Structured Iterative and Circulant MIMO Chip Equalizer Architectures with FFT-acceleration for CDMA Systems

Yuanbin Guo, Dennis McCain
Nokia Research Center
6000 Connections Drive, Irving, TX 75039, USA
Yuanbin.Guo, Dennis.McCain@nokia.com,
Joseph R. Cavallaro
Rice University
6100 Main Street, Houston, TX 77005, USA
cavallar@rice.edu

Abstract

In this paper, we propose a class of novel structured linear MMSE chip equalizer architectures for the MIMO CDMA systems using FFT-accelerations. First, a Conjugate Gradient (CG) algorithm is applied to avoid the Direct Matrix Inverse (DMI), which has $\mathcal{O}((NF)^3)$ complexity. By utilizing a revered form block-Toeplitz structure, the matrix multiplication in the CG iteration is accelerated by an equivalent frequency-domain FFT-based "overlap-save" architecture. The iteration rapidly refines a crude initial approximation to the actual final equalizer taps and significantly reduces complexity from $\mathcal{O}((NF)^2)$ to $\mathcal{O}(NF\log_2(F))$. Secondly, we propose a circulant architecture which also utilizes FFT-based acceleration by approximating the DMI with a block-circulant structure. An extensive comparative analysis in performance, numerical stability and complexity demonstrates promising performance/complexity tradeoff, especially for very long channels. Both algorithms not only reduce the complexity dramatically, but also provide unified parallel and pipelined structures, which is essential for practical real-time VLSI implementation of MIMO systems.

Key words: CDMA, MIMO, LMMSE equalizer, real-time implementation, iterative, Conjugate Gradient, circulant.

Part of the paper was presented in IEEE ICASSP'05 in Philadelphia in March 2005.

1 Introduction

Multiple-Input-Multiple-Output (MIMO) technology [1] using multiple antennas at both the transmitter and receiver sides has recently emerged as a significant breakthrough to dramatically increase the spectral efficiency. While the original invention is known as D-BLAST [2],

a more realistic V-BLAST [3] algorithm was proposed by nulling and canceling with better tradeoff between performance and complexity for real-time implementation. On the other hand, UMTS [4] and CDMA2000 extensions optimized for data services lead to the standardization of multi-code CDMA systems such as the High-Speed-Downlink-Packet-Access (HSDPA) and its equivalent 1X Evolution Data and Voice/Data Optimized (EV-DV/DO) standards [5]. This leads to an asymmetric capacity requirement where the downlink even plays a more essential role than the uplink because of the downloading features. Recently, MIMO extensions for the $3G$ wireless systems have received more and more attentions from the research community [5]. However, one of the major challenges in applying the MIMO CDMA systems in a realistic multipath-fading channel is that, the Multiple-Access-Interference (MAI) is introduced along with the Inter-Symbol-Interference (ISI) because the orthogonality of the spreading codes is destroyed.

The conventional Rake receiver could not provide satisfactory performance with a very short spreading gain to support the high speed data services. Linear-Minimum-Mean-Square-Error (LMMSE)-based chip equalizer is promising to restore the orthogonality of the spreading codes, so as to suppress both the ISI and MAI [6]. However, the LMMSE equalizers need to inverse a large covariance matrix with the $\mathcal{O}((NF)^3)$ complexity, where N is the number of Rx antenna and F is the channel length. This is prohibitively complex for real-time hardware implementation [14]. The fact that the receiver must be embedded into a portable device makes the design of low complexity and low cost mobile receivers very critical for widespread commercial deployment[7].

To avoid the Direct-Matrix-Inverse (DMI), adaptive stochastic gradient algorithms such as LMS could be applied. However, they suffer from stability problems because the convergence depends on the choice of a good step size. On the other hand, non-adaptive block-based algorithms such as the Levinson and Schur [10] [11] algorithms reduce the complexity to the order of $\mathcal{O}((NF)^2)$. Moreover, the standard iterative Conjugate Gradient (CG) algorithm has been applied to compute the linear system equations, e.g., in [12] and [13] for the SIMO and MIMO systems, respectively, also at the order of $\mathcal{O}((NF)^2)$. However, the required MIMO equalizer filter length is high and the covariance matrix has a very large eigenvalue spread when the signal dimension increases significantly in a MIMO-CDMA receiver. This makes the real-time hardware implementation still challenging as shown in [15].

In [17] [18], the authors presented superfast algorithms for many structured matrices. For an L_F-dimension matrix, an algorithm with $\mathcal{O}(L_F \log_2(L_F))$ is considered a superfast algorithm in [17]. However, the standard CG procedure do not apply any matrix structure in the system equation. In this paper, we show that the CG's iteration is essentially reduced to

performing matrix multiplication for the correlation matrix once per iterative step. We also observe that because the covariance matrix assumes a block-Toeplitz structure, we apply a divide-and-conquer methodology to the CG iteration to accelerate the computation. We first transform the direct form block-Toeplitz structure of the correlation matrix in the CG iteration to a reversed form block-Toeplitz structure. The new time-domain matrix multiplication is accelerated by equivalent FFT-based "overlap-save" computing architectures in the frequency domain. Superfast acceleration to the order of $\mathcal{O}(NF\log_2(F))$ is achieved.

Secondly, we show that the system equation can be approximated by a block-circulant structure by adding two corners. Thus, another FFT-based acceleration is proposed based on the features of block-circulant structure and Kronecker product [14] [15]. We then carry out an extensive comparative study in performance, numerical stability and complexity. It is shown that although the circulant approximation may increase the condition number and reduce the system stability, it has lower complexity and can provide reasonable performance for mild channel conditions. On the other hand, because no approximation of the matrix structure is necessary in the iterative CG procedure, the FFT-acceleration is at no cost of the performance and the strong numerical stability of the algorithm is retained for fixed-point implementation.

The comparison with other conventional algorithms demonstrates promising performance/complexity tradeoff for both proposed equalizers in different conditions. In summary, while the FFT-based circulant approximation has the lowest complexity and works fairly well in well-conditioned systems, the FFT acceleration for the iterative algorithm also reduces the complexity of the traditional Conjugate Gradient iteration to approach the $\mathcal{O}(NF\log_2(F))$ complexity and maintain the stability in badly conditioned environments. Both have significantly reduced complexity and structured computing architectures, which is of significant importance for the real-time VLSI implementation of the MIMO CDMA systems.

The rest of the paper is organized as follows. Section II gives the MIMO-CDMA downlink system model. Section III first applies the Conjugate Gradient algorithm for iterative tap solving. We then present the FFT-acceleration schemes for the iterative algorithm both for expanded form and reversed form structures. Section IV presents the FFT-acceleration using block circulant approximation. Section V presents the comparative study for alternative LMMSE-based chip equalizers in terms of performance, complexity and numerical stability. Section VI provides the conclusion.

2 System Model

The system model of the MIMO Multi-Code CDMA downlink using M Tx antennas and N Rx antennas is described as follows. First, the high data rate symbols are demultiplexed into KM lower rate substreams, where K is the number of spreading codes used for data transmission. The substreams are broken into M groups, where each substream in the group is spread with a spreading code of spreading gain G. Each group of substreams are then combined and scrambled with long scrambling codes and transmitted through the m^{th} Tx antenna. The chip level signal at the m^{th} transmit antenna is given by $d_m(i) = \sum_{k=1}^{K} s_m^k(g)c_m^k(i) + s_m^P(g)c_m^P(i)$ where g, i and k are the symbol, chip and code indices respectively. $s_m^k(g)$ is the g^{th} symbol of the k^{th} code at the m^{th} substream. In the following, we focus on the g^{th} symbol and omit the symbol index for simplicity. $c_m^k(i) = c_k(i)c_m^{(m)}(i)$ is the composite spreading code sequence for the k^{th} code at the m^{th} substream where $c_k(i)$ is the user specific spreading code and $c_m^{(m)}(i)$ is the antenna specific scrambling long code. $s_m^P(g)$ denotes the pilot symbols at the m^{th} antenna. $c_m^P(i) = c^P(i)c_m^{(m)}(i)$ is the composite spreading code for pilot symbols at the m^{th} antenna. The received chip level signal at the n^{th} Rx antenna is given by

$$r_n(i) = \sum_{m=1}^{M} \sum_{l=0}^{L} h_{m,n}(l) d_m(i - \tau_l) + z(i) \tag{1}$$

where $h_{m,n}(l)$ constructs the channel matrix between the m^{th} Tx and the n^{th} Rx antennas and L is the channel length.

By collecting the $L_F = 2F + 1$ consecutive chips with center at the i^{th} chip from each Rx antenna in $\mathbf{r}_n(i) = [r_n(i + F), \cdots, r_n(i), \cdots, r_n(i - F)]^T$ and packing the signal vectors from all antennas, we form a vector as $\mathbf{r}_A(i) = [\mathbf{r}_1(i)^T, \cdots, \mathbf{r}_n(i)^T, \cdots, \mathbf{r}_N(i)^T]^T$. Here F is the observation window length at the receiver to support the different channel models. The received signal is then given by

$$\mathbf{r}_A(i) = \sum_{m=1}^{M} \mathbf{H}_m \mathbf{d}_m(i) + \mathbf{v}(i), \tag{2}$$

where $\mathbf{v}(i)$ is the additive Gaussian noise and $\mathbf{d}_m(i) = [d_m(i+F), \cdots, d_m(i), \cdots, d_m(i-F-L)]^T$ is the transmitted chip vector for the m^{th} Tx antenna. $\mathbf{H}_m = [\mathbf{H}_{m,1}^T, \mathbf{H}_{m,2}^T, \cdots, \mathbf{H}_{m,N}^T]^T$ is the channel matrix with block Toeplitz structure and the subblock matrices are defined as

$$\mathbf{H}_{m,n} = \begin{bmatrix} h_{m,n}(0) & \cdots & h_{m,n}(L) & & 0 \\ 0 & h_{m,n}(0) & \cdots & h_{m,n}(L) & \\ & \ddots & & \ddots & \\ 0 & & h_{m,n}(0) & \cdots & h_{m,n}(L) \end{bmatrix}.$$

3 Frequency Domain Iterative Chip Equalizer

Linear MMSE based chip level equalization is one of the most promising receivers for the single-user CDMA downlink. The chip equalizer estimates the transmitted chip samples by a set of linear FIR filter coefficients as

$$\hat{\mathbf{d}}_m(i) = \hat{\mathbf{w}}_m^H(i)\mathbf{r}_A(i). \tag{3}$$

It is well known that the LMMSE chip equalizer is given by

$$\begin{aligned} \hat{\mathbf{w}}_m^{opt}(i) &= \arg\min_{\mathbf{w}_m} \mathrm{E}[||\mathbf{d}_m(i) - \hat{\mathbf{w}}_m^H(i)\mathbf{r}_A(i)||^2] \\ &= \sigma_d^2(i)\hat{\mathbf{R}}_{rr}(i)^{-1}\hat{\mathbf{h}}_m, \end{aligned} \tag{4}$$

where $\sigma_d^2(i)$ is the transmitted chip power. $\hat{\mathbf{R}}_{rr}(i)$ and $\hat{\mathbf{h}}_m(i)$ are the covariance estimation and channel estimation respectively. Here the covariance matrix is estimated by the time-average with ergodicity assumption as $\hat{\mathbf{R}}_{rr}(i) = \mathrm{E}[\mathbf{r}_A(i)\mathbf{r}_A^H(i)] = \frac{1}{N_B}\sum_{i=0}^{N_B-1}\mathbf{r}_A(i)\ \mathbf{r}_A^H(i)$, where N_B is the length for the time average. The channel coefficients are estimated as $\hat{\mathbf{h}}_m(i) = \mathrm{E}[\mathbf{r}_A(i)\mathbf{d}_m^H(i)]$ using the pilot symbols. In the HSDPA standard, about 10 % of the total transmit power is dedicated to the Common Pilot Channel (CPICH). This will provide acceptable channel estimation. By assuming that the channel is stationary over the observation window length, we can have a block-based operation by omitting the chip index in $\hat{\mathbf{R}}_{rr}(i)$, $\hat{\mathbf{h}}_m(i)$ and $\hat{\mathbf{w}}_m(i)$.

