Overloaded Array Processing with Spatially Reduced Search Joint Detection

by

James Hicks

Thesis submitted to the faculty of the

Virginia Polytechnic Institute & State University

in partial fulfillment of the requirements of the degree

MASTER OF SCIENCE
in

Electrical Engineering

Approved:

____

Dr. Jeffrey H. Reed, Chairman

______________________ ______________________

Dr. W. H. Tranter Dr. Brian D. Woerner

May 10, 2000

Blacksburg, Virginia

Keywords: Overloaded Array, Joint Detection, Antenna Array, Interference Mitigation

Overloaded Array Processing with Spatially Reduced Search Joint Detection

James Hicks

(ABSTRACT)

An antenna array is overloaded when the number of cochannel signals in its operating environment exceeds the number of elements. Conventional space-time array processing for narrow-band signals fails in overloaded environments. Overloaded array processing (OAP) is most difficult when signals impinging on the array are near equal power, have tight excess bandwidth, and are of identical signal type. In this thesis, we first demonstrate how OAP is theoretically possible with the joint maximum likelihood (JML) receiver. However, for even a modest number of interfering signals, the JML receiver's computational complexity quickly exceeds the real-time ability of any computer. This thesis proposes an iterative joint detection technique, Spatially Reduced Search Joint Detection, (SRSJD), which approximates the JML receiver while reducing its computational complexity by several orders of magnitude. This complexity reduction is achieved by first exploiting spatial separation between interfering signals with a linear pre-processing stage, and second, performing iterative joint detection with a (possibly) tail-biting and "time"-varying trellis. The algorithm is sub-optimal but is demonstrated to well approximate the optimum receiver in modest signal to interference ratios. SRSJD is shown to demodulate over 2M zero excess bandwidth synchronous QPSK signals with an M element array. Also, this thesis investigates a temporal processing technique similar to SRSJD, Temporally Reduced Search Joint Detection (TRSJD), that separates co-channel, asynchronous, partial response signals. The technique is demonstrated to separate two near equal power QPSK signals with r= .35 root raised-cosine pulse shapes.

ACKNOWLEDGMENTS

I would like to thank all those who have made this thesis possible: my advisor, Dr. Jeffrey H. Reed for his support, encouragement, tutelage, and advice; his breadth and scope of knowledge about interference mitigation techniques was invaluable while scooping the research in this thesis; Dr. Brian Agee for his perspicacious questions and indirect theoretical contributions to this work; Dr. Robert Boyle who first identified the Overloaded Array Processing problem within MPRG, closely managed our research in this area, and first proposed sub-optimal joint detection as a possible approach; Dr. William Ebel for identifying the connection between the approach taken in this thesis and similar work conducted by the coding theory community; Dr. William H. Tranter and Dr. Brian D. Woerner at MPRG as well as B. P. Paris at GMU whose commitment to education and research in wireless communications have given me the fundamentals required for this research; Tom Biedka for his advice and general guidance; Saffet Bayram, who helped in this research effort; and finally, Daniel Sharp and Aristos Dimitriou for their practical input and encouragement. Further, I would like to thank the graduate students at MPRG who have shared a legacy of outstanding research and whose stimulating discussions and ideas have contributed to this research including Dr. Matt Valenti, and Dr. Neiyer Correal-Mendoza, Ran Gozali, Fakhrul Alam, Bruce Puckett, Bror Peterson, and Rich Ertle. Most importantly, I would like to thank my wife, Rupashree Majali, whose love, support, and consideration have made this thesis possible.

CONTENTS

TABLE OF FIGURES

Chapter 1: INTRODUCTION

The steady growth of wireless market has created a huge demand for signal processing expertise. Increased demand for mobility and the reduced cost of telephone infrastructure has fueled the steady growth of the wireless market over the past decade. In 1997, more people signed up for mobile service than wire-line service. By 2003, wireless telephony is expected to reach 40% market penetration in the US, reaching a level of 110 million subscribers. Meanwhile, in Europe many countries have already exceeded 50% market penetration [47, 48].

Demand for increased capacity and increased data rates has been a major goal for mobile communications research. There are many ways of increasing capacity of a cellular system: more efficient data compression, increased spectrum usage, and the placement of multiple antennas at the receiver. Increasing system capacity by increasing the number of antennas requires very sophisticated signal processing, which must account for a variety of channel impairments: multipath (e.g. echoes), interference, and noise. This technique is commonly referred to as array processing or space-time adaptive processing (STAP) [39]. STAP for mobile communications has received a great deal of attention in the past decade, and great advances in this area have been made.

In any communication system the interference environment at hand determines the best choice of signal processing and receiver architecture. The signal to interference ratio (SIR) is *not* a complete description of an interference environment. A given SIR can involve a large number of small power interferers, or a small number of higher power users. In the former, the interference can be accurately modeled as Gaussian noise. In the latter, this assumption is not valid. We define an overloaded environment as one where the number of co-channel transmitters is greater than the number of antenna elements at the receiver, Overloaded Array Processing (OLAP) is most difficult when the interfering signals are near equal power, and have the tight excess bandwidth that modern communication protocols specify (e.g. IS-136).

Array processing in overloaded environments requires different considerations than traditional STAP. Most STAP receivers apply linear filtering techniques that break down in overloaded environments. A common belief amongst the array processing community is that signal extraction in an overloaded environment is not possible. However, recent research in this area suggests that it is. Array processing algorithms can be classified into two categories according to how they treat interference: interference rejection and multi-user detection. In the former, interfering signals are treated as noise that must be suppressed. In the latter, interfering signals are jointly estimated. Recent work suggests that multi-user detection performs better than interference rejection at a greatly increased cost in receiver complexity.

This thesis develops overloaded array processing algorithms with an emphasis on narrow-band signals. The question might arise: why focus on signal processing for narrowband signals? Despite the recent popularity of spread spectrum techniques, narrow-band standards are widely deployed and are expected to stay in existence beyond the next 10 years. In particular, an emerging 2.5G cellular standard, EDGE (Enhanced Data-rates for Global Evolution), will upgrade and unify the current digital standards in the US and Europe. The standard has already been accepted by both the European Telecom Standards Institute (ETSI) and the universal wireless communications consortium (UWCC) (a US standards committee). This standard has been well received by service providers and vendors alike and is expected to be deployed over the next year as upgrades to existing systems. Meeting the aggressive capacity and data-rate requirements of EDGE may require some sophisticated array processing. In short, narrowband communications is still very much alive.

The first part of this thesis provides an extensive survey of array processing techniques encompassing over 200 papers in the field. Thus far, few algorithms demonstrate a potential for overloaded array processing, and out of these, none have been applied in the literature for this purpose. In this expository second chapter, we first argue that overloaded array processing is indeed possible for known channels. The central contribution of this thesis is an algorithm that can perform OLAP with achievable complexity. This algorithm is presented in its simplest form in chapter 4. Chapter 3 provides sufficient background in trellis based processing: the Maximum-Likelihood Sequence Estimation (MLSE) for Viterbi Equalization (VEQ), Delayed Decision Feedback Sequence Estimation (DDFSE) as a method of reduced complexity VEQ, and finally, Tail-Biting MLSE (TB-MLSE) for Viterbi Equalization of circularly-convolutional channels. The algorithm presented in chapter 4 is only applicable for environments with symbol-synchronous signals. In chapter 5, we show how the approach can be extended to the asynchronous case. Finally, chapter 6 concludes the thesis with simulation results.

Chapter 2: ARRAY PROCESSING BACKGROUND

Increased capacity has long been a motivating factor behind the use of antenna arrays in communication systems. A natural question arises: what is the fundamental limit on the number of equal power users (interferers) that may be separated with an M element array? It is often thought that an M element array can separate a maximum of M equal power signals. This is because the majority of signal extraction algorithms found in the literature are based on linear filtering techniques. However, several interference rejection algorithms for single antenna receivers can extract a desired signal in the presence of multiple interferers. Moreover, recent contributions in antenna array research suggest that the number is much greater.

The capacity that any antenna array can achieve is highly dependent on the chosen receiver architecture. There are two basic approaches to dealing with interference in antenna arrays: interference rejection and joint detection. In the former, interference is treated as noise and is suppressed by the receiver. The receiver only extracts the (SOI). In the latter, interference is treated as a signal to be estimated. In a multi-user system, the benefits of joint detection are obvious. All interfering co-channel users are SOIs and the optimal receiver will demodulate all users simultaneously. However, if interference is not of direct interest, recent research suggests that multi-user detection will still out-perform interference suppression.

In the next section, we will first introduce the concept of the Joint Maximum A posteriori Probability Detector. In most cases, this receiver is not practical to implement, but it allows us to answer fundamental questions concerning capacity. Then we will focus on the limitations of more practical receivers. In our survey, we will first focus on the signal extraction techniques that employ a single receiver antenna and then move on to those techniques that employ receiver structures with multiple antennas.

2.1 THE OPTIMUM MULTI-ELEMENT RECEIVER

Figure 2.1 presents a block diagram of the Maximum A posteriori (MAP) Receiver. A number, *D*, of cochannel interferers impinge on the array from different angles of arrival. The array is matched filtered and sampled at some integer multiple of the baud rate. The sampled array signals are input to a multi-user detector, which performs a simultaneous MAP estimate of all users. A receiver of this type has been investigated by a number of researchers [205, 14, 20, 59, 13, 35, 203].

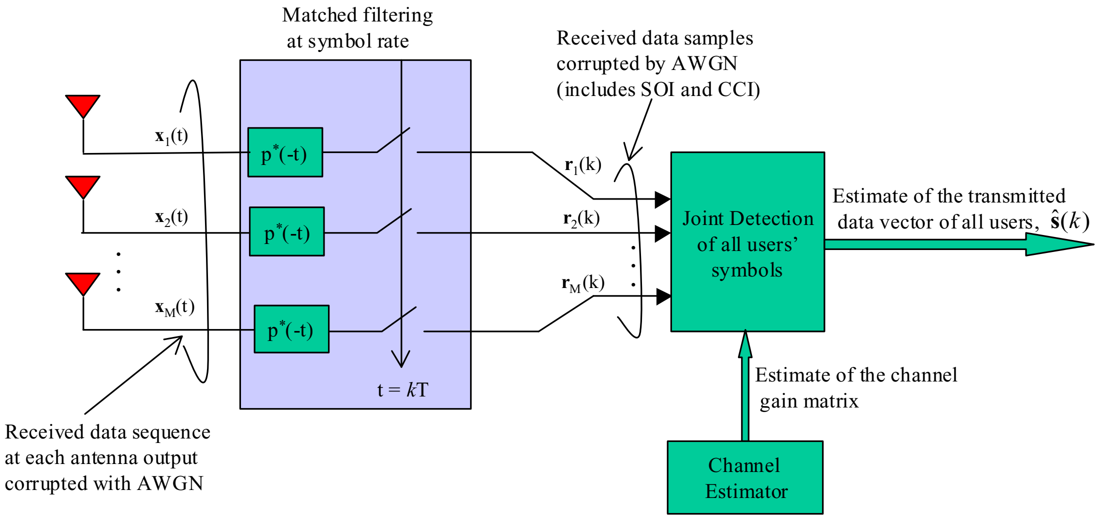

Figure 2.1: Multi-user receiver employing an *M* antenna array. Transmitted symbols of all users are separated and demodulated jointly using Joint-MAP (JMAP) algorithm. Channel estimates are obtained through the use of training sequences or pilot symbols (for CDMA type systems).

For simplicity, consider the case where the signals of all interfering users are synchronized in carrier frequency and baud. Furthermore, assume there is no ISI introduced by the channel. Finally, assume the signal is matched filtered and sampled at the symbol rate. Under these conditions, the received vector, **x**, can be written as

$$\mathbf{x} = \mathbf{H}\mathbf{s} + \mathbf{n} \tag{2.1}$$

where **s** is a vector of transmitted symbols and **H** is the array response, and **n** is a spatially uncorrelated AWGN vector. For example, for QPSK users, each element of **s** is drawn independently and equally likely from the alphabet *{1, -1, j, -j}*. Also, for a calibrated array, the

j^{th} column of **H** is the steering vector for the j^{th} user times its amplitude and phase offset. Although the assumption of synchronous users may seem pathological, it is, in some sense, a worst case scenario. In the synchronous case, there are fewer distinguishing features between co-channel users.

The joint-MAP receiver, attempts to choose the most likely set of transmitted signals, **s**, given the received vector and channel. That is

$$\hat{\mathbf{s}} = \arg\max_{\mathbf{s}}\{f_{\mathbf{s}|\mathbf{H},\mathbf{x}}(\mathbf{s} \mid \mathbf{H},\mathbf{x})\} \tag{2.2}$$

where $f_{\mathbf{s}|\mathbf{A},\mathbf{x}}(\mathbf{s} \mid \mathbf{A},\mathbf{x})$ is the likelihood of the vector **s** conditioned on the received vector and knowledge of the channel. In the general case, a MAP receiver is intractable. However, if signals associated with all users are equally likely and independent, then the MAP receiver is equivalent to the maximum likelihood (ML) receiver [170]:

$$\hat{\mathbf{s}} = \arg\max_{\mathbf{s}}\{f_{\mathbf{x}|\mathbf{H},\mathbf{s}}(\mathbf{x} \mid \mathbf{H},\mathbf{s})\} \tag{2.3}$$

In the case of spatially uncorrelated additive white Gaussian noise, this receiver reduces to the following processor:

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}}\left\|\mathbf{x} - \mathbf{Hs}\right\|^2 \tag{2.4}$$

The channel response, **H**, should be estimated with a training sequence. In this section we assume that the channel is perfectly known; however, in subsequent sections this assumption will be relaxed. If **s** were a continuous valued vector, the optimal receiver would be a least-squares estimator. However, since **s** is drawn from a finite number of possible values, differential calculus minimization techniques do not apply. Equation (2.4) tells us that the ML receiver picks the closest signal in array space to the received signal. This is equivalent to drawing decision boundaries in array space. These decision boundaries are similar to the decision drawn in I/Q space for MPSK. Before we discuss the optimization of Equation (2.4), let us discuss how to visualize what the ML receiver does. We will consider two examples. The first example illustrates joint ML detection for a single antenna. The second example illustrates joint ML detection for two elements.

Figure 2.2 illustrates two synchronous BPSK users of different powers impinging on a single antenna. In this case, the array manifold consists of a scaling and sum of the interfering users' symbols. We can describe the signals transmitted by all the users in two different spaces: user signal space and array space. The left set of axes in Figure 2.2 illustrates all possible combinations of user symbols. These axes should not be confused with I/Q space. The horizontal axis illustrates the possible values, s_1, transmitted by user 1. The vertical axis illustrates the possible values, s_2, transmitted by user 2. There are four possible combinations of transmitted signals. In contrast, we can visualize all possible received signals, x, if there is no noise in our channel. The relation $x = \mathbf{H}\underline{s}$ maps all signals on the left set of axes to the array space on the right set of axes. For instance, if $\mathbf{s} = [1, 1]^T$ is jointly transmitted, the received signal without noise will be *1.5*. If the receiver knows the powers of both signals, it can draw decision boundaries that separate the user's signals. If the symbols in the vector $\mathbf{s} = [-1, 1]^T$ are jointly transmitted, the receiver might observe $x = -.7 - j.7$ because of additive noise. So, the receiver will correctly guess that users transmitted $\mathbf{s} = [-1, 1]^T$ because it knows this value corresponds to $x = -.5$. One can easily see, that if the users are of equal power $P = 1$, $\mathbf{H} = [1\ 1]$ and the transmitted vectors $\mathbf{s} = [-1, 1]^T$ and $\mathbf{s} = [1, -1]^T$ result in the ambiguous point $x = 0$. In this case, the minimum probability of error for any receiver is $P_b = .25$.

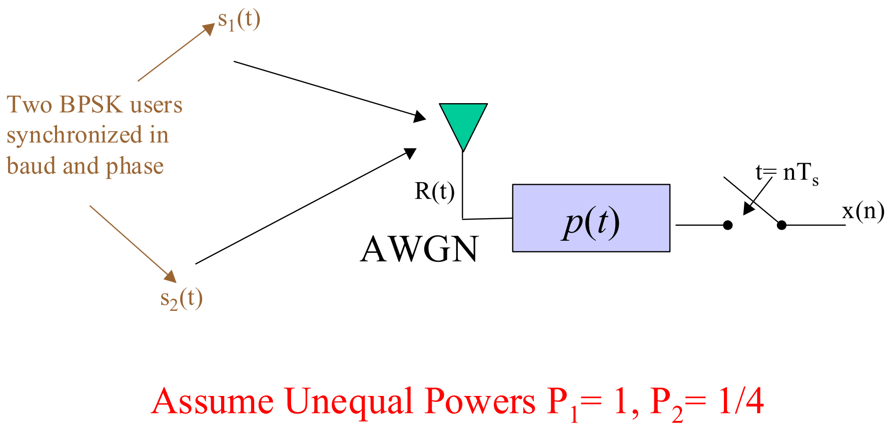
s1(t)
Two BPSK users synchronized in baud and phase
s2(t)
R(t)
AWGN
p(t)
t= nTs
x(n)
Assume Unequal Powers P1= 1, P2= 1/4

Figure 2.2: Two baud and phase synchronous BPSK signals $s_1(t)$ and $s_2(t)$ impinge on a single-element antenna. The received signals are pulse shaped and matched filtered at the baud rate.

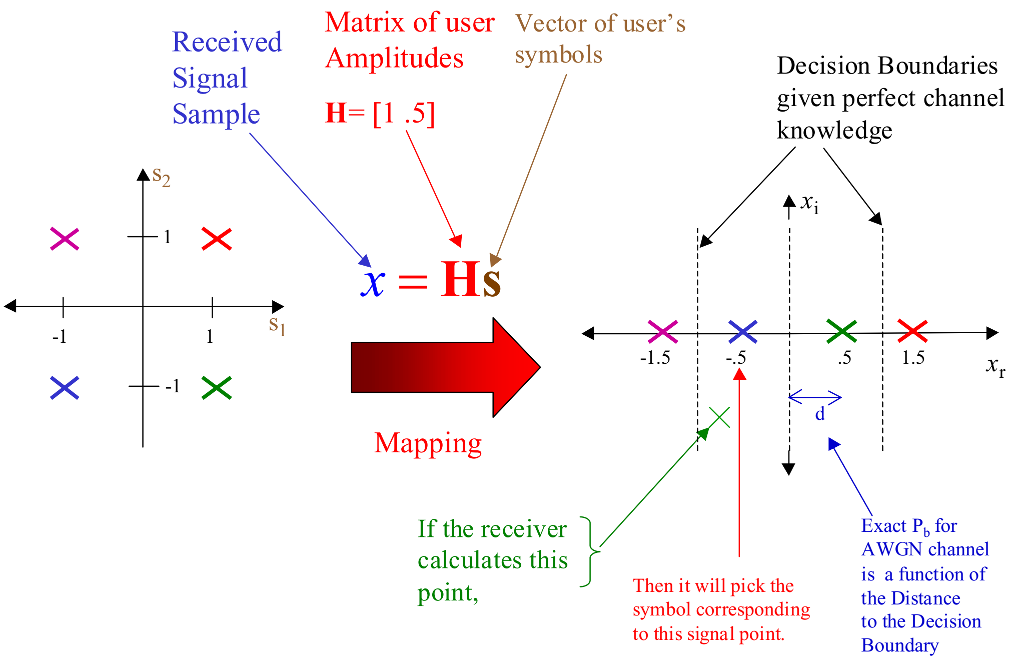

Figure 2.3: The signal space is projected on the array manifold. This mapping results in an ambiguous point, which can not be resolvable by the joint detection

In general, for a larger number of antenna elements, it becomes impossible to visualize the array space because there are too many dimensions. However, if we restrict ourselves to a simple example, we can visualize the array space in three dimensions. Now consider the scenario in Figure 2.3; four synchronous, equal power, BPSK users impinge on a perfectly calibrated, two element antenna array. This is an overloaded environment. For the sake of simplicity, we will say the four users are equally spaced in AOA over a range of 180^0. Because there are more users than dimensions, we cannot visualize the original signal space of **s**. However, because the array is perfectly calibrated, only the real part of the first element is of interest. We can visualize the signal points in array space by plotting the coordinates $x_r^{(1)}, x_r^{(2)}$, and $x_i^{(2)}$. This is illustrated in Figure 2.4. Here we see that all 2^4 possible combinations of user's signals generate distinct points in array space. In this case, decision boundaries are planes separating nearest neighbors. It is difficult to draw these planes without hiding signal points, but it should be easy to imagine these decision boundaries.

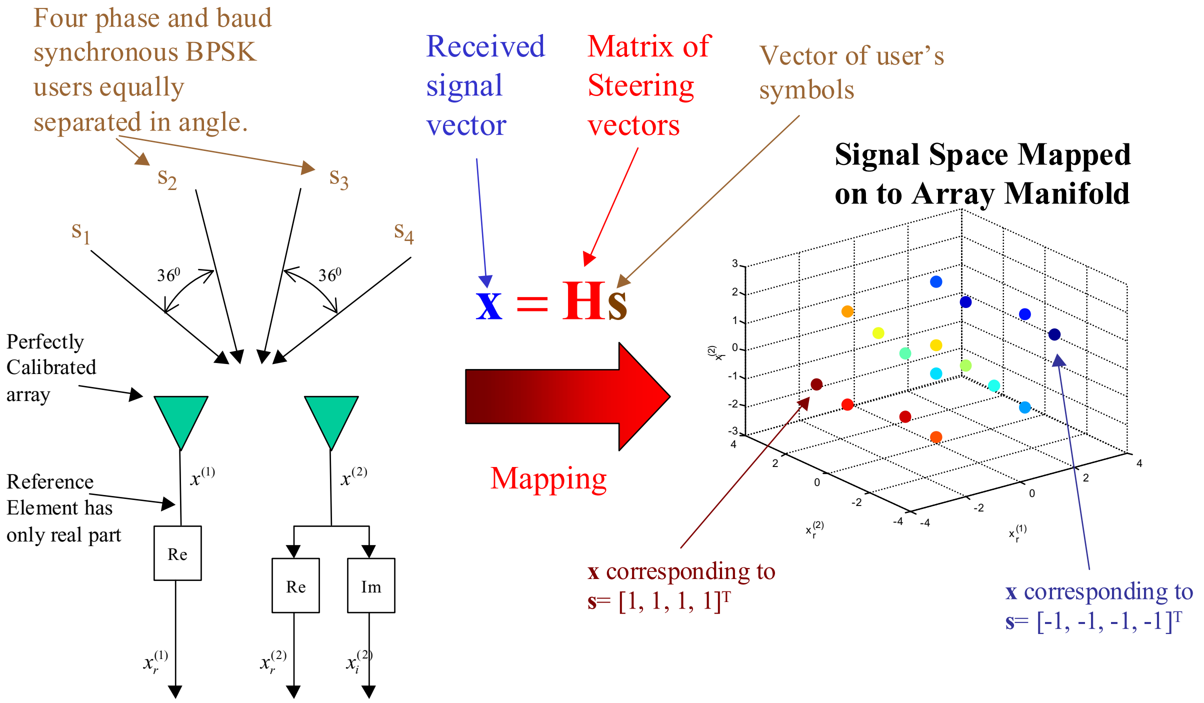
Four phase and baud synchronous BPSK users equally separated in angle.
s1
s2
s3
s4
36⁰
36⁰
Perfectly Calibrated array
Reference Element has only real part
x(1)
x(2)
Re
Re
Im
xr(1)
xr(2)
xi(2)
Received signal vector
Matrix of Steering vectors
Vector of user's symbols
x = Hs
Mapping
Signal Space Mapped on to Array Manifold
x corresponding to s= [1, 1, 1, 1]T
x corresponding to s= [-1, -1, -1, -1]T
xr(1)
xr(2)
xi(2)

Figure 2.4: Four equal power, baud/phase synchronous BPSK signals impinge on a two-element perfectly calibrated antenna array. Each possible transmitted vector of received symbols from all users is mapped to a distinct point in array space. The mapping results in no signal ambiguity.

If noise is sufficiently low, a joint ML receiver can reliably estimate the symbols from all users. The joint ML receiver's performance is limited by the distance between signal points in signal space with respect to the noise power. For many practical applications of interest, the SNR will be too high to jointly demodulate all users. However, this is fundamentally different than saying that signal extraction in overloaded environments is impossible.

In the first case, the decision boundaries were very simple and the ML receiver could be implemented with a simple threshold comparison. However, in general, ML decision boundaries for an array are difficult to describe in a compact form. A brute force method of minimizing equation (2.4) is an exhaustive search through all possible transmitted vectors, **s**. However, in the case of synchronous QPSK signals, this involves $M(D+1)4^D$ computations per symbol making this receiver prohibitively expensive for large D. This problem is closely related to the problem of multi-user detection for CDMA in which case no optimum algorithm has been found to reduce the receiver's complexity [207]. This has motivated many researchers in the CDMA community to find sub-optimal interference canceling receivers. For the asynchronous user case, the symbol of any given user can interfere with two consecutive symbols of any other user. Hence, maximum likelihood sequence detection must be performed with the Viterbi Algorithm.

Despite the JMAP receiver's prohibitive complexity, it yields the best possible performance in AWGN. It acts as a benchmark against which we can compare any other receiver. Inspired by the work of Verdu in CDMA multi-user detection [157, 158, 197], Grant and Cavers [13] have derived a closed form expression for a tight upper bound on error probability for synchronous users in fading channels. The derivation accounts for the possibility of imperfect channel estimates. Their results predict that a two-element array, in moderate SNR environments, can successfully demodulate up to six equal power users, even with an imperfect channel estimate. This prediction outperforms results achieved with linear space-time processing. Although these results were found under the synchronous user assumption, it has been found in [35, 203] that asynchronous users will help improve the JMAP's performance. No upper bound on error probability has been found for the asynchronous user case in the presence of multipath.

Despite the JMAP receiver's prohibitive complexity, its performance motivates research in efficient interference cancellation techniques for antenna arrays. In a sense, linear STAP can be considered an interference rejection technique because an *M*-element beamformer can place *M*-1 independent nulls in the direction of interfering users. However, just as CDMA non-linear interference canceling receiver can outperform a single correlator, the research indicates that interference canceling STAP can outperform linear STAP. The subsequent sections will provide an overview of signal extraction algorithms that are designed for interference-limited environment and an assessment of their performance in an overloaded array environment.

2.2 SINGLE-CHANNEL SIGNAL EXTRACTION ALGORITHMS

As mentioned before, single-channel signal extraction algorithms include *interference rejection* and *joint detection* techniques. In all these algorithms, only temporal processing is utilized since the receiver antenna at the base station contains only one element. Figure 2.5 shows a breakdown of the algorithms.

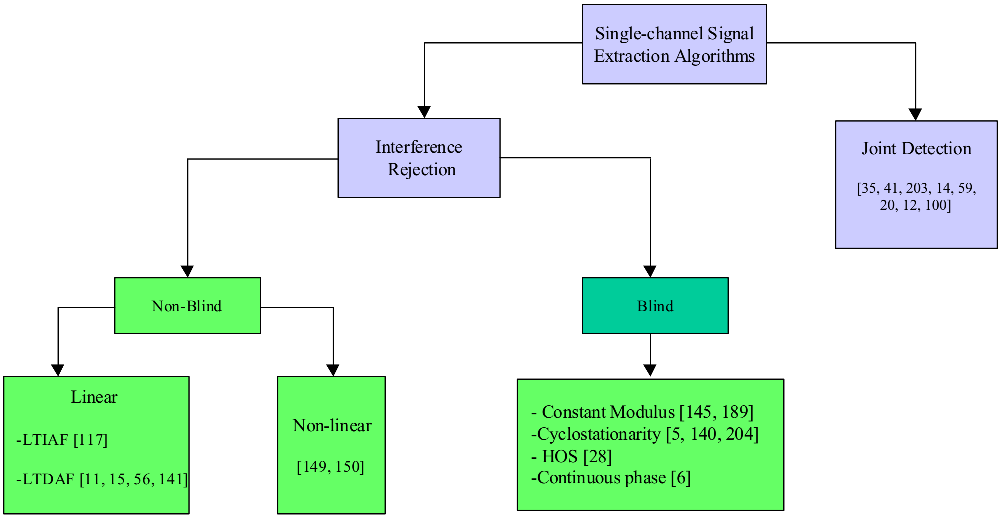

Figure 2.5:A breakdown of the single-channel signal extraction algorithms.

2.2.1 INTERFERENCE REJECTION ALGORITHMS

This section describes a variety of interference rejection techniques that employ only temporal processing. This survey is in no way exhaustive. We limit our discussion to algorithms that can be easily scaled to overloaded array processing. The reader is referred to [37] for a more detailed survey. It is interesting to note that many algorithms can separate two equal power cochannel interferers with only temporal processing. The section is broken into two major approaches: blind and non-blind algorithms. In the former, training sequences are assumed available in order to estimate channel and receiver parameters with adaptive processing. In the latter, training sequences are not available.

2.2.1.1 NON-BLIND TECHNIQUES

There are two aspects to any adaptive processor: the receiver architecture and the adaptation algorithm. Strictly speaking, all adaptive algorithms are non-linear and time varying. However, we call an adaptive filter linear time-independent, if the adaptive algorithm is intended to converge to a linear time invariant (LTI) filter. In contrast, when performing interference rejection, there are many reasons why one would not want to use a Linear Time-Independent Adaptive Filter (LTIAF). Linear time-dependent adaptive filtering (LTDAF) has shown to perform much better when trying to detect modulated signals in noise. Non-blind adaptive processing can be broken into three main approaches: LTIAF, LTDAF, and non-linear adaptive processing.

2.2.1.1.1 Linear time-independent adaptive Filtering (LTIAF)

The most fundamental tool in signal processing is the linear time invariant filter (LTI). Linear time invariant filters may be used to equalize ISI but, to some degree, LTI filters can also perform interference cancellation. With the aid of a training sequence the optimum LTI filter can be estimated with any standard adaptive algorithm. The reader is referred to [174] for a survey of adaptive equalization algorithms. An example of a linear time-independent adaptive filter used for equalization and interference cancellation is given in [117]. The equalizer consists of a tapped-delay line that works in a training mode and decision-directed mode. The output of the equalizer is decomposed into a Wiener Filter (WF) term and a misadjustment filter (MF) term. Interference rejection is done by creating a notch in the frequency response of the WF. MF improves the overall performance by compensating for the ISI produced by WF. Note that a linear time-independent filter can at best cancel only a narrow-band interferer.

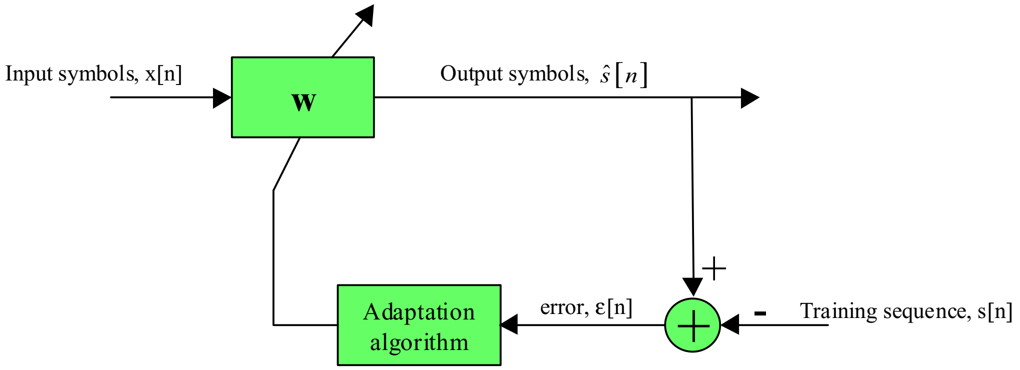

Figure 2.6: Linear Time-Independent Adaptive Filter. The error signal is generated from the estimated output symbols and known training sequence. The adaptive algorithm updates the filter taps in such a way that the mean square error is minimized.

2.2.1.1.2 Linear Time-Dependent Adaptive Filtering (LTDAF)

As previously mentioned, most digital waveforms and many analog waveforms can be classified as cyclostationary or conjugate cyclostationary. Interference rejection can be performed by exploiting the cyclostationary properties of the signal of interest and interfering signals. It is well known that, for stationary signals in additive white Gaussian noise, the optimal filter in the mean square sense is a linear, time-invariant filter. However, the optimal filter for cyclostationary signals is believed to be a polyperiodic filter [141]. This periodically time-varying filter exploits the spectral coherence of the signal of interest by combining discrete frequency-shifted and filtered versions of the received signal. The optimal choice of the discrete frequency shifts has been shown to be the cyclic frequencies of the signal of interest. The optimal choice of the filters (one for each cycle frequency) is called the cyclic Wiener filter. The cyclic Wiener filter is dependent on the properties of the signal of interest and its interferers. There are many implementations of polyperiodic filters, but perhaps, the simplest is the so-called FREquency SHift (FRESH) filter. Although, most man-made signals exhibit an infinite number of cyclic frequencies, near-optimum performance can be obtained by exploiting only a select few. If training sequences are available, the reduced complexity FRESH filter-bank can be solved via a host of adaptive algorithms e.g., RLS or LMS. Such an adaptive implementation is often called a Time-Dependent Adaptive Filter or TDAF. Several implementations of the TDAF are available such as the Time-Series-Representation TDAF (TSR TDAF), or the Frequency Domain TDAF (FD TDAF)[11]. All have identical optimal performance but exhibit different convergence properties [15].

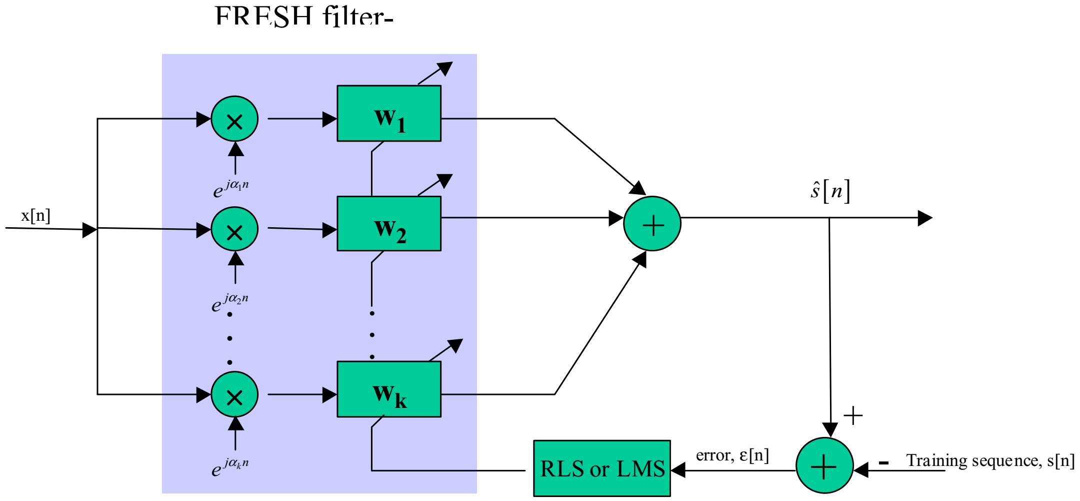

Figure 2.7: Linear Time-Dependent Adaptive Filter.

The performance of TDAFs is analyzed in [56]. They have been found to be particularly useful for rejecting interference that exhibits different cyclostationary properties than the SOI. For example, with an SNR of 30dB, a TDAF was found to be able to separate two equal power, co-channel, square pulse shaped, PAM signals whose baud-rates differed by 5%. The output SNR is 15dB in contrast to the optimal LTI filter, which yielded an SNR of ~3dB. Similar results are provided for SQPSK, and MSK signals. However, these results have been found to be highly dependent on the signal's excess bandwidth. Zero excess-bandwidth (e.g. r= 0 roll-off factor QPSK signals) cannot be effectively estimated directly with a TDAF. Nevertheless, in heavy interference environments, it is often useful to estimate a SNOI and cancel it from the received signal. This technique may be useful, even when the SOI has near zero excess bandwidth.

Another attractive feature of the TDAF is that it performs full waveform restoration of the SOI. Hence, a TDAF may be used as a front-end processor for other signal processing algorithms. In contrast, an MLSE makes hard decisions on output symbols, not on the SOI's waveform[1].

2.2.1.1.3 Non-linear adaptive processing

Despite the rich theoretical background underlying linear processing (e.g. Wiener or Cyclic-Wiener filtering), there are many reasons for implementing a non-linear processor. First, MMSE linear equalizers are not very efficient on channels with deep spectral nulls in the passband. This is because the linear equalizer places high gain near the spectral null in order to compensate for the distortion and thereby enhances the noise present in those frequencies. Nonlinear methods do not suffer from this phenomenon. The two most common forms of non-linear adaptive processors are the *Decision Feedback Equalizer (DFE)* or the *Maximum Likelihood Sequence Estimator (MLSE)*.

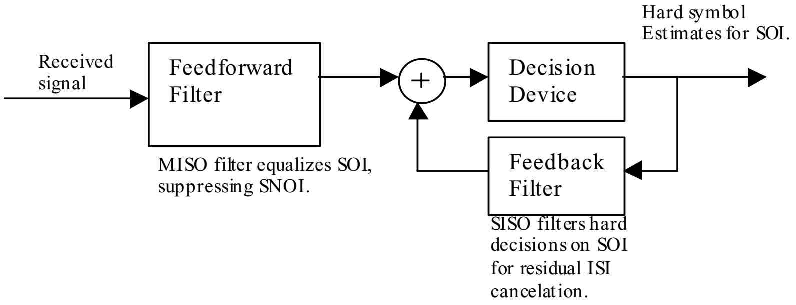
Received signal
Feedforward Filter
MISO filter equalizes SOI, suppressing SNOI.
+
Decision Device
Hard symbol Estimates for SOI.
Feedback Filter
SISO filters hard decisions on SOI for residual ISI cancelation.

Figure 2.8: Application of a DFE to signal extraction.

Decision feedback equalization is a well-known technique that has received much attention in the literature. The reader is referred to [136, 174, 170] for a detailed discussion. Perhaps, what is less well known, is that DFE can perform limited interference rejection. DFE consists of a

[1] The Soft Output Viterbi Algorithm (SOVA) [213] can provide "soft" log-likelihood probabilities. However, these have limited application for down-stream processing [214].

feedforward filter (tapped delay line) and a feedback filter (FBF). The FBF is driven by decisions on the output of the detector, and its coefficients can be adjusted to cancel ISI on the current symbol from past detected symbols (Figure 2.8). The weight update can be done using either the MMSE criterion (e.g. LMS algorithm) or the LS criterion (e.g. RLS algorithm) [136]. Lo *et al* [149,150] proposes an adaptive, fractionally spaced DFE to cancel interference (both co-channel and adjacent channel) and suppress ISI in the presence of a single, dominant co-channel signal and uncorrelated, additive Gaussian noise. The DFE consists of a fractionally-spaced feedforward filter and a symbol-spaced feedback filter that are both implemented as tapped delay lines. The authors show that a directly adapted RLS-DFE performs better than a computed DFE, which employs estimates of the channel impulse response and impairment (CCI + noise) autocorrelation estimates. However, the performance gain of the directly adapted RLS DFE degrades drastically as the noise power increases.

Non-linear processing is a powerful technique for separating interfering users. This discussion briefly reviews the concept of Viterbi Equalization: the application of the Viterbi algorithm to MLSE in multipath channels. We will expand upon this topic in subsequent sections when we discuss the application of MLSE for multi-user detection with antenna arrays. Since the reader is likely to be most familiar with the Viterbi Algorithm for convolutional decoding, we draw a parallel between Viterbi Equalization. Finally, we discuss a technique called Reduced State Sequence Estimation (RSSE) which is a sub-optimal but computationally efficient way to perform Viterbi Equalization.

The Viterbi Algorithm is an efficient MLSE implementation for channels with memory. Channel memory models the fact that in many wireless channels, a given received sample is a function of previously transmitted symbols.

In case of a convolutional code, the channel memory is imposed by the convolutional encoder. The mapping from channel state to code symbol is known apriori by the receiver. In the case of a multipath, channel memory is imposed by inter-symbol interference. This is illustrated in Figure 2.9 (a). In this case, the mapping from channel state to the received symbol is unknown to the receiver, and the channel must be estimated. This is illustrated in Figure 2.9 (b). In both figures, the channel state $\sigma[k]$, is the previous two transmitted symbols: $\sigma[k] = (s[k-1], s[k-2])$. Henceforth, we will discuss only the use of MLSE for equalization.

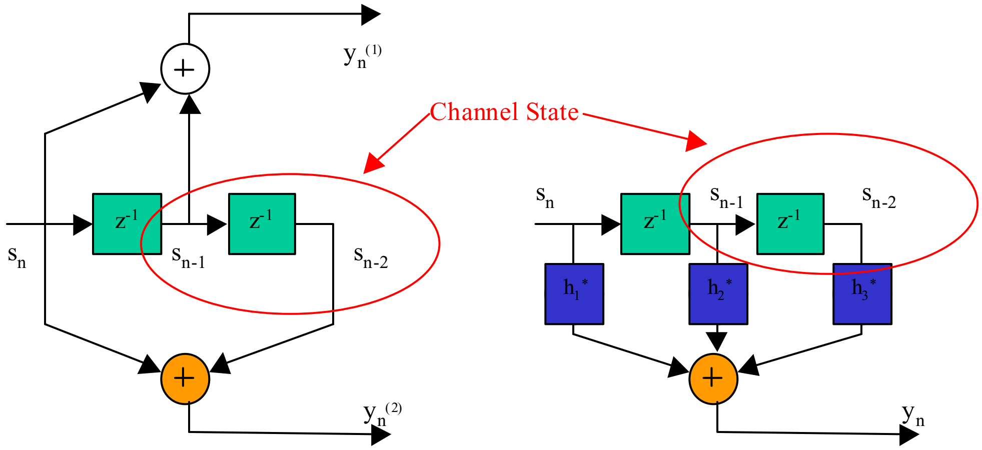

Figure 2.9: Similarity of a convolutional encoder and an FIR channel model. (a) (left) ½ convolutional encoder: binary bits ∈ {0,1} are shifted in a shift register and encoded into two channels. (b) (right) FIR digital equivalent channel model: binary BPSK symbols ∈ {-1,1} are shifted into a shift register. The current output is a linear combination of the current and previous symbols.

