

Fast Multi-user Detector for a Time-varying CDMA System

Suman Das and Joseph R. Cavallaro

Department of Electrical and Computer Engineering, Rice University
MS 366, Houston, TX 77251-1892 USA

ABSTRACT

This paper investigates methods to reduce the amount of computation needed to detect information bits using a linear detector for a CDMA system. We show windowing technique coupled with pipelining can reduce the amount of computation without significantly sacrificing the performance of linear feedback detector. We also describe efficient techniques to adapt to a dynamic system where the system parameters vary due to the change in delays associated with individual users.

Keywords: CDMA detector, Cholesky decomposition

1. INTRODUCTION

Code-Division Multiple-Access (CDMA), is steadily developing as the driving technology behind the wireless communication industry because of its superior channel utilization capacity. In the CDMA system each user is assigned a code sequence, which is used for the modulation of bits to be transmitted. The receiver receives a signal, which is the composite sum of the modulated information bits, transmitted by all the users and tries to estimate the information bits from this received signal sequence. However, with the increase in the number of users that the channel can support, there is a corresponding increase in the volume of data that needs to be processed by the receiver for successful retrieval of information bits from the received signal. Since CDMA is used to support voice data, real time processing of these data to extract the information bits is essential.

Previous signal processing studies on detection of data corrupted by background noise suggest, that the received signal should be passed through a matched filter to extract the information. If the received signal contains data from only one user, then this filtering operation can provide the best estimate of the data. This is also not a computationally complex operation, as real time digital and analog matched filters can be designed very easily. In a multiuser environment, the intuitive idea is to use a bank of filters matched to each individual user code. Although this filtering technique is computationally very simple, the performance of this type of detector in multiuser scenario is very poor. Verdu demonstrated¹ that the *optimal* multiuser detector can far outperform this conventional detector. However, the computational complexity of this detector is exponential in the size of the data as well as the number of users.

Thus it is evident that there is a tradeoff between the performance of the detector and the computational complexity that we can incur and still maintain real time response. This has led to the development of a number of suboptimal linear multiuser detectors which try to maintain a balance between these two approaches. Most of these linear detectors need to solve a system of linear equations involving the correlation matrix associated with the user code. Direct detection techniques require that we compute the inverse of the correlation matrix. However in an asynchronous system, the size of the matrix is NK where K is the number of users and N is the size of the data block and any direct method will involve $O((NK)^3)$ operations. For a large block length the computational effort can be significant and can prohibit real time response. In this paper we have shown that with careful windowing techniques coupled with pipelining, the computational effort can be reduced without sacrificing the performance quality significantly. Moreover, when the system parameters vary, updating algorithms may be a better alternative than fresh recomputation, under certain conditions, to adapt to the dynamic nature of the system.

The paper is organized as follows. In the following section we describe the system model. We then introduce the basic idea of linear feedback detector and show how windowing and pipelining can help in reducing the computation. Subsequently we describe our updating algorithm for this kind of detector and finally we present some concluding remarks.

Other author information: (Send correspondence to Suman Das)

Suman Das Email: suman@rice.edu; Telephone: 713-527-8750 x2899; Fax: 713-524-5237

2. SYSTEM DESCRIPTION

We will assume that there are K users transmitting data to the base station over a period of time. The k^{th} user is assigned a signature waveform $s_k(t)$ which extends over the symbol period $[0, T]$ and takes the value zero beyond it. Throughout our discourse, we will assume that the modulation of the signature waveform is linear. In other words, if the data stream of the k^{th} user is

$$b_k^t = [b_k(1), b_k(2), \dots, b_k(N)],$$

then the signal transmitted for this user is given as

$$A_k \sum_{i=1}^N b_k(i) s_k(t - iT),$$

where A_k is the received signal amplitude. Without loss of any generality we have implicitly used in the above relation the fact that the signature waveforms have unit energy and we will also assume that the transmitted bits are *antipodal* i.e. $b_k(i) \in \{-1, 1\}$.

In our system each user sends a data block of length N . In the asynchronous case there will be a delay associated with each user and let the delay of the k^{th} user be given by τ_k . The receiver receives the signal which is the composite sum of the signals transmitted by all the users and can be represented as

$$r(t) = \sum_{k=1}^K \sum_{i=1}^N A_k b_k(i) s_k(t - iT - \tau_k) + n, \quad (1)$$

where n is the background Additive White Gaussian Noise.