3.1 Conjugate Gradient Tap Solver

Fig. 1 gives the simple model of the LMMSE chip equalizer in the MIMO-CDMA downlink. In the figure, each dotted line block denotes one processing unit for one single antenna. The chip equalizer could be divided into two major steps: the tap solver to compute the equalizer filter taps; the FIR filters to equalize the multipath channel, which is marked the actual equalizer in the figure. A direct matrix inverse solution using Cholesky decomposition or Gaussian elimination involves complex multiplications and division operations at the order of $\mathcal{O}((NF)^3)$ for a NF-dimension matrix. This is very difficult for hardware real-time implementation. For this reason, adaptive solutions such as Least-Mean-Square (LMS) algorithms have been proposed for more practical implementations. However, when multiple receive antennas are used, the convergence speed of the adaptive chip-level algorithms becomes a limiting factor. In unfavorable channel conditions, the signal covariance matrix has a large spectral condition number. This slows the convergence rate dramatically and leads to unsatisfactory performance in a fading environment.

To deal with the large eigenvalue spread in the multi-antenna receiver, we apply an iterative Conjugate-Gradient algorithm to solve the equalizer taps. In the MIMO system, there are two steps involved with a total of J iterations for the m^{th} transmit antenna. The complete CG procedure is given as following:

1. *Initialization*

$$\mathbf{w}_{m,0} = \mathbf{0} \tag{5}$$

$$\boldsymbol{\gamma}_0 = \mathbf{h}_m; \quad \boldsymbol{\Delta}_0 = \mathbf{h}_m \tag{6}$$

$$\delta_0 = \boldsymbol{\gamma}_0^H \boldsymbol{\gamma}_0; \quad \delta_1 = \delta_0 \tag{7}$$

2. *Iteration: for* $j = 1 : J$

$$\boldsymbol{\Gamma}_j = \mathbf{R}_{rr}\boldsymbol{\Delta}_{j-1}; \quad \alpha = \delta_j/(\boldsymbol{\Delta}_{j-1}^H \boldsymbol{\Gamma}_j); \tag{8}$$

$$\mathbf{w}_{m,j} = \mathbf{w}_{m,j-1} + \alpha\boldsymbol{\Delta}_{j-1}; \tag{9}$$

$$\boldsymbol{\gamma}_j = \boldsymbol{\gamma}_{j-1} - \alpha\boldsymbol{\Gamma}_j; \tag{10}$$

$$\delta_{j+1} = \boldsymbol{\gamma}_j^H \boldsymbol{\gamma}_j; \quad \beta = \delta_{j+1}/\delta_j \tag{11}$$

$$\boldsymbol{\Delta}_j = \boldsymbol{\gamma}_j + \beta\boldsymbol{\Delta}_{j-1} \tag{12}$$

where $\boldsymbol{\gamma}_j, \boldsymbol{\Delta}_j$ are the feed-forward and feedback vector, respectively. $\boldsymbol{\Gamma}_j$ is the output vector of the matrix-vector product $\mathbf{R}_{rr}\boldsymbol{\Delta}_{j-1}$. δ_j, α and β are scalars for fine adjustment at the j^{th} iteration.

3.2 Time-domain Matrix-Vector Multiplication

The conventional CG algorithm involves the following significant computations in terms of complex multiplications: the dot product of $\delta_0 = \boldsymbol{\gamma}_0^H \boldsymbol{\gamma}_0; \delta_{j+1} = \boldsymbol{\Gamma}_j^H \boldsymbol{\Gamma}_j$; $\boldsymbol{\Delta}_{j-1}^H \boldsymbol{\Gamma}_j$; the matrix multiplication in $\boldsymbol{\Gamma}_j = \mathbf{R}_{rr}\boldsymbol{\Delta}_{j-1}$, and many vector scaling computations. Among these, the matrix-vector multiplication $\boldsymbol{\Gamma}_j = \mathbf{R}_{rr}\boldsymbol{\Delta}_{j-1}$ in the iteration has the dominant complexity. It could be shown that both the covariance matrix and the channel estimation vector could be partitioned to sub matrix and vectors according to the number of receive antennas. The covariance matrix has the size of $N(L+1) * N(L+1)$ and can be partitioned as

$$\mathbf{R}_{rr} = \begin{pmatrix} \mathbf{R}_{11} & \cdots & \mathbf{R}_{1N} \\ \vdots & \ddots & \vdots \\ \mathbf{R}_{N1} & \cdots & \mathbf{R}_{NN} \end{pmatrix}, \tag{13}$$

where each submatrix $\mathbf{R}_{n_1,n_2}$ is the cross-covariance matrix of two antennas. The $\boldsymbol{\Delta}_{j-1}$ vector is a $N(L+1)$ vector and can also be partitioned as $\boldsymbol{\Delta}_{j-1} = [\boldsymbol{\Delta}_{j-1,1}^T, \cdots, \boldsymbol{\Delta}_{j-1,N}^T]^T$, where the

second index in the element vectors is the receive antenna index. Thus, the matrix-vector multiplication $\mathbf{\Gamma}_j = \mathbf{R}_{rr}\mathbf{\Delta}_{j-1}$ is partitioned into sub-blocks as

$$\mathbf{\Gamma}_j = \begin{pmatrix} \sum_{n=1}^{N} \mathbf{R}_{1,n}\mathbf{\Delta}_{j-1,n} \\ \vdots \\ \sum_{n=1}^{N} \mathbf{R}_{N,n}\mathbf{\Delta}_{j-1,n} \end{pmatrix}. \tag{14}$$

The conventional VLSI architecture in hardware of the Direct-form sub Matrix-vector Multiplication (DMM) $\mathbf{\Gamma}_{j,n_1,n_2} = \mathbf{R}_{n_1,n_2}\mathbf{\Delta}_{j-1,n_2}$ is shown in Fig. 2, where $m(l)$ means the l^{th} element of the matrix/vector $\mathbf{M}$. The complexity for a single sub-block computation is shown in Table 6 if we assume the size of the correlation matrix is $L+1$. The memory access includes $(L+1)\times(L+1)$ to store the $\mathbf{R}_{n_1,n_2}$ matrix and $(L+1)$ for the $\mathbf{\Delta}$ and $\mathbf{\Gamma}$ vectors respectively.

Because the convergence of the CG algorithm depends on the spectral condition number of the signal covariance matrix, the MIMO covariance matrix with large eigenvalue spread will slow down the convergence rate. On the other hand, at least once or twice updates of the equalizer filters in one WCDMA slot (0.67 μs) are needed to capture the channel fading speed. Although the CG algorithm reduces the order of complexity to $\mathcal{O}(L^2)$, the hidden constant in the CG is still high when many iterations are required to reach convergence. This limits the real-time implementation in compact hardware. In the following, we propose equivalent FFT-based computing architectures to reduce the numerical complexity, which also provide some parallel and pipelined structures for efficient VLSI implementation.

3.3 Time-domain FIR Filtering-based Architecture

Using the stationarity of the channel and the convolution property, it is easy to show that the covariance matrix is a banded block Toeplitz matrix with only $N(L+1)$ independent elements. Each sub matrix has the form

$$\mathbf{R}_{n_1,n_2} = \begin{pmatrix} E_{n_1,n_2}(0) & \cdots & & E^*_{n_1,n_2}(L) \\ E_{n_1,n_2}(1) & \ddots & \vdots & E^*_{n_1,n_2}(L-1) \\ \vdots & \ddots & & \vdots \\ E_{n_1,n_2}(L) & \cdots & \vdots & E^*_{n_1,n_2}(0) \end{pmatrix}. \tag{15}$$

By defining the upper triangular corner matrix as

$$\mathbf{C}^L_{n_1,n_2} = \begin{pmatrix} E_{n_1,n_2}(L) & \cdots & E_{n_1,n_2}(1) \\ & \ddots & \vdots \\ 0 & & E_{n_1,n_2}(L) \end{pmatrix}, \tag{16}$$

we can expand the $\mathbf{R}_{n_1,n_2}$ matrix to a bigger Expanded-Form (EF) matrix by concatenation in the following way as

$$\mathbf{Z}^L_{n_1,n_2} = \left(\begin{pmatrix} \mathbf{C}^L_{n_1,n_2} \\ \mathbf{0}_{1\times L} \end{pmatrix} \quad \mathbf{R}_{n_1,n_2} \quad \begin{pmatrix} \mathbf{0}_{1\times L} \\ (\mathbf{C}^L_{n_1,n_2})^H \end{pmatrix} \right). \tag{17}$$

It can be shown that this is a Toeplitz matrix with size of $(L+1)\times(3L+1)$. Thus the matrix computation by the correlation matrix is equal to

$$\begin{aligned} \mathbf{\Gamma}_{j,n_1,n_2} &= \mathbf{R}_{n_1,n_2}\mathbf{\Delta}_{j-1,n_2} && (18) \\ &= \mathbf{Z}_{n_1,n_2} \cdot \begin{pmatrix} \mathbf{0}_{L\times 1} \\ \mathbf{\Delta}_{j-1,n_2} \\ \mathbf{0}_{L\times 1} \end{pmatrix}. && (19) \end{aligned}$$

From the sliding window feature of the $\mathbf{Z}_{n_1,n_2}$ matrix, the matrix-vector multiplication is actually a linear convolution, where the filter taps vector is $\mathbf{Y}_{n_1,n_2} = [E^*_{n_1,n_2}(L), \cdots, E^*_{n_1,n_2}(1), E(0), \cdots, E_{n_1,n_2}(L)]$ and the input data vector is $\mathbf{X}_{n_2} = [\mathbf{0}_{1\times L},\ \mathbf{\Delta}^T_{j-1,n_2},\ \mathbf{0}_{1\times L}]$. This linear convolution can be implemented using the time-domain linear FIR filtering either in the direct-form or transposed computation architecture. The general equation is given by

$$\Gamma_{j,n_1,n_2}(l) = \sum_{k=0}^{2L} y_{n_1,n_2}(k)\Delta_{j-1,n_2}(L-l+k) \tag{20}$$

for $\{\Delta_{j-1,n_2}(l) = 0, l<0 \text{ and } l>L\}$. A VLSI architecture example for the direct-form FIR filtering is given in Fig. 3.

