ON LIST DETECTION FOR OVERLOADED RECEIVERS

Michael Krause, Desmond P. Taylor and Philippa A. Martin
Department of Electrical and Computer Engineering, University of Canterbury,
Christchurch, New Zealand
Email: {michael.krause, taylor, p.martin}@elec.canterbury.ac.nz

ABSTRACT

Excessive co-channel interference (CCI) occurs in overloaded wireless environments where the number of transmitted signals exceeds the number of available receive antennas. We present a list-based multiuser detection scheme that effectively mitigates CCI in overloaded receivers with a circular array. The algorithm is named parallel symbol detection with reduced complexity interference estimation (PSD-RCIE). It employs an iterative parallel detector with list feedback of the best symbol estimates. Simulations show that our algorithm achieves near joint-maximum likelihood detection at a fraction of the complexity. Its good complexity-performance trade-off together with its parallel structure make PSD-RCIE well suited for practical implementation.

I. INTRODUCTION

Increasing capacity, data rate and reliability of data transfer are important issues in wireless communications. Multiple antennas at the receiver allow significant increases in capacity and reliability of wireless data transfer by exploiting spatial diversity. Wireless communication systems with multiple co-channel users suffer from severe co-channel interference (CCI). As a result, demodulation of the signals of interest at the receiver becomes a challenging task.

Multiuser detection (MUD) in overloaded receive environments has been analyzed by several authors. Grant et. al. [1] showed that signal extraction in overloaded receivers is possible using a complex joint maximum a posteriori algorithm. The work in [2, 3] analyzed maximum likelihood approaches for separating and estimating synchronous co-channel signals impinging on an antenna array. Bayram et. al. [4] compared the approaches of [2, 3] in the overloaded case. They showed that linear algorithms fail and that joint-maximum likelihood (JML) detection is the optimal approach. JML performs an exhaustive search over all possible symbol vectors, making it impractical. In [5], Hicks et. al. attempt to significantly reduce detection complexity while still approximating JML. Their spatially reduced search joint detection (SRSJD) algorithm combines a linear spatial pre-filter that mitigates CCI with a non-linear reduced search algorithm. SRSJD exploits delayed decision-feedback sequence estimation [6] to search only over user signals with high energy in each beam of the spatial pre-filter. It uses an adaptation of the Viterbi algorithm [7] referred to as iterative tail-biting delayed decision feedback sequence estimation (ITB-DDFSE) for symbol detection.

In this paper we develop a list-based iterative approach to MUD in overloaded environments. Our algorithm is named parallel symbol detection with reduced complexity interference estimation (PSD-RCIE). It outputs a list of most likely symbol vectors together with soft information, achieves near JML performance, and has a good complexity-performance tradeoff.

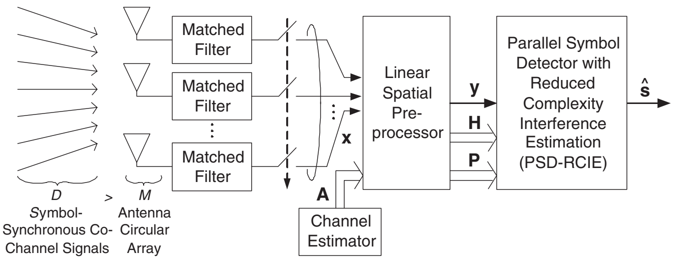

Matched Filter
Matched Filter
Matched Filter
x
A
Linear Spatial Pre-processor
Channel Estimator
y
H
P
Parallel Symbol Detector with Reduced Complexity Interference Estimation (PSD-RCIE)
ŝ
D > M
Symbol-Synchronous Co-Channel Signals
Antenna Circular Array

Figure 1: System model.

In Section II, we introduce the system model and the linear spatial preprocessor which mitigates CCI. Our PSD-RCIE algorithm is described in Section III. Simulation results are shown in Section IV. Computational complexity is considered in Section V and conclusions are drawn in Section VI.

II. SYSTEM MODEL AND SPATIAL PREPROCESSING

As an example, we consider a high altitude receiver node with an M element circular antenna array looking at the ground. Such a receiver is often referred to as the "base station in the sky" model [8, 9]. We assume $D > M$ co-channel complex PAM signals impinging on the circular receive array. Fig. 1 depicts the proposed receiver. The received signals are passed through a bank of filters matched to the transmitted pulse shapes and then sampled to give the received signal vector $\mathbf{x}$. A restriction[1] is made to the symbol-synchronous case for which, the received $M \times 1$ signal vector is given as

