Decision feedback with rollout for multiuser detection in synchronous CDMA

F. Tu, D. Pham, J. Luo, K.R. Pattipati and P. Willett

Abstract: Decision feedback (DF) is one of the most popular methods in multiuser detection due to its simplicity and outstanding performance. Despite the efficiency of the DF detector, there is usually a large performance gap between the DF detector and the optimal maximum likelihood detector. Rollout, an emerging technique from planning and optimisation, is employed to improve the performance of the decorrelator-based DF detector for synchronous code division multiple access channels. Simulation results show that the proposed algorithm significantly improves the joint error rate of the DF detector, and even outperforms the sequential group decision feedback detector for similar time complexities. Further, owing to the inherent parallel structure of the proposed algorithm, the method is particularly useful in applications where speed and accuracy are both important.

1 Introduction

Owing to the NP-hard nature of the general multiuser detection (MUD) problem in code-division multiple access (CDMA) channels [1], much research has been devoted to the development of suboptimal algorithms that provide reliable decisions with relatively low computational costs. Linear detectors, such as the decorrelator [2] and the linear minimum mean square error (LMMSE) detector [1], improve the performance of the conventional matched-filter detector significantly, while limiting their computational complexities to $O(K^2)$ where K is the number of users. Decision-driven detectors, which include the multistage detector [3], the decision feedback (DF) detector [4], and the group decision feedback (GDF) detector [5], generally provide significant improvement over linear detectors without raising the computational complexity beyond $O(K^2)$. Even so, there is still a large performance gap between these detectors and the optimal maximum likelihood (ML) detector. Therefore practical methods that provide a nice tradeoff between computational cost and optimality are desirable; rollout strategies [6] offer such a tradeoff.

Rollout algorithms [6] are a class of suboptimal solution methods inspired by the policy iteration of dynamic programming. The attractive aspects of rollout algorithms are simplicity, broad applicability, and suitability for online implementation. While rollout algorithms do not aspire to optimal performance, they typically result in a consistent and substantial improvement over the suboptimal algorithm that underlies them.

We employ rollout to improve the performance of the decorrelator-based DF detector for synchronous code division multiple access channels. Simulation results show that the proposed algorithm, which we refer to as the rollout decision feedback (RDF) detector, significantly improves the joint error rate (JER) of the DF detector, and even outperforms the sequential GDF detector for similar time complexities. Joint error rate was defined in [4] as the probability that at least one user is detected erroneously. Further, due to the inherent parallel structure of the proposed algorithm, the method is particularly useful in applications where speed and accuracy are both important.

2 Problem formulation and DF detector

A discrete-time model for the matched-filter outputs at the receiver of a CDMA channel is given by the K-length vector [1]

$$\boldsymbol{y} = \boldsymbol{RWb} + \boldsymbol{n} \tag{1}$$

where $\boldsymbol{b} \in \{-1, +1\}^K$ denotes the K-length vector of bits transmitted by the K active users, $\boldsymbol{R}$ is the symmetric normalised correlation matrix with unit diagonal elements, $\boldsymbol{W}$ is a diagonal matrix whose ith diagonal element, w_i is the square root of the received energy per bit of the ith user, and $\boldsymbol{n}$ is a real-valued zero-mean Gaussian random vector with covariance matrix $\sigma^2\boldsymbol{R}$. When all the user signals are equally probable, the optimal solution of (1) is the output of the ML detector [2]

$$\phi_{ML} : \hat{\boldsymbol{b}} = \arg \min_{\boldsymbol{b} \in \{-1,+1\}^K} (\boldsymbol{b}^T \boldsymbol{WRWb} - 2\boldsymbol{y}^T \boldsymbol{Wb}) \tag{2}$$

The ML detector has the property that it minimises, among all detectors, the probability that not all users' decisions are correct.