3.4 Expanded-Form (EF) Frequency Domain FFT Acceleration

Despite of its more regular VLSI architecture, the FIR filtering has the computation complexity of $(2L+1)*(L+1)$ multiplications where only $(L+1)*(L+1)$ multiplications are non-trivial for non-zero values. So the overall complexity of CG is still at the order of $\mathcal{O}(L^2)$. However, from the FIR filtering interpretation and the features of FFT algorithm, we can implement the linear convolution by circular convolution using FFT in the frequency-domain. To convert the circular convolution to the linear convolution, we need to take care of the edges to avoid frequency aliasing. It is known that there are two equivalent architectures for the FFT-based filtering of a long sequence, i.e., the "overlap-add" and "overlap-save" architectures [19]. Because of the simplicity in the control logic design, we derive the "overlap-save" based computing architecture. This basic procedure is shown in Fig. 4 and explained below.

First, zeros are padded to the filter coefficients and then an FFT is carried out for the zero-padded filter vector. A record is taken out from the long sequence and concatenated

with the $2L$ values from the previous record. The FFT result of this record is multiplied with the FFT result of the filter. An IFFT transform is then carried out to get the time-domain samples. Some samples need to be discarded from the result to avoid the alias. Because the length of the filter is $(2L+1)$, if we pad L zeros to the filter coefficients and take a $(3L+1)$ point FFT, the size of the FFT for the data vector will also be $(3L+1)$. We can consider it as a long sequence consisting of the $(L+1)$ non-zeros values of $\mathbf{\Delta}_{j-1,n_2}$. The overlap-save values from the first record become the first $2L$ samples of the FFT input as $[X_{n_2}(0), \cdots, X_{n_2}(2L-1)] = [\mathbf{0}_{1\times L}, \mathbf{\Delta}_{j-1,n_2}(0), \cdots, \mathbf{\Delta}_{j-1,n_2}(L-1)]$ and the $L+1$ new values in the overlap-save structure are $[\mathbf{\Delta}_{j-1,n_2}(L), \mathbf{0}_{1\times L}]$. The two FFT results are dot-multiplied in the frequency domain. Then after the IFFT, the final result needs to drop the $2L$ points to avoid the frequency alias. The final valid result is given by

$$\Delta_{j,n_1,n_2}(l) = \lambda_{n_1,n_2}(l+2L) \quad l \in [0, L]. \tag{21}$$

This architecture involves two FFTs and one IFFT of the length $(3L+1)$. Plus, there is one dot product of size $(3L+1)$. The complexity is given by $3(3L+1)\log_2(3L+1)/2 + (3L+1)$ complex multiplications. In this way, we have reduced the order of complexity to $\mathcal{O}\big(CL * \log_2(3L+1)\big)$ from the squared order $\mathcal{O}\big((L+1)^2\big)$, where C is a constant to indicate the complexity.

3.5 Reversed-Form FFT Acceleration Architecture

Although the order of complexity has been reduced, the constant C in front of the order number can be further reduced. Because the expanded matrix $\mathbf{Z}_{n_1,n_2}$ has many zero values, the inputs to the FFT contain many zeros. By pruning some branches in the FFT butterfly tree, we can reduce the state of the FFT computation. However, this involves many customizations in the FFT module, which is not desirable for VLSI implementation. Here we show a complexity reduction scheme to reduce the size of FFT by manipulating the matrix multiplication format. It can be shown that the matrix multiplication in $\mathbf{\Gamma}_{j,n_1,n_2} = \mathbf{R}_{n_1,n_2}\mathbf{\Delta}_{j-1,n_2}$ is equivalent to the following reversed form matrix-vector multiplication, where the elements of $\mathbf{\Delta}_{j-1,n_2}$ form the Topelitz matrix and the independent elements of the correlation matrix form the multiplying vector $\mathbf{Y}_{n_1,n_2} = \big(E^*_{n_1,n_2}(L), \cdots, E_{n_1,n_2}(0), \cdots, E_{n_1,n_2}(L)\big)^T$. The matrix-vector multiplication is then reformed as

$$\mathbf{\Gamma}_{j,n_1,n_2} = \begin{pmatrix} \Delta_{j-1,n_2}(L) & \cdots & \Delta_{j-1,n_2}(0) & 0 \\ 0 & & & \\ \vdots & \ddots & & \ddots \\ 0 & \Delta_{j-1,n_2}(L) & \cdots & \Delta_{j-1,n_2}(0) \end{pmatrix} * \mathbf{Y}_{n_1,n_2}. \tag{22}$$

This can be viewed as an operation where the data sequence $\mathbf{Y}_{n_1,n_2}$= $[E^*_{n_1,n_2}(L), \cdots, E_{n_1,n_2}(0), \cdots, E_{n_1,n_2}(L)]$ is filtered by the tap coefficients $\mathbf{\Delta}_{j-1,n_2}$. With the "overlap-save" FFT-based FIR filtering architecture, the filter $\mathbf{\Delta}_{j-1,n_2}$ is first padded with zeros to make a vector with length of $(2L+1)$. The vector $[E^*_{n_1,n_2}(L), \cdots, E^*_{n_1,n_2}(1)]$ is considered as the L "overlap-save" values from the first block. They are concatenated with the $(L+1)$ new values $[E_{n_1,n_2}[0], E_{n_1,n_2}[1], \cdots, E_{n_1,n_2}[L]]$ to make a length of $(2L+1)$ vector. This vector is input to the FFT module of length $(2L+1)$. The FFT results are multiplied with the FFT result of the filter taps. An IFFT module computes the interim results and the first L samples are dropped to generate the final result as

$$\Gamma_{j,n_1,n_2}(l) = \lambda_{n_1,n_2}(l+L) \quad l \in [0, L]. \tag{23}$$

This procedure is shown in Fig. 5. The number of complex multiplications is given by $(6L+3)\log_2(2L+1)/2 + (3L+1)$. After extracting the commonality, the optimized iterative algorithm with the reversed-form FFT acceleration is summarized here. For the m^{th} transmit antenna, the iteration is accelerated with FFT in the frequency domain with the initialization and iteration as:

1. *Initialization:* The initialization for $\mathbf{w}_{m,0}$, $\boldsymbol{\gamma}_0$, $\mathbf{\Delta}_0$ and δ_0, δ_1 is the same as the conventional CG algorithm. Furthermore, we initialize $\mathbf{\Psi}_{n_1,n_2} = FFT(\mathbf{Y}_{n_1,n_2})$ for $n_1 \in [1, N]$ and $n_2 \in [1, N]$. Note that for the MIMO partitioning, the initial vectors are actually given by

$$\mathbf{\Delta}_0 = \begin{pmatrix} \mathbf{\Delta}_{0,1} \\ \vdots \\ \mathbf{\Delta}_{0,N} \end{pmatrix} = \hat{\mathbf{h}}_m = \begin{pmatrix} \hat{\mathbf{h}}_{m,1} \\ \vdots \\ \hat{\mathbf{h}}_{m,N} \end{pmatrix}; \tag{24}$$

$$\mathbf{w}_{m,0} = \mathbf{0} \quad \boldsymbol{\gamma}_0 = \hat{\mathbf{h}}_m; \tag{25}$$

   and the initial scalars are given by $\delta_0 = \gamma_0^H \gamma_0; \quad \delta_1 = \delta_0$.

2. *Iteration (for $j = 1 : J$)*: To compute the MIMO matrix-vector multiplication $\mathbf{\Gamma}_j = \mathbf{R}_{rr}\mathbf{\Delta}_{j-1}$in frequency domain, we first initialize the partitioned vector $\mathbf{\Gamma}_j = [\mathbf{\Gamma}^T_{j,1}, \cdots, \mathbf{\Gamma}^T_{j,N}]^T$ $= \mathbf{0}$. Then, for $n_2 = 1 : N$, we compute the element-wise FFT $\mathbf{F}_{j,n_2} = FFT([\mathbf{\Delta}^T_{j-1,n_2}\ \mathbf{0}_{1\times L}])$. In the inner loop for $n_1 = 1 : N$, the frequency domain dot-product and the element-wise IFFT are computed as following:

$$\begin{aligned}
\mathbf{\Phi}_{n_1,n_2} &= \mathbf{F}_{j,n_2} \circ \mathbf{\Psi}_{n_1,n_2}; && (26)\\
\mathbf{\Theta}_{n_1,n_2} &= IFFT(\mathbf{\Phi}_{n_1,n_2}); && (27)\\
\Gamma_{j,n_1,n_2}(l) &= \theta_{n_1,n_2}(l+L) \quad l \in [0, L]; && (28)\\
\mathbf{\Gamma}_{j,n_1} &= \mathbf{\Gamma}_{j,n_1} + \mathbf{\Gamma}_{j,n_1,n_2}; && (29)
\end{aligned}$$

where "$\circ$" denotes the "dot-product". With the partitioned $\mathbf{\Gamma}_j$ vector, the scalars are computed as

$$\omega_j = \sum_{n=1}^{N} \mathbf{\Delta}_{j-1,n}^H \mathbf{\Gamma}_{j,n}; \quad (30)$$

$$\alpha = \delta_j / \omega_j \quad (31)$$

Finally, the equalizer taps are adjusted in the same manner as $\mathbf{w}_{m,j} = \mathbf{w}_{m,j-1} + \alpha \mathbf{\Delta}_{j-1}$, $\boldsymbol{\gamma}_j = \boldsymbol{\gamma}_{j-1} - \alpha \mathbf{\Gamma}_j$, then $\delta_{j+1} = \sum_{n=1}^{N} \boldsymbol{\gamma}_{j,n}^H \boldsymbol{\gamma}_{j,n}; \beta = \delta_{j+1}/\delta_j$ and $\mathbf{\Delta}_j = \boldsymbol{\gamma}_j + \beta \mathbf{\Delta}_{j-1}$ in the conventional CG iteration.

We can apply the commonality between different transmit antennas and within the iterations to further reduce the redundant computations. Because the FFT of the covariance vectors is not dependent on the iteration and the transmit antenna, we can first compute the FFT results of the covariance vectors in the initialization part common to all M transmit antennas and J iterations. The result is saved in distributed $\mathbf{\Psi}_{N \times N}$ sub-vectors for the N receive antennas. The channel vector is also partitioned into N sub-vectors. This saves the number of FFTs for this part from $(N^2 * J * M)$ with straightforward implementation to only N^2. In the iteration part, the $\mathbf{\Gamma}_j$ vector is first partitioned and initialized to zero vectors. By re-arranging the loop structure of n_2 and n_1, the FFT of the zero-padded $\mathbf{\Delta}_{j-1,n_2}$ vector is only necessary for the outer loop. Thus, we only need to compute the frequency domain dot-product and the IFFT for individual $\mathbf{\Gamma}_{j,n_1,n_2}$ in the inner loop. An accumulator will generate the $\mathbf{\Gamma}_{j,n_1}$ sub-vectors. After the n_1, n_2 loop, two scalars δ_{j+1} and ω_j are computed from the inner product. Because the δ_{j+1} is actually a computation of the norm of $\mathbf{\Gamma}_j$, there is no need to compute full complex multiplication here.