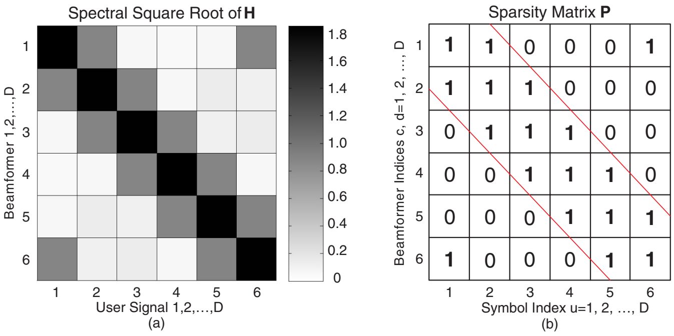
$$\mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{z} \tag{1}$$

where $\mathbf{s} = [s_1 \; s_2 \; \ldots \; s_D]^T$ is the vector containing the symbols transmitted by users 1 through D. Each s_d is independent and uniformly drawn from an alphabet $\mathcal{A}$. The symbol sets are multiplied by the $M \times D$ composite array response matrix $\mathbf{A}$, where the d-th column of $\mathbf{A}$ is the array steering vector for the d-th user's signal. The quantity $\mathbf{z}$ is an $M \times 1$ temporally uncorrelated noise vector with zero mean and autocorrelation $\mathbf{\Phi}_{zz} = E\left[\mathbf{z}\mathbf{z}^H\right]$. For spatially uncorrelated noise, $\mathbf{\Phi}_{zz} = \sigma_z^2\mathbf{I}$, where σ_z^2 denotes the noise variance and $\mathbf{I}$ is the $M \times M$ identity matrix. Note that any time dependance in (1) has been dropped for convenience. We consider only the overloaded case where $D > M$. For simplicity we assume that no intersymbol interference (ISI) is present in the channel and that perfect channel state information is available at the receiver.

As shown in Fig. 1, the array response matrix $\mathbf{A}$ and the received signal vector $\mathbf{x}$ are input to the linear spatial preprocessor. It exploits the fact that user signals impinging on the

[1] The extension to the non-synchronous case is straightforward.

1-4244-1144-0/07/$25.00©2007 IEEE



	1	2	3	4	5	6
1	1	1	0	0	0	1
2	1	1	1	0	0	0
3	0	1	1	1	0	0
4	0	0	1	1	1	0
5	0	0	0	1	1	1
6	1	0	0	0	1	1

Figure 2: Example of (a) spectral square root of $\mathbf{H}$ [5] and (b) sparsity matrix $\mathbf{P}$ for a $M = 5$ circular antenna array with $D = 6$ equal energy users uniformly spaced in AOA.

antenna array with greater spread in angle of arrival (AOA) interfere with each other less than signals closely spaced in AOA. As in [5], we define a $D \times D$ square matrix, $\mathbf{H}$, and a $D \times 1$ vector, $\mathbf{y}$, such that [5]

$$\begin{aligned} \mathbf{H}^H\mathbf{H} &= \mathbf{A}^H\mathbf{\Phi}_{zz}^{-1}\mathbf{A} \\ \mathbf{H}^H\mathbf{y} &= \mathbf{A}^H\mathbf{\Phi}_{zz}^{-1}\mathbf{x} \end{aligned} \quad (2)$$

where $(\cdot)^H$ denotes Hermitian transpose. Using (2) yields the JML detector defined as [5]

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}\in\mathcal{A}^D} \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2 = \arg\min_{\mathbf{s}\in\mathcal{A}^D} \sum_{d=1}^{D} |y[d] - \mathbf{h}[d]\mathbf{s}|^2 \quad (3)$$

where the quantity $\mathbf{h}[d]$ is the d-th row of $\mathbf{H}$ and $y[d]$ is the d-th element of $\mathbf{y}$. The minimization in (3) is over a discrete alphabet and requires a search over all $|\mathcal{A}|^D$ possible transmit symbol combinations, $\mathcal{A}^D$. Rearranging (2) gives the spatial filter output [5]

$$\mathbf{y} = \left(\left(\mathbf{H}^H\right)^\dagger \mathbf{A}^H\mathbf{\Phi}_{zz}^{-1}\right)\mathbf{x} = \mathbf{W}\mathbf{x} \quad (4)$$

where $(\cdot)^\dagger$ denotes pseudo-inverse and the $D \times M$ matrix $\mathbf{W}$ is a trellis oriented multiple-input multiple-output (MIMO) beam former. Each row of $\mathbf{W}$ implicitly puts a beam in the direction of one user. Fig. 2(a) shows an example of $\mathbf{H}$ for an $M = 5$ circular antenna array and $D = 6$ equal power users uniformly spaced in AOA. Note that darker shading indicates increased energy concentration. We see that most of the energy is contained on or near the main diagonal of $\mathbf{H}$ and in each row only a few elements contain most of the energy. From $\mathbf{H}$, a $D \times D$ matrix $\mathbf{P}$ is constructed as illustrated in Fig. 2(b). This matrix is called a sparsity matrix [5]. It contains unity entries for user signals that, after spatial filtering, have "high" energy contribution and zero entries for signals with "low" energy in the corresponding spatial beam. We define *enumeration sets*[2], $U_e[d]$, which contain the column indices of the nonzero elements of the d-th row of $\mathbf{P}$. The "high" and "low" energy symbols are distinguished by a predefined value of the desired energy to interference ratio (DEIR) [5]

$$DEIR[d] = \frac{E\left[|h_{dd}s_d|^2\right]}{E\left[\left|\sum_{u\in\overline{U}_e[d]} h_{du}s_u\right|^2\right]} = \frac{|h_{dd}|^2}{\left|\sum_{u\in\overline{U}_e[d]} h_{du}\right|^2} \quad (5)$$