A trellis is a way of visualizing all possible state sequences of a certain length. Figure 2.10(a) illustrates the k^{th} stage of the trellis for BPSK symbols transmitted across a channel with two symbols of ISI. This figure illustrates the fact that only certain transitions from one channel state to the next are possible. The entire trellis is constructed by concatenating the 0^{th} stage, with the 1^{st} stage, the 2^{nd}, stage, and so on. The state at the 0^{th} stage is assumed known. It should be obvious that for a channel with length, L_h (e.g. L_h symbols of ISI), and alphabet[2], $\mathcal{A}$, the size of the trellis is $|\mathcal{A}|^{L_h}$. In the case of figure 8, $L_h=3$, and $|\mathcal{A}|=2$.

In chapter 3, we will show that maximum likelihood sequence estimation is equivalent to minimizing a path through the trellis. A brute force method of minimizing a path through a trellis is to enumerate through all possible paths, picking the one with the least cost. However, the Viterbi algorithm reduces the complexity of the search by culling candidate paths at each stage of the trellis. After paths are culled at each stage, the remaining candidates are called survivors.

The Viterbi Algorithm has manageable complexity if $|\mathcal{A}|$, or L_h, is small. However, as we will see when we apply the Viterbi Algorithm to very long channels and/or higher order modulation

[2] For example for QPSK, the alphabet is $\mathcal{A}$={1, -1, j, -j}, and the alphabet size, $|\mathcal{A}|=4$.

schemes, the complexity quickly becomes unmanageable. This has motivated many researchers to seek sub-optimal methods of sequence estimation [85, 24, 64, 43]. The most popular method is Delayed Decision Feedback Sequence Estimation (DDFSE). The reduced-state trellis for BPSK is illustrated in . Here channel memory is not accounted for explicitly in the state trellis. Instead, channel memory is accounted for in the metric by looking back at the survivors in the trellis. In this example, the number of states in the trellis is reduced to $|\mathcal{A}| = 2$. DDFSE is no longer an optimum technique, but has been shown in many cases to approach the performance of a full-state MLSE. Henceforth, we will call Figure 2.10(a) a full-state trellis, and Figure 2.10 (b) as a reduced state trellis.

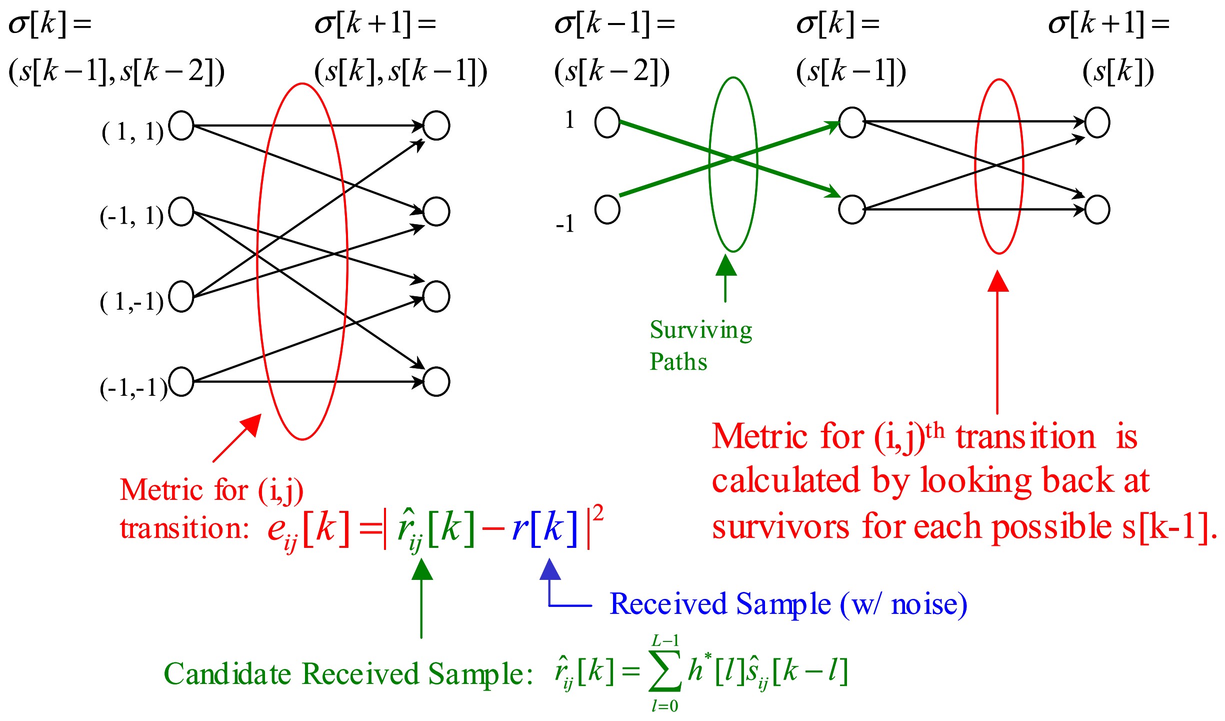
$\sigma[k]=$ $(s[k-1],s[k-2])$
$\sigma[k+1]=$ $(s[k],s[k-1])$
(1, 1)
(-1, 1)
(1,-1)
(-1,-1)
$\sigma[k-1]=$ $(s[k-2])$
$\sigma[k]=$ $(s[k-1])$
$\sigma[k+1]=$ $(s[k])$
1
-1
Surviving Paths
Metric for (i,j)th transition is calculated by looking back at survivors for each possible s[k-1].
Metric for (i,j) transition: $e_{ij}[k]=|\hat{r}_{ij}[k]-r[k]|^2$
Received Sample (w/ noise)
Candidate Received Sample: $\hat{r}_{ij}[k]=\sum_{l=0}^{L-1}h^*[l]\hat{s}_{ij}[k-l]$

Figure 2.10: (a) (left) kth stage of full-state MLSE trellis. Number of states is $|\mathcal{A}|^{L_h-1}$, where $|\mathcal{A}|$ is the alphabet size and L_h is the channel length. (b) kth stage of the reduced state trellis with survivors from the (k-1)th stage. In this case the number of states is reduced to $|\mathcal{A}|$.

In a previous section, we showed that the optimum receiver for an antenna array in AWGN can be implemented as an exhaustive search for the maximum likelihood solution. Assuming no ISI, the optimum receiver could perform symbol-by-symbol detection. In the case of ISI, the best choice of the current symbol must account for previous symbols. This necessitates maximum likelihood sequence estimation. The application of the MLSE for equalization has received great attention in the literature and is usually implemented with the Viterbi Algorithm [173].

A distinction should be made between MLSE and Joint MLSE (JMLSE). MLSE usually makes an implicit assumption that the interference is well modeled as Gaussian. This is true if there are a large number of interferers with a power much less than the SOI. In contrast, JMLSE jointly estimates all interfering user's signals. JMLSE is discussed in more detail in section 2.2.2.

2.2.1.2 BLIND TECHNIQUES

Blind algorithms must extract a signal by exploiting some other property of the modulated waveform. These properties may include but are not limited to the constant modulus property, the finite alphabet property, or cyclostationary properties.

Blind signal extraction gives rise to a major issue: channel identifiability. Is it possible to identify a channel with only a finite amount of data collection? In general, a channel cannot be estimated exactly without a training sequence, but often it can be estimated up to a multiplicative constant (including a possible phase shift) [106, 206]. In this case the channel is said to be identifiable. This is sufficient for a communication system that employs differential signaling. This issue has been tackled by many researchers. See [106] for a complete survey. In most wireless systems of practical interest, a channel can be blindly identified.

In environments with multiple signals of the same type, many blind algorithms suffer from the signal capture problem. For instance, suppose a receiver attempts to extract a signal by restoring its modulus. Further, suppose that there are several CM signals in the environment. Then, there is no guarantee that the algorithm will extract the SOI. *Signal Capture* occurs when a blind algorithm extracts the wrong signal. This is illustrated in Figure 2.11.

Constant modulus algorithms perform blind signal extraction by exploiting the constant modulation property of the SOI. They generally use steepest-descent approach or least squares based approach to minimize the a CM cost function. The simplest CM cost function is the following:

$$J = E\left[\left|y(n)\right|^2 - 1\right] \tag{2.5}$$

where $y(n) = \mathbf{w}^H \mathbf{x}$ is the estimate of the desired signal at the output, **x** is the received data vector at the antenna array, and **w** is the beamformer weight vector.

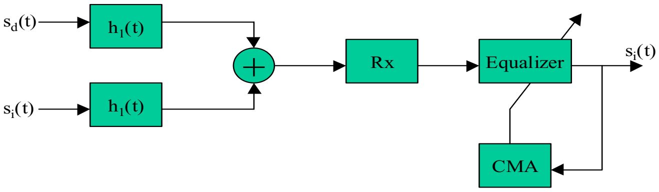

Figure 2.11: The signal capture problem: CMA captures the interfering signal, $s_i(t)$ instead of the desired signal, $s_d(t)$.

When coupled with differential signaling techniques, adaptive algorithms that exploit this signal property have been observed to effectively equalize many practical multipath channels. The simplicity of constant modulus algorithms and their near optimal performance have made them a very popular blind family of algorithms. However, it has been observed to suffer from the signal capture problem. In [189] Rude and Griffiths developed Linearly Constrained Constant Modulus Algorithm (LCCM). By itself, the LCCM is limited in performance by its fundamental linear architecture: the best a linear processor can hope to do is to null out its interferer while also rejecting a considerable amount of desired signal energy. In [145], the authors apply LCCM as a blind front end to a conventional decision feedback equalizer.

The original formulation for TDAF requires training sequences. However, blind TDAF algorithms have also been presented in the literature. Two blind TDAF algorithms were developed in [204]: a Time dependent CMA and a Spectral Correlation Discriminator. The former updates the TDAF filters with the standard constant modulus algorithm [174], the latter computes weight updates with an error signal derived from a spectral input/output cross-correlation coefficient. In an example, both were able to discriminate between two equal-power QPSK and BPSK signals, closely spaced in carrier frequency. For each, the BERs did not exceed 10^{-3}. The receiver achieved its performance by exploiting differences in their data rates and carrier frequencies. A blind adaptive FRESH filter and its convergence properties are described in [140]. A similar implementation [5] discussed an application for AMPS cellular signals which exploits the cyclostationarity induced in an FM modulated voice signal by a supervisory audio tone (SAT). A two stage TDAF was able to separate two co-channel equal power interferers with an output voice SNR of 30 dB.

FM signals have continuous phase property that a sequence estimator can exploit. This novel technique is analyzed by Hamkins [6]. The author developed a blind technique to separate co-channel FM signals by exploiting the *temporal correlation* of the modulated speech signal. The technique quantizes the slope of the message signal for all CCIs. The Viterbi Algorithm then attempts to find the maximum likelihood estimate of a sequence of slopes. The resulting size of the trellis is 2^{3n} where n is the number of co-channels signals. The algorithm has been found to successfully demodulate two equal power, cochannel interfering signals.

All of the above mentioned techniques exploit second order statistics, i.e. autocorrelation, cross-correlation and variance, of both the signals and noise. Blind Higher Order Statistics (HOS) techniques have also been developed in the literature [28] an have demonstrated superior steady state performance in channels with ISI. However, their slow convergence properties have prevented their use in wireless communication systems with dynamic channels.

2.2.2 JOINT DETECTION AND ESTIMATION ALGORITHMS

The interference rejection techniques mentioned in section 2.2.1 attempt to estimate interfering signals and then strip them from the total signal, leaving only the desired signal components plus noise. On the other hand, joint detection algorithms recover all the signals, desired and interfering, from the signal environment and then discard the latter. These algorithms are based on the *Maximum Likelihood (ML)* and *Maximum a Posteriori (MAP)* criteria for the joint recovery of the cochannel signals. These criteria are used to derive two important sequence estimation and symbol-by-symbol detection techniques, *Maximum Likelihood Sequence Estimation (MLSE)* and *Maximum a Posteriori Symbol Detection (MAPSD)* [171], respectively.

As previously discussed, the best possible receiver for an overloaded environment with AWGN is a Joint MAP receiver. We considered a simple example of multiple synchronous users impinging on an antenna array with no ISI present. The optimum receiver performs an exhaustive search through all possible combinations of symbols transmitted by all users. In the case when users are asynchronous or when ISI is present, the optimum choice of any given symbol is dependent on previous symbols. Therefore, sequence estimation must be implemented with the Viterbi Algorithm. The channel length, L_h, increases the channel memory so the number of states/stage is $|\mathcal{A}|^{D(L_h-1)}$, where $|\mathcal{A}|$ is the alphabet size, D is the number of interfering users. For two 16-QAM users and a channel length of 3, the number of states becomes ~ $65\text{x}10^3$! Such a sequence estimator would be several orders of magnitude greater than the largest commercially available Viterbi decoder. This has motivated many researchers to investigate *Reduced State Sequence Estimation (RSSE)* and Delayed Decision Feedback Sequence Estimation (DDFSE) [172]. In its simplest form, DDFSE reduces the number of states required by not accounting for the additional states generated by ISI. Instead DDFSE accounts for ISI in the error metrics of the Viterbi algorithm. This is a sub-optimal receiver, but can perform nearly as well as the full-state JMAPSD [43]. Other proposed methods attempt to reduce the number of states by estimating interferers of different powers in a multi-stage implementation.

The most computationally complex but straightforward implementation of joint detection is the so-called Interference Canceling Equalizer (ICE) of [35]. This name is misleading because the receiver is essentially a multi-user detector. The equalizer accounts for the fact that users may experience different dispersive channels before arriving at the receiver, which is consistent with a realistic mobile radio environment. The ICE uses a reduced state ML sequence estimator by accounting for ISI in the error metric of the Viterbi Algorithm. The channel is estimated from a set of training sequences with the RLS algorithm, and tracked with a decision feedback equalizer. The number of states in the reduced set implementation is $|\mathcal{A}|^D$. For a single antenna, narrowband receiver with two co-channel QPSK users, this processor has realistic complexity. For two equal power cochannel users, the system performance is limited by the ambiguous case when the two users completely destructively cancel each other. In this case the BER can be at best, 0.25. However, for realistic environments, the phases of interfering users seldom coincide, and adequate performance is achieved. Similar algorithms are presented in [41, 203].

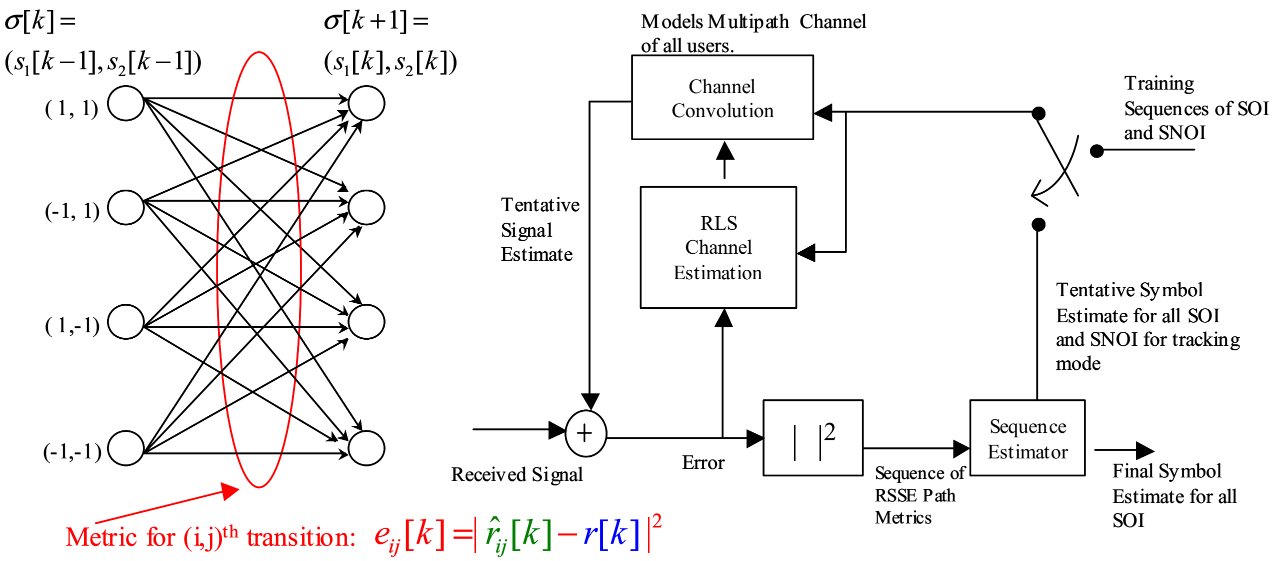
$\sigma[k] =$
$(s_1[k-1], s_2[k-1])$
$\sigma[k+1] =$
$(s_1[k], s_2[k])$
(1, 1)
(-1, 1)
(1,-1)
(-1,-1)
Metric for (i,j)th transition: $e_{ij}[k] = |\hat{r}_{ij}[k] - r[k]|^2$
Models Multipath Channel of all users.
Channel Convolution
RLS Channel Estimation
Tentative Signal Estimate
Training Sequences of SOI and SNOI
Tentative Symbol Estimate for all SOI and SNOI for tracking mode
+
Received Signal
Error
$|\ |^2$
Sequence of RSSE Path Metrics
Sequence Estimator
Final Symbol Estimate for all SOI

Figure 2.12: (a) (left) k^{th} stage of joint detection trellis (b) (right) Block diagram of ICE.

To mitigate the computational complexity of trellised based joint detection, Giridhar *et. al* [14, 59] extended the classical MLSE and MAPSD to *Joint MLSE (JMLSE)* and *Joint MAPSD (JMAPSD)* algorithms. The authors use each of these algorithm to jointly cancel cochannel interference with a suboptimal two-stage MAP detector. Both blind [20] and non-blind algorithms were investigated [14,59]. Simulations show that for low SNRs and known channels, the iterative two-stage MAP yields performance close to the MAP detector. The blind version employs an LMS stochastic gradient update for channel estimates.

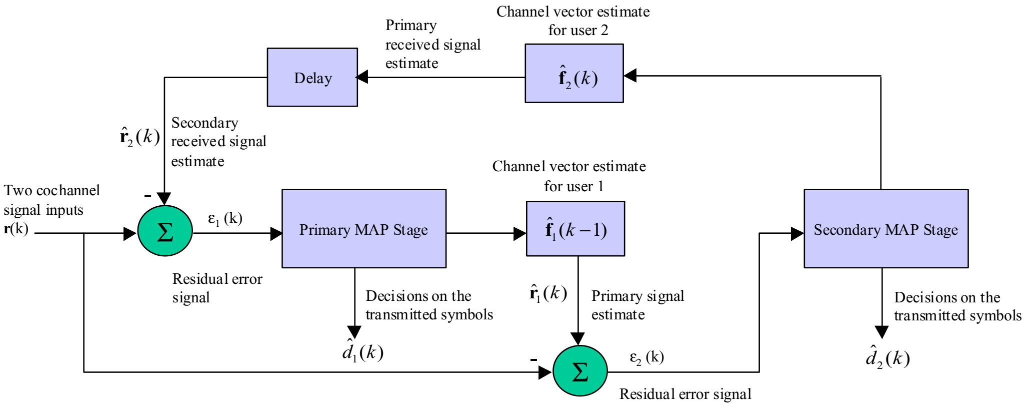
Channel vector estimate for user 2
Primary received signal estimate
Delay
$\hat{\mathbf{f}}_2(k)$
$\hat{\mathbf{r}}_2(k)$
Secondary received signal estimate
Channel vector estimate for user 1
Two cochannel signal inputs $\mathbf{r}$(k)
-
Σ
ε_1 (k)
Primary MAP Stage
$\hat{\mathbf{f}}_1(k-1)$
Secondary MAP Stage
Residual error signal
Decisions on the transmitted symbols
$\hat{\mathbf{r}}_1(k)$
Primary signal estimate
Decisions on the transmitted symbols
$\hat{d}_1(k)$
-
Σ
ε_2 (k)
$\hat{d}_2(k)$
Residual error signal

Figure 2.13: Two-Stage JMAPSD Algorithm.

Although trellised based joint detection techniques were originally developed for linear modulation, its use has been extended to non-linear modulation types such as GMSK, as well. In [12] Ranta et. Al develops a joint detection technique for symbol synchronous GSM. The algorithm employs a joint channel estimator with JMLSE in a per-survivor fashion (see [7] for an overview of per-survivor processing). The algorithm exploits the fact that in 120^0 sectored cellular system, there is usually only one dominant co-channel interferer. In this case, only two signals need to be jointly detected. The algorithm has been shown to outperform conventional interference rejection techniques for SINRs > 20dB. Other examples of JMLSE's application to non-linear modulation types are given in [100].

In the previous section, we cited several examples in the literature of signal extraction techniques that can successfully separate co-channel narrow band signals using only a single antenna. Most techniques exploit some separation in bandwidth or power. It seems tempting to apply many of these techniques to the overloaded array processing problem. Surprisingly, the application of many of these techniques to antenna arrays, LTDAF for example, have not been found in the literature. The simplest and most popular interference rejection techniques for antenna arrays are the beam-former, and the space-time equalizer. However, non-linear techniques have been applied as well, although not to the overloaded array problem. Although the focus of this section is on antenna arrays, we can benefit from the literature in many other multi-channel signal processing applications such as sonar, and multi-channel wire-line (cross-talk mitigation).

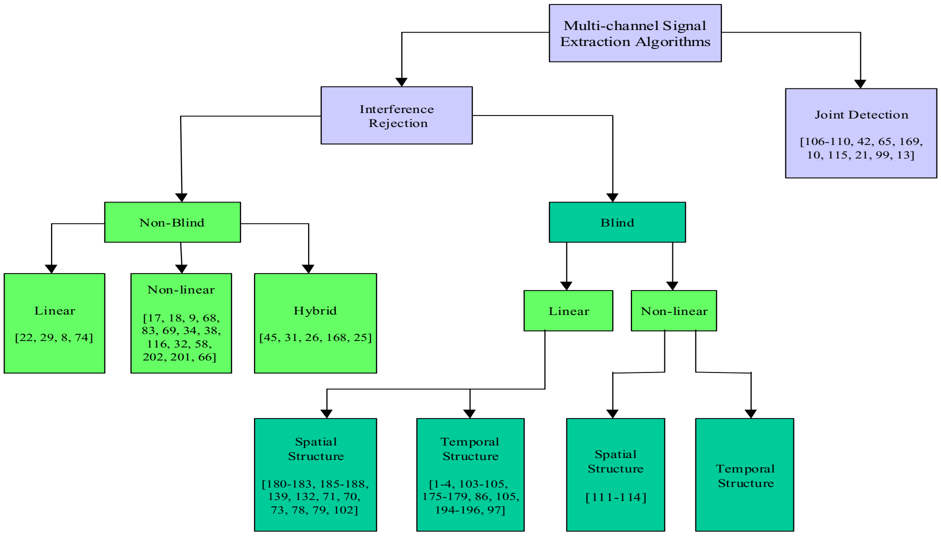

Figure 2.14: Chart showing the breakdown of the multi-channel signal extraction algorithms.

2.2.3 Interference Rejection Techniques

2.2.3.1 Non-blind Techniques

Non-blind algorithms require the use of training sequences to estimate the channel. Although the use of training sequences greatly simplifies the channel estimation problem, exploiting them can be difficult when interfering users transmit asynchronously. Non-blind interference rejection techniques are broken into linear, non-linear, and hybrid categories. Although linear interference rejection is well known to break down in overloaded environments, the development of these techniques are very mature. In the literature, several complete solutions for contemporary cellular systems have been reported that account for difficult problems such as synchronization. Although, non-linear interference rejection algorithms have not been applied to space division multiple access systems. However, they do provide a means for reducing interference from adjacent cells.

Linear Techniques

Linear signal extraction techniques in the literature can be classified as space-only processing techniques and spatio-temporal techniques where the antenna weights are optimized using MMSE criterion. The goal is to maximize the output signal-to-interference-plus-noise ratio (SINR). Although linear STAP is often used to demodulate multiple users, they are not really joint demodulation techniques because the fundamental receiver architecture can be separated into several single user beamformer/equalizers. This is not the case for any true multi-user detector[3].

Winters proposed a spatial processing algorithm for signal acquisition in IS-54 digital radio systems [22]. This algorithm uses an adaptive array to do classical beamforming, which null out the interferers and maximize the array output SINR. Two different weight adaptation techniques, LMS and Direct Matrix Inversion (DMI) are compared. DMI algorithm has a much faster convergence in the expense of higher computational complexity. In addition, DMI algorithm performs better in signal acquisition and interference suppression. The authors of [29] study the suppression of ACI, CCI and ISI by using linear zero-forcing equalizers/combiners, i.e., using antenna arrays with T_s-spaced tapped delay lines where T_s is the symbol period. The linear equalizer puts T_s-spaced zero crossings in the time-domain to reject ISI at the sampling instances, and the linear beamformer steers nulls in the direction of the interfering signals. The MMSE criterion is used for the adaptation of the linear equalizers/combiners.

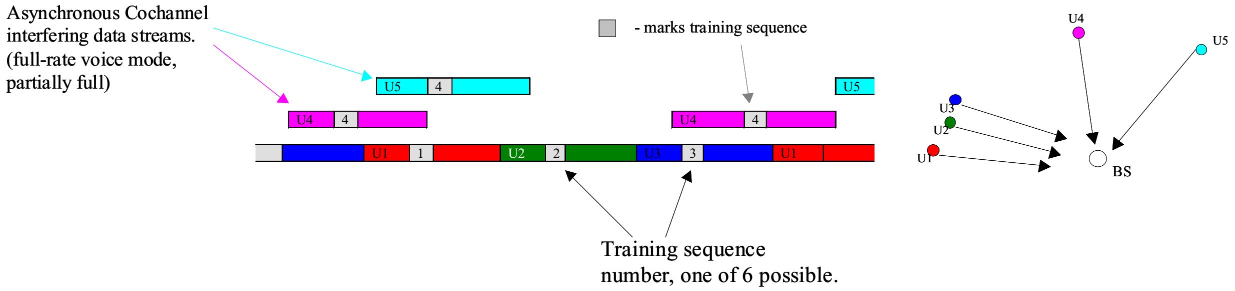

Figure 2.15: U1, U2, U3 are the synchronous users, U4 and U5 are the co-channel interferers. The optimal beamformer during U1's training sequence (training sequence 1) is no longer valid after U4's frame ends.

Separation of cochannel IS-54/136 signals using beamforming and linear equalization is considered in [8, 74]. Unlike many other algorithms in literature, these papers put emphasis on *asynchronous* TDMA frames. The authors show that in a bursty TDMA format, interference may not overlap the training sequence of the current slot. If this occurs, the optimal beamformer solution is no longer valid for that particular interferer (Figure 2.15). This is due to the misalignment of the TDMA frames. The authors propose a frame synchronization procedure followed by a *sequential* separation algorithm where the sources are captured and removed

[3] Our terminology here differs from [207].

sequentially via several stages of partial beamforming and signal cancellation. Frame synchronization is achieved by locating the peaks in the cross correlation of the beamformer outputs with modified training sequences. The beamformer weights are updated using LS criterion and ISI equalization is carried out using a fractionally spaced linear equalizer. The algorithm can effectively separate several users as long as the number of cochannel users does not exceed the number of antenna elements since a linear beamformer (*M* element array) can null out upto *M-1* users.

2.2.3.1.1 Nonlinear Techniques

Many signal extraction algorithms concatenate linear array processing with nonlinear temporal processing such as DFE and MLSE. In general, nonlinear adaptive array processing techniques perform better than the aforementioned linear techniques, especially in severe multipath fading environments.

In [9], Lee *et. al* provide a complete solution to the aforementioned asynchronous frame problem. Their decision-feedback based solution is similar to the approaches in [8,74]. The number of separated users is limited by the array size due to the linear beamforming operation.

MLSE based adaptive antenna array processing algorithms attempt to estimate the channel response for each signal as well as the covariance matrix of the impairment. They then use these estimates in its branch metric to search for the most likely desired transmitted sequence. In section 2.2.1.1.3, we made the distinction between the use of MLSE for interference rejection and JMLSE for joint detection. Several authors have investigated the use of MLSE for interference rejection [83, 68, 69, 32, 34, 38, 116, 202, 174, 16]. They differ in both their assumed operating environment and the calculation of the path metric for the Viterbi algorithm. In all MLSE approaches, interference is treated as additive Gaussian noise. If the MLSE algorithm works correctly, it will reject interference and demodulate the SOI. The myriad of approaches in the literature differ by how they account for the following effects:

1. Temporal correlation of interference induced by pulse shaping and multi-path.
2. Spatial correlation effects induced by the fact that interference is impinging on the array from certain directions of arrival.

3. Treatment of time-varying channel tracking.

MLSE techniques can be broken into two major approaches: metric combining (MC), and interference rejection combining (IRC). MC assumes that the interference is spatially uncorrelated (i.e. the interference comes from a ubiquitous direction). In such an environment, the branch metric for the Viterbi Algorithm is just the sum of the branch metric for each antenna. IRC makes no spatial assumption about the interference environment. Hence, the autocorrelation of the impairment must be estimated along with the SOI's channel.

Metric combining is treated in [83], [68], and [69]. Metric combining performs well when different antennas experience different fading processes but there is no interference present. Interference Rejection Combining, described in [34], [38], and [116], was developed for the current US Digital standard employing π/4-DQPSK. In [38, 39], only the temporal correlation of the interference over a symbol interval is accounted for. In [116] temporal correlation of the interference beyond a symbol interval is accounted for. In both, practical considerations including symbol, phase, and frequency synchronization are accounted for. An example for the European digital standard, GSM, is provided [58]. In [202] the authors compare several equivalent architectures for IRC. The performance of the multi-channel MLSE reception techniques necessitates for very accurate channel estimation. In [32] Bottomley and Molnar develop a low-complexity approach to cancel interference *prior* to channel estimation. This *pre-cancellation approach* is obtained through a series of approximations of Kalman filtering approach [174]. However, the performance of this algorithm is limited to very low Doppler spreads, e.g. less than 20-Hz Doppler spreads.

Any MLSE technique is susceptible to errors in channel state information. In [201], the authors show that imperfect CSI can create a floor in symbol error probability. Finally, in [66] Sheen and Stüber propose and analyze joint MLSE equalization and decoding of trellis-coded modulation employing a diversity array.

In [39, 116], neither MC nor IRC can reject more users than elements. This further supports the hypothesis that joint detection separates the SOI from interference better than interference rejection.

2.2.3.1.2 Hybrid

We have discussed the problems incurred by considering interference as Gaussian noise in an MLSE processor. The challenge is to account for the temporal correlation properties of interfering signals. To avoid this difficulty but still exploit MLSE's strong equalization abilities, many authors have proposed a hybrid approach: a linear space-time processor cascaded with a non-linear MLSE processor. This is illustrated in Figure 2.16.

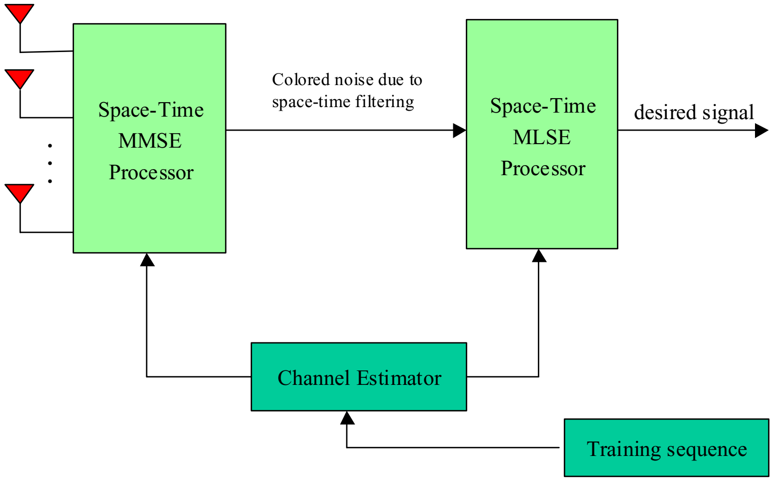

Figure 2.16: Linear ST-MMSE beamformer is concatenated with a nonlinear ST-MLSE processor. The linear beamformer attempts to cancel the interference whereas the following ST-MLSE processor gets rid of the ISI.

A theoretical analysis of these types of space time processors is presented in [26]. Complete solutions for GSM and EDGE systems are given in [26]. Complete solutions for GSM and EDGE systems are given in [45, 25, and 24].

2.2.3.2 Blind Techniques

Blind algorithms must extract a signal by exploiting properties that are specific to the SOI. These properties may include constant modulus properties, the finite alphabet properties, or cyclostationary properties.

In section 2.2.1.2, we have already discussed the issues of identifiability and signal capture. In Space Division Multiple Access (SDMA) systems, another issue arises. When a blind processor is trying to extract signals solely on their signal properties, there is no way to distinguish between signals of the same type. Hence, there is no guarantee that the j^{th} user will appear on the output of j^{th} output port of the signal processor. This is called the *port shuffle problem* [1, 106].

For digital systems, the port shuffle problem can often be solved by searching for some user id in the demodulated data. No easy solution exists for analog modulated systems.

The majority of blind techniques in the literature are based on linear space-time filtering. Although the filter update algorithm is often highly non-linear. The filtering operation that estimates the signal, is itself, a linear operation. Regardless of the non-linear nature of the adaptive algorithm, if the end convergence is a linear time invariant space-time filter, the blind algorithm cannot perform any better than the optimum linear time invariant solution. Again, it is well known that a linear STAP breaks down in overloaded environments regardless of the adaptive algorithm used. Nevertheless, we include a description of a variety of blind techniques to illustrate how blind signal extraction is possible.

2.2.3.2.1 CM Property

The term Constant modulus algorithms (CMA) describes a class of adaptive algirthms which blindly extract signals from interference by exploiting their constant modulus peroperties. The application of CMA to adaptive beam-forming has been widely studied by many researchers. Many different algorithms have been developed, each varying in its performance and complexity. The simplest version of CMA attempts to find a linear beamformer which minimizes the following cost function with a stochatic gradient decent:

Where $y[n]$ is the signal of interest estimated by a linear beamformer. This type of CMA algorithmhas been extensively analyzed in [104, 175, 176].

If other CM interferers in the environment are not accounted for, CMA based algorithms can suffer from an inherent signal capture problem. This problem has been addressed by many users. The lest computationally complex algorithm is a multi-user gradient decent algorithm proposed in [4]. The algorithm attempts to jointly estimate a bank of beamformers to extract all constant modulus signals in the environment. The classic CMA cost function above is modified to include cross correlation terms which de-correlate the outputs of all beamformers. The cost function was deomonstrated to contain a single local minimum.

As an alternative to the gradient decent approach, Agee et al develops the least squares CMA (LS-CMA) in [103]. This technique employs alternating projections in a block-fashion. An optimal beamfoermer is obtained in the two steps. First, the desired signal is projected onto a signal set with the desired CM signal property. Secondly, a new beamformer is derived which minimizes the squared modulus error. The algorithm has been well received in practice as well as in the literature. The block updates are more computatinoally intensive than a gradient decent algorithm but is numerically stable and usually exhibits faster convergence.

In [152] the author addresses the issue of signal capture with the LS-CMA approach. The proposed algorithm called Multi-Target LS-CMA, extracts all constant modulus signals in the environment by jointly estimating a bank of beamformers. The beamformers are forced to extract different signals with a Modified Gram-Schmidt Orthogonalization (MGSO) step. This receiver structure is illustrated in Figure 2.17.

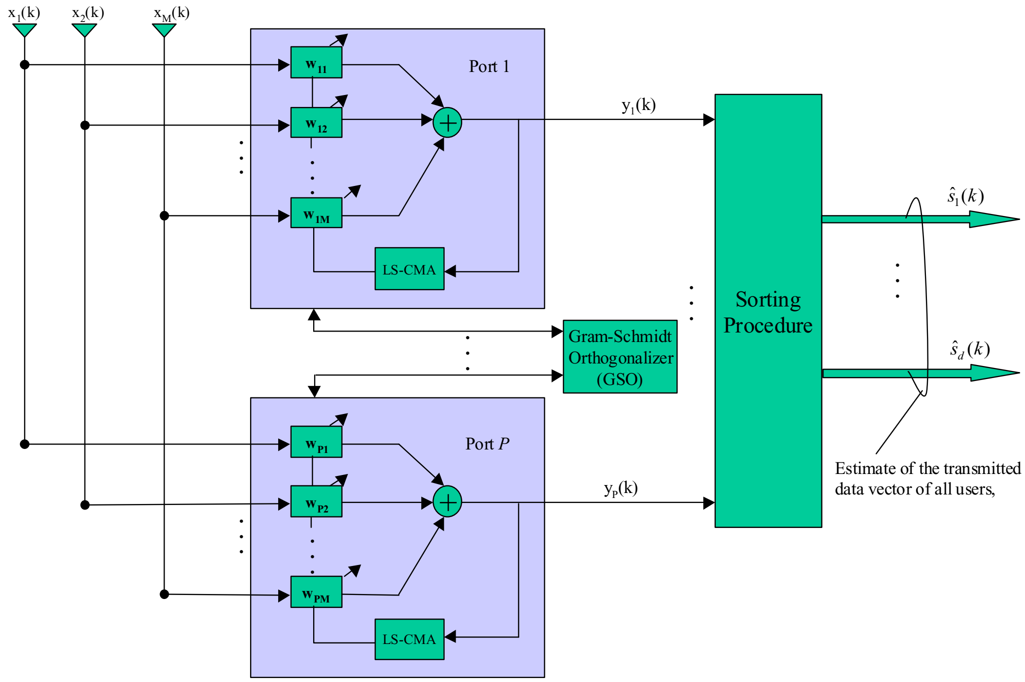

Figure 2.17: Multi-Target LS-CMA adaptive array. *M* is the number of antenna elements and it is usually equal to the number of ports, i.e. *M=P. P* different beamformer weights are adapted independently by LS-CMA technique. GSO orthogonalizes the weight vectors so that each port corresponds to a unique weight vector. Sorting procedure relates the port outputs to each user's signal. If number of users, *D*, is larger than the number of elements (or ports), then one output port may contain the signals of several users.

Another blind beamforming algorithm, *Analytic CMA (ACMA)*, is proposed in [2]. ACMA uses a subspace approach to solve the signal capture effect and slow convergence of traditional CMA.

ACMA involves the simultaneous diagonalization of matrices to solve the constant modulus factorization problem, i.e. factorizing **X=AS** given that **A** and **S** are full rank and the transmitted signals have constant modulus. Estimation of the number of sources is built in the algorithm. ACMA attempts to simultaneously demodulate all CM signals impinging on the antenna array. However, it is very computationally complex, i.e. the most efficient form has complexity given by $O(9D^4n+36M^2n)$, where D is the number of signals of interests, n is the data collect length, and M is the number of elements. The performance for ACMA on FM signals has not been analyzed in the literature. Furthermore, ACMA algorithm is derived for high-SNR conditions and the estimates of the ACMA beamformer are asymptotically biased. Improvements on this algorithm in the presence of low SNRs are presented in *Weighted ACMA (W-ACMA)* by Van der Veen [3].

2.2.3.2.2 FA Property

Another exploitable signal attribute for blind processing is the so called finite alphabet property. This property is thought to be much more powerful than the constant modulus property, but is more difficult to exploit. Exploiting the FA property requires casting the blind estimation problem in a tractable framework. In [106], the author investigates two algorithms, Iterative Least Squares with Projection (ILSP) and Iterative Least Squares with Enumeration (ILSE). ILSP and ILSE differ in that ILSP is an adaptive beamforming algorithm. Specifically, ILSP blindly estimates the signal and channel by iteratively performing a least squares estimate of channel and signal in two separate steps. The finite alphabet property is exploited at each step by making a hard decision on the least squares signal estimate. On the other hand ILSE is a true joint detection algorithm. Section 2.2.4 discusses ISLE in some detail.

This last contribution has motivated many researchers to build upon the ILSP framework. More computationally efficient but sub-optimal implementations of ILSP are given in [108-110]. Algorithms that account for ISI are presented in [42] and [65]. In both, ISI and cochannel interference is modeled with a set of linear equations. Blind deconvolution is viewed as a matrix factorization problem. Implicit in both algorithms is the assumption of a linear beamformer. Van der Veen and Paulraj [169] combine the CM property with the FA property of the signals and propose a blind channel identification algorithm using *Real ACMA* (special case of the ACMA technique applied to real signals) to initialize the ILSP algorithm. The authors use this blind channel algorithm to carry out space-time linear beamforming for the linear approximation of GMSK signals.

