Displacement MIMO Kalman Equalizer for CDMA Downlink in Fast Fading Channels

Yuanbin Guo, Jianzhong Zhang, Dennis McCain
Nokia Research Center
Irving, TX, 75039
{Yuanbin.Guo, Charlie.zhang, Dennis.McCain}@nokia.com

Joseph R. Cavallaro
Department of ECE, Rice University
Houston, TX, 77005
Email: Cavallar@rice.edu

***Abstract*— In this paper, a streamlined MIMO Kalman equalizer architecture is proposed to extract the commonality in the data path by jointly considering the displacement structure of the transition matrix and the block-Toeplitz structure of the channel matrix. Finally, an iterative Conjugate-Gradient based algorithm is proposed to avoid the inverse of the Hermitian symmetric innovation correlation matrix in Kalman gain processor. The proposed architecture not only reduces the numerical complexity to $O(F \log F)$ per chip, but also facilitates the parallel and pipelined VLSI implementation in real-time processing.**

I. Introduction

MIMO (Multiple-Input-Multiple-Output) technology using multiple antennas at both the transmitter and receiver sides has recently emerged as a significant breakthrough to increase the spectral efficiency dramatically. The original invention is known as D-BLAST [1] and a more realistic strategy is V-BLAST [2] by nulling and canceling with reasonable tradeoff between complexity and performance. Recently, MIMO extensions for the 3G wireless systems have received more and more attentions from the research community [3].

In a multipath-fading channel, the orthogonality of the spreading codes is destroyed and the Multiple-Access-Interference (MAI) along with the Inter-Symbol-Interference (ISI) are introduced. Linear Minimum-Mean-Square-Error (MMSE) based algorithms [4] [5] have demonstrated fairly good performance to suppress the interferences in slow-fading channels. However, they lack the tracking capability to the time-varying channels in fast-fading environments. Kalman filter [6] based on the state-space model of the dynamical system is known to provide the Best-Linear-Unbiased-Estimator (BLUE) of a linear system. The work of Iltis et al [7] developed the state-space model for estimation of the path gains and delays of multipath channels. Of notable importance is the work in [8], which associates the Kalman filter to the CDMA downlink system model and proposes the relevant state-space models for the channel equalization. Kalman filter is preferable in the fast-fading environments because of the following features: 1). It has the capability to track the non-stationarity in the time-variant channel, noise process etc; 2). It is an optimal BLUE estimator in the sense of MMSE.

However, the conventional Kalman equalizer has prohibitively high complexity for real-time hardware implementation. The Kalman filter involves an iterative computing structure to compute the Kalman gain and predict the state of the system. The complexity is dominated by numerous big size matrix-matrix multiplications and an inverse of the innovation correlation matrix in Kalman gain and new state estimation. For the purpose of real-time implementation, complexity reduction and the efficient architecture exploration are of essential importance.

In this paper, a parallel VLSI-oriented recursive architecture for MIMO Kalman equalizer is proposed by examining the timing relationship to extract the commonalities. We then explore the block-displacement structure in the state transition and Kalman gain to reduce the redundant multiplications dramatically. Numerical matrix-matrix multiplications with $O(F^3)$ complexity are eliminated by simple data loading process. A divide-and-conquer methodology is applied to partition the MIMO displacement structure into more tractable sub block architectures in the Kalman recursion. By utilizing the block Toeplitz structure of the channel matrix, an FFT-based acceleration is proposed to avoid direct matrix-matrix multiplications in the time domain for predicted state-error correlation matrix and Kalman gain. Finally, an iterative Conjugate-Gradient based algorithm is proposed to avoid the inverse of the Hermitian innovation correlation matrix in Kalman gain computer. The data path is streamlined by combining the displacement and block-Toeplitz structure. The proposed architecture not only reduces the numerical complexity to $O(F \log F)$ per chip, but also facilitates the parallel and pipelined real-time VLSI implementation. Simulation demonstrates significant complexity reduction and memory storage saving in the MIMO Kalman filter while keeping the performance gain over the conventional LMMSE-based chip equalizer.

II. MIMO CDMA System And State-Space Model

We consider the same MIMO CDMA downlink as in [5] based on spatial multiplexing with M Tx antennas and N Rx antennas. First, the high data rate symbols are demultiplexed into $U \times M$ lower rate substreams, where U is the number of spreading codes used in the system for data transmission. The substreams are broken into M groups, where each substream in the group is spreaded with a spreading code of spreading factor F. Each group of substreams are then combined and scrambled with long scrambling codes and transmitted through

0-7803-9415-1/05/$20.00 (C) 2005 IEEE

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2005 proceedings.

the t^{th} Tx antenna. The chip level signal at the t^{th} transmit antenna is given by $x_t(i) = \sum_{u=1}^{U} s_t^u(j)c_t^u(i)$ where j is the symbol index, i is chip index and u is the index of the composite spreading code. $s_t^u(j)$ is the j^{th} symbol of the u^{th} code at the t^{th} substream. In the following, we focus on the j^{th} symbol index and omit the index for simplicity. $c_t^u(i) = c^u(i)c_t^{(s)}(i)$ is the composite spreading code sequence for the u^{th} code at the t^{th} substream where $c^u(i)$ is the user specific Hadamard spreading code and $c_t^{(s)}(i)$ is the antenna specific scrambling long code. The received chip level signal at the r^{th} Rx antenna is given by

$$y_r(i) = \sum_{t=1}^{M}\sum_{l=0}^{L_{t,r}} h_{t,r}(l)x_t(i-\tau_l) + \nu_t(i). \quad (1)$$

