Structured Parallel Architecture for Displacement MIMO Kalman Equalizer in CDMA Systems

Yuanbin Guo, *Member, IEEE*, Jianzhong Zhang, *Member, IEEE*, Dennis McCain, *Member, IEEE*, and Joseph R. Cavallaro, *Senior Member, IEEE*

***Abstract*—A reduced complexity MIMO Kalman equalizer architecture is proposed in this brief by jointly considering the displacement structure and the block-Toeplitz structure. Numerical matrix–matrix multiplications with $\mathcal{O}(F^3)$ complexity are eliminated by simple data loading process, where F is the spreading factor. Finally, an iterative Conjugate-Gradient based algorithm is proposed to avoid the inverse of the Hermitian symmetric innovation covariance matrix in Kalman gain processor. The proposed architecture not only reduces the numerical complexity from $\mathcal{O}(F^2)$ to $\mathcal{O}(F\log_2 F)$ per chip, but also facilitates the parallel and pipelined VLSI implementation in real-time processing.**

***Index Terms*—Displacement, Kalman, multiple-input–multiple-output (MIMO), parallel architecture.**

I. Introduction

MULTIPLE-INPUT–multiple-output (MIMO) technology [1] using multiple antennas at both the transmitter and receiver sides has emerged as a significant breakthrough to increase the spectral efficiency dramatically. Linear minimum-mean-square-error (LMMSE)-based algorithms [2] have demonstrated fairly good performance to suppress the interferences in slow-fading multi-path channels. However, they lack the tracking capability to the time-varying channels in fast-fading environments. Kalman filter [3] based on the state-space model of the dynamical system is known to provide the best linear unbiased estimator (BLUE) of a linear system [4].

However, they apply the conventional Kalman equalizer has very high complexity for real-time hardware implementation, especially for the MIMO scenarios. 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 large size matrix–matrix multiplications and an inverse of the innovation covariance matrix in Kalman gain and new state estimation.

A fast Kalman algorithm is described in [6] to replace the Riccati recursions in the conventional Kalman filters by a different set of fast Chandrasekhar–Kailath–Morf–Sidhu (CKMS) recursions for complexity reduction. The CKMS recursion was proposed for general Kalman filter problem but does not apply the specific structures embedded in the MIMO Kalman equalizer. In this brief, we propose a structured parallel computing architecture for the symbol level MIMO Kalman equalizer proposed in [4] in CDMA systems. We explore the block-displacement structure in the state transition and Kalman gain to reduce the redundant multiplications dramatically. Combining multiple efforts, the proposed architecture not only reduces the numerical complexity from $\mathcal{O}(F^2)$ to $\mathcal{O}(F\log_2 F)$ per chip, but also facilitates the parallel and pipelined VLSI implementation.

Manuscript received September 13, 2005; revised June 15, 2006. Dr. Cavallaro was supported in part by Nokia Corporation, Texas Instruments Incorporated, the Texas Advanced Technology Program under Grant 1999-003604-080, and by the National Science Foundation under Grant ANI-9979465. This paper was recommended by Associate Editor B. C. Levy.

Y. Guo and D. McCain are with Nokia Networks, Irving, TX 75039 USA (e-mail: Yuanbin.Guo@nokia.com).

J. Zhang is with Motorola Inc., Fort Worth, TX 76137 USA.

J. R. Cavallaro is with The Department of Electrical and Computer Engineering and Computer Science, Rice University, Houston, TX 77005 USA.

Digital Object Identifier 10.1109/TCSII.2006.885075

II. MIMO CDMA System and State-Space Model

We consider the same MIMO CDMA downlink as in [2] and [4] based on spatial multiplexing with M Tx antennas, N Rx antennas and U users. The ith chip at the tth transmit antenna is given by $x_t(i)$. The received chip level signal at the rth Rx antenna is given by $y_r(i)$. By collecting the F consecutive chips at the kth 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 is given by

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

where $\mathbf{v}_r(k)$ is the additive Gaussian noise. The transmitted chip vector for the tth 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 tth transmit antenna and rth receive antenna.

