Adaptive Conjugate Gradient DFEs for Wideband MIMO Systems using Galerkin Projections

Vassilis Kekatos, Aris S. Lalos, Kostas Berberidis
Dept. of Computer Engineering & Informatics,
University of Patras
26500, Rio-Patras, Greece
Emails:{kekatos,lalos,berberid}@ceid.upatras.gr

***Abstract*— Three new adaptive equalization algorithms for wireless systems operating over frequency selective MIMO channels are proposed. The problem of the MIMO DFE design is formulated as a set of linear equations with multiple right-hand sides (RHS) evolving in time. By applying an adaptive modified conjugate gradient algorithm, originally proposed for a single linear system, to the problem at hand, we arrive at an equalizer of performance identical to RLS, numerically robust, but of higher computational cost. To reduce its complexity, two updating strategies of the equalizer filters are derived based on Galerkin projection in time and space respectively. The two alternative schemes exhibit a complexity lower than RLS while offering slightly inferior convergence properties.**

I. Introduction

Equalization of wireless MIMO frequency selective channel is a challenging task mainly due to the fact that the respective MIMO equalizers should cope with intersymbol, as well as interstream interference. When the channel is static and has already been estimated by the receiver, a MIMO DFE can be designed according to methods such as those in [1].

However, when the channel impulse response changes within a burst (a case arising in relatively long bursts and/or fast varying conditions), the above techniques fail to equalize the channel and efficient adaptive methods are required. To our knowledge, the only adaptive MIMO DFE designs are those proposed in [2] and [3], where the respective equalizers are updated using the recursive least squares (RLS) algorithm. The main problems in adaptive MIMO equalization are the increased size of the equalizer and the colored noise appearing due to interstream interference.

In the SISO case, adaptive algorithms based on the conjugate gradient (CG) method have already been proposed in [4], [5], [6]. These algorithms are numerically stable and exhibit convergence properties comparable to RLS in an equal or lower computational cost. In this work, first, we extend the adaptive modified CG (MCG) proposed in [4] to the MIMO case. Although the performance of this algorithm (MIMO-MCG) is identical to that of the MIMO RLS algorithm of [2], its computational complexity is increased. To reduce the complexity, an approximation of the MCG update has been derived based on the idea of Galerkin projections [7]. By exploiting the structure of the MIMO DFE problem, projections can be performed either in time, or in space. Hence, two alternative schemes are derived having convergence properties close to MIMO-MCG at a computational cost lower than MIMO RLS. Moreover, these schemes provide a flexible framework in MIMO adaptive equalization design to trade efficiency for performance.

This work is part of the 03ED838 research project, implemented within the framework of the "Reinforcement Programme of Human Research Manpower" (PENED) and co-financed by national and E.C. Funds.

II. System Model and Problem Formulation

Let us consider a MIMO communication system operating over a frequency selective wireless channel. The system employs M transmit and N receive antennas, with $M\leq N$, while spatial multiplexing is assumed. The signal transmitted across the M antennas at time k can be described by the vector

$$\mathbf{s}(k) = \begin{bmatrix} s_1(k) & \cdots & s_M(k) \end{bmatrix}^T \tag{1}$$

where $s_i(k),\ i = 1,\ldots,M$, are i.i.d. symbols of unit variance. By employing a discrete-time complex baseband model, the signal received at the n-th antenna can be expressed as

$$x_n(k) = \frac{1}{\sqrt{M}} \sum_{i=1}^{M} \sum_{l=0}^{L} h_{ni}(l) s_i(k-l) + \eta_n(k) \tag{2}$$

where $h_{ni}(l)$ for $l = 0,\ldots,L$, is the sampled impulse response between transmitter i and receiver n, $(L+1)$ is the channel length, and $\eta_n(k),\ n = 1,\ldots,N$, are white Gaussian complex noise samples of variance $N_0/2$ per dimension. The samples received at time k can be assembled in the vector

$$\underline{\mathbf{x}}(k) = \begin{bmatrix} x_1(k) & \cdots & x_N(k) \end{bmatrix}^T. \tag{3}$$