The channel is characterized by a channel matrix between the t^{th} Tx and the r^{th} Rx antenna as $h_{t,r}(t) = \sum_{l=0}^{L_{t,r}} h_{t,r}(l)\delta(t-\tau_{t,r,l})$ where $\delta(t)$ is the Kronecker function. By collecting the F consecutive chips at the k^{th} symbol from each of the N Rx antennas in a signal vector $\mathbf{y}_r(k) = [y_r(kF+F-1)\ \cdots\ y_r(kF)]^T$ and packing the signal vectors from each receive antenna, we form a signal vector as $\mathbf{y}(k) = [\mathbf{y}_1(k)^T\ \cdots\ \mathbf{y}_r(k)^T\ \cdots\ \mathbf{y}_N(k)^T]^T$. Here F is the spreading gain. In vector form, the received signal can be given by

$$\mathbf{y}_r(k) = \sum_{t=1}^{M}\mathbf{H}_{t,r}(k)\mathbf{x}_t(k) + \mathbf{v}_r(k) = \mathbf{H}_r(k)\mathbf{x}(k) + \mathbf{v}_r(k) \quad (2)$$

where $\mathbf{v}_r(k)$ is the additive Gaussian noise. The transmitted chip vector for the t^{th} transmit antenna is given by $\mathbf{x}_t(k) = [x_t(kF+F-1)\ \cdots\ x_t(kF)\ \cdots\ x_t(kF-D)]^T$ and the overall transmitted signal vector is given by stacking the substreams for multiple transmit antennas as $\mathbf{x}(k) = [\mathbf{x}_1(k)^T\ \cdots\ \mathbf{x}_t(k)^T\cdots\ \mathbf{x}_M(k)^T]^T$. D is the channel delay spread. The channel matrix from multiple transmit antennas is defined as $\mathbf{H}_r(k) = [\mathbf{H}_{1,r}(k)\ \cdots\ \mathbf{H}_{t,r}(k)\ \cdots\ \mathbf{H}_{M,r}(k)]$, where $\mathbf{H}_{t,r}(k)$ is the channel matrix from the t^{th} transmit antenna and r^{th} receive antenna.

The Kalman filter problem is a solution to estimate the state $\mathbf{x}(k)$ given the entire observed data $\mathbf{y}(1),\ \cdots,\ \mathbf{y}(k)$. It is derived from the state-space model consisting of a measurement equation and a process equation. The measure equation describes the generation model of the observation $\mathbf{y}$ from the state $\mathbf{x}$ in a stochastic noise process. The process equation describes the state transition of the new estimate $\mathbf{x}(k)$ at time k from the estimate $\mathbf{x}(k-1)$ at time $k-1$. By defining the transition matrix as $\mathbf{\Theta}(k)$, it is natural to have the measurement equation as the received signal model and the process equation as an excitation of some process noise:

$$\mathbf{y}(k) = \mathbf{H}(k)\mathbf{x}(k) + \mathbf{v}(k); \quad (3)$$
$$\mathbf{x}(k) = \mathbf{\Theta}(k)\mathbf{x}(k-1) + \mathbf{w}(k); \quad (4)$$

where the measure matrix is the overall MIMO channel matrix $\mathbf{H}(k)$ given by $\mathbf{H}(k) = [\mathbf{H}_1(k)^T, \cdots, \mathbf{H}_r(k)^T, \cdots, \mathbf{H}_N(k)^T]$. Moreover $\mathbf{v}(k)$ denotes the measurement noise and $\mathbf{w}(k)$ denotes the process noise.

TABLE I

SUMMARY OF THE COMMONALITY EXTRACTED KALMAN PROCEDURE

Init :
$\hat{\mathbf{x}}(0\|0) = E[\mathbf{x}(0)]$;
$\mathbf{P}(0\|0) = E\{[\mathbf{x}(0) - \hat{\mathbf{x}}(0\|0)][\mathbf{x}(0) - \hat{\mathbf{x}}(0\|0)]^H\}$
Input vector : $\mathbf{y}(k)$; Output vector : $\hat{\mathbf{x}}(k\|k)$;
Predefined parameters :
Transition matrix = $\mathbf{\Theta}(k)$; Measure matrix = $\mathbf{H}(k)$;
Correlation matrix of the process noise : $\mathbf{Q}_w(k) = E[\mathbf{w}(k)\mathbf{w}^H(k)]$;
Correlation matrix of the measure noise : $\mathbf{Q}_v(k) = E[\mathbf{v}(k)\mathbf{v}^H(k)]$.
Recursion for $k = 1, 2, \cdots$
(1). State transition equations :
$\hat{\mathbf{x}}(k\|k-1) = \mathbf{\Theta}(k)\hat{\mathbf{x}}(k-1\|k-1)$;
$\mathbf{P}(k\|k-1) = \mathbf{\Theta}(k)\mathbf{P}(k-1\|k-1)\mathbf{\Theta}(k)^H + \mathbf{Q}_w(k)$;
(2). Innovation generation :
$\alpha(k) = \mathbf{y}(k) - \mathbf{H}(k)\hat{\mathbf{x}}(k\|k-1)$;
$\mathbf{\Omega}(k) = \mathbf{H}(k)\mathbf{P}(k\|k-1)$;
(3). Kalman gain computation :
$\mathbf{R}(k) = \mathbf{H}(k)\mathbf{\Omega}^H(k) + \mathbf{Q}_v(k)$;
$\mathbf{G}(k) = \mathbf{\Omega}^H(k)\mathbf{R}^{-1}(k)$;
(4). State estimate & Predicted state error correlation update :
$\hat{\mathbf{x}}(k\|k) = \hat{\mathbf{x}}(k\|k-1) + \mathbf{G}(k)\alpha(k)$;
$\mathbf{P}(k\|k) = \mathbf{P}k\|k-1) - \mathbf{G}(k)\mathbf{\Omega}(k)$.