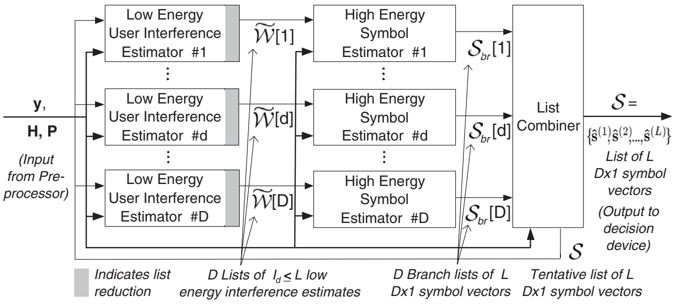

Low Energy User Interference Estimator #1
Low Energy User Interference Estimator #d
Low Energy User Interference Estimator #D
$\widetilde{\mathcal{W}}[1]$
$\widetilde{\mathcal{W}}[d]$
$\widetilde{\mathcal{W}}[D]$
High Energy Symbol Estimator #1
High Energy Symbol Estimator #d
High Energy Symbol Estimator #D
$\mathcal{S}_{br}[1]$
$\mathcal{S}_{br}[d]$
$\mathcal{S}_{br}[D]$
List Combiner
$\mathcal{S}$ =
$\{\hat{\mathbf{s}}^{(1)}, \hat{\mathbf{s}}^{(2)}, ..., \hat{\mathbf{s}}^{(L)}\}$
List of L Dx1 symbol vectors
(Output to decision device)
y,
H, P
(Input from Pre-processor)
$\mathcal{S}$
Indicates list reduction
D Lists of $I_d \leq L$ low energy interference estimates
D Branch lists of L Dx1 symbol vectors
Tentative list of L Dx1 symbol vectors

Figure 3: Block diagram of the PSD-RCIE detector.

where $u \in \overline{U}_e[d]$ defines the column indices of the zero elements in the d-th row of $\mathbf{P}$. The quantity h_{dd} is the d-th diagonal element of $\mathbf{H}$ and h_{du} denotes a non-diagonal matrix element with low energy. Similarly, s_d is the symbol value corresponding to the d-th diagonal element of $\mathbf{H}$ and $\left\{s_u | u \in \overline{U}_e[d]\right\}$ are the low energy symbols corresponding to the d-th row of $\mathbf{H}$. As an example, in row one of $\mathbf{H}$ in Fig. 2(a), $U_e[1] = \{6, 1, 2\}$ and $\overline{U}_e[1] = \{3, 4, 5\}$ are the column indices of elements with high and low energy, respectively. Hence, we can define $\tau[d]$ and $\omega[d]$ as the sets of high and low energy user symbols such that

$$\tau[d] = \{s_u | u \in U_e[d]\}\,, \quad \omega[d] = \left\{s_u | u \in \overline{U}_e[d]\right\}. \quad (6)$$

We refer to the low energy sets, $\omega[d]$, as interfering symbol sets, since they cause CCI which makes detection of the high energy symbol sets, $\tau[d]$, challenging.

III. Parallel Symbol Detection with Reduced Complexity Interference Estimation

We now describe the proposed PSD-RCIE algorithm, shown in Fig. 3. The detector takes the filtered receive vector $\mathbf{y}$, the matrix $\mathbf{H}$ and the sparsity matrix $\mathbf{P}$ as inputs. After iterative processing the overall detector outputs a list of symbol vectors, $\mathcal{S} = \left\{\hat{\mathbf{s}}^{(1)}, \hat{\mathbf{s}}^{(2)}, \ldots, \hat{\mathbf{s}}^{(L)}\right\}$, where $\hat{\mathbf{s}}^{(l)}$ is the l-th $D \times 1$ symbol vector in the list.

For each element in $\mathbf{y}$, a parallel detector branch computes a branch list $\mathcal{S}_{br}[d]$ of L ($D \times 1$) symbol vectors. The d-th branch list is given by $\mathcal{S}_{br}[d] = \left\{\hat{\mathbf{s}}_{br}^{(1)}[d], \hat{\mathbf{s}}_{br}^{(2)}[d], \ldots, \hat{\mathbf{s}}_{br}^{(L)}[d]\right\}$, where $\hat{\mathbf{s}}_{br}^{(k)}[d]$ is the k-th branch symbol vector in the d-th list, $k = 1, 2, \ldots, L$ and $d = 1, 2, \ldots, D$. The branch symbol vectors, $\hat{\mathbf{s}}_{br}^{(k)}[d]$, are computed in two stages. First, interfering symbol sets, $\omega[d]$, are estimated by a low energy user interference estimator which finds the list of L estimates $\left\{\hat{\omega}^{(1)}[d], \hat{\omega}^{(2)}[d], \ldots, \hat{\omega}^{(L)}[d]\right\}$. A *list reduction process*[3], indicated by the grey tinted blocks in Fig. 3, removes redundant elements from the initial list giving the reduced list $\widetilde{\mathcal{W}}[d] = \left\{\tilde{\hat{\omega}}^{(1)}[d], \tilde{\hat{\omega}}^{(2)}[d], \ldots, \tilde{\hat{\omega}}^{(I_d)}[d]\right\}$. We denote the remaining list elements as $\tilde{\hat{\omega}}^{(i)}[d]$, where $i = 1, 2, \ldots, I_d$ is the index variable and $1 \leq I_d \leq L$. Using the estimates in $\widetilde{\mathcal{W}}[d]$, a high energy symbol estimator computes the branch list $\mathcal{S}_{br}[d]$ by searching over all high energy symbol sets, $\tau[d]$. Hence, we decompose

[2] The term *enumeration set* was chosen in [5] because the detection algorithm enumerates over all combinations of user symbols $\{s_u | u \in U_e[d]\}$.