The DF detector makes its decisions sequentially, one user at a time, using successive interference cancellation. Consequently, the quality of the DF solution is strongly dependent on the detection order. The optimal user ordering for the decorrelator-based DF detector is given in theorem 1 of [4], and we assume throughout this paper

IEE Proceedings online no. 20040425

doi:10.1049/ip-com:20040425

Paper first received 7th October 2003 and in revised form 9th February 2004. Originally published online: 7th July 2004

F. Tu is with Qualtech Systems, Inc., Putnam Park, Suite 501, 100 Great Meadow Road, Wethersfield, CT 06109, USA

D. Pham, K.R. Pattipati and P. Willett are with the Electrical and Computer Engineering Department, University of Connecticut, U-1157, Storrs, CT 06269, USA

J. Luo is with the Institute for Systems Research, University of Maryland, MD 20742, USA

that the users have been ordered according to this rule unless stated otherwise.

Assuming that $\boldsymbol{R}=\boldsymbol{L}^T\boldsymbol{L}$ the system can also be represented by a white noise model

$$\tilde{\boldsymbol{y}} = \boldsymbol{L}\boldsymbol{W}\boldsymbol{b} + \tilde{\boldsymbol{n}} \tag{3}$$

where $\tilde{\boldsymbol{y}} = \boldsymbol{L}^{-T}\boldsymbol{y}$ and $\tilde{\boldsymbol{n}} = \boldsymbol{L}^{-T}\boldsymbol{n}$ is a white Gaussian noise vector with zero mean and covariance matrix $\sigma^2\boldsymbol{I}$. The decorrelator-based DF solution is then given as

$$\phi_{DFD} : \hat{b}_i = \text{sign}\left(\tilde{y}_i - \sum_{j=1}^{i-1} (LW)_{ij}\hat{b}_j\right) \tag{4}$$

where the sequential nature of the bit decisions is clear.

3 RDF

The MUD problem can be viewed as a tree search wherein the nodes represent users (with each level of the tree being occupied by a single user) and the branches emanating from a node are the possible signals that the user of that node can assume (i.e. either '+1' or '−1' in the binary case we study here). In a worst-case scenario, the optimal detector traverses the entire tree (i.e. enumerates all 2^K possible solutions) and selects the best solution using the performance metric given by (2). This is precisely what we are seeking to avoid.

To reduce the computational cost, rather than traverse the entire tree, in each iteration we only expand the tree m levels deeper; and for each leaf node obtained (2^m of them) we apply a suboptimal multiuser detector to determine which path to take among all the paths emanating from that node. The total number of paths (or solutions) in the tree is equal to 2^m where m is chosen such that $2^K \gg 2^m$ (also see Fig. 1). In this case, the MUD technique that we employ to complete the solutions is the DF detector because of its low computational complexity and comparatively good performance. (This is the base policy in the rollout.)

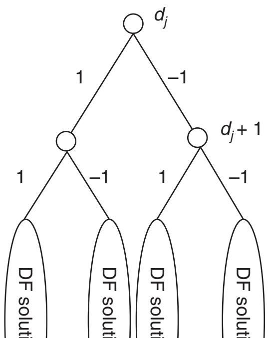

Fig. 1 *Illustration of tree with m = 2 levels*

One can best view the RDF detector as a conditional DF detector where the decisions of some of the users are assumed known *a priori*. For the RDF detector to be significantly better than the original DF detector, none of the 2^m solutions is claimed as a final solution. Instead we make decisions only on the users who are occupying the m levels of the tree who are assigned signal values from the path with the smallest ML cost. Naturally, the degree to which the DF solution agrees with the truth is key in making a reliable decision about the m users.

Consider the DF solution for the ith user. In the ideal case where $\hat{b}_1$, $\hat{b}_2$, …,$\hat{b}_{i-1}$ are correct there is no user interference and the decision for the ith user is

$$\tilde{y}_i = (LW)_{ii}b_i + \tilde{n}_i \tag{5}$$

Then the probability of making an error on user i for the ideal case, which we denote as $P_e(b_i)$, is

$$P_e(b_i) = Pr(\tilde{n}_i > (LW)_{ii}) = Q\left(\frac{(LW)_{ii}}{\sigma}\right) \tag{6}$$