III. DISPLACEMENT MIMO KALMAN EQUALIZER

A. Commonality Extracted Architecture

It is assumed that the readers have basic knowledge of Kalman filter theory. In this section, we extract the commonality in the conventional Kalman procedure and streamline the computing architecture with a study of the timing dependency and the physical meaning of the Kalman procedure. An innovation process and the correlation matrix of the innovation process are defined by

$$\alpha(k) = \mathbf{y}(k) - \hat{\mathbf{y}}(k|k-1), \quad (5)$$
$$\mathbf{R}(k) = E[\alpha(k)\alpha^H(k)]. \quad (6)$$

whose physical meaning represents the new information in the observation data $\mathbf{y}(k)$. $\hat{\mathbf{y}}(k|k-1)$ denotes the MMSE estimation of the observed data at time k, given all the past observed data from time 1 to $k-1$. It is shown that $\mathbf{R}(k) = \mathbf{H}(k)\mathbf{P}(k|k-1)\mathbf{H}^H(k) + \mathbf{Q}_v(k)$ where the $\mathbf{P}(k|k-1)$ matrix is the predicted state error correlation matrix defined by $\mathbf{P}(k|k-1) = E[\varepsilon(k|k-1)\varepsilon^H(k|k-1)]$. Here $\varepsilon(k|k-1) = \hat{\mathbf{x}}(k) - \hat{\mathbf{x}}(k|k-1)$ is the predicted state error vector at time k using data up to time $k-1$. By defining a Kalman gain as $\mathbf{G}(k) = E[\mathbf{x}(k)\alpha^H(k)]\mathbf{R}^{-1}(k)$, the new state estimate can be given by

$$\hat{\mathbf{x}}(k|k) = \mathbf{\Theta}(k)\mathbf{x}(k-1|k-1) + \mathbf{G}(k)\alpha(k). \quad (7)$$

The Riccati equation provides a recursive computation procedure of the predicted state error correlation matrix $\mathbf{P}(k|k-1)$ and the Kalman gain. By analyzing the data dependency and the timing relationship, the streamlined procedure is given in Table I.

B. MIMO Displacement Structure

Despite of the streamlining to reduce redundancy, the computation complexity still remains the same order. Both the matrix inverse and the matrix-matrix multiplication have

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2005 proceedings.

$O(F^3)$ complexity for an $F \times F$ matrix. In this section, we will show that because the transition matrix has some displacement structure, the matrix multiplication complexity can be dramatically reduced. Some explicit matrix multiplications are eliminated by simple data-loading process of a small portion of the full matrix. It can be shown that the transition matrix can be designed as following

$$\mathbf{\Theta}(k) = \mathbf{I}_M \otimes \tilde{\mathbf{\Theta}}(k) \quad (8)$$

$$\tilde{\mathbf{\Theta}}(k) = \begin{pmatrix} \mathbf{0}_{F\times D} & \mathbf{0}_{F\times F} \\ \mathbf{I}_{D\times D} & \mathbf{0}_{D\times F} \end{pmatrix}. \quad (9)$$

where $\otimes$ denotes the Kronecker product. It is assumed that $D < F$ in most situations. The process noise is then given by $\mathbf{w}(k) = [\mathbf{w}_1(k)^T \ \cdots \ \mathbf{w}_t(k)^T \ \cdots \ \mathbf{w}_M(k)^T]^T$ where the process noise for the t^{th} transmit antenna is given by $\mathbf{w}_t(k) = [x_t(kF+F-1) \ \cdots \ x_t(kF) \ 0 \ \cdots \ 0]^T$. It is easy to verify that to pre-multiply a matrix with $\tilde{\mathbf{\Theta}}(k)$ is equivalent to shifting the first D rows of the matrix to the bottom and adding F rows of zeros to the upper portion. To post-multiply a matrix with is equivalent to shifting the first D columns of the matrix to the right and adding F rows of zeros to the left portion. For the MIMO case, the feature forms a block-displacement structure and will be applied to related computations.

1) State transition equation: It is shown that the state transition equation can be partitioned into M transmit antennas using the Kronecker product. $\hat{\mathbf{x}}(k|k-1) = [\hat{\mathbf{x}}_1(k|k-1)^T \ \cdots \ \hat{\mathbf{x}}_t(k|k-1)^T \ \cdots \hat{\mathbf{x}}_M(k|k-1)^T]^T$. Thus, the t^{th} sub block of the transition is given by $\hat{\mathbf{x}}_t(k|k-1) = \tilde{\mathbf{\Theta}}(k)\hat{\mathbf{x}}_1(k-1|k-1) = [\mathbf{0}_{1\times F} \ \hat{\mathbf{x}}_1^U(k-1|k-1)^T]^T$, where $\hat{\mathbf{x}}_t^U(k-1|k-1) \equiv [\hat{x}_t(k-1|k-1,0) \ \cdots \ \hat{x}_t(k-1|k-1,D-1)]^T$ is the upper D rows of the previous state. This process is shown in the Fig. 1.