The intersymbol and interstream interference involved in the system described by (2) can be mitigated through a MIMO Decision Feedback Equalizer (DFE) [2]. The proposed equalizer architecture is a structure of M MISO DFEs operating in parallel. The i-th MISO DFE is designated to extract the i-th stream $s_i(k)$, and it consists of a feedforward and a feedback filter of temporal span K_f and K_b taps, respectively. The input of the feedforward filters $\mathbf{f}_i(k)$, for $i = 1,\ldots,M$, can be described by the $NK_f \times 1$ vector

$$\mathbf{x}(k) = \begin{bmatrix} \underline{\mathbf{x}}^T(k-K_f+1) & \cdots & \underline{\mathbf{x}}^T(k) \end{bmatrix}^T. \tag{4}$$

Similarly, if $\tilde{d}_i(k)$ denotes the output of the i-th DFE, and $d_i(k) = f\{\tilde{d}_i(k)\}$ is the corresponding decision device output, then the input of the feedback filters, $\mathbf{b}_i(k)$ for $i = 1, \ldots, M$, can be expressed by the $MK_b \times 1$ vector

$$\mathbf{d}(k) = \left[\begin{array}{ccc} \underline{\mathbf{d}}^T(k-K_b) & \cdots & \underline{\mathbf{d}}^T(k-1) \end{array}\right]^T \tag{5}$$

where the $\underline{\mathbf{d}}(k)$ is defined as

$$\underline{\mathbf{d}}(k) = \left[\begin{array}{ccc} d_1(k) & \ldots & d_M(k) \end{array}\right]^T.$$

By using the above definitions, the output of the i-th DFE can be compactly expressed as

$$\begin{aligned} \tilde{d}_i(k) &= \mathbf{w}_i^H(k)\mathbf{y}(k), \qquad (6)\\ \mathbf{w}_i(k) &= \left[\begin{array}{cc} \mathbf{f}_i^T(k) & \mathbf{b}_i^T(k) \end{array}\right]^T, \\ \mathbf{y}(k) &= \left[\begin{array}{cc} \mathbf{x}^T(k) & \mathbf{d}^T(k) \end{array}\right]^T, \; i = 1, \ldots, M. \end{aligned}$$

Notice that all the MISO DFEs have a common input, $\mathbf{y}(k)$, of dimension $K = NK_f + MK_b$.

The MIMO DFE may be found by using a least squares (LS) approach. Provided that all previous decisions are correct, each equalizer $\mathbf{w}_i(k)$ can be computed as the minimizing argument of the cost function

$$J\left(\mathbf{w}, \mathbf{\Phi}(k), \mathbf{z}_i(k)\right) = \frac{\mathbf{w}^H \mathbf{\Phi}(k)\mathbf{w}}{2} - Re\{\mathbf{w}^H \mathbf{z}_i(k)\} \tag{7}$$

with respect to $\mathbf{w}$. Matrix $\mathbf{\Phi}(k)$ stands for the $K \times K$ exponentially time-averaged input data autocorrelation matrix, and $\mathbf{z}_i(k)$ for the crosscorrelation vector, which are defined as

$$\begin{aligned} \mathbf{\Phi}(k) &= \lambda\mathbf{\Phi}(k-1) + \mathbf{y}(k)\mathbf{y}^H(k), \qquad (8)\\ \mathbf{z}_i(k) &= \lambda\mathbf{z}_i(k-1) + \mathbf{y}(k)d_i^*(k) \qquad (9) \end{aligned}$$

where λ is a forgetting factor $(0 < \lambda \leq 1)$.

Equivalently, the minimizers of (7) can be derived as the solution of a set of linear equations with multiple RHS, i.e.,

$$\mathbf{\Phi}(k)\mathbf{W}(k) = \mathbf{Z}(k) \tag{10}$$

where the $K \times M$ matrices $\mathbf{W}(k)$ and $\mathbf{Z}(k)$ are defined as

$$\begin{aligned} \mathbf{W}(k) &= \left[\begin{array}{ccc} \mathbf{w}_1(k) & \cdots & \mathbf{w}_M(k) \end{array}\right], \\ \mathbf{Z}(k) &= \left[\begin{array}{ccc} \mathbf{z}_1(k) & \cdots & \mathbf{z}_M(k) \end{array}\right]. \end{aligned}$$

III. Conjugate Gradient and Galerkin Projections

Before presenting the proposed CG-based algorithms, it is necessary to make a brief introduction into the use of CG methods for solving single and multiple linear systems.

