Reduced-Rank Equalization for EDGE Via Conjugate Gradient Implementation of Multi-Stage Nested Wiener Filter

Guido Dietl, Michael Joham, and Michael D. Zoltowski

54th IEEE Vehicular Technology Conference (VTC Fall 2001)
Atlantic City, USA

volume 3, pp. 1912–1916, October 7th–11th, 2001

Munich University of Technology
Institute for Circuit Theory and Signal Processing
http://www.nws.ei.tum.de

Reduced-Rank Equalization for EDGE Via Conjugate Gradient Implementation of Multi-Stage Nested Wiener Filter

Guido Dietl, Michael D. Zoltowski, and Michael Joham

Abstract—**The Wiener filter solves the Wiener-Hopf equation and may be approximated by the Multi-Stage Nested Wiener Filter (MSNWF) which lies in the Krylov subspace of the covariance matrix of the observation and the crosscorrelation vector between the observation and the desired signal. Moreover, since the covariance matrix is Hermitian, the Lanczos algorithm can be used to compute the Krylov subspace basis.**

The Conjugate Gradient (CG) method is another approach in order to solve a system of linear equations. In this paper, we derive the relationship between the CG method and the Lanczos based MSNWF and finally transform the formulas of the MSNWF into those of the CG algorithm. Consequently, we present a CG based MSNWF where the filter weights and the Mean Square Error (MSE) are updated at each iteration step.

The obtained algorithm is used to linearily equalize the received signal in an Enhanced Data rates for GSM Evolution (EDGE) system. Simulation results demonstrate the ability of the MSNWF to reduce receiver complexity while maintaining the same level of system performance.

Keywords—**Adaptive Filtering, Conjugate Gradients, EDGE, Multipath Propagation, Reduced-Rank Equalization, Space-Time Processing, Wireless Communications.**

I. INTRODUCTION

THE *Wiener filter* (WF) [1] estimates an unknown signal $d_0[n]$ from an observation signal $\boldsymbol{x}_0[n]$ by minimizing the *Mean Square Error* (MSE) and needs only second order statistics. Due to the necessity of solving the *Wiener-Hopf equation* the computational complexity is high, especially for observations $\boldsymbol{x}_0[n]$ of high dimensionality.

The *Principal Component* (PC) *method* [2] was the first approach to approximate the WF. The eigenvectors corresponding to the principal eigenvalues of the covariance matrix of the observation are composed to a pre-filter matrix which is applied to the observation signal. Then, a WF of reduced dimensionality estimates the desired signal from the transformed observation. An alternative approach uses the *Cross-Spectral* (CS) *metric* [3] instead of the eigenvalue to choose the eigenvectors which compose the pre-filter matrix. More recently, Goldstein et. al. developed the *Multi-Stage Nested Wiener Filter* (MSNWF) [4] where the columns of the pre-filter matrix are no longer eigenvectors of the covariance matrix. The improvement made by the MSNWF shows that dimensionality reduction of the observation signal based on eigenvectors is generally suboptimal.

Honig et. al. [5] observed that the MSNWF is the solution of the Wiener-Hopf equation in the *Krylov subspace* of the covariance matrix of the observation and the crosscorrelation vector between the observation and the desired signal. Thus, the *Arnoldi algorithm* may be used to generate the columns of the pre-filter matrix. Moreover, since the covariance matrix is Hermitian, the Arnoldi algorithm can be replaced by the *Lanczos algorithm*. The resulting order-recursive version of the MSNWF [6] updates the filter weights and the *Mean Square Error* (MSE) at each iteration step.

Our contribution is to derive the relationship between the Lanczos based implementation of the MSNWF and the *Conjugate Gradient* (CG) *method* which was introduced by Hestenes and Stiefel [7] in order to solve a system of linear equations. Moreover, we transform the formulas of the Lanczos based MSNWF algorithm to yield a formulation of the CG algorithm, and finally present a new implementation of the MSNWF where the filter weights and the *Mean Square Error* (MSE) between the estimated and desired signal are updated at each iteration step.

In the next section, we recall the Lanczos based MSNWF. Before we show the relationship between the considered algorithms in Section IV, we review the CG algorithm in Section III. Finally, we present a new formulation of the MSNWF algorithm in Section V and apply it to an EDGE system in Section VI.

Throughout the paper the covariance matrix of a vector $\boldsymbol{x}[n]$ is denoted by $\boldsymbol{R}_{\boldsymbol{x}} = \mathrm{E}\{\boldsymbol{x}[n]\boldsymbol{x}^{\mathrm{H}}[n]\}$, the crosscorrelation of a vector $\boldsymbol{x}[n]$ and a scalar $d[n]$ is $\boldsymbol{r}_{\boldsymbol{x},d} = \mathrm{E}\{\boldsymbol{x}[n]d^*[n]\}$, and the variance of a scalar $d[n]$ is $\sigma_d^2 = \mathrm{E}\{|d[n]|^2\}$.