2) Filtered state estimation output & feedback: This displacement structure can be further applied in the filtered state estimation and feedback process. Similarly we can partition the update equation $\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{G}(k)\alpha(k)$ into $\hat{\mathbf{x}}_t(k|k) = \hat{\mathbf{x}}_t(k|k-1) + \mathbf{G}_t(k)\alpha(k)$ where $\hat{\mathbf{x}}(k|k) = [\cdots \ \hat{\mathbf{x}}_t(k|k) \ \cdots]$ and $\mathbf{G}(k) = [\cdots \ \mathbf{G}_t(k)^T \ \cdots]$. We further partition the element-wise state estimate and the Kalman gain into three sub-blocks, the upper D rows, the lower D rows and the rest rows in the center as $\hat{\mathbf{x}}_t(k|k) = [\hat{\mathbf{x}}_t^U(k|k)^T \ \hat{\mathbf{x}}_t^C(k|k)^T \ \hat{\mathbf{x}}_t^L(k|k)^T]^T$ and $\mathbf{G}_t(k)^T = [\mathbf{G}_t^U(k)^T \ \mathbf{G}_t^C(k)^T \ \mathbf{G}_t^L(k)^T]^T$. We define the effective transition state vector as the lower D rows of the state at time $(k-1)$. It can be shown from the transition that the upper and center portions of the new states do not need to add the previous states. Only the lower portion is updated from the previous state with the Kalman gain. Then the new effective transition state vector is simply a copy of the new upper portion of the state. In the real-time implementation, only this portion is stored and feeded back to form the state transition.

$$\hat{\mathbf{x}}_t^U(k|k) = \mathbf{G}_t^U(k)\alpha(k) \quad (10)$$

$$\hat{\mathbf{x}}_t^C(k|k) = \mathbf{G}_t^C(k)\alpha(k) \quad (11)$$

$$\hat{\mathbf{x}}_t^L(k|k) = \chi_t^L(k-1) + \mathbf{G}_t^L(k)\alpha(k) \quad (12)$$

$$\chi_t^L(k) = \hat{\mathbf{x}}_t^U(k|k). \quad (13)$$

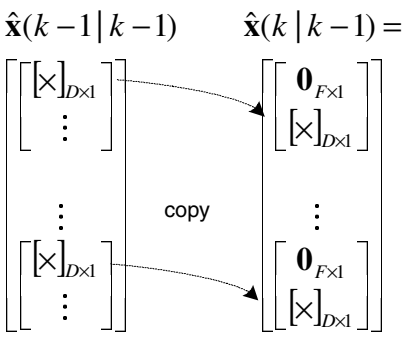

Fig. 1. The displacement based architecture for state transition.

3) Predicted state error correlation matrix: Another process involved with the transition matrix is the computation of the predicted state error correlation matrix $\mathbf{P}(k|k-1) = \mathbf{\Theta}(k)\mathbf{P}(k-1|k-1)\mathbf{\Theta}(k)^H + \mathbf{Q}_w(k)$. It is shown that the process noise correlation is given by

$$\mathbf{Q}_w(k) = E\{\mathbf{w}(k)\mathbf{w}(k)^H\} = \mathbf{I}_M \otimes \mathbf{Q}(k) \quad (14)$$

$$\mathbf{Q}(k) = \begin{pmatrix} \tilde{\mathbf{Q}}_w(k) & \mathbf{0}_{F\times D} \\ \mathbf{0}_{D\times F} & \mathbf{0}_{D\times D} \end{pmatrix}. \quad (15)$$

Thus if we span the MIMO correlation matrix from the sub blocks as $\mathbf{P}(k|k-1) = \{\mathbf{P}_{t_1,t_2}(k|k-1)\}$ and $\mathbf{P}(k-1|k-1) = \{\mathbf{P}_{t_1,t_2}(k-1|k-1)\}$ for t_1 and $t_2 \in [1, M]$, we can get the partitioned sub blocks given by $\mathbf{P}_{t_1,t_2}(k|k-1) = \tilde{\mathbf{\Theta}}(k)\mathbf{P}_{t_1,t_2}(k-1|k-1)\tilde{\mathbf{\Theta}}(k)^H + \mathbf{Q}_{t_1,t_2}(k)$ and $\mathbf{Q}_{t_1,t_2}(k) = \mathbf{Q}(k)\delta(t_1-t_2)$.

With the feature of the pre-multiplication and post-multiplication with the displacement transition matrix, we can show that the new state error correlation matrix is given by the following partitioning,

$$\begin{cases} \mathbf{P}_{t_1,t_2}(k|k-1, F:F+D-1, F:F+D-1) = \rho_{t_1,t_2}(k-1) \\ \mathbf{P}_{t,t}(k|k-1, 0:F-1, 0:F-1) = \tilde{\mathbf{Q}}_w(k) \\ \mathbf{P}_{t_1,t_2}(k|k-1, i, j) = 0, \qquad o.w. \end{cases}$$

where the sub block matrix $\rho_{t_1,t_2}(k-1)$ is a $D \times D$ left-upper corner of the partitioned correlation matrix defined by $\rho_{t_1,t_2}(k-1) = \mathbf{P}_{t_1,t_2}(k-1|k-1, 0:D-1, 0:D-1)$.

Thus, the matrix multiplications and additions in computing $\mathbf{P}(k|k-1)$ from $\mathbf{P}(k-1|k-1)$ are all eliminated. Logically we only need to copy some small sub-blocks of $\mathbf{P}(k-1|k-1)$ to $\mathbf{Q}_w(k)$ following the special pattern. Actually, the storage of the full matrix is not necessary since the matrix is sparse with many zero entries. This displacement procedure is demonstrated by the data loading process in Fig. 2.