2.2.3.2.3 Cyclostationarity Property

Previously, we discussed how the cyclostationary of digitally modulated signals can be exploited to perform interference rejection with a TDAF. In this case, TDAF was updated with a training sequence. However, cyclostationarity is also an exploitable property for blind processing. All cyclostationary signals have second order statistics that are degraded in the presence of noise and interference. A class of blind beamforming algorithms called Spectral Self-Coherence Restoral (SCORE) algorithms [97] update a beamformer by attempting to restore the spectral correlation at a known cycle frequency. SCORE algorithms are powerful because they applicable to any cyclostationary signal (not just CM or FA). Also, SCORE makes no assumption about the array manifold or the interference environment. There are several different SCORE algorithms that operate with slightly different cost functions and receiver architectures, but the basic operating principle of each is similar. In particular, the performance of an algorithm called cross-SCORE has been shown to converge to that of a non-blind optimal SINR beamformer if the number of interfering signals with the same spectral correlation frequencies (including echoes) in the environment does not exceed the number of elements. The convergence of the algorithm is highly sensitive to the spectral correlation coefficient and data collection time. When the spectral correlation coefficient at a chosen cyclic frequency is near one, cross-SCORE has convergence performance near that of a non-blind least squares algorithm. However, in most practical cases, the spectral correlation coefficient will be less. The data collect time should be chosen to be large enough to discriminate between interfering signals with closely spaced spectral correlation frequencies but small enough to assure signal coherence.

2.2.4 Joint Detection Techniques

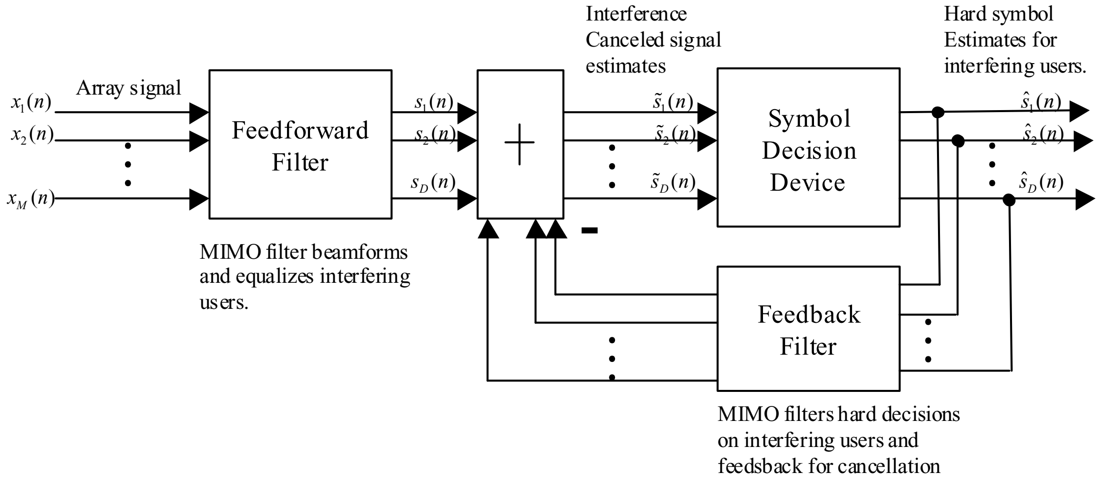
Array signal
$x_1(n)$
$x_2(n)$
$x_M(n)$
Feedforward Filter
MIMO filter beamforms and equalizes interfering users.
$s_1(n)$
$s_2(n)$
$s_D(n)$
+
-
Interference Canceled signal estimates
$\tilde{s}_1(n)$
$\tilde{s}_2(n)$
$\tilde{s}_D(n)$
Symbol Decision Device
Hard symbol Estimates for interfering users.
$\hat{s}_1(n)$
$\hat{s}_2(n)$
$\hat{s}_D(n)$
Feedback Filter
MIMO filters hard decisions on interfering users and feedsback for cancellation

Figure 2.18: Multi-user detector employing a multiple input multiple output (MIMO) Decision Feedback Equalizer (DFE). The MIMO feedforward filter acts as a beamformer/ equalizer. Symbol decision device is a hard limiter whose output (hard limited symbol estimates for the interference) is fed back using a MIMO feedback filter.

In [21, 98, 99], a comparison of interference rejection and joint detection is analyzed for a multi-element decision-feedback equalizer/beamformer. Here, an interference rejecting DFE feeds back a decision for the SOI only. A joint detection DFE feeds back the decision signals. The number of near equal power interferers is assumed known apriori. In simulation trials of a four element array, joint detection was found to outperform interference suppression by an order of magnitude.

The most straightforward implementation of MLSE applied to antenna arrays is the multi-element extension of ICE (see section 2.2.2). The multi-element version of ICE is identical to that of the single channel version, except more channels must be estimated. A single element ICE receiver must estimate the *d* FIR impulse responses from each transmitter to the receiver. A multi-element receiver must estimate the *M•d* FIR impulse response from each transmitter to each antenna element. Because there are a greater number of channel parameters, longer training sequences may be required. But from Grant's work [13], this requirement should be relaxed by the fact that more elements require less channel accuracy. One interesting attribute of ICE is that if each user is synchronous and experiences no ISI, the receiver reduces to an exhaustive maximum likelihood search outlined in the introduction. Also, the authors found that for equal-power synchronous users, the ambiguity problem present in the single-element receiver is not present in the two-element receiver. Results for more users than elements were not reported in the paper. Finally, the performance of the algorithm was found to be sensitive to Doppler shift: as the channel changes more quickly, it becomes more difficult to track. The authors did not comment on the receiver's tracking performance for minimum-phase or non-minimum phase channels. A detailed understanding of channel tracking performance and overloaded performance are open areas of research.

In [106], a blind joint detection technique was proposed that can exploit the finite-alphabet property. This property is thought to be much more powerful than the constant modulus property, but is more difficult to exploit. Exploiting the FA property requires casting the blind estimation problem in a tractable framework. The author investigates the comparative performance of ILSP and ILSE.

The operating principles of the algorithms are derived by answering two questions:

1. If the channel is known, what are the optimum signal estimates of all users?

2. If the signal estimates are known, what is the optimum channel estimate?

These two questions are fundamentally different from each other because the answer to question 1 involves an optimization over a finite and discrete number of possible transmitted signals. In contrast, for most practical channels, question number 2 involves an optimization over a continuous, complex set of variables. Both ILSE and ILSP attempt to blindly estimate the channel by iteratively answering the questions one and two. Hence, both ILSE and ILSP can be described with the following algorithm:

1. Estimate the best signal estimates of all the users.

2. Estimate the optimum channel estimate.

3. Repeat 1) until convergence is achieved

ILSE and ILSP differ in the approach to step 1. ILSE finds the true ML estimate of all user's signals by *enumerating* over all possible transmitted signals. This step is similar to the approach described in the introduction and is very costly. In contrast, ILSP approximates the ML estimate by first performing a least squares (projection) and then quantizing the least squares solution to a finite alphabet. The least-squares projection of ILSP implicitly performs a beamforming operation because each estimated symbol is made from a linear combination of received signal samples.

The resulting algorithms suffer from several local minima. However, they converge very quickly and hence, a global minima can usually be found by trying different initialization points. Firstly, ILSE and ILSP illustrate the difference between the ML solution and the beamforming solution. In a sequel paper [107], ILSE was found to out-perform ILSP. Also, when ILSE converged to a global maxima, it was found to perform near the ML receiver if the channel was known.

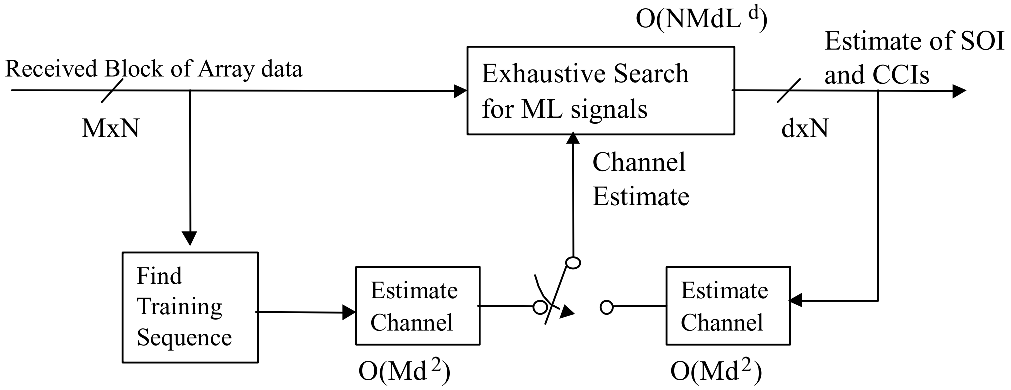

Figure 2.19: *L* ~ Alphabet size, *N* ~ frame size, *M* ~ array size, *D* ~ number of interferers. ILSE iteratively finds ML estimate of channel and data by brute force optimization over all the users. Overall complexity is prohibitively high, i.e. $O(ML^D)$.

ILSE and ILSP are not practical for most systems, because it requires all users to be synchronized in baud and frequency. Also, the algorithms do not account for ISI or a time-varying channel. However, both algorithms are extremely valuable for the following reasons. First, they act as a first step in answering many questions of uniqueness and identifiability for bind channel identification. Second, the fact that ILSE nearly performs as well as a non-blind algorithm clearly illustrates the power of non-linear array processing. Third, they provide a framework in which the FA property is easily exploitable.

Talwar's ILSE is warranted in multi-user environments where there is no spatially correlated background interference. However, in many environments, the interference is spatially correlated but temporally unstructured. Two examples include a large number of low power interferers transmitting from a specific direction, and a large power interferer whose bandwidth is much greater than the SOIs. A receiver for this type of environment is discussed in [10]. Here a blind maximum likelihood cost function is derived that accounts for the possibility of temporally unstructured but spatially correlated noise in addition to multiple SOIs. Two cost functions are presented: one that finds the maximum likelihood signal estimate, and a sub-optimal receiver based on alternating directions optimization. The sub-optimal receiver was found to exhibit a performance near the ML receiver's. Like Talwar's ILSE, users are assumed to be baud-synchronous. The algorithm was found to successfully demodulate up to ten cochannel CPFSK signals with a four element circular array.

2.3 Conclusion

Array processing in overloaded environments requires different considerations than underloaded environments. Overloaded array processing provides two benefits. First, multi-user systems will benefit from an increase in capacity. Second, overloaded array processing can make receivers more robust to interference external to the communication system. The former is of great interest to commercial communication systems. The later benefit is of great interest to military applications. Although, many well-established STAP algorithms break down in overloaded environments, one would still expect to be able to extract more signals than elements because single element interference canceling receivers have been known for a long time. We have described many examples of interference canceling receivers for single element receivers. The literature suggests that time varying and non-linear receivers will perform better than non-linear solutions. In particular, the joint MAP receiver is guaranteed to separate cochannel signals with a lower probability of symbol error than any other receiver.

The fundamental limit on array capacity is given by the probability of error for the JMAP receiver. This receiver will have a symbol error probability better than any other receiver. Upper bounds on the probability of error for the JMAP receiver suggest that the capacity of an M-element array is much greater than M. However, achieving large capacity with the JMAP receiver is difficult because its complexity increases exponentially with the number of users. This motivates a search for sub-optimal receivers that might approach the JMAP's performance. In this survey, we have described a host of array processing techniques. It is well known that linear STAP breaks down in overloaded environments. However, even non-linear interference suppression tends to break down in homogeneous overloaded environments. The success of time-varying interference suppression techniques, such as FRESH filtering for single element receivers, suggests that this may also be a successful approach for overloaded antenna arrays. However, for overloaded homogenous environments , joint detection schemes are expected to yield the best success. Signal extraction algorithms that exploit cyclostationarity (e.g. FRESH) have been found to yield small SINR gains for tight excess bandwidth signals. ML joint detection becomes impractical for large number of elements. For arrays of four or more elements, demodulating a large number of users is the only motivation for joint detection. No practical implementation of an overloaded array processor for a large number of users has been found in the literature. However, just as CDMA research has produced many sub-optimal multi-user detectors that can effectively mitigate multiple-access interference, we expect that sub-optimal multi-user detectors for antenna arrays can support a much greater capacity than conventional STAP. A simple example of a sub-optimal multi-user joint detection technique is the multi-user DFE. Practical multi-user detection is the focus of the rest of this thesis.

Chapter 3: VITERBI EQUALIZATION

The Viterbi Algorithm is an efficient method of finding the least cost path through a trellis. In its original application [173] it was found to be a maximum-likelihood decoder for convolutional codes over binary symmetric channels. Shortly later, it was found to yield the maximum likelihood estimate of a symbol sequence transmitted over a channel with finite memory [54]. Since then, many variants of Forney's formulation have been developed for the purpose of equalization. Many of these applications are sub-optimal, asymptotic approximations to a true maximum-likelihood receiver. In this chapter we will first define the optimum Viterbi Equalizer for known channels, the Maximum Likelihood Sequence Estimator (MLSE). The purpose of this section is to provide a basis upon which other less well-known equalization techniques can be built. Then we will define a sub-optimal but computationally efficient variant of MLSE called Delayed Decision Feedback Sequence Estimation (DDFSE). Finally, we discuss two trellis-based algorithms for equalizing circularly-convolutional channels. All of these approaches fit under the general aegis of Viterbi Equalization (VEQ).

3.1 MAXIMUM LIKELIHOOD SEQUENCE ESTIMATION

In this section we formally define the application of Viterbi Equalization to MLSE. We more formally define the notion of a trellis and the relation of a minimum cost path to a maximum likelihood estimate. Most importantly, we introduce a visualization tool that will help us quickly build equalization (and later joint detection) trellises. This same visualization tool will help us build efficient joint detection trellises for overloaded array processing. Admittedly, there are numerous tutorials on the Viterbi Algorithm in the literature. The purpose of this section is not to provide a comprehensive treatment of the Viterbi Algorithm, but to formally define a notation and intuition upon which other more heuristic trellis-based detection techniques can be based. The simplest form of Viterbi Equalization is Maximum Likelihood Sequence Estimation (MLSE). MLSE yields the maximum likelihood estimate of a symbol sequence in (AWGN). We'll first introduce the concept of FIR channel model. Then we will show how a maximum likelihood estimate of a sequence can be obtained. Finally, we will show how the Viterbi Algorithm yields this estimate. The type of channel under consideration will be the FIR channel model illustrated in Figure 3.1. We will consider linear modulation only. However, approximate MLSE receivers have been applied to non-linear modulation types. Here *s[n]* represents of sequence of symbols drawn from an Alphabet, $\mathcal{A}$. If the signals, *s*[*n*], is a BPSK signal, $\mathcal{A}$ = {1, -1}. If *s*[*n*] is a QPSK signal, $\mathcal{A}$ = {1, -1, j, -j}. Assume that the channel can be modeled as an FIR channel with impulse response *h*[*n*], of length L_h. The received match-filtered and sampled signal can by modeled as:

$$r[n] = y[n] + z[n] \tag{3.1}$$

where

$$y[n] = \sum_{l=0}^{L_h-1} h^*[l]s[n-l] \tag{3.2}$$

and *z*[*n*] is additive complex Gaussian noise with zero mean and variance $\sigma_z^2 = E\left[|z[n]|^2\right]$. The channel convolution can be written in matrix form by defining the following vectors. Let $\underline{h}$ be a vector of "conjugate and flipped" channel coefficients:

$$\underset{1\times L_h}{\underline{h}} = \left[h^*[L_h-1] \quad h^*[L_h-2] \quad \cdots \quad h^*[0]\right] \tag{3.3}$$

Then a vector of received samples can be written as $\underline{r} = \underline{y} + \underline{z}$, where $\underline{z}$ is a vector of the noise process, *z[n]*, and the signal component vector, $\underline{y}$, is expressed in terms of a length *N* frame of symbols as:

$$\underbrace{\begin{bmatrix} y[0] \\ y[1] \\ \vdots \\ y[N-L_h] \end{bmatrix}}_{\substack{\underline{y} \\ (N-L_h+1)\times 1}} = \underbrace{\begin{bmatrix} \boxed{\qquad \underline{h} \qquad} & & \mathbf{0} \\ 0 \; \boxed{\qquad \underline{h} \qquad} & & \\ \mathbf{0} & \ddots & \boxed{\qquad \underline{h} \qquad} \end{bmatrix}}_{\substack{\mathbf{H} \\ (N-L_h+1)\times N}} \underbrace{\begin{bmatrix} s[0] \\ s[1] \\ \vdots \\ \vdots \\ s[N-1] \end{bmatrix}}_{\substack{\underline{s} \\ N\times 1}} \tag{3.4}$$

Note that the length of the channel output vector, $L_r = L_y = N - L_h + 1$, is not the same as the input. Before we charge forth into the maximum likelihood sequence estimator, we will first introduce a visualization tool that will be useful later for constructing more complicated trellises: we can plot the magnitude of the elements of **H** in a checkerboard plot. Let us illustrate this with the following example.

3.1.1 EXAMPLE

Consider the length L_h= 3 channel illustrated Figure 3.1(a). A length N=10 frame of symbols is transmitted. A checkerboard plot of the **H**-matrix is illustrated in Figure 3.1(b). Note that the symbols at the beginning and end of the frame have less energy in $\underline{y}$ than the other symbols; this phenomenon will result in less reliable estimates for these symbols. Beginning and ending a frame with L_h-1 known header and tail symbols can mitigate this effect.

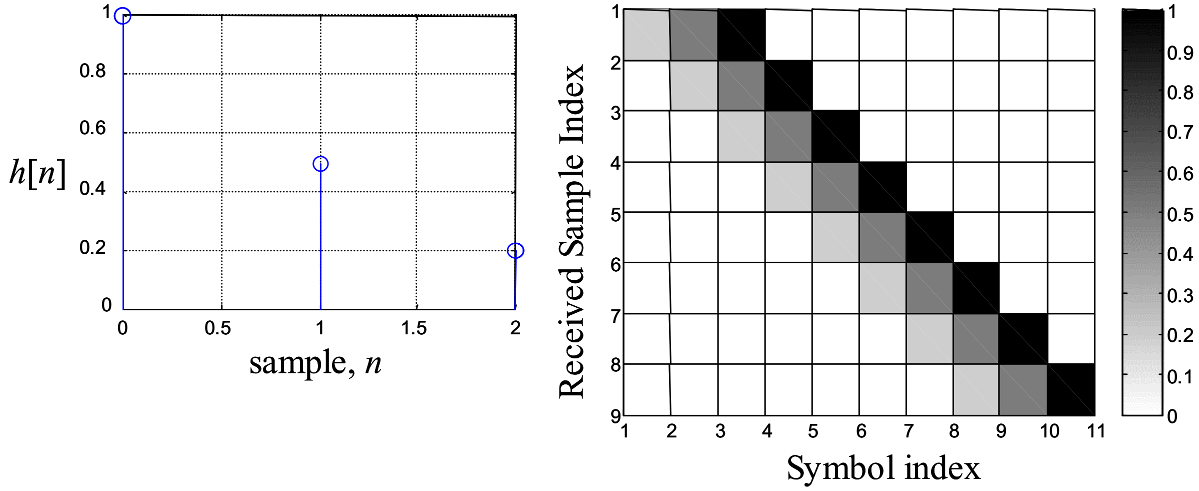

Figure 3.1: (a) (left) FIR channel impulse response. (b) (right) Checkerboard plot of Toeplitz channel matrix.

3.1.2 Development of MLSE

We will now define and develop a MLSE based equalizer. The maximum likelihood estimate of the symbol sequence, $\underline{s}$, maximizes the likelihood of $\underline{r}$ given $\underline{s}$. Let $f(\underline{r}\,|\,\underline{s},\underline{h})$ be the probability density function (pdf) of $\underline{r}$ given knowledge of $\underline{s}$ and $\underline{h}$. Then the maximum-likelihood estimate of $\underline{s}$, $\hat{\underline{s}}$, is given as

$$\hat{\underline{s}} = \underset{\underline{s}\in\mathcal{A}^N}{\arg\max}\, f(\underline{r}\,|\,\underline{s},\underline{h}) \tag{3.5}$$

where $\mathcal{A}^N$ denotes all possible length N sequences of symbols drawn from the alphabet $\mathcal{A}$. If all possible $\underline{s}\in\mathcal{A}^N$ are equally likely, the maximum likelihood estimate is equivalent to finding the most probable symbol sequence for a particular received signal. That is:

$$\hat{\underline{s}} = \underset{\underline{s}\in\mathcal{A}^N}{\arg\max}\, p(\underline{s}\,|\,\underline{r},\underline{h}) \tag{3.6}$$

which is a more intuitive criterion.

The maximum likelihood estimate requires perfect channel knowledge. Although in practical applications, the channel impulse response is not usually known, we will initially assume that it is, and address this issue later. The conditional pdf, $f(\underline{r}\,|\,\underline{s},\underline{h})$, is multi-dimensional Gaussian with mean $\underline{y}=\mathbf{H}\underline{s}$ and covariance $\mathbf{\Phi}_{\underline{zz}} = E\left[\underline{z}\underline{z}^H\right]$.

$$f(\underline{r} \mid \underline{s}, \underline{h}) = \frac{1}{(\pi\sigma_z^2)^{L_r/2}} \exp\left\{-\| \underline{r} - \mathbf{H}\underline{s} \|^2 / \sigma_z^2\right\} \qquad (3.7)$$

Maximization of equation (3.7) is equivalent to the following estimators

$$\begin{aligned} \hat{\underline{s}} &= \underset{\underline{s}\in \mathrm{A}^N}{\arg\max}\, f(\underline{r} \mid \underline{s}, \underline{h}) \\ &= \underset{\underline{s}\in \mathrm{A}^N}{\arg\max} \left\{ \ln f(\underline{r} \mid \underline{s}, \underline{h}) \right\} \\ &= \underset{\underline{s}\in \mathrm{A}^N}{\arg\max}\, \| \underline{r} - \mathbf{H}\underline{s} \|^2 \end{aligned} \qquad (3.8)$$

If we knew nothing bout the desired signal, that is, if *s[n]* could be any value in the complex plane, then the maximum likelihood estimate would be equivalent to a least- squares solution such as the Penrose pseudo-inverse. However, if we exploit the finite alphabet property of $s[n]$, then we can achieve a much more accurate estimate [106]. The brute force maximum likelihood estimate is to exhaustively search through all possible values of $\underline{s} \in \mathcal{A}^N$. This would require us to search over $|\mathcal{A}|^N$ possible combinations of transmitted symbols. For typical wireless voice frame lengths, N~160 symbols, $|\mathcal{A}| = 4$, in which case $|\mathcal{A}|^N \approx 2 \cdot 10^{96}$ iterations is well beyond the computational ability of any computer. However, this search can be performed much more efficiently with the Viterbi algorithm that exploits the sparse nature of the matrix, **H**.

Instead of finding the symbol sequence that yields the least squared error, the Viterbi Algorithm works with the sequence of states, $\sigma[n] = (s[n-1], s[n-2], \ldots, s[n+L_h-1])$ illustrated in Figure 3.2. The VA finds the sequence of states that yields the least squared error. Obviously, if we know the least-cost state sequence, we can pull out the least cost symbol sequence from the state definition. The two are equivalent.

The operating principle of the Viterbi algorithm works on the principle of a trellis. A trellis is a way of visualizing all possible sequence of states. An example trellis for a BPSK signal with a length *N* frame of information symbols transmitted over a length $L_h = 3$ channel is illustrated in Figure 3.2. Following the observation of example 3.1.1, a header and tail of $L_h - 1 = 2$ symbols is assumed. Each circle in the diagram represents a particular sample of $\sigma[n]$. A black line from the left-most circle to the right-most circle depicts a unique state sequence. States are connected if their values do not conflict. For a particular time index, *n*, a state, $\sigma[n]$ can be named by its value in symbols (e.g. $\sigma[n]=(-1,1)$) or its index (e.g. $\sigma[n]=2$). The particular interpretation should be evident in context. Figure 3.2 illustrates an example path.

Before we continue to describe the Viterbi Algorithm, let us define some fundamental constructs. Define a partial path through the trellis, $\rho_0^k=\{\sigma[0],\sigma[1],\ldots,\sigma[k]\}$. Two example partial paths are illustrated in Figure 3.3. We say that a particular state value, *i*, at time index n, $\sigma_0[n]$, is in path ρ_0^k, $\sigma_0[n]\in\rho_0^k, n\le k$, if the n^{th} component of ρ_0^k, $\sigma[n]=\sigma_0[n]$. Furthermore, we say that a particular symbol value at time *n*, $s_0[n]$ is in a path, ρ_0^k, $s_0[n]\in\rho_0^k, n\le k-1$, if there is a state $\sigma_0[n+1]\in\rho_0^k$ and $\sigma[n+1]=(s_0[n],\ldots)$.

We will now relate the partial path concept to the least squares criterion. The least squares criterion can be broken up into additive terms with the following identity

$$\|\underline{r}-\mathbf{H}\underline{s}\|^2=\sum_n |r[n]-\hat{r}[n]|^2 \tag{3.9}$$

where

$$\hat{r}[n]=\sum_l h^*[l]\hat{s}[n-l] \tag{3.10}$$

Each stage of the Viterbi Algorithm will deal with exactly one of the squared error terms in the sum of (3.9). To describe the operating principle of the Viterbi Algorithm, let us define the following quantities.

Let $\hat{r}_{ij}[k]$ be the candidate signal component of the received signal corresponding to a $\sigma[k]=i\rightarrow\sigma[k+1]=j$ transition.

$$\hat{r}_{ij}[k]=\sum_{l=0}^{L_h-1} h^*[l]s[k-l] \tag{3.11}$$

where

$$\sigma[k] = (\hat{s}[k-1], \hat{s}[k-2], \ldots, \hat{s}[k-L_h+1]) = i$$
$$\sigma[k+1] = (\hat{s}[k], \hat{s}[k-1], \ldots, \hat{s}[k-L_h+2]) = j$$

Define the $(i, j)^{th}$ transition cost on the k^{th} stage, $e^{(i,j)}[k]$, to be the error between the received signal and the $(i, j)^{th}$ candidate signal component.

$$e^{(i,j)}[k] = | r[k] - \hat{r}_{ij}[k] |^2 \tag{3.12}$$

Further, define the cost of a partial path, ρ_0^k

$$\varepsilon[\rho_0^k] = \sum_{l \leq k} e^{(\sigma[l], \sigma[l+1])}[l]. \tag{3.13}$$

The Viterbi Algorithm reduces the complexity of search by culling candidate paths at each stage. It is built upon the following obvious axiom: if two paths converge on the same node, then the difference in their cost can be computed from their partial costs at that node. This principle is illustrated in Figure 3.3. Two candidate paths converge on the same state at stage *k* and continue to share the same route thereafter. At this stage, if the cumulative cost of path two's partial path is greater than the cumulative cost of path one's, then there is no way that path two can catch up. Then, at the k^{th} stage the Viterbi Algorithm will cull path two and declare path one as the survivor.

We will now define the Viterbi Algorithm recursively. Define the accumulative cost metric at state i, $\xi^{(i)}[k]$, as follows. If the sequence is preceded with a known header (e.g. $(s[-1], \ldots, s[-L_h+1])$ is known).

$$\xi^{(i)}[0] = \left\{ \begin{matrix} 0 & \sigma[0] = (s[-1], \ldots, s[-L_h+1]) \\ \infty & else \end{matrix} \right] \tag{3.14}$$

otherwise, $\xi^{(i)}[0] = 0$, $\forall\ i = 0, 1, \ldots, |\mathcal{A}|^{L_h - 1} - 1$. Now define $\xi^{(i)}[k]$, for *k*>0,

$$\xi^{(i)}[k+1] = \min_{i \in \mathcal{T}_j} \{ \xi^{(i)}[k] + e^{(i,j)}[k] \}. \tag{3.15}$$

where $\mathcal{T}_j$ is the set of all allowed transitions into the j^{th} state. Define the surviving transition into the j^{th} state at the k^{th} stage as:

$$i_s^{(j)}[k] = \arg\min_{i \in \mathcal{T}_j} \{\xi^{(i)}[k] + e^{(i,j)}[k]\} \tag{3.16}$$

Finally, recursively define the surviving partial path at stage k,

$$\begin{aligned} &\rho_0^0[i] = \{\sigma[0] = i\} \\ &\rho_0^{k+1}[j] = \{\rho_0^k[i_s^{(j)}[k]], \sigma[k+1] = j\} \end{aligned} \tag{3.17}$$

From the preceding definitions, it follows that $\xi^{(i)}[k] = \varepsilon[\rho_0^k[i]]$. At any point in the Viterbi Algorithm, a surviving path can be reconstructed from a list of survivors, $i_s^{(j)}[k]$, as follows:

$$\begin{aligned} &\rho_k^k[j] = i_s^{(j)}[k] \\ &\rho_n^k[j] = \{i_s^{(\sigma[n+1])}[n], \rho_{n+1}^k\},\ \sigma[n+1] \in \rho_{n+1}^k,\ n = k-1,\ldots,0 \end{aligned} \tag{3.18}$$

This process is better known as *tracing back.*

At the end of a received frame, we can find the least cost path in two steps: first, find the last state in the least cost path; then traceback. Terminate the trellis in the following manner. If the frame contains a known "tail", $\sigma[N + L_h - 1] = (s[N + L_h - 2], \ldots, s[N]) = j_{term}$, then the least cost path through the trellis *must* end with this state. Otherwise, terminate the trellis by searching for the minimum path metric $\xi^{(j)}[N + L_h - 1]$. That is, the

$$j_* = \left\{ \begin{array}{cc} j_{term} & \text{trellis is terminated} \\ \arg\min_j \xi^{(j)}[N + L_h - 1] & \textit{else} \end{array} \right] \tag{3.19}$$

Finally, the least cost path can be reconstructed from a list of survivors, $i_s^{(j)}[k]$ by "tracing back" with equation (3.18) starting with $i_s^{(j_*)}[N + L_h - 2]$.

3.1.3 SUMMARY MLSE WITH VITERBI EQUALIZATION (KNOWN CHANNEL)

1. Allocate an $|\mathcal{A}|^{L_h-1}\times 1$ array of cumulative partial path metrics[4], $\xi^{(i)}[k]$. Initialize $\xi^{(i)}[0]$ according to equation (3.14).

2. Allocate an $|\mathcal{A}|^{L_h-1}\times N+L_h$ array of surviving transitions into the $\sigma[k+1]=j^{th}$ state at the k^{th} stage, $i_s^{(j)}[k]$.

3. Start the iterative recursion

```
For each stage, $k=0,1,\ldots,N+L_h-1$,
    For each stage, $j=0,1,\ldots,|\mathcal{A}|^{L_h-1}-1$,
        Find the survivor (3.12), (3.11), (3.16).
        Update the list of cumulative partial path metrics, $\xi^{(i)}[k]$, (3.15).
```

4. Terminate the trellis with equation (3.19).

5. Traceback: after the last stage of the trellis, reconstruct the least cost path with equation (3.18).

6. Translate a state sequence into a symbol sequence.

[4] Even though there is a time index on $\xi^{(i)}[k]$, the algorithm never requires knowledge of prior path metrics, so you won't need to store them.

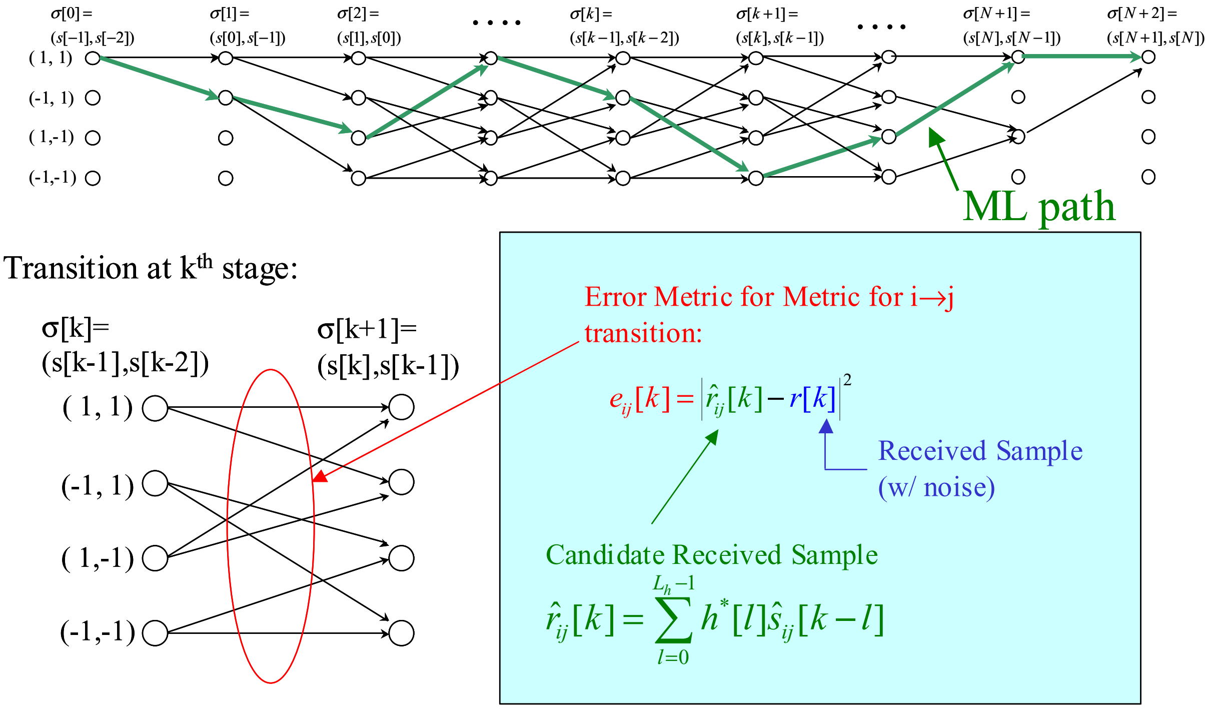
$\sigma[0]=(s[-1],s[-2])$
$\sigma[1]=(s[0],s[-1])$
$\sigma[2]=(s[1],s[0])$
$\sigma[k]=(s[k-1],s[k-2])$
$\sigma[k+1]=(s[k],s[k-1])$
$\sigma[N+1]=(s[N],s[N-1])$
$\sigma[N+2]=(s[N+1],s[N])$
(1, 1)
(-1, 1)
(1,-1)
(-1,-1)
ML path
Transition at kth stage:
σ[k]= (s[k-1],s[k-2])
σ[k+1]= (s[k],s[k-1])
Error Metric for Metric for i→j transition:
$e_{ij}[k]=\left|\hat{r}_{ij}[k]-r[k]\right|^2$
Received Sample (w/ noise)
Candidate Received Sample
$\hat{r}_{ij}[k]=\sum_{l=0}^{L_h-1}h^*[l]\hat{s}_{ij}[k-l]$

Figure 3.2:Summary of the Viterbi Algorithm

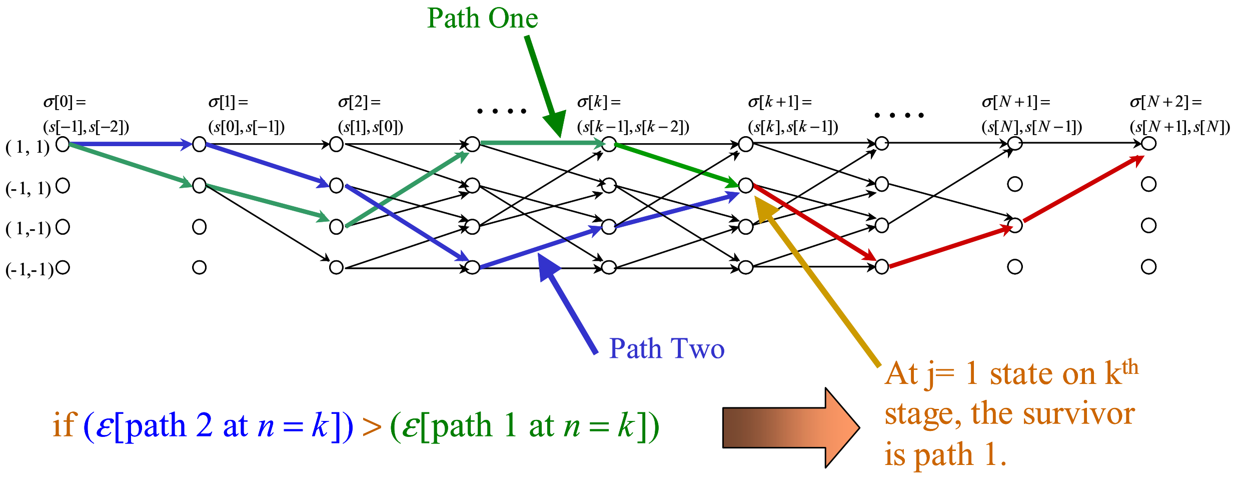
Path One
$\sigma[0]=(s[-1],s[-2])$
$\sigma[1]=(s[0],s[-1])$
$\sigma[2]=(s[1],s[0])$
$\sigma[k]=(s[k-1],s[k-2])$
$\sigma[k+1]=(s[k],s[k-1])$
$\sigma[N+1]=(s[N],s[N-1])$
$\sigma[N+2]=(s[N+1],s[N])$
(1, 1)
(-1, 1)
(1,-1)
(-1,-1)
Path Two
if $(\varepsilon[\text{path 2 at } n=k]) > (\varepsilon[\text{path 1 at } n=k])$
At j= 1 state on kth stage, the survivor is path 1.

Figure 3.3: Operating principle of the Viterbi Algorithm

3.1.4 Application of H-Matrix to Build Trellis

As promised, we can build the Viterbi Equalization Trellis from the Toeplitz channel matrix by inspection. A trellis is completely specified by the sequence of state definitions. If we know the definition of each state, $\sigma[k]$, then we can draw a trellis. In the case of a time-invariant linear FIR channel, the state definition is obvious: let $\sigma[k]$ be the past symbols in the FIR channels tapregisters. However, for more complicated channels, the answer may be come less obvious. So it behooves us to develop a visualization tool with a simple example. Each row of the H-matrix is associated with one term of the sum

$$\| \underline{r} - \mathbf{H}\underline{s} \|^2 = \sum_n | r[n] - \hat{r}[n] |^2$$

So each row of the **H**-matrix is associated with one stage of the trellis. To illustrate this fact, we'll return to the Toeplitz **H** matrix from a previous example. The **H** matrix says that the second received signal sample, $r[1]$, has symbol components from[5] s[-1], s[0], and s[1]. This suggests that the trellis for this stage should account for all possible combinations of s[-1], s[0], and s[1]. This is satisfied by defining the state $\sigma[1] = (s[0], s[\text{-}1])$, and $\sigma[2] = (s[1], s[0])$. A similar argument for each trellis stage will lead to a state definition $\sigma[k] = (s[k\text{-}1], s[k\text{-}2])$.

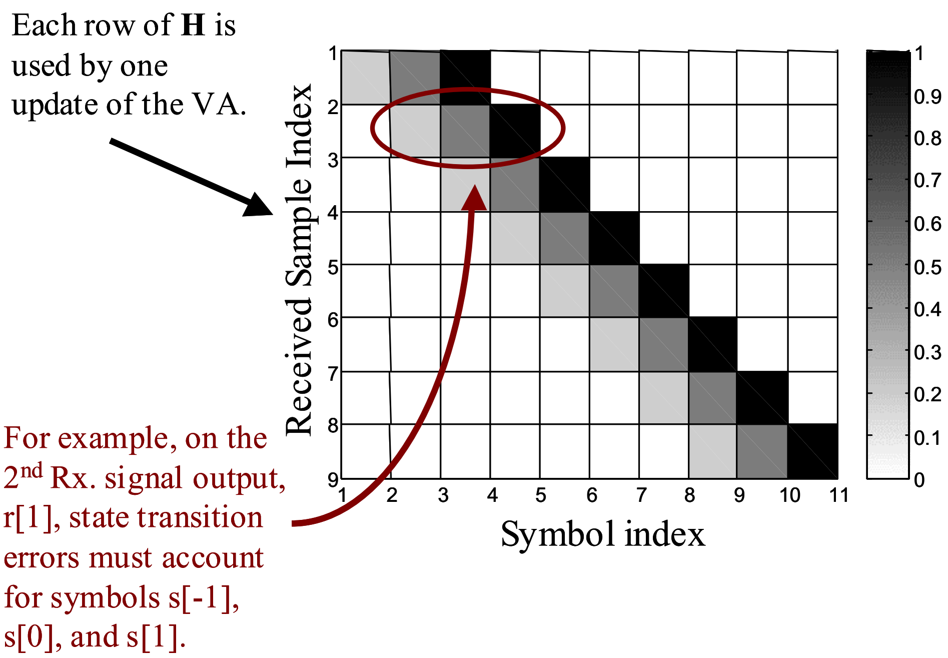
Each row of **H** is used by one update of the VA.
Received Sample Index
1
2
3
4
5
6
7
8
9
1
2
3
4
5
6
7
8
9
10
11
Symbol index
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
For example, on the 2nd Rx. signal output, r[1], state transition errors must account for symbols s[-1], s[0], and s[1].

Figure 3.4: Illustration of how a trellis can be constructed stage by stage from the channel transfer matrix.

[5] Here we deviate slightly from the format of equation (3.4) by assuming that header symbols have been transmitted.

There are a few other issues to consider when defining states. For one, the states must define a trellis such that each forward path corresponds to a distinct symbol sequence. We will address this issue in a later section.

3.1.5 Channel Estimation Issues

Until now, we have assumed that our channel is known. In practical VEQ implementations, the channel must be estimated. Frequently, a channel is estimated with a training sequence before Equalization [54]. Other methods include a channel estimation procedure in tandem with the VA [7]. Channel estimation for VEQ is beyond the scope of this thesis but is still an important issue.