G. Dietl and M. Joham, Institute for Circuit Theory and Signal Processing, Munich University of Technology, D-80333 Munich, Germany, e-mail: {gudi,mijo}@nws.ei.tum.de

M. D. Zoltowski, School of Electrical Engineering, Purdue University, West Lafayette, IN 47907-1285, USA, e-mail: mikedz@ecn.purdue.edu

II. LANCZOS BASED MSNWF

Applying the linear filter $\boldsymbol{w} \in \mathbb{C}^N$ to the observation signal $\boldsymbol{x}_0[n] \in \mathbb{C}^N$ leads to the estimate $\hat{d}_0[n] = \boldsymbol{w}^{\mathrm{H}}\boldsymbol{x}_0[n]$ of the desired signal $d_0[n] \in \mathbb{C}$. The *mean square error*

$$\mathrm{MSE}_0 = \sigma_{d_0}^2 - 2\mathrm{Re}\left\{\boldsymbol{w}^{\mathrm{H}}\boldsymbol{r}_{\boldsymbol{x}_0,d_0}\right\} + \boldsymbol{w}^{\mathrm{H}}\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{w} \qquad (1)$$

is the variance of the error $d_0[n] - \hat{d}_0[n]$. The *Wiener filter* (WF) minimizes MSE_0 leading to the *Wiener-Hopf equation*

$$\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{w}_0 = \boldsymbol{r}_{\boldsymbol{x}_0,d_0}, \qquad (2)$$

whose solution, the WF $\boldsymbol{w}_0$, achieves the *minimum mean square error* $\mathrm{MMSE}_0 = \sigma_{d_0}^2 - \boldsymbol{r}_{\boldsymbol{x}_0,d_0}^{\mathrm{H}}\boldsymbol{R}_{\boldsymbol{x}_0}^{-1}\boldsymbol{r}_{\boldsymbol{x}_0,d_0}$.

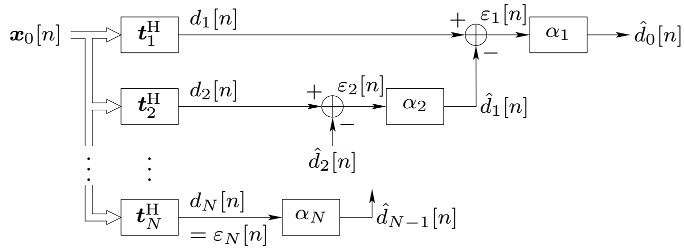

Fig. 1. MSNWF as Filter Bank

Figure 1 shows the block diagram of the *Multi-Stage Nested Wiener Filter* (MSNWF, [4]) that solves Equation (2). The first filter $\boldsymbol{t}_1$ is the normalized matched filter $\boldsymbol{r}_{\boldsymbol{x}_0,d_0}/\|\boldsymbol{r}_{\boldsymbol{x}_0,d_0}\|_2$ and the i-th filter $\boldsymbol{t}_i$ maximizes the real part of the correlation between its output $d_i[n]$ and the output $d_{i-1}[n]$ of the previous filter $\boldsymbol{t}_{i-1}$. If we restrict the filters $\boldsymbol{t}_i$ to be orthonormal, the i-th filter can be computed with following optimization [6]:

$$\begin{aligned} &\boldsymbol{t}_i = \arg\max_{\boldsymbol{t}} \mathrm{E}\{\mathrm{Re}(d_i[n]d_{i-1}^*[n])\} \\ &\mathrm{s.\,t.:} \quad \boldsymbol{t}^{\mathrm{H}}\boldsymbol{t} = 1 \quad \text{and} \quad \boldsymbol{t}^{\mathrm{H}}\boldsymbol{t}_k = 0, k = 1, \ldots, i-1. \end{aligned} \tag{3}$$

whose solution is the well known *Arnoldi iteration* (e. g. [8]):

$$\boldsymbol{t}_i = \frac{\left(\prod_{k=1}^{i-1} \boldsymbol{P}_k\right) \boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{t}_{i-1}}{\left\|\left(\prod_{k=1}^{i-1} \boldsymbol{P}_k\right) \boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{t}_{i-1}\right\|_2} \in \mathbb{C}^N, \tag{4}$$

with the projector $\boldsymbol{P}_k = \mathbf{1}_N - \boldsymbol{t}_k\boldsymbol{t}_k^{\mathrm{H}}$ onto the space orthogonal to $\boldsymbol{t}_k$ and $\mathbf{1}_N$ denotes the $N \times N$ identity matrix. Since $\boldsymbol{R}_{\boldsymbol{x}_0}$ is Hermitian, we can use the *Lanczos algorithm*

$$\boldsymbol{t}_i = \frac{\boldsymbol{P}_{i-1}\boldsymbol{P}_{i-2}\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{t}_{i-1}}{\|\boldsymbol{P}_{i-1}\boldsymbol{P}_{i-2}\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{t}_{i-1}\|_2} \tag{5}$$

which leads to a tridiagonal covariance matrix $\boldsymbol{R}_{\boldsymbol{d}}$ of the pre-filtered observation vector $\boldsymbol{d} = [d_1[n], \ldots, d_N[n]]$. The scalar WFs α_i estimate the output of the previous filter $d_{i-1}[n]$ from the error signal $\varepsilon_i[n]$.

For a reduced rank MSNWF we use the first D basis vectors to build the pre-filter matrix $\boldsymbol{T}^{(D)} = [\boldsymbol{t}_1, \ldots, \boldsymbol{t}_D] \in \mathbb{C}^{N\times D}$ which leads to the length L observation vector $\boldsymbol{d}^{(D)}[n] = \boldsymbol{T}^{(D),\mathrm{H}}\boldsymbol{x}_0[n]$. The WF to estimate $d_0[n]$ from $\boldsymbol{d}^{(D)}[n]$ can be written as $\boldsymbol{w}_{\boldsymbol{d}}^{(D)} = \left(\boldsymbol{T}^{(D),\mathrm{H}}\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{T}^{(D)}\right)^{-1}\boldsymbol{T}^{(D),\mathrm{H}}\boldsymbol{r}_{\boldsymbol{x}_0,d_0}$, thus, the rank D approximation of the WF reads as

$$\boldsymbol{w}_0^{(D)} = \boldsymbol{T}^{(D)}\left(\boldsymbol{T}^{(D),\mathrm{H}}\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{T}^{(D)}\right)^{-1}\boldsymbol{T}^{(D),\mathrm{H}}\boldsymbol{r}_{\boldsymbol{x}_0,d_0}, \tag{6}$$

which yields to the mean square error

$$\mathrm{MSE}^{(D)} = \sigma_{d_0}^2 - \boldsymbol{w}_0^{(D),\mathrm{H}}\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{w}_0^{(D)}. \tag{7}$$