4) Update State Error Correlation Matrix: Jointly considering the feedback data path of $\mathbf{P}(k|k)$ and the displacement structure in $\mathbf{P}(k|k-1)$, it is clear that only the upper left corner $\rho_{t_1,t_2}(k-1)$ are utilized for the element matrix $\mathbf{P}_{t_1,t_2}(k-1|k-1)$. The other elements are redundant information that will be dropped during the displacement procedure. Thus, there is no need to compute and keep these components. Because there is matrix multiplication of the Kalman gain with $\mathbf{\Omega}(k)$ as in $\mathbf{P}(k|k) = \mathbf{P}(k|k-1) - \mathbf{G}(k)\mathbf{\Omega}(k)$, we define an

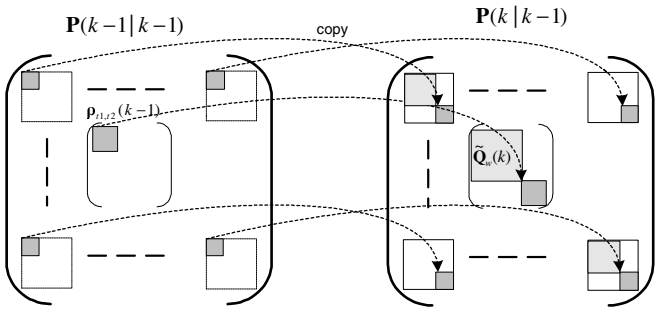

Fig. 2. The displacement based architecture for state transition.

intermediate variable $\mathbf{\Psi}(k)$ for the multiplication and partition it to MIMO sub-blocks as $\mathbf{\Psi}(k) = \mathbf{G}(k)\mathbf{\Omega}(k) = \{\mathbf{\Psi}_{mm}(k)\}$ for $m = [1, M]$. Instead of computing the full matrix of $\mathbf{P}(k|k)$, we only need to compute the relevant submatrices given by $\rho_{t_1,t_2}(k) = \mathbf{P}_{t_1,t_2}(k|k-1, 0:D-1, 0:D-1) + \mathbf{\Psi}_{t_1,t_2}(k, 0:D-1, 0:D-1)$.

We also partition the Kalman Gain $\mathbf{G}(k)$ and the $\mathbf{\Omega}(k)$ matrices into MIMO sub blocks as $\mathbf{G}(k) = \{\mathbf{G}_{t,r}(k)\}$ and $\mathbf{\Omega}(k) = \{\mathbf{\Omega}_{t,r}(k)\}$ where $\mathbf{G}_{t,r}(k) = [\mathbf{G}^U_{t,r}(k)^T \quad \mathbf{G}^L_{t,r}(k)^T]^T$ is further partitioned into the upper and lower sub-matrices while $\mathbf{\Omega}_{r,t}(k) = [\mathbf{\Omega}^L_{r,t}(k) \quad \mathbf{\Omega}^R_{r,t}(k)]$ is partitioned into the left and right sub-matrices of the following sizes as

$$\mathbf{G}^U_{t,r}(k) : upper \quad D \times F \tag{16}$$

$$\mathbf{G}^U_{t,r}(k) : lower \quad F \times F \tag{17}$$

$$\mathbf{\Omega}^L_{r,t}(k) : left \quad F \times D \tag{18}$$

$$\mathbf{\Omega}^R_{r,t}(k) : right \quad F \times F \tag{19}$$

It is clear that the element block in the $\mathbf{\Psi}(k)$ is given by

$$\mathbf{\Psi}_{t_1,t_2}(k) \quad = \quad \sum_{r=1}^{N} \mathbf{G}_{t_1,r}(k)\mathbf{\Omega}_{r,t_2}(k) \tag{20}$$

Comparing the displacement structure, only the left-upper corner of size $D \times D$ is necessary, which is given by $\widetilde{\mathbf{\Psi}}_{t_1,t_2}(k) = \mathbf{\Psi}_{t_1,t_2}(k, 0:D-1, 0:D-1) = \sum_{r=1}^{N} \mathbf{G}^U_{t_1,r}(k)\mathbf{\Omega}^L_{r,t_2}(k)$. This is only associated with the upper part of $\mathbf{G}_{t_1,r}(k)$ and left part of $\mathbf{\Omega}_{r,t_2}(k)$. As a summary, the updated effective state error correlation is simplified by adding the correction item to the $D \times D$ corner of $\tilde{\mathbf{Q}}_w(k)$ which is constant to the transmit antenna elements t_1 and t_2 as $\rho_{t_1,t_2}(k) = \tilde{\mathbf{Q}}_w(k, 0:D-1, 0:D-1)\delta(t_1 - t_2) + \widetilde{\mathbf{\Psi}}_{t_1,t_2}(k)$. This optimization not only saves many computations and memory storage but also fastens the update and feedback time.

IV. FFT-Acceleration

In the innovation and the omega matrix generation, there are some pre-multiplications by the channel matrix $\mathbf{H}(k)$ as in and $\alpha = \mathbf{y}(k) - \mathbf{H}(k)\hat{\mathbf{x}}(k|k-1)$ and $\mathbf{\Omega} = \mathbf{H}(k)\mathbf{P}(k|k-1)$. We define the estimated observation and partition it into the sub-vectors for the multiple receive antennas as $\hat{\mathbf{y}}(k) = \mathbf{H}(k)\hat{\mathbf{x}}(k|k-1) = [\sum_{t=1}^{M} \mathbf{H}_{t,1}(k)\hat{\mathbf{x}}_t(k|k-1), \cdots, \sum_{t=1}^{M} \mathbf{H}_{t,N}(k)\hat{\mathbf{x}}_t(k|k-1)]^T$. Since the channel matrix from the t^{th} transmit antenna and r^{th} receive antenna $\mathbf{H}_{t,r}(n)$ has the Toeplitz structure as

$$\mathbf{H}_{t,r}(n) = \begin{pmatrix} h^r_{t,0} & \cdots & h^r_{t,D} & & 0 \\ & \ddots & & \ddots & \\ 0 & & h^r_{t,0} & \cdots & h^r_{t,D} \end{pmatrix}, \tag{21}$$

