

# LIST DETECTION FOR OVERLOADED RECEIVERS WITH A LINEAR ARRAY

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

*We consider an overloaded wireless multiuser communication system where the number of transmitters exceeds the number of receive antennas. Each user transmits data through a single antenna to a receiver with an  $M$ -element uniform linear array. The received signals are combined using maximum ratio diversity combining. Severe co-channel interference (CCI) occurs in such a system due to multiple co-channel users. We present an iterative parallel multiuser detector with list feedback of the best estimates to separate and detect the user symbols. CCI is estimated in the detector using a novel list-based parallel interference cancellation scheme. Simulations show that our algorithm approximates joint-maximum likelihood detection at lower complexity and is well suited for practical application.*

## 1. INTRODUCTION

Space-time processing allows the capacity and reliability of data transfer in wireless communication systems to be increased. This is achieved by exploiting spatial diversity using multiple antennas at the receiver. Diversity techniques combine multiple replicas of the same information-bearing signal to improve performance. Maximum ratio combining (MRC) is a classical diversity technique that maximizes the instantaneous signal-to-noise ratio (SNR) at the combiner output [1]. It requires knowledge of the channel coefficients at the receiver and is an effective tool to combat the effects of severe multipath fading.

An overloaded wireless communication system has more transmitters than receive antennas. Multiple co-channel users cause severe co-channel interference (CCI). As a result, the demodulation of the signals of interest at the receiver can be challenging.

Signal separation in overloaded receivers was first studied by Grant and Cavers. In [2], they showed that the extraction of the signals of interest is possible using a complex joint maximum a posteriori algorithm. Talwar et. al. [3, 4] analyzed maximum likelihood ap-

proaches for estimating synchronous co-channel signals impinging on an antenna array in non-overloaded receivers. Bayram et. al. [5] extended this to the overloaded case and showed that joint-maximum likelihood (JML) detection is optimal. JML uses an exhaustive search over all possible symbol combinations, making it impractical for most real-world applications. Hicks et. al. [6] proposed a reduced complexity algorithm that approximates the JML detector for receivers with an uniform circular array (UCA). Their spatially reduced search joint detection (SRSJD) algorithm combines a linear spatial pre-filter with a non-linear reduced search algorithm. The pre-filter is a linear beam former which mitigates CCI. Complexity savings are achieved by only searching over high energy symbols in each spatial beam of the pre-filter. The search algorithm is a spatial adaptation of the Viterbi algorithm [7], known as iterative tail-biting delayed decision feedback sequence estimation (ITB-DDFSE). It uses a search trellis and is thus restricted to certain array geometries. Unfortunately, receivers employing other array geometries, e.g. the uniform linear array (ULA), do not necessarily have trellis oriented channel matrices.

In [8], we proposed an iterative list-based multiuser detection (MUD) algorithm for use with circular arrays. The algorithm, named parallel symbol detection with reduced complexity interference estimation (PSD-RCIE), uses the preprocessor of [6] and computes lists of most likely symbol estimates. It approximates JML detection much better than SRSJD. The computational complexity of PSD-RCIE is low compared to JML but higher than SRSJD.

In this paper we extend the work of [8] to overloaded receivers with an ULA. Unlike [8], which uses a linear beam former as a preprocessor and is restricted to circular antenna arrays, our detector employs MRC as a preprocessor. Symbol detection differs from [8] in that we use a list-based parallel interference cancellation scheme for CCI estimation. We name the resulting algorithm parallel symbol detection with parallel inter-

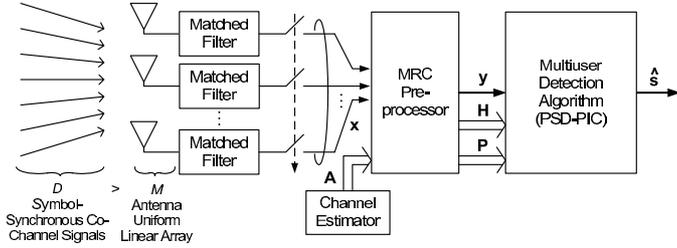


Figure 1: Proposed system model.

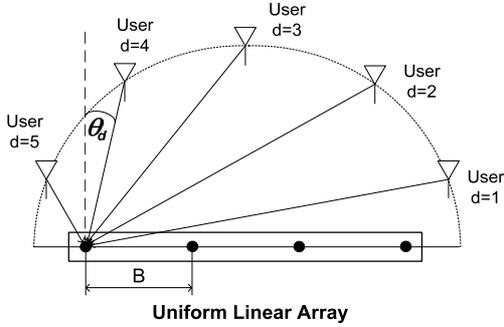


Figure 2: ULA with  $M = 4$  receive antennas and  $D = 5$  single antenna users.

ference cancellation (PSD-PIC). Simulations show that it approximates JML with lower complexity and that it provides an excellent complexity-performance trade-off.

In Section 2, the system model is introduced. Section 3 explains the preprocessor. The detection algorithm is described in Section 4 and simulation results are shown in Section 5. Section 6 considers computational complexity. Conclusions are drawn in Section 7.