If this received signal is passed through a bank of code matched filters then the output of the filter bank can be expressed² in the vector form as

$$y = R_N b + \eta, \quad (2)$$

where R_N is the code-correlation matrix of the form

$$R_N = \begin{bmatrix} R(0) & R^t(1) & 0 & \dots & 0 & 0 \\ R(1) & R(0) & R^t(1) & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \dots & 0 & R(1) & R(0) & R^t(1) \\ 0 & 0 & \dots & 0 & R(0) & R^t(1) \end{bmatrix},$$

with each individual block being defined as

$$R_{kl}(m) = \int A_k s_k(t - \tau_k) A_l s_l(t + mT - \tau_l) dt.$$

In the symbol-synchronous case the correlation matrix takes a simpler form and the output of the bank of matched filters can be expressed as

$$y = \rho A b + \eta, \quad (3)$$

with the scalar elements of the correlation matrix given by

$$\rho_{ij} = \int s_i(t) s_j(t) dt.$$

It has been shown (see^{2,3}) that the outputs of a bank of K matched filters provide sufficient statistic to estimate the information bits

$$b = [b^t(1), b^t(2), \dots, b^t(N)]^t.$$

Most of the linear detectors try to solve some variation of this linear equation to detect the transmitted bits.

2.1. CDMA detectors

In the previous section we have seen that to support personal voice telephony using CDMA, the receiver should be capable of decoding the transmitted bits in real time. Since the optimal detector requires exponential computational effort and the matched filter suffers from significant performance degradation the suboptimal linear multiuser detectors are the obvious choice.

We have also seen that most of these suboptimal detectors try to solve the linear system described by the correlation matrix. The naive way of solving a linear system of equations requires computations cubic in the size of the matrix and for an asynchronous CDMA system the size of the correlation matrix is NK . Thus this simple method may not be computationally feasible for a large system.

The linear detectors apply a linear transformation on the output of the bank of matched filters and we can consider this transformation equivalent to a filtering operation. For example, the inversion of the correlation matrix is done by a decorrelating filter. In an ideal implementation, the memory length of these filters equals the data packet length, which can potentially approach infinity. Not only does it necessitate unwieldy computation, but also unacceptable system response time to aggregate the data and then detect them. One approach is to limit the block size of the data by leaving certain transmission intervals unused. This is achieved by the insertion of isolation bits.⁴ However this implies that the channel is not utilized to its maximum capacity and this defeats one of the main purposes of CDMA.

In their paper⁵ Juntti et.al. showed that the amount of computation needed to estimate the information bits can be curbed down by approximating the infinite memory length filter by finite memory length linear multi-user detectors. In the following sections we will describe their idea in a greater detail. We will also extend the idea to the linear feedback type of detectors. We will show that greater performance benefits can be achieved by using this approximation on a linear feedback detector. We will also suggest some modifications to improve the performance. We will formalize these ideas to a new detection scheme which far outperforms both the windowed version of decorrelating and linear feedback detector.

Another issue in the design of the multiuser detector is the update of the correlation matrix. In a practical scenario, the delays of the users will change continuously. And this will change the correlation matrix. This means that we need to periodically update the system we plan to solve. There are two approaches to solve the problem:

- Recompute the solution afresh,
- Update the solution based on the perturbations of the system.

We will describe methods to update the system and identify threshold points to decide between the two options.

3. LINEAR FEEDBACK DETECTOR

The suboptimal multiuser detectors try to estimate the informations bits by solving some variation of eqn (2). In the decorrelating detector this estimate is obtained by taking the inverse of a correlation matrix. However this is not the unique way of achieving the solution.

In a multiuser environment, there are two causes of performance degradation - the background noise and the MAI. We have no control over the background noise; however we can combat the MAI by using the information of the estimated signals of other users in detection of data bits. Now under the assumption that the bit sequence b was sent, the bit error probability for the i^{th} user in case of the conventional detector is given by²

$$P_i(b) = Q \left(\sqrt{\frac{A_i - \sum_{j \neq i} A_j \rho_{ij} b_j}{\sigma}} \right),$$

where

$$Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt$$

is the error function and ρ_{ij} is the signature correlation between the user i and j and σ is the noise power spectral density. It is evident that this error depends on the relative signal strength of the i^{th} user (A_i) with respect to the other users. Thus it can be said that the user with stronger signal strength will have less bit-error probability than the user with weaker signal strength. Thus if the user signals are of unequal strength, then the user with the strongest signal will be least affected by the signals from the other users and the information bits can be detected with least amount of error compared to other users. Now if we use the information of the signal of this strongest user in the detection of the other users, then we can reduce the bit error probability for the weaker users. The weaker users can benefit from the knowledge of the estimated signal contributed by the strongest user to improve the detection of their data bits. This idea has led to the development of the linear feedback detector.⁶ In the symbol-synchronous case the following steps are performed:

- Detect, with the conventional detector, the bit sent by the strongest user,
- Regenerate an estimate of the received signal for the strongest user using:
 - Estimated data,
 - The knowledge of the signature waveform,
 - Estimate of the delay and amplitude of the user,
- Remove this signal estimate from the received signal to estimate the data bit for the next strongest user and so on.