the matrix-vector multiplication can be viewed as a FIR filter with the channel impulse response $[h^r_{t,D}, \cdots, h^r_{t,0}]$. This can be implemented in the time domain by delayed tap line architecture as a conventional FIR. It is well known that the time-domain FIR filtering can also be implemented by FFT-based circular convolution in the frequency domain. The similar architecture can be applied directly to the Kalman filtering problem in this paper. This achieves $O((F+D)\log(F+D))$ complexity algorithm versus $O((F+D)^2)$ for the matrix-vector multiplication and $O((F+D)^2\log(F+D))$ versus $O((F+D)^3)$ for the matrix-matrix multiplications in the innovation estimation and the Kalman Gain computer. The procedure is described briefly as following:

1) Take the FFT of the zero-padded channel impulse response $[h^r_{t,D}, \cdots, h^r_{t,0}, 0, \cdots, 0]$;
2) Take the FFT of the right-product vector $\hat{\mathbf{x}}_t(k|k-1)$;
3) Compute the dot product of the frequency-domain coefficients;
4) Take the IFFT of the product;
5) Truncate the result to get the valid coefficients as the matrix-vector multiplication result.

First, the element-wise FFT bank computes the frequency coefficients of each of the zero-padded MIMO channel impulse response. Simultaneously, another FFT bank computes the dimension-wise coefficients of the estimated state. The dot product of the two groups of coefficients are computed according to the transmit antenna index t. Then the results are grouped by the receive antenna index r by summing the result for all the transmit antennas. A dimension-wise FFT-bank with N IFFTs computes the dot products correspondingly and truncates the result according to the "over-lap save" architecture to generate estimated observation. For a matrix-matrix multiplication involved with the block-Toeplitz channel matrix, we can extend the matrix-vector multiplication architecture to multiple vectors in a straightforward way. Note that we only need to take FFT once for each channel impulse response. For the multiple vectors to be filtered, we can form a pipelined FFT computation to use the hardware resource efficiently.

V. Iterative Inverse Solver

With the afore-mentioned optimizations, the complexity has been reduced dramatically. However, there is one last hard work in computing the Kalman gain as in

$$\mathbf{G}(k) = \mathbf{\Omega}^H \mathbf{R}^{-1}(k). \tag{22}$$

It is known that a Gaussian elimination can be applied to solve the matrix inverse with complexity at the order of $O[(NF)^3]$. Moreover, Cholesky decomposition can also be applied to accelerate the speed by reducing the hidden constant factor in

TABLE II

SUMMARY OF THE CG PROCEDURE FOR THE $\mathbf{\Phi}(k) = \mathbf{R}^{-1}\alpha(k)$

(1). Initialization
$\mathbf{\Phi}_0 = \mathbf{0}$;
$\gamma_0 = \alpha(k)$; $\quad \mathbf{\Delta}_0 = \alpha(k)$;
$\delta_0 = \gamma_0^H \cdot \gamma_0$; $\quad \delta_1 = \delta_0$;
(2). For an iteration from $j = 1 : J$ until convergence:
$\mathbf{\Gamma}_j = \mathbf{R} \cdot \mathbf{\Delta}_{j-1}$; $\quad \delta_{j+1} = \mathbf{\Gamma}_j^H \mathbf{\Gamma}_j$;
$\mu = \delta_j / \mathbf{\Delta}_{j-1}^H \mathbf{\Gamma}_j$; $\quad \nu = \delta_{j+1}/\delta_j$
$\mathbf{\Phi}_j = \mathbf{\Phi}_{j-1} + \mu \mathbf{\Delta}_{j-1}$;
$\gamma_j = \gamma_{j-1} - \mu \mathbf{\Gamma}_j$;
$\mathbf{\Delta}_j = \mathbf{\Gamma}_j + \nu \mathbf{\Delta}_{j-1}$.

the order of complexity. However, since these two solutions do not use the structure of the matrix, the complexity is at the same order as to solve the inverse of a general matrix.

We made the observation that (1). $\mathbf{R}$ is a $(NF \times NF)$ Hermitian symmetric matrix. This can be easily verified as $\mathbf{R}^H(k) = \mathbf{\Omega}(k)\mathbf{H}(k)^H + \mathbf{Q}_v^H(k) = \mathbf{H}(k)\mathbf{P}(k|k-1)\mathbf{H}(k)^H + \mathbf{Q}_v(k) = \mathbf{R}(k)$ because $\mathbf{P}(k|k-1) = \mathbf{P}(k|k-1)^H$ and $\mathbf{Q}_v(k) = \mathbf{Q}_v^H(k)$ are also Hermitian symmetric. It is known that the iterative Conjugate Gradient algorithm can solve the inverse of this type of matrix more efficiently. (2). The full matrix of the $\mathbf{G}$ is not necessary from the displacement structure of the state transition matrix. Only the lower $D \times NF$ ($\mathbf{G}_t^L$) and the left upper $D \times D$ ($\mathbf{G}_{t,r}^U$) corner are required. This feature can also be used to optimize the matrix inverse and the matrix multiplication involved in the Kalman Gain computation.