In previous examples, we have constructed trellises for the equalization of BPSK signals. However, the VA can be applied to many different modulation types simply by changing the trellis. As an example, one stage of the trellis for equalization of a π/4 DQPSK signal across a 1st Order FIR channel is illustrated below. Here, instead of labeling states in terms of symbols, we have labeled them in terms of phases of those symbols. The trellis accounts for the fact that symbols must change by odd multiples of 45^0.

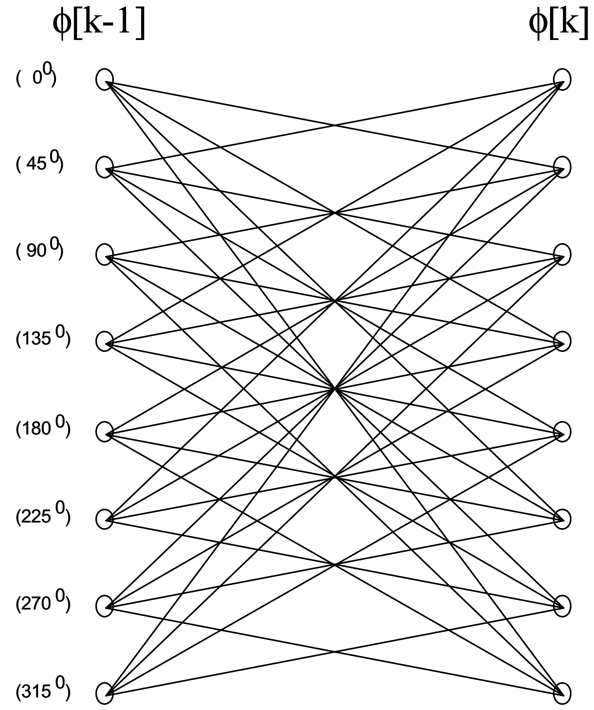

Figure 3.5: Trellis stage for π/4 DQPSK.

3.1.6 Complexity

The complexity of the VA is often measured in terms of the number of states [208]. However, the number of transitions per stage is a better measure since the number of computations required is directly proportional to this number. Moreover, two trellises with the same state-size can have a different number of state transitions, and hence different complexities. Henceforth, we will refer to the number of state transitions/stage as the trellis size, and the number of states/stage as the state-size or trellis depth. The trellis size can be calculated by first calculating the number of states and then the number of incoming transitions per state. For linear modulated signals there is one distinct state value for each possible combination of symbols in $\sigma[n]=(s[n-1],s[n-2],\ldots,s[n-L_h+1])$. There are thus $|\mathcal{A}|^{L_h-1}$ possible values. Likewise, for a given state, $\sigma[n]$, there is one incoming transition for each possible value of $s[n-L_h+1]$. There are $|\mathcal{A}|$ such values. Hence, the trellis size for linear modulated signals is $|\mathcal{A}|^{L_h}$. In contrast, π/4-DQPSK signals can only transition by odd multiples of π/4. Hence, π/4 DQPSK has a state-size as large as 8-PSK but its trellis is more sparsely interconnected.

3.2 Delayed Decision Feedback Sequence Estimation

MLSE has realizable complexity for small channel lengths and small alphabets; however, for long channels, and higher-order alphabets, it becomes extremely complex. This complexity prohibits MLSE's application to very high data rate applications (where the channel delay spread is much longer than a symbol period) or to the equalization of IIR channels, such as those that occur in magnetic recording media. However, the VEQ's ability to equalize channels with very deep nulls without noise emphasis motivates a generalization of MLSE that covers non-linear equalization of very long channels. Many authors have addressed this topic [24]. There are two major approaches to reducing the trellis size of a Viterbi Sequence Estimator:

- Channel Truncation: ignore a portion of the channel in the error metric calculation.
- Decision Feedback: Account for longer channel length by tracing back through survivors.

Before we formally define DDFSE, we will first illustrate a difference between these two approaches with an example.

3.2.1 Example

Consider the channel model from the previous example. The last sample $h[2]$= .2 is much less than the other samples. Truncation would define a channel state with just one symbol $\sigma[n] = (s[n-1])$. Hence, when computing error metrics, only $h[0]$, and $h[1]$ is accounted for $h[2]$= .2 is neglected. The neglected symbol of ISI appears as additive noise in the path metric calculations. In contrast, DDFSE accounts for this extra symbol of ISI by looking back through the trellis for survivors. This is illustrated in Figure 3.6. In this figure, two stages are illustrated. The survivors have already been chosen for the (*k-1*)th stage and the DDFSE algorithm is ready to compute the error metrics for the k^{th} stage. The survivors from the (*k-1*)th stage are clearly labeled. In this particular case, the transitions exiting the $\sigma[k] = (1)$ state look back to find the candidate symbol $s[k-2] = -1$ for error metric computation. Similarly, the transitions exiting the $\sigma[k] = (-1)$ state look-back to find the candidate symbol $s[k-2] = 1$.

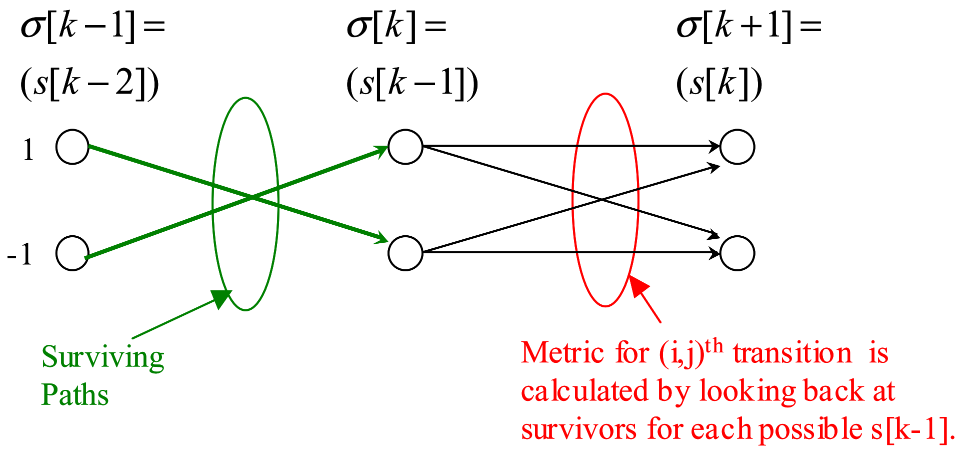
$\sigma[k-1] =$
$(s[k-2])$
$\sigma[k] =$
$(s[k-1])$
$\sigma[k+1] =$
$(s[k])$
1
-1
Surviving Paths
Metric for (i,j)th transition is calculated by looking back at survivors for each possible s[k-1].

Figure 3.6: Looking back through the trellis for delayed decision feedback.

We will now formally define Delayed Decision Feedback Sequence Estimation (DDFSE) and describe its application to the equalization of long FIR channels; although, its principle has been extended to IIR channels as well, we will neglect this application for brevity. DDFSE differs from MLSE only in the calculation of the state metrics. Here, the state size, μ, is a parameter left to the algorithm designer to specify in order to meet some cost/complexity tradeoff. Let the DDFSE state be defined as $\sigma[k] = (s[k-1], s[k-2], \ldots, s[k-\mu])$. For MLSE, $\mu = L_h - 1$ but for DDFSE $\mu \le L_h - 1$. Then the error metric is calculated as follows:

$$e^{(i,j)}[k] = | r[k] - \hat{r}[k] |^2 \tag{3.20}$$

where

$$\hat{r}_{ij}[k] = \sum_{l=0}^{\mu} h^*[l]\hat{s}[k-l] + \sum_{l=\mu+1}^{L_h-1} h^*[l]\hat{s}[k-l] \tag{3.21}$$

and

$$\begin{aligned} \sigma[k] &= (\hat{s}[k-1],\ldots,\hat{s}[k-\mu]) = i \\ \sigma[k+1] &= (\hat{s}[k],\ldots,\hat{s}[k-\mu+1]) = j \\ v[k] &= (\hat{s}[k-\mu-1],\ldots,\hat{s}[k-L_h+1]) \in \rho_0^k[i] \end{aligned}$$

The quantity $v[k]$ is called the feedback state. The value of $v[k]$ is determined only from the surviving partial path into the state $\sigma[k] = i$. The set of indices $\mathcal{U}_e = (0,1,\ldots,\mu)$ is called the enumeration set and the set of indices $\mathcal{U}_{fb} = (\mu+1,\ldots,L_h-1)$ is called the feedback set.

The performance of DDFSE has been analyzed extensively in [24]. Its performance is sensitive to the choice of state size parameter, m. In general, if the feedback set is chosen over a region of *h[n]* with small energy, then DDFSE can well approximate a full-state MLSE. Its application is particularly powerful for minimum-phase channels.

3.2.2 Summary of DDFSE

7. Allocate an $|\mathcal{A}|^{\mu} \times 1$ array of cumulative partial path metrics[6], $\xi^{(i)}[k]$. Initialize $\xi^{(i)}[0]$ according to equation (3.14).

8. Allocate an $|\mathcal{A}|^{\mu} \times N + L_h$ array of surviving transitions into the $\sigma[k+1] = j^{th}$ state at the k^{th} stage, $i_s^{(j)}[k]$.

[6] Even though there is a time index on $\xi^{(i)}[k]$, the algorithm never requires knowledge of prior path metrics, so you won't need to store them.

9. Start the iterative recursion:

    For each stage, $k = 0,1,\ldots, N + L_h - 1$

        For each stage, $j = 0,1,\ldots,|\mathcal{A}|^{\mu} - 1$

            Find the survivor (3.20),(3.21),(3.16).

            Update the list of cumulative partial path metrics, $\xi^{(i)}[k]$, (3.15).

10. Terminate the trellis with equation (3.19)

11. Traceback: after the last stage of the trellis, reconstruct the least cost path with equation (3.18).

12. Translate the resulting state sequence into a symbol sequence.

3.3 CIRCULAR CONVOLUTION

The previous sections discussed Viterbi Equalization of time invariant FIR channel models. Let us now consider a different channel: circular convolutional channels. Circular convolution occurs when an Discrete Fourier Transform (DFT) is used to calculate a convolution [215]. Usually, input signals are zero padded so circular convolution yields exactly the same result as the linear convolution model from the previous section. To the author's knowledge, to date, circular convolution has only been applied in the literature toward understanding the properties of the DFT. Circular convolution should never occur as an impairment in practical communication systems[7], so it is no wonder that it has not been treated in the literature. However, an equalizer for circular convolutional channels will provide a fundamental processor structure for overloaded array processing so it is worth our study.

[7] A special case, some Orthogonal Frequency Division Multiplexed (OFDM) systems force a linear convolutional channel to be circular and then *exploits* the properties of circular convolution with sophisticated modulation.

We will now formally define circular convolution and its corresponding matrix representation. Let $s[d], h[d] \in \mathbb{C}$, be discrete time, complex, sequences of length D_u. Then circular convolution is defined as:

$$y[d] = s[d] \otimes h[d] \triangleq \sum_{l=0}^{D_u-1} h^*[l] s[(d-l) \operatorname{mod} D_u],\ 0 \le d \le D_u - 1 \tag{3.22}$$

where the symbol, $\otimes$, denotes circular convolution and "$\operatorname{mod} D_u$" denotes modulo-D_u indexing. Let the $D_u \times 1$ vectors, $\underline{y} = [y[0] \quad y[1] \quad \ldots \quad y[D_u-1]]^T$ and $\underline{s} = [s[0] \quad s[1] \quad \ldots \quad s[D_u-1]]^T$. Then a linear system of equations, $\underline{y} = \mathbf{H}\underline{s}$ can be developed similar to equation (3.4). In the case of equation (3.4), the matrix, **H**, is Toeplitz. In the case of a circ-conv channel, the matrix, **H**, is circulant: each row of **H** is a circular shift of the previous row. We will now consider an example.

3.3.1 Example

Let D_u= 6, and let $h[d]$ be given by the sequence in Figure 3.7(a). A checkerboard plot of the elements of **H** are given in Figure 3.7(b).

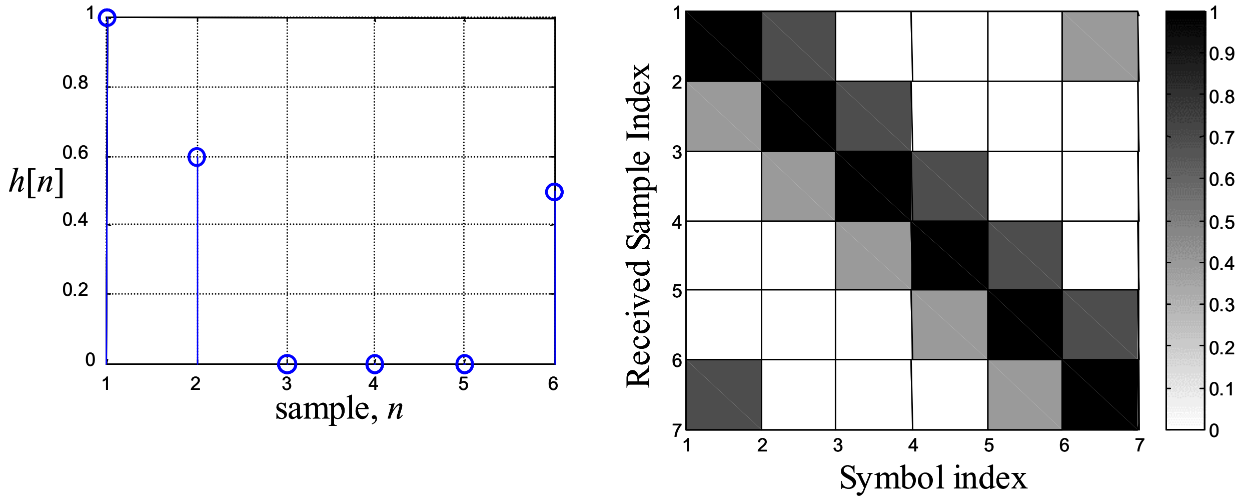

Figure 3.7: (a) (left) Discrete-time FIR circularly-convolutional channel. (b) (right) Checkerboard plot of the channel transfer matrix for $h[n]$ in (a).

Now consider a sequence of symbols from some alphabet $s[d] \in \mathcal{A}$ transmitted over a circularly-convolutional channel. The received signal with noise is

$$r[d] = y[d] + z[d],\ 0 \le d \le D_u - 1 \tag{3.23}$$

where $z[d]$ is AWGN with variance, σ_z^2. Similar to the linear convolution model, the maximum likelihood estimate of the transmitted signal for the received vector, $\underline{r} = \left[r[0] \quad r[1] \quad \ldots \quad r[D_u - 1]\right]^T$ is given as follows:

$$\begin{aligned} \hat{\underline{s}} &= \arg\max_{\underline{s} \in \mathcal{A}^N} f(\underline{r} \mid \underline{s}, \underline{h}) \\ &= \arg\max_{\underline{s} \in \mathcal{A}^N} \left\{ \ln f(\underline{r} \mid \underline{s}, \underline{h}) \right\} \\ &= \arg\max_{\underline{s} \in \mathcal{A}^N} \| \underline{r} - \mathbf{H}\underline{s} \|^2 \end{aligned} \tag{3.24}$$

Also similar to before, the best estimate of the transmitted signal should constrain $\hat{\underline{s}}$ to a finite alphabet. We can construct a trellis that illustrates the ML search by directly observing the structure of **H**. Again, we can define a state sequence that satisfies the following:

- The d^{th} stage of the trellis enumerates over the non-zero entries on the (d+1)th row[8] of **H**.
- There exists a unique path through the trellis implied by this state sequence that yields the maximum likelihood sequence estimate.

A state definition for the previous example that satisfies the above two criterion is:

$$\sigma[d] = (s[d-2], s[d-1]),\ 0 \le d \le D_u - 1 \tag{3.25}$$

where the indexing of the $s[d]$ is performed modulo-D_u. For instance $\sigma[0] = (s[D_u - 1], s[0])$. Figure 3.8 illustrates the trellis implied by this state definition for a QPSK alphabet. The circulant structure of **H** imposes a strange structure to the trellis: it wraps around upon itself. These so-called tail-biting trellises (TBT) have become a hot topic in the area of error correction coding [208,209]. In general, an equalization trellis will be tail-biting if the H-matrix has corners. As we will see in the next chapter, this matrix does not need to be purely circulant.

[8] The (d+1) occurs because we have numbered the rows of **H** with positive integers.

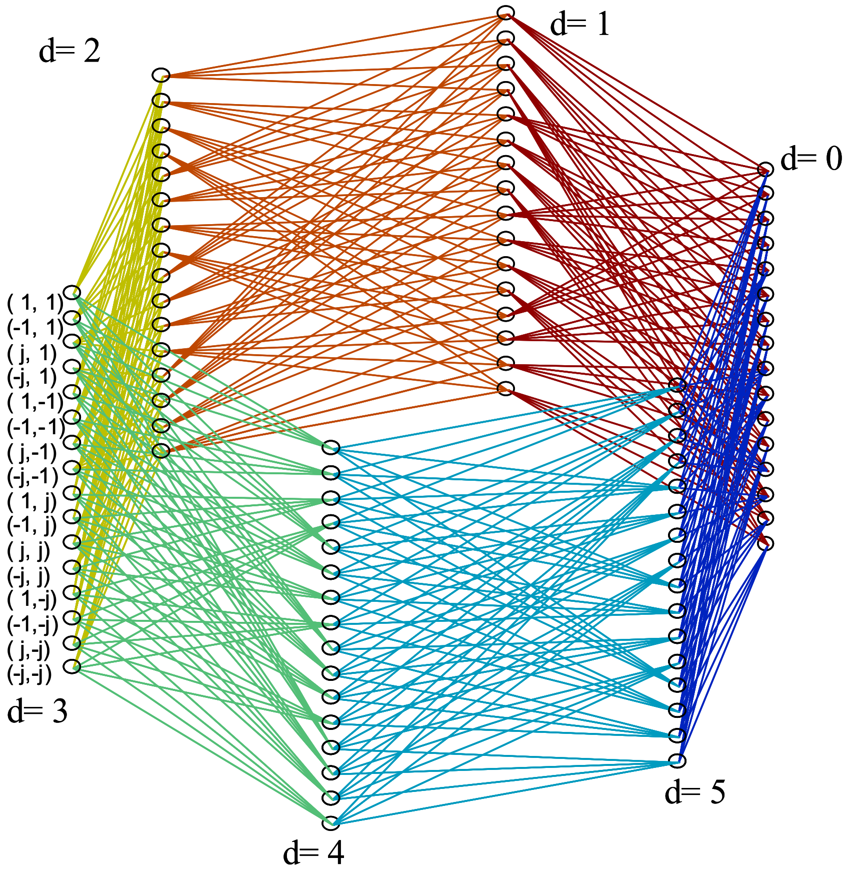

Figure 3.8: Tail-biting trellis for QPSK symbols transmitted over the channel of Equation (3.23)

Although the area of error–correction coding is beyond the scope of this thesis, it is worthwhile to mention, briefly, the application of TBTs to error correction coding. Tail-biting structures have been studied in the coding literature for three reasons: tail-biting convolutional codes are a bandwidth-efficient way of implementing high constraint length convolutional codes with small block lengths. This is because tail-biting convolutional codes do not require trellis termination. Secondly, maximum-likelihood detection of most block codes is equivalent to finding the least cost path around a tail-biting trellis. Finally, and perhaps most importantly, tail-biting trellis has been posed as a fundamental stepping-stone to understanding more complicated decoding algorithms that are not yet fully understood [208]. Iterative decoding of Turbo Codes [216] can be thought of as a sub-optimal approximation to a maximum a posteriori probability (MAP) decoding algorithm on a graph with multiple cycles with large diameters.

3.3.2 Tail-Biting MLSE (TB-MLSE)

The maximum likelihood estimate of $\underline{s}$ corresponds to the least cost *closed* path through the TBT. When equalizing linear convolutional channels, the equalization trellis is flat. In this case, observing that paths can be culled at each stage allows us to apply the Viterbi Algorithm. However, in this case, there is a "chicken before the egg" dilemma; paths cannot strictly be culled at each stage because the concept of a cumulative path metric is not well defined. The true least squares path around a TBT can be obtained from a variant of the Viterbi Algorithm [208]. First let $\mathcal{P}$ denote all closed paths through the TBT. Then let $\mathcal{P}_{\sigma_0[0]}$ denote all closed path through a particular node, $\sigma_0[0]$ at stage zero. We can write Equation (3.24) another way:

$$\begin{aligned} \min_{\underline{s}\in\mathcal{A}^{D_u}} \| \underline{r} - \mathbf{H}\underline{s} \|^2 &= \min_{(\sigma[0],\ldots,\sigma[D_u-1])\in\mathcal{P}} \| \underline{r} - \mathbf{H}\underline{s} \|^2 \\ &= \min_{\sigma_0[0]} \left\{ \min_{(\sigma[0],\ldots,\sigma[D_u-1])\in\mathcal{P}_{\sigma_0[0]}} \| \underline{r} - \mathbf{H}\underline{s} \|^2 \right\} \end{aligned} \qquad (3.26)$$

This tells us that we can find the least cost path through a trellis in two steps:

- Pick a particular value of and form of a sub-trellis of the TBT consisting of all closed paths through this node. Find the minimum closed path through this state by "unwrapping" the sub-trellis and finding the least-cost path with the VA.
- Repeat step one for every possible value of σ[0] and choose the global minimum of all σ[0].

We will call this algorithm Tail-Biting MLSE (TB-MLSE). For the example TBT of Figure 3.8, there are 16 possible values of σ[0]. By the above prescription, TB-MLSE will involve 16 calls of the Viterbi Algorithm. In general, TB-MLSE's complexity is squared that of MLSE for a flat trellis with the same state size, μ.

3.3.3 Iterative Tail-Biting Viterbi Algorithm (ITB-VA)

TB-MLSE's complexity prohibits its use in long channels. However, the VA has been observed by many researchers [209] to converge to a maximum likelihood path after a handful of stages even after improper initialization. This property has motivated many coding researchers to apply the Viterbi-Algorithm iteratively around the TBT with an all-zeros initialization of the cumulative partial path metric [209]. We will call this algorithm the Iterative Tail-Biting Viterbi Algorithm (ITB-VA). Usually, only 2 or three iterations around the TBT is enough to converge [217]. ITB-VA runs the risk of converging to a sub optimal path around the TBT but has a complexity several orders of magnitude less than TB-MLSE.

3.4 Conclusion

This chapter has introduced several trellised based algorithms. The most straight-forward is MLSE with the Viterbi Algorithm. However, this solution becomes prohibitively complex for long channels. The more flexible but sub-optimal DDFSE provides the power of non-linear signal processing with a greatly reduced complexity. DDFSE equalizes channels effectively and efficiently when the majority of the channel's energy is grouped at low delays (e.g. minimum-phase channels). The last section introduced trellis-based algorithms for equalizing circularly-convolutional channels. In particular, the ITB-VA can approximate TB-MLSE with an order of magnitude less complexity. Overloaded array processing will borrow from both DDFSE and ITB-VA.

Chapter 4: Spatially Reduced Search Joint Detection (SRSJD)

4.1 Introduction

Overloaded Array Processing is an attractive option to increase the capacity of wireless systems. In many wireless applications such as “base-station in the sky”, it is prohibitively expensive to increase the capacity of a geographic area by dividing it up into small cells serviced by multiple aircraft. Moreover, aerodynamics as well as size-and-weight requirements impose limitations on the type of antenna used. Hence, it is not possible to increase user capacity with multiple pencil-beam antennas. The focus of this section will be on narrow band linear modulated signals; however, it is the author’s belief that this algorithm can be extended to other signal types with achievable complexity.

Figure 4.1 illustrates the desired capabilities of an overloaded array processing algorithm and our proposed approach. Overloaded array processing should be able to separate more signals than elements in the presence of spatially uncorrelated noise and directive noise. Spatially uncorrelated noise models background noise and interference impinging on the array from a ubiquitous direction. Directive noise can model a large number of low power, co-channel interferers impinging on the array from a specific direction. Or it can model a single co-channel interferer of an unknown type impinging on the array from a specific direction [221]. The structured signals can have varying inter-AOAs, different powers, and may be asynchronous in phase, frequency, and baud.

The innovative approach that we pursue is to perform joint detection with a linear space-time processor followed by a sub-optimal Viterbi-based joint detector. We chose a joint-detection architecture because all narrow-band interference rejection based algorithms found in the literature were reported to have one or both of the following limitations: they fail when co-channel signals of the same type are nearly the same power; or they provide marginal SNR improvement when all homogeneous signals have tight excess bandwidth. Contrastingly, in interference-limited environments, joint detection algorithms can separate near equal power, near-zero excess bandwidth co-channel signals by exploiting only difference in their received phase and amplitude. In overloaded array processing, it is necessary to separate signals in homogeneous environments[9] with exactly the same power and nearly the same AOA. Moreover, in most practical wireless communication systems, excess bandwidth is minimized for maximum spectral efficiency [16]. For these reasons, we have chosen a joint detection approach.[10]

The contributions of this thesis are two fold:

- Reduced Span Linear Filtering: linear space-time processors that lead to efficient sub-optimal joint detection algorithms. This entails separating signals into several overlapping groups with linear processing to facilitate efficient non-linear post-processing.
- A sub-optimal iterative joint detection algorithm that takes full advantage of the above linear pre-processing.

The goal of this thesis is not to find the best Overloaded Array processor, but to demonstrate the potential of this approach. Toward this end, we will investigate the properties of select combinations of cascaded linear/non-linear processors. For simplicity, we will begin by focusing on the problem of separating symbol synchronous co-channel signals. Then, in a following chapter, we will demonstrate the applicability of a similar approach to separate asynchronous co-channel signals with one antenna.

[9] That is, environments with signals all of the same type.

[10] This does not mean that interference rejection techniques are not useful for Overloaded Array processing. We anticipate their application to separating heterogeneous environments. But we do not anticipate their use to signal separation in homogeneous environments.

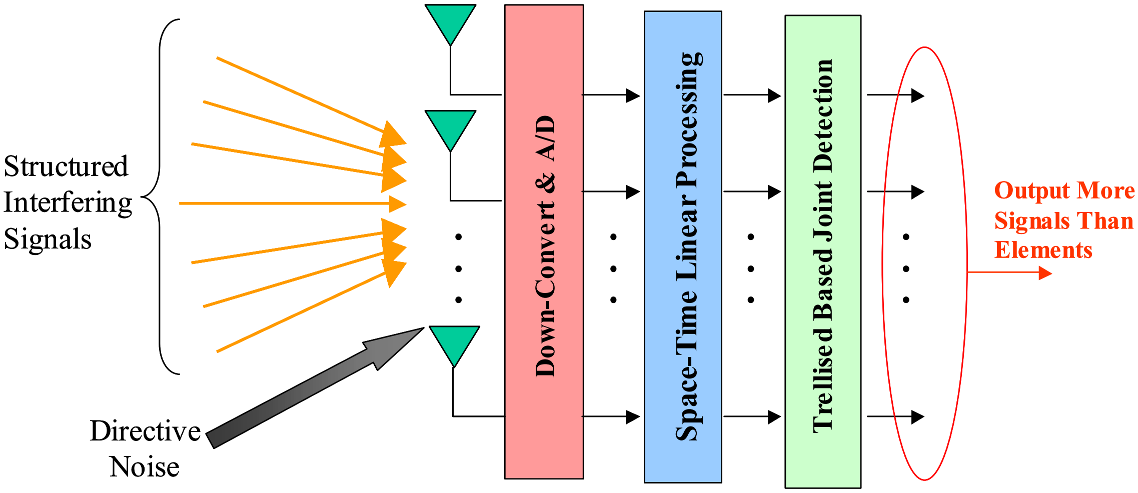

Figure 4.1: Overloaded array processing can separate more structured signals than elements in highly complex signal environments.

In chapter 3, we discussed the joint maximum likelihood detector for synchronous signals impinging on an antenna array. We will see in a following chapter that this receiver has a theoretical performance that facilitates overloaded array processing; however, its complexity is prohibitively expensive. This leads to a natural question: is it necessary to enumerate over all possible signal combinations at once? Intuitively, we know that signals impinging on a calibrated array from widely separated AOAs do not significantly interfere with each other. It then seems possible that we can demodulate a select few signals by only jointly estimating a subset at a time. This section outlines a sub optimal algorithm, which approximates the joint ML receiver with a greatly reduced complexity. The algorithm exploits a wide separation in AOA between angles of arrival by factoring quadratic terms of the maximum-likelihood criterion into trellis-oriented form. For simplicity, in our examples, we consider QPSK signals of known AOAs impinging on a calibrated array. However, we believe that the algorithm is not limited to this case. Several examples are considered including environments with spatially uncorrelated noise.

4.2 A SUBOPTIMAL APPROXIMATION TO THE JOINT MAXIMUM LIKELIHOOD CRITERION

As previously discussed, the channel model for multiple synchronous QPSK signals impinging on an antenna array is as follows

$$\underline{x}[n] = \mathbf{A}\underline{s}[n] + \underline{z}[n] \tag{4.1}$$

where $\underline{s}[n]$, is a $D_u \times 1$ vector of the n^{th} set of symbols impinging on the array from users 1 through D_u. $\underline{x}[n]$ is a $M \times 1$ vector of the n^th received array signal. The matrix **A** is the array response. Finally, $\underline{z}[n]$ is the $M \times 1$ noise vector with zero mean and Auto –correlation $\mathbf{\Phi}_{\underline{zz}} = E\left[\underline{zz}^H\right]$. The auto-correlation matrix can model background thermal noise as well as directional noise (see Appendix B.3).

If no ISI is present on the channel, then the joint-ML criterion for detecting synchronous users impinging on an antenna array can be reduced to a symbol-by-symbol detector. The joint-ML (JML) detector for this case is:

$$\hat{\underline{s}}[n] = \arg\min_{\underline{s}} \left(\underline{x}[n] - \mathbf{A}\underline{s}\right)^H \mathbf{\Phi}_{\underline{zz}}^{-1} \left(\underline{x}[n] - \mathbf{A}\underline{s}\right) \tag{4.2}$$

In the case of spatially uncorrelated noise, $\Phi_{\underline{zz}} = \sigma_z^2 \mathbf{I}$, and the joint-ML detector reduces to least squares enumeration. We seek an alternate form of the ML detector, which can exploit large differences in AOA. Expanding products and dropping terms independent of the optimization, equation (4.2) is equivalent to:

$$\hat{\underline{s}}[n] = \arg\min_{\underline{s}} \left\{ \underline{s}^H \mathbf{A}^H \Phi_{\underline{zz}}^{-1} \mathbf{A}\underline{s} - 2\Re\mathfrak{e}\left\{ (\Phi_{\underline{zz}}^{-1} \mathbf{A}\underline{s})^H \underline{x} \right\} \right\} \tag{4.3}$$

Now, define a $D_u \times D_u$ matrix, **H**, and $D_u \times 1$ matrix, $\underline{y}$, such that

$$\begin{aligned} \mathbf{H}^H \mathbf{H} &= \mathbf{A}^H \Phi_{\underline{zz}}^{-1} \mathbf{A}, \\ \mathbf{H}^H \underline{y} &= \mathbf{A}^H \Phi_{\underline{zz}}^{-1} \underline{x} \end{aligned} \tag{4.4}$$

Further, let $\underline{h}[d]$ be the d^{th} row of **H**. Then, the joint ML receiver can be written as

$$\begin{aligned} \hat{\underline{s}}[n] &= \arg\min_{\underline{s}} \left\| \underline{y} - \mathbf{H}\underline{s} \right\|^2 \\ &= \arg\min_{\underline{s}} \sum_{d=1}^{D_u} \left| y[d] - \hat{y}[d] \right|^2 \end{aligned} \tag{4.5}$$

where $\hat{y}[d] = \underline{h}[d]\underline{s}$. For a reduced complexity search, the matrix **H** should be chosen such that the energy of each row is concentrated on as few symbols as possible. We will consider a brief example.

4.2.1 Example

Consider D_u= 6 synchronous QPSK users equally spaced in angle of arrival impinging on an M= 5 element calibrated circular array with a radius, R_a= $.2\lambda$ in the presence of spatially uncorrelated additive white Gaussian noise. Let[11] $\mathbf{H} = (\mathbf{A}^H\mathbf{A})^{1/2}$. This scenario is illustrated in Figure 4.2.

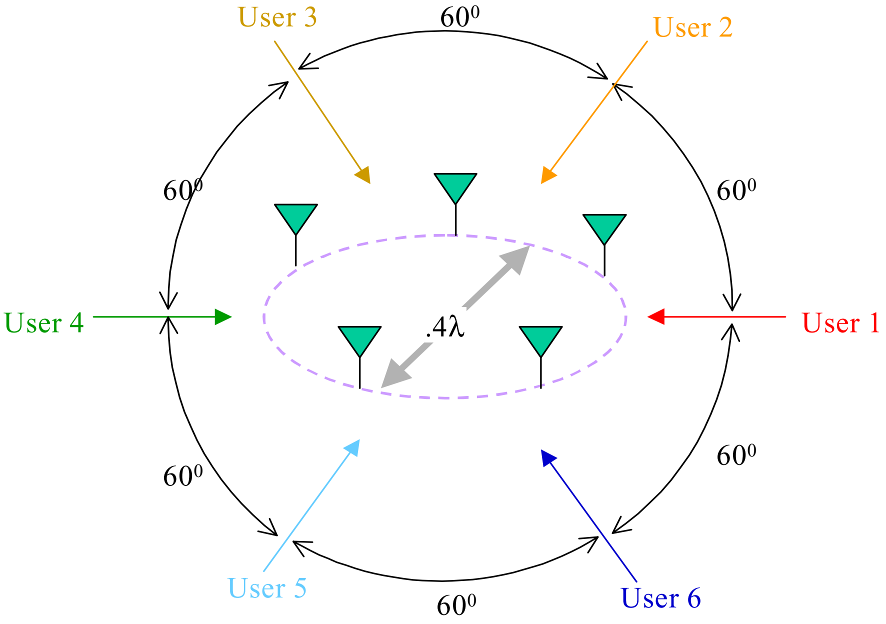

Figure 4.2: Illustration of Example Scenario

Since the matrix $\mathbf{A}^H\mathbf{A}$ is Hermitian symmetric, $\mathbf{H} = \mathbf{H}^H$ is also Hermitian symmetric. Further, let $\mathbf{W} = \mathbf{H}^\dagger\mathbf{A}^H$ and $\underline{y} = \mathbf{W}\underline{x}$. Then $\mathbf{H}$ and $\underline{y}$ satisfy equation (4.4). Instead of explicitly giving **H** numerically, we will plot the magnitude of each element of **H**. This is given in Figure 4.3.

[11] Here $(\ )^{1/2}$ denotes the spectral square root defined more formally later.

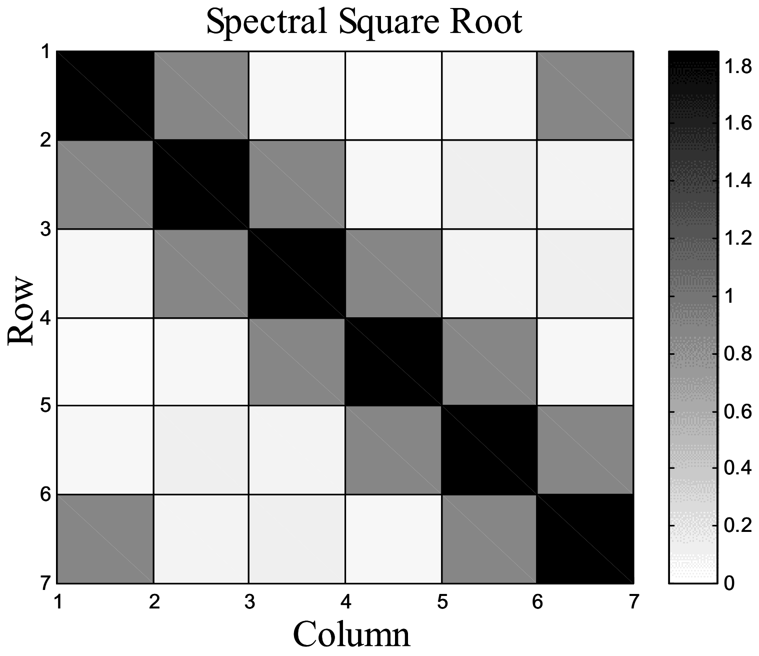

Figure 4.3: Checkerboard plot of the matrix **H**. Each i,jth square displays the magnitude of the i,jth element of **H**.

Note that the most of the energy of **H** is concentrated along the diagonal. Each row of **H** is used in one summation term of equation (4.5). The matrix **H** resembles a circulant matrix[12]. That is, the equation $\underline{y} = \mathbf{H}\underline{s}$ resembles the matrix form of circular convolution $\hat{y}[d] = h[d] \otimes s[d]$ defined in a previous chapter. The off diagonal rows where |i-j| > 2 (with the exception of the corners) are near zero. In the overloaded case, the matrix **H** is rank *M*. The d^{th} row of the matrix **W** may be considered as a beam-former for the d^{th} signal. The beam pattern corresponding to each row is illustrated in Figure 4.4.

[12] A circulant matrix is a matrix where each row is a circular shift of the previous row.

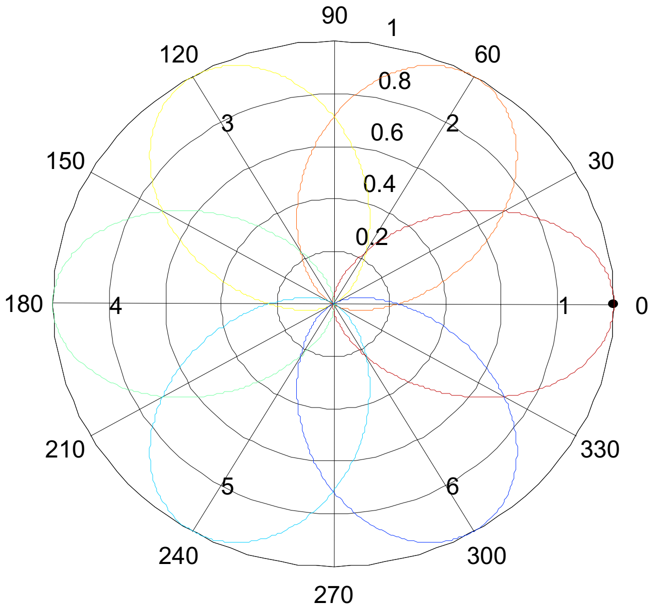

Figure 4.4: A polar plot of the implicit beams formed by the operation $\underline{y} = \mathbf{W}\underline{x}$. Angles of arrival are labeled in degrees. Beams are normalized to their peak amplitude. Each beam is labeled with its corresponding row in **W**. Clearly, in this case the dth beam focuses on the dth user.

In the previous example, we call **W** a reduced-span spatial filter. The object of the linear processing stage is to separate the signals in the environment into a series of overlapping groups. These groups should be made as small as possible in order to reduce the complexity of the subsequent joint-detection stage. We have derived such a processor by factoring a matrix of channel parameters into a *trellis-oriented form* where the energy on each row is focused on a specific column. We will now discuss the class of such factorization that preserve the maximum-likelihood criterion.

The square root of a Hermitian symmetric matrix is not unique. However, a particular square root can be described in terms of the eigen-value decomposition of:

$$\mathbf{A}^H \mathbf{\Phi}_{\underline{zz}}^{-1} \mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^H \tag{4.6}$$

where $\mathbf{\Lambda}$ is a diagonal matrix of eigen-values and $\mathbf{U}$ are an orthonormal set of eigen-vectors.

$$\mathbf{H} = (\mathbf{A}^H \mathbf{\Phi}_{\underline{zz}}^{-1} \mathbf{A})^{1/2} = \mathbf{V}\mathbf{\Lambda}^{1/2}\mathbf{V}^H \tag{4.7}$$

Hence, a straightforward way of taking a square root of a Hermitian symmetric matrix is to take the square root of its eigen-values (all-real). However, if **A** is rank *M*, (e.g. all of the AOAs are distinct) then, by Sylvester's Rank Inequality [222], the matrix $\mathbf{A}^H\mathbf{\Phi}_{\underline{zz}}^{-1}\mathbf{A}$ will also be rank *M* and, by equation (4.7), **H** will be rank *M*.

