Published in IET Communications
Received on 17th April 2008
Revised on 22nd June 2008
doi: 10.1049/iet-com:20080210

ISSN 1751-8628

Partial regularisation approach for detection problems in underdetermined linear systems

X.-W. Chang[1] *X. Yang*[1] *T. Le-Ngoc*[2] *P. Wang*[2]

[1]*School of Computer Science, McGill University, Montreal, Canada, QC H3A 2A7*
[2]*Department of Electrical and Computer Engineering, McGill University, Montreal, Canada, QC H3A 2A7*
E-mail: xyang35@cs.mcgill.ca

Abstract: The maximum likelihood detection problem in many underdetermined linear communications systems can be described as an underdetermined integer least squares (ILS) problem. To solve it efficiently, a partial regularisation approach is proposed. The original underdetermined ILS problem is first transformed to an equivalent overdetermined ILS problem by using part of the transmit vector to do the regularisation. Then the overdetermined ILS problem is solved by conventional sphere decoding algorithms. Simulation results indicate that this approach can be much more efficient than other approaches for any square constellation higher than 4QAM.

1 Introduction

In many linear communications systems, the received signal vector can be represented as a linear combination of the channels corrupted by additive noise. The relation between the received signal vector and the transmit signal vector is written as a complex linear system

$$\tilde{\boldsymbol{y}} = \tilde{\boldsymbol{H}}\tilde{\boldsymbol{x}} + \tilde{\boldsymbol{v}} \tag{1}$$

where $\tilde{\boldsymbol{H}} \in \mathbb{C}^{N_r \times N_t}$ represents the system matrix with N_t inputs and N_r outputs for the linear system, $\tilde{\boldsymbol{v}} \in \mathbb{C}^{N_r}$ is the white Gaussian noise vector with distribution $CN(\boldsymbol{0}, 2\sigma^2\boldsymbol{I})$, and $\tilde{\boldsymbol{x}} \in C^{N_t}$ is the unknown signal vector and its elements are odd numbers in the finite set $\tilde{\mathcal{X}}_k = \{k_1 + k_2 j : k_1, k_2 = \pm 1, \pm 3, \ldots, \pm(2^k - 3), \pm(2^k - 1)\}$, where $j^2 = -1$. Note that $k = 1, 2, 3$ correspond to QPSK (i.e. 4QAM), 16QAM, 64QAM constellations, respectively. The detection problem is to recover (or estimate) the transmitted vector $\tilde{\boldsymbol{x}}$ in (1). Such detection problems arise from many communication applications, including lattice decoding [1], multi-input multi-output (MIMO) detection problems [2], multi-user detection problems in direct-sequence code-division multi-access (DS-CDMA) [3] and multi-carrier code-division multi-access (MC-CDMA) [4].

In order to avoid complex computations in the detection process of the transmitted vector $\tilde{\boldsymbol{x}}$, we first transform the complex linear system (1) to the following real one (see, e.g. [5], about the transformation)

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{v} \tag{2}$$

where $\boldsymbol{y} \in \mathbb{R}^n$, $\boldsymbol{H} \in \mathbb{R}^{m \times n}$ with $m \triangleq 2N_r$ and $n \triangleq 2N_t$, $\boldsymbol{v} \sim N(0, \sigma^2\boldsymbol{I}_m)$, and $\boldsymbol{x} \in \mathcal{X}_k^n$ with $\mathcal{X}_k = \{\pm 1, \pm 3, \ldots, \pm(2^k - 3), \pm(2^k - 1)\}$.

To find the maximum likelihood (ML) estimation of the complex transmit vector $\tilde{\boldsymbol{x}}$ in (1) or the real 'transmit' vector $\boldsymbol{x}$ in (2), one solves the following minimisation problem

$$\min_{\boldsymbol{x} \in \mathcal{X}_k^n} \|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|_2^2 \tag{3}$$

which we refer to as a box-constrained integer least squares (ILS) problem. When the matrix $\boldsymbol{H}$ has full column rank, a conventional sphere decoding (SD) algorithm can be employed to find the optimal solution of (3) (see, e.g. [2–8]). In this paper, we are interested in the case that $\boldsymbol{H}$ has full row rank, that is, (3) is a constrained underdetermined ILS problem. One such application is the multiple-antenna communication systems where there are more transmitting antennas than receiving antennas. For this case, a conventional SD algorithm cannot be applied directly. To solve this problem, Damen *et al.* [9], Dayal and Varanasi [10], Yang *et al.* [11] and Chang and Yang [12] proposed various generalised SD (GSD) algorithms. All those algorithms

mainly consider how to generate a sequence of determined sub-ILS problems. For the convenience of the reader, we give more details about GSD algorithms in the appendix.

Cui and Tellambura [13] proposed a different approach, which transforms the underdetermined problem (3) to an equivalent overdetermined problem so that a conventional SD algorithm can then be applied. We refer to this approach as a regularisation approach and the corresponding algorithm as regularized SD algorithm (RSD). The idea of this approach is first transforming the entire vector $\boldsymbol{x}$ into a new vector $\bar{\boldsymbol{x}}$ whose entries are either 1 or -1, and then transforming the underdetermined problem to an overdetermined problem by adding a constant term which involves the 2-norm of the entire vector $\bar{\boldsymbol{x}}$ to the objective function of the problem. For the constant-modulus QPSK constellation (i.e. 4QAM) $\bar{\boldsymbol{x}} = \boldsymbol{x}$, the dimension of the equivalent overdetermined problem is the same as that of the original underdetermined problem. Here, dimension means the number of entries in the unknown vector to be estimated. However, for the non-constant high-order QAM (i.e. $k \geq 2$ in $\mathcal{X}_k$), the transformation from $\boldsymbol{x}$ to $\bar{\boldsymbol{x}}$ enlarges the dimension of the problem from n to nk. This makes the search process for solving the equivalent over-determined problem less efficient, although the transformation shrinks the constellation set.