To avoid the direct inverse of $\mathbf{R}$ using the iterative CG algorithm, the Kalman gain computation and the state update is re-partitioned to generate the following new problem.

$$\begin{aligned} \chi(k) &= \mathbf{G}(k)\alpha(k) = \mathbf{\Omega}^H(k)[\mathbf{R}^{-1}(k)\alpha(k)] = \mathbf{\Omega}^H(k)\mathbf{\Phi}(k) \\ \mathbf{\Psi}(k) &= \mathbf{G}(k)\mathbf{\Omega}(k) = \mathbf{\Omega}^H(k)[\mathbf{R}^{-1}(k)\mathbf{\Omega}(k)] = \mathbf{\Omega}^H(k)\mathbf{\Pi}(k) \end{aligned}$$

where $\mathbf{\Phi}(k) = \mathbf{R}^{-1}(k)\alpha(k)$ and $\mathbf{\Pi}(k) = \mathbf{R}^{-1}(k)\mathbf{\Omega}(k)$ respectively. With this changed order of computation, the iterative procedure of the CG-based algorithm is shown as following.

A. Computation of $\mathbf{\Phi}(k) = \mathbf{R}^{-1}(k)\alpha(k)$

The computation of $\mathbf{\Phi}(k)$ is a direct application of the iterative Conjugate-Gradient algorithm. The procedure is shown in Table II.

Thus, the inverse of the $\mathbf{R}$ matrix is reduced to performing matrix-vector multiplication in the recursive structure. The Kalman gain is not computed explicitly. Note that the vector $\chi(k) = \mathbf{\Omega}^H(k)\mathbf{\Phi}(k)$ can also be partitioned into the $\chi(k) = [\cdots, \chi_t(k)^T, \cdots]^T$. Using the displacement structure for the filtered state estimate discussed in section III, the element vector $\chi_t(k)$ can still be partitioned into the upper, center and lower portion as $\chi_t(k) = [\chi_t^U(k)\ \chi_t^C(k)\ \chi_t^L(k)]$, where

$$\chi_t^U(k) = \mathbf{\Omega}^U(k)^H \mathbf{\Phi}(k); \tag{23}$$

$$\chi_t^C(k) = \mathbf{\Omega}^C(k)^H \mathbf{\Phi}(k); \tag{24}$$

$$\chi_t^L(k) = \mathbf{\Omega}^L(k)^H \mathbf{\Phi}(k); \tag{25}$$

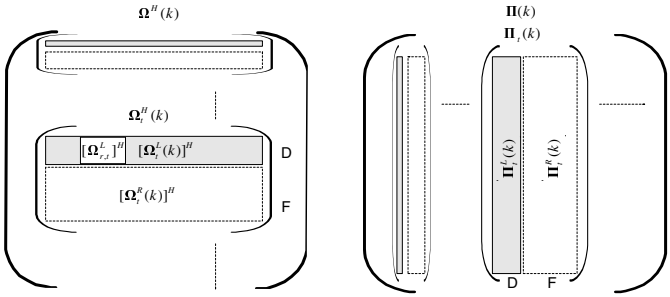

Fig. 3. Effective data in the Omega matrix.

B. Update of Predicted State Error Correlation

Another computation involving the Kalman gain and the inverse of the correlation matrix of the innovation is the update of the predicted state error correlation $\mathbf{P}(k|k)$. With the definition of $\mathbf{\Pi}(k) = \mathbf{R}^{-1}(k)\mathbf{\Omega}(k)$, the CG procedure will need to be applied to the column vectors of $\mathbf{\Pi}(k)$ and $\mathbf{\Omega}(k)$. It can be shown that $\mathbf{\Psi}(k) = \mathbf{\Omega}^H(k)\mathbf{\Pi}(k)$ can also be partitioned into sub-block matrices for the MIMO configuration. The element is given by $\mathbf{\Psi}_{t_1,t_2}(k) = \sum_{r=1}^{N} [\mathbf{\Omega}_{r,t_1}(k)]^H \mathbf{\Pi}_{r,t_2}(k)$ where the $\mathbf{\Omega}_{r,t_1}(k)$ is the element of the omega matrix and $\mathbf{\Pi}(k)$ is partitioned to $\mathbf{\Pi}_{r,t_2}(k) = \{\mathbf{\Pi}_{r,t}(k)\}$. Since only the left upper corner in $\Phi_{t_1,t_2}(k)$ is of interest, the full matrix of $\mathbf{\Pi}(k)$ is not necessary and the whole matrix multiplication by $\mathbf{\Omega}^H(k)$ is redundant. Thus, if the $\mathbf{\Pi}(k)$ is defined by column sub-matrices as $\mathbf{\Pi}(k)$=$[\mathbf{\Pi}_1(k) \cdots \mathbf{\Pi}_t(k) \cdots \mathbf{\Pi}_M(k)]$, and each $\mathbf{\Pi}_t(k)$ is further partitioned into the left portion and right portion as $\mathbf{\Pi}_t(k) = [\mathbf{\Pi}_t^L(k)\ \mathbf{\Pi}_t^R(k)]$, we only need to calculate the left portion from the CG iterative algorithm. Because the iterative algorithm finally reduces to matrix-vector multiplications in a loop, the columns of interest can be easily identified and picked up by simply ignoring the right portions. The effective data for both the $\mathbf{\Omega}^H(k)$ and $\mathbf{\Pi}(k)$ are shown in Fig. 3 as the shaded portion. The iterative procedure to compute the matrix inverse and multiplication $\mathbf{\Pi}(k) = \mathbf{R}^{-1}(k)\mathbf{\Omega}(k)$ is only necessary for the effective data. The detailed procedure is similar to Table II and omitted here. Thus, the direct-matrix inverse is of $\mathbf{R}$ is avoided and the "inverse + multiplication" is reduced to a small portion of the matrix-vector multiplications in an iteration loop.