Mathematically, the users are arranged in the descending order of their signal strength,

$$A_1 \geq A_2 \geq \dots \geq A_K.$$

Since the correlation matrix is positive definite, we can decompose it into a product of lower and upper triangular matrices using Cholesky decomposition⁷ and rewrite eqn (3) as,

$$\rho Ab = L^t L A b = y - \eta,$$

where L is a lower triangular matrix. Pre-multiplying both sides by L^{-t} we get

$$\tilde{y} = L^{-t} y = L A b + \tilde{n}. \quad (4)$$

Expanding the individual equations, we get the following iteration scheme for the first two users

$$\begin{aligned} \tilde{y}_1 &= L_{11} A_1 b_1 \\ \tilde{y}_2 &= L_{22} A_2 b_2 - L_{21} A_1 \tilde{b}_1. \end{aligned}$$

And in general for the i^{th} user

$$\tilde{y}_i = L_{ii} A_i b_i - \sum_{j=1}^{i-1} L_{ij} A_j \tilde{b}_j,$$

where the previously estimated data bits are given by $\tilde{b}_i = \text{sgn}(\tilde{y}_i)$.

We will extend this idea to the asynchronous case. However in the asynchronous system, the i^{th} bit for a particular user is not only affected by the same bits from other users, but the adjacent bits of other users can also corrupt the signal. This is evident from the figure 1. Here the first data bit of user 3 is affected by the first and second bit of user 1 and the first bit of user 2. This also leads to some other problems which are absent in the synchronous system.

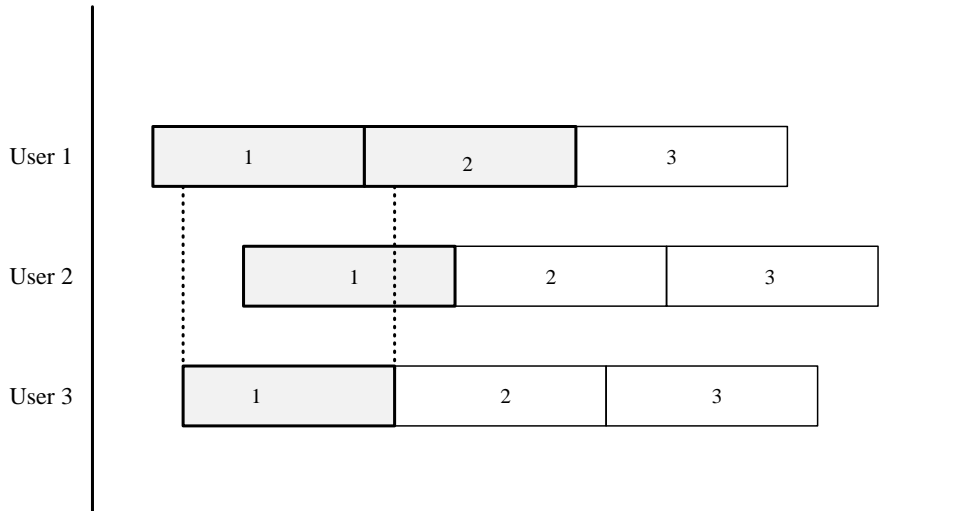


Figure 1. Timing diagram of an asynchronous channel

4. WINDOWING OF DATA

In the asynchronous case the one shot output of matched filter is no longer a sufficient statistic for multiuser detection. We have to consider the output of the matched filters extending over the entire data packet. If the size of the data packet is too large the processing of this enormous volume of data may be computationally expensive.

In an ideal infinite memory length filter the detector receives the huge data sequence and tries to detect the entire information simultaneously. To reduce the computational effort and delay we may divide the received signal sequence into manageable segments and try to extract the information bits from these segments. If the information sequence is N bits long, in this approach,⁵ the vector is partitioned into P bit long windows of data and they are decoded independently.

Thus instead of considering an $NK \times NK$ correlation matrix we will consider a windowed version of it. The penalty of taking a windowed version of the correlation matrix is that the performance of this detector degrades due to the edge effect. Since in the windowed version of decorrelating detector, with window size P , we are not considering the full effect of adjacent bits from other users for the first and the P^{th} bits, the probability of error in detecting these bits will be higher. As the window size is increased the edge effect progressively becomes less significant, but an increase in data size will increase the computation. So a trade-off between these two factors is necessary. Juntti et.al. have found⁵ through simulations that for the *decorrelating detector* with reasonable number of users, a small window size gives satisfactory performance.

We have conducted a similar study (fig 2) for the linear feedback detector case. We have used 15 users and modulated each user with gold codes of length 31. The results show that for a window size 5-8 we almost approach the same performance as the infinite window detector. We have used the result of decorrelating detector for the infinite memory length detector.