4 FFT-acceleration with Circulant Approximation

In this section, we present another FFT-based acceleration by using the circulant approximation to the block-Toeplitz structure of the covariance matrix. To facilitate the structure approximation, we define the signal vectors in a slightly different way. By packing the received chips from all the receive antennas in a vector $\mathbf{r}(i) = [r_1(i), \cdots, r_n(i), \cdots, r_N(i)]^T$ and collecting the $L_F = 2F+1$ consecutive chips with center at the i^{th} chip from all the N Rx antennas, we form the signal vector as $\mathbf{r}_A = [\mathbf{r}(i+F)^T, \cdots, \mathbf{r}(i)^T, \cdots, \mathbf{r}(i-F)^T]^T$. In the vector form, the received signal can also be given by $\mathbf{r}_A(i) = \sum_{m=1}^{M_t} \mathbf{H}_m \mathbf{d}_m(i) + \mathbf{z}(i)$, where $\mathbf{H}_m$ is a block Toeplitz matrix constructed from the channel coefficients as shown in [15]. The multiple receive antennas' channel vector is defined as $\mathbf{h}_m(l) = [h_{m,1}(l), \cdots, h_{m,n}(l), \cdots, h_{m,N}(l)]^T$.

The transmitted chip vector for the m^{th} transmit antenna is given by $\mathbf{d}_m(i) = [d_m(i+F), \cdots, d_m(i), \cdots, d_m(i-F-L)]^T$. Using the stationarity of the channel and the convolution property, it is easy to show that the correlation matrix under the new received signal vector is a banded block Toeplitz matrix as

$$\mathbf{R}_{rr} = \begin{pmatrix} \mathbf{E}[0] & \cdots & \mathbf{E}^H[L] & \cdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \cdots & \vdots \\ \mathbf{E}[L] & \ddots & \ddots & \cdots & \mathbf{E}^H[L] \\ \vdots & \ddots & \cdots & \vdots & \ddots \\ \mathbf{0} & \cdots & \mathbf{E}[L] & \cdots & \mathbf{E}[0] \end{pmatrix}$$

where $\mathbf{E}[l]$ is an $N \times N$ block matrix with the correlation coefficients. It is known that a circulant matrix $\mathbf{S}$ can be diagonalized [23] by the FFT operation as $\mathbf{S} = \mathbf{D}^H \mathbf{\Lambda} \mathbf{D}$ where $\mathbf{D}$ is the FFT phase coefficient matrix and $\mathbf{\Lambda}$ is a diagonal matrix whose diagonal elements are the FFT result of the first column of the circulant matrix $\mathbf{S}$. This known lemma can be applied to simplify the MIMO equalizer computation dramatically. It is shown that the correlation matrix $\mathbf{R}_{rr}$ can be approximated by a block-circulant matrix after we add two corner matrices as

$$\mathbf{C}_{rr} = \mathbf{R}_{rr} + \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{C}_L^H \\ \vdots & \ddots & \mathbf{0} \\ \mathbf{C}_L & \mathbf{0} & \mathbf{0} \end{pmatrix}$$

where

$$\mathbf{C}_L = \begin{pmatrix} \mathbf{E}^H[L] & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \mathbf{0} \\ \mathbf{E}^H[1] & \cdots & \mathbf{E}^H[L]. \end{pmatrix}.$$

Using the extension of the diagonalization theorem, the block-circulant matrix can be decomposed as

$$\mathbf{C}_{rr} = \Big(\mathbf{D}^H \otimes \mathbf{I}\Big)\Big(\sum_{i=0}^{L_F-1} \mathbf{W}^i \otimes \mathbf{E}[i]\Big)\Big(\mathbf{D} \otimes \mathbf{I}\Big) \tag{32}$$

where $\mathbf{W} = \text{diag}(1, W_{L_F}^{-1} \ \cdots \ W_{L_F}^{-(L_F-1)})$ and $W_{L_F} = e^{j(2\pi/L_F)}$is the phase factor coefficient for the DFT computation. $\otimes$ denotes the *Kronecker* product and $\mathbf{D}$ is the DFT matrix. For MIMO system, it can be shown that the MIMO equalizer taps are computed by

$$\hat{\mathbf{w}}_m^{opt} \approx (\mathbf{D}^H \otimes \mathbf{I}) \cdot \mathbf{F}^{-1} \cdot (\mathbf{D} \otimes \mathbf{I})\hat{\mathbf{h}}_m. \tag{33}$$

In (32) the operation to compute $\mathbf{F} = \Big(\sum_{i=0}^{L_F-1} \mathbf{W}^i \otimes \mathbf{E}[i]\Big)$ is implemented by some parallel element-wise FFTs of the covariance vectors. $\mathbf{D} \otimes \mathbf{I}$ is implemented as some parallel dimension-wise FFT while $\mathbf{D}^H \otimes \mathbf{I}$ is implemented as some dimension-wise IFFT operations.

$\mathbf{F} = \text{diag}(\mathbf{F}_0,\ \mathbf{F}_1, \cdots, \mathbf{F}_{L_F})$ is a block-diagonal matrix with elements taken from the element-wise FFT of the first column of a circulant matrix. For an $(M \times N)$ MIMO system, this reduces the inverse of an $(NL_F \times NL_F)$ matrix to the inverse of sub-block matrices with size $(N \times N)$. This is much more tractable for hardware implementation than the original direct inversion of the covariance matrix. Thus, the equalizer computation is accelerated by FFT-based operations. A block diagram of this circulant approximation based FFT-acceleration is shown in Fig. 6.

5 Comparative Performance and Complexity Analysis

5.1 BER Performance under Well/Mild Conditions

We first compare the performance of four different schemes: the LMS adaptive algorithm, the FFT-accelerated CG algorithm, the FFT-based algorithm with circulant approximation and the DMI using Cholesky decomposition in some well and mild conditioned systems. The performance is evaluated in an MIMO-HSDPA simulation chain for different antenna configurations. We simulated the Pedestrian-A and Pedestrian-B channels following the I-METRA channel model [20] [21]. The chip rate for the transmit signal is 3.84 Mcps, which is in compliance with the 3GPP standard. Orthogonal-Variable-Spreading-Factor (OVSF) codes are generated from the Hadamard sequence. Notice that unlike the conventional CDMA downlink, all the K codes are assigned to one single user to achieve high rate downlink service in the multi-code CDMA system. The channel state information is estimated from the Common Pilot Channel (CPICH) at the receiver. 10% of the total transmit power is dedicated to the pilot training symbols.

We provide the simulation results for QPSK modulation with antenna configuration in the form of $(M \times N)$. In the figures, L_h is the channel delay spread. Fig. 7 shows the fully loaded system for Pedestrian-A channel with 2×2 configuration, while Fig. 8 shows a fully loaded system for 2×2 Pedestrian-B channel. Fig. 9 shows the simulation results for Pedestrian-A with 4×4 configuration. It can be seen that for Fig. 7, the FFT-based algorithm with circulant approximation overlaps with both the DMI and the CG at 5 iterations very closely. In a 2×2 case for Pedestrian-B channel, both the CG and FFT-based algorithm shows very small divergence from the DMI at the very high SNR range in Fig. 7. For a fully loaded system, CG with 5 iterations seems to be slightly better than FFT-based algorithm. In the 4×4 case as shown in Fig. 9, the CG algorithm with 5 iterations has close performance to the FFT-based circulant approximation as well as the DMI. In all cases, the DMI, CG and FFT-based algorithms significantly outperform the LMS adaptive algorithm.

5.2 BER Performance for Badly Conditioned Channels

In this section, we provide the simulation results in more difficult channel conditions. The I-METRA modified Pedestrian-B channel models with 13 and 15 paths and mobile speed of 3 km/h [20] [21] are simulated. Fig. 10 and Fig. 11 show the performance for $L_h = 13$ and $L_h = 15$ respectively. To show the potential of the iterative algorithm, 8 iterations are computed. Both the CG and FFT-based algorithm show small divergence from the DMI at the small to medium SNR range. Both the CG and the circulant approximation FFT solution are much better than the LMS solution. For the very high SNR range, we see the performance degradation with the circulant approximation in this very bad channel situation. In the range of SNR $>$ 16 dB, the FFT-based circulant approximation even fails because the high condition number corrupts the stability of the system equation. But the FFT accelerated iterative algorithm follows the DMI solution very closely.

5.3 Numerical Stability

The system stability is determined by the condition number of the system matrix. The condition number of a matrix is defined as the ratio of the maximal eigenvalue over the minimum eigenvalue,

$$\kappa(\mathbf{R}_{rr}) = \frac{\lambda_{max}(\mathbf{R}_{rr})}{\lambda_{min}(\mathbf{R}_{rr})}. \tag{34}$$

If the condition number is large, the matrix tends to be ill conditioned with the convention that $\kappa = \infty$ for a singular matrix. An ill-conditioned matrix has very bad numerical stability in matrix inverse operations. In the following, we analyze the 2-norm condition number of the original covariance matrix and the circulant approximation matrix used in the FFT-based algorithm in [15] for difference SNR range. We also look at the reciprocal of the condition of $\mathbf{R}$ in the 1-norm obtained by the matlab built-in function LAPACK condition estimator. If the matrix is well conditioned, the reciprocal of the condition is near 1.0. Otherwise, if the matrix is badly conditioned, the reciprocal is near zero.

In Table 2, the condition number $\kappa(\mathbf{R}_{rr})$ and the reciprocal of the 1-norm condition number $\kappa_{rc}(\mathbf{R}_{cir})$ are shown for an $L_h = 13$ channel for different SNR range. The $\mathbf{R}_{rr}$ and $\mathbf{R}_{cir}$ are the original covariance matrix and the circulant matrix after adding the corners, respectively. It could be seen that in the range of SNR=0 to 16 dB, the original matrix is reasonably well or mild conditioned. The condition number increases for higher SNR while the κ_{rc} decreases when the SNR increases. After the corner is added to make the $\mathbf{R}_{cir}$, the condition number get worse for each SNR. This is compatible with the analysis and reduces

the numerical stability in the equalizer tap solver. In Table 3, we analyze the condition for a worse channel case when $L_h = 15$. First we notice that the condition number is higher than the $L_h = 13$ case respectively. After adding the corner for $\mathbf{R}_{cir}$, the condition number increases dramatically. Especially when SNR=24 dB, $\kappa(\mathbf{R}_{cir})$ becomes $3.29e+3$. This may lead to the singularity of the circulant matrix for numerical operation with limited word length.

5.4 Complexity

The complexity is one of the most important considerations for real-time implementation. Although the complete equalizer system consists of the correlation/channel estimation, the tap solver and the FIR filtering, we focus on the complexity of the three tap solvers with similar performance, i.e., the DMI, the standard CG, the FFT-accelerated CG and the FFT-based circulant approximation algorithm. The other two parts are common for the algorithms presented here. Cholesky decomposition is assumed for the DMI. The complexity is compared in terms of the number of equivalent complex multiplications and additions.