where

$$Q(x) = \int_x^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz$$

Intuitively, it makes sense to occupy the top levels of the tree (as opposed to the lower levels that are relegated to solution by DF) with the users who are least likely to be detected correctly by the DF detector. In doing so, the quality of the 2^m DF solutions (see Fig. 1) may be improved since incorrect decisions on the part of the aforementioned users can lead to incorrect decisions for other users depending on the detection order. Using (6) as the criterion for determining which users should occupy the levels of the tree first, we define $\boldsymbol{D} = (d_1, d_2, \ldots, d_K)$ as the sequence of error prone users where $P_e(b_{d_1}) \geq P_e(b_{d_2}) \geq \ldots \geq P_e(b_{d_K})$. Without the perfect feedback assumption the probability of error of user signal b_i in the DF solution is bounded asymptotically by $Q(\min_{j\leq i}[(LW)_{jj}]/\sigma) \geq P_e(b_i) \geq Q((LW)_{ii}/\sigma)$. Therefore the proposed sequence $\boldsymbol{D}$, which is derived from criterion (6), is an approximation to the decreasing order on the error probabilities of the user signals. Let D_j represent a set of users, $|D_j|$ denote the cardinality of D_j and define $\boldsymbol{b}_{D_j}$ as the user signals for D_j. The RDF algorithm is described as follows:

Step 1: Sort users according to the user ordering criterion proposed for the DF detector in [4] (especially theorem 1 of [4]).

Step 2: Determine the sequence $\boldsymbol{D}$ and partition it as $\boldsymbol{D} = \{D_1, D_2, \ldots, D_F\}$, where $|D_j| = m$ for $j = 1, 2, \ldots, F-1$ and $|D_F| \leq m$ with $F = \lceil |D_j|/m \rceil$.

Step 3: Initialise the stage counter $j = 1$.

Step 4: For each realisation of $\boldsymbol{b}_{D_j}$ ($\boldsymbol{b}_{D_j} \in \{1, -1\}^{|D_j|}$), obtain the decisions for the still undecided users using the DF detector, compute and store the associated ML cost for the overall solution, *viz.* $\hat{\boldsymbol{b}}^T \boldsymbol{W}\boldsymbol{R}\boldsymbol{W}\hat{\boldsymbol{b}} - 2\boldsymbol{y}^T\boldsymbol{W}\hat{\boldsymbol{b}}$, assuming $\hat{\boldsymbol{b}}$ is the overall DF solution.

Step 5: Select the realisation of $\boldsymbol{b}_{D_j}$ that is associated with the best path and claim this as the solution for the users in D_j.

Step 6: If $j < F$, let $j = j+1$ and go to step 4. Otherwise, STOP.

In practice we usually terminate the program early if the best cost does not improve after a certain number of iterations. As all the related computations for a given path are independent of the other paths, the RDF detector is particularly computationally efficient if parallel processing is employed.

4 Simulation results and discussion

For all the examples we generate the user signal amplitudes randomly according to $w_i \sim N(4.5,4)$ and limit them to the range of [3, 6]. Theorem 1 of [4] was applied for the user ordering in the DF, RDF, and probabilistic data association (PDA) detectors [7, 9]. The optimal solutions (ML decisions) are obtained via a fast branch-and-bound algorithm that employs depth-first search on an optimal user ordering [10].

In the first two examples the depth m of the RDF detector was set to 2. Figure 2 shows the performance comparison based on 10^5 Monte-Carlo runs for a six-user system with 11-length randomly generated spreading sequences. Figure 3 shows the performance comparison based on 300 000 Monte-Carlo runs for a 20-user system with 25-length randomly generated spreading sequences. For the aforementioned examples the algorithm is terminated early if the best cost does not improve after four consecutive stages.

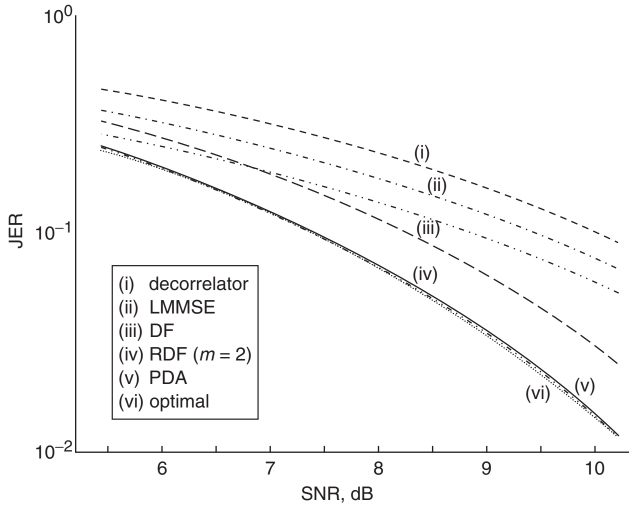

Fig. 2 *Performance comparison, six users, 11-length randomly generated codes, 10^5 Monte-Carlo runs with importance sampling*