In the previous example, we satisfied equation (4.4) with a spectral square-root factorization of $\mathbf{A}^H\mathbf{\Phi}_{\underline{zz}}^{-1}\mathbf{A} \propto \mathbf{A}^H\mathbf{A}$. However, at this juncture, its not understood if this is the only applicable factorization. For instance all unitary[13] rotations, **U,** satisfy equation (4.4). This is true because if we denote the spectral square-root as

$$\tilde{\mathbf{H}} = (\mathbf{A}^H\mathbf{\Phi}_{\underline{zz}}^{-1}\mathbf{A})^{1/2} = \tilde{\mathbf{H}}^H \tag{4.8}$$

and if we let

$$\mathbf{H} = \mathbf{U}\tilde{\mathbf{H}} \tag{4.9}$$

for any unitary, **U**,

$$\begin{aligned}\mathbf{H}^H\mathbf{H} &= (\mathbf{U}\tilde{\mathbf{H}})^H\,\mathbf{U}\tilde{\mathbf{H}} \\ &= \tilde{\mathbf{H}}^H\mathbf{U}^H\mathbf{U}\tilde{\mathbf{H}} \\ &= \tilde{\mathbf{H}}^H\tilde{\mathbf{H}} = \tilde{\mathbf{H}}^2 \\ &= \mathbf{A}^H\mathbf{\Phi}_{\underline{zz}}^{-1}\mathbf{A}\end{aligned} \tag{4.10}$$

which satisfies equation (4.4). Since unitary rotations preserve rank, the matrix **H** will have the rank properties of its Hermitian symmetric cousin, $\tilde{\mathbf{H}}$. There is an infinite number of **U** we can choose, and hence, there is an infinite number of **H** which satisfy equation (4.4). On a final note, we should briefly discuss our choice of $\underline{y}$. The vector $\underline{y}$ must follow our choice of **H**. We choose **H** to minimize the complexity of our joint detection algorithm. Then, we choose $\underline{y}$ to complete the square of our cost function. Hence, when we come to our choice of $\underline{y}$, the quantities on the right side of equation (4.4)(b) are known and the matrix **H** is known. In general, since **H** is not full rank, the system $\mathbf{H}\underline{y} = \underline{b}$ may not have a solution. However, in Appendix A, we show that the system (4.4) (b) always has a solution. This existence has to do

[13] A matrix is unitary if $\mathbf{U}^\mathrm{H}\mathbf{U}= \mathbf{I}$

with the fact that no matter which matrix, **H**, we choose, the row-space of **H** is the same as the row-space of **A**. In this thesis we will always choose the pseudo-inverse:

$$\underline{y} = \left(\left(H^{H}\right)^{\dagger} \mathbf{A}^{H} \mathbf{\Phi}_{\underline{zz}}^{-1}\right)\underline{x} = \mathbf{W}\underline{x} \tag{4.11}$$

4.3 ITB-DDFSE

This section discusses the Viterbi-based joint detection portion of the receiver: Iterative Tail-Biting Delayed Decision Feedback Sequence Estimation (ITB-DDFSE). We will now explain how the factored matrix, **H** can be applied in equation (4.5) to reduce the complexity of an ML-like search. Comparing with the plot in the last example of chapter 3, we see that **H** is very similar to a circularly-convolutional channel; both are tail-biting. The one subtle difference is that the off-diagonal elements of **H** in this case are not exactly zero. We will develop an ITB-DDFSE algorithm that provides an approximation to equation (4.5).

Let us divide the estimate $\hat{y}[d] = \underline{h}[d]\underline{s}$ into two components: an enumeration term, $\hat{y}_e[d]$, and feedback term, $\hat{y}_{fb}[d]$. So, $\hat{y}[d] = \hat{y}_e[d] + \hat{y}_{fb}[d]$. Let $\mathcal{U}_e[d]$ be the set of signal indices to jointly detect in order to minimize the d^{th} term of equation (4.5). Let $\mathrm{U}_{fd}[d]$ be the set of signal indices to cancel with feedback in d^{th} term of equation (4.5). Finally, let $\bar{\mathcal{U}}_e[d]$ be the compliment of $\mathcal{U}_e[d]$ such that $\mathcal{U}_e[d] \cup \bar{\mathcal{U}}_e[d] = \{1, 2, \ldots, D_u\}$. Then $\hat{y}_e[d]$ and $\hat{y}_{fb}[d]$ are defined as follows:

$$\begin{aligned} \hat{y}_e[d] &= \sum_{u \in \mathcal{U}_e} h[d,u]s[u] \\ \hat{y}_{fb}[d] &= \sum_{u \in \mathcal{U}_{fb}} h[d,u]s[u] \end{aligned} \tag{4.12}$$

If the set of non-enumeration signals $\left\{s[d] \mid s[d] \in \bar{\mathcal{U}}_e\right\}$ considered at the d^{th} stage are assumed to be zero, equation (4.5) is equivalent to finding the path with the minimum cost through a trellis, which wraps around upon itself. Such a trellis for Example 4.2.1 is illustrated in Figure 4.5 and Figure 4.6. One stage of the trellis is illustrated Figure 4.5. The entire trellis is illustrated in Figure 4.6.

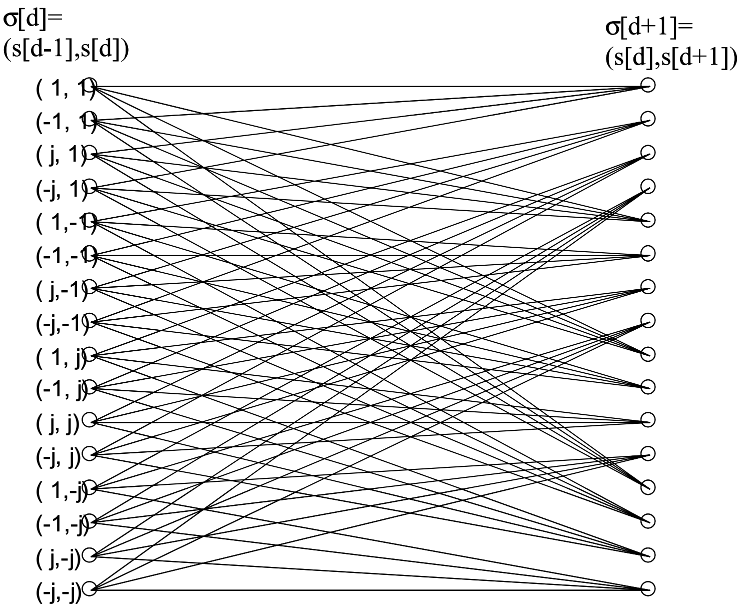

Figure 4.5: One stage of the reduced search trellis for Example 4.2.1.

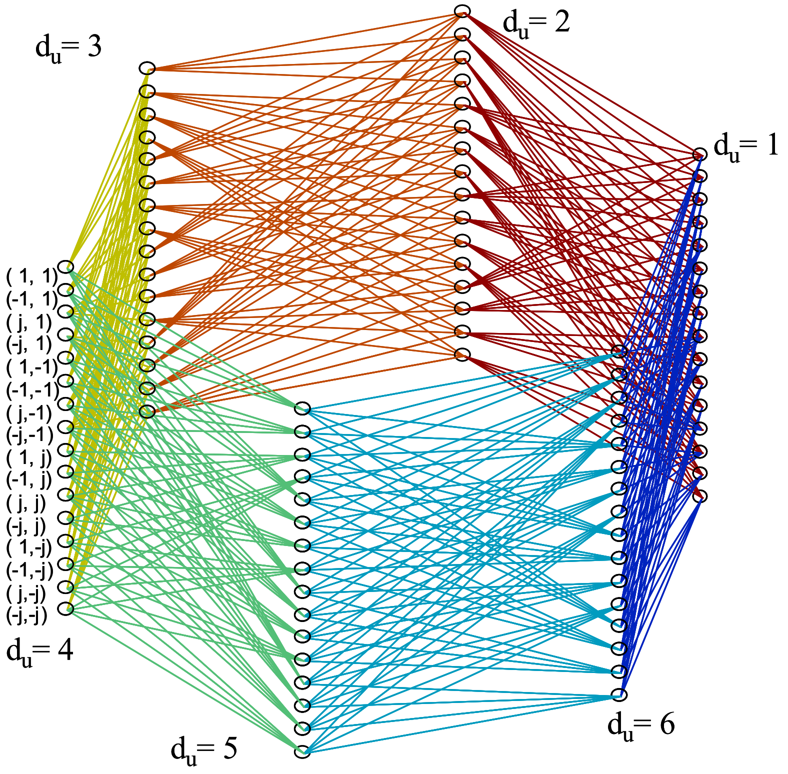

Figure 4.6: Reduced Search Trellis for Example 4.2.1. Each face of the trellis is identical to Figure 4.5. The dth face can be associated with the joint detection of the dth user with a select number of dominant interferers.

The tail-biting property of the joint detection trellis arises from the fact that each signal experiences significant interference from signals adjacent in AOA. Had the sixth user not interfered with the first user, the reduced search trellis would be flat.

As we will see, the sequence of sets, $\mathcal{U}_e[d]$, completely describe the joint detection trellis. Usually $\mathcal{U}_e[d]$ is chosen to include the dominant interfering symbols in each element of $y[d]$ (that is, the entries on the d^{th} row of **H** with the most energy).

The estimate of the interfering signals can be derived in several ways:

- Full Decision Feedback: estimate the signals $\bar{\mathcal{U}}_e[d]$ by looking back through survivors in the cylindrical trellis, i.e. $\mathcal{U}_{fb}[d] = \bar{\mathcal{U}}_e[d]$.
- Truncation: assume that symbols in the set $\bar{\mathcal{U}}_e[d]$ are zero: i.e. $\mathcal{U}_{fb}[d] = \varnothing$.
- Partial Decision Feedback: a combination of the two approaches, i.e. $\mathcal{U}_{fb}[d] \subset \bar{\mathcal{U}}_e[d]$.

At this point, it is not clear which method, if any is best. In the subsequent examples, we will begin with truncation and apply decision feedback as reliable estimates of interfering signals become available.

We will conclude our discussion of the algorithm with two more examples. The first example presents results for an 8-element circular array with a radius of $R_a = \lambda/4$. The second example presents the same number of users with spatially uncorrelated AWGN.

4.3.1 EXAMPLE

Consider D_u= 16 signals, equally spaced in AOA impinging on an M=8 element calibrated circular array with a radius, R_a= .25λ and spatially uncorrelated AWGN. Let $\mathbf{H} = (\mathbf{A}^H\mathbf{A})^{1/2}$ where the $(\;)^{\frac{1}{2}}$ operator denotes the spectral square root. A checkerboard plot of the matrix **H** is provided in Figure 4.7.

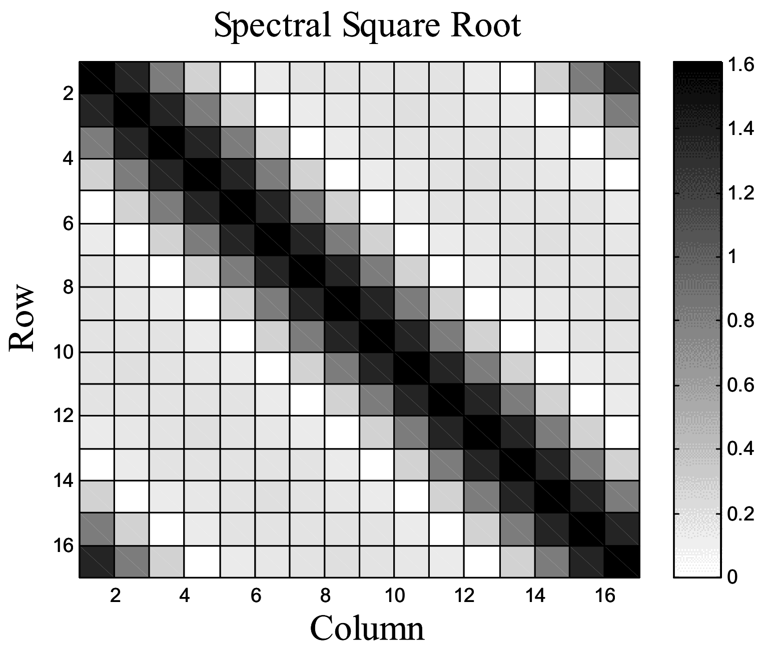

Figure 4.7: Checkerboard plot of the spectral-factorization, **H,** for environment of Example 4.3.1.

Similar to the case in Example 4.2.1, most of the energy of **H** is concentrated along the diagonal. Also, like Example 4.2.1, **H** is in tail-biting form. However, there are two important differences. Firstly, there are more large off-diagonal values. Secondly, the small off-diagonal values have a larger magnitude than the small off-diagonal values of Example 4.2.1. The reduced search trellis for synchronous QPSK users will consist of a cylindrical trellis with D_u= 16 faces and 4^4= 256 states/stage. This is still a great computational savings over a brute-force search compared to $4^{16} \approx 4 \bullet 10^9$ possible interfering signal values.

4.3.2 Example

Consider D_u= 17 signals equally spaced in angle of arrival impinging on an M= 8 element calibrated circular array with a radius, R_a= $\lambda/2$. Let $\mathbf{H} = (\mathbf{A}^H \mathbf{A})^{1/2}$ where the $(\ )^{1/2}$ operator denotes the spectral square root. In addition to D_u =16 equally spaced users, an AWGN signal impinges on the array from an AOA of 0^0. The power of this AWGN is equal to that of the other users. This scenario is illustrated in Figure 4.8. A checkerboard plot of the matrix **H** is provided in Figure 4.9.

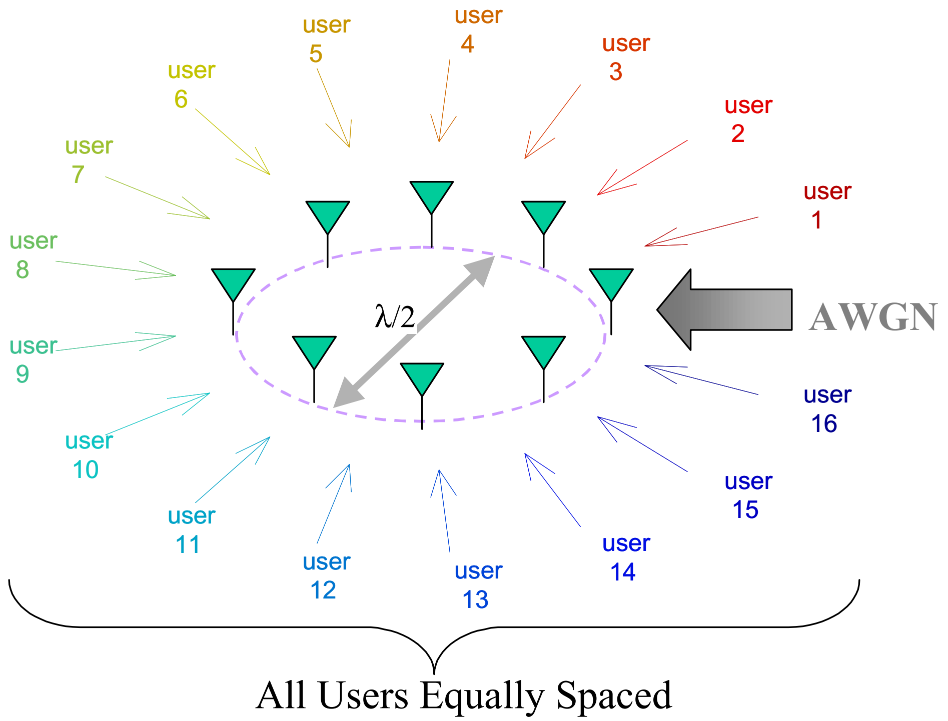

Figure 4.8: illustration of the scenario considered in Example 4.3.2.

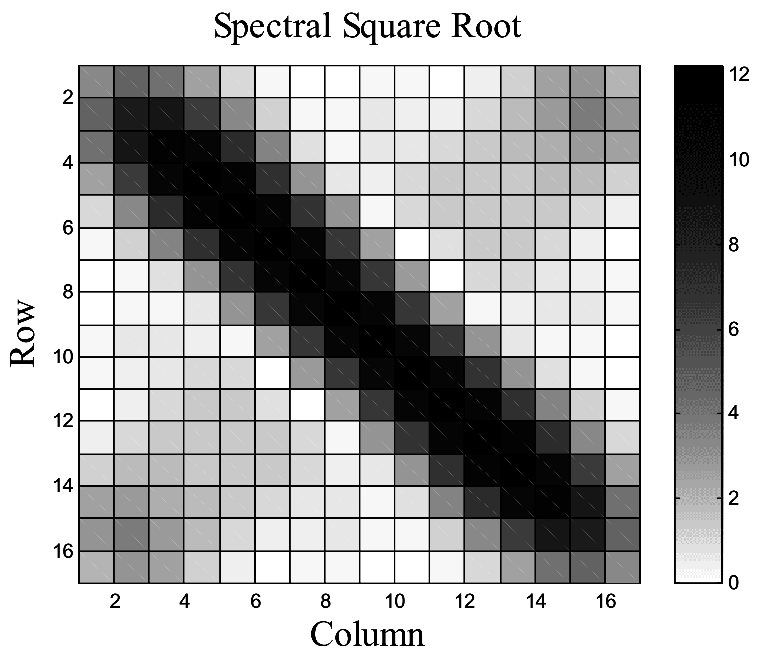

Figure 4.9: Spectral square root factorization, **H**, for example 4.3.2

Note in Figure 4.8 that the spectral square root is distorted by the spatially correlated AWGN source. The spatial-“whitening” of the noise source shows up as weak entries for users close to the directive noise in AOA: users 1,2,15,16. The smaller SINR experienced by these users is not an artifact of the spectral square-root factorization, but a physical consequence of directive noise. Figure 4.9 suggests a reduced search QPSK trellis with 16 faces and 4^4=256 labels/face.

4.3.3 EXAMPLE

Consider 16 equal power signals impinging on an *M*= 8 element calibrated circular array with a radius, R_a= *λ/2* and spatially uncorrelated AWGN. The users are bunched around an AOA of $180^{0.}$ The AOA degree spacing between users are as follows:

$[36^0\ \ 29^0\ \ 23^0\ \ 19^0\ \ 15^0\ \ 12^0\ \ 11^0\ 10^0\ \ 11^0\ \ 12^0\ \ 15^0\ \ 19^0\ \ 23^0\ \ 29^0\ 36^0\ \ 60^0]$

where the first represents the spacing between user 1 and user 2 the second between user 2 and user 3 and so on. The last AOA value is the difference between user 16 and user 1. This scenario is illustrated in Figure 4.10. A checkerboard plot of the matrix **H** is provided in Figure 4.11.

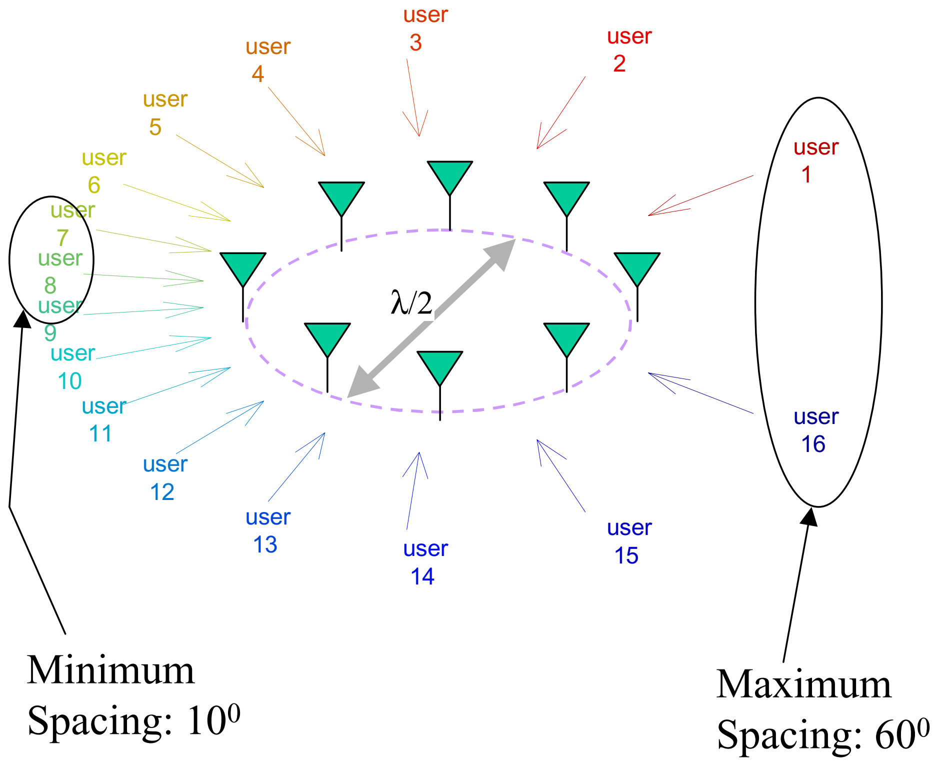

Figure 4.10:Illustration of the scenario considered in Example 4.3.3.

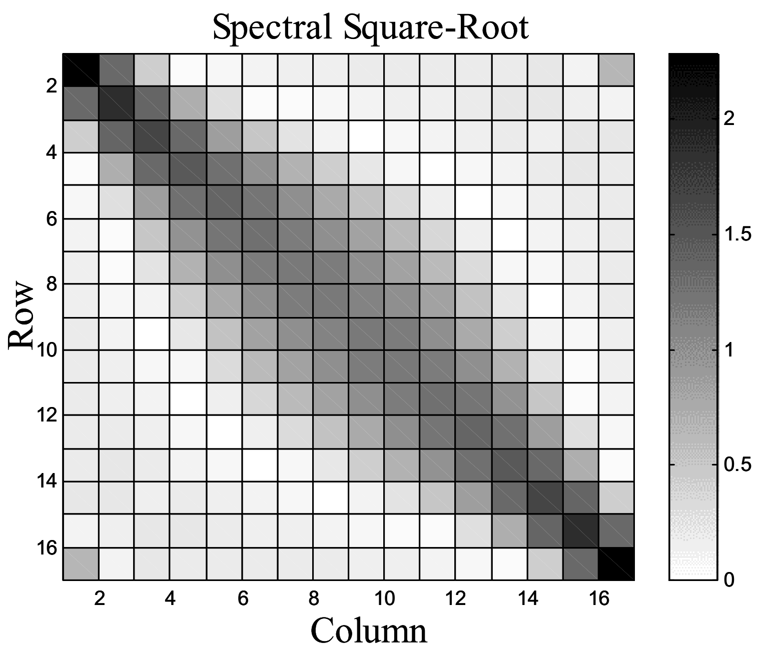

Figure 4.11: Spectral square root factorization for the scenario in Example 4.3.3.

This figure suggests a lopsided tail-biting trellis whose size varies with each stage. The first set of states will consist of all possible combinations of signals 1, and 16. The second set of states in the trellis will consist of all possible combinations of users 1,2, and possibly 3. The size of the trellis will grow until the 8th stage that will consist of all possible combinations of users signals 5-10. Thereafter the trellis size will shrink.

We will conclude our discussion of the algorithm with a few comments on factorizations for other array geometries. Firstly, it has been found that the spectral factorization is highly sensitive to array radius. It should be noted that this is not true for the brute-force JML search which has been found to be very insensitive to array geometry. The difference is that SRSJD relies on a front-end linear transformation to reduce the complexity of the search. We expect that this sensitivity to array radius can be improved with factorizations other than the spectral square root. However, for large array sizes $M > 5$, this dependence seems to be less important. Secondly, for a given overloading ratio, D_u/M, spectral factorizations suggest a lower μ for higher M_{el}. Moreover, for small array sizes, e.g. $M= 2, 3$, very little reduction in complexity is achievable.

4.3.4 TRELLIS CONSTRUCTION

We will now discuss how to construct a trellis from a sequence of enumeration sets: $\mathcal{U}_e[d]$. Before we do, we will introduce the concept of a sparsity pattern. A sparsity pattern is a way of visualizing the sequence of enumeration sets, $\mathcal{U}_e[d]$. It can be obtained from the matrix, **H**, simply by coloring the near-zero entries of **H** with light, and the non-zero entries of **H** with dark (e.g. one-bit b/w plot for **H**). For purposes of illustration, we will limit our sparsity patterns to sizes of small D_u. Consider the matrix of Example 4.2.1. There are three dominant interfering signals on each element of $\underline{y}$. Hence, the sequence of enumeration sets is:

$$\begin{aligned}
\mathcal{U}_e[1] &= \{6,1,2\} \\
\mathcal{U}_e[2] &= \{1,2,3\} \\
\mathcal{U}_e[3] &= \{2,3,4\} \\
\mathcal{U}_e[4] &= \{3,4,5\} \\
\mathcal{U}_e[5] &= \{4,5,6\} \\
\mathcal{U}_e[6] &= \{5,6,1\}
\end{aligned}$$

From the sequence of enumeration sets, $\mathcal{U}_e[d]$, we construct a sparsity pattern by coloring the dominant entries of the d^{th} row of **H** in $\mathcal{U}_e[d]$ dark, and the rest light. This is illustrated in Figure 4.12.

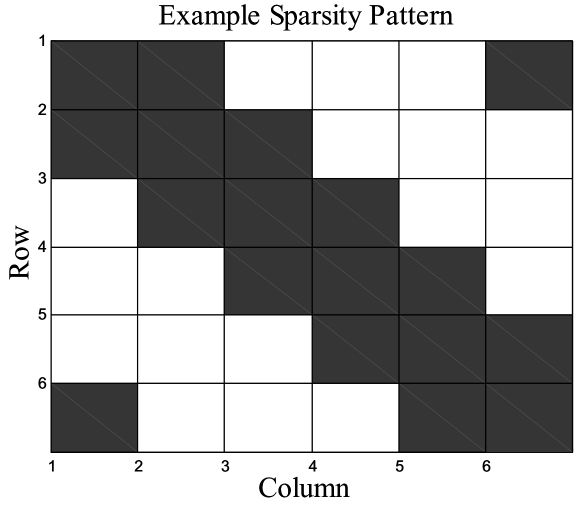

Figure 4.12: Sparsity pattern corresponding to example 4.2.1

It should be obvious how this sparsity pattern can be constructed visually from **H**. Conversely, the sequence of enumeration sets, $\mathcal{U}_e[d]$, can be constructed directly from a sparsity pattern by inspection. In [208], Calderbank illustrates how a tail-biting trellis may be constructed from a sparsity pattern. The method is based on that of Kschischang and Sorokine, which forms a cross product of elementary trellises, obtained from the columns of **H**. The method has the drawback that there is not a direct relationship between states and symbols that form those states. We propose an alternative, more straightforward, technique that facilitates delayed decision feedback. The method assumes that $\mathcal{U}_e[d] \subseteq \mathcal{U}_e[d-1] \cup \mathcal{U}_e[d+1]$[14]. This is equivalent to saying that if user *d+1* has a dominant entry in the d^{th} row of **H**, then, the d^{th} user will have a dominant entry in the $(d+1)^{\text{th}}$ row of **H**. Under this assumption we can apply the following state definition:

$$\sigma[d] = \mathcal{U}_e[d-1] \cap \mathcal{U}_e[d] \tag{4.13}$$

This assures that at the d^{th} stage of the trellis, we will be enumerating over

[14] Of course, indexes are wrapped back into the index set $\{1, 2, \ldots, D_u\}$

$$\begin{aligned}\sigma[d]\cup\sigma[d+1] &= (\mathcal{U}_e[d-1]\cap\mathcal{U}_e[d])\cup(\mathcal{U}_e[d]\cap\mathcal{U}_e[d+1]) \\ &= \mathcal{U}_e[d]\end{aligned} \tag{4.14}$$

which is the desired result. For convenience, let us denote the size of the state definition of the d^{th} state as $\mu[d]=|\sigma[d]|$. In the previous example, this state definition will result in the state sequence:

$$\begin{aligned}\sigma[1]&=\{s[6],s[1]\}\\ \sigma[2]&=\{s[1],s[2]\}\\ \sigma[3]&=\{s[2],s[3]\}\\ \sigma[4]&=\{s[3],s[4]\}\\ \sigma[5]&=\{s[4],s[5]\}\\ \sigma[6]&=\{s[5],s[6]\}\end{aligned}$$

The size of all the states are uniform: $\mu[d]=2,\ \forall d$. This state sequence implies a TBT illustrated in Figure 4.6.

With this state definition in hand, we can now describe Iterative Tail Biting Delayed Decision Feedback Sequence Estimation (ITB-DDFSE) and its application to the overloaded array problem. Let $\rho_{d+1}^{d}[i]$ denote the surviving partial path around[15] the TBT on the $\sigma[d]=i^{th}$ state of the d^{th} stage. This path starts at the $d+1^{\text{th}}$ stage and ends at the d^{th} stage. Then the cost of the $(i\rightarrow j)^{th}$ transition is

$$e^{(i,j)}[d]=y[d]-\hat{y}_{ij}[d] \tag{4.15}$$

where

$$\begin{aligned}\hat{y}_{ij}[d]&=\hat{y}_e^{(i,j)}[d]+\hat{y}_{fb}^{(i,j)}[d]\\ \hat{y}_e^{(i,j)}[d]&=\sum_{u\in\mathcal{U}_e[d]}h[d,u]\hat{s}[u]\\ \hat{y}_{fb}^{(i,j)}[d]&=\sum_{u\in\mathcal{U}_{fb}[d]}h[d,u]\hat{s}[u]\end{aligned} \tag{4.16}$$

and the candidate symbol values in the above expression are taken from the state values

[15] Of course this path may wrap around so $\rho_{d+1}^{d}=(\sigma[d+1],\ldots,\sigma[D_u],\sigma[1],\ldots,\sigma[d])$

$$\begin{aligned} &\sigma[d] = i \\ &\sigma[d+1] = j \\ &v[d] = \{s[u], u \in \mathcal{U}_{fb}[d]\} \in \rho_{d+1}^{d}[i] \end{aligned} \qquad (4.17)$$

Then, the Viterbi Algorithm update is the same as for MLSE:

$$i_s^{(j)}[k] = \arg\min_{i \in \mathcal{A}_j} \left\{ \xi^{(i)}[k] + e^{(i,j)}[k] \right\} \qquad (4.18)$$

$$\xi^{(j)}[k+1] = \min_{i \in \mathcal{A}_j} \left\{ \xi^{(i)}[k] + e^{(i,j)}[k] \right\} \qquad (4.19)$$

4.3.5 Example: Sparsity-Pattern for Lop-Sided Trellises

We will illustrate the construction of a tail-biting trellis for a more complicated result. Consider the sparsity pattern illustrated in Figure 4.13.

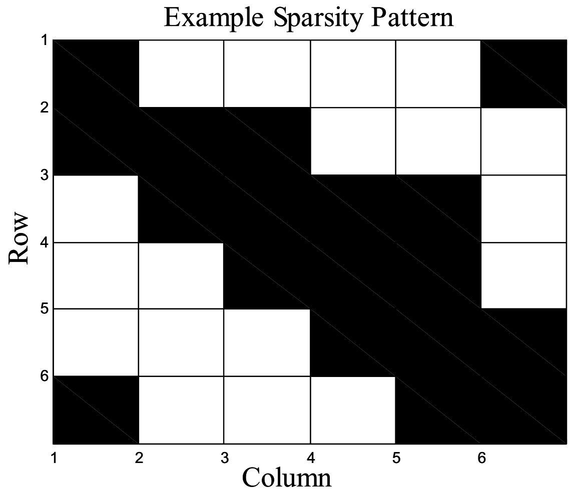

Figure 4.13: Example sparsity pattern for a lop-sided TBT.

This sparsity pattern is analogous to the one generated by the **H** matrices in Figure 4.11 in that it has a row-span that differs on each row. This type of sparsity pattern might occur when users 3, 4, and 5 are closely spaced together but the other users are relatively far apart. The TBT generated by this sparsity pattern for an environment with all BPSK signals is illustrated in Figure 4.14. In this case, the trellis is drawn flat with the first set of states being drawn at the beginning and end of the diagram. The state definition sequence generated by equation (4.13) is clearly labeled at the top of each stage. The sequence of state sizes is obviously

$\mu[d] = \{2,1,2,3,2,2\}$. The value of each state is also labeled in vector form. The deepest portion of the trellis corresponds to the row with the widest span.

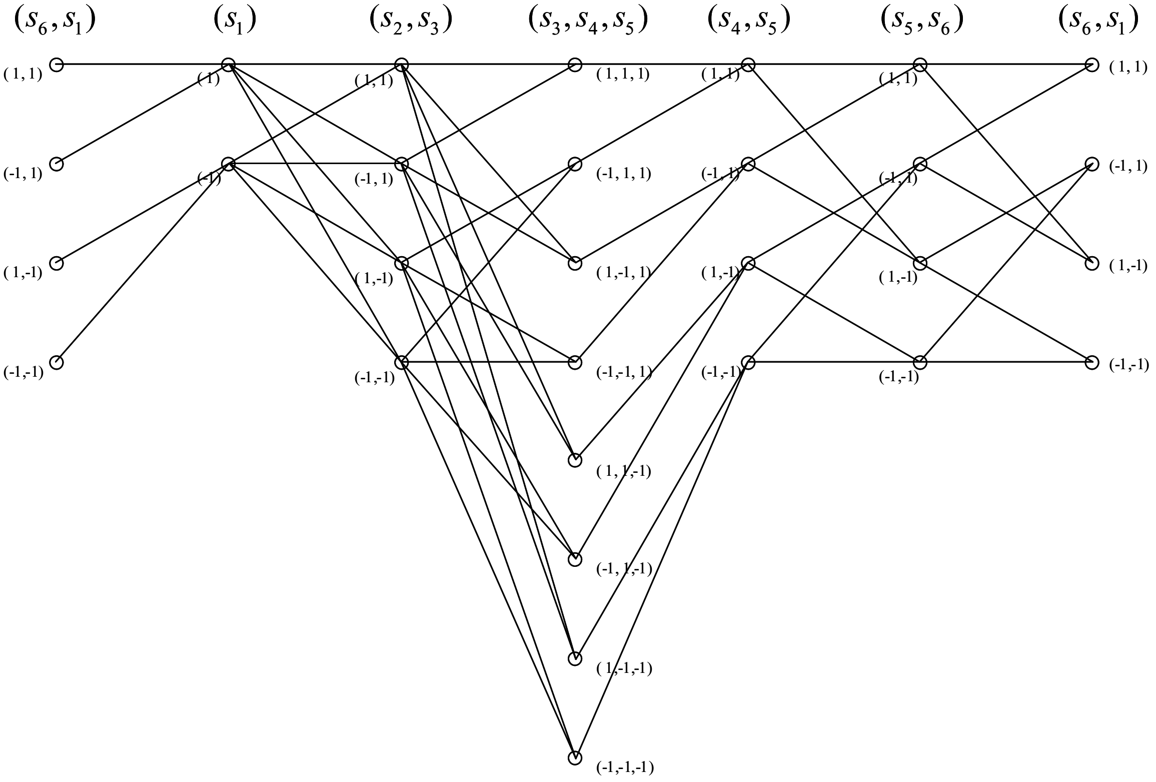

Figure 4.14:Tail-biting trellis for a sparsity pattern in Figure 4.13 and Example 4.3.5.

4.3.6 EXAMPLE

As a final example, we will consider a situation where the joint detection problem is separable. That is, the signal environment consists of two isolated sets of interfering signals. This situation is suggested by the sparsity pattern in Figure 4.15. Here, users 4,5, and 6 do not interfere with users 1,2, and 3. The resulting joint detection trellis contains stages with just one state. At these stages a final decision is made for one interference set. Because these stages contain one state, the joint detection trellis is no longer tail biting. However, if the original **H** matrix that suggests this sparsity pattern has non-zero elements outside the sparsity-pattern, treating the trellis like a TBT and applying the ITB-DDFSE algorithm may yield better results.

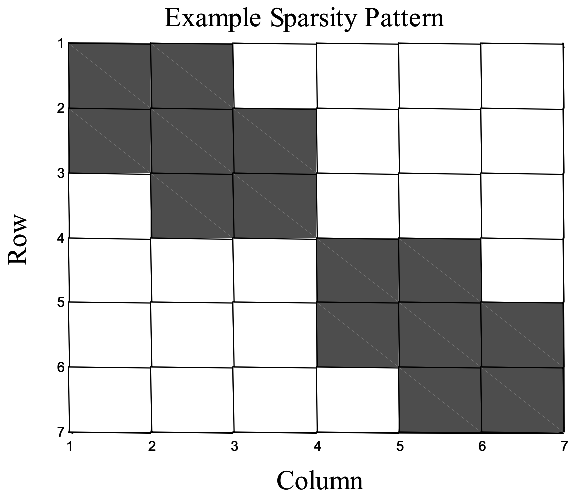

Figure 4.15:Example sparsity pattern.

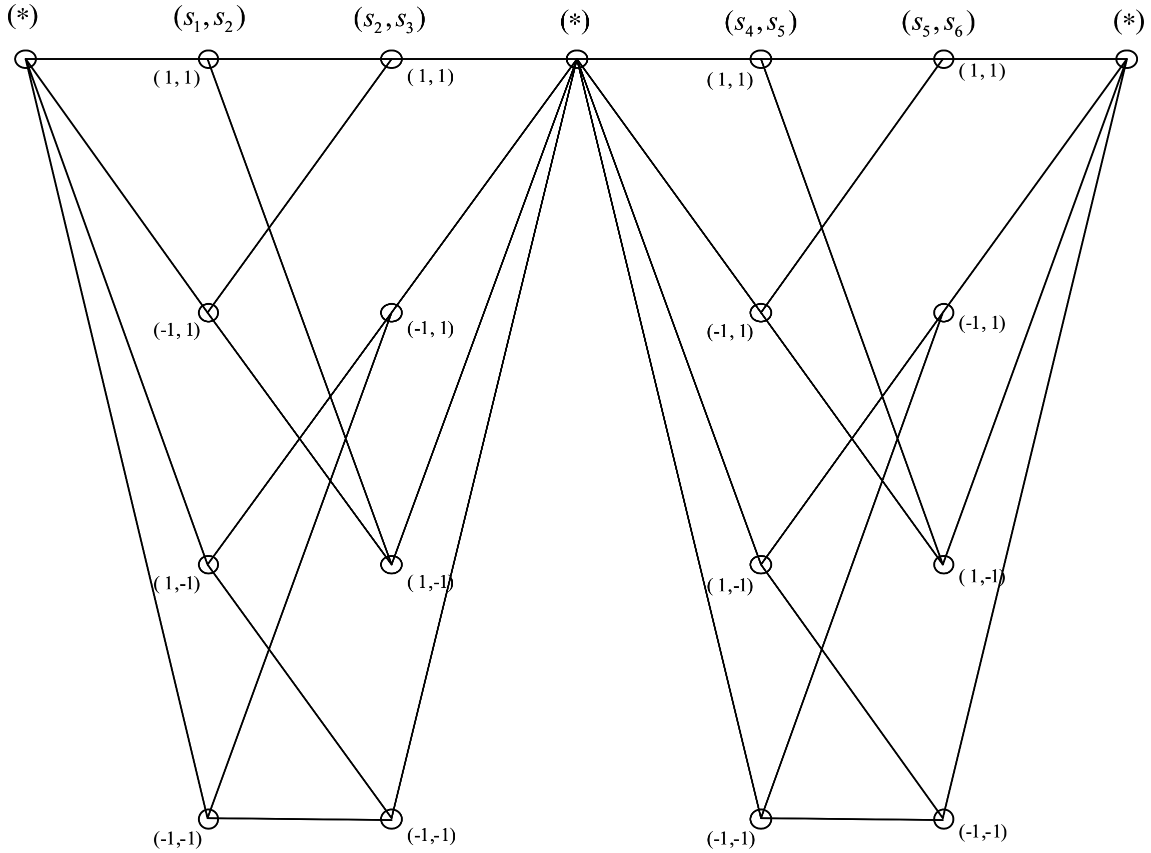

Figure 4.16: Trellis corresponding to the sparsity pattern in Figure 4.15.

4.3.7 Summary of ITB-DDFSE (Assumed Known Channel)

1. Allocate an $|\mathcal{A}|^{\mu_{\max}} \times 1$ array of cumulative partial path metrics, $\xi^{(i)}[d]$. Initialize $\xi^{(i)}[d=1]=0,\ \forall i=0,1,\ldots,|\mathcal{A}|^{\mu_{\max}}-1$.

2. Allocate a $D_u \times 1$ list of $|\mathcal{A}|^{\mu[d]} \times 1$ arrays. This list of arrays stores surviving transitions into the $\sigma[d+1]=j^{th}$ state at the dth stage, $i_s^{(j)}[d]$.

3. Start the iterative recursion, continue for a specified number of iterations around the trellis, N_{round}.

   For each stage, $d=1,2,\ldots,D_u,1,2,\ldots$

   > For each stage, $j=0,1,\ldots,|\mathcal{A}|^{\mu[d+1]}-1$
   >
   > > Find the survivor (4.16), (4.17), (4.18).
   > >
   > > Update the list of cumulative partial path metrics, $\xi^{(i)}[k]$, (4.19)

4. Terminate the trellis.

5. Trace back: after the last stage of the trellis, reconstruct the least cost path from the survivor list, $i_s^{(j)}[d]$.

6. Translate a state sequence into a symbol sequence.

4.4 Choosing a Sparsity Pattern

The choice of sparsity pattern for a particular trellis-oriented factorization is a difficult one. Generally, the linear filter in SRSJD will not yield an **H** that is strictly banded. That is, the elements of **H** outside the main super-diagonal are not strictly zero. We would like to find a joint detection trellis of reasonable complexity that provides an adequate approximation to the joint ML receiver in some sense. This requires the following: on each row of **H**, we must define a threshold, γ_t, for forming a sparsity pattern. That is the $(i,j)^{th}$ element of the sparsity pattern, **P**, is defined as

$$p_{i,j} = \begin{cases} 1 & |h_{i,j}| > \gamma_t \\ 0 & else \end{cases} \Bigg] \tag{4.20}$$

Finding the best way to choose this threshold is difficult. Obviously, we would like to choose a threshold that yields a symbol error probability that meets some specification. However, since the probability of error expressions for ITB-DDFSE are not available, this is a difficult specification to meet. Hence, we would like to define a sub-optimal criterion. Define the Desired Signal Energy to Interference Ratio (DEIR) for the d^{th} beam-former output, $y[d]$, to be

$$DEIR_d = \frac{E\left[\left|h[d,d]s[d]\right|^2\right]}{E\left[\left|\sum_{u \in \bar{\mathcal{U}}_e[d]} h[d,u]s[u]\right|^2\right]} = \frac{\left|h[d,d]\right|^2}{\sum_{u \in \bar{\mathcal{U}}_e[d]} \left|h[d,u]\right|^2} \tag{4.21}$$

The DEIR$_d$ is just the ratio of the dominant interferers in the d^{th} beam-former output due to signals in the enumeration set to the energy of the signals outside that set. One possible way to choose a sparsity pattern is to choose the enumeration set at each stage by finding the smallest $\mathcal{U}_e[d]$ which meets some specified DEIR$_d$. Henceforth, we will call this the *DEIR-Rule*. For instance, a 6-dB DEIR-Rule forms a sparsity pattern that assures DEIR$_d$ > 6 dB for each *d*.