## 2. SYSTEM MODEL

We assume  $D$  co-channel QAM signals impinging on a receiver with an  $M$ -element ULA. The receiver is overloaded with overload factor  $f = D/M$ , where  $D > M$ . The system model is shown in Fig. 1. At each antenna, the received signal is passed through a filter matched to the transmitted pulse shape. The received  $M \times 1$  signal vector  $\mathbf{x}$  is input to the MRC preprocessor. For simplicity, we restrict consideration to the symbol-synchronous case with no intersymbol interference present in the channel. The signal model is given as

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{z} \quad (1)$$

where  $\mathbf{s} = [s_1 \ s_2 \ \dots \ s_D]^T$  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 vector  $\mathbf{a}(\theta_d)$  is the steering vector for the  $d$ -th user's signal. Fig. 2 shows a model of the ULA signal envi-

ronment. For an  $M$ -element ULA, the array steering vector  $\mathbf{a}(\theta_d) = [a_1 \ a_2 \ \dots \ a_M]^T$  is given by

$$a_m = \exp\left(-j2\pi \frac{(m-1)B}{\lambda} \sin(\theta_d)\right) \quad (2)$$

where

- $\theta_d$  azimuthal angle of arrival (AOA)
- $B$  spacing of the antenna elements
- $\lambda$  wavelength of the carrier frequency
- $m$  antenna element index,  $m = 1 \dots M$  [9].

Note that for simplicity, only azimuth angles are considered. The extension to three dimensions is straightforward. Finally, the quantity  $\mathbf{z}$  in (1) represents an  $M \times 1$  temporally uncorrelated noise vector with zero mean and autocorrelation  $\mathbf{\Phi}_{zz} = E[\mathbf{z}\mathbf{z}^H]$ , where  $E[\cdot]$  is the expectation operator and  $(\cdot)^H$  denotes Hermitian transpose. For spatially uncorrelated noise,  $\mathbf{\Phi}_{zz} = \sigma_z^2 \mathbf{I}$ , where  $\sigma_z^2$  denotes the noise variance and  $\mathbf{I}$  is the  $M \times M$  identity matrix. Throughout this paper, any time dependence in equations is dropped for convenience. We further assume that perfect channel information is available at the receiver but not at the transmitters.

## 3. MRC PREPROCESSOR

MRC is a diversity technique where the output is a linear combination of the weighted signal replicas. Hence, the MRC combiner output is written as

$$\mathbf{y} = \mathbf{W}^H \mathbf{x} \quad (3)$$

where  $\mathbf{y}$  is the  $D \times 1$  output vector and  $\mathbf{W}^H$  are the  $D \times M$  complex optimum weights. When the receiver has perfect channel state information,  $\mathbf{W} = \mathbf{A}$ . Towards this end, we define a  $D \times D$  matrix  $\mathbf{H}$  such that

$$\mathbf{H} = \mathbf{A}^H \mathbf{A}. \quad (4)$$

Using (4), the JML receiver can be written as

$$\begin{aligned} \hat{\mathbf{s}} &= \arg \min_{\mathbf{s} \in \mathcal{A}^D} \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2 \\ &= \arg \min_{\mathbf{s} \in \mathcal{A}^D} \sum_{d=1}^D |y[d] - \mathbf{h}[d]\mathbf{s}|^2 \end{aligned} \quad (5)$$

where  $\mathbf{h}[d]$  is the  $d$ -th row of  $\mathbf{H}$  and  $y[d]$  is the  $d$ -th element of  $\mathbf{y}$ . Note that (5) appears similar to Eqn. (6) in [6] with the difference being the definitions of  $\mathbf{y}$  and  $\mathbf{H}$ . This is because here we consider an ULA with element spacing in the order of the wavelength  $\lambda$  which makes MRC more suited than a linear beam former at the preprocessing stage. MRC maximizes the SNR at the output by combining multiple signal replicas whereas the linear beam former of [6] places a beam

in the direction of each user, thereby mitigating CCI. Both preprocessors focus the signal energy to only a few elements in each row of  $\mathbf{H}$ . This suggests that reduced complexity algorithms, such as PSD-RCIE [8], can be used for the detection of the user symbols.

We define enumeration sets,  $U_e[d]$ , that contain the column indices of the high energy symbols in the  $d$ -th row of  $\mathbf{H}$ . The high and low energy user symbols are distinguished by choosing the column indices  $u \in U_e[d]$  of the matrix elements with the highest energy

$$U_e[d] = \left\{ \arg \max_{1 \leq u \leq D}^{(\nu)} |h_{du}|^2 \right\}, \quad 1 \leq \nu \leq \mu[d] \quad (6)$$

where  $\max^{(\nu)}$  denotes the  $\nu$ -th greatest value and  $\mu[d]$  is the number of column indices considered for  $U_e[d]$  in the  $d$ -th row of  $\mathbf{H}$ . The quantity  $h_{du}$  denotes the matrix element of  $\mathbf{H}$  at position  $(d, u)$ . Note that the choice of  $\mu[d]$  affects both performance and complexity of the subsequent symbol detector. We wish to choose  $\mu[d]$  such that the column indices of the  $d$ -th user and all other users with high energy are contained in  $U_e[d]$ . The optimum choice of  $U_e[d]$  may be difficult to find. We use the strongest energy to interference ratio (SEIR) for each row of  $\mathbf{H}$  as a measurement of the energy contribution from low energy user signals. We define the SEIR as

$$\begin{aligned} SEIR[d] &= \frac{E \left[ \max_{1 \leq u \leq D} |h_{du}s_u|^2 \right]}{E \left[ \sum_{v \in \bar{U}_e[d]} |h_{dv}s_v|^2 \right]} \\ &= \frac{\max_{1 \leq u \leq D} |h_{du}|^2}{\sum_{v \in \bar{U}_e[d]} |h_{dv}|^2} \end{aligned} \quad (7)$$

where the numerator is the signal power of the strongest user in the  $d$ -th row of  $\mathbf{H}$  and the denominator is the power of the signals outside the enumeration set  $U_e[d]$ . If a computed SEIR value is below a predefined threshold, the parameter  $\mu[d]$  is increased to ensure good performance. Additionally, we must guarantee that  $\mu[d] \leq \mu_{\max}$  with  $1 \leq \mu_{\max} \leq D$ . The parameter  $\mu_{\max}$  denotes the maximum number of high energy users and determines the upper bound on computational complexity in subsequent detection stages. Due to the restriction on the size of  $\mu[d]$ , the SEIR may be below the threshold with no “strong” users in that row.