By observing the dimension-increasing problem lying in algorithm RSD which decreases its efficiency, we propose an efficient partial regularisation approach to relieve that problem. The key idea is to choose part of the transmit vector $\boldsymbol{x}$ to do the transformation and then to do the regularisation. This can greatly reduce the computational complexity for high QAM constellations. We will also discuss some issues involved in this approach and propose our strategies. The resulting method will be referred to as partial RSD (PRSD) algorithm.

It is also noted that some methods to find a sub-optimal solution to (3), for example, [14, 15], were proposed to reduce computational complexity, which will not be discussed in this paper.

The rest of this contribution is organised as follows. In Section 2, we present our partial regularisation approach. In Section 3, we compare our method with algorithm RSD and discuss a few efficiency issues. Section 4 gives simulation results to show the efficiency of this approach. Finally, conclusions are given in Section 5.

2 Partial regularisation approach

The regularisation approach presented in [13] transforms the underdetermined ILS problem (3) to an overdetermined ILS problem. For high-order QAM, this approach enlarges the problem size and becomes less efficient. In this section, we will present a modified approach to improve efficiency.

Partition $\boldsymbol{H}$ and $\boldsymbol{x}$ as follows

$$\boldsymbol{H} = [\underset{n-l}{\boldsymbol{H}_1}, \underset{l}{\boldsymbol{H}_2}]m, \quad \boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} \begin{matrix} n-l \\ l \end{matrix} \tag{4}$$

We will discuss how to choose l in the next section. Following [13], we can write $\boldsymbol{x}_2 \in \mathcal{X}_k^l$ as a linear combination of $\boldsymbol{x}_2^{(i)} \in \mathcal{X}_1^l$ for $0 \leq i \leq k-1$:

$$\boldsymbol{x}_2 = \sum_{i=0}^{k-1} 2^i \boldsymbol{x}_2^{(i)} \tag{5}$$

Define

$$\begin{aligned} \bar{\boldsymbol{H}}_2 &\triangleq [\boldsymbol{H}_2, 2\boldsymbol{H}_2, \ldots, 2^{k-1}\boldsymbol{H}_2] \in \mathbb{R}^{m \times kl}, \\ \bar{\boldsymbol{x}}_2 &\triangleq \begin{bmatrix} \boldsymbol{x}_2^{(0)} \\ \boldsymbol{x}_2^{(1)} \\ \vdots \\ \boldsymbol{x}_2^{(k-1)} \end{bmatrix} \in \mathbb{R}^{kl} \end{aligned} \tag{6}$$

Notice that $\|\bar{\boldsymbol{x}}_2\|_2^2 = kl$, so $\|\bar{\boldsymbol{x}}_2\|_2^2$ is a constant. Then from (4)–(6), we see that the original ILS problem (3) is equivalent to

$$\min_{\boldsymbol{x}_1 \in \mathcal{X}_k^{n-l}, \bar{\boldsymbol{x}}_2 \in \mathcal{X}_1^{kl}} \left\| \boldsymbol{y} - [\boldsymbol{H}_1, \bar{\boldsymbol{H}}_2] \begin{bmatrix} \boldsymbol{x}_1 \\ \bar{\boldsymbol{x}}_2 \end{bmatrix} \right\|_2^2 + \alpha^2 \|\bar{\boldsymbol{x}}_2\|_2^2 \tag{7}$$

where we refer to α as a regularisation parameter and its choice will be discussed later. Therefore with

$$\begin{aligned} \bar{\boldsymbol{H}} &\triangleq \begin{bmatrix} \boldsymbol{H}_1 & \bar{\boldsymbol{H}}_2 \\ \boldsymbol{0} & \alpha \boldsymbol{I} \end{bmatrix} \in \mathbb{R}^{(m+kl) \times (n+(k-1)l)} \\ \bar{\boldsymbol{x}} &\triangleq \begin{bmatrix} \boldsymbol{x}_1 \\ \bar{\boldsymbol{x}}_2 \end{bmatrix} \in \mathbb{R}^{n+(k-1)l} \\ \bar{\boldsymbol{y}} &\triangleq \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{0} \end{bmatrix} \in \mathbb{R}^{m+kl} \end{aligned} \tag{8}$$

$$\bar{\mathcal{X}} \triangleq \left\{ \begin{bmatrix} \boldsymbol{x}_1 \\ \bar{\boldsymbol{x}}_2 \end{bmatrix} : \boldsymbol{x}_1 \in \mathcal{X}_k^{n-l}, \ \bar{\boldsymbol{x}}_2 \in \mathcal{X}_1^{kl} \right\} \tag{9}$$

the problem (7) can be rewritten as

$$\min_{\bar{\boldsymbol{x}} \in \bar{\mathcal{X}}} \|\bar{\boldsymbol{y}} - \bar{\boldsymbol{H}}\bar{\boldsymbol{x}}\|_2^2 \tag{10}$$

When $l \geq n - m$, this is a box-constrained overdetermined ILS problem and can be solved by any conventional SD algorithm which can handle the constraints. We will use the V-BLAST column re-ordering strategy in the reduction (or preprocessing) process (see [6, 8]) and apply the Schnorr–Euchner strategy-based search algorithm given in [8] for search. The reduction algorithm transforms $\bar{\boldsymbol{H}}$ to a $(n + (k-1)l) \times (n + (k-1)l)$ upper triangular matrix by a QR factorisation with column re-ordering, and the search

doi: 10.1049/iet-com:20080210

algorithm is an SD algorithm and its cost, which depends on the number of columns of $\bar{\boldsymbol{H}}$, dominates the cost of the whole algorithm to solve the problem (10).