Note that the rank D MSNWF is equivalent [5], [6] to solving the Wiener-Hopf equation in the D-dimensional *Krylov subspace* $\mathcal{K}^{(D)}(\boldsymbol{R}_{\boldsymbol{x}_0}, \boldsymbol{r}_{\boldsymbol{x}_0,d_0})$ and $\mathcal{K}^{(D)}(\boldsymbol{A}, \boldsymbol{b}) = \mathrm{span}\left(\left[\boldsymbol{b}, \boldsymbol{A}\boldsymbol{b}, \ldots, \boldsymbol{A}^{(D-1)}\boldsymbol{b}\right]\right)$.

In [6], a Lanczos algorithm based order-recursive MSNWF (cf. Algorithm I) was presented which exploits the tri-diagonality of the covariance matrix $\boldsymbol{R}_{\boldsymbol{d}}^{(D)}$ of $\boldsymbol{d}^{(D)}[n]$ and the simple structure of the crosscorrelation vector $\boldsymbol{r}_{\boldsymbol{d},d_0}^{(D)} = \left[\|\boldsymbol{r}_{\boldsymbol{x}_0,d_0}\|_2, \mathbf{0}^{\mathrm{T}}\right]^{\mathrm{T}}$ to compute the WF in reduced dimensions $\boldsymbol{w}_{\boldsymbol{d}}^{(D)}$ from the previous one $\boldsymbol{w}_{\boldsymbol{d}}^{(D-1)}$. Additionally, the MSE is computed at each step.

III. Conjugate Gradient Algorithm

The *Conjugate Gradient* (CG) *algorithm* [7] is an iterative method in order to solve a system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ of N linear equations in N unknowns where the matrix $\boldsymbol{A} \in \mathbb{C}^{N\times N}$ is assumed to be Hermitian and positive definite. If an exact solution exists, it is obtained after N steps. Stopping the iteration after $D < N$ steps yields an approximate solution of the problem.

ALGORITHM I

LANCZOS BASED MSNWF

$\boldsymbol{t}_0 = \mathbf{0}$
$\boldsymbol{t}_1 = \boldsymbol{r}_{\boldsymbol{x}_0,d_0}/\|\boldsymbol{r}_{\boldsymbol{x}_0,d_0}\|_2$
$r_{0,1} = 0, \quad r_{1,1} = \beta_1 = \boldsymbol{t}_1^{\mathrm{H}}\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{t}_1$
$c_{\mathrm{first}}^{(1)} = c_{\mathrm{last}}^{(1)} = r_{1,1}^{-1}$
$\mathrm{MSE}^{(1)} = \sigma_{d_0}^2 - \|\boldsymbol{r}_{\boldsymbol{x}_0,d_0}\|_2^2\, c_{\mathrm{first}}^{(1)}$
for $i = 2$ to D **do**
$\boldsymbol{v} = \boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{t}_{i-1} - r_{i-1,i-1}\boldsymbol{t}_{i-1} - r_{i-2,i-1}\boldsymbol{t}_{i-2}$
$r_{i-1,i} = \|\boldsymbol{v}\|_2$
$\boldsymbol{t}_i = \boldsymbol{v}/r_{i-1,i}$
$r_{i,i} = \boldsymbol{t}_i^{\mathrm{H}}\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{t}_i$
$\beta_i = r_{i,i} - r_{i-1,i}^2\beta_{i-1}^{-1}$
$\boldsymbol{c}_{\mathrm{first}}^{(i)} = \begin{bmatrix} \boldsymbol{c}_{\mathrm{first}}^{(i-1)} \\ 0 \end{bmatrix} + \beta_i^{-1}c_{\mathrm{last},1}^{(i-1)}\begin{bmatrix} r_{i-1,i}^2\boldsymbol{c}_{\mathrm{last}}^{(i-1)} \\ -r_{i-1,i} \end{bmatrix}$
$\boldsymbol{c}_{\mathrm{last}}^{(i)} = \beta_i^{-1}\begin{bmatrix} -r_{i-1,i}\boldsymbol{c}_{\mathrm{last}}^{(i-1)} \\ 1 \end{bmatrix}$
$\mathrm{MSE}^{(i)} = \sigma_{d_0}^2 - \|\boldsymbol{r}_{\boldsymbol{x}_0,d_0}\|_2^2\, c_{\mathrm{first},1}^{(i)}$
end for
$\boldsymbol{T}^{(D)} = \begin{bmatrix} \boldsymbol{t}_1 & \cdots & \boldsymbol{t}_D \end{bmatrix}$
$\boldsymbol{w}_0^{(D)} = \|\boldsymbol{r}_{\boldsymbol{x}_0,d_0}\|_2\,\boldsymbol{T}^{(D)}\boldsymbol{c}_{\mathrm{first}}^{(D)}$