A. Solution of a Single Linear System

The conjugate gradient (CG) method is an iterative Krylov subspace method for solving linear systems of the form

$$\mathbf{\Phi}\mathbf{w} = \mathbf{z}. \tag{11}$$

where $\mathbf{\Phi}$ is a $K \times K$ Hermitian positive definite matrix [8]. The CG method minimizes the quadratic function $J(\mathbf{w}, \mathbf{\Phi}, \mathbf{z})$ defined in (7) by iteratively updating the parameters' vector as

$$\mathbf{w}(k) = \mathbf{w}(k-1) + \alpha(k)\mathbf{p}(k). \tag{12}$$

The search direction vectors $\mathbf{p}(k)$ for $k \geq 1$ are designed to be $\mathbf{\Phi}$-orthogonal to each other, i.e. $\mathbf{p}^H(k)\mathbf{\Phi}\mathbf{p}(l) = 0$ for $k \neq l$. Moreover, the step sizes, $\alpha(k)$, are selected as the minimizing arguments of $J(\mathbf{w}(k), \mathbf{\Phi}, \mathbf{z})$ with respect to $\alpha(k)$.

To obtain the $\mathbf{\Phi}$-orthogonal direction vectors, the Gram-Schmidt conjugation process should be applied on a set of orthogonal vectors. The CG method selects the successive negative gradients (or residuals) of the cost function, i.e.,

$$\mathbf{g}(k) = -\frac{\partial J(\mathbf{w}(k), \mathbf{\Phi}, \mathbf{z})}{\partial \mathbf{w}^H(k)} = \mathbf{z} - \mathbf{\Phi}\mathbf{w}(k), \tag{13}$$

to be the basis vectors. By using the properties of the gradients, the direction vectors can be updated as

$$\mathbf{p}(k+1) = \mathbf{g}(k) + \beta(k)\mathbf{p}(k), \tag{14}$$

and $\beta(k)$ are chosen to ensure the $\mathbf{\Phi}$-orthogonality among the direction vectors [8]. Finally, it can be shown that the CG method converges in at most K iterations [8].

B. Solution of Multiple Linear Systems

Conjugate gradient optimization methods may also be employed for solving multiple linear systems of the form $\mathbf{\Phi}_i\mathbf{w}_i = \mathbf{z}_i$ for $i = 1, \ldots, M$. A straightforward approach is to treat each system independently and apply to it the CG method presented in subsection III-A. However, more sophisticated methods have been proposed in the literature, and can be organized in two groups: the block-Krylov subspace solvers [9], and those using projections [7].

Block-Krylov subspace solvers apply only in the case of a linear system with multiple RHS, i.e. $\mathbf{\Phi}_i = \mathbf{\Phi}$ for all i. They generalize the idea of the CG method and enforce $\mathbf{\Phi}$-orthogonality of direction vectors not only across a single system, but among the other systems as well, that is $\mathbf{p}_i^H(k)\mathbf{\Phi}\mathbf{p}_j(l) = 0$ for all systems i, j, and iterations k, l. Hence, by adding more constraints in the optimization problem, these algorithms achieve faster convergence at the expense of increased complexity, since two matrix inversions are required per iteration.

In this work, our main goal is to derive a computationally efficient algorithm for updating the MIMO DFE filters. At the same time, note that the order of the linear systems in (10), K, is much smaller than the time needed for the statistics of the problem (8)-(9) to converge to their steady state values. Thus, convergence in fewer than K iterations is not an issue in the MIMO DFE case. For these two reasons, we choose not to follow a block-Krylov subspace approach, but emphasis has been given on the projection methods which are presented briefly below.

According to the projection methods, one of the linear systems is selected as the 'seed' system. Let the j-th one, $\mathbf{\Phi}_j\mathbf{w}_j = \mathbf{z}_j$, be the seed system. This system is solved by using the conventional CG method of subsection III-A until it converges to its solution. During these iterations, the rest of the systems do not perform any CG iteration, but rather they use the search direction of the seed system, $\mathbf{p}_j(k)$, to update their solution as

$$\mathbf{w}_i(k) = \mathbf{w}_i(k-1) + \alpha_i(k)\mathbf{p}_j(k), \; i \neq j \tag{15}$$

where the step size $\alpha_i(k)$ is again selected as the minimizing argument of $J(\mathbf{w}_i(k), \mathbf{\Phi}_i, \mathbf{z}_i)$,

$$\alpha_i(k) = \frac{\mathbf{p}_j^H(k)\mathbf{g}_i(k-1)}{\mathbf{p}_j^H(k)\mathbf{\Phi}_i\mathbf{p}_j(k)}, \qquad (16)$$

and the gradient is defined according to (13) as $\mathbf{g}_i(k) = \mathbf{z}_i - \mathbf{\Phi}_i\mathbf{w}_i(k)$. Upon convergence of the seed system, a new system is selected as the seed system and the whole procedure is repeated until all systems have been solved.