3 Efficiency issues

In order to obtain an overdetermined ILS problem (4), l in (4) has to be at least $n-m$. In algorithm RSD given in [13], $\boldsymbol{x}_2$ was chosen as the entire vector $\boldsymbol{x}$, that is, in (4) $l=n$. In our partial regularisation approach, we choose $l=n-m+1$. The justification will be discussed in the next paragraph. Thus the matrix $\bar{\boldsymbol{H}}$ in (8) is $(m+k(n-m+1))\times(n+(k-1)(n-m+1))$ in our modified approach, whereas it is $(m+kn)\times(n+(k-1)n)$ in algorithm RSD. When $k=1$, the numbers of columns of $\bar{\boldsymbol{H}}$ in the two approaches are the same, whereas the number of rows is smaller in the modified approach. Thus the reduction process will cost less for the modified approach, although this is not a big deal, since the search process dominates the cost of the entire algorithm. When $k\geq 2$, the number of columns of $\bar{\boldsymbol{H}}$ in the modified approach is smaller than that in algorithm RSD, while the constraint set for each entry of $\boldsymbol{x}_1$ is still $\mathcal{X}_k$, which is larger than $\mathcal{X}_1$, the constraint set for each entry of the unknown vector $\bar{\boldsymbol{x}}$ in algorithm RSD. In other words, compared with Algorithm RSD, when $k\geq 2$, our modified approach decreases the dimension of the overdetermined ILS problem, while it increases the ranges of part of the constraint sets. Simulation results in Section 4 will show that this strategy can significantly decrease the computational complexity. Thus the dimension has more significant effect on the efficiency of the search process than the ranges of the constraints. This is related to the nature of the search process. The Schnorr–Euchner strategy-based search process can be regarded as a depth-first search process on a tree (see e.g. [5]), and our strategy actually decreases the height of the search tree, making the search process more efficient (see, e.g. [16 Chapters 12 and 13]) for explanations.

To make the number of columns of $\bar{\boldsymbol{H}}$ in the overdetermined ILS problem (10) as small as possible, l in (4) should be chosen to be $n-m$. It turns out, however, that this choice make the search algorithm less efficient. We can explain this from the structure of the $\boldsymbol{R}$ factor of the QR factorisation of $\bar{\boldsymbol{H}}$. When $l=n-m$, $\bar{\boldsymbol{H}}$ is $(m+k(n-m))\times(m+k(n-m))$. If we do not do any column re-ordering, then the $k(n-m)\times k(n-m)$ bottom right block of the $\boldsymbol{R}$ factor of the QR factorisation of $\bar{\boldsymbol{H}}$ is a diagonal matrix (more specifically, it is $\sigma\boldsymbol{I}$). Then, when the search algorithm goes from level $m+k(n-m)$ (at which $\bar{x}_{m+k(n-m)}$ is determined) to level $m+1$ (at which $\bar{x}_{m+1}$ is determined), no node of the search tree can be pruned, that is, no branch can be cut off, which makes the search process inefficient. If we use the V-BLAST column re-ordering strategy, usually the bottom right block of the $\boldsymbol{R}$ factor is still a diagonal matrix although its dimension is usually smaller than $k(n-m)$ (our limited numerical experiments indicated averagely its dimension is about half of $k(n-m)$). So, we still have an efficiency problem. By choosing l to be $n-m+1$, however, we usually avoid the diagonal structure problem in the $\boldsymbol{R}$ factor, which makes the search process more efficient. Certainly, choosing l to be larger than $n-m+1$ will also avoid this problem, but it would increase the dimension of the overdetermined ILS problem.

In (4), we use the last $n-m+1$ entries of $\boldsymbol{x}$ to define $\boldsymbol{x}_2$. But choosing a different $\boldsymbol{x}_2$ from $\boldsymbol{x}$ will make the efficiency of the algorithm to solve (10) different. Thus, we would like to determine how to choose a good $\boldsymbol{x}_2$ from $\boldsymbol{x}$, or how to re-order the columns of the matrix $\boldsymbol{H}$ so that we have a good $\boldsymbol{x}_2$ corresponding to the last $n-m+1$ columns of the re-ordered $\boldsymbol{H}$. From simulations, we found that sorting the columns of $\boldsymbol{H}$ in decreasing order with respect to the 2-norm helps to reduce the computational complexity of the search process when $k\geq 2$. The reason is that such ordering is likely to make the strict upper triangular part of the $\boldsymbol{R}$ factor of the QR factorisation of $\bar{\boldsymbol{H}}$ smaller, which can make the search process more efficient (see, [5, 7]), for some discussion on this issue.

In the ILS problem (7), α is a parameter and its choice has an effect on the efficiency and performance (i.e. the accuracy of the computed solution) of the algorithm to solve the problem. On one hand, α cannot be too small, otherwise $\bar{\boldsymbol{H}}$ in (8) will be nearly singular and the computed solution of (10) may not be the true solution (i.e. the ML estimate) due to rounding errors; on the other hand, α cannot be too large, otherwise, the search region, that is, the hyper-sphere determined by the first integer point found by the search algorithm, will be large, which will slow down the search process. It appears difficult if not impossible to find an optimal α that leads to the lowest computational complexity. In [13], α was chosen to be 1 or σ [the standard deviation of the noise $\boldsymbol{v}$ in the linear model (2)] in the simulations, and there was no other suggestion about how to choose a good α. We found from simulations that the optimal α in our modified approach depends strongly on σ and weakly on the dimension difference $n-m$ and k. If $n\leq 2m$ and $k\leq 3$ (this is often the case in applications), our simulations suggest that $\alpha=2^{7/4}\sigma$ is a good choice with respect to the efficiency and it has no negative effect on the performance.

