop from peak to e^{-1} at sample 8) with a length 8 spreading code. Results for the normalized stochastic gradient algorithm and normalized known transmit filter algorithms are shown in Figure 8. Note that the known channel transmit filter algorithm shows better initial convergence and steady-state performance but at a higher computational cost per step as discussed in Section 4.

7 Summary and Conclusions

In this paper, it has been shown how a direct-sequence spread spectrum system sampled at the chip rate can be mapped to a discrete oversampled system with the spreading code as the transmit filter. This motivates the search for efficient data-adaptive blind channel identification algorithms when the oversampling rate is high. Stochastic gradient and conjugate gradient algorithms have been presented that are only linear in the oversampling rate. However, if one wants to take advantage of the knowledge of the transmit filter to reduce the dimensionality of the search space, the algorithms become quadratic in the oversampling rate. Finally, for comparison a trellis-searching algorithm for performing non-oversampled equalization has been extended to the oversampled case.

Numerical results suggest that there is quite a bit of performance loss between the off-line SRM algorithm and the data-adaptive implementations of it. Furthermore, it appears that the gain of the conjugate gradient algorithms over the stochastic gradient algorithms is not as prominent as was shown in the Pisarenko's harmonic retrieval; however, the conjugate gradient algorithm showed markedly faster initial convergence in general. The huge gains shown in [10] for the known transmit filter are also not evident for the case that was considered in this paper. In general, the trellis-searching algorithms perform much better than the SRM-based algorithms while still only being linear in the oversampling rate. However, while still only linear in the oversampling rate, the complexity is much higher if a large number of paths are kept per state, which appears to be a prerequisite for acceptable performance. If the SRM-based algorithms are allowed to close this complexity gap, they can take into account the current error estimate at each iteration and use a multiple starting point SRM-based algorithm to narrow the performance gap considerably.

References

[1] S. Haykin, Ed., *Blind Deconvolution,* Prentice Hall, Englewood Cliffs, 1994.

[2] H. Liu, G. Xu, and L. Tong, "A Deterministic Approach to Blind Equalization," *Conference Record of the Twenty-Seventh Asilomar Conference on Signals, Systems, and Computers (1993),* pp. 751-755.

[3] L. Baccalá and S. Roy, "Time-Domain Blind Channel Identification Algorithms," *Proceedings of the 1994 Conference on Information Science and Systems,* Vol. 2, pp. 863-867.

[4] S. Schell and D. Smith, "Improved Performance of Blind Equalization Using Prior Knowledge of Transmitter Filter," *Conference Record of the 1994 IEEE Military Communications Conference,* Vol. 1, pp. 128-132.

[5] L. Tong, G. Xu, and T. Kailath, "Blind Identification and Equalization Based on Second-Order Statistics: A Time Domain Approach," *IEEE Transactions on Information Theory,* Vol. 40, pp. 340-349, March 1994.

[6] E. Moulines, P. Duhamel, J. Cardoso, and S. Mayrargue, "Subspace Methods for the Blind Identification of Multichannel FIR Filters," *IEEE Transactions on Signal Processing,* Vol. 43, pp. 516-525, February 1995.

[7] K. Meraim, P. Duhamel, D. Gesbert, P. Loubaton, S. Mayrargue, E. Moulines, and D. Slock, "Prediction Error Methods for Time-Domain Blind Identification of Multichannel FIR Filters," *Proceedings of the 1995 International Conference on Acoustics, Speech, and Signal Processing,* Vol. 3, pp. 1968-1971.

[8] G. Giannakis and S. Halford, "Performance Analysis of Blind Equalizers Based on Cyclostationary Statistics," *Proceedings of the 1994 Conference on Information Science and Systems,* Vol. 2, pp. 873-876.

[9] D. Slock and C. Papadias, "Further Results on Blind Identification and Equalization of Multiple FIR Channels ," *Proceedings of the 1995 International Conference on Acoustics, Speech, and Signal Processing,* Vol. 3, pp. 1964-1967.

[10] S. Schell, D. Smith, and S. Roy, "Blind Channel Identification using Subchannel Response Matching," *Proceedings of the 1994 Conference on Information Science and Systems,* Vol. 2, pp. 858-862.

[11] L. Baccalá and S. Roy, "A New Blind Time-Domain Channel Identification Method Based on Cyclostationarity," *IEEE Signal Processing Letters,* Vol. 1, pp. 89-91, June 1994.

[12] P. Thompson, "An Adaptive Spectral Analysis Technique for Unbiased Frequency Estimation in the Presence of White Noise," *Conference Record of the Thirteenth Asilomar Conference on Circuits, Systems, and Computers (1979),* pp. 529-533.

[13] H. Chen, T. Sarkar, S. Dianat, J. Brulé, "Adaptive Spectral Estimation by the Conjugate Gradient Method," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* Vol. 34, pp. 272-284, April 1986.

[14] M. Larimore and R. Calvert, "Convergence Studies of Thompson's Unbiased Adaptive Spectral Estimator," *Conference Record of the Fourteenth Asilomar Conference on Circuits, Systems, and Computers (1980)*, pp. 258-262.