Using  $\mathbf{H}$  and  $U_e[d]$ , we construct a  $D \times D$  sparsity matrix,  $\mathbf{P}$  which contains zeros and ones in positions corresponding to elements with low energy and high energy in  $\mathbf{H}$ , respectively. We find the sets of high and low energy symbols,  $\tau[d]$  and  $\omega[d]$ , respectively, as [8]

$$\tau[d] = \{s_u | u \in U_e[d]\}, \quad \omega[d] = \{s_u | u \in \bar{U}_e[d]\}. \quad (8)$$

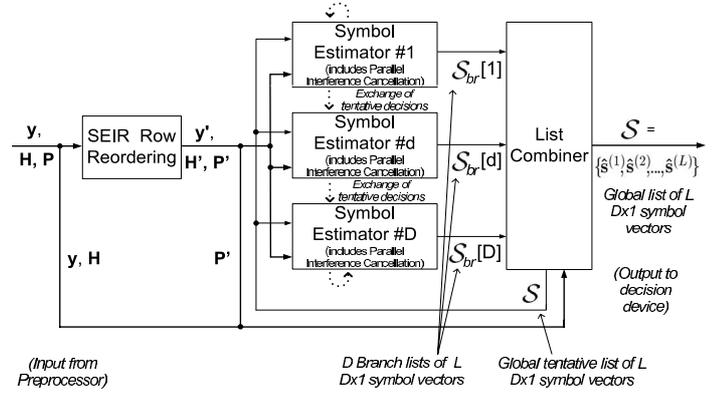


Figure 3: Block diagram of the PSD-PIC detector.

#### 4. DETECTION ALGORITHM

In this section, we describe the proposed PSD-PIC detection algorithm, shown in Fig. 3. It takes the MRC output vector  $\mathbf{y}$ , the matrix  $\mathbf{H}$  and the sparsity matrix  $\mathbf{P}$  as inputs. The rows of  $\mathbf{y}$ ,  $\mathbf{H}$  and  $\mathbf{P}$  are reordered<sup>1</sup> from largest to smallest SEIR value and stored in  $\mathbf{y}'$ ,  $\mathbf{H}'$  and  $\mathbf{P}'$ , respectively. After iterative processing, a list of symbol vectors  $\mathcal{S} = \{\hat{\mathbf{s}}^{(1)}, \hat{\mathbf{s}}^{(2)}, \dots, \hat{\mathbf{s}}^{(L)}\}$  is output, where  $\hat{\mathbf{s}}^{(l)}$  is the  $l$ -th  $D \times 1$  symbol vector in the list. The iterative detection process is divided into two stages: symbol estimation with parallel interference cancellation of low energy users and list combining.

The detector has  $D$  independent parallel detection branches such that the  $d$ -th element in  $\mathbf{y}'$  corresponds to the  $d$ -th branch. In each iteration and branch, a branch list  $\mathcal{S}_{br}[d]$  containing  $L$  ( $D \times 1$ ) symbol vectors is computed. The  $d$ -th branch list is given by  $\mathcal{S}_{br}[d] = \{\hat{\mathbf{s}}_{br}^{(1)}[d], \hat{\mathbf{s}}_{br}^{(2)}[d], \dots, \hat{\mathbf{s}}_{br}^{(L)}[d]\}$ , where  $\hat{\mathbf{s}}_{br}^{(k)}[d]$  is the  $k$ -th symbol vector in the  $d$ -th branch list,  $k = 1, 2, \dots, L$  and  $d = 1, 2, \dots, D$ . Each branch symbol vector  $\hat{\mathbf{s}}_{br}^{(k)}[d]$  contains estimates of the high and low energy symbol sets  $\tau[d]$  and  $\omega[d]$ , respectively. Hence,  $\hat{\mathbf{s}}_{br}^{(k)}[d]$  can be decomposed into estimates of the low and high energy user symbol sets  $\hat{\omega}^{(k)}[d]$  and  $\hat{\tau}^{(k)}[d]$  such that

$$\hat{\mathbf{s}}_{br}^{(k)}[d] = \hat{\tau}^{(k)}[d] \cap \hat{\omega}^{(k)}[d]. \quad (9)$$

The branch symbol vectors are computed iteratively using the symbol estimators shown in Fig. 3. Each symbol estimator searches over all possible combinations of high energy symbol sets  $\tau[d]$ . The  $k = 1, 2, \dots, L$  best estimates  $\hat{\tau}^{(k)}[d]$  are used to update the  $d$ -th branch list  $\mathcal{S}_{br}[d]$ . This list is stored in the  $d$ -th symbol estimator and is passed to the  $(d+1)$ -th symbol estimator in the next iteration. Symbol values from the input

<sup>1</sup>Reordering is beneficial for detection performance due to the employment of interference cancellation techniques in subsequent detection stages.