4 Illustrative results

In this section, we compare the computational cost of our algorithm outlined in Section 2 with algorithm RSD in [13] and the recursive GSD (RGSD) algorithm in [12], which, to be referred to as algorithm RGSD for convenience, is faster than other GSD algorithms given in [9–11]. We consider the flat-fading MIMO system and overloaded frequency-selective fading group-orthogonal MC-CDMA (GO-MC-CDMA) system (see [18] and therein) to illustrate the merits of our proposed algorithm. All the simulations were performed in MATLAB 7.0. The three algorithms employ the same

conventional SD algorithm for solving any overdetermined or determined box-constrained ILS problems. This conventional SD algorithm uses the V-BLAST column re-ordering strategy (see [6, 8]) in the reduction process and the search algorithm given in [8] in the search process. The complexity of the three algorithms is measured by the number of flops. Only the flops of the search process are counted, since the cost of the reduction process is relatively negligible. For algorithm RSD, the parameter α was set to 1 or σ as in [13], and for algorithm PRSD, α was set to $2^{7/4}\sigma$. For each case, we performed 200 runs and counted the average flops.

4.1 Flat-fading MIMO systems

We constructed the data for a flat-fading MIMO system according to the real model (2). The elements of $\boldsymbol{H} \in \mathbb{R}^{m\times n}$ were drawn from an i.i.d. Gaussian distribution $N(0, 1/2)$, elements of $\boldsymbol{x} \in \mathbb{Z}^n$ were generated independently and uniformly according to the set $\mathcal{X}_k$ and the elements of the noise vector $\boldsymbol{v} \in \mathbb{R}^m$ were drawn from an i.i.d Gaussian distribution $N(0, \sigma^2)$.

Figs. 1–3 show the average flops of the three algorithms against $N_t - N_r$ for 4QAM ($k = 1$), 16QAM ($k = 2$) and 64QAM ($k = 3$), respectively, and in all of these cases $\sigma = 0.05$ and $N_r = 4$. Since it is too time consuming for the 64QAM constellation, the simulations were performed only for $N_t - N_r = 1, 2, 3$. For 4QAM, we see that there is only a small difference between the cost of algorithm RSD (for either $\alpha = 1$ or $\alpha = \sigma$) and the cost of algorithm PRSD. This is because the numbers of columns of $\bar{\boldsymbol{H}}$ in the two overdetermined ILS problems [see (10)] solved by the two algorithms, respectively, are the same when $k = 1$. When $N_t - N_r = 1, 2$, both algorithms RSD and PRSD cost more than algorithm RGSD, and when $N_t - N_r = 3, 4$, the former cost (much) less than the latter. For 16QAM and 64QAM, algorithm PRSD is the most efficient one among the three decoding algorithms. For example, in Fig. 2, when $N_t - N_r = 1$, the average flops costed by algorithms PRSD and RSD with $\alpha = \sigma$ are 1.2×10^3 and 1.3×10^4, that is, Algorithm PRSD is about 11 times as fast as algorithm RSD; and in Fig. 3, when $N_t - N_r = 1$, the average flops costed by algorithms PRSD and RSD with $\alpha = 1$ are 1.1×10^4 and 2.4×10^5, respectively, that is, algorithm PRSD is about 22 times as fast as algorithm RSD. If $N_t - N_r$ is fixed, when k increases, Figs. 1–3 indicate that algorithm PRSD becomes more and more efficient than algorithm RSD. This is because the difference $(k-1)(m-1)$ between the numbers of columns of the matrices $\bar{\boldsymbol{H}}$ in the two overdetermined ILS problems solved by algorithms RSD and PRSD, respectively, becomes larger and larger.