[3] List reduction reduces the complexity in subsequent detection stages.

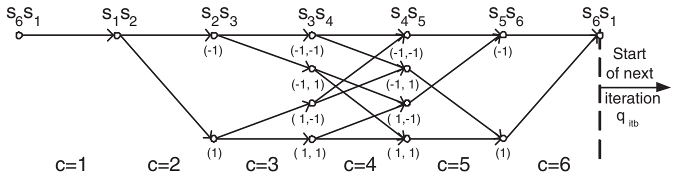


Figure 4: ITB-DDFSE trellis for CCI estimation in row one of Fig. 2 using BPSK signals.

each $\hat{\mathrm{s}}_{br}^{(k)}[d] \in \mathcal{S}_{br}[d]$ into $\hat{\mathrm{s}}_{br}^{(k)}[d] = \left\{ \widetilde{\hat{\omega}}^{(k)}[d], \hat{\tau}^{(k)}[d] \right\}$, where $\widetilde{\hat{\omega}}^{(k)}[d] \in \widetilde{\mathcal{W}}[d]$ and $\hat{\tau}^{(k)}[d]$ are estimates of the symbol sets $\omega[d]$ and $\tau[d]$, respectively. The D ordered branch lists $\mathcal{S}_{br}[d]$ are input to a list combiner where the list elements are searched and combined to find the tentative ordered[4] list, $\mathcal{S}$, of the L most likely symbol vectors $\hat{\mathrm{s}}^{(l)}$. This list is stored in the list combiner and is fed back to the D detector branches as input for the next iteration. After a sufficient number of iterations, Q, the detector outputs the ordered list of the L most likely symbol vectors, $\mathcal{S}$. Typically, PSD-RCIE requires $Q = 2$ or 3 iterations. A decision device selects the first list element $\hat{\mathrm{s}}^{(1)} \in \mathcal{S}$ as the best estimate of s.

A. *Low Energy User Interference Estimation*

In overloaded environments a linear spatial filter cannot remove all CCI due to interfering symbol sets, $\omega[d]$. As a result, part of the CCI remains in the filtered received vector, $\mathbf{y}$. This increases with the overload factor $f = D/M$. If the exact CCI were known to the detector, all remaining interference could be cancelled and detection of the high energy symbol sets, $\tau[d]$, would be straightforward. Hence, the purpose of the low energy user interference estimators in Fig. 3 is to compute low complexity estimates of the remaining CCI.

We employ the ITB-DDFSE algorithm of [5] for CCI estimation. It uses the sparsity matrix $\mathbf{P}$ to construct a trellis across space and applies the Viterbi algorithm to find the best path. As an example, Fig. 4 shows the trellis used by low energy user interference estimator #1 in Fig. 3 for the $M = 5$ antenna, $D = 6$ user environment of Fig. 2. The trellis is shown for BPSK signaling. The extension to other signal types is straightforward. We use the index $c = 1, 2, \ldots, D$ to distinguish trellis stages from detection branches $d = 1, 2, \ldots, D$. The states at the c-th trellis stage are defined as

$$\sigma[c] = \{s_u | u \in U_e[c-1] \cap U_e[c]\} = \tau[c-1] \cap \tau[c]. \quad (7)$$

Note that in row $c = 1$ of $\mathbf{H}$ (Fig. 2(a)), $\tau[1] = \{s_6 s_1 s_2\}$ are the high energy symbols which are estimated separately. Thus, all symbols in $\tau[1]$ are represented by fixed states. The state sequence is $\sigma[1] = \{s_6 s_1\}$, $\sigma[2] = \{s_1 s_2\}$, ..., $\sigma[6] = \{s_5 s_6\}$ and the number of variable symbols in $\sigma[c]$ is $\{0, 0, 1, 2, 2, 1\}$.

For the q-th overall iteration, $q > 1$, and each detector branch d, ITB-DDFSE is applied to all L vectors in the tentative list $\mathcal{S}$. Hence, L estimates of the interfering symbol sets, $\omega[d]$, for a given list $\mathcal{S}$ are computed in each branch and iteration. We have observed that for different high energy symbol sets, $\hat{\tau}^{(j)}[d]$, ITB-DDFSE often finds the same estimates $\hat{\omega}^{(1)}[d] = \hat{\omega}^{(2)}[d] = \ldots = \hat{\omega}^{(i)}[d]$, where $1 \leq i, j \leq L$. We can therefore reduce each list by removing redundant symbol sets, to form the reduced list $\widetilde{\mathcal{W}}[d] = \left\{ \widetilde{\hat{\omega}}^{(1)}[d], \widetilde{\hat{\omega}}^{(2)}[d], \ldots, \widetilde{\hat{\omega}}^{(I_d)}[d] \right\}$, where $1 \leq I_d \leq L$. Note that for the first iteration, $q = 1$, the tentative list is $\mathcal{S} = \{\emptyset\}$ and only one estimate $\widetilde{\mathcal{W}}[d] = \widetilde{\hat{\omega}}^{(1)}[d]$ is computed in each branch. The D reduced lists $\widetilde{\mathcal{W}}[d]$ are input to the high energy symbol estimators as shown in Fig. 3.