The Kalman filter estimates the state $\mathbf{x}(k)$ given the entire observed data $\mathbf{y}(1),\cdots,\mathbf{y}(k)$ using the state-space model. In [4], both symbol level and chip level Kalman filters are proposed for the CDMA downlink. We consider the symbol-level Kalman Equalizer in this brief because of its superior performance over the chip level equalizer. It is natural to have the measurement equation as the received signal model and the process equation as an excitation of some process noise

$$\mathbf{y}(k) = \mathbf{H}(k)\mathbf{x}(k) + \mathbf{v}(k) \tag{2}$$

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 = $\boldsymbol{\Theta}$; Measure matrix = $\mathbf{H}(k)$;

Covariance matrix of the process noise : $\mathbf{Q}_w(k) = E[\mathbf{w}(k)\mathbf{w}^H(k)]$;

Covariance 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 :

$\hat{\mathbf{x}}(k|k-1) = \boldsymbol{\Theta}\ \hat{\mathbf{x}}(k-1|k-1)$;

$\mathbf{P}(k|k-1) = \boldsymbol{\Theta}\ \mathbf{P}(k-1|k-1)\boldsymbol{\Theta}^H + \mathbf{Q}_w(k)$;

(2). Innovation generation :

$\alpha(k) = \mathbf{y}(k) - \mathbf{H}(k)\hat{\mathbf{x}}(k|k-1)$;

$\boldsymbol{\Omega}(k) = \mathbf{H}(k)\mathbf{P}(k|k-1)$;

(3). Kalman gain computation :

$\mathbf{R}(k) = \mathbf{H}(k)\boldsymbol{\Omega}^H(k) + \mathbf{Q}_v(k)$;

$\mathbf{G}(k) = \boldsymbol{\Omega}^H(k)\mathbf{R}^{-1}(k)$;

(4). State estimate & Predicted state error covariance update :

$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{G}(k)\alpha(k)$;

$\mathbf{P}(k|k) = \mathbf{P}(k|k-1) - \mathbf{G}(k)\boldsymbol{\Omega}(k)$.

$$\mathbf{x}(k) = \boldsymbol{\Theta}\mathbf{x}(k-1) + \mathbf{w}(k) \tag{3}$$

where the measure matrix is the overall MIMO channel matrix $\mathbf{H}(k)$ given by $\mathbf{H}(k) = [\mathbf{H}_1(k)^T, \cdots, \mathbf{H}_r(k)^T, \cdots, \mathbf{H}_N(k)^T]^T$. $\boldsymbol{\Theta}$ is a constant state transition matrix whose details are given later. $\mathbf{v}(k)$ denotes the measurement noise and $\mathbf{w}(k)$ denotes the process noise.

III. DISPLACEMENT MIMO KALMAN EQUALIZER

A. Modular-Based Architecture

An innovation process and its covariance matrix of the innovation process are defined by $\alpha(k) = \mathbf{y}(k) - \hat{\mathbf{y}}(k|k-1)$ and $\mathbf{R}(k) = E[\alpha(k)\alpha^H(k)]$. $\hat{\mathbf{y}}(k|k-1)$ 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 covariance matrix of the predicted state error 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 is given by

$$\hat{\mathbf{x}}(k|k) = \boldsymbol{\Theta}\mathbf{x}(k-1|k-1) + \mathbf{G}(k)\alpha(k). \tag{4}$$

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

The logic block diagram of the VLSI oriented architecture is shown in Fig. 1. The architecture is constructed with two parallel feedback loop structures that are associated with a common Kalman gain $\mathbf{G}(k)$. On the top is the one step predictor of the state $\hat{\mathbf{x}}(k|k)$ using the input observation $\mathbf{y}(k)$. A MUX first selects either the init state or the delayed feedback state estimate for $\hat{\mathbf{x}}(k-1|k-1)$, where $z^{-1}\mathbf{I}$ in the figure denotes the delay of the vector or matrix. The $\hat{\mathbf{x}}(k-1|k-1)$ is first pre-multiplied (PRM) by the transition matrix to generate $\hat{\mathbf{x}}(k|k-1)$