5. PIPELINED CHOLESKY ALGORITHM

In the asynchronous multiuser system the i^{th} bit of the k^{th} user can be affected by the $i - 1^{th}$, i^{th} and the $i + 1^{th}$ bits of other users and hence any windowed version of the linear multiuser detector must have a correlation matrix of size $3K \times 3K$. However there is one benefit in using the linear feedback detector over the decorrelating detector. If the estimated bits of the previous users (used for feedback) are correct then the linear feedback detector can outperform the decorrelating detector. In the matched filter output the probability of error for the k^{th} user is given by

$$P(e|b) = Q\left(\frac{A_k b_k - \sum_{i \neq k} A_i \rho_{ik} b_i}{\sigma}\right).$$

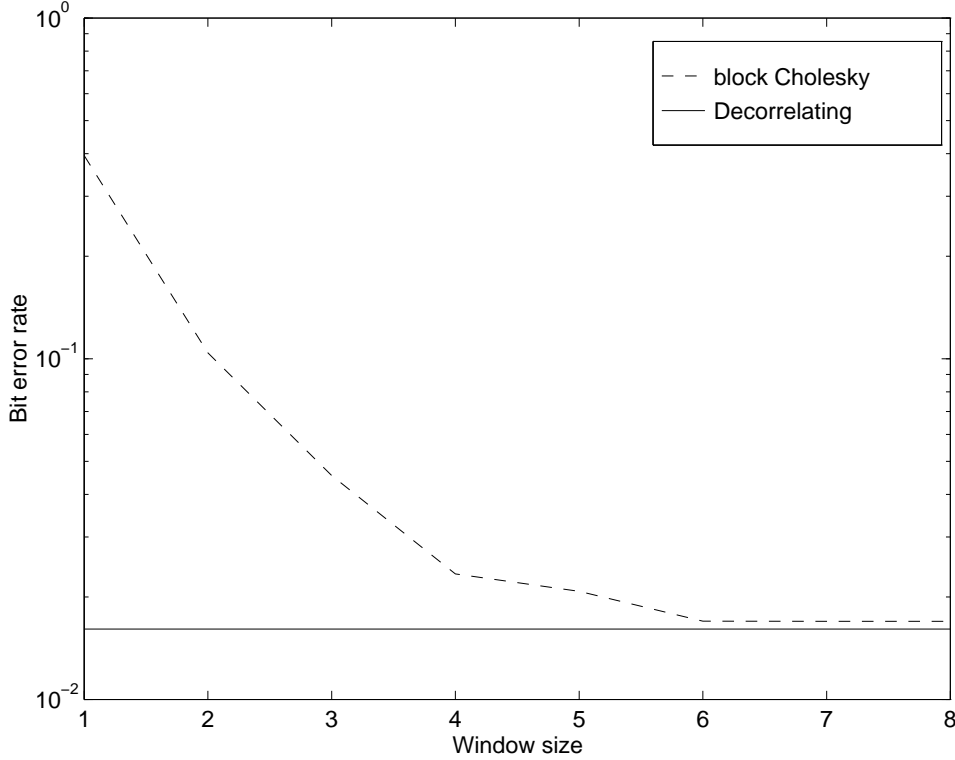


Figure 2. Performance of windowed linear feedback detector

If we feedback the correct information for the users $[1 \cdots (k-1)]$, then the probability of error reduces to

$$P(e|b) = Q\left(\frac{A_k b_k - \sum_{i=k+1}^K A_i \rho_{ik} b_i}{\sigma}\right).$$

These are the results in the synchronous case. Similar results are available for the asynchronous system too. Thus the linear feedback detector can outperform the conventional detector. The comparative simulation study shows that it can outperform the decorrelating detector too. In this simulation we have plotted the bit-error rate of the 4th weakest user out of 15 users for both the decorrelating and linear-feedback detector in the asynchronous case.

We exploited this observation as the motivation for the design of our pipelined detector. However, another important insight can be obtained if we look at the breakdown of the error. We did a simulation with block length 3 to study of the error distribution for linear feedback detector. It (4) shows that the major contribution to the error comes from the two edge blocks and the middle K data bits are relatively error free.

However this feature is not so glaring in case of the decorrelating detector. Thus if we take a block length of 3 to detect only the middle K bits then these middle K data bits will be relatively error free. It should also be noted that the error estimate for the middle K data blocks is pessimistic since it is influenced by the estimation of data bits in the first block, which are estimated with a higher probability of error. However if in the second step we use the estimated middle K blocks for the estimation of the next K blocks the error due to feedback of wrongly estimated data will be reduced. Thus the detector will have improved performance due to two effects - detected bits are the middle K bits for the windowed feedback detection scheme which are relatively error free and the use of their information will improve performance in the estimation of subsequent bits. This observation leads to a natural pipelined scheme of detection:

- Use the data block extending over the $(i-1)^{th}$, i^{th} and $(i+1)^{th}$ data bits to detect the i^{th} information bits.