For the DMI, the complexity is at the order of $\mathcal{O}((N(F+1))^3)$ for the inverse of $\mathbf{R}_{rr}$ and $\mathcal{O}((N(F+1))^2 M)$ for the matrix multiplication in $(\mathbf{R}_{rr})^{-1}\mathbf{h}_m$. For the standard CG algorithm, there are $\mathcal{O}\{MJ[N(F+1)]^2 + M(5J+1)N(F+1)\}$ complex multiplications and $\mathcal{O}\{MJ[N(F+1)]^2 + 8MJN(F+1)\}$ complex additions. Usually, $J = 5$ iterations for the CG algorithm will suffice for converging to the DMI solution. For the FFT-based algorithm with circulant approximation, the overall complexity is $\mathcal{O}\{(N^2 + 2MN)L_F(\log_2 L_F)/2 + (N^3 + MN^2)L_F\}$. For the FFT-based circulant approximation algorithm, we usually require $L_F \geq 2F + 1$.

Now we examine the complexity of the FFT-accelerated iterative equalizer. The complexity of matrix-vector multiplication of size $(L+1)$ in the CG iteration with different computing architectures is shown in Table 4. Note that L is the channel correlation window size which is usually chosen the same as the channel length. DMM and the Reversed-Form FIR architectures have the same multiplication complexity. FIR-EF refers to the FIR-based architecture for the expanded-form matrix. The original CG algorithm using time-domain computing architecture has the complexity at the order of $\mathcal{O}(MJ(N(L+1))^2$. For the frequency-domain architectures, FFT-EF indicates the FFT-based architecture with the Expanded-Form matrix. FFT-RF denotes the FFT-based architecture for the reversed-form matrix. It is seen that the constant of the FFT-RF complexity is reduced.

The trend of the complexity for a single sub-matrix operation is shown in Fig. 12 for increasing channel length. The benefit of the FFT-RF is obvious for very long correlation

length. From the summary of FFT-accelerated iterative tap solver, the number of FFTs in the iterative MIMO chip equalizer is $(N^2+M*N*J)$ and the number of IFFTs is $M*(N^2+J)$. Moreover, there are $M*N^2*J*(2L+1)$ complex multiplications for the frequency-domain dot product and $1.5*M*N*J*(2L+1)$ complex multiplications for the inner product. For an L_F-point FFT/IFFT, the number of complex multiplications is "$L_F * \log_2(L_F)/2$". Altogether, the complexity is $[(M+1)*N^2+MJ*(N+1)]*(2L+1)/2*\log_2(2L+1)+1.5*M*N*J*(2L+1)$ for the reversed form FFT-based iterative tap solver. For simplicity, the dominant complexity is given by the order of $\mathcal{O}\{(M*N^2+MNJ)/2*(2L+1)*\log_2(2L+1)\}$. It could be seen that the FFT-RF has significantly accelerated the algorithm by reducing the number of multiplications from $\mathcal{O}((NL)^2)$ to $O(N^2L*\log_2(L))$ with a small order constant.

The complexity of different tap solvers is summarized in Table 5. For simplicity, we only list the most significant part of equivalent number of complex multiplications. An example is given for the 4×4 case with $F = 10,\ J = 5$. The length of FFT $L_F = 32$ will suffice for both Pedestrian-A and Pedestrian-B channels. In Fig. 13, we show the complexity trend for different J and different L_F versus the channel length for a 4×4 system. Although the standard CG algorithm has reduced complexity compared with the DMI, the FFT acceleration significantly reduces the standard CG algorithm complexity by an order of magnitude. The complexity is slightly higher than the FFT-based solution with the circulant approximation, while both have the same complexity order. On the other hand, the numerical stability of the iterative algorithm in bad conditioned channels is better. In summary, these FFT-accelerated architectures provide very promising performance and complexity tradeoff for hardware implementation.

6 Conclusion

In this paper, we present a class of novel FFT-accelerated iterative and circulant architectures to reduce the complexity of the LMMSE-based MIMO chip equalizer to $\mathcal{O}\big(NL*\log_2(L)\big)$. We first apply an iterative CG algorithm to a reversed form block-Toeplitz structure to avoid direct matrix inverse. The time-domain matrix-vector multiplication is accelerated by an equivalent frequency-domain circular convolution with FFT-based "overlap-save" architecture. The iteration rapidly refines a crude initial approximation to the actual final equalizer taps. We also propose an FFT-acceleration using circulant approximation. Both the FFT-based circulant approximation and the FFT iterative algorithm proposed in this paper significantly reduce the computation complexity and provide parallel and pipelined architecture for efficient VLSI implementation. Although the circulant-approximation in-

creases the condition number by adding corners, the FFT-accelerated iterative algorithm does not increase the condition number of the system and demonstrates strong numerical stability in badly conditioned channels.

Acknowledgment

The authors would like to thank Dr. Behnaam Aazhang and Dr. Richard A. Tapia from Rice University for their instructive comments. J. R. Cavallaro was supported in part by NSF under grants ANI-9979465, EIA-0224458 and EIA-0321266.

References

[1] D. Gesbert, M. Shafi, D. Shiu, P. J. Smith and A. Naguib, *From Theory to Practice: An Overview of MIMO SpaceTime Coded Wireless Systems*, IEEE JSAC, VOL. 21, No.3, pp. 281-302, 2002.

[2] G. J. Foschini, *Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas*, Bell Labs Tech. J., pp. 41-59, 1996.

[3] G. D. Golden, J. G. Foschini, R. A. Valenzuela and P. W. Wolniansky, *Detection algorithm and initial laboratory results using V-BLAST space-time communication architecture*, Electron. Lett., vol. 35, pp.14-15, Jan. 1999.

[4] H. Holma, A. Tolskala, *Wideband CDMA for UMTS*, John Wiley and Sons, New York, NY, USA, 2000.

[5] A. Wiesel, L. Garca, J. Vidal, A. Pags and Javier R. Fonollosa, *Turbo linear dispersion space time coding for MIMO HSDPA systems*, 12th IST Summit on Mobile and Wireless Communications, Aveiro, Portugal, June 15-18, 2003.

[6] K. Hooli, M. Juntti, M. J. Heikkila, P. Komulainen, M. Latva-aho and J. Lilleberg, *Chip-level channel equalization in WCDMA downlink*, EURASIP Journal on Applied Signal Processing, pp. 757-770, Aug.2002.

[7] S. Das, C. Sengupta and J. R. Cavallaro, *Hardware design issues for a mobile unit for next generation CDMA systems*, Proc. SPIE Conf. Advanced Signal Processsing: algorithms, architectures and implementations, Vol.3461, San Diego, CA, July, 1998.

[8] S. Rajagopal, B. A. Jones, J. R. Cavallaro, *Task Partitioning Wireless Base-station Receiver Algorithms on Multiple DSPs and FPGAs*, International Conference on Signal Processing Applications and Technology (ICSPAT), Dallas, TX, October, 2000.

[9] L. Scharf, *Statistical Signal Processing: Detection, Estimation and Time Series Analysis*, Addison Wesley,1990.

[10] T. Kailath and J. Chun, *Generalized displacement structure for block-Toeplitz, Toeplitz-block, and Toeplitz-derived matrices*, SIAM J. Matrix Analysis and Applications, Vol. 15, No. 1. Jan. 1994.

[11] S. Chandrasekarnan and Ali H. Sayed, *Stablizing the generalized schur algorithm*, SIAM J. Matrix Analysis. Appl. Vol. 17, No. 4, pp 950-983, October 1996.

[12] M. J. Heikkila, K. Ruotsalainen and J. Lilleberg, *Space-time equalization using conjugate-gradient algorithm in WCDMA downlink*, IEEE Proceeding in PIMRC, pp. 673-677, 2002.

[13] P. Radosavljevic, J. R. Cavallaro and A. D. Baynast, *Implementation of channel equalization for MIMO systems in WCDMA downlink*, in proceeding of VTC fall, 2004.

[14] Y. Guo, J. Zhang, D. McCain, J. R. Cavallaro, *Scalable FPGA architectures for LMMSE-based SIMO chip equalizer in HSDPA downlink*, IEEE Asilomar Conference on Signals, Systems, and Computers, Vol. 2 , pp.2171-2175, Monterey, CA, November 9-12, 2003.

[15] Y. Guo, J. Zhang, D. McCain and J. R. Cavallaro, *Efficient MIMO equalization for downlink multi-code CDMA: complexity optimization and comparative study*, in the Proceeding of IEEE GlobeCom 2004, pp. 2513-2519, Dallas, TX, USA, Nov. 2004.

[16] Y. Guo, G. Xu, D. McCain, J. R. Cavallaro, *Rapid scheduling of efficient VLSI architectures for next-generation HSDPA wire-less system using Precision-C synthesizer*, Proc. IEEE Intl. Workshop on Rapid System Prototyping'03, San Diego, CA, pp. 179-185, June 2003.

[17] V. Y. Pan, A. Zheng, *Superfast algorithms for Cauchy-like matrix computations and extensions*, Linear algebra and its applications, 310, 83-108, 2000.

[18] V. Y. Pan, *Structured matrices and polynomials: unified superfast algorithms*, springer, 2001.

[19] J. G. Proakis, *Digital signal processing: princiles, algorithms and applications:* 3^{rd} *edition*, Prentice-Hall, 1996.

[20] I-METRA project consortium, *The IST-2000-30148 I-METRA project*, in http://www.ist-imetra.org.

[21] J. P. Kermoal, L. Schumacher, K. Pedersen and P. Mogensen, *A Stochastic MIMO Radio Channel Model with Experimental Validation*, IEEE JSAC, VOL. 20, NO. 6, pp.1211-1226,AUGUST 2002.

[22] H. Nguyen, J. Zhang, B. Raghothaman, *A Kalman-filter approach to equalization of CDMA downlink channels*, EURASIP J. Applied Sig. Proc., accepted for publication in 2004.

[23] R. A. Horn, C. R. Johnson, *Matrix analysis*, Cambridge University Press, New York, NY, USA, 1985.

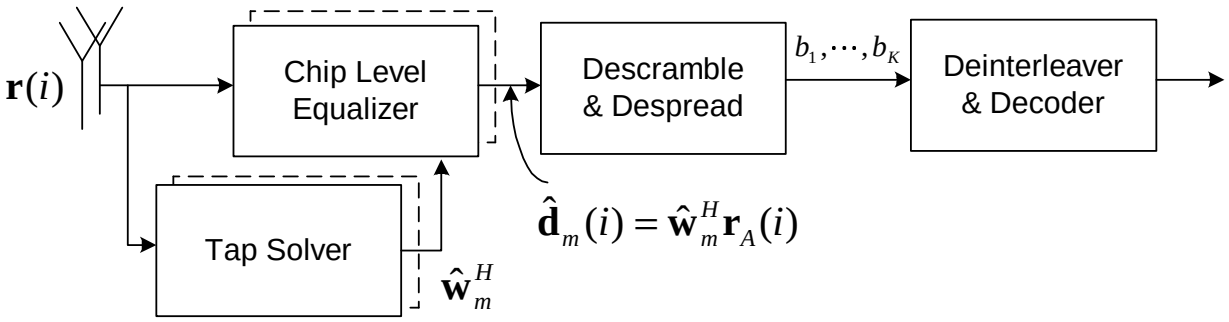

Figure 1: The block diagram of the LMMSE chip equalizer.