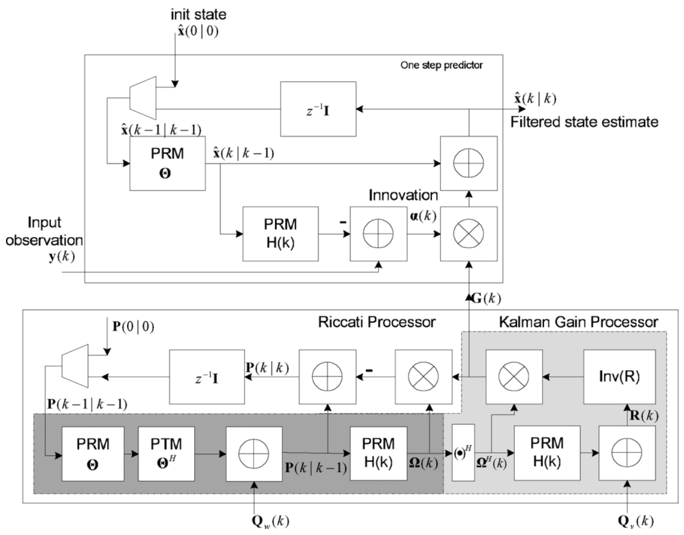

Fig. 1. Modularized data path block diagram of the Kalman filter.

and then PRM by the channel matrix $\mathbf{H}(k)$. The result is subtracted from the input observation $\mathbf{y}(k)$ to generate the innovation process $\alpha(k)$. The innovation is then multiplied by the Kalman gain $\mathbf{G}(k)$ and added to the $\hat{\mathbf{x}}(k|k-1)$ to finally generate the filtered state estimate $\hat{\mathbf{x}}(k|k)$. The dynamical transition is repeated for each time k to get state sequence estimate.

On the bottom is the feedback loop of the covariance $\mathbf{P}(k|k)$ and the Kalman gain processor. Similarly, a MUX first selects from the init value $\mathbf{P}(0|0)$ or the delayed feed $\mathbf{P}(k|k)$ for the $\mathbf{P}(k-1|k-1)$. It is PRM by $\boldsymbol{\Theta}$ and then post-multiplied (PTM) by $\boldsymbol{\Theta}^H$. The covariance of the process noise $\mathbf{Q}_w(k)$ is then added to form an intermediate covariance $\mathbf{P}(k|k-1)$. This is PRM by $\mathbf{H}(k)$ to generate the $\boldsymbol{\Omega}(k)$, whose result is then Hermitian transposed. Note that the Hermitian transpose is a virtual operation with no time/memory resource usage because the subsequential operations can use the structure of $\boldsymbol{\Omega}^H(k)$ explicitly. $\boldsymbol{\Omega}^H(k)$ is PRM by $\mathbf{H}(k)$ and the result is added to the measurement noise covariance matrix $\mathbf{Q}_v(k)$ to form the innovation covariance $\mathbf{R}(k)$. The Kalman gain is produced as the result of the pre-multiplication of $\boldsymbol{\Omega}^H(k)$ with the inverse of $\mathbf{R}(k)$. The $\mathbf{P}(k|k)$ is then updated in the Riccati processor accordingly.

B. MIMO Displacement Structure

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 follows:

$$\boldsymbol{\Theta} = \mathbf{I}_M \otimes \tilde{\boldsymbol{\Theta}} \tag{5}$$

$$\tilde{\boldsymbol{\Theta}} = \begin{pmatrix} \mathbf{0}_{F\times D} & \mathbf{0}_{F\times F} \\ \mathbf{I}_{D\times D} & \mathbf{0}_{D\times F} \end{pmatrix} \tag{6}$$

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 tth 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 $\tilde{\boldsymbol{\Theta}}$ 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 $\tilde{\mathbf{\Theta}}^H$ 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 step by step according to Table I.

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 tth sub block of the transition is given by $\hat{\mathbf{x}}_t(k|k-1) = \tilde{\mathbf{\Theta}}\hat{\mathbf{x}}_t(k-1|k-1) = [\mathbf{0}_{1\times F}\hat{\mathbf{x}}_t^U(k-1|k-1)^T]^T$, 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.

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 $\hat{\mathbf{x}}(k|k) = [\cdots\ \hat{\mathbf{x}}_t(k|k)^T\ \cdots]^T$ and $\mathbf{G}(k) = [\cdots\ \mathbf{G}_t(k)^T\ \cdots]^T$. 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 remaining 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 $(k-1)$. 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.