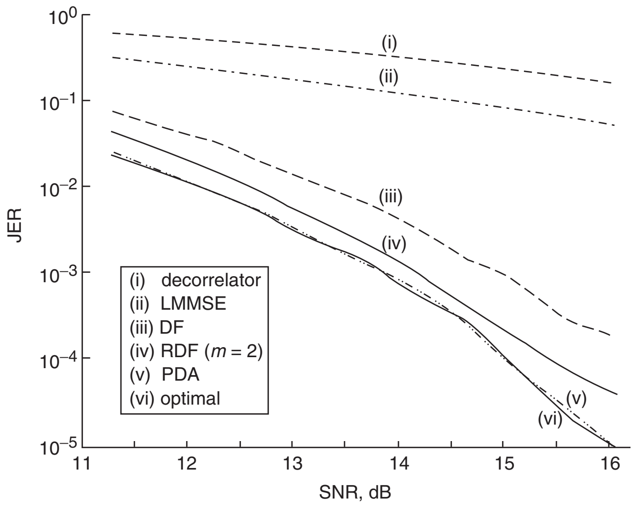

Fig. 3 *Performance comparison, 20 users, 25-length randomly generated codes, 300 000 Monte-Carlo runs with importance sampling*

In the third example we consider a 20-user system with 21-length randomly generated spreading sequences. For the RDF detector we varied m from 1 to 3 and we terminate the algorithm early if the best cost does not improve after four consecutive stages. Figure 4 shows the performance comparison based on 300 000 Monte-Carlo runs. Compared with the previous two examples the level of improvement is reduced as the number of users comes close to the code length.

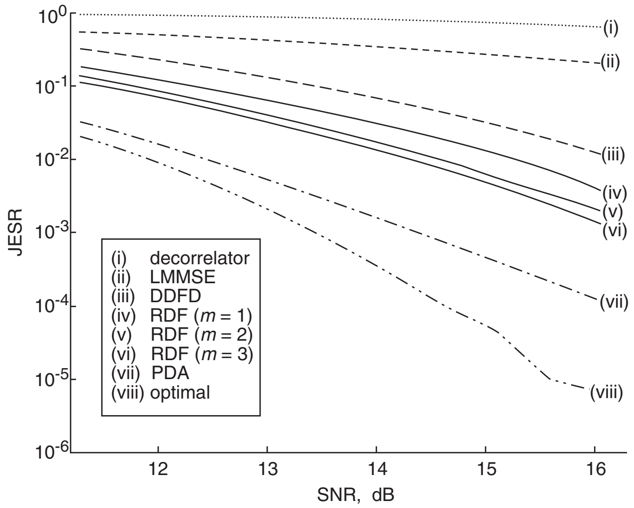

Fig. 4 *Performance comparison, 20 users, 21-length randomly generated codes, 300 000 Monte-Carlo runs with importance sampling*

In the final example we fixed the system SNR to 12dB and vary the number of users from 5 to 55. The spreading sequences are randomly generated and the ratio between the spreading factor and the number of users is fixed at 1.2 where K varies from 5 to 55 in increments of 5. For the RDF detector, we use $m = 4$ and we terminate the algorithm early if the best cost does not improve after five consecutive stages. In Fig. 5, the computational cost of the RDF detector is computed under the assumption that the computations for all the paths of the tree are done in parallel. The PDA detector, whose performance is typically close to optimal, has a computational complexity of $O(K^3)$ [8]. Hence, with the availability of parallel processing, the complexity of the RDF detector is substantially less than $O(K^3)$. The simulation is based on 1000 Monte-Carlo runs.