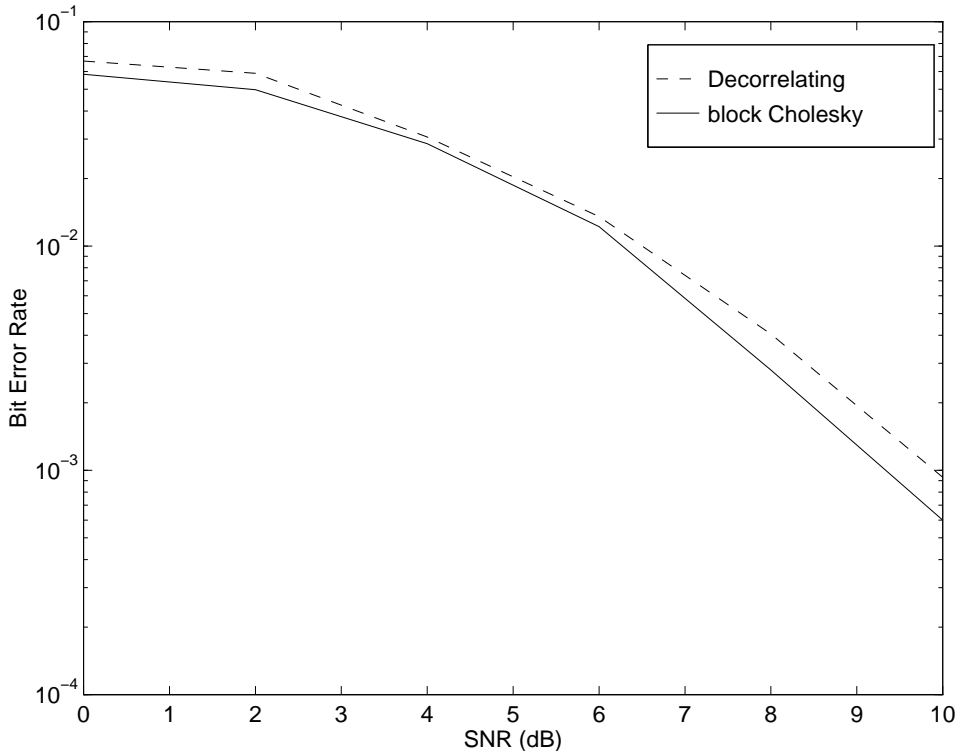


Figure 3. Comparative study of the decorrelating and linear feedback detector

- In the next stage for the detection of the $(i + 1)^{th}$ bits we use the information of the i^{th} and $(i + 2)^{th}$ bits.

However since after Cholesky decomposition we end up with a lower triangular matrix similar to eqn (4), we need information from the i^{th} bits only after this transformation for the estimation of the $(i + 1)^{th}$ bit. Since the i^{th} bits are estimated relatively error free, the feedback of this information reduces the probability of error for the subsequent bit estimation.

5.1. Simulation Results

We have performed extensive simulation with the three schemes. We have compared the result of our pipelined Cholesky algorithm with the windowed version of decorrelating detector and the windowed version of the linear feedback detector (where the data over entire window are detected simultaneously). The window size is three. For simulation we have used 15 users with each user being modulated by Gold Code sequences of length 30. The energies of the users were assigned randomly, with the energy of the strongest user being 20dB stronger than the weakest user. The delays for the users were chosen randomly. For comparison we have also plotted the error in case of the ideal infinite memory length decorrelating detector.

As expected, the simulation shows that both the decorrelating and linear feedback scheme with window size of 3 perform much worse than the ideal non-windowed version of the decorrelating detector. This is in accordance with the findings of Juntti et.al.⁵ However the pipelined Cholesky scheme significantly outperforms both these detectors. Even though it did not match the bit error rate of the ideal decorrelating detector, the performance degradation is not so significant. Thus using the proposed pipelined Cholesky algorithm we can get the performance comparable to a decorrelating detector at a much lesser computational complexity.

6. UPDATING ALGORITHM

In a mobile communication environment, the delays of the various user change continuously. Other system parameters like the amplitude of the users can also vary with time. Since the correlation matrix is directly determined by these

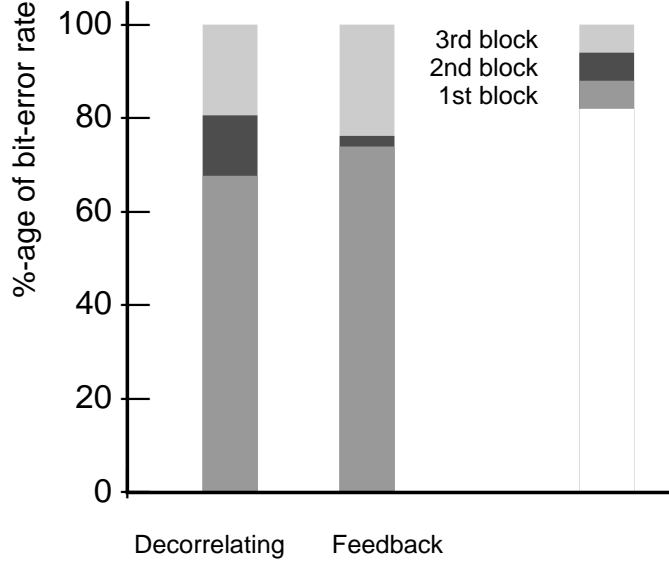


Figure 4. Split of error in the decorrelating and linear feedback detection scheme

system parameters, this means that the system we need to process is dynamic. When the changes in these parameters become significant, assumption of a static correlation matrix may lead to an erroneous detection process. There are two methods to combat this dynamic system,

- Recompute the detectors whenever the system parameters vary,
- Try to update the correlation matrix from the previous realization and update the detectors accordingly.