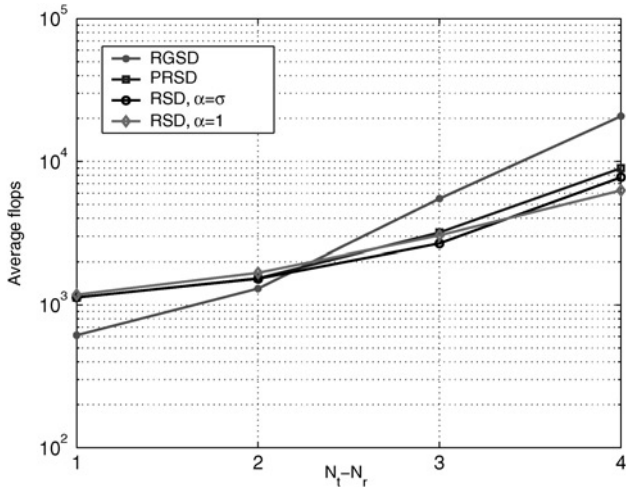

Figure 1 *Average flops against $N_t - N_r$ (4QAM, flat-fading MIMO)*

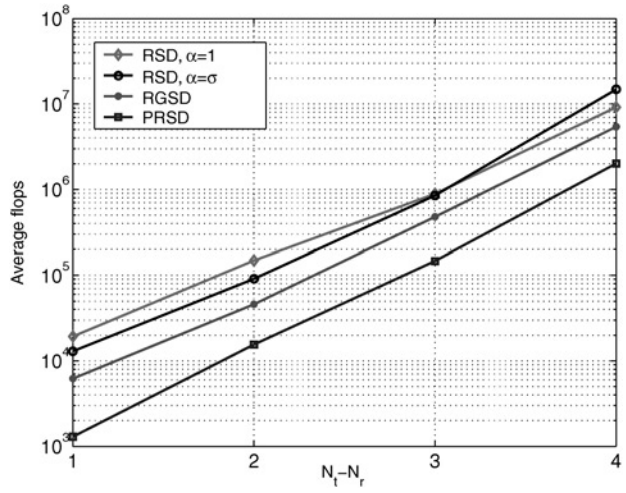

Figure 2 *Average flops against $N_t - N_r$ (16QAM, flat-fading MIMO)*

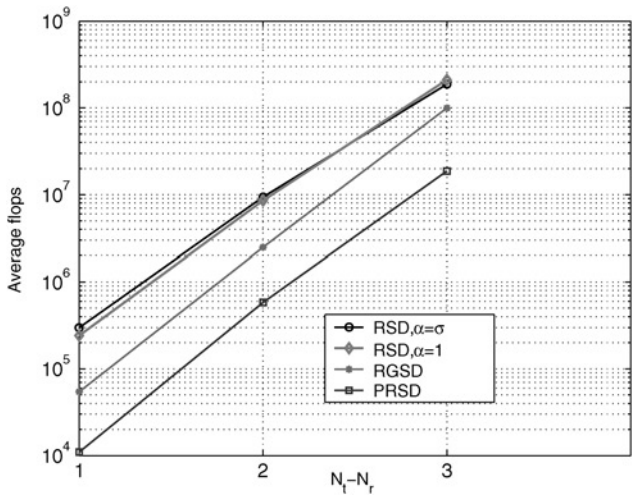

Figure 3 *Average flops against $N_t - N_r$ (64QAM, flat fading MIMO)*

Fig. 4 shows the average flops of the three algorithms against different signal-to-nose-ratios (SNRs) for 16QAM, $N_r = 4$ and $N_t = 5$. Here, for M-QAM, SNR is defined by $\text{SNR} = 10\log_{10}(((M-1)/3)/2\sigma^2)$. When SNR increases,

doi: 10.1049/iet-com:20080210

Fig. 4 indicates that algorithm PRSD becomes more and more efficient than algorithm RSD. For SNR = 12 and 27 dB, the average flops costed by algorithm RSD with $\alpha = \sigma$ is about 7 times and 10 times as high as that by algorithm PRSD, respectively.

In Section 2, we introduced two strategies to improve the efficiency of our approach. One is about the choice of the dimension of $\boldsymbol{x}_2$ and the other is about re-ordering the columns of $\boldsymbol{H}$ to find $\boldsymbol{x}_2$. Fig. 5 shows the effects of these two strategies on efficiency (again here $\sigma = 0.05$ and $N_r = 4$). All the three curves represent the cost of algorithm PRSD. The top curve is for the case that the dimension l of $\boldsymbol{x}_2$ was set to $n - m$ and there was no column re-ordering of $\boldsymbol{H}$ in finding $\boldsymbol{x}_2$; the middle curve is for the case that l was set to $n - m + 1$ and there was no re-ordering in finding $\boldsymbol{x}_2$ either; and the bottom curve is for the case that l was set to $n - m + 1$ and $\boldsymbol{x}_2$ was found by the re-ordering strategy. From Fig. 5, we see that indeed those two strategies can reduce the computational complexity of Algorithm PRSD.

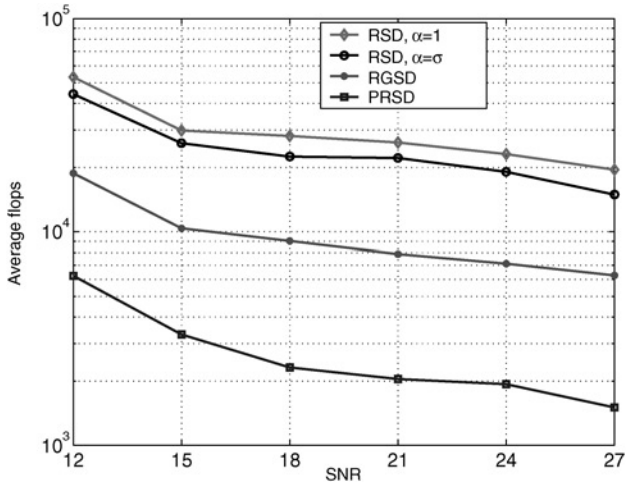

Figure 4 *Average flops against SNR (16QAM, flat-fading MIMO)*

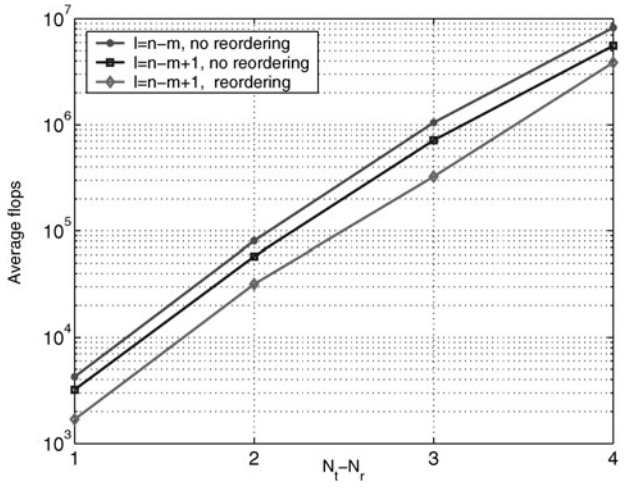

Figure 5 *Average flops against $N_t - N_r$ (16QAM, flat-fading MIMO)*

In order to see how different regularisation parameter α in (10) affect the efficiency of algorithms RSD and PRSD, Fig. 6 shows the costs of these two algorithms against limited α with different settings. Note that the horizontal axis in Fig. 6 uses different scales for different parts. For each part where an optimal α is located, more points are displayed. We observe that algorithm RSD is more sensitive to α than algorithm PRSD when $\alpha > 2$. When $\sigma = 0.02,\ 0.05,\ 0.5$, our choice $\alpha = 2^{7/4}\sigma \simeq 0.07,\ 0.17,\ 1.7$. We see from Fig. 6 that these values are close to the optimal values. This suggests that $\alpha = 2^{7/4}\sigma$ is a good choice. From Fig. 6, we find that for the same setting ($\sigma = 0.05$, 16QAM), algorithm PRSD with $\alpha = 0.17$ costs much less than algorithm RSD even with the optimal α, which appears to be around 0.1.