[15] M. Larimore, "Adaptive Convergence of Spectral Estimation Based on Pisarenko Harmonic Retrieval," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 31, pp. 955-962, August 1983.

[16] V. Reddy, B. Egardt, and T. Kailath, "Least Squares Type Algorithm for Adaptive Implementation of Pisarenko's Harmonic Retrieval Method," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 30, pp. 399-405, June 1982.

[17] G. Mathew, S. Dasgupta, and V. Reddy, "Improved Newton-Type Algorithm for Adaptive Implementation of Pisarenko's Harmonic Retrieval Method and Its Convergence Analysis," *IEEE Transactions on Signal Processing*, Vol. 42, pp. 434-437, February 1994.

[18] R. Schmidt, "Multiple Emitter Location and Signal Parameter Estimation," *IEEE Transactions on Antennas and Propagation*, Vol. 34, pp. 276-280, March 1986.

[19] M. Kaveh and A. Barabell, "The Statistical Performance of the MUSIC and the Minimum-Norm Algorithms in Resolving Plane Waves in Noise," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 34, pp. 331-341, April 1986.

[20] D. Torrieri and K. Bakhru, "The Recursive Suppression Algorithm for Adaptive Superresolution," *IEEE Transactions on Antennas and Propagation*, Vol. 40, pp. 926-932, August 1992.

[21] J. Yang and M. Kaveh, "Adaptive Eigensubspace Algorithms for Direction or Frequency Estimation and Tracking," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 36, pp. 241-251, February 1988.

[22] N. Seshadri, "Joint Data and Channel Estimation Using Blind Trellis Search Techniques," *IEEE Transactions on Communications*, Vol. 42, pp. 1000-1011, February/March/April 1994.

[23] X. Yang, T. Sarkar, and E. Arvas, "A Survery of Conjugate Gradient Algorithms for Solution of Extreme Eigen-Problems of a Symmetric Matrix," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 37, pp. 1550-1555, October 1989.

[24] J. Marsden and A. Tromba, *Vector Calculus*, W. H. Freeman, New York, 1988.

[25] J. Proakis, *Digital Communications*, McGraw-Hill, New York, 1989.

[26] S. Verdú, B. Anderson, and R. Kennedy, "Blind Equalization without Gain Identification," *IEEE Transactions on Information Theory*, Vol. 39, pp. 292-297, January 1993.

[27] D. Goeckel, A. Hero III, and W. Stark, "Blind Channel Identification for Direct-Sequence Spread-Spectrum Systems," *Conference Record of the 1995 IEEE Military Communications Conference*, pp. 368-372.

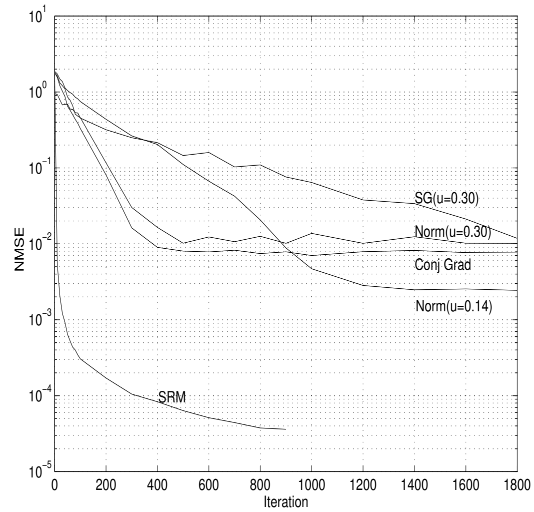

Figure 2: Comparison of the SRM-based algorithms for the "Good" channel, SNR=20 dB

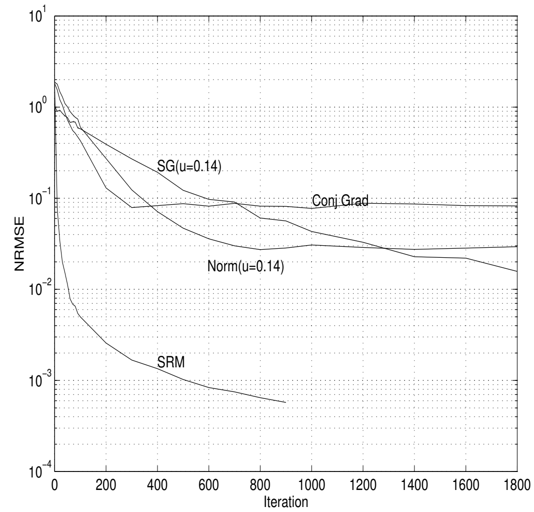

Figure 3: Comparison of the SRM-based algorithms for the "Good" channel, SNR=10 dB