The choice depends on the number of parameters varying, i.e the degree of perturbation. Whenever the parameters of the system change it will affect the correlation matrix corresponding to the system. We can write this perturbed system in the following form

$$\tilde{R}_N = R_N + X,$$

where X is the perturbation matrix. In the following sections we will bound the effort of updating the system and its inverses based on the rank of X . Though the change in amplitude also affects the correlation matrix it is not as dramatic as the perturbation due to change in the delays. Hence we will limit our discussions to variations in the delays.

LEMMA 61. *If the rank of the correlation matrix is NK , where N is the block length of data and K is the number of users, and if the delay of only one user changes then the rank of the perturbation matrix cannot exceed $2N$.*

Proof. Let the delays of the perturbed system be given by $\tilde{\tau}$ and let the delay of the i^{th} user change; i.e. $\tilde{\tau}_i \neq \tau_i$. We can write the perturbed correlation matrix as

$$\tilde{R}_N = \begin{bmatrix} \tilde{R}_a(0) & \tilde{R}_a^t(1) & 0 & \cdots & 0 & 0 \\ \tilde{R}_a(1) & \tilde{R}_a(0) & \tilde{R}_a^t(1) & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & \tilde{R}_a(1) & \tilde{R}_a(0) & \tilde{R}_a^t(1) \\ 0 & 0 & \cdots & 0 & \tilde{R}_a(0) & \tilde{R}_a^t(1) \end{bmatrix},$$

and each of the sub-block can be written as $\tilde{R}_a(i) = R_a(i) + X(i)$.

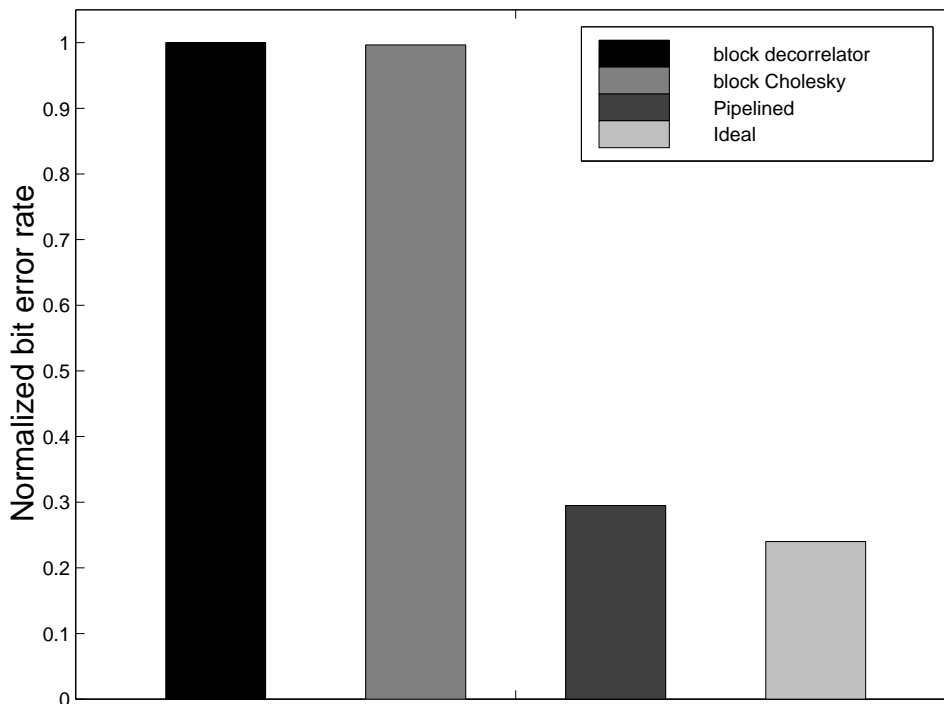


Figure 5. Performance evaluation of the three detection scheme

Now the entries for the sub-blocks are given by

$$\tilde{R}_{kl}(m) = \int A_k s_k(t - \tilde{\tau}_k) A_l s_l(t + mT - \tilde{\tau}_l) dt$$

and if only $\tilde{\tau}_i$ varies from τ_i only the i^{th} row and columns of the sub-blocks $\tilde{R}_a(i)$ will differ from the unperturbed sub-blocks. Thus the matrix X will have the same block-Toeplitz structure as the original matrix R_N with nonzero entries corresponding to the i^{th} rows and columns of the sub-blocks. The figure 6 shows the shape of the matrix X when $N = 5$, $K = 6$ and the third user undergoes a change in its delay.