ALGORITHM II

CONJUGATE GRADIENT METHOD

$\boldsymbol{x}^{(0)} = \mathbf{0}$
2: $\boldsymbol{p}_1 = -\boldsymbol{r}_1 = \boldsymbol{b}$
for $i = 1$ to D **do**
4: $\gamma_i = -\left(\boldsymbol{p}_i^{\mathrm{H}}\boldsymbol{r}_i\right)/\left(\boldsymbol{p}_i^{\mathrm{H}}\boldsymbol{A}\boldsymbol{p}_i\right)$
$\boldsymbol{x}^{(i)} = \boldsymbol{x}^{(i-1)} + \gamma_i\boldsymbol{p}_i$
6: $\boldsymbol{r}_{i+1} = \boldsymbol{r}_i + \gamma_i\boldsymbol{A}\boldsymbol{p}_i$
$\delta_i = \left(\boldsymbol{p}_i^{\mathrm{H}}\boldsymbol{A}\boldsymbol{r}_{i+1}\right)/\left(\boldsymbol{p}_i^{\mathrm{H}}\boldsymbol{A}\boldsymbol{p}_i\right)$
8: $\boldsymbol{p}_{i+1} = -\boldsymbol{r}_{i+1} + \delta_i\boldsymbol{p}_i$
end for

Algorithm II is one possible implementation of the CG method. The fundamental recursion formula which updates the approximate solution of the system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ is Line 5 of Algorithm II, where $\boldsymbol{p}_i$ is the *search direction* at iteration step i and γ_i is its *weight factor*. The choice of γ_i (cf. Line 4 of Algorithm II) ensures that the approximate solution $\boldsymbol{x}^{(i)}$ minimizes the error function

$$\begin{aligned} e(\boldsymbol{x}) &= \left(\boldsymbol{x}^{(N)} - \boldsymbol{x}\right)^{\mathrm{H}}\boldsymbol{A}\left(\boldsymbol{x}^{(N)} - \boldsymbol{x}\right) \\ &= \boldsymbol{x}^{\mathrm{H}}\boldsymbol{A}\boldsymbol{x} - \boldsymbol{x}^{\mathrm{H}}\boldsymbol{b} - \boldsymbol{b}^{\mathrm{H}}\boldsymbol{x} + \boldsymbol{x}^{(N),\mathrm{H}}\boldsymbol{b} \end{aligned} \tag{8}$$

on the line $\boldsymbol{x} = \boldsymbol{x}^{(i-1)} + \gamma\boldsymbol{p}_i$. $\boldsymbol{x}^{(N)}$ denotes the exact solution of the system. Line 6 of Algorithm II calculates the *residual* $\boldsymbol{r}_{i+1} = \boldsymbol{A}\boldsymbol{x}^{(i)} - \boldsymbol{b}$ which belongs to the approximation $\boldsymbol{x}^{(i)}$. The index mismatch is useful for the derivations we make in Section IV. Finally, the remaining Lines 7 and 8 of Algorithm II update the search direction $\boldsymbol{p}_i$ and ensure its $\boldsymbol{A}$-conjugacy to any vector $\boldsymbol{p}_j$, $j \neq i$, i. e.

$$\boldsymbol{p}_i^{\mathrm{H}} \boldsymbol{A} \boldsymbol{p}_j = 0 \qquad \forall i \neq j. \tag{9}$$

Thus, the CG algorithm belongs to the family of *Conjugate Directions* (CD) *methods*. Moreover, the CG method is a special case of the CD methods because the residuals are mutually orthogonal. It holds

$$\boldsymbol{r}_i^{\mathrm{H}} \boldsymbol{r}_j = 0 \qquad \forall i \neq j. \tag{10}$$

The proofs of Equations (9) and (10) [7], [9] are not shown in this paper due to space limitations.

IV. Relationship Between MSNWF and CG Algorithm

In numerous papers and books [6], [10] it was mentioned that the Lanczos algorithm which is used by our implementation of the MSNWF is only a version of the CG algorithm. In the following, the system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ solved by the CG algorithm is replaced by the Wiener-Hopf equation (cf. Equation 2). Consequently, the optimization functions in Equations (1) and (8) are the same except for a constant which does not change the minimum. Moreover, the solution at each step i is searched in the same Krylov subspace $\mathcal{K}^{(D)}(\boldsymbol{R}_{\boldsymbol{x}_0}, \boldsymbol{r}_{\boldsymbol{x}_0,d_0})$ (e. g. [8], [9]). To establish the equivalence of both algorithms, we transform [11], [9] the formulas of the Lanczos based MSNWF to those of the CG algorithm.

Assume that $D \geq i \geq 2$. Using Lines 17 and 12 of Algorithm I, and by setting $\boldsymbol{T}^{(i)} = [\boldsymbol{T}^{(i-1)}, \boldsymbol{t}_i]$, it holds for $\boldsymbol{w}_0^{(i)}$ that

$$\begin{aligned} \boldsymbol{w}_0^{(i)} &= \left\|\boldsymbol{r}_{\boldsymbol{x}_0,d_0}\right\|_2 \boldsymbol{T}^{(i)} \boldsymbol{c}_{\text{first}}^{(i)} = \boldsymbol{w}_0^{(i-1)} + \eta_i \boldsymbol{u}_i, \\ \boldsymbol{w}_0^{(1)} &= \left\|\boldsymbol{r}_{\boldsymbol{x}_0,d_0}\right\|_2 r_{1,1}^{-1} \boldsymbol{t}_1, \end{aligned} \tag{11}$$

where η_i and $\boldsymbol{u}_i$ are defined as

$$\eta_i = \left\|\boldsymbol{r}_{\boldsymbol{x}_0,d_0}\right\|_2 \varrho_i^{-1} \beta_i^{-1} r_{i-1,i} c_{\text{last},1}^{(i-1)}, \tag{12}$$