4.2 Overloaded GO-MC-CDMA systems

The proposed partial regularisation approach was also applied to the overloaded GO-MC-CDMA systems with frequency-selective fading wireless channels. At time index i, the received vector $\boldsymbol{r}(i) \in \mathbb{C}^{N_r}$ for the group of interest is

$$\boldsymbol{r}(i) = \boldsymbol{C}(i)\boldsymbol{D}_a\boldsymbol{b}(i) + \boldsymbol{\eta}(i)$$

where the channel matrix $\boldsymbol{C}(i) \in \mathbb{C}^{N_r \times N_t}$ is decided by two factors: the channel frequency response and the spreading sequences applied, $\boldsymbol{D}_a = \mathrm{diag}(a_1, \ldots, a_{N_t})$ and $a_j \in \mathbb{R}$ is the signal amplitude of user j, $j = 1, \ldots, N_t$ and the complex i.i.d. AWGN vector $\boldsymbol{\eta}(i)$ has N_r elements with zero mean and variance $2\sigma^2$. N_t also indicates the group load. For an overloaded GO-MC-CDMA system, the load N_t can exceed the processing gain N_r to accommodate more active users.

In our simulations, the overloaded GO-MC-CDMA system was in a typical 3-tap SUI-3 [19] fading channel with power profile of [0 −5 −10] db, delay profile of

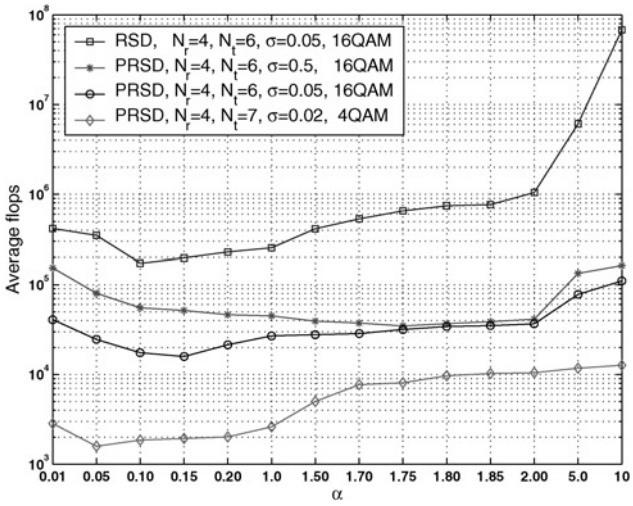

Figure 6 *Average flops against α (flat-fading MIMO)*

[0 0.2 0.4] μs and Rician fading K-factor [1 0 0]. Number of subcarriers per group was set to $N_r = 4$. The sampling frequency was assumed to be 5 MHz for an FFT size of 512. It was assumed that all users had equal-power transmission, that is, $\boldsymbol{D}_a$ was an identity matrix. Two sets of orthogonal carrier interferometry sequences proposed in [20] were used for spreading, one for the first N_r columns of $\boldsymbol{C}(i)$, and the other for the remaining $N_t - N_r$ columns.

Figs. 7 and 8 show the average flops of the three algorithms against $N_t - N_r$ for 4QAM and 16QAM, respectively, and in both cases the symbol energy to noise ratio $E_s/N_0 = 24$ dB. Fig. 7 indicates that for $k = 1$, algorithm PRSD is slightly less efficient than algorithm RSD, but Fig. 8 shows that for $k = 2$, algorithm PRSD is significantly faster than algorithm RSD. Comparing these two figures with Figs. 1 and 2, respectively, we see that there is no significant difference between the flat-fading MIMO system and the overloaded GO-MC-CDMA system with respect to the efficiency of the three decoding algorithms.

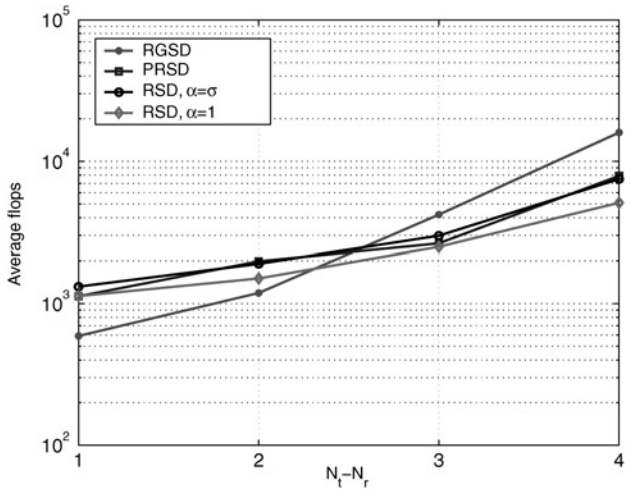

Figure 7 *Average flops against $N_t - N_r$ (4QAM, GO-MC-CDMA)*

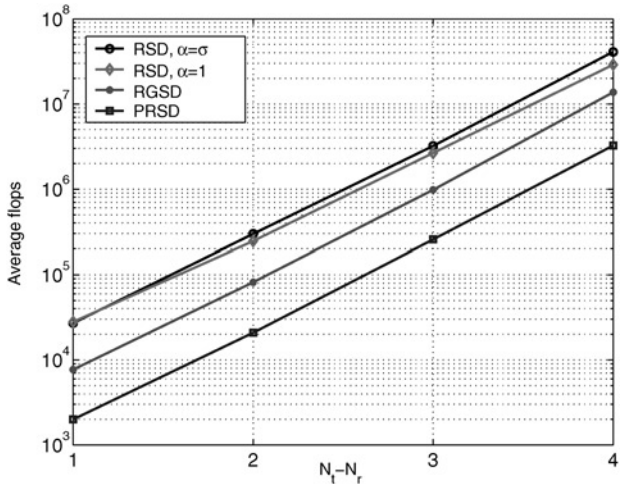

Figure 8 *Average flops against $N_t - N_r$ (16QAM, GO-MC-CDMA)*

5 Conclusions

A PRSD approach has been proposed to improve the efficiency of the RSD approach given in [13] for detection problems in underdetermined linear systems. We showed how to choose some entries from the transmit vector for regularisation. We also suggested an empirical formula for the optimal regularisation parameter. Simulation results on flat-fading MIMO channels and frequency-selective-fading group-orthogonal MC-CDMA systems indicated that our new PRSD approach is much more efficient than the regularisation approach given in [13] and is also more efficient than the RGSD approach presented in [12] for 16QAM and 64QAM constellations.