4.5 Complexity

The complexity of the reduced search ML-like algorithm is dependent on several parameters. Among these are:

- The size of the reduced-search trellis $\mu[d]$.
- The number of iterations around the cylindrical-trellis, N_{round}

The required size of the reduced state trellis depends on the signal environment at hand. As an example, we will consider the case where users are equally spaced in AOA around a circular array. This environment yields uniform-depth joint detection trellises such as those of Figure 4.6. We further assume that hardware cost of multiplies far exceeds the cost of data accesses or additions; so it is meaningful to speak in terms of multiplies/sec. Finally, we will assume every possible combination of $h[d,u]$, and $s[u]$ is pre-computed and stored in a $D_u \times D_u \times |\mathcal{A}|$ array. These computations are negligible compared to the cost of ITB-DDFSE. The remaining multiplies are associated with the cost of each squared-error computation $|e^{(i,j)}[d]|^2$, requiring two real multiplies. There are hence, $2|\mathcal{A}|^{(\mu+1)}$ multiplies/stage and there are $N_{round}D_u$ stages/channel/symbol. Hence , ITB-DDFSE requires $2N_{round}D_u|\mathcal{A}|^{(\mu+1)}$ multiplies/channel/symbol to demodulate a single transmitted symbol from each co-channel interfering signal. Compare this with the $2M|\mathcal{A}|^{D_u}$ multiplies/channel/symbol required by the brute-force JML search.

Now consider the special case of QPSK signals impinging on an M=8 element circular array. The required number of iterations around the trellis is not yet well understood, but we have had success in a later chapter with N_{round}= 2. A judicious choice of μ depends on the number of users and the number of elements. Table 6.3.1 1 lists a recommended choice of μ for different D_u. Here, an M= 8 element calibrated circular array with a radius of R_a =λ/4 is considered. For each case, the choice of μ is chosen by looking at the number of relatively large values in the matrix **H**. Also, in this table the required multiplies/second/channel is given for each D_{u} and recommended μ. Also, for 24ksymbols/sec signals (IS-136's symbol rate) the multiplies/sec are given. Finally, the factor savings over a brute-force ML search is given. For large D_{u} the reduced search provides a computational savings factor in excess of 10^3.

Table 4.1: Computational Complexity of the reduced search vs. number of equal AOA-spaced, equal power QPSK users impinging on an M= 8 element array. Mults/sec assume IS-136 data rates.

D_u	recommended μ	Mults/ symbol/ channel	Mults/ sec	Factor Savings
9	2	2E+03	6E+07	2E+03
10	2	3E+03	6E+07	7E+03
11	2	3E+04	7E+08	2E+04
12	4	5E+04	1E+09	5E+03
13	4	5E+04	1E+09	2E+04
14	4	6E+04	1E+09	7E+04
15	4	6E+04	1E+09	3E+05
16	4	7E+04	2E+09	1E+06
17	4	7E+04	2E+09	4E+06
18	6	1E+06	3E+10	9E+05

4.6 CONCLUSION

In this section, we have outlined the application of SRSJD to overloaded array processing. The proposed algorithm here relies on two steps: reduced span linear filtering and ITB-DDFSE. Reduced span filtering attempts to find a MIMO beam-former that reduces the complexity of later non-linear processing stages while still preserving the JML criterion. We have proposed a particular method of reduced-span filtering obtained through a spectral square-root factorization of (assumed) known channel parameters. However, we have recognized that other factorizations exist which may yield better results. Improved reduced span linear filtering is an area of future research. Finally, we have recognized that, although a MIMO beam-former can group the energy on each row of **H**, it cannot force elements off of the main diagonal of **H** to zero. Hence, strict application of the ITB-VA is not appropriate. For these reasons, we have proposed the ITB-DDFSE algorithm, which accounts for near zero (but not necessarily zero) off diagonal elements of **H** with decision-feedback.

Chapter 5: TEMPORALLY REDUCED SEARCH JOINT DETECTION (TRSJD)

In the last chapter, we have shown that overloaded array processing is possible for symbol synchronous signals. In this chapter, we show that a similar approach can be used to jointly detect asynchronous signals with a single antenna. Jointly detecting asynchronous, narrow-band, linear modulated signals with zero partial response pulse-shapes is a well-studied problem in the literature [207]. However, joint detection of linear modulated signals with partial-response signaling has been neglected. In this chapter, we will explain how signals employing a square-root raised cosine pulse shape can be jointly detected with an achievable complexity.

5.1 INTRODUCTION

As shown in the previous chapter, SRSJD can greatly reduce the number of computations required for joint detection by exploiting the spatial distance properties. The algorithm is limited to environments where users are symbol-synchronous. Although course frame synchronization is a common feature in modern TDMA cellular systems, synchronizing users on the symbol level imposes difficult and expensive system requirements. We would like to extend the SRSJD algorithm to the asynchronous case. Toward this end, this section investigates the possibility of jointly detecting two asynchronous Root-Raised Cosine Pulse shaped QPSK signals impinging on a single antenna.

One difficulty with jointly detecting asynchronous users is that the k^{th} symbol transmitted from any given user is interfered with by both the k^{th} and the $(k+1)^{\text{th}}$ symbol from other users. This difficulty can be mitigated with a joint sequence estimator. The development of joint detection trellises require two major steps [8]:

- Treat the sequence of all users as one long sequence to be estimated.
- Expand the definition of the channel state to incorporate all users.

First, we will briefly overview Verdu's approach to joint detection. Next, we will describe a new joint detection algorithm that reduces the complexity of Verdu's processor in two ways:

- It requires fewer A/D converters on the front end.
- It attempts to reduce the required states of a sequence detector.

5.2 Verdu's Joint-ML sequence detector

Consider D_u asynchronous cochannel signals impinging on a single antenna. The continuous time received signal can be written as

$$r(t)=\sum_{d=1}^{D_u}\sum_{l=-M_{gq}}^{N_{fr}-M_{gq}} A_d s^{(d)}[l]p(t-lT_s-\tau_d)+z(t) \tag{5.1}$$

where $s^{(d)}[l]$, A_d, $p_d(t)$ and τ_d are the symbol sequence, complex amplitude, pulse, and symbol sequence of the d^{th} user. Verdu showed that if the channel parameters are assumed known, the joint estimate of all the user's symbol sequences which maximizes the likelihood of a continuous time received signal, $r(t)$, is

$$\begin{aligned}\{\hat{s}^{(d)}[k]\} &= \underset{\{s^{(d)}[k]\}}{\arg\min}\int_{-\infty}^{\infty} |r(t)-\hat{r}(t)|^2\, dt \\ \hat{r}(t) &= \sum_{d=1}^{D_u}\sum_{l} A_d s^{(d)}[l]p_d(t-lT_s-\tau_d)\end{aligned} \tag{5.2}$$

Verdu showed that a bank of D_u matched filters provides a sequence of sufficient statistics, $r^{(d)}[k]$, for joint ML detection. That is a ML criterion can be written in terms of $r^{(d)}[k]$ that is equivalent to Equation (5.1). Expanding the ML criterion for $r^{(d)}[k]$ results in a joint detection algorithm that exploits the temporal dependence between interfering signals.

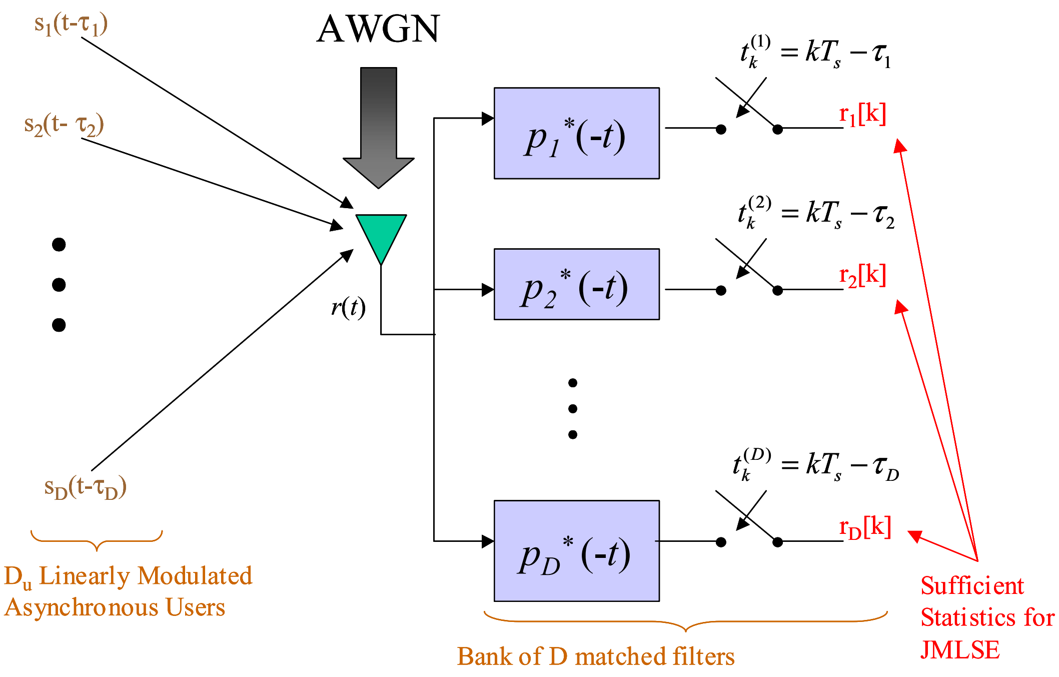
$s_1(t-\tau_1)$
$s_2(t-\tau_2)$
$s_D(t-\tau_D)$
AWGN
$r(t)$
$p_1^*(-t)$
$p_2^*(-t)$
$p_D^*(-t)$
$t_k^{(1)} = kT_s - \tau_1$
$t_k^{(2)} = kT_s - \tau_2$
$t_k^{(D)} = kT_s - \tau_D$
$r_1[k]$
$r_2[k]$
$r_D[k]$
D_u Linearly Modulated Asynchronous Users
Bank of D matched filters
Sufficient Statistics for JMLSE

Figure 5.1: Front-end processor for Verdu's Maximum Likelihood Joint detector.

5.3 TRSJD

Although Verdu's ML Joint detector will obtain the optimal performance in Additive White Gaussian Noise, it has two drawbacks:

- It requires a separate A/D synchronized to each interfering signal.
- Its trellis does not try to mitigate the complexity imposed by non-zero partial response signaling (.e.g. Root-Raised Cosine Signaling)

To simplify the matched filter for the special purpose of detecting narrow-band signals, we propose to replace the bank of matched filters with a single fractionally spaced sampling A/D. This is illustrated in Figure 5.2. Let Q denote the oversampling factor.

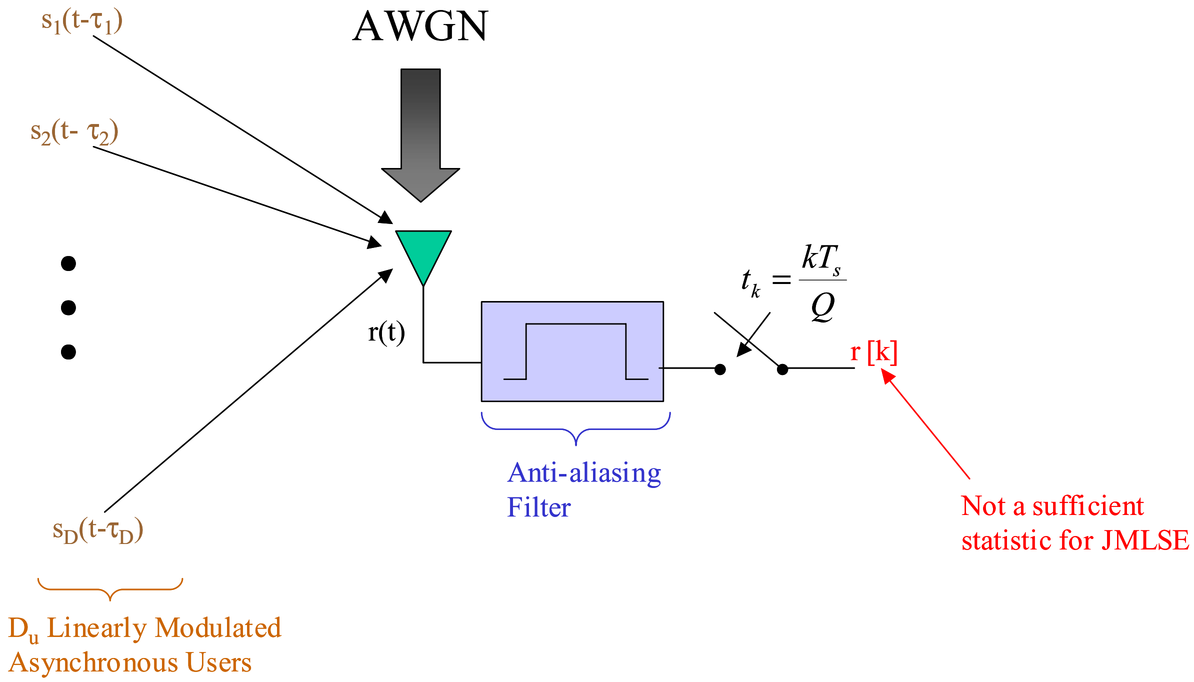

Figure 5.2: The front-end processor for TRSJD.

An discrete equivalent channel model, $g^{(d)}[k]$ can be developed for $r[k]$:

$$\begin{aligned} r[k] &= \sum_{d=1}^{D_u} \sum_{l=-M_{gq}}^{N_{fr}+M_{gq}} s^{(d)}[l] g^{(d)}[k-lQ] + z[k] \\ g^{(d)}[k] &= A_d p(t) * f(t)\Big|_{t_k=\frac{kT_s}{Q}-\tau_d} \end{aligned} \tag{5.3}$$

where the length of each $g^{(d)}[k]$ is $Q(2M_{gq}+1)$ and N_{fr} denotes the frame length. The function $f(t)$ is the front-end anti-aliasing filter and for IS-136 $p(t)$ is a Root-Raised Cosine (RRCOS) pulse with r= 0.35 rolloff. If we group the outputs of the received signal $r[k]$ into Q-tuplets, we can form a poly-phase version of the channel model:

$$r[pQ+q] = \sum_{d=1}^{D_u} \sum_{\Delta p=-M_{gq}}^{M_{gq}} s^{(d)}[p-\Delta p] g_q^{(d)}[\Delta p] + z[pQ+q] \tag{5.4}$$

The quantity $g_q^{(d)}[\Delta p]$ is the qth poly-phase filter for the dth user given by:

$$g_q^{(d)}[\Delta p] = g^{(d)}[q+\Delta pQ] \tag{5.5}$$

This model is illustrated in Figure 5.3.

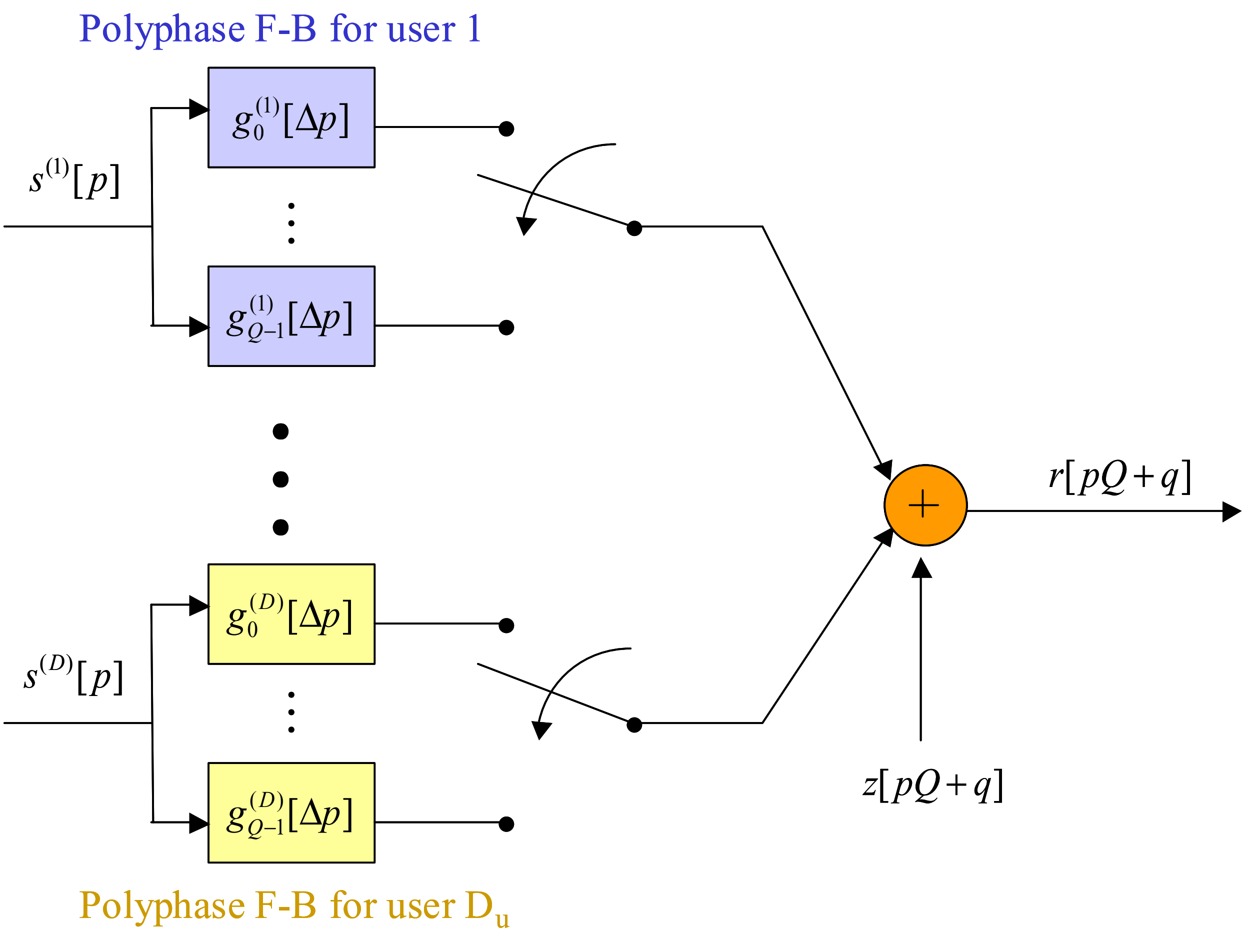

Figure 5.3: Polyphase filter model of multiple-access channel.

Using this poly-phase model, a joint detection algorithm can be devised by considering the symbols from all interferers as one symbol-stream. Towards this end, we form a $D_u \times 1$ vector,

$$\underline{s}[l] = \left[s^{(1)}[l] \quad s^{(2)}[l] \quad \cdots \quad s^{(D_u)}[l] \right]^T$$

by stacking the l^{th} symbols transmitted by each user. We form a $D_u N_{fr} \times 1$ vector , $\underline{s}$, by stacking all user's symbols first in order of user and secondly in order of transmission. Then, we stack all the received symbols into one vector, $\underline{r}$. With these definitions in hand, we can write a block-Toeplitz relation between transmitted and received symbols.

$$\underbrace{\begin{bmatrix} r[0] \\ r[1] \\ \vdots \\ r[QN_{pkt}-1] \end{bmatrix}}_{\substack{\underline{r} \\ QN_{pkt}\times 1}} = \underbrace{\begin{bmatrix} \boxed{\underset{Q\times D_u L_{gq}}{\mathbf{G}_0}} & & 0 \\ \underset{QxD_u}{\mathbf{0}}\ \boxed{\underset{Q\times D_u L_{gq}}{\mathbf{G}_0}} & & \\ & \ddots & \\ 0 & & \boxed{\underset{Q\times D_u L_{gq}}{\mathbf{G}_0}} \end{bmatrix}}_{\substack{\mathbf{G} \\ QN_{pkt}\times D_u N_{fr}}} \underbrace{\begin{bmatrix} \vdots \\ \cdots\cdots \\ \underline{s}[0] \\ \cdots\cdots \\ \underline{s}[1] \\ \cdots\cdots \\ \vdots \end{bmatrix}}_{\substack{\underline{s} \\ D_u N_{fr}\times 1}} + \underbrace{\underset{QN_{pkt}\times 1}{\underline{z}}}_{\text{AWGN}} \tag{5.6}$$

The Toeplitz sub-block, **G₀**, is formed by interleaving the conjugate-flipped poly-phase filters from all the users. The joint estimate that maximizes the likelihood of the received sampled signal is

$$\hat{\underline{s}} = \arg\min_{\underline{s}} \left\| \underline{r} - \mathbf{G}\underline{s} \right\|^2 \tag{5.7}$$

because of the ISI introduced by the Root-Raised Cosine pulse shape, a direct minimization of Equation (5.7) results in an inefficient trellis. To simplify the trellis we perform a matrix factorization similar to the one performed by SRSJD. That is, we will form a $D_uN_{fr}XD_uN_f$ matrix **H** and a vector $\underline{y}$, which satisfy the following conditions:

$$\begin{aligned} \mathbf{G}^H\,\mathbf{G} &= \mathbf{H}^H\,\mathbf{H}, \\ \mathbf{W} &= \left(\mathbf{H}^H\right)^{\dagger}\mathbf{G}^H \\ \underset{D_u N_{fr} x1}{\underline{y}} &= \mathbf{W} \underset{QN_{pkt}x1}{\underline{r}} \end{aligned} \tag{5.8}$$

If these conditions hold, then $\underline{y}$ is a sufficient statistic. That is,

$$\begin{aligned} \hat{\underline{s}} &= \arg\min_{\underline{s}} \| \underline{r} - \mathbf{G}\underline{s} \|^2 \\ &= \arg\min_{\underline{s}} \left\| \underline{y} - \mathbf{H}\underline{s} \right\|^2 \end{aligned} \tag{5.9}$$

It advantageous to choose an **H** that minimizes the required ML-trellis as much as possible.

5.3.1 Example

Consider D_u= 2 QPSK users w/ r= .35 RRCOS pulse shaping, $\{\tau_d\}$= $\{-.2T_s, .15T_s\}$, Q= 2, User 2's power is 3dB down from user 1. Each user transmits N_{pkt}=10 symbols. A checkerboard plot of the spectral factor, $\mathbf{H}= (\mathbf{G}^H\mathbf{G})^{(1/2)}$ is illustrated in Figure 5.4.

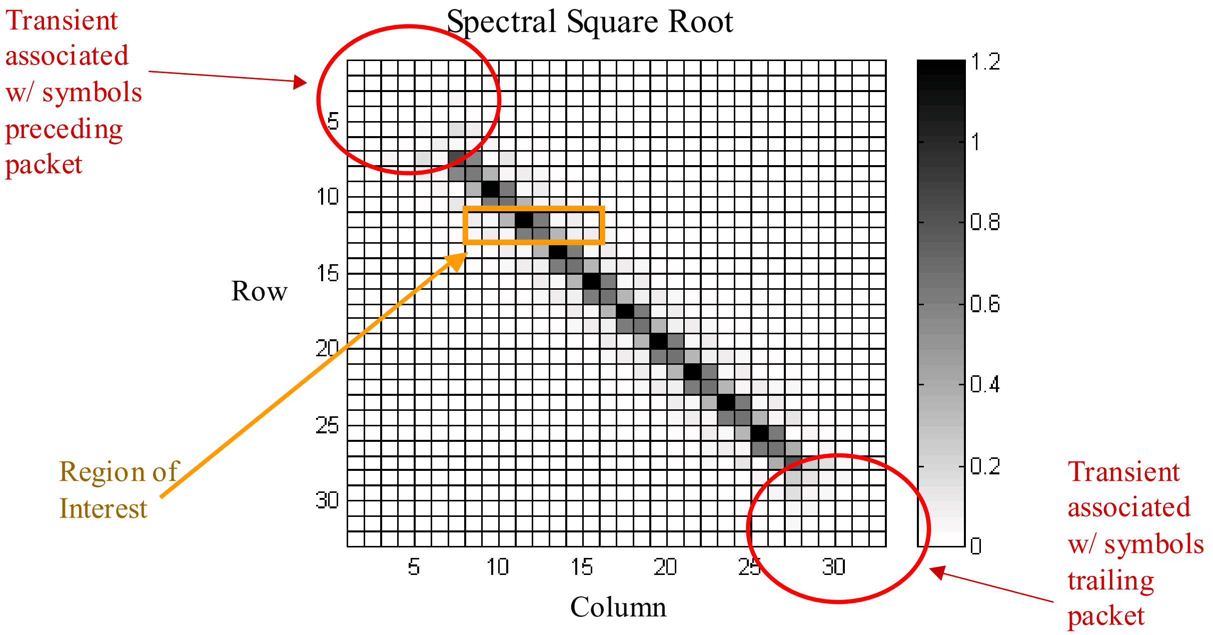

Figure 5.4: Example spectral factorization for a 10-symbol block. The number of Non-transient rows in the factorization is D_uN_{pkt}= 20. The remaining rows D_uM_{gq}= 6 top/bottom rows are transients. Judicious application of the trellis will operate on rows containing non-negligible entries.

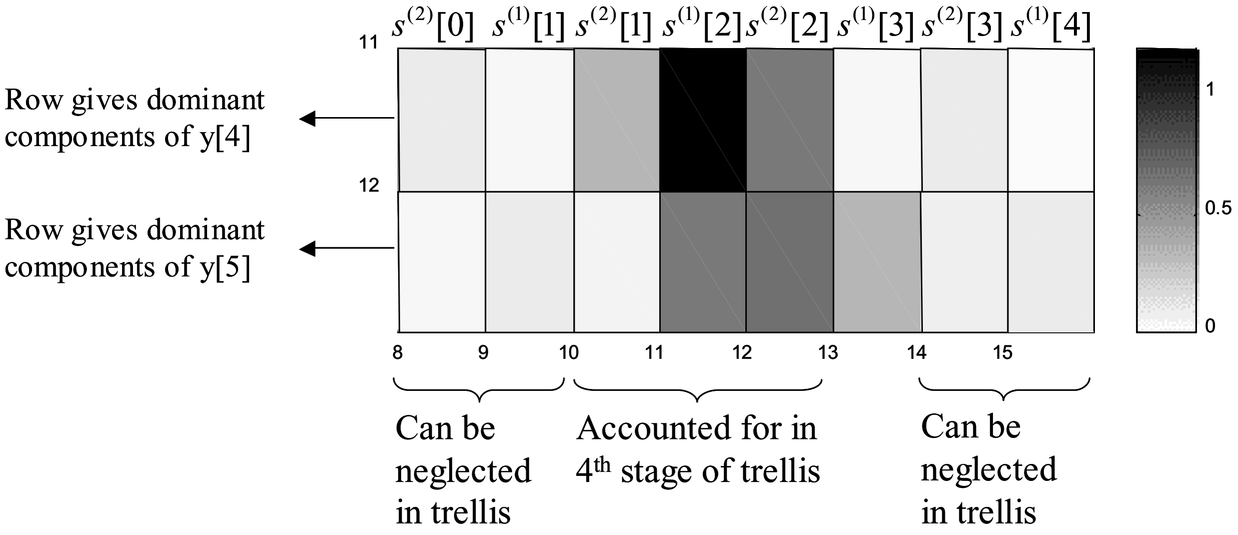

Figure 5.5: Zoom in on sub-block of the matrix illustrated in Figure 5.4.

To illustrate how the structure of **H** can be applied to a trellis, we will zoom in on a particular sub-block. This is illustrated in Figure 5.5. The 11th row of **H** yields the dominant components of $y[4]$ and the 12th row of **H** yields the dominant signal components of $y[5]$. Clearly, $y[4]$ consists of a large component of the symbol $s^{(1)}[2]$ which is interfered with by temporally adjacent symbols $s^{(2)}[1]$ and $s^{(2)}[2]$. For this reason, the trellis for the 4th stage should account for all possible combinations of symbols: $s^{(1)}[2]$, $s^{(2)}[1]$ and $s^{(2)}[2]$. Such a trellis would have a complexity of $4^{(\mu+1)} = 64$ transitions per stage.

5.4 Conclusion

In this section we have outlined an approach for the joint detection of signals with root-raised-cosine signaling. If only one co-channel signal were present, a simple matched filter would mitigate all the ISI induced by a pulse shape. However, since there are multiple signals present in the channel, an A/D matched to any particular user will contain several symbols of ISI for other signals. We have avoided the use of multiple A/D's synchronized to the delays of different users. Also, we have mitigated the multi-user ISI problem with a spectral square root factorization of the channel cross-correlation matrix. Although this factorization is costly for long block lengths, it demonstrates that trellis based joint detection of asynchronous partial-response signals can be made possible with linear pre-processing.

Chapter 6: SIMULATION RESULTS

This section compares simulation results for the algorithms discussed in the previous chapters:

1. Optimum-SINR Linear Beamformer.
2. Joint Maximum-Likelihood Detector.
3. Spatially Reduced Search Joint Detection.
4. Temporally-Reduced Search Joint Detection.

If not specified otherwise, SNR is taken as the ratio of signal variance to noise variance experienced at the receiver:

$$SNR = 10\log_{10}\frac{\sigma_s^2}{\sigma_z^2} \tag{6.1}$$

6.1 JOINT MAXIMUM-LIKELIHOOD RECEIVER

Simulation of the ML receiver is prohibitively expensive for a large number of users. However, it is instructive to investigate its performance for a small range of signals and SNRs. Consider the M= 5 element circular array of radius Ra=.2λ in Figure 4.2. Consider D_u > 5 equal power symbol and phase synchronous QPSK signals impinging on the array from equally spaced angles of arrival. Figure 6.1 plots the symbol error probability verse the number of equal-AOA spaced QPSK signals impinging on the array. For JML detection with a circular array, we have observed large degradation in performance when signals impinge on the array from opposite angles of arrival. This is not an anomaly of the algorithm but a fundamental limit of the circular array. It is expected that signals impinging on the array from opposite angles of arrival result in more closely spaced signal points in array signal space. This is not an idiosyncrasy of the algorithm but rather a fundamental limitation of the chosen array geometry.

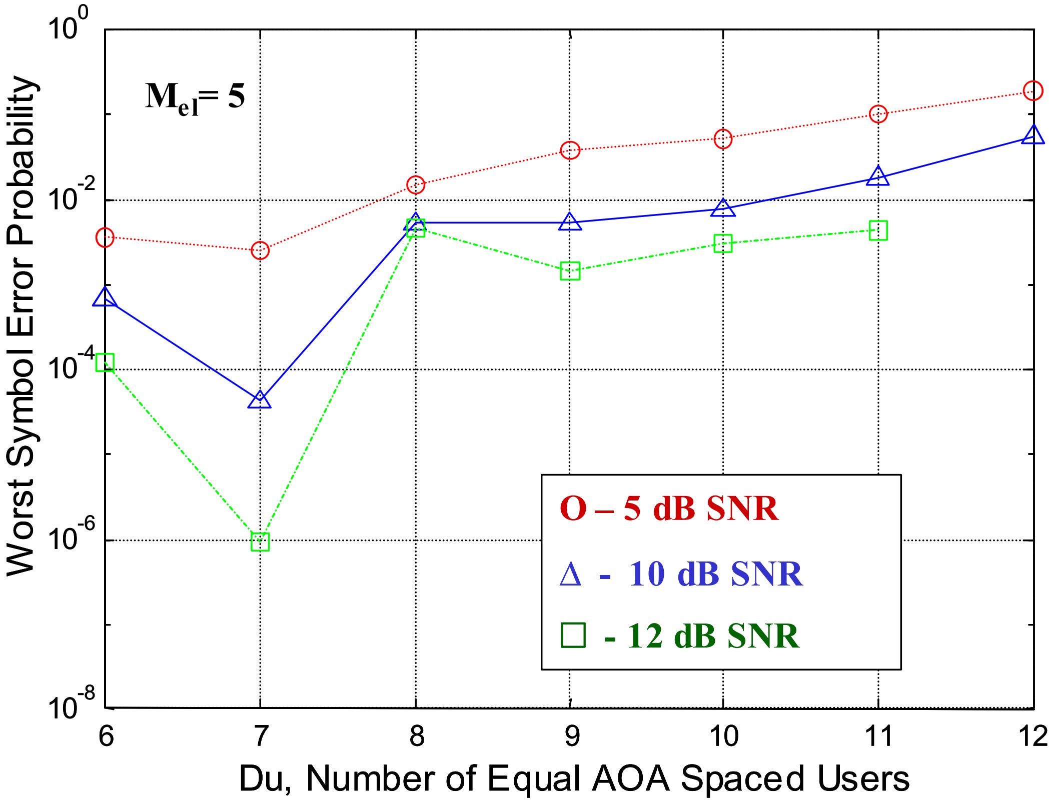

Figure 6.1: Signal capacity simulation of brute-force maximum-likelihood search for M =5 elements.

6.2 SRSJD

Simulation results have been compiled for SRSJD. Firstly, we will compare the reduced search ML-like approach to the brute-force ML approach for equally spaced users impinging on an M= 5 element array. We will find that for moderate SNRs, the reduced state approach achieves a huge reduction in complexity with light performance penalty. Finally, we will present simulation results for an eight-element array.

6.2.1 Symmetric Interference Environment

Figure 6.2 illustrates capacity curve for scenarios similar to Example 4.2.1. We consider a number , D_u, of equal power QPSK signals impinging on an array with equally spaced AOAs from 0^{o} to 360^{o}. Again, the signals are assumed to be symbol and phase synchronous. The array is a 5-element circular with radius R_a= .2λ. Three algorithms are compared: the optimum SINR linear beam-former, a brute-force ML search and the reduced search ML-like algorithm. The SNR per signal per element, as defined in Equation (5.1) is kept at 10dB. Since the users are uniformly distributed in AOA, the reduced search trellis for each case is tail-biting and uniform in complexity. We will describe the complexity of the reduced search trellis in terms of a parameter μ: the trellis size will be $4^{\mu+1}$ on each face of the cylindrical trellis. For each case, the parameter μ is set by looking at the number of non-zero elements on each row of **H**. This will be investigated in more detail later. Full decision feedback was used. Finally, N_{round}= 2 ITB-DDFSE iterations around the TBT were used. When the ITB-DDFSE terminates, symbol estimates are pulled out of the least-cost path by *tracing-back* through survivors in the trellis. In general, the ITB-DDFSE often yields a path through the tail-biting trellis that is not closed. Symbol estimates are taken from the least cost path in reversed order. If the ITB-DDFSE terminates on the d_{term}^{th} stage, then the ITB-DDFSE uses the symbol exiting the state $\sigma[d_{term}]$.

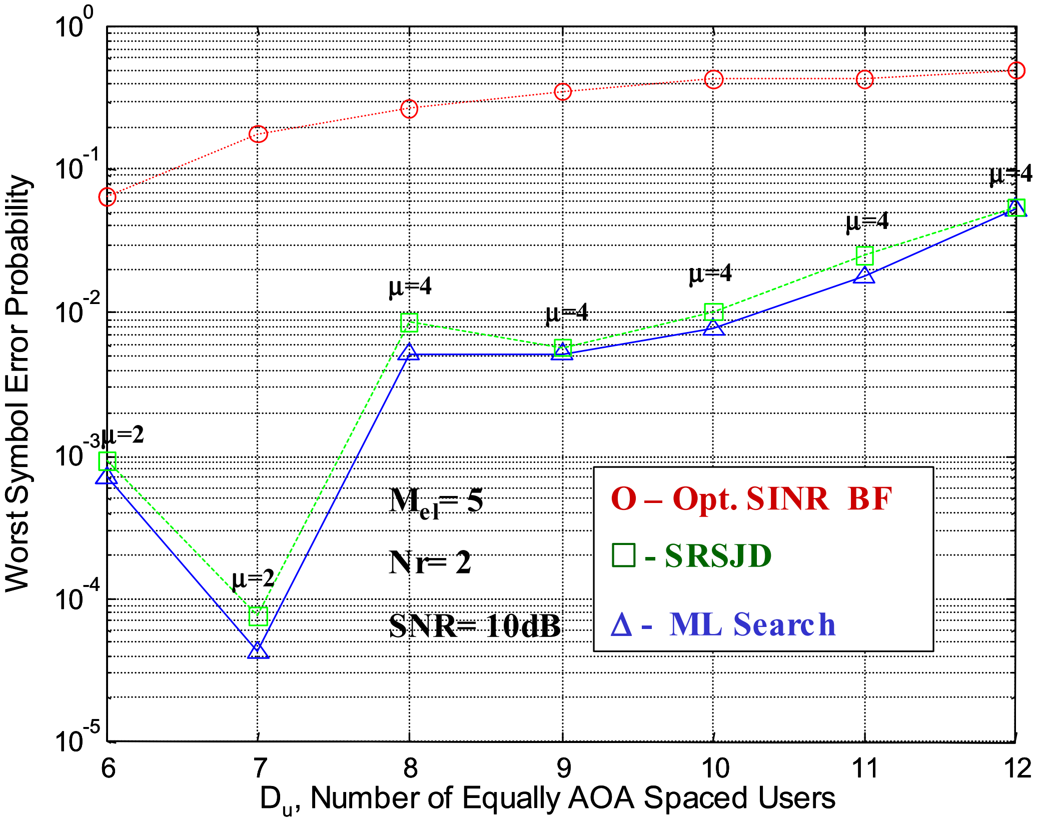

Figure 6.2: Signal Capacity comparison of three receivers for a synchronous QPSK users, equally spaced in AOA. $M_{el} = 5$ element array.

The SNR per signal per element is defined in Equation (5.1). We have plotted symbol error rate (SER) vs. D_u for two values of SNR= 5dB, 10dB. For each simulation, we have reported the worst symbol error rate experienced by any user in the simulation. The complexity of the trellis is labeled for each case. For the maximum-likelihood receiver, the simulation was run until 20 errors were experienced by the worst user. The optimal SINR beamformer is discussed in Appendix B. For the optimum SINR and SRSJD, the simulations were run until 50 errors were experienced. Figure 6.3 illustrates the simulation results for SNR= 10dB. For JML detection with a circular array, we have observed large degradation in performance when signals impinge on the array from opposite angles of arrival. This is not an anomaly of the algorithm but a fundamental limit of the circular array. In the case of the linear beam-former, symbol error rates are marginal at small overloading factors but errors approach 50% D_u= 9. In contrast, the JML detector and SRSJD can support up to D_u= 11 users acceptable error rates . Symbol errors were reasonably uniform across all users. There is marginal performance penalty for the search reduction over all signal capacities. We conclude from these simulations that the reduced-search algorithm can achieve a huge reduction in complexity with a small performance penalty.

However, at lower SNRs the picture is different. For example, Figure 6.3 illustrates the simulation results for SNR= 5dB. For a fair comparison, the same complexity and number of iterations were used as in the SNR= 10dB case. Also, decision feedback and trace-back are performed in the same manner. Here, the symbol error probability for SRSJD can be an order of magnitude worse than the true JML detector. However, in this case, even the JML receiver's performance is marginal for users $D_u > 7$.

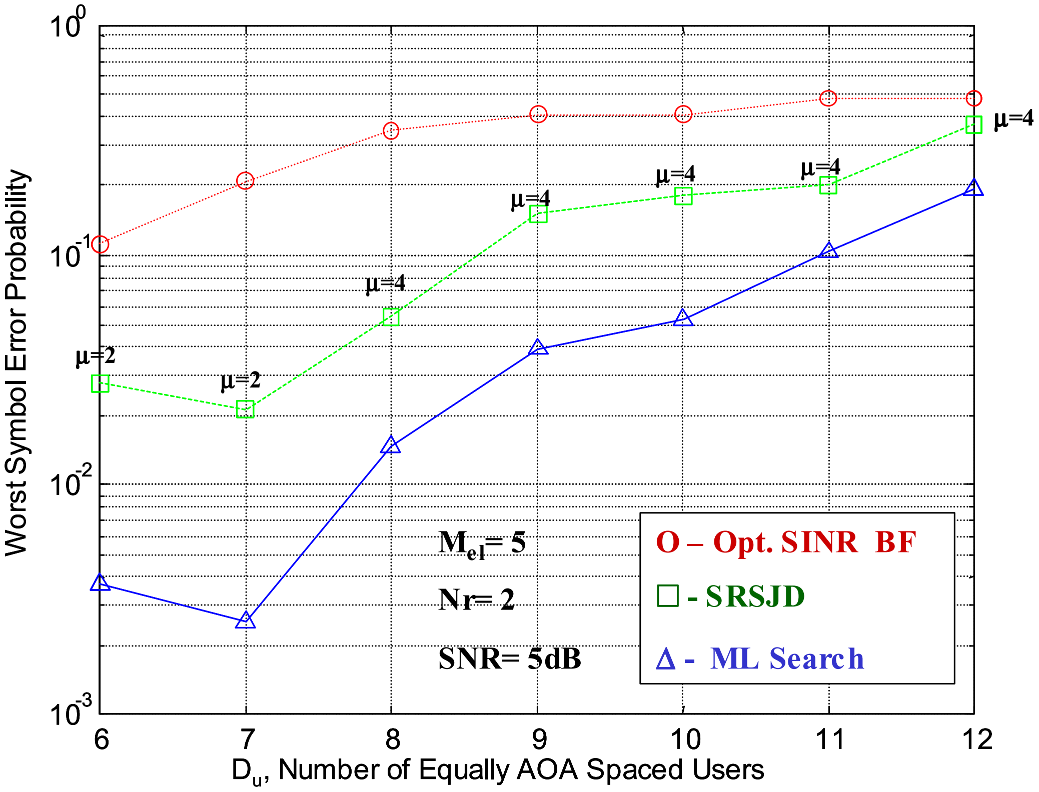

Figure 6.3: Signal Capacity comparison of three receivers for a synchronous QPSK users, equally spaced in AOA. $M_{el} = 5$ element array. The signal to background noise ratio, SNR, is less than the previous case.