By using the update of (15), it can be shown [7] that the solution of a non-seed system at the k-th iteration, $\mathbf{w}_i(k)$ is restricted to lie in the subspace where the respective seed system's solution, $\mathbf{w}_j(k)$, lies. The solution of a system is eventually refined when it is, in turn, selected as the seed system. This method, known as the Galerkin projection method, has been utilized to reduce the computational cost of multiple linear system solvers by exploiting possible relations among the linear system parameters $\mathbf{\Phi}_i$ and $\mathbf{z}_i$ [7].

IV. Algorithm Derivation

When the system parameters are known a-priori and kept static, then the CG methods of the previous section can be employed. However, the solution of MIMO DFE in (10) can be adaptively updated and track any variations in the input data autocorrelation matrix and the corresponding crosscorrelation vectors. Adaptive CG algorithms for single-input single-output (SISO) systems have already been proposed in [4], [5], [6], where either a single, or multiple CG iterations are performed per sample time k. In [4], the former approach has been followed, and by properly modifying the original CG method an adaptive algorithm (modified CG, MCG) of performance comparable to RLS was derived. In this section, we extend this algorithm to the MIMO case, and apply the idea of Galerkin projections to reduce its complexity.

A. Adaptive MIMO Modified Conjugate Gradient

By generalizing the adaptive MCG algorithm of [4] to the MIMO case, the solution of (10) can be time updated as

$$\mathbf{W}(k) = \mathbf{W}(k-1) + \mathbf{P}(k)\mathbf{A}(k) \qquad (17)$$

where the columns of $\mathbf{P}(k)$ are the search directions for each of the M systems, and $\mathbf{A}(k)$ is a $M \times M$ diagonal matrix having as i-th diagonal element, $\alpha_i(k)$, the step size of the corresponding system. If $\mathbf{e}(k) = \mathbf{s}(k) - \mathbf{W}^H(k-1)\mathbf{y}(k)$ is the a-priori estimation error, then the gradients of the systems can be derived by (17), (8), and (9) as

$$\begin{aligned}\mathbf{G}(k) &= \mathbf{Z}(k) - \mathbf{\Phi}(k)\mathbf{W}(k) \\ &= \mathbf{T}(k) - \mathbf{\Phi}(k)\mathbf{P}(k)\mathbf{A}(k) \qquad (18)\end{aligned}$$

where $\mathbf{T}(k)$ is defined as

$$\mathbf{T}(k) = \lambda\mathbf{G}(k-1) + \mathbf{y}(k)\mathbf{e}^H(k). \qquad (19)$$

The step sizes $\alpha_i(k)$ are selected as the minimizing arguments of $J\left(\mathbf{w}_i(k), \mathbf{\Phi}(k), \mathbf{z}_i(k)\right)$, i.e.,

$$\alpha_i(k) = \frac{\mathbf{p}_i^H(k)\mathbf{t}_i(k)}{\mathbf{p}_i^H(k)\mathbf{\Phi}(k)\mathbf{p}_i(k)}, \; i = 1, \ldots, M, \qquad (20)$$

TABLE I

Summary of MIMO-MCG

Initialization: $\mathbf{G}(0) = \mathbf{W}(0) = \mathbf{0}_{K\times M}$, $\mathbf{B}(0) = \mathbf{0}_M$ $\mathbf{P}(1) = \mathbf{y}(1)\mathbf{s}^H(1)$, and $\mathbf{\Phi}(0) = \delta\mathbf{I}_K$ where δ is a small positive constant.
1) Update matrix $\mathbf{\Phi}(k)$ (8).
2) Update matrix $\mathbf{T}(k)$ (19).
3) Compute the step sizes by using (20).
4) Update the equalizer filters from (17).
5) Update the gradients (18).
6) Compute the $\beta_i(k)$ for $i = 1, \ldots, M$, from (22).
7) Update the search directions as in (21).