$$\boldsymbol{u}_i = \varrho_i \left(r_{i-1,i} \boldsymbol{T}^{(i-1)} \boldsymbol{c}_{\text{last}}^{(i-1)} - \boldsymbol{t}_i \right), \tag{13}$$

and where $\varrho_i \in \mathbb{R} \setminus \{0\}$ is so far an arbitrary factor which we explain later. Multiplying Line 13 of Algorithm I on the left side by $\boldsymbol{T}^{(i)}$ and using Equation (13) leads to $\boldsymbol{T}^{(i)} \boldsymbol{c}_{\text{last}}^{(i)} = -\varrho_i^{-1} \beta_i^{-1} \boldsymbol{u}_i$ which we replace in Equation (13) in order to get the recursion formula for $\boldsymbol{u}_{i+1}$ and $D > i \geq 1$

$$\boldsymbol{u}_{i+1} = \psi_i \boldsymbol{u}_i - \boldsymbol{g}_{i+1}, \quad \boldsymbol{u}_1 = -\varrho_1 \boldsymbol{t}_1, \tag{14}$$

where

$$\psi_i = -\varrho_{i+1} \varrho_i^{-1} \beta_i^{-1} r_{i,i+1}, \tag{15}$$

$$\boldsymbol{g}_{i+1} = \varrho_{i+1} \boldsymbol{t}_{i+1}. \tag{16}$$

In Equations (11) and (14) we observe the analogies to Lines 5 and 8 of Algorithm II if we substitute the vectors $\boldsymbol{u}_i$ and $\boldsymbol{g}_i$ by the search direction $\boldsymbol{p}_i$ and the residual $\boldsymbol{r}_i$ of the CG algorithm, respectively. In the sequel, we prove that the remaining formulas of the CG algorithm can be obtained from the MSNWF, too.

Proposition 1: The vectors $\boldsymbol{g}_i$ for $D \geq i \geq 1$ can be updated by the recursion formula

$$\begin{aligned} \boldsymbol{g}_{i+1} &= \boldsymbol{g}_i + \eta_i \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{u}_i, \\ \boldsymbol{g}_1 &= \varrho_1 \boldsymbol{t}_1, \quad \varrho_1 = -\left\|\boldsymbol{r}_{\boldsymbol{x}_0,d_0}\right\|_2, \end{aligned} \tag{17}$$

where the definition of η_i is given in Equation (12) and $\eta_1 := r_{1,1}^{-1}$. Therefore, the vectors $\boldsymbol{g}_i$ are residual vectors for $D+1 \geq i \geq 1$, i. e.

$$\boldsymbol{g}_i = \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{w}_0^{(i-1)} - \boldsymbol{r}_{\boldsymbol{x}_0,d_0} \tag{18}$$

and the absolute value of the factor ϱ_i is its length.

Proof: First, we prove Equation (17) by induction. Set $i = 1$. Recall that $\boldsymbol{u}_1 = \|\boldsymbol{r}_{\boldsymbol{x}_0,d_0}\|_2 \boldsymbol{t}_1 = -\boldsymbol{g}_1$ and we get by using $\eta_1 := r_{1,1}^{-1}$ and Lines 7 to 9 of Algorithm I

$$\begin{aligned} \boldsymbol{g}_1 + \eta_1 \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{u}_1 &= \left\|\boldsymbol{r}_{\boldsymbol{x}_0,d_0}\right\|_2 r_{1,1}^{-1} \left(\eta_1^{-1} \boldsymbol{t}_1 - \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{t}_1 \right) \\ &= \left\|\boldsymbol{r}_{\boldsymbol{x}_0,d_0}\right\|_2 r_{1,1}^{-1} r_{1,2} \boldsymbol{t}_2 = \varrho_2 \boldsymbol{t}_2 = \boldsymbol{g}_2, \end{aligned}$$

where $\varrho_2 := \|\boldsymbol{r}_{\boldsymbol{x}_0,d_0}\|_2 r_{1,1}^{-1} r_{1,2}$. Now assume that Equation (17) holds for $i = n-1$. This leads to $\boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{u}_{n-1} = \eta_{n-1}^{-1}(\boldsymbol{g}_n - \boldsymbol{g}_{n-1})$. To prove Equation (17) for $i = n$ if it holds for $i = n-1$, we need the product between $\boldsymbol{R}_{\boldsymbol{x}_0}$ and Equation (14), i. e. $\boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{u}_n = \psi_{n-1} \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{u}_{n-1} - \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{g}_n$. Finally, we get by using the expression for $\boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{u}_n$ and $\boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{u}_{n-1}$

$$\begin{aligned} \boldsymbol{g}_n + \eta_n \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{u}_n = &- \varrho_n \eta_n \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{t}_n \\ &+ \varrho_n \left(1 + \eta_{n-1}^{-1} \eta_n \psi_{n-1} \right) \boldsymbol{t}_n \\ &- \varrho_{n-1} \eta_{n-1}^{-1} \eta_n \psi_{n-1} \boldsymbol{t}_{n-1}. \end{aligned} \tag{19}$$

On the other hand, multiplying the recursion formula in Lines 7 to 9 of Algorithm I by ϱ_{n+1} yields

$$\begin{aligned} \varrho_{n+1} \boldsymbol{t}_{n+1} = &+ \varrho_{n+1} r_{n,n+1}^{-1} \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{t}_n \\ &- \varrho_{n+1} r_{n,n+1}^{-1} r_{n,n} \boldsymbol{t}_n \\ &- \varrho_{n+1} r_{n,n+1}^{-1} r_{n-1,n} \boldsymbol{t}_{n-1}. \end{aligned} \tag{20}$$