6 Acknowledgment

This work was supported by NSERC of Canada grant RGPIN217191-07 for the first two authors. Part of this work was presented in IWCMC 2007, Honolulu, 12–16 August 2007.

7 References

[1] VITERBO E., VIKALO H.: 'A universal lattice code decoder for fading channels', *IEEE Trans. Inf. Theory*, 1999, **45**, pp. 1639–1642

[2] DAMEN M., CHKEIF A., BELFIORE J.C.: 'Lattice code decoder for space–time codes', *IEEE Commun. Lett.*, 2000, **4**, (5), pp. 161–163

[3] BRUNEL L., BOUTROS J.: 'Lattice decoding of joint detection in direct-sequence CDMA systems', *IEEE Trans. Inf. Theory*, 2003, **49**, (4), pp. 1030–1037

[4] BRUNEL L.: 'Multiuser detection techniques using maximum likelihood sphere decoding in multicarrier CDMA systems', *IEEE Trans. Wirel. Commun.*, 2004, **3**, (3), pp. 949–957

[5] MURGUAN A.D., EL GAMAL H., DAMEN M., CAIRE G.: 'A unified framework for tree search decoding: rediscovering the sequential decoder', *IEEE Trans. Inf. Theory*, 2006, **52**, (3), pp. 933–953

[6] DAMEN M., EL GAMAL H., CAIRE G.: 'On maximum-likelihood detection and the search for the closest lattice point', *IEEE Trans. Inf. Theory*, 2003, **49**, (10), pp. 2389–2402

[7] VIKALO H., HASSIBI B., KAILATH T.: 'Iterative decoding for MIMO channels via modified sphere decoding', *IEEE Trans. Wirel. Commun.*, 2004, **3**, (6), pp. 2299–2311

doi: 10.1049/iet-com:20080210

[8] CHANG X.-W., HAN Q.: 'Solving box-constrained integer least squares problems', *IEEE Trans. Wirel. Commun.*, 2008, **7**, pp. 277–287

[9] DAMEN M., ABED-MERAIM K., BELFIORE J.: 'Generalized sphere decoder for asymmetrical space-time communication architecture', *Electron. Lett.*, 2000, **36**, pp. 166–167

[10] DAYAL P., VARANASI M.: 'A fast generalized sphere decoder for optimum decoding of under-determined MIMO systems'. Proc. 41st Annu. Allerton Conf. Communication, Control, and Computing, Monticello, IL, October 2003, pp. 1216–1225

[11] YANG Z., LIU C., HE J.: 'A new approach for fast generalized sphere decoding in MIMO systems', *IEEE Signal Process Lett.*, 2005, **12**, (1), pp. 41–44

[12] CHANG X.-W., YANG X.: 'A new fast generalized sphere decoding algorithm for underdetermined MIMO systems'. Proc. 23rd Queen's Biennial Symp. Communications, Kingston, CA, 2006, pp. 18–21

[13] CUI T., TELLAMBURA C.: 'An efficient generalized sphere decoder for rank-deficient MIMO systems', *IEEE Commun. Lett.*, 2005, **9**, (5), pp. 423–425

[14] DAMEN M., EL GAMEL H., CAIRE G.: 'MMSE-GDFE lattice decoding for solving under-determined linear systems with integer unknowns'. Proc. IEEE Int. Symp. Inform. Theory, Chicago, USA, June–July 2004

[15] WANG P., LE-NGOC T.: 'A low complexity generalized sphere decoding approach for underdetermined MIMO systems'. Proc. IEEE Int. Conf. Communications, June 2006, pp. 4266–4271

[16] CORMEN T.H., LEISERSON C.E., RIVEST C.: 'Introduction to algorithms' (MIT Press, 2001)

[17] AGRELL E., ERIKSSON T., VARDY A., ZEGER K.: 'Closest point search in lattices', *IEEE Trans. Inf. Theory*, 2002, **48**, (8), pp. 2201–2214

[18] WANG P., LE-NGOC T.: 'An efficient multi-user detection scheme for overloaded group-orthogonal MC-CDMA systems'. IEEE ISWPC07, February 2007, pp. 280–284

[19] ERCEG V., HARI K., SMITH M.S.: 'Channel models for fixed wireless applications'. IEEE 802.16.3c-01/29r4, July 2001

[20] NATARAJAN B., WU Z., NASSAR C.R., SHATTIL S.: 'Large set of CI spreading codes for high-capacity MC-CDMA', *IEEE Trans. Commun.*, 2004, **52**, (11), pp. 1862–1866

[21] GOLUB G.H., VAN LOAN C.F.: 'Matrix computations' (The Johns Hopkins University Press, Baltimore, MD, 1996, 3rd edn.)

8 Appendix: Basics of GSD algorithms

To solve the underdetermined problem (3), we compute the QR factorisation of the matrix $\boldsymbol{H}$: $\boldsymbol{H} = \boldsymbol{Q}\boldsymbol{R}$ where $\boldsymbol{Q} \in \mathbb{R}^{m\times m}$ is orthogonal and $\boldsymbol{R} \in \mathbb{R}^{m\times n}$ is an upper trapezoidal matrix. This QR factorisation can be computed by the Householder transformations, (see, e.g. [21, Section 5.2]). Then with $\tilde{\boldsymbol{y}} \triangleq \boldsymbol{Q}^{\mathrm{T}}\boldsymbol{y}$, the objective function in (3) can be written as

$$\|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|_2^2 = \|\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{y} - \boldsymbol{R}\boldsymbol{x}\|_2^2 = \|\tilde{\boldsymbol{y}} - \boldsymbol{R}\boldsymbol{x}\|_2^2$$

So the problem (3) is transformed to

$$\min_{\boldsymbol{x}\in\mathcal{X}_k^n} \|\tilde{\boldsymbol{y}} - \boldsymbol{R}\boldsymbol{x}\|_2^2$$

The above process is called reduction or preprocessing.