list are used as estimates of the low energy user interference. After a sufficient number of iterations for interference cancellation, typically  $Q_{pic} = 2$  or  $3$ , the  $D$  branch lists  $\mathcal{S}_{br}[d]$  are output by the symbol estimators and input to a list combiner. The list combiner searches and combines the input list elements to find the ordered<sup>2</sup> global tentative list<sup>3</sup> of most likely symbol vectors,  $\mathcal{S} = \{\hat{\mathbf{s}}^{(1)}, \hat{\mathbf{s}}^{(2)}, \dots, \hat{\mathbf{s}}^{(L)}\}$ . This is stored and fed back to the  $D$  detector branches for the next iteration. At the end of the  $Q$ -th iteration, the list  $\mathcal{S}$  is output by the detector as an estimate of the ordered global list of most likely symbol vectors. Typically,  $Q = 2$  or  $3$  overall iterations are necessary to obtain good results. A decision device selects the first element  $\hat{\mathbf{s}}^{(1)} \in \mathcal{S}$  as the best estimate. Alternatively, the output list  $\mathcal{S}$  can be used to calculate soft information for subsequent stages such as error control decoders.

#### 4.1. Symbol Estimation and PIC

We now consider the symbol estimation stage in the proposed PSD-PIC detector. Overloaded receivers suffer from severe CCI in the received signal, and pre-processors employing linear operations such as MRC and spatial pre-filtering cannot totally cancel it. The amount of CCI increases with the receiver overload factor  $f$ . We consider all symbols contained in the low energy symbol sets  $\omega[d]$  as residual CCI. Symbol detection is based on the idea that if the exact amount of CCI is known to the receiver, it can be removed from the received signal. Therefore, if the CCI from low energy users is removed, the probability of correct detection of the high energy user symbols in subsequent detection stages is high. We employ an iterative parallel interference cancellation algorithm for CCI estimation, as shown in Fig. 4. The algorithm takes the quantities  $\mathbf{y}'$ ,  $\mathbf{H}'$  and  $\mathbf{P}'$  as inputs. After iterative processing for  $Q_{pic}$  iterations, it outputs the  $D$  branch lists  $\mathcal{S}_{br}[d]$ .

In each iteration  $q_{pic}$ ,  $q_{pic} = 1, 2, \dots, Q_{pic}$ , the  $D$  branch lists  $\mathcal{S}_{br}[d]$  are updated by estimating the high energy symbol sets  $\tau[d]$ . This is done by the  $D$  parallel high energy symbol estimators shown in Fig. 4. Each updated list  $\mathcal{S}_{br}[d]$  is stored and serves as input to the  $(d+1)$ -th high energy symbol estimator during the  $(q_{pic}+1)$ -th iteration. Note that during the initial iteration,  $q_{pic} = 1$ , we choose the global tentative list  $\mathcal{S}$  as input to all symbol estimators (see Fig. 3). Each input list contains  $L$  ( $D \times 1$ ) symbol vectors.

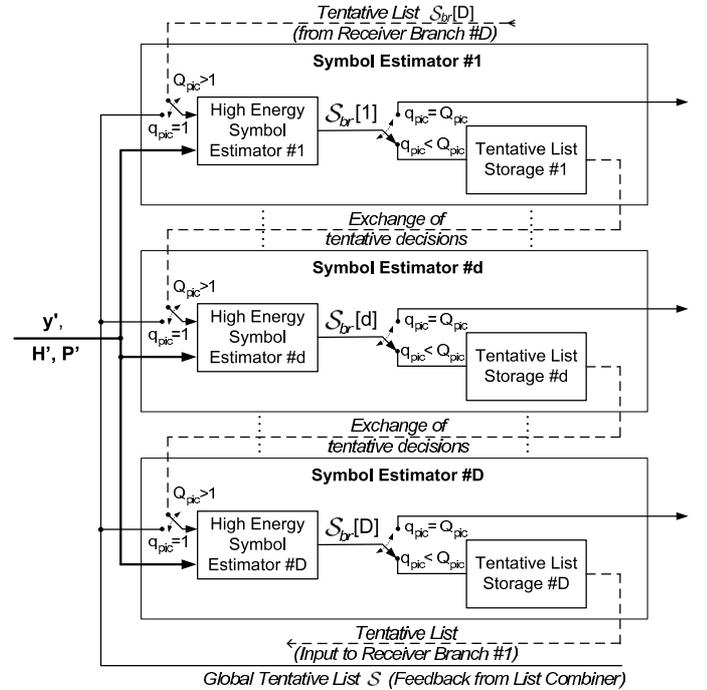


Figure 4: Block diagram of the symbol estimation stage using parallel interference cancellation.

We apply the sparsity matrix  $\mathbf{P}'$  to the input lists to find  $D$  lists of low energy symbol set estimates  $\{\hat{\omega}^{(1)}[d], \hat{\omega}^{(2)}[d], \dots, \hat{\omega}^{(L)}[d]\}$ . Note that each list often contains one or more identical symbol sets  $\hat{\omega}^{(1)}[d] = \hat{\omega}^{(2)}[d] = \dots = \hat{\omega}^{(i)}[d]$  with  $1 \leq i \leq L$ . We can therefore remove any redundant sets to find the  $D$  reduced<sup>4</sup> lists of low energy symbol set estimates  $\widetilde{\mathcal{W}}[d] = \{\tilde{\omega}^{(1)}[d], \tilde{\omega}^{(2)}[d], \dots, \tilde{\omega}^{(I_d)}[d]\}$ , where  $I_d$  denotes list size with  $1 \leq I_d \leq L$ . The list elements  $\tilde{\omega}^{(i)}[d] \in \widetilde{\mathcal{W}}[d]$  contain the current estimates of the CCI in the  $d$ -th symbol estimator. Using these estimates, the  $d$ -th high energy symbol estimator performs an exhaustive search over all sets  $\tau^{(j)}[d], j = 1, 2, \dots, |\mathcal{A}|^{|\tau[d]|}$  and finds the improved list of  $L$  branch symbol vectors  $\hat{\mathbf{s}}_{br}^{(k)}[d] \in \mathcal{S}_{br}[d]$  with lowest Euclidean error metric. We define the error metric as