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

$$\mathbf{Q}_w(k) = E\left\{\mathbf{w}(k)\mathbf{w}(k)^H\right\} = \mathbf{I}_M \otimes \mathbf{Q}(k) \tag{7}$$

$$\mathbf{Q}(k) = \begin{pmatrix} \tilde{\mathbf{Q}}_w(k) & \mathbf{0}_{F\times D} \\ \mathbf{0}_{D\times F} & \mathbf{0}_{D\times D} \end{pmatrix}. \tag{8}$$

It is also pointed out in [4] that the covariance matrix $\tilde{\mathbf{Q}}_w(k)$ can be estimated as a unitary matrix determined by the total transmitted power when *a priori* is unknown because $\mathbf{w}(k)$ is determined by the transmitted symbols. Thus, if we span the MIMO covariance 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 $\mathbf{P}_{t_1,t_2}(k|k-1) = \tilde{\mathbf{\Theta}}\mathbf{P}_{t_1,t_2}(k-1|k-1)\tilde{\mathbf{\Theta}}^H + \mathbf{Q}_{t_1,t_2}(k)$ and $\mathbf{Q}_{t_1,t_2}(k) = \mathbf{Q}(k)\delta(t_1-t_2)$.

Using the feature of the pre-multiplication and post-multiplication with the displacement transition matrix, we can show that the new covariance matrix is given by the following partitioning:

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

where $(a:b)$ in one dimension of the $\mathbf{P}$ matrix denotes the ath to bth elements in this dimension. The sub block matrix

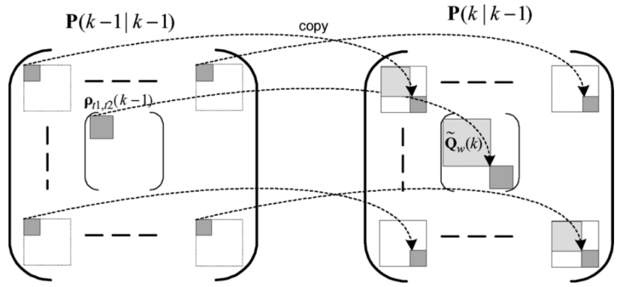

Fig. 2. Displacement-based architecture for $\mathbf{P}(k|k-1)$.

$\rho_{t_1,t_2}(k-1)$ is a $D\times D$ left-upper corner of the partitioned covariance 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. This displacement procedure is demonstrated by the data loading process in Fig. 2.

4) Update State Error Covariance 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. 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 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 $m \in [1,M]$. Instead of computing the full matrix of $\mathbf{P}(k|k)$, we only need to compute the relevant submatrices given by $\boldsymbol{\rho}_{t_1,t_2}(k) = \mathbf{P}_{t_1,t_2}(k|k-1, 0:D-1, 0:D-1) + \mathbf{\Psi}_{t_1,t_2}(k, 0:D-1, 0:D-1)$.

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 $\mathbf{G}_{t,r}(k) = [\mathbf{G}_{t,r}^U(k)^T\ \ \mathbf{G}_{t,r}^L(k)^T]^T$ is further partitioned into the upper and lower sub-matrices while $\mathbf{\Omega}_{r,t}(k) = [\mathbf{\Omega}_{r,t}^L(k)\ \ \mathbf{\Omega}_{r,t}^R(k)]$ is partitioned into the left and right sub-matrices. It is clear that the element block in the $\mathbf{\Psi}(k)$ is given by

$$\mathbf{\Psi}_{t_1,t_2}(k) = \sum_{r=1}^{N} \mathbf{G}_{t_1,r}(k)\mathbf{\Omega}_{r,t_2}(k) \tag{9}$$

Comparing the displacement structure, only the left-upper corner of size $D\times D$ is necessary, which is given by $\widetilde{\mathbf{\Psi}}_{t_1,t_2}(k) = \mathbf{\Psi}_{t_1,t_2}(k, 0:D-1, 0:D-1) = \sum_{r=1}^{N}\mathbf{G}_{t_1,r}^U(k)\mathbf{\Omega}_{r,t_2}^L(k)$. 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 covariance 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 $\alpha = \mathbf{y}(k) - \mathbf{H}(k)\hat{\mathbf{x}}(k|k-1)$ 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 $\hat{\mathbf{y}}(k) = \mathbf{H}(k)\hat{\mathbf{x}}(k|k-1) = [(\sum_{t=1}^{M}\mathbf{H}_{t,1}(k)\hat{\mathbf{x}}_t(k|k-1))^T, \cdots, (\sum_{t=1}^{M}\mathbf{H}_{t,N}(k)\hat{\mathbf{x}}_t(k|k-1))^T]^T$. Since the channel matrix from the tth transmit antenna and rth receive antenna $\mathbf{H}_{t,r}(n)$ has the Toeplitz structure as shown in [4], the matrix–vector multiplication can be viewed as an FIR filter with the channel impulse response $[h_{t,r}^D, \cdots, h_{t,r}^0]$, where $h_{t,r}^l$ is the lth path channel coefficient between the tth transmit and rth receive antenna. This can be implemented in the time domain by tapped delay 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 brief. This achieves $\mathcal{O}((F+D)\log_2(F+D))$ complexity algorithm versus $\mathcal{O}((F+D)^2)$ for the direct matrix–vector multiplication and $\mathcal{O}((F+D)^2\log_2(F+D))$ versus $\mathcal{O}((F+D)^3)$ for the direct matrix–matrix multiplications in the innovation estimation and the Kalman gain processor. The procedure is described briefly as follows.