After the above process, a GSD algorithm tries to find the optimal integer point over a hyper-ellipsoid

$$\|\tilde{\boldsymbol{y}} - \boldsymbol{R}\boldsymbol{x}\|_2^2 < \beta^2 \tag{11}$$

Note that this is a hyper-sphere with radius β in terms of $\boldsymbol{R}\boldsymbol{x}$. A good choice of β was proposed in [12].

Partition the upper trapezoidal matrix $\boldsymbol{R} \in \mathbb{R}^{m\times n}$ and the vector $\boldsymbol{x} \in \mathbb{R}^n$ as

$$\underset{m}{\boldsymbol{R}} = [\underset{m}{\boldsymbol{R}_1}, \underset{n-m}{\boldsymbol{R}_2}], \quad \boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}^{(1)} \\ \boldsymbol{x}^{(2)} \end{bmatrix} \begin{matrix} m \\ n-m \end{matrix} \tag{12}$$

Notice that $\boldsymbol{R}_1$ is non-singular upper triangular. Then, we have

$$\min_{\boldsymbol{x}\in\mathcal{X}_k^n} \|\tilde{\boldsymbol{y}} - \boldsymbol{R}\boldsymbol{x}\|_2^2 = \min_{\boldsymbol{x}^{(2)}\in\mathcal{X}_k^{n-m}} \left(\min_{\boldsymbol{x}^{(1)}\in\mathcal{X}_k^m} \|(\tilde{\boldsymbol{y}} - \boldsymbol{R}_2\boldsymbol{x}^{(2)}) - \boldsymbol{R}_1\boldsymbol{x}^{(1)}\|_2^2 \right) \tag{13}$$

The GSD algorithm in [9] first fixed $\boldsymbol{x}^{(2)}$, then employs a conventional SD to solve the bracketed minimisation problem in (13). This is done by exhaustively trying every possible $\boldsymbol{x}^{(2)}$. Thus the GSD algorithm in [9] has an exponential complexity in $n-m$ independent of the SNR. Obviously, it is time consuming to try each possible candidate for $\boldsymbol{x}^{(2)}$. Motivated by this observation, Dayal and Varnasi [10], Yang *et al.* [11] and Chang and Yang [12] proposed different strategies to improve the efficiency.

Instead of partitioning (12), in [10, 12], the vector $\tilde{\boldsymbol{y}} \in \mathbb{R}^m$, the upper trapezoidal matrix $\boldsymbol{R} \in \mathbb{R}^{m\times m}$ and the

vector $\boldsymbol{x} \in \mathbb{R}^n$ are partitioned as follows

$$\bar{\boldsymbol{y}} = \begin{bmatrix} \bar{\boldsymbol{y}}^{(1)} \\ \bar{y}_m \end{bmatrix} \begin{matrix} m-1 \\ 1 \end{matrix}, \quad \boldsymbol{R} = \begin{bmatrix} \boldsymbol{R}_1 & \boldsymbol{R}_2 \\ \underset{m-1}{\mathbf{0}} & \underset{n-m+1}{\boldsymbol{r}^{\mathrm{T}}} \end{bmatrix} \begin{matrix} m-1 \\ 1 \end{matrix},$$

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}^{(1)} \\ \boldsymbol{x}^{(2)} \end{bmatrix} \begin{matrix} m-1 \\ n-m+1 \end{matrix} \tag{14}$$

Let $N \triangleq n - m + 1$. Then, we have

$$\min_{\boldsymbol{x} \in \mathcal{X}_k^n} \|\bar{\boldsymbol{y}} - \boldsymbol{R}\boldsymbol{x}\|_2^2 = \min_{\boldsymbol{x}^{(2)} \in \mathcal{X}_k^N} [(\bar{y}_m - \boldsymbol{r}^{\mathrm{T}}\boldsymbol{x}^{(2)})^2 + \min_{\boldsymbol{x}^{(1)} \in \mathcal{X}_k^{m-1}} \|(\bar{\boldsymbol{y}}^{(1)} - \boldsymbol{R}_2\boldsymbol{x}^{(2)}) - \boldsymbol{R}_1\boldsymbol{x}^{(1)}\|_2^2] \tag{15}$$

Notice that if $\boldsymbol{x}^{(2)}$ is fixed, one can solve the corresponding ILS sub-problem within the brackets in (15) to obtain the optimal $\boldsymbol{x}^{(1)}$.

Suppose that (11) holds, then it follows that

$$(\bar{y}_m - \boldsymbol{r}^{\mathrm{T}}\boldsymbol{x}^{(2)})^2 = \left(\bar{y}_m - \sum_{j=m}^{n} r_{mj}x_j\right)^2 < \beta^2 \tag{16}$$

This inequality is crucial in the GSD algorithms given in [10–12]. By using this inequality, only part of possible candidates for $\boldsymbol{x}^{(2)}$ will be enumerated. Yang *et al.* [11] derived a series of inequalities for each entry of $\boldsymbol{x}^{(2)}$ based on (16). From these inequalities, a slightly modified SD algorithm is employed to obtain a possible $\boldsymbol{x}^{(2)}$. Dayal and Varanasi [10] and Chang and Yang [12] partitioned the candidate set for $\boldsymbol{x}^{(2)}$ into disjoint ordered subsets. Then by using the inequality (16), only part of the disjoint subsets need to be enumerated to obtain possible $\boldsymbol{x}^{(2)}$. The GSD algorithm given in [12], which incorporates a column re-ordering strategy and has recursive nature, can reduce more the number of the sub-ILS problems within the brackets in (15) which need to be solved than the GSD algorithm given in [10]. Our simulation results showed that the GSD algorithm given in [12] is faster than those given in [10, 11].

IET Commun., 2009, Vol. 3, Iss. 1, pp. 17–24
doi: 10.1049/iet-com:20080210