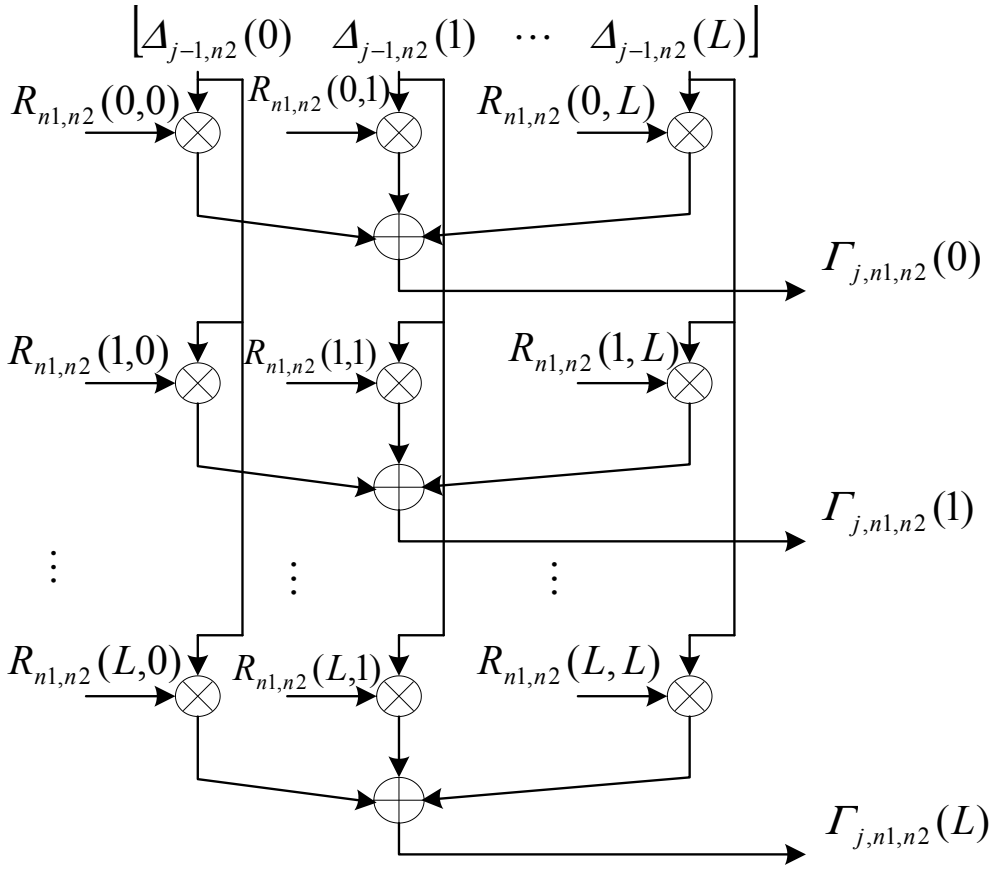

Figure 2: The straight-forward VLSI architecture for the DMM computation

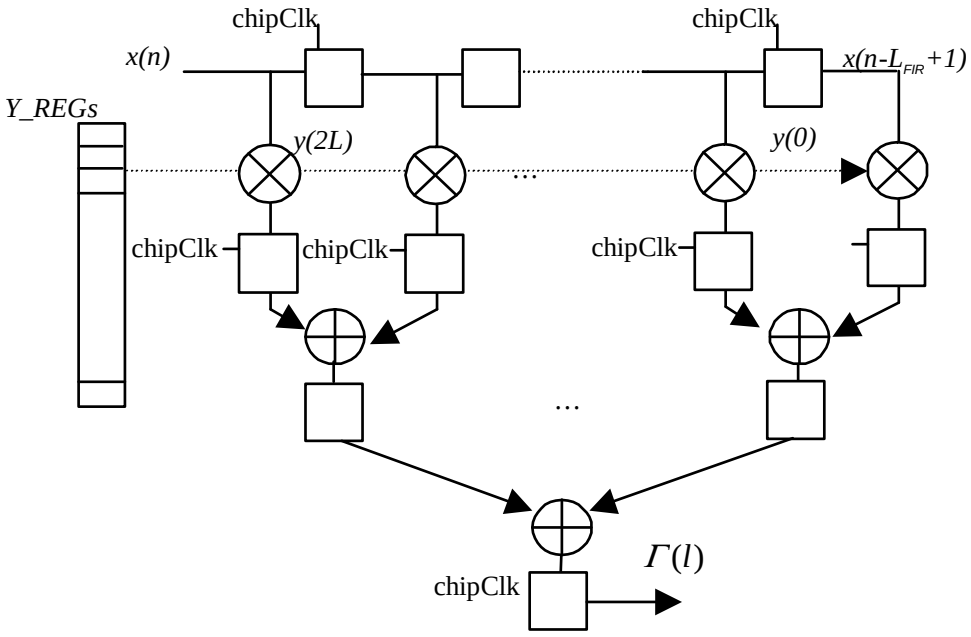

Figure 3: Direct-Form FIR filtering architecture for the matrix-vector multiplication in CG tap solver.

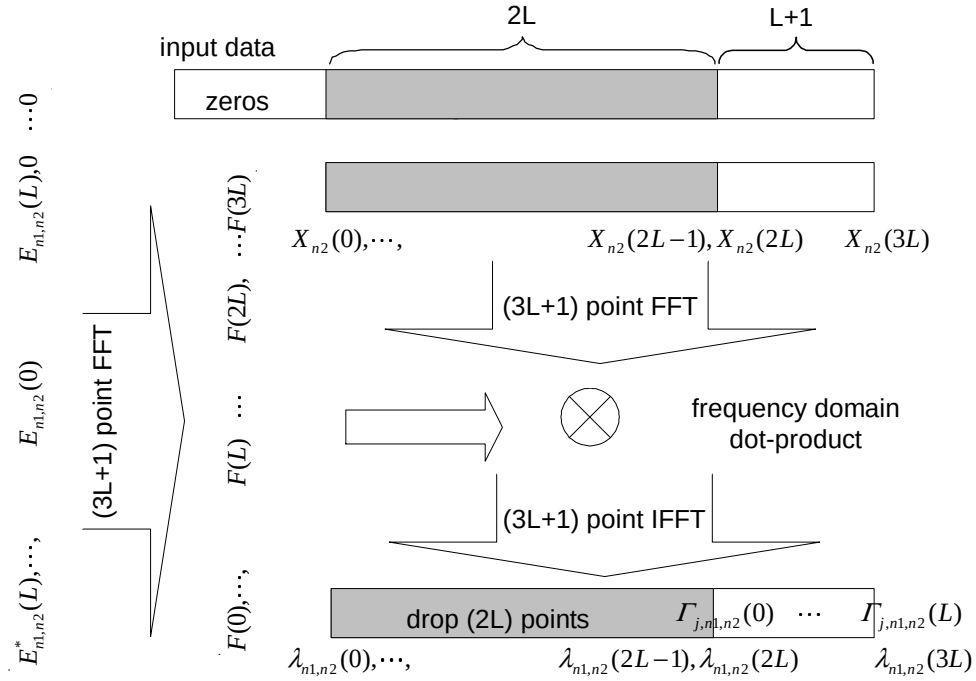

Figure 4: The "overlap-save" interpretation of the Expanded-Form FFT-acceleration architecture.

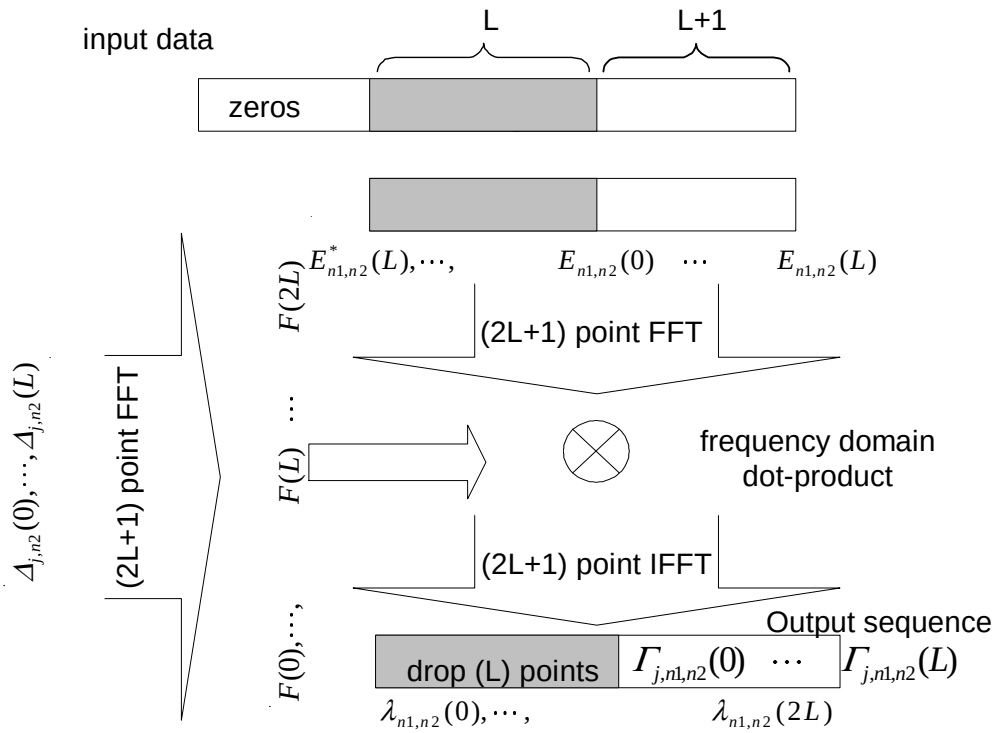

Figure 5: "Overlap-save" interpretation of the reversed-form FFT-acceleration architecture for complexity reduction.

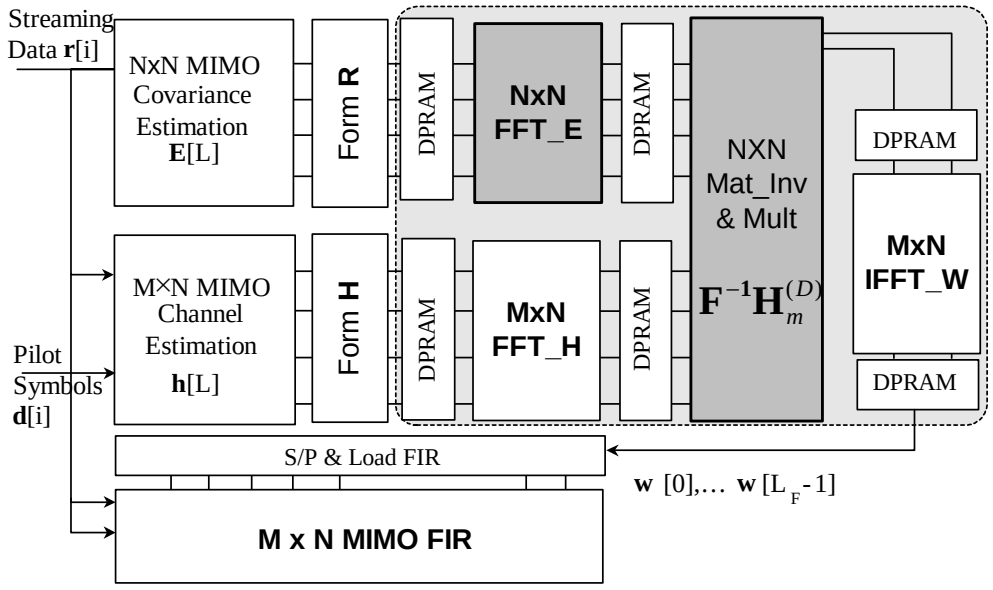

Figure 6: The block diagram of the FFT-acceleration MIMO equalizer using circulant approximation.