$$e^{(i,j)}[d] = \left| y'[d] - \hat{y}'^{(i,j)}[d] \right|^2 \quad (10)$$

where  $y'[d]$  is the  $d$ -th component of  $\mathbf{y}'$  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”  $\hat{y}'_{if}^{(i)}[d]$  given as

$$\hat{y}'^{(i,j)}[d] = \hat{y}'_e^{(j)}[d] + \hat{y}'_{if}^{(i)}[d]$$

<sup>2</sup>The lists  $\mathcal{S}_{br}[d]$  and  $\mathcal{S}$  are ordered from most to least likely.

<sup>3</sup>The list  $\mathcal{S}$  is initialized with random values chosen from the alphabet  $\mathcal{A}$  at the beginning of the detection process.

<sup>4</sup>List reduction reduces the complexity of high energy symbol estimation because the estimates of the low energy symbol sets are combined with an exhaustive search over high energy symbols.

$$\begin{aligned}\hat{y}'_e^{(j)}[d] &= \sum_{u \in U_e[d]} h'_{du} s_u \\ \hat{y}'_{if}^{(i)}[d] &= \sum_{u \in \bar{U}_e[d]} h'_{du} \hat{s}_u^{(i)}\end{aligned}\quad (11)$$

where  $h'_{du}$  is a component of  $\mathbf{H}'$ . Symbol values  $s_u$  for  $\hat{y}'_e^{(j)}[d]$  are drawn from the  $j$ -th symbol set  $\tau^{(j)}[d]$ , whereas symbol values  $\hat{s}_u^{(i)}$  for  $\hat{y}'_{if}^{(i)}[d]$  are drawn from the  $i$ -th list element  $\tilde{\omega}^{(i)}[d] \in \tilde{\mathcal{W}}[d]$ . We find the estimates  $\hat{\mathbf{s}}_{br}^{(k)}[d] \in \mathcal{S}_{br}[d]$ ,  $k = 1, 2, \dots, L$ , by drawing symbol values from the  $(i, j)$  symbol combination with the  $k$ -th smallest error metric

$$(i, j)^{(k)} = \arg \min_{\substack{1 \leq i \leq I_d \\ 1 \leq j \leq |\mathcal{A}|^{\tau[d]}}} \left\{ e^{(i, j)}[d] \right\}, \quad k = 1, 2, \dots, L. \quad (12)$$

where  $\min^{(k)}$  denotes the  $k$ -th smallest value. Each list  $\mathcal{S}_{br}[d]$  is stored in the corresponding symbol estimator for the next iteration,  $q_{pic} \leq Q_{pic}$ . After a sufficient number of iterations, typically  $Q_{pic} = 2$  to  $5$ , the  $D$  branch lists  $\mathcal{S}_{br}[d]$  are output by the symbol estimators as estimates of the user symbols in the  $d$ -th branch.

## 4.2. List Combining

The symbols in each branch vector  $\hat{\mathbf{s}}_{br}[d] \in \mathcal{S}_{br}[d]$  contain estimates of the low and high energy symbol sets  $\omega[d]$  and  $\tau[d]$ , respectively. Instead of a complete search as in (5), the symbol sets are estimated using the error metric (10) and iterative parallel interference cancellation. As a result, the symbol vector  $\hat{\mathbf{s}}$  that satisfies (5) may not be included in the  $D$  branch lists  $\mathcal{S}_{br}[d]$ . However, by searching and combining the previously estimated symbol sets, we may find improved estimates of  $\mathbf{s}$ . We therefore employ the list combining algorithm of [8] to find the global list  $\mathcal{S}$  of the  $L$  most likely symbol estimates  $\hat{\mathbf{s}}^{(l)} \in \mathcal{S}$ ,  $l = 1, 2, \dots, L$ .

The list combiner, shown in Fig. 3, produces the global tentative list  $\mathcal{S}$  and the corresponding tentative list of error metrics  $\mathcal{E} = \{e^{(1)}, e^{(2)}, \dots, e^{(L)}\}$ . In the  $q$ -th overall iteration,  $q < Q$ ,  $\mathcal{S}$  is fed back to the  $D$  detector branches. If  $q = Q$ ,  $\mathcal{S}$  is output by the detector.