Now we need to find the rank of this perturbation matrix X . We know that if a matrix is written as the sum of two matrices then the rank of the matrix cannot exceed the sum of the ranks of the two matrices. We can write the perturbation matrix X as the sum of two matrices,

$$X = X_{col} + X_{row},$$

where the nonzero rows of X_{row} corresponds to the i^{th} , $(K + i)^{th}$, $(2k + i)^{th}$, \dots rows of the matrix X and similarly X_{col} consists of the similar columns of X . We split the $(j, j)^{th}$ elements of X , where $j \in [i, K + i, 2K + i, \dots]$, and distribute them equally to X_{row} and X_{col} . Now since X is a symmetric matrix this makes

$$X_{col} = X_{row}^t \quad (5)$$

The rank of a matrix is given by the number of independent rows of that matrix and the number of independent rows cannot exceed the number of nonzero rows of that matrix. Thus the rank of X_{row} cannot exceed N . Also since the rank of a matrix and its transpose are same, the rank of X_{col} cannot exceed N either. And hence the rank of the perturbation matrix X cannot exceed $2N$. \square

As a *corollary* to the above Lemma we can easily show that when the delays of k users vary with $k \leq K/2$ then the rank of the perturbation matrix is at most $2kN$

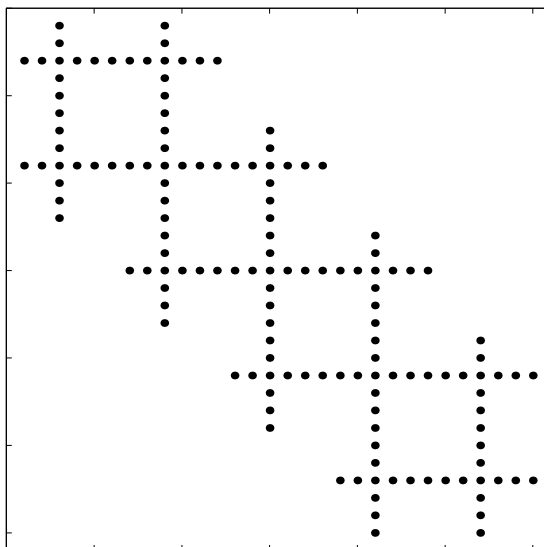


Figure 6. Shape of the perturbation matrix

6.1. Updating the Linear Feedback Detector

If $X = \tilde{R}_N - R_N$ is a rank m perturbation of the correlation matrix R_N of size NK . Let us also assume that the original system $R_N b = y$ is solved via Cholesky decomposition with $R_N = LDL^t$, where L is a lower triangular matrix with unit diagonal and D is a diagonal matrix. If we can express the symmetric matrix

$$X = Y\Sigma Y^t,$$

where Y is an $n \times m$ matrix and C is $m \times m$, then our aim is to find the Cholesky decomposition of the new system \tilde{R}_N . The perturbed system can be expressed as

$$\tilde{R}_N = LDL^t + Y\Sigma Y^t = \tilde{L}\tilde{D}\tilde{L}^t,$$

where \tilde{L} is the updated lower triangular matrix and \tilde{D} is the updated diagonal matrix.

A number of methods have been proposed to compute the updates of the Cholesky decomposition. We will consider Bennett's method⁸ for updating the lower triangular and diagonal system and will analyze the cost of the update. We will also assume that we are considering a windowed correlation matrix of window size 3 (as described for the pipelined Cholesky Algorithm). We will also restrict our analysis for the case where the perturbation in the correlation matrix is due to the change in the delay of only one (i) user.

LEMMA 62. *The update of the Cholesky decomposition of the windowed correlation matrix R_N can be done in $O(K^2)$ steps where K is the number of users.*

Proof. We first describe a method to obtain the factorization

$$X = Y\Sigma Y^t$$

for the perturbation matrix without any costly computation. X has a block Toeplitz structure with its elements $X_{j,k} \neq 0$ if and only if either the row or the column index is in $[i, 2K + i, 3K + i]$. Now we can rearrange this matrix such that all the nonzero elements have the column or the row index in $[1-3]$. Let this matrix be \hat{X} ,

$$X = P^t \hat{X} P,$$

where P is the desired permutation of the rows and columns to satisfy the above criteria. The matrix \hat{X} is also symmetric since the original matrix X is also symmetric and permutation preserves the symmetric property.

We can express this new matrix as

$$\hat{X} = \begin{pmatrix} T & L^t \\ L & 0 \end{pmatrix},$$

where T is a 3×3 symmetric matrix, and L is a $3(K-1) \times 3$ matrix.