and $\mathbf{t}_i(k)$ is the i-th column of $\mathbf{T}(k)$. Then, the search directions for the next update are computed as

$$\mathbf{P}(k+1) = \mathbf{G}(k) + \mathbf{P}(k)\mathbf{B}(k) \qquad (21)$$

where $\mathbf{B}(k)$ is again a $M \times M$ diagonal matrix. By employing the Polak-Ribiere method [4], the diagonal elements of $\mathbf{B}(k)$ can be computed as

$$\beta_i(k) = \frac{\left(\mathbf{g}_i(k) - \mathbf{g}_i(k-1)\right)^H \mathbf{g}_i(k)}{\mathbf{g}_i^H(k-1)\mathbf{g}_i(k-1)}, \; i = 1, \ldots, M, \qquad (22)$$

for $k \geq 2$, and $\mathbf{g}_i(k)$ is the i-th column of the gradient matrix $\mathbf{G}(k)$. The proposed algorithm, called hereafter MIMO-MCG, is summarized in Table I.

Following standard practice in DFE design, a decision delay should be inserted between equalizer decisions and transmitted symbols. As in [1], we consider a decision delay parameter Δ common for all streams, and set it to $\Delta = K_f - 1$. Hence, the decision $d_i(k)$ corresponds to symbol $s_i(k-\Delta)$.

The MIMO-MCG can be viewed as the application of the SISO-MCG algorithm of [4] to each of the linear systems of (10) independently. By using the rationale of [4], it can be shown that the MIMO-MCG converges to the solution of the system at steady state. As it will be shown by simulations, the performance of MIMO-MCG in terms of mean square error (MSE) is identical to that of the RLS algorithm. To reduce the complexity, we incorporate Galerkin projections into the MCG algorithm as explained next.

B. Galerkin Projection-Based MIMO-MCG

Recall that the problem at hand in (10) can be considered as a set of multiple linear equations evolving in time. According to MIMO-MCG, all linear systems $\mathbf{\Phi}(k)\mathbf{w}_i(k) = \mathbf{z}_i(k), \; i = 1, \ldots, M$, at the k-th time instant, are updated by the MCG algorithm. By utilizing the idea of Galerkin projections, an approximate solution can be obtained by MCG updating just a single system at each time instant k, while the others are updated through Galerkin projections. A round-robin policy is engaged for the selection of the system to be MCG-updated. Hence, the equalizer $\mathbf{w}_i$ is updated by the MCG method only when $i = mod(k, M) + 1$.

Then, one has to specify how the Galerkin projections of (15) should be performed. Due to the space-time nature of the problem, there are two choices, which are explained below.

TABLE II

SUMMARY OF MIMO-MCG-TP AND MIMO-MCG-SP

Select seed system as $j = mod(k, M) + 1$.

A. MCG Update

1) Update matrix $\mathbf{\Phi}(k)$ by using (8).
2) $\mathbf{t}_j(k) = \lambda \mathbf{g}_j(k-1) + \mathbf{y}(k) e_j^*(k)$.
3) Compute the step size $\alpha_j(k)$ from (20).
4) $\mathbf{w}_j(k) = \mathbf{w}_j(k-1) + \alpha_j(k)\mathbf{p}_j(k)$.
5) $\mathbf{g}_j(k) = \mathbf{t}_j(k) - \alpha_j(k)\mathbf{\Phi}(k)\mathbf{p}_j(k)$.
6) Compute the $\beta_j(k)$ from (22).
7) $\mathbf{p}_j(k+M) = \mathbf{g}_j(k) + \beta_j(k)\mathbf{p}_j(k)$.