B. *High Energy Symbol Estimation*

The purpose of each high energy symbol estimator is to find the branch list $\mathcal{S}_{br}[d] = \left\{ \hat{\mathrm{s}}_{br}^{(1)}[d], \hat{\mathrm{s}}_{br}^{(2)}[d], \ldots, \hat{\mathrm{s}}_{br}^{(L)}[d] \right\}$ given the reduced list of interfering symbol sets $\widetilde{\mathcal{W}}[d]$.

In the d-th beam of the spatial beam former the symbol sets $\tau[d]$ contain most of the energy. We employ an exhaustive search over all sets $\tau^{(j)}[d]$, $j = 1, 2, \ldots, |\mathcal{A}|^{|\tau[d]|}$. For each set $\tau^{(j)}[d]$ and each interfering symbol set $\widetilde{\hat{\omega}}^{(i)}[d] \in \widetilde{\mathcal{W}}[d]$, $i = 1, 2, \ldots, I_d$, we do an exhaustive search using the error metric

$$e^{(i,j)}[d] = \left| y[d] - \hat{y}^{(i,j)}[d] \right|^2 \quad (8)$$

where $y[d]$ is a component of $\mathbf{y}$ in (4) and the candidate component $\hat{y}^{(i,j)}[d]$ is the sum of an "enumeration component" $\hat{y}_e^{(j)}[d]$ and an "interference component" $\hat{y}_{if}^{(i)}[d]$ defined as

$$\begin{aligned} \hat{y}^{(i,j)}[d] &= \hat{y}_e^{(j)}[d] + \hat{y}_{if}^{(i)}[d] \\ \hat{y}_e^{(j)}[d] &= \sum_{u \in U_e[d]} h_{du} s_u \\ \hat{y}_{if}^{(i)}[d] &= \sum_{u \in \overline{U}_e[d]} h_{du} \hat{s}_u^{(i)}. \end{aligned} \quad (9)$$

Symbol values for $\hat{y}_e^{(j)}[d]$ are drawn from the j-th symbol set $\tau^{(j)}[d]$, whereas symbol values for $\hat{y}_{if}^{(i)}[d]$ are drawn from the i-th list element $\widetilde{\hat{\omega}}^{(i)}[d] \in \widetilde{\mathcal{W}}[d]$. Therefore, using a reduced list $\widetilde{\mathcal{W}}[d]$ of size $I_d \leq L$ significantly decreases the overall number of (i, j) symbol combinations.

The k-th branch symbol vector $\hat{\mathrm{s}}_{br}^{(k)}[d]$ in the d-th branch list $\mathcal{S}_{br}[d]$ is found by drawing symbol values from the (i, j) symbol combination with the k-th smallest error metric

$$(i,j)^{(k)} = \underset{\substack{1 \leq i \leq I_d \\ 1 \leq j \leq |\mathcal{A}|^{|\tau[d]|}}}{\arg\min^{(k)}} \left\{ e^{(i,j)}[d] \right\}, \ k = 1, 2, \ldots, L \quad (10)$$

where $\min^{(k)}$ denotes the k-th smallest value. As shown in Fig. 3, the D branch lists $\mathcal{S}_{br}[d]$ are passed from the high energy symbol estimators to a list combining stage.

C. *List Combining*

In previous detection stages the high and low energy symbol sets of each vector, $\hat{\mathrm{s}}_{br}^{(k)}[d] = \left\{ \widetilde{\hat{\omega}}^{(k)}[d], \hat{\tau}^{(k)}[d] \right\}$, in the d-th branch list $\mathcal{S}_{br}[d]$ have been estimated. The two symbol sets have different probabilities of containing the correct symbols. We combine the high energy symbol sets $\hat{\tau}^{(k)}[d]$ of each branch list $\mathcal{S}_{br}[d]$ to increase the probability of finding the best overall list of symbol vectors, $\mathcal{S}$. The combiner stores the tentative

[4] The lists $\mathcal{S}$ and $\mathcal{S}_d$ are ordered from most to least likely.

symbol vector list $\mathcal{S}$ and the corresponding tentative list of L error metrics, $\mathcal{E} = \left\{e^{(1)}, e^{(2)}, \ldots, e^{(L)}\right\}$. For the q-th iteration, $q < Q$, $\mathcal{S}$ is fed back to the D detector branches. Otherwise, if $q = Q$, $\mathcal{S}$ is output by the PSD-RCIE detector.

We first define D lists of L branch error metrics $\mathcal{E}_{br}[d] = \left\{e_{br}^{(1)}[d], e_{br}^{(2)}[d], \ldots, e_{br}^{(L)}[d]\right\}$ corresponding to $\mathcal{S}_{br}[d]$ and compute the elements

$$e_{br}^{(k)}[d] = \left|\mathbf{y} - \mathbf{H}\hat{\mathbf{s}}_{br}^{(k)}[d]\right|^2 = \sum_{c=1}^{D} \left| y[c] - \sum_{u=1}^{D} h_{cu}\hat{s}_u^{(k)} \right|^2 \quad (11)$$

where $k = 1, 2, \ldots, L$ is the list index and $\hat{\mathbf{s}}_{br}^{(k)}[d] \in \mathcal{S}_{br}[d]$ is the k-th branch symbol vector with symbol values $\hat{s}_u^{(k)}$.