VI. PERFORMANCE & COMPLEXITY

A. Performance

The performance is evaluated in a CDMA2000 1X EV-DV simulation chain. The focus is to optimize the complexity without sacrificing the performance. Both the Bit-Error-Rate (BER) and the Block-Error-Rate (BLER) are compared from a link level simulation. The ITU Veh-A channel model is applied. Forward error correcting (FEC) code of rate 0.7083 and 0.5156 is applied for the QPSK and 16-QAM respectively. In the simulation, the spreading gain is 32 and U=25 codes are applied for data transmission. Fig. 4 shows the performance for $M = 2, N = 1$ with vehicular speed of $30km/h$. Fig. 5 shows the performance for the 16-QAM for 2×2 MIMO

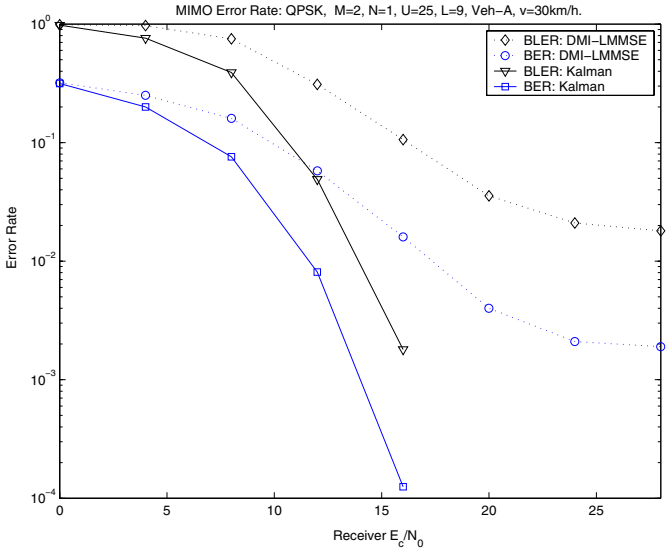

Fig. 4. Performance of the QPSK MIMO link with $M = 2, N = 1, U = 25, L = 9$, Veh-A at speed of $v = 30km/h$.

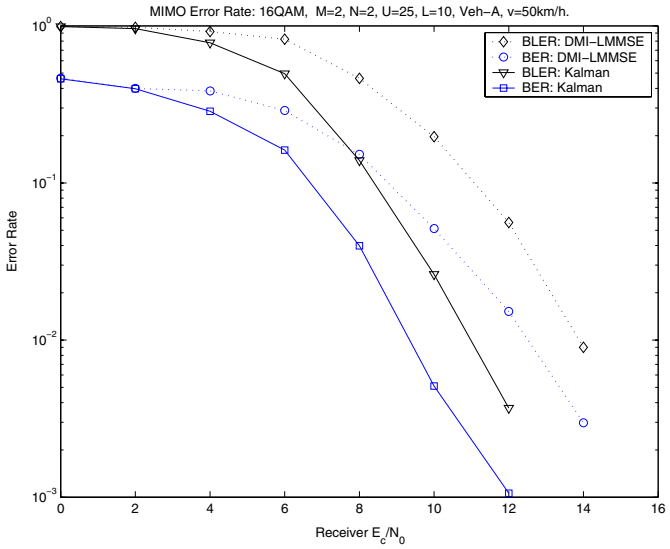

Fig. 5. Performance of the 16-QAM MIMO link with $M = 2, N = 2, U = 25, L = 10$, Veh-A at speed of $v = 50km/h$.

configuration with speed of 50 km/h. The superiority of the Kalman filter over the LMMSE chip equalizer is obvious.

B. Numerical Complexity

In this section, we briefly summarize the complexity reduction achieved from the displacement structure, FFT-based acceleration and the Conjugate Gradient iterative solver for only the effective elements. It is clear that the conventional Kalman procedure has the complexity of $O(F^3)$ for each the matrix-matrix multiplication and the inverse of the Kalman gain. The procedure is applied for symbol duration. Thus, the complexity of the conventional procedure is $O(F^2)$ per chip. After using the displacement structure, many matrix multiplications involving the transition matrix are replaced by simple data loading procedures. Moreover, the FFT acceleration of the matrix multiplication for the channel matrix reduces the complexity to several FFT operations. Finally, the conjugate gradient procedure with only the effective sub blocks avoids the direct matrix inverse and reduces the complexity to some matrix vector multiplications. Overall, the complexity per chip becomes $O(F \log_2 F)$. Moreover, the proposed architecture has more parallel structure, which is suitable for VLSI real-time implementation.

VII. Conclusion

In this paper, efficient VLSI-oriented recursive architecture for MIMO Kalman equalizer is proposed by exploring the block-displacement structure and block Toeplitz structure of the channel matrix to reduce the redundant multiplications in the state transition and Kalman gain dramatically. The proposed architecture not only reduces the numerical complexity to $O(F \log_2 F)$ per chip, but also facilitates the parallel and pipelined real-time VLSI implementation.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] Y. Guo, J. Zhang, D. McCain and J. R. Cavallaro, "Efficient MIMO equalization for downlink multi-code CDMA: complexity optimization and comparative study", *accepted by IEEE GlobeCom 2004*.

[6] M. H. Hayes, "Statistical Digital Signal Processing and Modeling", *John Wiley & Sons*, Inc.: New York, NY, 1996.

[7] K. J. Kim and R. A. Iltis, "Joint detection and channel estimation algorithms for QS-CDMA signals over time-varying channels", *IEEE Trans. on Com.*, vol. 50, pp. 845-55, May 2002.

[8] H. Nguyen, J. Zhang and B. Raghothaman, "Linear Minimum MSE Equalization of SISO and MIMO CDMA Downlink Channels Via The Kalman and FIR Filters", *37th IEEE Asilomar Conference on Signals, Systems and Computers*, 2003.

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