For $i = 1, \ldots, M,\ i \neq mod(k, M) + 1$,

B1. Time Project

1) $\mathbf{t}_i(k) = \lambda \mathbf{g}_i(k-1) + \mathbf{y}(k) e_i^*(k)$.
2) Update the matrix-vector product $\mathbf{\Phi}(k)\mathbf{p_i}(l) = \lambda \mathbf{\Phi}(k-1)\mathbf{p_i}(l) + \mathbf{y}(k)\mathbf{y}^H(k)\mathbf{p}_i(l)$.
3) Compute $\alpha_i(k)$ from (24).
4) Update the equalizer filter according to (23).
5) $\mathbf{g}_i(k) = \mathbf{t}_i(k) - \alpha_i(k)\mathbf{\Phi}(k)\mathbf{p}_i(l)$.

or **B2. Space Project**

1) $\mathbf{t}_i(k) = \lambda \mathbf{g}_i(k-1) + \mathbf{y}(k) e_i^*(k)$.
2) Compute $\alpha_i(k)$ from (26).
3) Update the equalizer filter according to (26).
4) $\mathbf{g}_i(k) = \mathbf{t}_i(k) - \alpha_i(k)\mathbf{\Phi}(k)\mathbf{p}_j(k)$.

End

1) Time Projection Scheme (MIMO-MCG-TP): According to the Time Projection scheme (MIMO-MCG-TP), each stream is treated separately. The i-th stream is updated by the MCG algorithm only at the l-th time instant where $i = mod(l, M) + 1$, and $\mathbf{\Phi}(l)\mathbf{w}_i(l) = \mathbf{z}_i(l)$ is considered as the seed system. Hence, during the intermediate time instants, the equalizer $\mathbf{w}_i(k)$ is updated by using the direction $\mathbf{p}_i(l)$ as

$$\begin{aligned} \mathbf{w}_i(k) &= \mathbf{w}_i(k-1) + \alpha_i(k)\mathbf{p}_i(l), \\ \mathbf{p}_i(l) &= \mathbf{p}_i\left(\lceil (k-i)/M \rceil\right) \end{aligned} \quad (23)$$

for $i \neq mod(k, M) + 1$. The corresponding step size can be found similarly to (16) and (20) as

$$\alpha_i(k) = \frac{\mathbf{p}_i^H(l)\mathbf{t}_i(k)}{\mathbf{p}_i^H(l)\mathbf{\Phi}(k)\mathbf{p}_i(l)}. \quad (24)$$

Notice that the matrix-vector product in the denominator of (24) can be efficiently computed by utilizing the rank-1 updates of $\mathbf{\Phi}(k)$ (8) for $k = l+1, \ldots, l+M-1$.

2) Space Projection Scheme (MIMO-MCG-SP): In the Space Projection scheme (MIMO-MCG-SP), the adaptations of the equalizer filters are performed jointly. Similarly to the TP scheme, at the k-th time instant, the j-th system is selected as the seed system, where $j = mod(k, M) + 1$. After the seed system has been MCG-updated, the non-seed systems are updated according to Galerkin projections as

$$\mathbf{w}_i(k) = \mathbf{w}_i(k-1) + \alpha_i(k)\mathbf{p}_j(k), \quad (25)$$

$$\alpha_i(k) = \frac{\mathbf{p}_j^H(k)\mathbf{t}_i(k)}{\mathbf{p}_j^H(k)\mathbf{\Phi}(k)\mathbf{p}_j(k)}. \quad (26)$$

Note that the denominator in (26) has already been computed during the MCG update of the j-th equalizer (seed system), and thus the complexity of SP is even smaller than TP scheme.

TABLE III

COMPARISON OF COMPLEXITIES

Algorithm	Complex Multiply-Add Operations
MIMO-RLS	$\frac{9}{2}K^2 + 9MK + O(K)$
MIMO-MCG	$\left(2M + \frac{3}{2}\right)K^2 + 17MK + O(K)$
MIMO-MCG-TP	$\frac{7}{2}K^2 + 15MK + O(K)$
MIMO-MCG-SP	$\frac{7}{2}K^2 + 11MK + O(K)$

The two schemes described above are presented in Table II, where the initialization can be performed as in Table I. Since only a single system is MCG-updated at each time instant, while the non-seed systems are updated according to efficient Galerkin projections, both schemes have a computational complexity of $O(K^2 + MK)$. Note that the complexities of the proposed MCG schemes can be further reduced, by extending the idea of computing efficiently Toeplitz matrix - vector products, followed in [6], to the MIMO case. The complexities, in numbers of complex multiply-add operations per symbol period for all the proposed algorithms, as well as the MIMO-RLS of [2] implemented in a square-root fashion to avoid numerical instability [3], are presented in Table III.