Figure 6.4 compares the performance of several SRSJD processors with different complexities. In this case the SNR= 10dB. Again, μ denotes state size. In this case, there appears to be no benefit in choosing $\mu = 6$ over the range of signal capacities yielding acceptable performance. Choosing $\mu = 2$ is clearly acceptable until $D_u = 8,9$ where the interference from second nearest neighbors exceeds that which can be reliably mitigated with decision feedback.

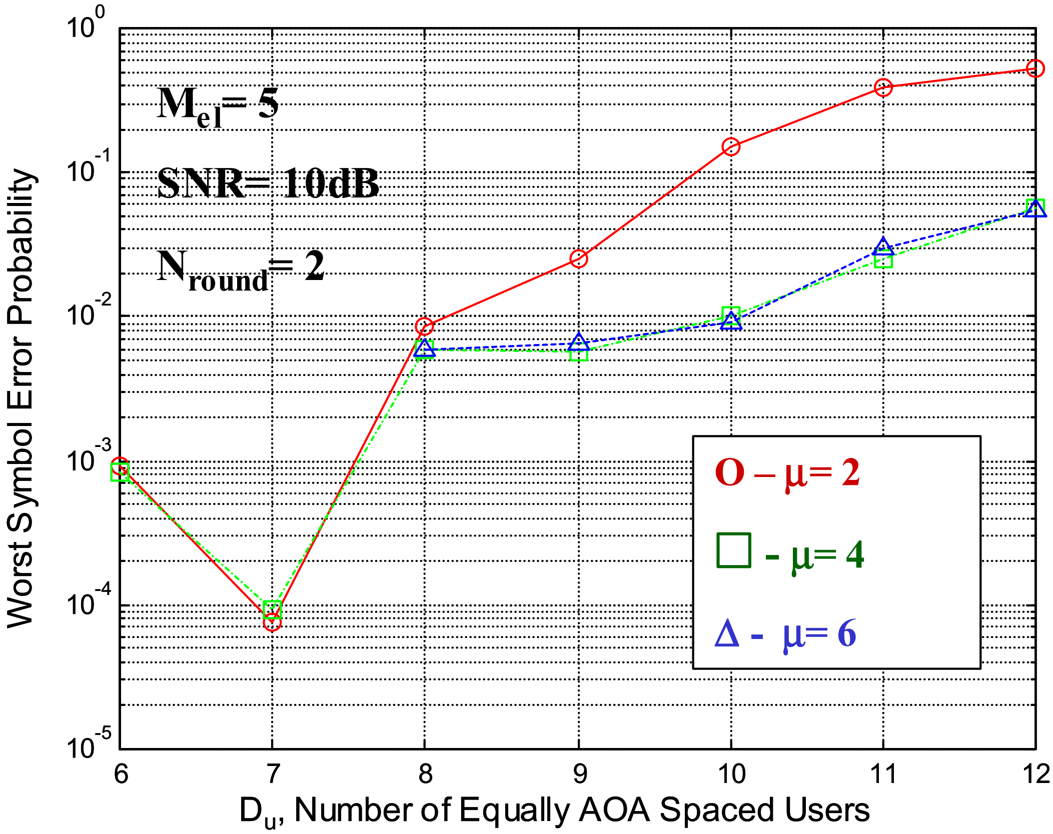

Figure 6.4: The effect of trellis size on SRSJD's symbol error rate performance. Signal Capacity curve for equally AOA spaced QPSK signals impinging on an M= 5 element antenna array.

Figure 6.5 illustrates a capacity curve for an *M*= 8 element calibrated cylindrical array with a radius of $R_a = \lambda / 4$. Again, we consider D_u equal power baud-synchronous, QPSK signals impinging on an array with equally spaced AOAs over 360^o. This time, however, the phase of each user has been assumed random and uniform over $[0,2\pi)$. Decision feedback and trace-back are performed in the same manner as the 5 element array. Several different SRSJD receivers of varying complexity are compared against the optimal SINR beam-former. For the large number of users that an eight-element array affords, simulation times for the brute-force ML search become excessively long. Hence, for this case, we have simulated only the reduced search algorithm. Again, the optimal SINR BF exhibits poor performance for even small overloading factors. However, SRSJD provides adequate SERs (with FEC) up to D_u= 19 users. Recall that with an $M = 5$ element array, we incurred difficulties with signals impinging from opposite AOAs. However, for a larger array size and randomized received phases, this problem is not apparent. The eight- element array can handle overloading factors of two but with a higher complexity than the circular array. Also note, as with the 5 element array, a gradual increase in μ with D_{u} allows a smooth cost/performance tradeoff.

Figure 6.6 illustrates the effect of feedback depth on SRSJD's performance. In this case, we have chosen a varying state size, μ, to yield a graceful degradation of performance vs. signal capacity, D_{u}. Three different feedback schemes are compared:

1. Full Feedback – all symbols not in the enumeration set (interference set) are accounted for in the trellis. $\mathcal{U}_{fb} = \mathcal{U}_I$, $|\mathcal{U}_{fb}| = D_u - (\mu + 1)$.
2. Partial Feedback – only a certain number of symbols in the interference set are feed-back. In this case, $\mathcal{U}_{fb} \subset \mathcal{U}_I$, $|\mathcal{U}_{fb}| = \left\lfloor \frac{D_u - (\mu+1)}{2} \right\rfloor$.
3. No Feedback - interference outside of the enumeration set is assumed zero (Truncation). $\mathcal{U}_{fb} = \varnothing$.

In this case the SNR= 10dB. Surprisingly Full Feedback can outperform partial feedback by two orders of magnitude. It is difficult to say; however, how much this translates to environments with lower SNRs or less symmetric interference geometries.

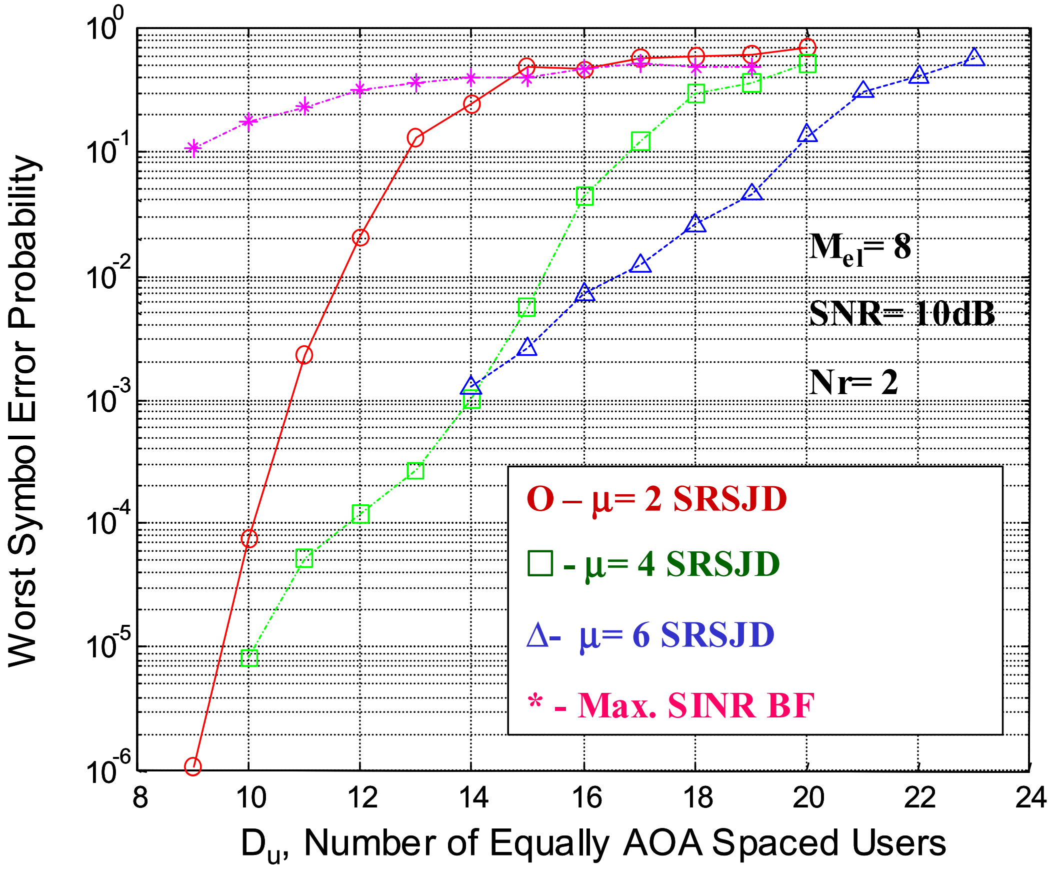

Figure 6.5: Effect of trellis state size on the performance of SRSJD. All SRSJD receivers are compared to a Maximum SINR Beam-former (see Appendix B) as a baseline. Signal Capacity curve for equal-AOA spaced QPSK signals impinging on an *M*= 8 element antenna array.

On a final note, we will investigate the effect of the number of iterations. To provide a fair comparison across a wide range of DU, we report the number of iterations around the trellis, N_{round} (as opposed to the counting the number of stags). N_{round} is a fractional

number if the number of stages traversed is not an integer multiple of D_u. It seems reasonable that, in order for all users to benefit from an extra iteration around the trellis, the following must be done: all the energy from each user's signals, must be accounted

for in the Viterbi algorithm. For this reason we report N_{round} in terms of another parameter, N_r.

$$N_{round} = N_r + \left\lfloor \frac{\mu+1}{2D_u} \right\rfloor, \; N_r = 1,2,3,\ldots \qquad (6.2)$$

There is a surprising improvement in performance on the second iteration but little benefit thereafter. Again, it is difficult to generalize these results to harsher signal

environments.

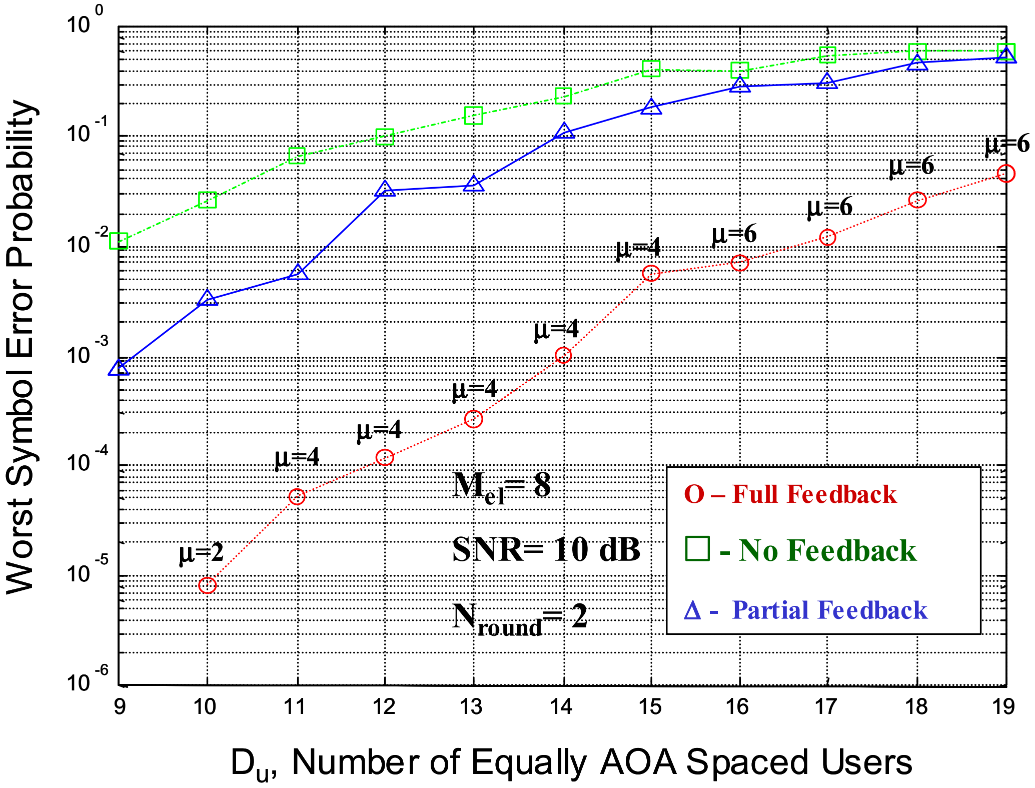

Figure 6.6: The effect of feedback on SRSJD's symbol error probability performance. Signal Capacity curve for equally AOA spaced QPSK signals impinging on an M_{el} = 8 element antenna array.

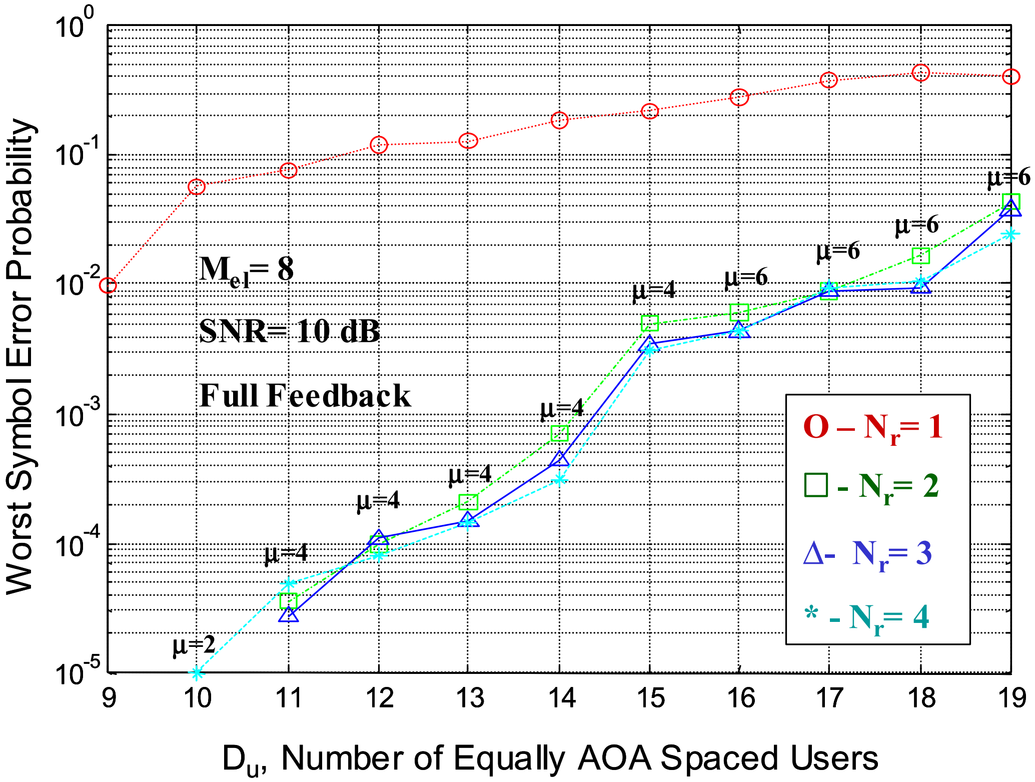

Figure 6.7: The effect of the number of iterations, N_{round}, on SRSJD's error rate performance. Signal Capacity curve for equally AOA spaced QPSK signals impinging on an M = 8 element antenna array. The chosen state size for each number of users is labeled. For a given D_u, the state size is the same for all curves (i.e. all N_r).

6.2.2 Non-Uniform Environment

In chapter 4, we found that non-uniformly spaced AOAs translate to trellis-oriented factors, **H**, that are fat on some rows and skinny on others. In this section we will consider the interference environment of example 4.3.3. We will consider two receivers: one with the sparsity pattern displayed in Figure 6.8(a), the other with the sparsity pattern displayed in Figure 6.8(b). The former was generated with a 6dB DEIR rule, and the later with a 10dB DEIR rule. This time, we will consider symbol synchronous but phase asynchronous BPSK modulated signals. Symbol error rates were evaluated with simulations; the simulations were run until 20 errors were experienced by the *best* user. The symbol error probabilities for select users are plotted vs. SNR in Figure 6.9. Here, statistics are given for three select users: user 1, user 14, and user 8. User 1 is widely spaced in AOA from other users. User 4 is moderately spaced in AOA from other users. User 14 is widely spaced in AOA from other users. The SRSJD receiver employing the 6dB DEIR rule cannot reliably separate closely spaced users such as user 8, even at high SNRs.

We can safely conclude that for the chosen receiver complexity, user 8 is interference limited. Similarly, user 1 experiences steady improvement in SER vs. SNR until about 10 dB when an error-floor effect occurs. Surprisingly, this error floor was not experienced by user 14. There are two possible reasons for this. Firstly, user one impinges on the array from a direction opposite the majority of the interference. It is possible that user 1's error floor is an artifact of a cross-array interference effect we observed with the maximum-likelihood receiver in section 6.1. Under this hypothesis, the error floor is a fundamental limitation of the circular array. No receiver could do better. A second possibility is that user 14 enjoys more receiver complexity allocated to detecting its symbols. User 14's stage is 2^5=32 times more complex than user 1. In additional simulations, none of the users benefited from less decision feedback or longer traceback depths.

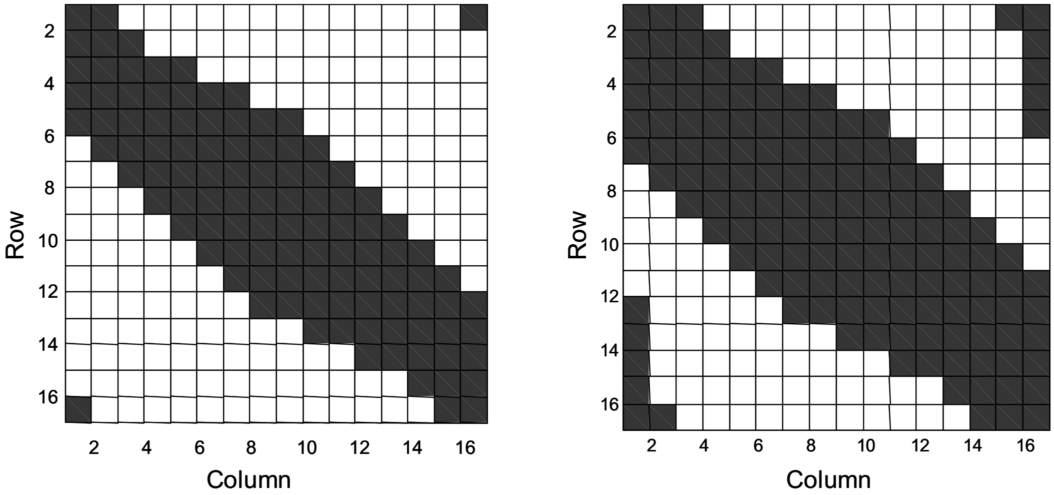

Figure 6.8: Two sparsity patterns for the asymmetric interference environment considered in Figure 4.10 of example 4.3.3 (a) (left) the sparsity pattern generated by a 6dB DEIR Rule. (b) (right) the sparsity pattern generated by a 10dB DEIR Rule.

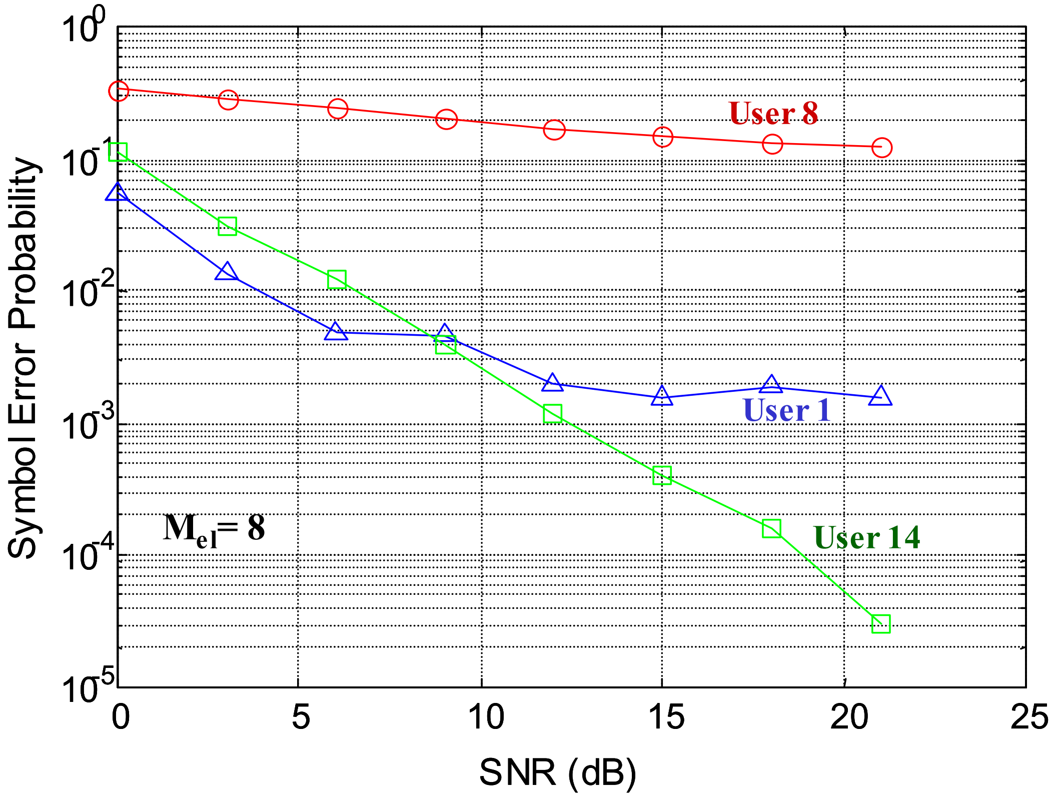

Figure 6.9: Performance of SRSJD employing the sparsity pattern of Figure 6.8(a) subject to asymmetric interference geometries.

Figure 6.11 illustrates the performance of SRSJD employing the 10dB DEIR rule (see Figure 6.8). Here all users enjoy a greater receiver complexity at higher SNRs. Again, we observe a crossover in the performance curves of user 1 and user 14 at high SNRs.

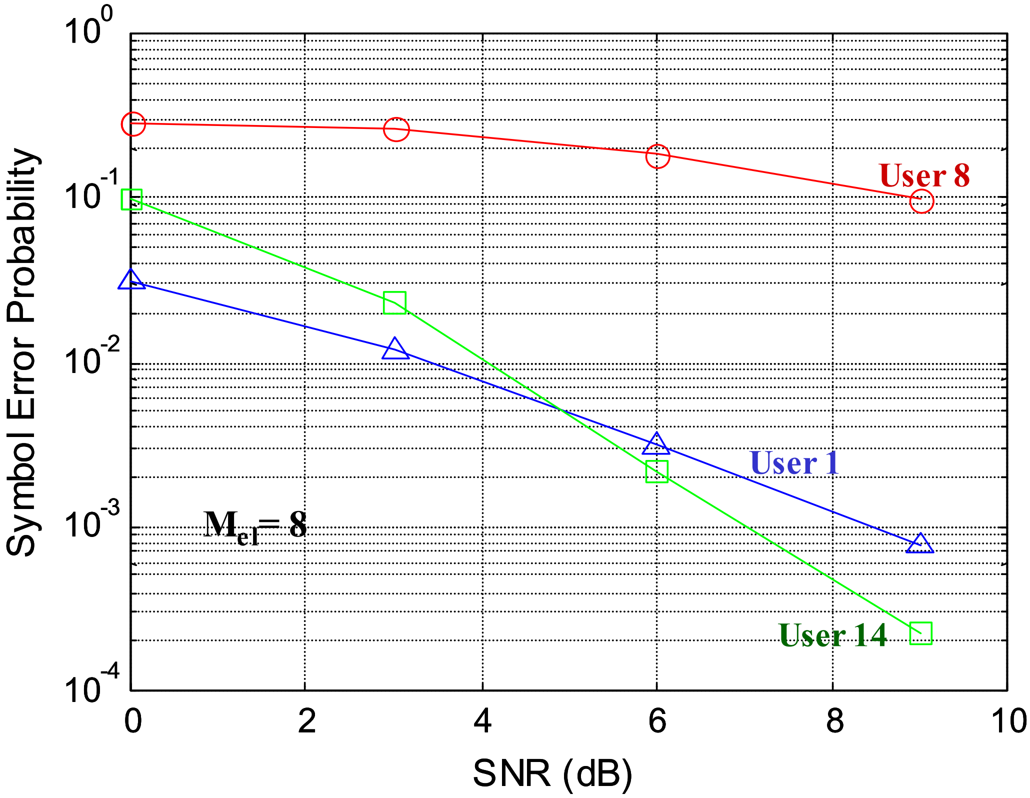

Figure 6.10: Performance of SRSJD employing the sparsity pattern of Figure 6.8 (b) subject to asymmetric interference geometries

6.3 TRSJD Results

Simulations of TRSJD applied to the above example have been run for packet sizes of N_{pkt} = 160 symbols. Again, user 1's received power is 3dB higher than user 2's and their delays relative to the receiver clock are $\tau_l = -.2T_s$ and $\tau_l = .15T_s$, respectively. Symbol error rates have been evaluated through simulation. As a baseline comparison, we have compared the performance of TRSJD to the ML receiver for a single user. Joint detection favors the user with the highest power, but, in this case, there is at most a 1 dB difference in separation. Adequate symbol error rates for voice encoding, less than 10^{-2} can be obtained for both users for $E_s^{(1)}/N_0$ in excess of 15dB. As expected, TRSJD cannot achieve the same SNR performance as a single user receiver.

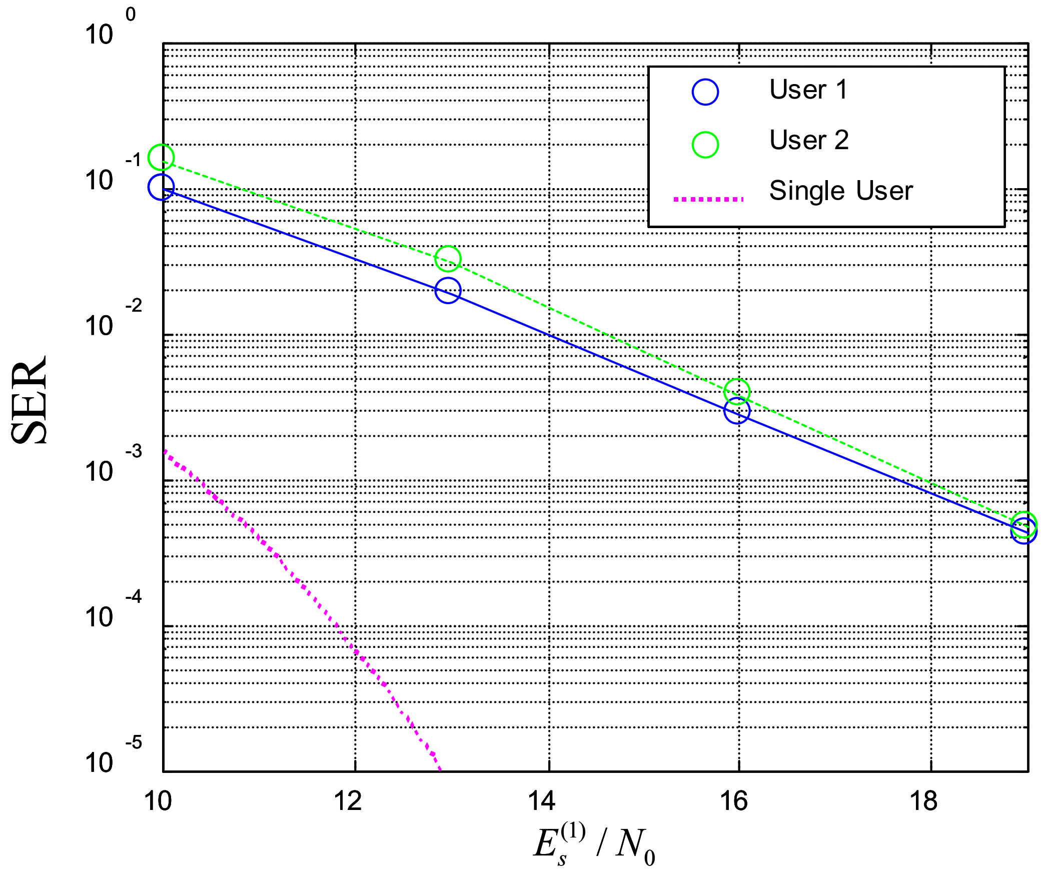

Figure 6.11: Symbol Error Rate curve for TRSJD for example 5.3.1

6.4 Conclusion

This chapter has evaluated, through simulation, the algorithms proposed in this thesis. In the case of the SRSJD, the optimal JML receiver and the optimal linear beamformer have been included as base-line comparisons. We have observed that, at moderate SNRs, SRSJD can achieve the performance of the optimal receiver but at low SNRs, SRSJD falls short. However, in both cases, SRSJD far outperforms the optimal linear receiver. This chapter demonstrates SRSJD's performance with small array sizes in order to compare its performance o the JML receiver. This chapter also demonstrates its application to larger array sizes and, in particular, investigated its performance in asymmetric interference geometries. We have found that in reasonable environments, all users enjoy an increase in SRSJD receiver complexities but this trend exhibits diminishing returns. We have found in symmetric interference geometries, that SRSJD can separate signals with overloading factors in excess of $2M$ in modest signal to noise ratios. This is an important result for systems that can reliably assign users to SDMA/FDMA channels. In the asymmetric case, we have observed particular receiver sparsity patterns can favor users with less inter-AOA spacing. Moreover, if not enough receiver complexity is allocated to a particular user, an error flooring effect occurs in a user's SNR curve when the receiver becomes interference limited. Increasing the receiver's complexity can mitigate this error floor. Finally, TRSJD has been observed to successfully separate partial response users closely spaced in power (SIR= 3dB) but slightly favors (in SER) the user with the largest power.

Chapter 7: CONCLUSION AND FUTURE WORK

This thesis has proposed a novel approach to Overloaded Array Processing: Spatially Reduced Search Joint Detection (SRSJD). In addition, we have shown that a similar approach, Temporally Reduced Search Joint Detection, can be applied to separate asynchronous signals employing partial response pulse-shapes.

In moderate signal to noise ratios (SNR), SRSJD provides was shown through simulation to well approximate the joint maximum likelihood receiver, yielding acceptable voice-grade bit error rates with overloading factors in excess of two. We have identified the most difficult operating environment for OLAP: all signals are co-channel, near equal power, with tight excess bandwidth and identical modulation types. Although SRSJD is much more expensive than conventional linear beam-forming, it succeeds where beam-forming fails, and is several order of magnitude less complex than any other known OLAP solution.

Analyzing the performance of SRSJD is difficult, because it can be applied to many different environments and array geometries. Further analysis of SRSJD will surely illuminate its strengths and weaknesses. For one, we relied on simulation for performance evaluation. No analytical performance bounds have been derived. Indeed, analysis of SRSJD's performance is difficult because it borrows from beam-forming, iterative detection, sequence estimation, and decision feedback. The author expects that these elements will interact in interesting ways. Duel-Hallen's work with DDFSE [24] should prove helpful. Moreover, all simulation results assume a known channel. Channel estimation in tandem with SRSJD is an area of future work. Talwar's ILSE [106] is a potential candidate scheme. Future analysis should account for imperfect channel estimates, synchronization errors, and other channel degradations such as Doppler carrier shifts.

SRSJD exhibits the following strengths: scalability, and generality. The algorithm is scalable with overloading factor and array size. True, if array size is fixed and overloading factor is increased, then SRSJD has exponential complexity; however, if the overloading factor is fixed the complexity increases roughly linearly with the number of users. In addition, the general formulation of SRSJD makes it applicable to many different *linear* modulation types of the same data rate. This feature makes it attractive for hybrid cellular systems such as EDGE. Future work in this area will be SRSJD's application to non-linear modulation types (e.g. GMSK or π/4-DQPSK) and joint detection of signals employing different data rates.

Nevertheless, in its still nascent stages of development, SRSJD still exhibits three major limitations. The most important limitation is the baud-synchronization requirement: signals must arrive at the receiver perfectly aligned in symbol. We have partially addressed this issue by showing that SRSJD's time-domain counterpart, TRSJD, can successfully separate asynchronous co-channel signals employing partial-response signaling. However, a Space-Time version, say STRSJD, which allows for separation of asynchronous signals is yet to be developed. Indeed such a development presents new challenges because the power of both SRSJD and TRSJD is derived from trellis-based non-linear processing. In the former, a trellis is constructed, in some sense, over space. In the later, a trellis constructed over time. Since trellises are, by definition, directed graphs, reduced-search trellises cannot be constructed over both space and time simultaneously. Hence, space-time overloaded array processing, in the general case, necessitates a new joint-detection framework analogous to, but different from, trellis based approaches. Factor-graphs and iterative joint detection may provide such a framework [216].

Secondly, it has been observed in the literature that joint multi-user detection and forward error correction well out-perform these operations in cascade. Iterative decoding employs trellised based algorithms with soft decision metrics and has been found in [214] to provide an attractive cost/complexity tradeoff. SRSJD outputs hard symbol decisions, and hence, is not compatible with iterative detection. Instead, Joint-Maximimum A Posteriori Probability (JMAP) receivers [207, 216] may provide a bridge between iterative detection and a factor-graph based joint detection framework.

Finally, SRSJD was observed to yield poor complexity reductions for some array geometries. This may be overcome with improved channel factorization algorithms. We have observed that the spectral square-root is not the only factorization that facilitates SRSJD. For one, unitary rotations of the spectral square root also preserve the maximum-likelihood criterion. This fact licenses the use of well known matrix factorization tools such as Jacobi Rotations, Givens Rotations, and Householder transforms [222].

Overall, joint detection has found a surprising application in a new research area: overloaded array processing. The algorithms proposed in this thesis provide a baseline approach for further development. In particular, this chapter outlines some possible new approaches. Although, there are likely to be many more. With so many research directions to take, Overloaded Array Processing should prove to be a fruitful area of future research.

Appendix A: PROOF OF CONSISTENCY

In this appendix, we will show that the solution, $\underline{y}$, to the following equation exists:

$$\mathbf{H}^H \underline{y} = \mathbf{A}^H \mathbf{\Phi}_{zz}^{-1} \underline{x} \tag{A.1}$$

for the chosen class of solutions:

$$\begin{aligned} \tilde{\mathbf{H}} &= \left(\mathbf{A}^H \mathbf{\Phi}_{zz}^{-1} \mathbf{A}\right)^{1/2} \\ \mathbf{H} &= \mathbf{Q}\tilde{\mathbf{H}}, \quad \mathbf{Q}^H \mathbf{Q} = \mathbf{I} \end{aligned} \tag{A.2}$$

Remark: In general, the $D_u \times D_u$ system of equations

$$\mathbf{H}^H \underline{y} = \underline{b} \tag{A.3}$$

may not have a solution if the matrix, **H**, is not full rank. Indeed, in overloaded case, **H**, is rank $M < D_u$. From the fundamental theorem of algebra [222] we know that equation (A.3) will have a solution if and only if $\underline{b} \in \mathcal{R}ange\left\{\mathbf{H}^H\right\}$. We will now show that this is the case for equation (A.1).

We will start our argument with the spectral square root. Then we will show that if the claim is true for a spectral square-root, then it is true for any unitary rotation. It is enough to show that $\mathcal{R}ange\left\{\mathbf{A}^H\right\} \subseteq \mathcal{R}ange\left\{\mathbf{H}^H\right\}$ because if this is true, for every $\underline{x}_0 \in \mathbb{C}^{M_{el}}$, there exists a $\underline{y} \in \mathbb{C}^{D_u}$ such that $\mathbf{H}^H \underline{y} = \mathbf{A}^H \underline{x}_0$. But obviously, for every $\underline{x} \in \mathbb{C}^M$, there exists a $\underline{x}_0 = \mathbf{\Phi}_{zz}^{-1} \underline{x} \in \mathbb{C}$.

We thus, complete our proof by showing that , in fact, $\mathcal{R}ange\left\{\mathbf{A}^H\right\} = \mathcal{R}ange\left\{\mathbf{H}^H\right\}$ by construction. Toward this end, let us define the Hermitian symmetric matrix and its eigen-decomposition.

$$\mathbf{R}_A = \mathbf{A}^H \mathbf{\Phi}_{zz}^{-1} \mathbf{A} = \mathbf{V} \mathbf{\Lambda} \mathbf{V}^H \tag{A.4}$$

where $\mathbf{V} = \left[\underline{v}_1 \quad \underline{v}_2 \quad \cdots \quad \underline{v}_M \quad \cdots \quad \underline{v}_{D_u}\right]$ is a matrix whose columns are the normalized eigen-vectors of $\mathbf{R}_A$. Also, the matrix, $\mathbf{\Lambda}$, is a diagonal matrix of the eigen-values of $\mathbf{R}_A$. Since $\mathbf{R}_A$ is Hermetian, then $\mathbf{U}$ is unitary [222]. Now, the set $\{\underline{v}_1 \quad \underline{v}_2 \quad \cdots \quad \underline{v}_M\}$ provides an orthonormal basis for $\mathcal{R}ange\{\mathbf{R}_A\}$.

Now, define the spectral square root factorization

$$\tilde{\mathbf{H}} = \left(\mathbf{R}_A\right)^{1/2} = \left(\mathbf{A}^H \mathbf{\Phi}_{zz}^{-1} \mathbf{A}\right)^{1/2} \triangleq \mathbf{U} \mathbf{\Lambda}^{1/2} \mathbf{U}^H \tag{A.5}$$

Since $\tilde{\mathbf{H}}^H$ has the same eigen-vectors as $\mathbf{R}_A$, the set $\{\underline{v}_1 \quad \underline{v}_2 \quad \cdots \quad \underline{v}_M\}$ is also an orthonormal basis for $\mathcal{R}ange\left\{\tilde{\mathbf{H}}^H\right\}$. Hence, $\mathcal{R}ange\left\{\tilde{\mathbf{H}}^H\right\} = \mathcal{R}ange\{\mathbf{R}_A\}$ since they have the same basis.

We will now show that $\mathcal{R}ange\left\{\mathbf{A}^H\right\} = \mathcal{R}ange\{\mathbf{R}_A\}$. For every $\underline{x} \in \mathcal{R}ange\{\mathbf{R}_A\}$, there exists a $\underline{x}_0$ such that $\underline{x} = \mathbf{A}^H \mathbf{\Phi}_{zz}^{-1} \mathbf{A} \underline{x}_0$. Hence, there exists a $\underline{y} = \mathbf{\Phi}_{zz}^{-1} \mathbf{A} \underline{x}_0$ such that $\underline{x} = \mathbf{A}^H \underline{y}$. Hence, we know at least $\mathcal{R}ange\left\{\mathbf{A}^H\right\} \subseteq \mathcal{R}ange\{\mathbf{R}_A\}$. Conversely, for every $\underline{y} \in \mathcal{R}ange\left\{\mathbf{A}^H\right\}$ there exists a $\underline{x}_0$ such that $\underline{y} = \mathbf{A}^H \underline{x}_0$. But since the matrices $\mathbf{\Phi}_{zz}^{-1}$ and $\mathbf{A}$ are rank *M*, by Sylvester's Inequality, $\mathbf{\Phi}_{zz}^{-1}\mathbf{A}$ is rank *M* system of equations $\mathbf{\Phi}_{zz}^{-1} \mathbf{A} \underline{y}_0 = \underline{x}_0$ has a solution. Thus, for every $\underline{y} \in \mathcal{R}ange\left\{\mathbf{A}^H\right\}$ there exists an $\underline{y}_0$ such that $\underline{y} = \mathbf{R}_A \underline{y}_0$ and hence, $\mathcal{R}ange\left\{\mathbf{A}^H\right\} \supseteq \mathcal{R}ange\{\mathbf{R}_A\}$. But, before we found that $\mathcal{R}ange\left\{\mathbf{A}^H\right\} \subseteq \mathcal{R}ange\{\mathbf{R}_A\}$ so $\mathcal{R}ange\left\{\mathbf{A}^H\right\} = \mathcal{R}ange\{\mathbf{R}_A\}$.

We have shown thus far that for the spectral square root,

$$\mathcal{R}ange\left\{\mathbf{A}^H\right\} = \mathcal{R}ange\{\mathbf{R}_A\} = \mathcal{R}ange\left\{\tilde{\mathbf{H}}^H\right\}.$$

We will now consider unitary rotations $\mathbf{H} = \mathbf{Q}\tilde{\mathbf{H}}$. If a solution exists to the system $\tilde{\mathbf{H}}^H \underline{y} = \mathbf{A}^H \mathbf{\Phi}_{zz}^{-1} \underline{x}$, then a solution, $\underline{y}_0 = \mathbf{Q}\underline{y}$, exists to the system of equations $\mathbf{H}^H \underline{y}_0 = \mathbf{A}^H \mathbf{\Phi}_{zz}^{-1} \underline{x}$. Since the unitary matrix, **Q**, is invertible, the converse is also true. Hence, $\mathcal{R}ange\left\{\mathbf{A}^H\right\} = \mathcal{R}ange\left\{\mathbf{H}^H\right\}$ which completes our proof.