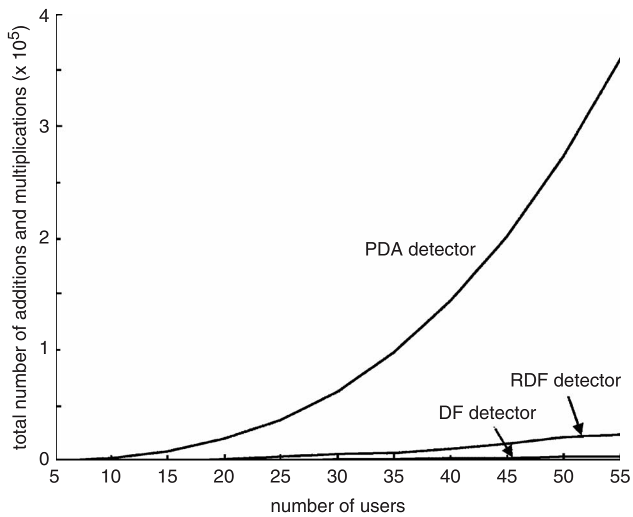

Fig. 5 *Computational cost comparison, random signature sequences, spreading factor = 1.2K, SNR = 12 dB, 1000 Monte-Carlo runs*

5 Conclusions

Significant reductions in the JER of the DF detector have been achieved by applying rollout, which requires only minor modifications to the DF detector. The inherent parallel architecture of the RDF detector allows for the possibility of achieving even lower JERs without raising the time complexity. In cases where the system is nearly overloaded, the RDF detector in general does not attain near-optimal performance. However, for other cases, the RDF detector has a JER that comes reasonably close to that of the ML detector. Future research will focus on improving the DF detector via an adaptive rollout strategy where the number of tree levels m will vary from stage to stage. Rollout as underpinned by DF has been explored here for the MUD problem. Underlying suboptimal algorithms other than DF can be used for the MUD problem, and we believe that the rollout philosophy may have wide applicability in communications problems.

6 Acknowledgment

This work is supported by the Office of Naval Research under contract 00014-00-1-0101.

7 References

1 Verdu, S.: 'Multiuser detection' (Cambridge University Press, New York, 1998)
2 Lupas, R., and Verdu, S.: 'Linear multiuser detectors for synchronous code-division multiple-access channels', *IEEE Trans. Inf. Theory*, 1989, **35**, pp. 123–136
3 Varanasi, M., and Aazhang, B.: 'Near-optimum detection in synchronous code-division multiple-access systems', *IEEE Trans. Commun.*, 1991, **39**, pp. 725–736
4 Varanasi, M.: 'Decision feedback multiuser detection: a systematic approach', *IEEE Trans. Inf. Theory*, 1999, **45**, pp. 219–240
5 Varanasi, M.: 'Group detection for synchronous gaussian code-division multiple-access channels', *IEEE Trans. Inf. Theory*, 1995, **41**, pp. 1083–1096
6 Bertsekas, D.P., Tsitsiklis, J.N., and Wu, C.: 'Rollout algorithms for combinatorial optimization', *J. Heuristics*, 1997, **3**, (3), pp. 245–262
7 Luo, J., Pattipati, K., Willett, P., and Hasegawa, F.: 'Near optimal multiuser detection in synchronous CDMA using probabilistic data association', *IEEE Commun. Lett.*, 2001, **5**, pp. 361–363
8 Luo, J., Pattipati, K., and Willett, P.: 'Optimal grouping algorithm for a group decision feedback detector in synchronous CDMA communications', *IEEE Trans. Commun.*, 2003, **51**, (3), pp. 341–346
9 Hasegawa, F., Luo, J., Pattipati, K., Willett, P., and Pham, D.: 'Speed and accuracy comparison of techniques for multi-user detection in synchronous CDMA', *IEEE Trans. Commun.*, (to be published)
10 Luo, J., Pattipati, K., Willett, P., and Brunel, L.: 'Branch-and-bound-based fast optimal algorithm for multiuser detection in synchronous CDMA'. Proc. IEEE Int. Conf. on Communications, Anchorage, AK, USA, May 2003