Comparing the two projection alternatives, one should notice that in both schemes each system is MCG-updated every M time instants. However, during the rest of the time, TP scheme utilizes a single direction, i.e., $\mathbf{p}_i(l)$ to update its solution, while SP utilizes the $(M-1)$ quite different directions determined by the rest of the systems. Thus, when space projections are performed, the solution has much more degrees of freedom to move into the K-dimensional space and converges faster to its solution. Several combinations of the three algorithms proposed in this section can be performed, and space and/or time projections can substitute some MCG updates to trade convergence for computational complexity.

V. PERFORMANCE EVALUATION

The performance of the proposed equalizers was evaluated through computer simulations. We considered a system transmitting uncoded QPSK symbols of duration T_s=$0.25\mu sec$ over a wireless channel modelled according to the UMTS Vehicular Channel Model A [10]. This channel model consists of six independent, Rayleigh faded paths, with a power delay profile described in [10]. All transmitter-receiver links were considered independent. The SNR was defined as the expected SNR per bit (over the ensemble of channel realizations) on each receive antenna. The feedforward and feedback filters had a temporal span of K_f=20, and K_b=10 taps, respectively.

Initially, to study the convergence of the equalizers, the Doppler effect was ignored and the channel was kept static for an interval of $4096T_s$. An M=N=3 antenna configuration operating at SNR=16dB was simulated, while the system was constantly in training mode. Six different MIMO DFE algorithms were tested: (1) the MIMO-RLS of [2] implemented in a square root RLS fashion to avoid numerical instability [3], (2) the MIMO-MCG, (3) the MIMO-MCG-TP, (4) the MIMO-MCG-SP, (5) a scheme that executes only one

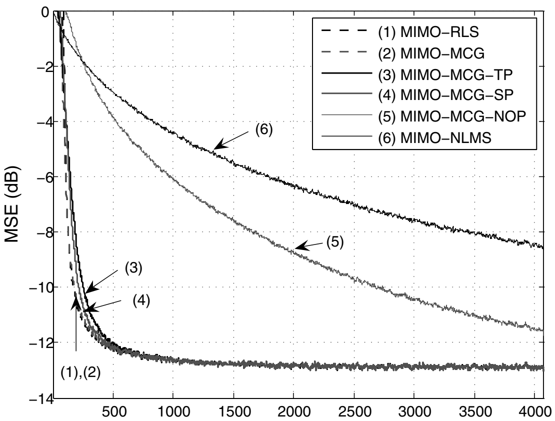

Fig. 1. Convergence of adaptive equalizers for a 3×3 static MIMO channel at SNR=16dB (λ=0.9989).

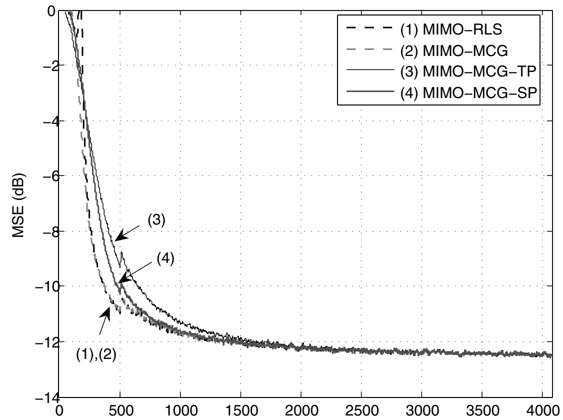

Fig. 2. Convergence of adaptive equalizers for a 6×6 static MIMO channel at SNR=16dB (λ=0.9994).

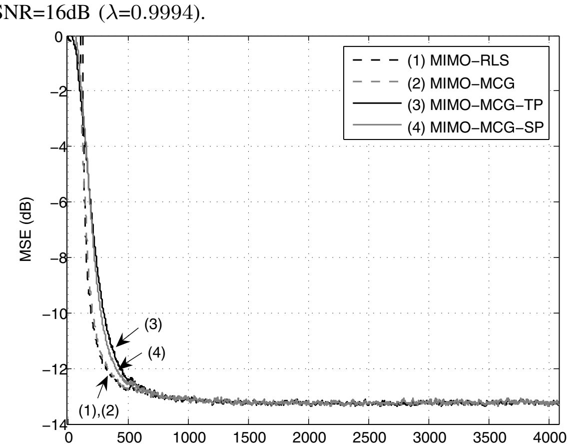

Fig. 3. Convergence of adaptive equalizers for a 4×4 slow fading MIMO system at SNR=16dB (λ=0.998).