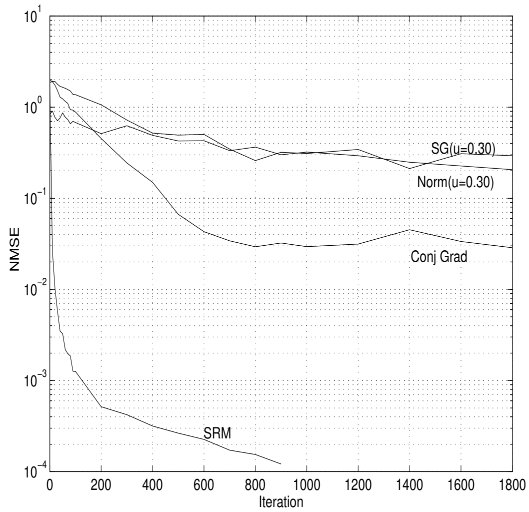

Figure 4: Comparison of the SRM-based algorithms for the "Bad" channel, SNR=20 dB

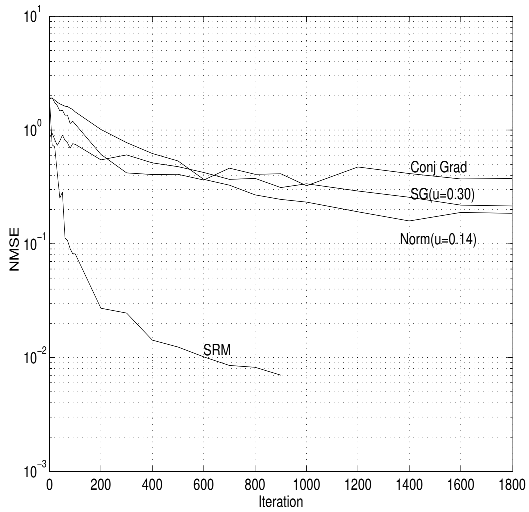

Figure 5: Comparison of the SRM-based algorithms for the "Bad" channel, SNR=10 dB

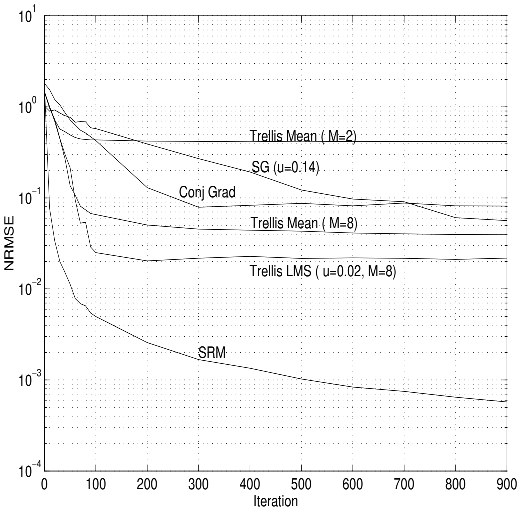

Figure 6: Comparison of SRM-based stochastic gradient algorithms with the trellis searching algorithm, "Good" channel, SNR=10 dB

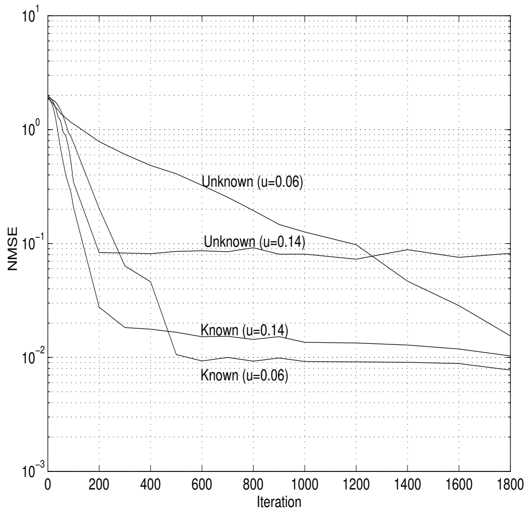

Figure 8: Comparison of the normalized stochastic gradient algorithm with that when the transmit filter is known, $N = 8, L = 3$, SNR = 10 dB

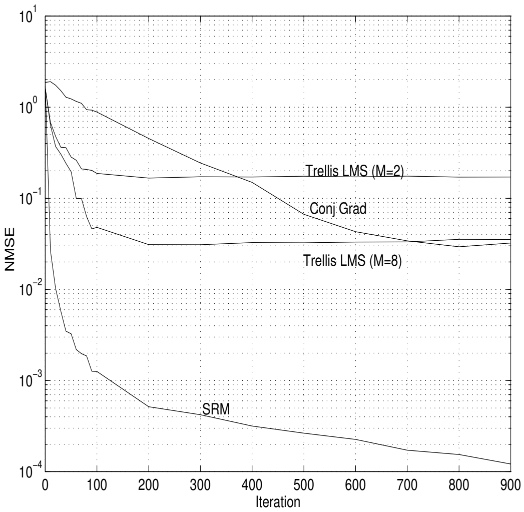

Figure 7: Comparison of SRM-based stochastic gradient algorithms with the trellis searching algorithm, "Bad" channel, SNR=20 dB