We combine the  $D$  lists of branch symbol vectors,  $\mathcal{S}_{br}[d]$ , so as to produce a non-redundant list of  $(D \times 1)$  branch symbol vectors,  $\tilde{\mathcal{S}}_{br} = \{\tilde{\mathbf{s}}_{br}^{(1)}, \tilde{\mathbf{s}}_{br}^{(2)}, \dots, \tilde{\mathbf{s}}_{br}^{(K)}\}$ . The corresponding list of error metrics is defined by  $\mathcal{E}_{br} = \{e_{br}^{(1)}, e_{br}^{(2)}, \dots, e_{br}^{(K)}\}$ , where the list size  $K$  is bounded by  $1 \leq K \leq LD$ . The list elements  $\tilde{\mathbf{s}}_{br}^{(k)} \in \tilde{\mathcal{S}}_{br}$  are found by copying the non-redundant symbol vectors

Table 1: Iterative List Combining Algorithm

1. Allocate a list of  $L$  non-redundant  $D \times 1$  candidate symbol vectors,  $\tilde{\mathcal{S}}_{cand} = \{\tilde{\mathbf{s}}_{cand}^{(1)}, \tilde{\mathbf{s}}_{cand}^{(2)}, \dots, \tilde{\mathbf{s}}_{cand}^{(L)}\}$ , and a list of corresponding error metrics,  $\mathcal{E}_{cand} = \{e_{cand}^{(1)}, e_{cand}^{(2)}, \dots, e_{cand}^{(L)}\}$ .
2. For each iteration  $q_{lc} = 1, 2, \dots, Q_{lc}$  and all  $j = 1, 2, \dots, J_d$  elements  $\tilde{\tau}^{(j)}[d]$  of the  $d = 1, 2, \dots, D$  high energy symbol set lists  $\tilde{\mathcal{T}}[d]$ ,
  - Using  $\mathbf{P}'$ , find the symbols that, for the  $d$ -th list  $\tilde{\mathcal{T}}[d]$ , are the estimates of the low energy symbol sets  $\omega[d]$  in the global tentative list  $\mathcal{S}$ . Copy the  $k = 1, 2, \dots, L_d$  non-redundant low energy symbol sets from  $\mathcal{S}$  into the candidate list  $\tilde{\mathcal{S}}_{cand}$ . Note that  $L_d$  always  $1 \leq L_d \leq L$ .
  - For each element  $\tilde{\mathbf{s}}_{cand}^{(k)} \in \tilde{\mathcal{S}}_{cand}$ ,  $k = 1, 2, \dots, L_d$ , do
    - Copy the high energy symbol set estimate  $\tilde{\tau}^{(j)}[d]$  into  $\tilde{\mathbf{s}}_{cand}^{(k)}$ .
    - Compute the error metric
 
$$e_{cand}^{(k)} = \left\| \mathbf{y} - \mathbf{H} \tilde{\mathbf{s}}_{cand}^{(k)} \right\|^2.$$
  - Update the tentative list of minimum error metrics,  $\mathcal{E}_{min}$ , by finding the  $l$  smallest metrics,
 
$$e_{min} = \min_{\substack{1 \leq i \leq L \\ 1 \leq k \leq L_d}} \left\{ e_{cand}^{(k)}, e^{(i)} \right\}, \quad l = 1, 2, \dots, L$$
 where  $e^{(i)} \in \mathcal{E}$  is the  $i$ -th tentative metric in  $\mathcal{E}$ .
  - Update the corresponding list of symbol vectors,  $\mathcal{S}_{min}$ , by picking the  $l = 1, 2, \dots, L$  symbol vectors from  $\tilde{\mathcal{S}}_{cand}$  and  $\mathcal{S}$  with minimum error metric  $e_{min}^{(l)}$ .
  - Set  $\mathcal{S} = \mathcal{S}_{min}$  and  $\mathcal{E} = \mathcal{E}_{min}$ .
3. Terminate the list combining algorithm. Set  $q = q + 1$ .

from the  $D$  lists  $\mathcal{S}_{br}[d]$ . For each element  $\tilde{\mathbf{s}}_{br}^{(k)} \in \tilde{\mathcal{S}}_{br}$ , we compute the error metric  $e_{br}^{(k)} \in \mathcal{E}_{br}$  as

$$e_{br}^{(k)} = \left\| \mathbf{y} - \mathbf{H} \tilde{\mathbf{s}}_{br}^{(k)} \right\|^2 = \sum_{c=1}^D \left| y[c] - \sum_{u=1}^D h_{cu} \hat{s}_u^{(k)} \right|^2 \quad (13)$$

where  $\hat{s}_u^{(k)}$  are the symbol values of the branch vector  $\tilde{\mathbf{s}}_{br}^{(k)} \in \tilde{\mathcal{S}}_{br}$  with  $k = 1, 2, \dots, K$ . Using the list of branch error metrics  $\mathcal{E}_{br}$  and the stored tentative list of error metrics  $\mathcal{E}$ , we find the tentative list of  $L$  minimum error metrics  $\mathcal{E}_{min} = \{e_{min}^{(1)}, e_{min}^{(2)}, \dots, e_{min}^{(L)}\}$  by searching over

$$e_{min}^{(l)} = \min_{\substack{1 \leq i \leq L \\ 1 \leq k \leq K}} \left\{ e_{br}^{(k)}, e^{(i)} \right\}, \quad l = 1, 2, \dots, L \quad (14)$$

where  $e^{(i)} \in \mathcal{E}$  denotes the  $i$ -th element in the tentative list of error metrics from the previous  $(q - 1)$ -th iteration for  $q > 1$ . For  $q = 1$ , we choose  $\mathcal{E} = \{\infty\}$ . We define the list of  $L$  ( $D \times 1$ ) symbol vector estimates corresponding to  $\mathcal{E}_{min}$  as  $\mathcal{S}_{min} = \{\hat{\mathbf{s}}_{min}^{(1)}, \hat{\mathbf{s}}_{min}^{(2)}, \dots, \hat{\mathbf{s}}_{min}^{(L)}\}$ .