Consequently, since $\boldsymbol{g}_{n+1} = \varrho_{n+1} \boldsymbol{t}_{n+1}$, Equation (17) of Proposition 1 is proven if the following three equations are true:

$$-\varrho_n \eta_n = \varrho_{n+1} r_{n,n+1}^{-1}, \tag{21}$$

$$\varrho_n \left(1 + \eta_{n-1}^{-1} \eta_n \psi_{n-1} \right) = -\varrho_{n+1} r_{n,n+1}^{-1} r_{n,n}, \tag{22}$$

$$\varrho_{n-1} \eta_{n-1}^{-1} \eta_n \psi_{n-1} = \varrho_{n+1} r_{n,n+1}^{-1} r_{n-1,n}. \tag{23}$$

Take Equation (21), replace $n+1$ by i and solve it for ϱ_i by using Equation (12) and the expression $c_{\text{last},1}^{(i-1)} = -r_{i-2,i-1} \beta_{i-1}^{-1} c_{\text{last},1}^{(i-2)}$ (cf. Line 13 of Algorithm I). It follows for $D > i \geq 2$

$$\begin{aligned} \varrho_i &= -\varrho_{i-1} \eta_{i-1} r_{i-1,i}, && (24) \\ &= \left\|\boldsymbol{r}_{\boldsymbol{x}_0,d_0}\right\|_2 r_{i-1,i} c_{\text{last},1}^{(i-1)}. && (25) \end{aligned}$$

If we plug Equations (25) and (24) in Equations (12) and (15), respectively, we get simpler formulas for η_i and ψ_i

$$\eta_i = \beta_i^{-1}, \qquad D \geq i \geq 1, \tag{26}$$

$$\psi_i = \beta_i^{-2} r_{i,i+1}^2, \qquad D > i \geq 1. \tag{27}$$

Finally, with Equations (24), (26), (27) and Line 11 of Algorithm I, it can easily be shown [11], [9] that Equations (22) and (23) hold. Thus, Equation (17) is established.

It remains to verify Equation (18) in order to complete the proof of Proposition 1. Again, this is done by induction. Set $i = 1$. It holds that $\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{w}_0^{(0)} - \boldsymbol{r}_{\boldsymbol{x}_0,d_0} = -\|\boldsymbol{r}_{\boldsymbol{x}_0,d_0}\|_2\boldsymbol{t}_1 = \boldsymbol{g}_1$, where $\boldsymbol{w}_0^{(0)} = \boldsymbol{0}$. Assume that Equation (18) holds for $i = n-1$. Hence, we get for $i = n$ (cf. Equation 11)

$$\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{w}_0^{(n-1)} - \boldsymbol{r}_{\boldsymbol{x}_0,d_0} = \boldsymbol{g}_{n-1} + \eta_{n-1}\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{u}_{n-1} = \boldsymbol{g}_n.$$

The last equality holds because of Equation (17) which we have already proven. ■

Up to now we showed, in addition to the already found anologies, that the residuals $\boldsymbol{g}_i$ are updated by a similar formula as the residuals $\boldsymbol{r}_i$. To derive the remaining problem, we need to show that the computation of the weight factors ψ_i and η_i is similar to those of the CG algorithm.

Proposition 2: The factor ψ_i in Equation (14) satisfies the relation

$$\psi_i = \frac{\boldsymbol{u}_i^{\mathrm{H}}\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{g}_{i+1}}{\boldsymbol{u}_i^{\mathrm{H}}\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{u}_i} \tag{28}$$

and thus, $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_D\}$ is a set of $\boldsymbol{R}_{\boldsymbol{x}_0}$-conjugate vectors.

Proof: Use Equation (14) and the fact that $\boldsymbol{u}_j\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{g}_i = 0$ for $j \leq i-2$ [11], [9], to get

$$\boldsymbol{u}_i^{\mathrm{H}}\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{g}_{i+1} = -\varrho_i\varrho_{i+1}r_{i,i+1}. \tag{29}$$

To optain a similar expression for the denominator on the right side of Equation (28), substitute $\boldsymbol{u}_i$ given by Equation (14) and use Equations (16), (27), (29), and Line 10 of Algorithm I.

$$\boldsymbol{u}_i^{\mathrm{H}}\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{u}_i = \varrho_i^2\left(r_{i,i} - r_{i-1,i}^2\beta_{i-1}^{-1}\left(2 - \frac{\boldsymbol{u}_{i-1}^{\mathrm{H}}\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{u}_{i-1}}{\varrho_{i-1}^2\beta_{i-1}}\right)\right)$$

Comparing the equation above with Line 11 of Algorithm I yields the denominator $\boldsymbol{u}_i^{\mathrm{H}}\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{u}_i = \varrho_i^2\beta_i$ and Equation (28) is established. Thus, similar to δ_i in Algorithm II, ψ_i and Equations (14) ensure that the vectors $\boldsymbol{u}_i$ are mutually $\boldsymbol{R}_{\boldsymbol{x}_0}$-conjugate. ■

Proposition 3: It holds for $D \geq i \geq 1$ that

$$\eta_i = -\frac{\boldsymbol{u}_i^{\mathrm{H}}\boldsymbol{g}_i}{\boldsymbol{u}_i^{\mathrm{H}}\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{u}_i}. \tag{30}$$

Thus, with $\boldsymbol{w} = \boldsymbol{w}_0^{(i-1)} + \eta_i\boldsymbol{u}_i$ the value of η_i above minimizes the error function $e(\boldsymbol{w})$.