Define the tentative lists of L minimum error metrics $\mathcal{E}_{\min} = \left\{e_{\min}^{(1)}, e_{\min}^{(2)}, \ldots, e_{\min}^{(L)}\right\}$ and corresponding $D \times 1$ symbol vectors $\mathcal{S}_{\min} = \left\{\hat{\mathbf{s}}_{\min}^{(1)}, \hat{\mathbf{s}}_{\min}^{(2)}, \ldots, \hat{\mathbf{s}}_{\min}^{(L)}\right\}$. The metric list $\mathcal{E}_{\min}$ is found by calculating

$$e_{\min}^{(l)} = \min_{\substack{1 \le i,k \le L \\ 1 \le d \le D}}{}^{(l)} \left\{e_{br}^{(k)}[d], e^{(i)}\right\}, \quad l = 1, 2, \ldots, L \quad (12)$$

where $e^{(i)} \in \mathcal{E}$ denotes the i-th element in the tentative list of error metrics from the previous iteration. For $q = 1$, we choose $\mathcal{E} = \{\infty\}$. The tentative list of symbol vectors, $\mathcal{S}_{\min}$, is found by picking corresponding symbol vectors from $\mathcal{S}_{br}[d]$ and $\mathcal{S}$. The list $\mathcal{S}_{\min}$ contains the improved estimates and $\mathcal{E}_{\min}$ holds the corresponding error metrics. Both lists are copied into the tentative lists $\mathcal{S}$ and $\mathcal{E}$, respectively, as initial estimates of these quantities in the q-th iteration. We next find D lists of non-redundant high energy symbol sets, $\widetilde{\mathcal{T}}[d] = \left\{\widetilde{\tau}^{(1)}[d], \widetilde{\tau}^{(2)}[d], \ldots, \widetilde{\tau}^{(J_d)}[d]\right\}$. The list elements $\widetilde{\tau}^{(j)}[d]$, $1 \le j \le J_d$, are copied from the corresponding branch list $\mathcal{S}_{br}[d]$. Hence, each list $\widetilde{\mathcal{T}}[d]$ has size $1 \le J_d \le L$. We can now describe the iterative list combining algorithm as shown in Table 1, which takes as inputs the tentative lists $\mathcal{S}$, $\mathcal{E}$ and the lists $\widetilde{\mathcal{T}}[d]$. It typically performs $Q_{lc} = 2$ or 3 iterations.

After list combining, the improved lists of most likely symbol vectors $\mathcal{S}$ and corresponding error metrics $\mathcal{E}$ are stored by the list combiner. For the q-th overall iteration, $q < Q$, $\mathcal{S}$ is fed back to the D detector branches. Otherwise, for $q = Q$, $\mathcal{S}$ is output by the detector and passed to the decision device.

IV. Simulation Results

We compare performance of the PSD-RCIE, JML [4] and SRSJD [5] algorithms using simulation results. We assume that D users transmit QPSK signals and that the receiver has perfect channel knowledge. The signal-to-noise ratio (SNR) is defined as the ratio of signal and noise variances at each receive antenna, namely $SNR = 10\log_{10}\left(\sigma_s^2/\sigma_z^2\right)$, where σ_s^2 denotes the average received signal power. Results are shown as the symbol error rate (SER) of the worst user versus the number of users for different SNRs. Simulations were stopped after one user experienced 50 errors. Fig. 5 presents the SER for a receiver with a $M = 5$ element circular antenna array at $SNR = 5dB$ and $10dB$. The array radius is $R_a = 0.2\lambda$, where λ is the wavelength of the received signals. The users are equally spaced around the antenna array. Note that this causes performance degradation for even numbers of users due to signals impinging from opposite AOAs. The user signals are assumed to be symbol and phase synchronous with equal power. For the SRSJD and PSD-RCIE algorithms, the DEIR was chosen to give the same sparsity matrices $\mathbf{P}$ with $|\tau[d]| = 3$ high energy symbols in each spatial beam. The ITB-DDFSE algorithm employed by SRSJD and PSD-RCIE performs $Q_{itb} = 2$ trellis iterations. The PSD-RCIE detector is set to do $Q_{lc} = 2$ list combining and $Q = 2$ overall iterations. As seen in Fig. 5, for a list $\mathcal{S}$ of size $L = 2D$, PSD-RCIE approximates JML very well and clearly outperforms both, the linear beam former of [5] and the SRSJD detector. Furthermore, for $D > 8$ users PSD-RCIE still approximates JML whereas SRSJD fails.