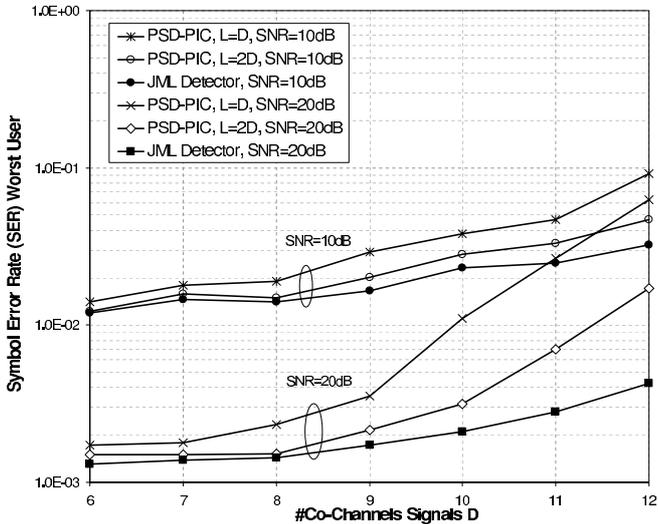


Figure 5: SER performance of the worst user for a  $M = 5$ -element ULA and various sizes  $L$  of the global list  $\mathcal{S}$ .

The list elements  $\hat{\mathbf{s}}_{\min}^{(l)} \in \mathcal{S}_{\min}$  are found by picking symbol values from the lists  $\tilde{\mathcal{S}}_{br}$  and  $\mathcal{S}$ .  $\mathcal{S}_{\min}$  and  $\mathcal{E}_{\min}$  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 the  $D$  lists of non-redundant high energy symbol sets,  $\tilde{\mathcal{T}}[d] = \{\tilde{\tau}^{(1)}[d], \tilde{\tau}^{(2)}[d], \dots, \tilde{\tau}^{(J_d)}[d]\}$  by copying the non-redundant high energy symbol set estimates from the corresponding branch list  $\mathcal{S}_{br}[d]$ . Hence, each reduced list  $\tilde{\mathcal{T}}[d]$  has size  $1 \leq J_d \leq L$ .

We can now describe the iterative list combining algorithm which takes the lists  $\mathcal{S}$ ,  $\mathcal{E}$  and  $\tilde{\mathcal{T}}[d]$  as inputs. The algorithm is summarized in Table 1. It typically requires  $Q_{lc} = 2$  or 3 iterations.

## 5. SIMULATION RESULTS

We use simulation to compare the performance of our detection algorithm with JML.  $D$  users are assumed to transmit QPSK (4-QAM) signals to a receiver with an  $M$  antenna linear array. The antenna elements are spaced at distance  $B = \lambda$  apart. We use a random allocation of the users into  $D$  equal size sectors<sup>5</sup> within the array's view angle of  $\theta_{\max} = \pm 60^\circ$ . The user signals are equal power symbol synchronous signals with random phase. We further assume that the receiver has perfect knowledge of the user's channels. The detection algorithm is configured to perform  $Q_{pic} = 3$  and  $Q_{lc} = Q = 2$  iterations. We choose a SEIR threshold of  $10dB$  to distinguish the high and low energy user sym-

<sup>5</sup>The ULA is highly selective in AOA for non-fading memoryless channels. We therefore use equal size sectors and random user spacing to obtain comparable results for different numbers of users.

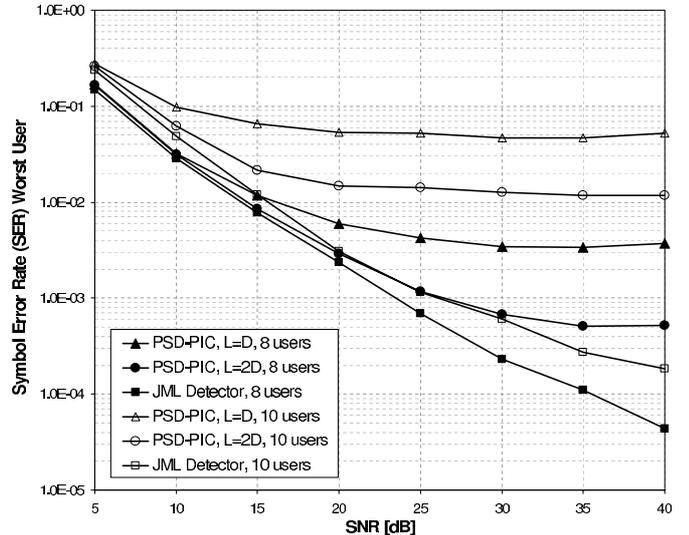


Figure 6: SER performance of the worst user for a  $M = 4$ -element ULA with  $D = 8$  and 10 equal energy users and global list  $\mathcal{S}$  of sizes  $L = D$  and  $2D$  at various SNR.

bol. In addition, the maximum number of high energy symbols in each receiver branch is limited to  $\mu_{\max} = 3$ . Performance results are shown as the symbol error rate (SER) of the worst user at different SNRs and sizes of the global estimated symbol list  $\mathcal{S}$ . The SNR at each receive antenna is defined as the ratio of signal and noise variances,  $SNR = 10 \log_{10} (\sigma_s^2 / \sigma_z^2)$ , where  $\sigma_s^2$  is the average received signal power. Simulations were stopped after 50 errors were experienced by a single user.