It should be said that since **H** is not full rank, *infinitely many* solutions exist. In this thesis, we will choose a particular solution: the pseudo-inverse, which is the minimum-norm solution. That is, the pseudo-inverse [222], denoted as

$$\underline{y}_* = \left(\mathbf{H}^H\right)^\dagger \mathbf{A}^H \mathbf{\Phi}_{zz}^{-1} \underline{x} \tag{A.6}$$

finds the solution to equation (A.1) with the smallest L$_2$-norm, $\| \underline{y}_* \|$.

Appendix B: BACKGROUND IN ANTENNA ARRAYS

This thesis will consider a circular antenna array illustrated in Figure B. 1. Consider a single wave-front carrying a signal s(t), impinging on a circular array with a depression angle of ε_d, and an azimuth of θ_d. The vector of complex, baseband, signals experienced by each element, $\underline{x}(t)$, can be expressed as

$$\underline{x}(t) = \underline{a}(\varepsilon_d, \theta_d)s(t) + \underline{z}(t) \tag{B.1}$$

where $\underline{z}(t)$ is a vector of spatially uncorrelated additive white complex Gaussian noise. The mapping of the received signal (w/o noise) to an array response vector is described with a steering vector, $\underline{a}(\varepsilon_d, \theta_d)$.

For simplicity, we will assume that the array elements are perfectly isotropic (i.e. receive with equal gain in all directions). The steering vector for an M element λ/2 spaced circular array is given as

$$\underline{a}(\varepsilon_d, \theta_d) = [a_1 \quad \cdots \quad a_M]^T, \quad a_m = \exp\left(-j\kappa_R \sin\left(\theta_d - {}^{2\pi(m-1)}\!\!\diagup\!_M \right) \cos\varepsilon_d \right) \tag{B.2}$$

where the constant $\kappa_R = 2\pi R_a / \lambda_0$ and λ_0 is the wavelength of the carrier frequency.

A linear beam-former forms a single output signal from a linear combination of the antenna elements.

$$y(t) = \underline{w}^H \underline{x}(t) \tag{B.3}$$

where $\underline{w}$ is carefully chosen vector of element weights. Consider a single signal impinging on the array from a depression angle of ε_d and an azimuth of θ_d. The optimal-SNR beam-former in spatially uncorrelated additive white complex Gaussian noise is given as [223]

$$\underline{w}_{opt} = \underline{a}(\varepsilon_d, \theta_d) \tag{B.4}$$

The gain pattern of a beam former indicates the gain in the power of y(t) for a signal received from a certain direction. For a given weight vector, $\underline{w}$, the gain pattern as a function of azimuth θ, and depression, ε, is given as:

$$G(\varepsilon,\theta) = |\, \underline{w}^H \underline{a}(\varepsilon,\theta)\,|^2 \tag{B.5}$$

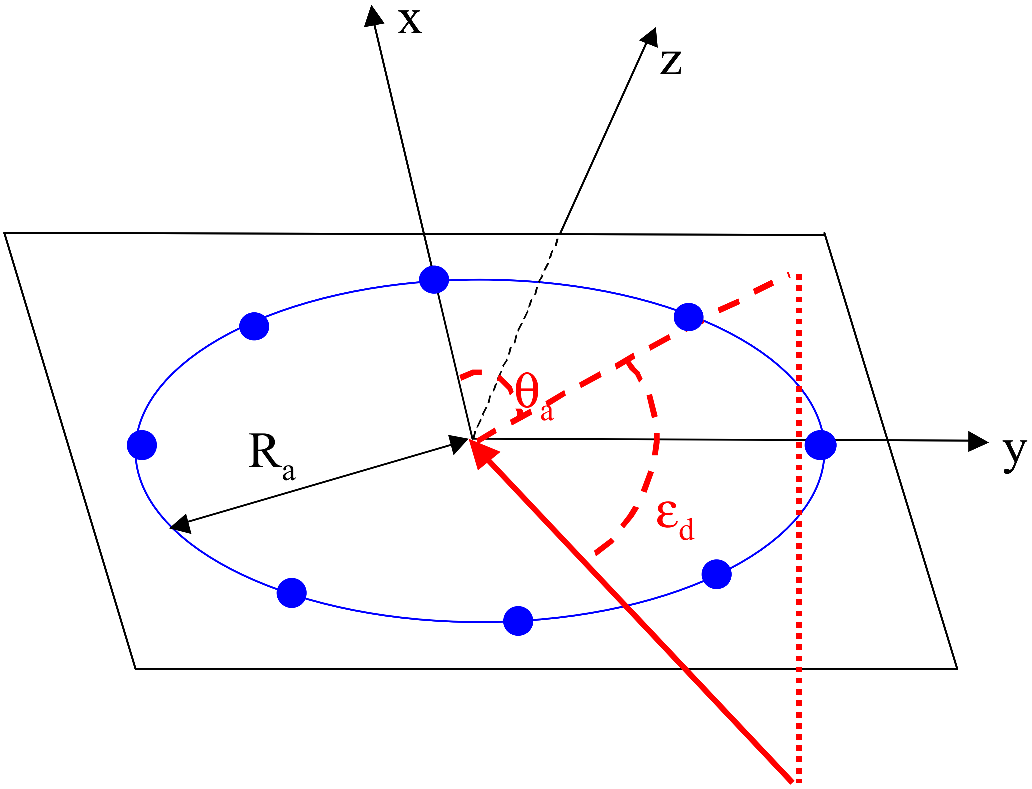

Figure B. 1: Bottom view of an eight element circular array with a planar wave impinging on the array at a depression angle of ε_d and an azimuth angle of θ_d. Each blue dot is assumed to be a perfectly isotropic antenna.

B.1 Optimal SINR Solution

In the subsequent discussion, and for the entirety of this thesis, we will assume that all impinging signals arrive with depression, ε_d= 0. Consequently we will refer to the azimuth angle as the *Angle of Arrival* (AOA).

Now consider the following situation: a number, D_u, of signals impinge on the circular array, each with AOA, θ_d. Further, let these signals be synchronized in baud, frequency, and phase. Let $\underline{x}[n]=\begin{bmatrix} x_1 & x_2 & \cdots & x_M \end{bmatrix}^T$ be a vector of matched filtered and synchronously sampled array signals, $x_m(t)$. Further, let **A** be the $M \times D_u$ matrix whose d^{th} column is the steering vector for the d^{th} signal:

$$\mathbf{A}=\begin{bmatrix} \underline{a}(\theta_1) & \underline{a}(\theta_2) & \cdots & \underline{a}(\theta_{D_u}) \end{bmatrix} \tag{B.6}$$

Then, a discrete equivalent channel can relate the symbols modulated by the d^{th} signal $s_d[n]$ as follows. Define a vector of transmitted signals:

$$\underline{s}[n]=\begin{bmatrix} s_1[n] & s_2[n] & \cdots & s_{D_u}[n] \end{bmatrix}^T$$

and let

$$\underline{z}[n]=\begin{bmatrix} z_1[n] & z_2[n] & \cdots & z_M[n] \end{bmatrix}^T$$

be a vector of Additive White Gaussian Noise (AWGN) match filtered an sampled along with the signals of interest. Then $\underline{z}[n]$ has a stationary auto-correlation matrix $\mathbf{\Phi}_{zz} \triangleq E\left[\underline{z}[n]\underline{z}[n]^H\right]$. The discrete equivalent channel model is given by the following linear equation:

$$\underline{x}[n]=\mathbf{A}\underline{s}[n]+\underline{z}[n] \tag{B.7}$$

We will now define the optimal SINR beamformer, $\underline{w}_d$ for a particular signal, $s_d[n]$. Separate Equation (B.7) into desired and interference terms. For convenience, we will drop the time index, *n*, in the following equations.

$$\underline{x}=\underline{d}_d+\underline{i}_d \tag{B.8}$$

where

$$\underline{d}_d = \underline{a}(\theta_d)s_d \tag{B.9}$$

and

$$\underline{i}_d = A_{\bar{d}}\underline{s}_{\bar{d}} + \underline{z} \tag{B.10}$$

Then the power of the desired signal, s_d, collected by the beamformer, $\underline{w}_d$, is given as

$$P_d \triangleq E\left[\left|\underline{w}_d^H \underline{d}_d\right|^2\right] = \underline{w}_d^H \mathbf{R}_{dd} \underline{w}_d \tag{B.11}$$

where

$$\mathbf{R}_{dd} = \sigma_s^2 \underline{a}(\theta_d)\underline{a}(\theta_d)^H \tag{B.12}$$

Similarly, the power of the interference with the desired signal, s_d, is

$$P_d^{(I)} \triangleq E\left[|\,\underline{w}_d^H \underline{i}_d\,|^2\right] = \underline{w}_d^H \mathbf{R}_{ii}^{(d)} \underline{w}_d \tag{B.13}$$

where

$$\mathbf{R}_{ii}^{(d)} = \sigma_s^2 \mathbf{A}_{\bar{d}} \mathbf{A}_{\underline{d}}^H + \mathbf{\Phi}_{zz} \tag{B.14}$$

and the matrix $\mathbf{A}_{\bar{d}}$ is the matrix, **A**, with its d^{th} column removed. Then the Signal to Interference and Noise Ratio (SINR) is just the ratio

$$SINR \triangleq \frac{P_d}{P_d^{(I)}} = \frac{\underline{w}_d^H \mathbf{R}_{dd} \underline{w}_d}{\underline{w}_d^H \mathbf{R}_{ii}^{(d)} \underline{w}_d} \tag{B.15}$$

It is shown in [223] that this quotient is maximized when

$$\underline{w}_{d,opt} \propto \mathbf{R}_{ii}^{-1} \underline{a}(\theta_d) \tag{B.16}$$

The proportionality indicates that a scalar multiple of any optimum SINR beam-former is itself an optimum SINR beamformer. Usually, we will scale $\underline{w}_{d,opt}$ to have unity norm.

Appendix C: BIBLIOGRAPHY

[1] T.E. Biedka, M.F. Kahn, “Methods for Constraining a CMA Beamformer to Extract a Cyclostationary Signal,” *Second Workshop on Cyclostationary Signals,* Monterey, CA, Aug. 1994.

[2] Van der Veen, Paulraj, “An Analytical Constant Modulus Algorithm,” *IEEE Transactions on Signal Processing*, vol. 44, no. 5, May 1996.

[3] Van der Veen, “Weighted ACMA,” *ICASSP ’99, 1999.*

[4] Castedo, Escudero, Depena, “A Blind Signal Separation Method for Multiuser Communications,” *IEEE Transactions on Signal Processing,* vol. 45, no. 5, May 1997.

[5] J.H. Reed, R. He, “Spectral Correlation of AMPS Signals and its Application to Interference Rejection,”*Vehicular Technology Conference*, 1994.

[6] J. Hamkins, “A Joint Viterbi Algorithm to Separate Cochannel FM Signals,” *ICASSP 1998*

[7] R. Raheli, A. Polydoros, C. Tzou, “Per-Survivor Processing: A General Approach to MLSE in Uncertain Environments,” *IEEE Transactions on Communications,* Vol. 43, No. 2/3/4, February/March/April 1995, pp. 354-507.

[8] A.V. Keerthi, J.J. Shynk, "Separation of Cochannel Signals in TDMA Mobile Radio," *IEEE Transactions on Signal Processing,* Vol.46, No.10, October 1998, pp.2684-2697.

[9] Y.K. Lee, R. Chandrasekaran, J.J. Shynk, "Separation of Cochannel GSM Signals Using an Adaptive Array," *IEEE Transactions on Signal Processing,* Vol. 47, No.7, July 1999, pp.1977-1987.

[10] Agee, Bruzzone, Bromberg, "Exploitation of Signal Structure in Array Based Blind Copy and Copy-Aided DF Systems," *Vehicular Technology Conference,* June 1998.

[11] E.R. Ferrara, Jr. "Frequency domain implementations of periodically time-varying adaptive filters," *IEEE Transactions on Acoustics Speech and Signal Processing,* Vol. 33, No. 8, August 1985, pp. 883-892.

[12] Ranta, Honkasalo, "Co-Channel Interference Cancelling Receiver for TDMA Mobile Systems," *-Proc. of IEEE ICI, Seattle,* 1995, pp. 17-21.

[13] Grant, Cavers, "Performance Enhancement Through Joint Detection of Cochannel Signals Using Diversity Arrays," *IEEE Transactions on Communications*, Vol. 46, No. 8, August 1998.

[14] Giridhar, Chari, Shynk, Gooch, "Joint Demodulation of Cochannel Signals Using MLSE and MAPSD Algorithms," *Proc. of ICASSP, Minneapolis,* 1993, Vol. IV, pp. 160-163, 1993.

[15] J.H. Reed, A.A. Quilici, and T.C. Hsia, "A frequency domain time-dependent adaptive filter for interference rejection," *IEEE Military Communications Conference,* October 1988, pp. 391-397.

[16] *TDMA Cellular/PCS –Radio Interface- Mobile Station –Base station Compatibility – Traffic Channels and FSK Control Channel.* TIA/EIA/IS-136.2-A.

[17] Lindskog, Ahlen, Sternad, “Combined Spatial and Temporal Equalization Using and Adaptive Antenna Array and a Decision Feedback Equalization Scheme,” *Proc. Of Int. Conf. On Acoustics, Speech, and Signal Processing,* May 1995.

[18] Lndskog, Ahlen, Sternad, “Spatio-Temporal Equalization for Multipiath Environments in Mobile Radio Applications,” *Proc. of the 45th IEEE Vehicular Technology Conference*,pp. 775-779 July, 1995.

[19] Torlak, Hansen, Xu, “A Fast Blind Source Separation for Digital Wireless Applications,” *29th Asilomar Conference on Signals, Systems, & Computers* 1998.

[20] Giridhar, Shynk, Mathur, Chari, Gooch, “Nonlinear Techniques for the Joint Estimation of Cochannel Signals,” *IEEE Transactions on Communications,* Vol 45, No. 4, pp. 473-483, April 1997.

[21] Tidestav, Ahlen, Sternad, “A Comparison of Interference Rejection and Multiuser Detection,” IEEE International Symposium on Personal, Indoor and Mobile Radio Communications., pp. 732-736, 1998

[22] Winters, “Signal Acquisition and Tracking with Adaptive Arrays in the Digital Mobile Radio System IS-54 with Flat Fading,” *IEEE Transactions on Vehicular Technology*, Vol. 42, November 1993.

[24] Hallen, Heegard, “Delayed Decision-Feedback sequence Estimation,” *IEEE Transactions on Communications*, Vol. 37, No. 5, pp. 435, May 1989.

[25] Ariyavistakul, Winters, “Joint Equalizatoin and Interference Suppression for High Data Rate Wireless Systems,” *Vehicular Technology Conference,* February 1999.

[26] Ariyavisitakul, Winters, Lee, “Optimum Space-Time Processors with Dispersive Interference: Unified Analysis and Required Filter Span,” *IEEE Transactions on Communications*, Vol 47, No. 7, July 1999.

[27] Heidari, Nikias, “Co-Channel Interference Mitigation in the Time-Scale Domain: The CIMTS Algorithm,” *IEEE Transactions on Signal Processing*, Vol. 44, No. 9, September 1996.

[28] Shin, Nikias, “Adaptive Interference Canceler for Narrowband and Wideband Interferences Using Higher Order Statistics,” *IEEE Transactions on Signal Processing, Vol. 42, No. 10, October 1994.*

[29] Petersen, Falconer, “Suppression of Adjacent-Channel, Cochannel, and Intersymbol Interference by Equalizers and Linear Combiners,” *IEEE Transactions on Communications,* Vol 42, No. 12, pp. 3109-3117 December 1994.

[30] Edepalli, Andayam, “Combined Equalizatoin and Cochannel Interference Cancellation for the Downlink Using Tentative Decisions,” *Proc. ICASSP,* 1999.

[31] Ratnavel, Paulraj, Constantinides, “MMSE Space-Time Equalization for GSM Cellular Systems,” *Vehicular Technology Conference,* pp. 331-335, 1996.

[32] Gregory E. Bottomley, Karl J. Molnar, “Adaptive Channel Estimation for Multichannel MLSE Receivers,” *IEEE Communication Letters,* Vol.3, No.2, February 1999, pp.40 – 42.

[33] F. Pipon, P. Chevalier, P. Vila, D. Pirez, “Practical Implementation of a Multichannel Equalizer for a Propagation with ISI and CCI – Application to a GSM Link,” *Proc. 47th IEEE Vehicular Technology Conf.,* May 1997, pp. 889-893.

[34] G.E. Bottomley, K. Jamal, “Adaptive Arrays and MLSE Equalization,” 45th *IEEE Vehicular Technology Conference,* Volume 1, pages 50-54, 1991.

[35] H. Yoshino, K. Fukawa, H. Suzuki, “Interference Canceling Equalizer (ICE) for Mobile Radio Communication,” *IEEE Transactions on Vehicular Technology,* Vol.46, No.4, November 1997, pp. 849-861.

[36] S.M. Redl, M.K. Weber, and M.W. Oliphant, *An Introduction to GSM,* Mobile Communications Series, Artech House, Inc., 1995.

[37] J.D. Laster and J.H. Reed, “Interference Rejection in Digital Wireless Communications,” *IEEE Signal Processing Magazine*, pp. 37-62, May 1997.

[38] K.J. Molnar, G.E. Bottomley, “Adaptive Array Processing MLSE Receivers for TDMA Digital Cellular/PCS Communications,” *IEEE Jounal on Selected* Areas *in Communications,* Vol. 16, No.8, October 1998, pp. 1340-1351.

[39] A.J. Paulraj, B.C. Ng, “Space-Time Modems for Wireless Personal Communications,” *IEEE Personal Communications,* February 1998, pp. 36-48

[40] K. Fukawa, H. Suzuki, “Blind Interference Canceling Equalizer for Mobile Radio Communication,” *IEICE Transactions on Communications,* Vol.E77-B, No.5, May 1994, pp. 849-861.

[41] B. C. W. Lo, K.B. Letaief, “Adaptive Equalization and Interference Cancellation for Wireless Communication Systems,” *IEEE Trans. On Communications*, Vol. 47, No. 4, April 1999.

[42] Alle-Jan vand der Veen, Shilpa Talwar, A. Paulraj, “Blind Estimation of Multiple Digital Signals Transmitted over FIR Channels,” *Signal Processing Letters*, Vol 2, No. 5., May 1995.

[43] G. Paparisto, K.M. Chugg, “PSP Array Processing for Multipath Fading Channels,” *IEEE Transactions on Communications,* Vol. 47, No. 4, April 1999, pp.504-507.

[45] J. Liang, A.J. Paulraj, “Two Stage CCI/ISI Reduction with Space-Time Processing in TDMA Cellular Networks,” *Conference Record of Thirtieth Asilomar Conference on Signals, Systems and Computers*, pp. p.607-611

[46] S. Ratnavel, A.Paulraj, A.G. Constantinides, “MMSE Space-Time Equalization for GSM Cellular Systems,” *1996 IEEE 46th Vehicular Technology Conference,* pp.331-335, vol.1, 1996.

[47] CTIA Web pages, http://www.wow-com.com/wirelesssurvey/.

[48] Strategis Group Web page, http://www.strategisgroup.com/.

[49] S. Anderson, M.Millnert, B. Wahlberg, “An Adaptive Array for Mobile Communication Systems,” *IEEE Transactions on Vehicular Technology,* Vol. 40, No. 1, February 1991, pp. 231-236.

[53] T.Wu, C. Schlegel, “Interference Cancellation for Narrowband Mobile Communication Systems,” *Vehicular Technology Conference ’99.*

[54] Forney, “Maximum-Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference,” *IEEE Transactions on Information Theory*, pp.363-378, May 1972.

[55] Gottfried Ungerboeck, “Adaptive Maximum-Likelihood Receiver for Carrier-Modulated Data-Transmission Systems,” *IEEE Transactions on Communications*, vol 22. No. 5, pp. 624-636, May 1974.

[56] Reed, Hsia, "The Performance of Time-Dependent Adaptive Filters for Interference Rejection," *IEEE Trans. On Acoustics, Speech, and Signal Processing,* Vol. 38, No. 8, August 1990

[57] W.A. Gardner, "Cyclic Wiener Filtering: Theory and Method," *IEEE Transactions on Communications,* Vol. 41, No. 1, January 1993, pp. 151-163.

[58] J. Karlsson, J. Heinegard, "Interference Rejection Combining for GSM," *Proc. 5th IEEE ICUPC,* September 1996, pp. 433-437.

[59] Giridhar, Chari, Shynk, Gooch, Artman, "Joint Estimation Algorithms for Cochannel Signal Demodulation," *Proc. of IEEE ICC, Geneva,* 1993, pp. 1497-1501.

[60] Hedstrom, Kirlin, "Co-Channel Signal Separation Using Coupled Digital Phase-Locked Loops," *IEEE Transactions on Communications, vol. 44, no. 10, October 1996.*

[64] Hashimoto, "A List-Type Reduced-Constraint Generalization of the Viterbi Algorithm," *IEEE Transactions on Information Theory*, vol. 33, no. 6, pp. 866-976, November 1987.

[65] Van der Veen, Talwar, Paulraj, "A subspace Approach to Blind space-Time Signal Processing for Wireless Communication Systems," *IEEE Transactions on Signal Processing*, vol. 45, no. 1, pp. 173-190 January 1997.

[66] Sheen, Stuber, "MLSE Equalization and Decoding for Multipath-Fading Channels," *IEEE Transactions on Communications*, vol. 39, no. 10, pp. 1455-1464 October 1991.

[67] Liu, Xu, "Smart Antennas in Wireless Systems: Uplink Multiuser Blind Channel and Sequence Detection," *IEEE Trans. On Comm.* Vol. 45, no. 2, pp. 187-199 Feb. 1997.

[68] Jamal, Brismark, "Adaptive MLSE Performance on D-AMPS 1900 Channel," *IEEE Transactions on Vehicular Technology,* vol. 46, no. 3, pp. 634-641 August 1997.

[69] Krenz, Wesolowski, “Comparative Study of space-Diversity Techniques for MLSE Receivers in Mobile Radio,” *IEEE Trans. Vehicular Technology,* vol. 46, no. 3, pp. 653-663 August 1997.

[70] Godara, “Applications of Antenna Arrays to Mobile Communications, Part 1: Performance Improvement, Feasibility, and system Considerations,” vol. 85, no. 7, pp. 1031-1060 July 1997.

[71] Godara, “Application of Antenna Arrays to Mobile Communications, Part II: Beam-Formaing and Direction-of-Arrival Considerations,” vol. 85, no. 8, pp. 1195-1245 August 1997.

[72] Raleigh, Boros, “Joint Space-Time Parameter Estimation for Wireless Communication Channels,” *IEEE Transactions on signal Processing,* vol 46, no. 5, pp. 1333-1343 May 1998.

[73] Izzo, Paura, Poggi, “An Interference-Tolerant Algorithm for Localization of Cyclostationary-Signal Sources,” *IEEE Trans. on Signal Processing*, vol. 40, no. 7, pp. 1682-1686, July 1992.

[74] R. Chandrasekaran, J.J. Shynk, K. Lai, “A Subspace Method for Separating Cochannel TDMA Signals,” *ICASSP 1999.*

[78] M. Yao, L. Jin, Q. Yin, “Selective Direction Finding for Cyclostationary Signals by Exploitation of New Array Configuration,” *ICASSP 1999.*

[79] V.B. Manimohan, W.J. Fitzgerald, “Direction Estimation Using Conjugate Cyclic Cross-Correlation: More Signals than Sensors,” *ICASSP 1999.*

[81] G. Xu, A. Paulraj, Y. Cho, T. Kailath, “ Maximum Likelihood Detection of Co-channel Communication Signals via Exploitation of Spatial Diversity,” *26th Asilomar Conference on Signals, Systems and Computers, Vol. 2, 1992.*

[83] J.W. Modestino, V. Eyuboglu, “Integrated Multielement Receiver Structures for Spatially Distributed Interference Channels,” *IEEE Transactions on Information Theory,* Vol. IT-32, No. 2, March 1986, pp. 195-219.

[86] A. Paulraj, G.G. Raleigh, “Time Varying Vector Channel Estimation for Adaptive Spatial Equalization,” *IEEE Globecomm,* Vol.1, 1995.

[97] Brian G. Agee, Stephan V. Schell, William Gardner, “Spectral Self-Coherence Restoral: A New Approach to Blind Adaptive Signal Extraction Using Antenna Arrays,” *IEEE Proceedings*, Vol 74, No. 40, April 1990.

[98] C. Tidestav, M. Sternad, A. Ahlen, “ Reuse Within a Cell – Intreference Rejection or Multiuser Detection,” *IEEE Transactions on Communications,* Vol. 47, No. 10, pp. 1511-1522, October 1999.

[99] Tidestav, Sternad, Ahlen, “Reuse Within a Cell-Interference Rejection or Multiuser Detection?” *Vehicular Technology Conference ’99.*

[100] Ready, Chari, “Demodulation of Cochannel FSK Signals Using Joint Maximum Likelihood Sequence Estimation,” *27th Asilomar Conference,* Vol. 2, 1993.

[102] Tsuji, Xin, Yoshimoto, “Detection of Direction and Number of Impinging Signals in Array Antennas Using Cyclostationarity,” *Electronics and Communications in Japan,* Part 1, Vol. 82, No. 10, 1999.

[103] Agee, “The Least-Squares CMS: A New Technique for Correction of Constant Modulus Signals,” *IEEE ICASSP*, April 1986, pp. 953-956.

[104] Shynk, Keerthi, Mathur, “Steady-State Analysis of the Multistage Constant Modulus Array,” *IEEE Transactions on Signal Processing,* Vol.44, No.4, April 1996.

[105] Shynk, Keerthi, Mathur, “Convergence Properties of the Multistage Constant Modulus Array for Correlated Sources,” *IEEE Transactions on Signal Processing,* Vol.45, No.1, January 1997.

[106] Talwar, Viberg, Paulraj, “Blind Separation of Synchronous Co-Channel Digital Signals Using an Antenna Array – Part I: Algorithms,” *IEEE Transactions on Signal Processing,* Vol. 44, No.5, May 1996, pp. 1184-1197.

[107] Talwar, Viberg, Paulraj, “Blind Separation of Synchronous Co-Channel Digital Signals Using an Antenna Array – Part II: Performance Analysis,” *IEEE Transactions on Signal Processing,* Vol. 45, No.3, March 1997, pp. 706-718.

[108] Ranheim A., “A Decoupled Approach to Adaptive Signal Separation Using an Antenna Array,” *IEEE Transactions on Vehicular Technology,* Vol. 48, No. 3, May 1999, pp. 676-682.

[109] Talwar, Viberg, Paulraj, “Reception of Multiple Co-Channel Digital Signals using Antenna Arrays with Applications to PCS,” *SUPERCOMM/ICC,* Vol. 2, pp. 790-794, 1994.

[110] Hansen L.K., Xu G., “A Fast Algorithm for the Blind Separation of Digital Co-Channel Signals,” *31st Asilomar Conference,* Vol.2, 1997.

[111] Dogan M.C, Mendel J.M., “Applications of cumulants to array processing I: Aperture extension and array calibration,” *IEEE Transactions on Signal Processing,* Vol. 43, No.5, May 1995, pp. 1200-1216.

[112] Dogan M.C, Mendel J.M., “Applications of cumulants to array processing IV: Direction finding in coherent signals case,” *IEEE Transactions on Signal Processing,* Vol. 45, No.9, September 1997, pp. 2265-2276.

[113] Dogan M.C, Mendel J.M., “Applications of cumulants to array processing III: Blind beamforming for coherent signals,” *IEEE Transactions on Signal Processing,* Vol. 45, No.9, September 1997, pp. 2252-2264.

[115] Agee B.G., “Exploitation of Signal Structure in Array-Based Blind Copy and Copy-Aided DF Systems,” *ICASSP Presentation,* May 13, 1998.

[116] Bottomley G.E., Molnar K.J., Chennakeshu S., “Interference Cancellation with an Array Processing MLSE Receiver,” *IEEE Transactions on Vehicular Technology,* Vol.48, No.5, September 1999, pp. 1321-1331.

[117] R.C. North, R.A. Axford, J.R. Zeidler, “The performance of adaptive equalization for digital communication systems corrupted by interference,” *Asilomar1993,* Vol.2, pp. 1548-1554.

[132] R. Roy, T. Kailath, “ESPRIT-Estimation of Signal Parameters Via Rotational Invariance Techniques,” *IEEE Transactions on Acoustics, Speech, and Signal Processing,* Vol.37, No.7, July 1989, pp. 984-994.

[136] T.S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice Hall, New Jersey, 1996.

[137] P. Petrus, “Novel Adaptive Array Algorithms and Their Impact on Cellular System Capacity,” in *Ph.D. Dissertation, Virginia Polytechnic Institute and State University, Blacksburg,* March 1997.

[138] J.C. Liberti, T.S. Rappaport, *Smart Antennas for Wireless Communications: 1S-95 and Third Generation CDMA Applications,* Prentice Hall, New Jersey, 1999.

[139] R.O. Schmidt, "Multiple Emitter Location and Signal Parameter Estimation," *Proc. of RADC Spectrum Estimation Workshop,* Griffiss AFB, NY, pp. 243-258, 1979.

[140] J. Zhang, K.M. Wong, Z.Q. Luo, P.C. Ching, "Blind Adaptive FRESH Filtering for Signal Extraction," *IEEE Transactions on Signal Processing,* Vol.47, No.5, May 1999, pp.1397 – 1402.

[141] W.A. Gardner, *Cyclostationarity in Communications and Signal Processing,* IEEE Press, NY, 1994.

[145] M.J. Rude, L.J. Griffiths, "An Untrained, Fractionally-Spaced Equalizer for Co-Channel Interference Environments," *24th Asilomar Conference on Signals, Systems and Computers,* 1992.

[149] N.W.K. Lo, D.D. Falconer, A.U.H. Sheikh, "Adaptive Equalization for a Multipath Fading Environment with Interference and Noise," *VTC'94*, Vol. 1, 1994.

[150] N.W.K. Lo, D.D. Falconer, A.U.H. Sheikh, "Adaptive Equalization Techniques for Multipath Fading and Co-Channel Interference," *VTC'93*, 1993.

[152] B.G. Agee, "Blind Separation and Capture of Communication Signals Using a Multitarget Constant Modulus Beamformer," *Proc. MILCOM,* May 1989, pp. 340-346.

[157] R. Lupas, S. Verdu, "Linear Multiuser Detectors for Synchronous Code-Division Multiple Access Channels," *IEEE Transactions on Information Theory,* Vol. 35, No. 1, January 1989, pp. 123-136.

[158] M. Honig, U. Madhow, S. Verdu, "Blind Adaptive Multiuser Detection," *IEEE Transactions on Information Theory,* Vol. 41, No. 4, July 1995, pp. 944-960.

[169] A. Van der Veen, A. Paulraj, "Singular Value Analysis of Space-Time Equalization in the GSM Mobile System," *ICASSP'96,* Vol. 2, pp.1073-1076, 1996.

[170] J.G. Proakis, *Digital Communications,* McGraw-Hill, New York, 3rd Ed., 1995.

[171] K. Abend, B.D. Fritchman, "Statistical Detection for Communication Channels with Intersymbol Interference," *Proc. IEEE,* Vol. 58, May 1970, pp. 779-785.

[172] M.V. Eyuboglu, S.U.H. Qureshi, "Reduced-state Sequence Estimation with Set Partitioning and Decision Feedback," *IEEE Transactions on Communications,* Vol. 36, January 1988, pp. 13-20.

[173] G.D. Forney, "The Viterbi Algorithm," *Proceedings of IEEE,* Vol. 61, No.3, March 1973, pp. 268-278.

[174] S. Haykin, *Adaptive Filter Theory,* Third Edition, Prentice-Hall, 1996.

[175] J.J. Shynk, R.P. Gooch, "The Constant Modulus Array for Cochannel Signal Copy and Direction Finding," *IEEE Transactions on Signal Processing,* Vol. 44, No. 3, March 1996.

[176] R.P. Gooch, J.D. Lundell, "The CM Array: An Adaptive Beamformer for Constant Modulus Signals," *Proc. ICASSP, Tokyo,Japan,* April 1986.

[177] B.J. Sublett, R.P. Gooch, S.H. Goldberg, "Separation and Bearing Estimation of Cochannel Signals," *Proc. of IEEE Military Communications Conference,* May 1989, pp. 629-634.

[178] R. Lonski, R.P. Gooch, "An experimental angle of arrival system," *Proc. of the Asilomar Conf. on Signals, Systems, and Computers,* November 1991, pp. 969-973.

[179] R.D. Hughes, E.J. Lawrence, L.P. Withers, "A Robust CMA Adaptive Array for Multiple Narrowband Sources," *Proc. of the Asilomar Conf. on Signals, Systems, and Computers,* November 1992, pp. 35-39.

[180] J. Capon, "High Resolution Frequency-Wavenumber Spectral Analysis," *Proc. of IEEE*, Vol. 57, No. 8, August 1969, pp. 1408-1418.

[181] J. Capon, "Maximum Likelihood Spectral Estimation," *Nonlinear Methods of Spectral Analysis,* Ed. By S. Haykin, Springler, NY, 1979.

[182] A.J. Barabell, "Improving the Resolution Performance of Eigenstructure-based Direction Finding Algorithms," *Proc. of ICASSP –83,* 1983, pp. 336-339.

[183] S.V. Schell, Calabretta, W.A. Gardner, B.G. Agee, "Cyclic MUSIC Algorithms for Signal Selective DOA Estimation," *Proc. of ICAASP –89,* 1989, pp. 2278-2281.

[184] D. Feldman, L.J. Griffiths, "A Constraint Projection Approach for Robust Adaptive Beamforming," *Proc. of ICASSP,* May 1991, pp. 1381-1384.

[185] J.E. Evans, J.R. Johnson, D.F. Sun, "High Resolution Angular Spectrum Estimation Techniques for Terrain Scattering Analysis and Angle of Arrival Estimation in ATC Navigation and Surveillance System," *MIT Lincoln Lab.,* Lexington, MA, Rep. 582, 1982.

[186] T.J. Shan, M. Wax, T. Kailath, "On Spatial Smoothing for Estimation of Coherent Signals," *ICASSP,* Vol. ASSP-33, August 1985.

[187] K. Takao, N. Kikuma, "An Adaptive Array Utilizing an Adaptive Spatial Averaging Technique for Multipath Environments," *IEEE Trans. on Antennas and Propagation,* Vol. AP-35, No. 12, December 1987, pp. 1389-1396.

[188] F. Haber, M. Zoltowski, "Spatial Spectrum Estimation in a Coherent Signal Environment Using an Array in Motion," *IEEE Trans. on Antennas and Propagation,* Vol. AP-34, March 1986, pp. 301-310.

[189] M.J. Rude, L.J. Griffiths, "Incorporation of Linear Constraints into the Constant Modulus Algorithm," *Proc. of ICASSP,* Glasgow, Scotland, UK, May 1989.

[190] W.A. Gardner, "Simplification of MUSIC and ESPRIT by Exploitation of Cyclostationarity," *Proc. of IEEE,* Vol. 76, July 1988, pp. 845-847.

[192] G. Gelli, L. Izzo, "Minimum-Redundancy Linear Arrays for Cyclostationary-based Source Location," *IEEE Transactions on Signal Processing,* Vol. 45, October 1997, pp.2605-2608.

[194] S.V. Schell, B.G. Agee, "Application of the SCORE algorithm and SCORE extensions to sorting in the rank-L spectral self coherence environment," *Proc. of the 22nd Asilomar Conf. on Signals, Systems, and Computers,* December 1988, pp. 274-278.

[195] S.V. Schell, W.A. Gardner, "Maximum likelihood and common factor analysis-based blind adaptive spatial filtering for cyclostationary signals," *Proc. ICAASP*, Minneapolis, MN, April 1993, pp. 292-295.

[196] T.E. Biedka, "Subspace constrained SCORE algorithms," *Proc. of Asilomar Conf. on Signals, Systems, and Computers,* November 1993, pp. 716-720.

[197] S. Verdu, "Minimum Probability of Error for Asynchronous Gaussian Multiple-Access Channels," *IEEE Transactions on Information Theory*, vol IT-32, no. 1, January 1986.

[198] G.J. Bierman, *Factorization Method for Discrete Sequential Estimation,* Academic Press, New York, 1977.

[201] S.N. Diggavi, A. Paulraj, "Performance of Multisensor Adaptive MLSE in fading channels," *Proc. IEEE VTC*, pp. 2148-2152, May 1997.

[202] E. Lindskog, "Multi-channel Maximum Likelihood Sequence Estimation," *Proc. IEEE VTC*, pp. 715-719, May 1997.

[203] K. Fukawa, H. Suzuki, "Blind Interference Canceling Equalizer for Mobile Radio Communications," *IEICE transactions on communications,* Vol. E77-B, No. 5, May 1994.

[204] R. Mendoza, J.H. Reed, T.C. Hsia, B.G. Agee, " Interference Rejection Using the Time-Dependent Constant Modulus Algorithm (CMA) and the Hybrid CMA/Spectral Correlation Discriminator," *IEEE Transactions on Signal Processing,* Vol. 39, No. 9, September 1991, pp. 2108 – 2111.

[205] Van Etten, "Maximum Likelihood Receiver for Multiple Channel Transmission Systems," *IEEE Transactions on Communications*, pp. 276, vol. 24, February, 1976.

[206] Liu, Xu, Tong, Kailath, "Recent developments in Blind Channel Equalization: from Cyclostationarity to Subspaces," *Signal Processing*, pp. 83-89, vol 50, April 1996.

[207] Sergio Verdu, *Multi-user Detection*, Cambridge, UK: Cambridge University Press, 1998.

[208] Liu, Xu, Tong, Kailath, "Recent developments in Blind Channel Equalization: from Cyclostationarity to Subspaces," *Signal Processing*, pp. 83-89, vol 50, April 1996.

[209] G.D. Forney, Jr., "The Forward-Backward Algorithm", Procedings of 34th Annual Allerton Conference on Communication, Control, and Computing, Univ. Illinois. 1996, pp. 432-446.

[210] A. Fernandez, K. Efe, "Generalized Algorithm for Parallel Sorting on Product Networks", *IEEE Transactions on Parallel and Distributed Systems,* vol. 8, no. 12, Dec. 1997.

[211] Ulukus, "Optimum Multiuser Detection Is Tractable for Synchronous CDMA Systems Using M-Sequences", vol. 2, no. 4, April 1998.

[212] C. Sankaran, A. Ephremides, "Solving a Class of Optimum Multiuser Detection Problems with Polynomial Complexity," *IEEE Transactions on Information Theory*, vol. 44, no. 5, Sept. 1998.

[213] J. Hagenauer and P. Hoeher, "A Viterbi Algorithm with soft-decision outputs and its applications, " Proc. IEEE Globecom, pp. 1680-1686, 1989.

[214] M. Moher, "An Iterative Multi-user Decoder for Near-Capacity Communications", IEEE Transactions on Communications, vol. 46, no. 7, July 1998, pp. 870-880.

[215] J. G. Proakis, D.G. Manolakis, Digital Signal Processing, Principles, Algorithms, and Applications, Third Edition, Prentice Hall, Upper Saddle River, NJ, © 1996.

[216] B. Frey, *Graphical Models for Machine Learning and Digital Communication*, MIT Press Cambridge, MA, © 1998.

[217] A. Reznik, *Iterative Decoding of Codes Defined on Graphs*, MIT Thesis, June 1998.

[218] S. Bayram, J. Hicks, J.H. Reed, B. Boyle, "Overloaded Array Processing in Wireless Airborne Communication Systems", to be published in MILCOM, October 22-25, 2000, Los Angeles.

[219] S. Bayram, J. Hicks, J.H. Reed, B.Boyle, "Overloaded Array Processing: Non-Linear vs. Linear Signal Extraction Techniques", to be published in Wireless 2000 Conference, Calgary, July 10-12, 2000.

[220] S. Bayram, J. Hicks, J.H. Reed, B. Boyle, " Joint ML Approach in Overloaded Array Processing," to be published in *Vehicular Technology Conference*, Sept. 24-28, Boston.

[221] B. Agee, *The Property-Restoral Approach to Blind Adaptive Signal Extraction*, University of California Davis Dissertation, © 1989.

[222] R. Horn, C. Johnson, Matrix Analysis, © 1985, Camgridge University Press, NY, NY.

[223] R. Monzingo, T. Miller, Introduction to Adaptive Arrays, © 1980, John Wiley & Sons, Inc. , NY, NY.

[224] J. Litva, T. Lo, *Digital Beamforming in Wireless Communications,* © 1996, Artech House, Boston, MA.

VITA

James Hicks was born in Fairfax, VA on September 13, 1974. He received his Bachelor of Science degree in Electrical Engineering from George Mason University, Fairfax, VA (GMU) in May 1997. While pursuing his undergraduate studies at GMU, he completed two Cooperative Education programs. From June 1994 to June 1996, James worked at the United States Naval Research Laboratory as a graphics programmer for real-time tactical warfare simulation. From June 1996 to June 1997, he worked at Stanford Telecommunications where he helped develop several satellite propagation model tools. James has been consulting part time for Information Systems Laboratories (ISL) in Vienna, VA since 1997. While at ISL, he has developed algorithms and system analysis for single satellite position location systems. James is currently pursuing his Ph.D. in electrical engineering at Virginia Tech, Blacksburg, VA as a Bradley Fellow. His research interests are digital signal processing and system modeling for wireless communications with a special interest in antenna arrays, spread-spectrum, and Markov modeling.