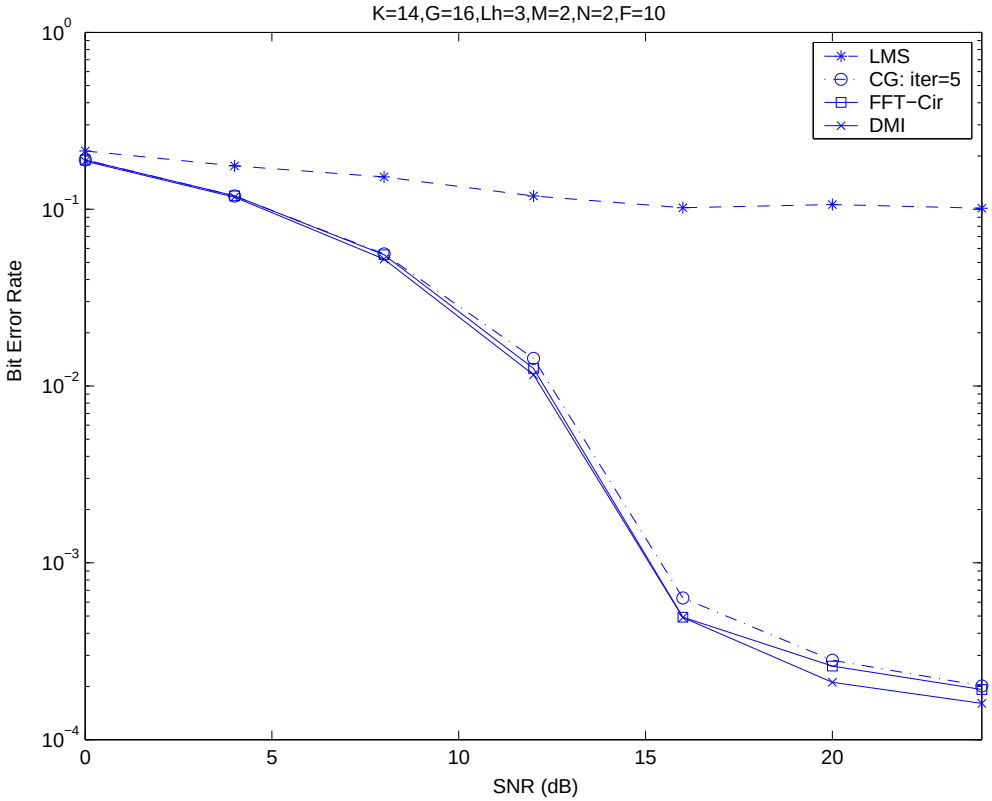

Figure 7: BER performance of 2×2 in Pedestrian-A channel.

Table 1: Complexity of Direct Matrix Multiplication (DMI)

Operation	Complexity
MULT	$(L+1)*(L+1)$
ADD	$(L+1)*L$
MEM	$(L+1)*(L+1)+2(L+1)$

Table 2: Condition number and reciprocal versus SNR: $L_h = 13$

SNR(dB)	0	4	8	12	16	20	24
$\kappa(\mathbf{R}_{rr})$	2.30	3.90	7.89	14.29	28.69	44.9	79.7
$\kappa(\mathbf{R}_{cir})$	2.70	5.09	10.99	21.48	53.52	417.8	1280
$\kappa_{rc}(\mathbf{R}_{rr})$	0.31	0.19	0.075	0.045	0.019	0.012	0.008
$\kappa_{rc}(\mathbf{R}_{cir})$	0.18	0.11	0.049	0.025	0.013	$5e-3$	$2.6e-3$

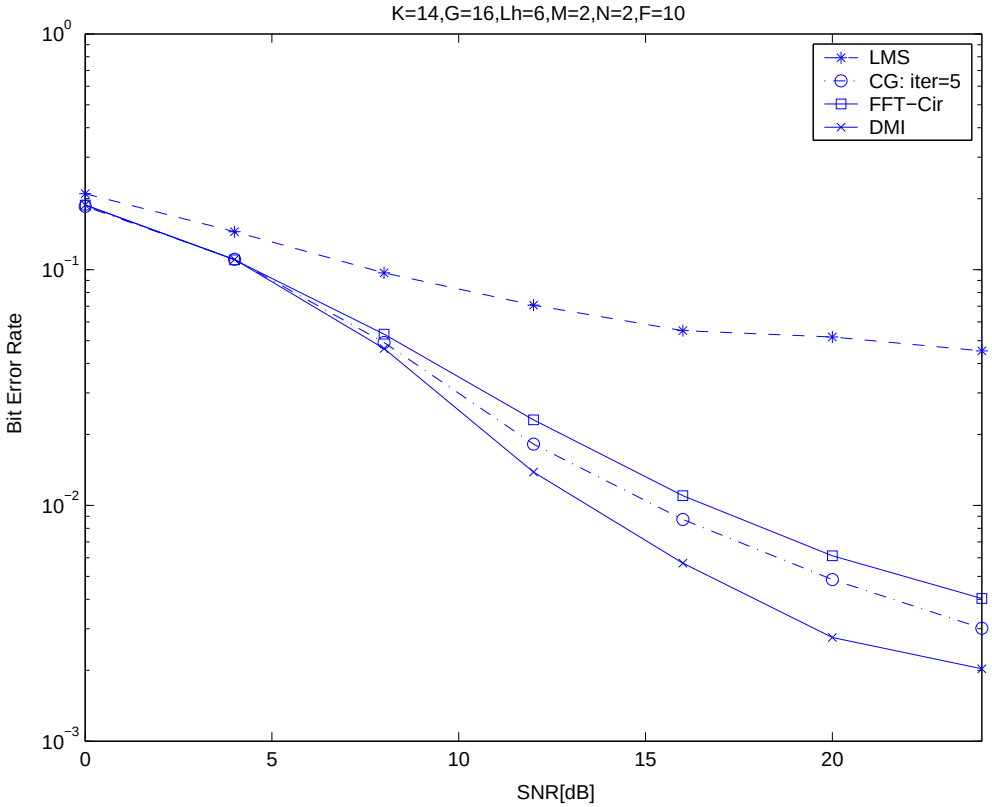

Figure 8: BER performance of 2×2 in Pedestrian-B channel case 2: K=10 codes.

Table 3: Condition number and reciprocal versus SNR: $L_h = 15$

SNR(dB)	0	4	8	12	16	20	24
$\kappa(\mathbf{R}_{rr})$	3.84	7.89	16.56	32.06	55.32	64.33	92.20
$\kappa(\mathbf{R}_{cir})$	5.98	14.07	33.66	75.15	$1.9e+2$	$1.54e+3$	$3.29e+3$
$\kappa_{rc}(\mathbf{R}_{rr})$	0.097	0.042	0.019	$1.06e-2$	$6.8e-3$	$5.7e-3$	$4.3e-3$
$\kappa_{rc}(\mathbf{R}_{cir})$	0.064	0.023	0.011	$5.4e-3$	$2.1e-3$	$2.76e-4$	$1.5e-4$

Table 4: Algorithm complexity comparison: Matrix-Vector Multiplication

Architecture	Number of Complex Multiplication
DMM/FIR-RF	$(L+1)*(L+1)$
FIR-EF	$(2*L+1)*(L+1)$
FFT-EF	$(3L+1)[3/2*\log_2(3L+1)+1]$
FFT-RF	$(2L+1)[3/2*log_2(2L+1)+1]$

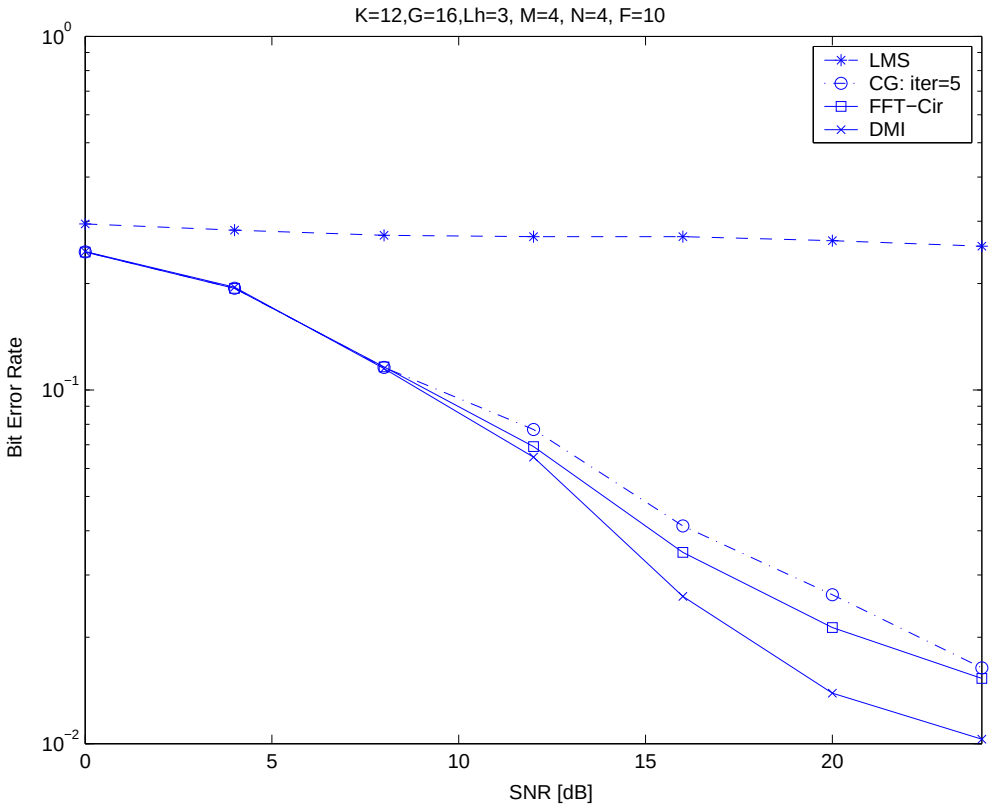

Figure 9: BER performance of 4×4 in Pedestrian-A channel.