This matrix can be expressed as

$$\begin{aligned} \begin{pmatrix} T & 0 \\ L & L \end{pmatrix} \begin{pmatrix} T^{-1} & 0 \\ 0 & -T^{-1} \end{pmatrix} \begin{pmatrix} T & L^t \\ 0 & L^t \end{pmatrix} &= \begin{pmatrix} I & 0 \\ LT^{-1} & -LT^{-1} \end{pmatrix} \begin{pmatrix} T & L^t \\ 0 & L^t \end{pmatrix} \\ &= \begin{pmatrix} T & L^t \\ L & LT^{-1}L^t - LT^{-1}L^t \end{pmatrix} \\ &= \hat{X}. \end{aligned}$$

Hence we can write the perturbed matrix as

$$X = \underbrace{P^t}_{Y} \begin{pmatrix} T & 0 \\ L & L \end{pmatrix} \underbrace{\begin{pmatrix} T^{-1} & 0 \\ 0 & -T^{-1} \end{pmatrix}}_{\Sigma} \underbrace{\begin{pmatrix} T & L^t \\ 0 & L^t \end{pmatrix}}_{Y^t} P.$$

This decomposition requires a trivial calculation of the inverse of a 3×3 matrix.

Now the Bennett's method proceeds by first finding the first column of the updated lower triangular matrix \tilde{L} and the first element \tilde{D}_{11} . The computation of each element of the updated lower triangular matrix requires a matrix vector multiplication involving the matrix Σ and requires m steps, where m is the size of the matrix. Thus the first step requires $O(mn)$ computation, where n is the length of the first column of \tilde{L} . After the first step we are left with a matrix of size $(n-1) \times (n-1)$ which we need to update recursively. So if the whole computation takes $T(n)$ time we can write

$$\begin{aligned} T(n) &= T(n-1) + mn \\ &= T(n-2) + m(n+n+1) \\ &\vdots \\ &= m(1+2+3+\dots+n) = mn^2 \end{aligned}$$

Thus the whole update takes $O(mn^2)$ operations. If the original matrix is banded with a bandwidth b this computation can be further reduced to $O(mnb)$ operations. In our case

$$\begin{aligned} n &= NK \\ m &= 2N \\ b &= 2K \\ N &= 3. \end{aligned}$$

Thus the whole computation can be done in

$$mnb = (6K)(3K)(2K) = 36K^2$$

steps. Thus the update can be done in $O(K^2)$ operations. \square

If we consider that the delays of k user change then this result can be easily extended to show that it requires $O(kK^2)$ steps. However for larger value of k the inversion takes more computational effort. On the other hand the re-computation of the whole Cholesky decomposition from scratch will take $O((3K)^3)$ operations. So there is a threshold for the number of users whose delays may vary below which update is advantageous over the re-computation scheme. In our analysis there will be some lower order terms which will also contribute to the exact computation of this threshold. Also the relative cost of each arithmetic operation will also determine this threshold. As a rule of thumb we have found that for $k \leq K/4$ update scheme gives clear benefit over the re-computation effort.

7. SUMMARY

The exponential complexity of the optimal multiuser detector and the poor performance of the conventional detector has led to the development of the linear detectors. The decorrelating detector which estimates the transmitted bit by inverting the correlation matrix will require computation cubic in the size of the data block length and the number of users. For a potentially infinite data block size this technique fails to provide real time response. In this paper we have explored the linear feedback detection scheme and have shown that the clever windowing and pipelining scheme can help reduce the computational effort. We have proposed a new detector which can reduce the total computational effort without significantly degrading the performance. We have also demonstrated that to adapt to the dynamic behavior of the system updating algorithms can sometimes provide a superior alternative to fresh recomputation. Our future research aim is directed towards the parallelization of the above algorithms.

8. Acknowledgments

This work was supported in part by Nokia Corporation, by the Texas Advanced Technology Program under grant #003604-049, and by NSF under grant NCR 9506681.

REFERENCES

1. S. Verdu, "Minimum Probability of Error for Asynchronous Gaussian Multiple-Access Channels," *IEEE Transactions on Information Theory* **32**(1), pp. 85–96, 1986.
2. J. G. Proakis, *Digital Communications*, McGraw-Hill, 1995.
3. S. Verdu, "Multiuser Detection," *IEEE Transactions on Automatic Control*, 1990.
4. F. C. Zheng and S. K. Barton, "Near-Far Resistant Detection of CDMA Signals via Isolation Bit Insertion," *IEEE Transactions on Communication* **43**, pp. 1313–1317, 1995.
5. M. Juntti and B. Aazhang, "Linear Finite Memory-length Multiuser Detectors," in *Proc. of Communication Theory Miniconference at IEEE Global Telecommunication Conference*, Nov. 1995.
6. A. Duel-Hallen, "Decorrelating Decision Feedback Multiuser Detector for Synchronous Code-Division Multiple-Access Channel," *IEEE Transactions on Communications* **41**, pp. 285–290, Feb. 1993.
7. G. W. Stewart, *Introduction to Matrix Computations*, Academic Press, 1973.
8. J. M. Bennett, "Triangular Factors of Modified Matrices," *Numerical Mathematics* **7**, pp. 217–221, 1965.