1) Take the FFT of the zero-padded channel impulse response $[h_{t,r}^D, \cdots, h_{t,r}^0, 0, \cdots, 0]$.
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.

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. The computation of the covariance matrix of innovation as $\mathbf{R}(k) = \mathbf{H}(k)\mathbf{\Omega}^H(k) + \mathbf{Q}_v(k)$ is also accelerated by the FFT-based computing architecture with similar procedure.

V. Iterative Inverse Solver

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

$$\mathbf{G}(k) = \mathbf{\Omega}^H\mathbf{R}^{-1}(k). \tag{10}$$

It is known that a Gaussian elimination can be applied to solve the matrix inverse with complexity at the order of $\mathcal{O}[(NF)^3]$. Moreover, Cholesky decomposition can also be applied to accelerate the speed by reducing the hidden constant factor in 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 $\mathbf{R}$ is a $(NF \times NF)$ Hermitian symmetric matrix. It is known that the iterative Conjugate Gradient (CG) algorithm can solve the inverse of this type of matrix more efficiently [5]. Secondly, the full matrix of $\mathbf{G}$ 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.

TABLE II
Summary of the CG Procedure for the $\mathbf{\Phi}(k) = \mathbf{R}^{-1}\alpha(k)$

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

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:

$$\chi(k) = \mathbf{G}(k)\alpha(k) = \mathbf{\Omega}^H(k)\left[\mathbf{R}^{-1}(k)\alpha(k)\right] = \mathbf{\Omega}^H(k)\mathbf{\Phi}(k)$$
$$\mathbf{\Psi}(k) = \mathbf{G}(k)\mathbf{\Omega}(k) = \mathbf{\Omega}^H(k)\left[\mathbf{R}^{-1}(k)\mathbf{\Omega}(k)\right] = \mathbf{\Omega}^H(k)\mathbf{\Pi}(k)$$

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 follows.

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, where J denotes the number of CG iteration.

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 $\chi_t^U(k) = \mathbf{\Omega}^U(k)^H\mathbf{\Phi}(k)$; $\chi_t^C(k) = \mathbf{\Omega}^C(k)^H\mathbf{\Phi}(k)$ and $\chi_t^L(k) = \mathbf{\Omega}^L(k)^H\mathbf{\Phi}(k)$.

B. *Update of Predicted State Error Covariance*

Another computation involving the Kalman gain and the inverse of the covariance matrix of the innovation is the update of the predicted state error covariance $\mathbf{P}(k|k)$. 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 $\mathbf{\Omega}$ 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 $\mathbf{\Pi}(k)$ 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

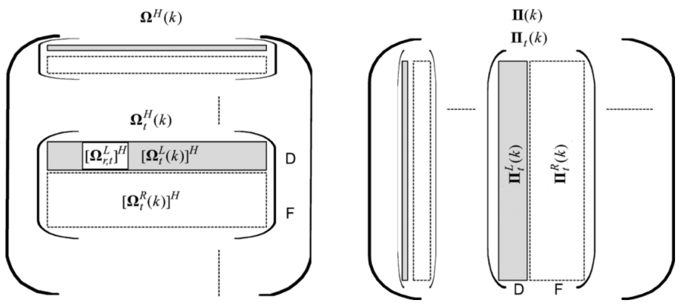

Fig. 3. Effective data for matrix multiplication and conjugate gradient matrix inversion in $\mathbf{\Omega}(k)$ and $\mathbf{\Pi}(k)$.