Proof: Replacing $\boldsymbol{g}_i$ by Equation (18) and $\boldsymbol{u}_i^{\mathrm{H}}\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{u}_i$ by $\varrho_i^2\beta_i$, and using recursively Equation (11) yields

$$-\frac{\boldsymbol{u}_i^{\mathrm{H}}\boldsymbol{g}_i}{\boldsymbol{u}_i^{\mathrm{H}}\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{u}_i} = \varrho_i^{-2}\beta_i^{-1}\boldsymbol{u}_i^{\mathrm{H}}\boldsymbol{r}_{\boldsymbol{x}_0,d_0},$$

because the vectors $\boldsymbol{u}_i$ are mutually $\boldsymbol{R}_{\boldsymbol{x}_0}$-conjugate (cf. Proposition 2). Now, complete the proof by induction. Set $i = 1$. It holds by using $\boldsymbol{u}_1 = -\varrho_1\boldsymbol{t}_1$ and Equation (26) that $\varrho_1^{-2}\beta_1^{-1}\boldsymbol{u}_1^{\mathrm{H}}\boldsymbol{r}_{\boldsymbol{x}_0,d_0} = \beta_1^{-1} = \eta_1$. Then, set $D \geq i \geq 2$ and substitute $\boldsymbol{u}_i$ by Equation (13). It follows

$$\varrho_i^{-2}\beta_i^{-1}\boldsymbol{u}_i^{\mathrm{H}}\boldsymbol{r}_{\boldsymbol{x}_0,d_0} = \varrho_i^{-1}\beta_i^{-1}r_{i-1,i}c_{\mathrm{last},1}^{(i-1)}\|\boldsymbol{r}_{\boldsymbol{x}_0,d_0}\|_2 = \eta_i,$$

if we remember that $\boldsymbol{t}_i^{\mathrm{H}}\boldsymbol{r}_{\boldsymbol{x}_0,d_0} = 0$ for all $D \geq i \geq 2$. ■

Thus, we see that the remaining Equations (30) and (28) which compute the weight factors ψ_i and η_i are similar to Lines 4 and 7 of Algorithm II. To put it in a nutshell, the Lanczos based MSNWF uses the same formulas as the CG algorithm if we make the following equivalences:

$$\text{approximate solution:} \quad \boldsymbol{w}_0^{(i)} \leftrightarrow \boldsymbol{x}^{(i)}, \qquad \eta_i \leftrightarrow \gamma_i, \tag{31}$$

$$\text{search directions:} \quad \boldsymbol{u}_i \leftrightarrow \boldsymbol{p}_i, \qquad \psi_i \leftrightarrow \delta_i, \tag{32}$$

$$\text{residuals:} \quad \boldsymbol{g}_i \leftrightarrow \boldsymbol{r}_i. \tag{33}$$

V. CG based MSNWF

In Section IV we have derived that the Lanczos based MSNWF can be expressed by the CG algorithm. Thus, Equations (30) and (28) can be replaced [7], [9] as follows

$$\eta_i = \frac{\boldsymbol{g}_i^{\mathrm{H}}\boldsymbol{g}_i}{\boldsymbol{u}_i^{\mathrm{H}}\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{u}_i}, \qquad \psi_i = \frac{\boldsymbol{g}_{i+1}^{\mathrm{H}}\boldsymbol{g}_{i+1}}{\boldsymbol{g}_i^{\mathrm{H}}\boldsymbol{g}_i}. \tag{34}$$

This reduces computational complexity because the matrix vector multiplication $\boldsymbol{R}_{\boldsymbol{x}_0}\boldsymbol{g}_{i+1}$ in Equation (28) is avoided. The resulting computational complexity for a rank D MSNWF is $O(N^2D)$, since a matrix vector multiplication with $O(N^2)$ has to be performed at each step.

The Lanczos implementation of the MSNWF computes the mean square error $\mathrm{MSE}^{(i)}$ at each step. In order to get a recursion formula for $\mathrm{MSE}^{(i)}$ in the CG implementation consider the first elements in Line 12 of Algorithm I and use this equation to replace $c_{\mathrm{first},1}^{(i)}$ in Line 14 of Algorithm I. This yields for $D \geq i \geq 1$

$$\mathrm{MSE}^{(i)} = \mathrm{MSE}^{(i-1)} - \eta_i\boldsymbol{g}_i^{\mathrm{H}}\boldsymbol{g}_i, \quad \mathrm{MSE}^{(0)} = \sigma_{d_0}^2. \tag{35}$$

Finally, summarizing Equations (34), (11), (35), (17), and (14) leads to a CG implementation of the MSNWF which is given by Algorithm III. Note that P. S. Chang and A. N. Willson, Jr., presented a similar algorithm to solve the Wiener-Hopf equation in [12]. However, we derived the CG algorithm from the MSNWF and in addition, our implementation computes the mean square error at each iteration step.

VI. Application to an EDGE System

In the following we consider an EDGE system with 8PSK modulation and *Laurent pulse shaping*. The Laurent impulse is a linearized GMSK impulse [13] which has a duration of five symbol times. Thus, we have severe intersymbol interference even without channel distortion. The symbol time $T = 3.69\,\mu$s and the two antennas of the mobile station (MS) receive the signal of a base station which propagates over Rayleigh multipath fading channels with a delay spread of $\tau_{\max} = 10\,\mu$s or three symbol times. We assume a constant channel during one burst with 148 symbols (excluding guard symbols). The CG based implementation of the MSNWF is used as a linear equalizer filter for the received signal at the MS. We sample two times during one symbol duration and take 20 samples of each antenna to build the space-time observation vector $\boldsymbol{x}_0[n]$, thus, its dimension $N = 40$.