Fig. 5 depicts the SER versus the number of users for a receiver with  $M = 5$  antennas at  $SNR = 10dB$  and  $20dB$ . It can be seen that the SER increases with the number of users. This is due to the increasing amount of CCI remaining in the received signal after MRC preprocessing. Fig. 5 shows that our PSD-PIC detector well approximates JML detection up to  $D = 8$  users for list size  $L = D$  and up to  $D = 10$  users for list size  $L = 2D$ . The better performance for  $L = 2D$  is at the cost of increased complexity.

For the second set of simulations we used a receiver with  $M = 4$  antennas and  $D = 8$  and 10 users at various SNR. All other parameters remain the same. Fig. 6 depicts the SER results. For  $D = 8$  users PSD-PIC approximates JML up to  $SNR = 15dB$  and  $25dB$  for list sizes  $L = D$  and  $2D$ , respectively. For  $D = 10$  users, detection performance is worse than JML. This is due to insufficient list size  $L$  and the low number of high energy users considered in each detector branch. Increasing  $L$  and/ or  $\mu_{\max}$  can significantly improve performance. This represents a complexity-performance trade off.

Table 2: Comparison of computational complexity

$D$	JML				PSD-PIC	
	Receive Antennas				List Size of $\mathcal{S}$	
	$M = 4$	$M = 5$	$M = 8$	$M = 10$	$L = D$	$L = 2D$
5	8.2E3	–	–	–	1.1E4	1.9E4
6	3.3E4	4.1E4	–	–	1.8E4	3.4E4
7	1.3E5	1.6E5	–	–	3.2E4	6.2E4
8	5.2E5	<b>6.6E5</b>	–	–	<b>5.0E4</b>	9.9E4
9	2.1E6	2.6E6	4.2E6	–	6.9E4	1.4E5
10	8.4E6	<b>1.1E7</b>	1.7E7	–	9.4E4	<b>2.0E5</b>
11	3.4E7	4.2E7	6.7E7	8.4E7	1.2E5	2.6E5
12	1.3E8	1.7E8	2.7E8	3.4E8	1.6E5	3.5E5

## 6. COMPLEXITY

Algorithm complexity depends on several parameters. Among these are the number of users  $D$ , the alphabet size  $|\mathcal{A}|$ , the SEIR threshold, the maximum number of high energy symbols  $\mu_{\max}$ , the number of iterations  $Q_{pic}$ ,  $Q_{lc}$  and  $Q$ , and the sizes of the lists  $\mathcal{S}_{br}[d]$  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 indicate complexity. Table 2 shows the average number of real squaring operations for the simulations in Section 5. Note that the computation of each error metric requires two real squarings. JML requires  $2M|\mathcal{A}|^D$  real squarings.

It can be seen from Table 2 that our algorithm achieves complexity savings of up to several orders of magnitude over JML detection. It is further evident that JML complexity increases exponentially with  $D$ , whereas our list approach essentially has a more linear increase. For only slightly overloaded scenarios, e.g. ( $M = 4$ ,  $D = 5$ ) with list size  $L = D$  or ( $M = 5$ ,  $D = 6$ ) with  $L = 2D$ , complexity is slightly worse than JML. This is due to the list size  $L$  being unnecessarily large. Clearly, the correct choice of the detection parameters is crucial to achieve complexity savings without sacrificing performance. Close to JML performance is obtained for an  $M = 5$ -element ULA with ( $D = 8$ ,  $L = D$ ) and ( $D = 10$ ,  $L = 2D$ ). In both cases complexity savings between one and two orders of magnitude are achieved. Savings in computational complexity become more significant when the number of receive antennas is large.

## 7. CONCLUSIONS

In this paper, a MUD algorithm for overloaded receivers with a linear receive array is presented. It uses MRC as a preprocessing stage and an iterative parallel sym-

bol detector with list feedback of the best candidates in the detection stage. The MRC preprocessor maximizes the instantaneous SNR at the preprocessor output. The detector has  $D$  branches, one for each user. In each branch, a symbol estimator computes a branch list of symbol vectors by searching over user symbols with high energy. The low energy symbols account for residual CCI and are estimated using a novel list-based parallel interference cancellation scheme. The branch lists are then searched and combined to find the global list of symbol vectors with minimum Euclidean error metric. This list is fed back to the detection branches to obtain improved estimates. After a sufficient number of iterations, the symbol vector with minimum error metric is output by the detector.

Our detection algorithm is shown to approximate JML detection with reduced complexity. Its parallel processing structure makes it well suited for practical application.

## 8. REFERENCES

- [1] D. G. Brennan, "Linear diversity combining techniques," *Proc. IRE*, vol. 47, pp. 1075-1102, June 1959.
- [2] 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.
- [3] S. Talwar, M. Viberg and A. Paulraj, "Blind separation of synchronous co-channel digital signals using an antenna array—Part I: Algorithms," *IEEE Trans. Signal Processing*, vol. 44., pp. 1184-1197, May 1996.
- [4] 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.
- [5] S. Bayram, J. Hicks, R. J. Boyle and J. H. Reed, "Joint ML approach in overloaded array processing," in *Proc. VTC*, pp. 394-400, Sept. 2000.
- [6] 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.
- [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. Krause, D. P. Taylor, and P. A. Martin, "On list detection for overloaded receivers," *to appear in PIMRC*, Athens, Greece, Sept. 2007.
- [9] J.-A. Tsai, R. M. Buehrer, and B. D. Woerner, "BER performance of a uniform circular array versus a uniform linear array in a mobile radio environment," *IEEE Trans. Wireless Commun.*, vol. 3, pp. 695-700, May 2004.