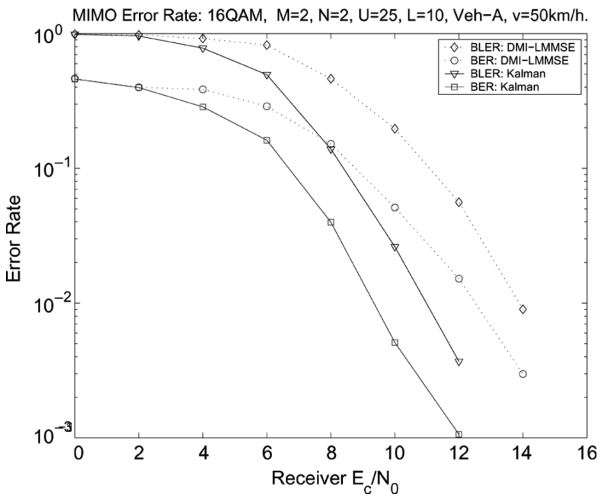

Fig. 4. Performance of the 16-QAM MIMO link with $M = 2, N = 2, U = 25, L = 10$, Veh-A at speed of $v = 50$ km/h, $J = 4$.

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 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. 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 0.5156 is applied. In the simulation, the spreading gain is 32 and U = 25 codes are applied for data transmission. Fig. 4 shows the performance for the 16 quadrature amplitude modulation (QAM) for the 2×2 MIMO configuration with a speed of 50 km/h with four CG iterations. The performance 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, the FFT-based

TABLE III
Complexity Comparison Before/After Complexity Reduction: The Complexity in Each Step is for a Symbol Duration

Operation	DMM/DMI Complexity	Reduced complexity
$\mathbf{\Theta P}(k\|k)$	$\mathcal{O}\{[M(F+D)]^3\}$	-
$\mathbf{H}(k)\mathbf{P}(k\|k-1)$	$\mathcal{O}\{NF[M(F+D)]^2\}$	$\mathcal{O}\{N[MF]^2 log_2 F\}$
$\mathbf{\Omega}^H(k)\mathbf{R}^{-1}(k)$	$\mathcal{O}\{[M(F+D)+NF](NF)^2\}$	$\mathcal{O}\{MDJ[NF]^2\}$
$\mathbf{G}(k)\mathbf{\Omega}(k)$ or $\mathbf{\Omega}^H(k)\mathbf{\Pi}(k)$	$\mathcal{O}\{[M(F+D)]^2 NF\}$	$\mathcal{O}\{[MD]^2 NF\}$
Overall/chip	$\mathcal{O}\{(F+D)^2\}$	$\mathcal{O}\{F log_2 F\}$

acceleration and the effective Conjugate Gradient iterative solver. The complexity using the conventional algorithms and using the proposed fast algorithms are compared in Table III for the dominant operations in each step. It is clear that the original Kalman procedure has the complexity of $\mathcal{O}(F^3)$ for each the matrix–matrix multiplication and the inversion of the Kalman gain. Since these operations are for a symbol duration, the complexity of the original procedure is $\mathcal{O}(F^2)$ 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 with complexity of $\mathcal{O}(F \log_2 F)$. 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 is reduced to $\mathcal{O}(F \log_2 F)$. Moreover, the proposed architecture has more parallel structure, which is more suitable for VLSI real-time implementation.

VII. Conclusion

In this brief, 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 $\mathcal{O}(F \log_2 F)$ 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] Y. Guo, J. Zhang, D. McCain, and J. R. Cavallaro, "Efficient MIMO equalization for downlink multi-code CDMA: complexity optimization and comparative study," in *Proc. IEEE GlobeCom 2004*, Dallas, TX, Nov. 2004, vol. 4, no. 7, pp. 2513–2519.

[3] M. H. Hayes, *Statistical Digital Signal Processing and Modeling*. New York: Wiley, 1996.

[4] ——, "A Kalman-filter approach to equalization of CDMA downlink channels," *EURASIP J. Appl. Signal Process.*, vol. 2005, pp. 611–625, Apr. 2005.

[5] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *J. Res. Nat. Bureau Standards*, vol. 49, pp. 409–436, Dec. 1952.

[6] T. Kailath, A. H. sayed, and B. Hassibi, *Linear Estimation*. Upper Saddle River, NJ: Prentice-Hall, 2000.

[7] X. Wang and P. H. Vincent, "Blind joint equalization and multiuser detection for DS-CDMA in unknown correlated noise," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 46, no. 7, pp. 886–895, Jul. 1999.