MCG update every M iterations and does not perform any projection (MIMO-MCG-NOP), and (6) a MIMO normalized LMS (MIMO-NLMS) algorithm. In Fig. 1, the MSE, averaged over all streams and over 1000 independent runs, is plotted. By setting the parameter λ of the MCG algorithms equal to the forgetting factor of the RLS and adjusting accordingly the step size of the NLMS, all the equalizers were tuned to converge at the same steady state error. As it is shown, the MIMO-MCG curve coincides with that of MIMO-RLS, while MIMO-MCG-TP and MIMO-MCG-SP lie very close to it. The SP scheme is better than the TP as it was expected, while the MIMO-MCG-NOP and MIMO-NLMS exhibit very slow convergence. Note that the projections can offer significant improvement in performance at a limited additional cost.

Error propagation effects in decision-directed mode and the impact of MCG updates frequency were studied by simulating a system that operates over a 6×6 static channel. A training period of $512T_s$ was employed. In Fig. 2 the error curves of the proposed MCG equalizers, as well as the MIMO-RLS are plotted. As it can be seen, all equalizers are robust to error propagation effects. Moreover, by performing the MCG update less frequently (every M=6 iterations instead of 3 in Fig. 1), a slight degradation in performance can be observed. Finally, the tracking performance of the algorithms was studied by simulating a system that operates over a 4×4 slow fading channel. A normalized Doppler frequency $f_D T_s$=1.1 10^{-5} was simulated by using the Jakes method. As illustrated in Fig. 3, all the proposed equalizers successfully track channel variations.

VI. Conclusions

Three adaptive algorithms for updating a MIMO DFE have been developed. By extending the algorithm of [4], we derived an adaptive CG MIMO DFE. To reduce its complexity, we employed the idea of Galerkin projections, and two schemes of convergence close to RLS have been proposed. As shown by simulations, all the new equalizers can be successfully employed in practical MIMO wideband systems. The numerical stability of the MCG schemes as compared to the RLS algorithm and the impact of using a more sophisticated policy for choosing the system to be updated by MCG, are some of the issues that are currently being investigated.

References

[1] N. Al-Dhahir and A. H. Sayed, "The finite length multi-input multi-output MMSE-DFE," *IEEE Trans. Signal Processing*, vol. 48, no. 10, pp. 2921–2936, Oct. 2000.

[2] A. Maleki-Tehrani, B. Hassibi, and J. M. Cioffi, "Adaptive equalization of multiple-input multiple-output (MIMO) channels," in *Proc. IEEE Int. Conf. Communications*, New Orleans, LA, June 2000, pp. 1670–1674.

[3] V. Kekatos, A. A. Rontogiannis, and K. Berberidis, "Cholesky factorization - based adaptive BLAST DFE for wideband MIMO channels," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, no. 45789, pp. 1–11, 2007.

[4] P. S. Chang and A. N. Willson, "Analysis of conjugate gradient algorithms for adaptive filtering," *IEEE Trans. Signal Processing*, vol. 48, no. 2, pp. 409–418, 2000.

[5] J. S. Lim and C. K. Un, "Block conjugate gradient algorithms for adaptive filtering," *Signal Processing*, vol. 55, no. 1, pp. 65–77, 1996.

[6] A. S. Lalos and K. Berberidis, "A frequency domain conjugate gradient algorithm and its application to channel equalization," in *Proc. EUSIPCO*, Florence, Sept 2006.

[7] T. F. Chan and M. K. Ng, "Galerkin projection methods for solving multiple linear systems," *SIAM Journal on Scientific Computing*, vol. 21, no. 3, pp. 836–850, 2000.

[8] R. Fletcher, *Practical Methods of Optimization*, 2nd ed. Chichester, U.K.: Wiley, 2000.

[9] I. Groh, G. Dietl, and W. Utschick, "Preconditioned and rank-flexible block conjugate gradient implementations of MIMO Wiener decision feedback equalizers," in *Proc. IEEE ICASSP*, Toulouse, May 2006.

[10] "Selection procedures for the choice of radio transmission technologies of the UMTS," ETSI, Technical Report 101.112, Apr. 1998.