Table 1: Iterative List Combining Algorithm

1. Allocate a list of L candidate symbol vectors of size $D \times 1$, $\mathcal{S}_{\text{cand}} = \left\{\hat{\mathbf{s}}_{\text{cand}}^{(1)}, \hat{\mathbf{s}}_{\text{cand}}^{(2)}, \ldots, \hat{\mathbf{s}}_{\text{cand}}^{(L)}\right\}$, and a list of corresponding candidate error metrics, $\mathcal{E}_{\text{cand}} = \left\{e_{\text{cand}}^{(1)}, e_{\text{cand}}^{(2)}, \ldots, e_{\text{cand}}^{(L)}\right\}$.
2. For each iteration $q_{lc} = 1, 2, \ldots, Q_{lc}$ and all elements $\widetilde{\tau}^{(j)}[d]$, $j = 1, 2, \ldots, J_d$, of the $d = 1, 2, \ldots, D$ high energy symbol set lists $\widetilde{\mathcal{T}}[d]$
   - Copy the tentative list $\mathcal{S}$ into the candidate list $\mathcal{S}_{\text{cand}}$.
   - For each list element $\hat{\mathbf{s}}_{\text{cand}}^{(k)} \in \mathcal{S}_{\text{cand}}$ and $k = 1, 2, \ldots, L$ do
     - Copy the high energy symbol set $\widetilde{\tau}^{(j)}[d]$ into $\hat{\mathbf{s}}_{\text{cand}}^{(k)}$.
     - Compute the error metric, $e_{\text{cand}}^{(k)} = \left|\mathbf{y} - \mathbf{H}\hat{\mathbf{s}}_{\text{cand}}^{(k)}\right|^2$.
   - Update the tentative list of minimum error metrics, $\mathcal{E}_{\min}$, by finding the l smallest metrics,
     $$e_{\min}^{(l)} = \min_{1 \le i,k \le L}{}^{(l)} \left\{e_{\text{cand}}^{(k)}, e^{(i)}\right\}, \quad l = 1, 2, \ldots, L$$
     where $e^{(i)} \in \mathcal{E}$ is the i-th tentative error metric in the list $\mathcal{E}$.
   - Update the corresponding list of symbol vectors, $\mathcal{S}_{\min}$, by picking the $l = 1, 2, \ldots, L$ symbol vectors from $\mathcal{S}_{\text{cand}}$ and $\mathcal{S}$ with minimum error metric $e_{\min}^{(l)}$.
   - Set $\mathcal{S}_{\min} = \mathcal{S}$ and $\mathcal{E}_{\min} = \mathcal{E}$.
3. Terminate the list combining algorithm. Set $q = q + 1$.

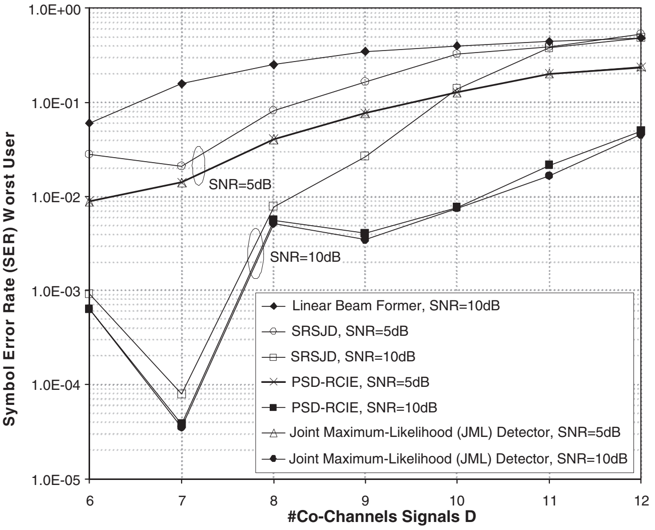


Figure 5: SER performance of the worst user for a $M = 5$ antenna array using list size $L = 2D$ for PSD-RCIE.

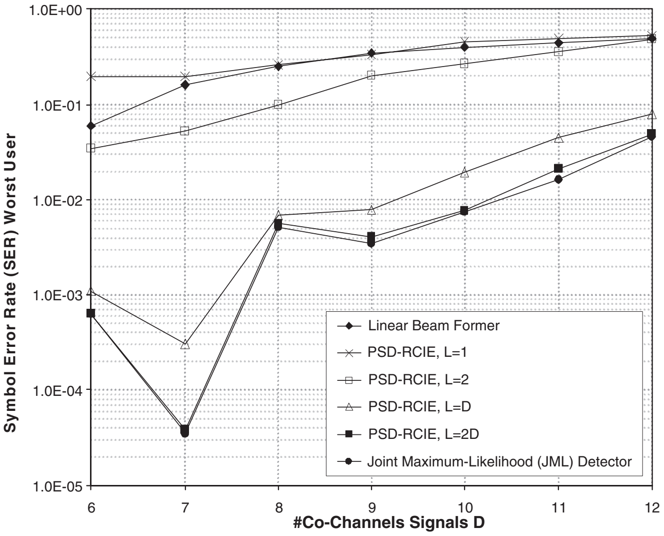


Figure 6: PSD-RCIE SER performance of the worst user for a $M = 5$ antenna array at $SNR = 10dB$ and various sizes of the tentative symbol vector list.

Table 2: Comparison of computational complexity for a receiver with $M = 5$ antenna array at $SNR = 10dB$

D	JML	SRSJD	PSD-RCIE		
			$L = 2$	$L = D$	$L = 2D$
6	$4.10E4$	$1.54E3$	$1.08E4$	$2.95E4$	$6.87E4$
7	$1.64E5$	$1.79E3$	$1.83E4$	$5.78E4$	$1.34E5$
8	$6.55E5$	$2.05E3$	$2.75E4$	$9.94E4$	$2.29E5$
9	$2.62E6$	$2.30E3$	$3.83E4$	$1.57E5$	$3.63E5$
10	$1.05E7$	$2.56E3$	$5.08E4$	$2.34E5$	$5.44E5$
11	$4.19E7$	$2.82E3$	$6.48E4$	$3.32E5$	$7.78E5$
12	$1.68E8$	$3.07E3$	$8.05E4$	$4.56E5$	$1.08E6$

Similarly, at $SNR = 5dB$, PSD-RCIE has almost the same performance as JML and outperforms SRSJD detection.