Table 5: The overall tap solver complexity comparison

	Equivalent Complex Multiplication	Example
DMI	$\mathcal{O}\{(M+NF)(NF)^2\}$	92928
standard CG	$\mathcal{O}\{JM[(NF)^2+5NF]\}$	43120
FD Iterative	$\mathcal{O}[(MN^2+MNJ)L_F \log_2 L_F]$	15360
FFT Circulant	$\mathcal{O}\{[(N^2/2+2MN)\log_2 L_F+(N^3+MN^2)]L_F/2\}$	5248

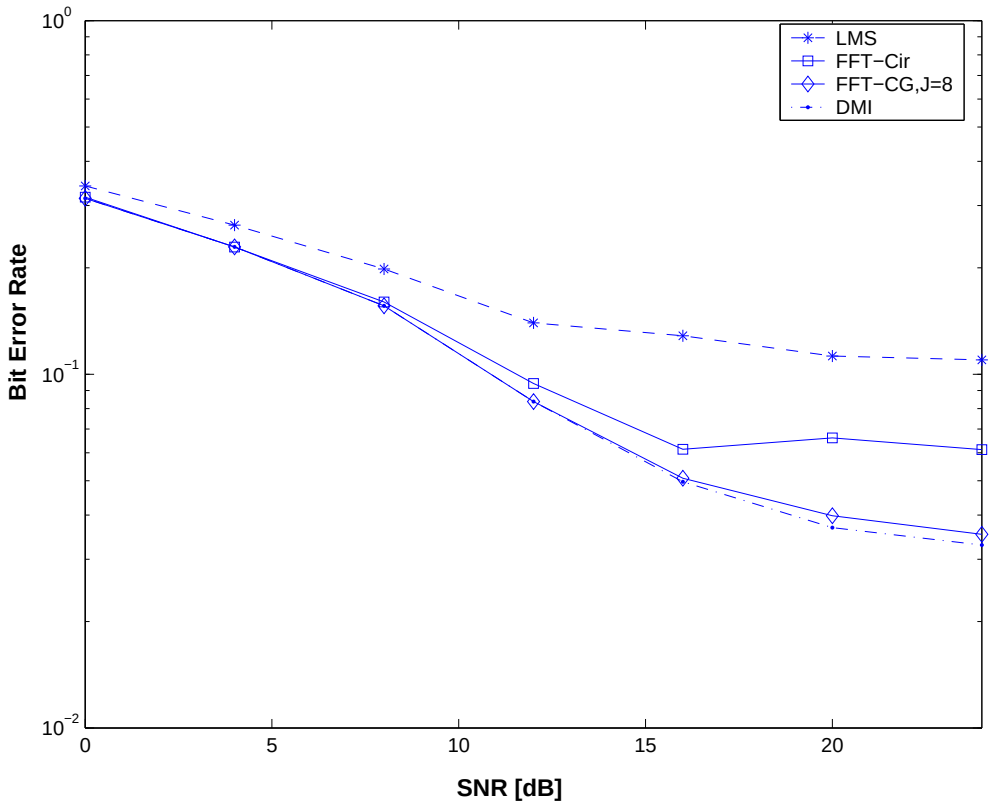

Figure 10: The BER performance of a fully loaded system ($K = 14$) under 13 taps modified Pedestrian-B channel for a 2×2 MIMO system.

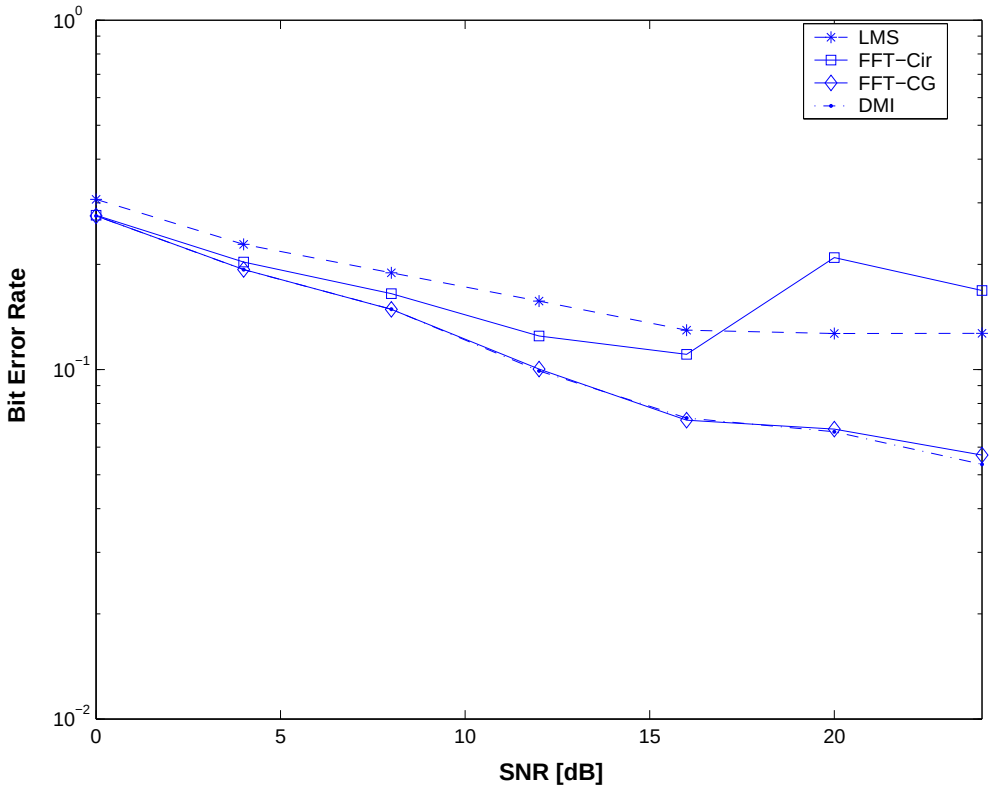

Figure 11: The BER performance of a fully loaded system ($K = 14$) under 15 taps modified Pedestrian-B channel for a 2×2 MIMO system.

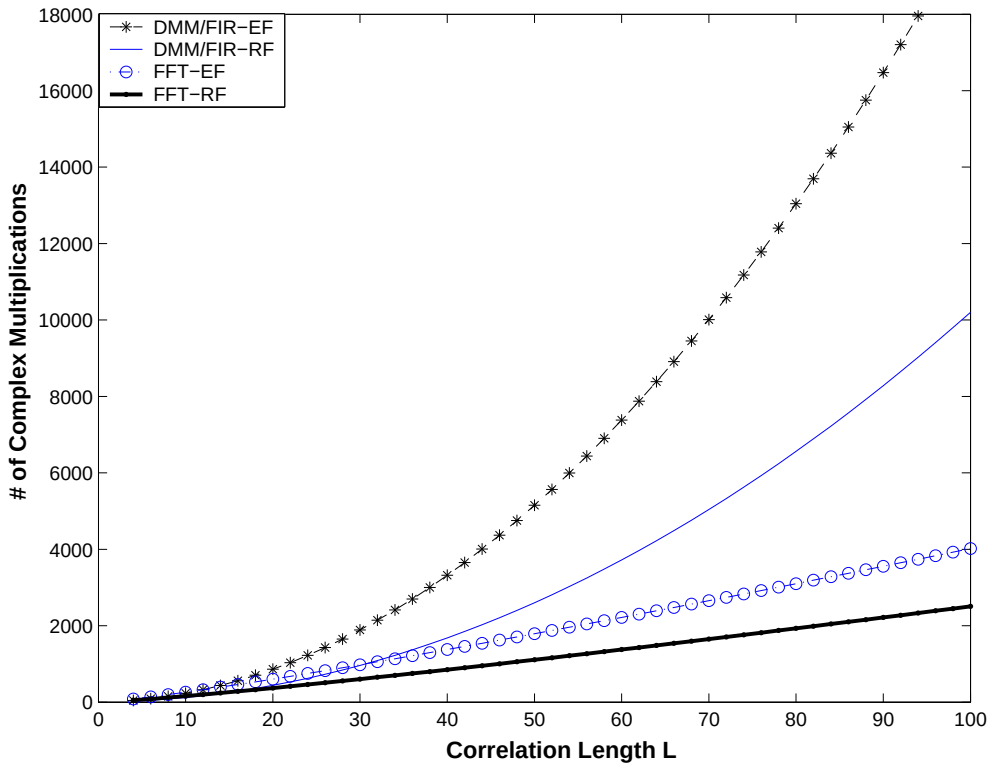

Figure 12: The complexity trend of the equivalent matrix-vector multiplication architectures.

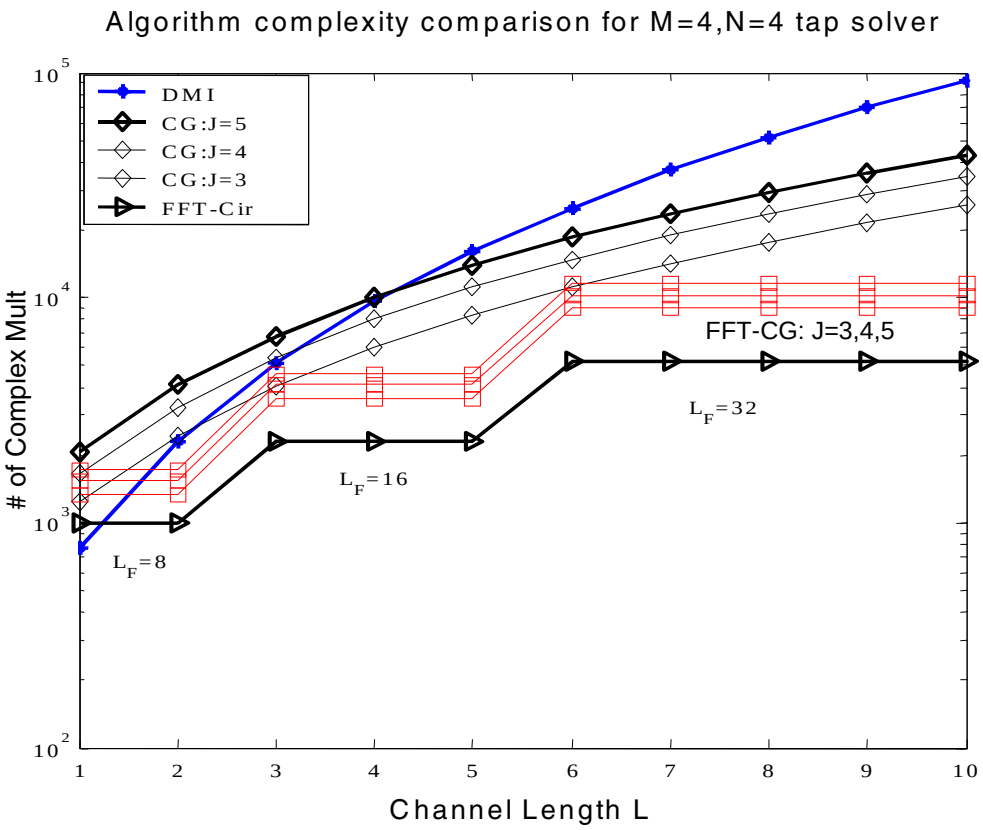

Figure 13: Overall tap solver complexity comparison.