Fig. 6 depicts the SERs of PSD-RCIE for various sizes of the list $\mathcal{S}$. It can be seen that a list $\mathcal{S}$ of size $D \leq L \leq 2D$ obtains good performance whereas for $L = 1, 2$ results are poor. Note that increasing L also increases the complexity of list combining and low energy user interference estimation.

V. Complexity

The complexity of the PSD-RCIE algorithm depends on several parameters. Among these are the number of users D, the alphabet size $|\mathcal{A}|$, the number of iterations Q_{itb}, Q_{lc} and Q, the number of high energy user symbols $|\tau[d]|$, and the sizes of the lists $\widetilde{\mathcal{W}}[d]$, $\mathcal{S}_{br}[d]$, $\widetilde{\mathcal{T}}[d]$ and $\mathcal{S}$.

Since the number of real squaring operations in the Euclidean error metric computation is usually the most hardware intensive operation, we use it to evaluate complexity. We present average complexities for the simulations of Section IV. Table 2 shows the number of real squaring operations per estimated output symbol vector for the JML, SRSJD and PSD-RCIE algorithms at $SNR = 10dB$. Note that the PSD-RCIE values represent the complexity of estimating an ordered list of symbol vectors from which the best vector is chosen as output. PSD-RCIE achieves complexity savings of up to several orders of magnitude over JML detection for a large number of users or a small list size.

PSD-RCIE has higher overall complexity than SRSJD. This is due to multiple application of the ITB-DDFSE algorithm in the low energy interference estimators whereas SRSJD employs ITB-DDFSE only once. The advantages of PSD-RCIE are that it approximates JML much better than SRSJD and that the decoder design allows parallel implementation.

VI. Conclusion

A novel MUD algorithm for use in overloaded receivers with a circular antenna array is presented. The detection process is divided into two stages: linear preprocessing and PSD-RCIE. The linear preprocessor is the spatial filter of [5], and uses the spatial separation of users to mitigate CCI. PSD-RCIE is a list type iterative detection algorithm. Symbol detection is split into D parallel branches, one for each user. In each branch, residual CCI is estimated using the ITB-DDFSE algorithm [5]. Using these estimates, high energy symbol estimators compute D branch lists of symbol vectors through an exhaustive search over high energy symbols. The vectors in the branch lists are then combined to find the list of symbol vectors with minimum Euclidean error metric. This list is fed back to obtain improved estimates.

PSD-RCIE is shown to estimate the true JML symbol vector with high probability after only two or three iterations. Thus, it outperforms other detection algorithms such as those in [5]. PSD-RCIE complexity is evaluated in terms of the average number of real squaring operations. Simulations show that it can achieve complexity savings of up to several orders of magnitude compared to JML, but it has higher complexity than that of [5]. The parallel detector structure enables parallel processing which may compensate for the higher complexity.

References

[1] S. J. Grant and J. K. Cavers, "Performance enhancement through joint detection of cochannel signals using diversity arrays," *IEEE Trans. Commun.,* vol. 46, pp. 1038-1049, Aug. 1998.

[2] S. Talwar, M. Viberg and A. Paulraj, "Blind separation of synchronous cochannel digital signals using an antenna array–Part I: Algorithms," *IEEE Trans. Signal Processing,* vol. 44., pp. 1184-1197, May 1996.

[3] S. Talwar and A. Paulraj, "Blind separation of synchronous co-channel digital signals using an antenna array–Part II: Performance Analysis," *IEEE Trans. Signal Processing,* vol. 45., pp. 706-718, March 1997.

[4] S. Bayram, J. Hicks, R. J. Boyle and J. H. Reed, "Joint ML approach in overloaded array processing," in *Proc. VTC,* Boston, MA, vol. 1., pp. 394-400, Sept. 2000.

[5] J. Hicks, S. Bayram, W. H. Tranter, R. J. Boyle and J. H. Reed, "Overloaded array processing with spatially reduced search joint detection," *IEEE J. Select. Areas Commun.,* vol. 19, pp. 1584-1593, Aug. 2001.

[6] A. Duel-Hallen and C. Heegard, "Delayed decision-feedback sequence estimation," *IEEE Trans. Commun.,* vol. 37, pp. 428-436, May 1989.

[7] G. Forney, "Maximum-Likelihood sequence estimation of digital sequences in the presence of intersymbol interference", *IEEE Trans. Inform. Theory,* vol. 18, pp. 363-378, May 1972.

[8] M. J. Colella, J. N. Martin and F. Akyildiz, "The halo network", *IEEE Commun. Mag.,* vol. 38, pp. 142-148, June 2000.

[9] S. Bayram, J. Hicks, R. J. Boyle and J. H. Reed, "Overloaded array processing in wireless airborne communication systems," in *Proc. MILCOM,* pp. 24-29, Los Angeles, CA, Oct. 2000.