Figure 2 shows the measered MSE using the CG based MSNWF for $D \in \{6, 8, 10\}$ steps compared to the MMSE

ALGORITHM III

CG MSNWF

$\boldsymbol{w}_0^{(0)} = \mathbf{0}$

$\boldsymbol{u}_1 = -\boldsymbol{g}_1 = \boldsymbol{r}_{\boldsymbol{x}_0,d_0}$

$l_1 = \boldsymbol{g}_1^{\mathrm{H}} \boldsymbol{g}_1$

$\mathrm{MSE}^{(0)} = \sigma_{d_0}^2$

for $i = 1$ to D **do**

$\boldsymbol{v} = \boldsymbol{R}_{\boldsymbol{x}_0} \boldsymbol{u}_i$

$\eta_i = l_i / \left(\boldsymbol{u}_i^{\mathrm{H}} \boldsymbol{v} \right)$

$\boldsymbol{w}_0^{(i)} = \boldsymbol{w}_0^{(i-1)} + \eta_i \boldsymbol{u}_i$

$\mathrm{MSE}^{(i)} = \mathrm{MSE}^{(i-1)} - \eta_i l_i$

$\boldsymbol{g}_{i+1} = \boldsymbol{g}_i + \eta_i \boldsymbol{v}$

$l_{i+1} = \boldsymbol{g}_{i+1}^{\mathrm{H}} \boldsymbol{g}_{i+1}$

$\psi_i = l_{i+1} / l_i$

$\boldsymbol{u}_{i+1} = -\boldsymbol{g}_{i+1} + \psi_i \boldsymbol{u}_i$

end for

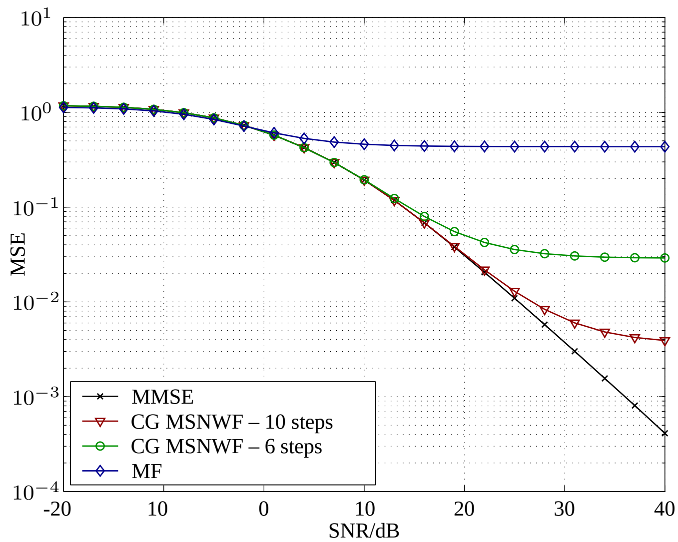

Fig. 2. Measured MSE for estimated statistic using CG MSNWF

equalizer or WF which corresponds to the MSNWF with $D = 40$ steps. The statistics are derived from the *least squares* estimation of the channel where we used the 26 training symbols of a burst. We observe that the MSNWF with $D = 10$ steps is very close to the MMSE equalizer even for high SNR values.

VII. Conclusions

In this paper we transformed the formulas of the Lanczos based implementation of the MSNWF to those of the CG algorithm and obtained a new implementation of MSNWF. Simulation results of an application to an EDGE system showed that despite the reduced computational complexity, the CG based MSNWF yields almost the same results as the MMSE equalizer and outperforms other methods of dimension reduction like the PC or CS method.

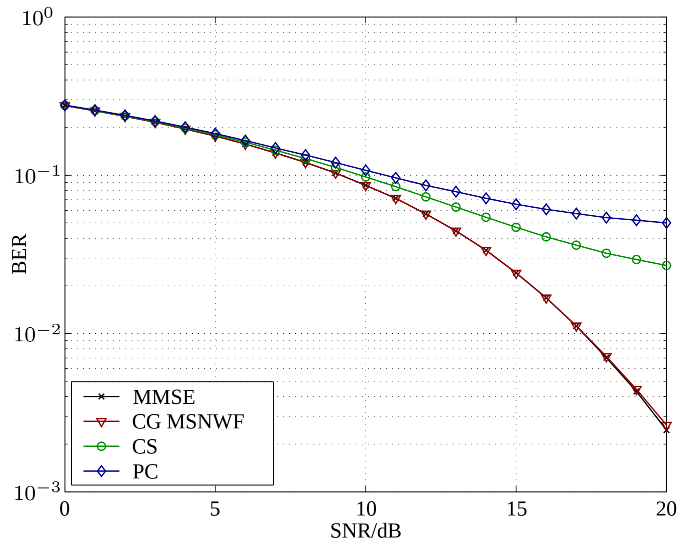

Fig. 3. BER for estimated statistic using different equalizer ($D = 10$)

References

[1] L. L. Scharf, *Statistical Signal Processing*, Addison-Wesley, 1991.

[2] H. Hotelling, "Analysis of a Complex of Statistical Variables into Principal Components," *Journal of Educational Psychology*, vol. 24, no. 6/7, pp. 417–441, 498–520, September/October 1933.

[3] J. S. Goldstein and I. S. Reed, "Subspace Selection for Partially Adaptive Sensor Array Processing," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 2, pp. 539–543, April 1997.

[4] J. S. Goldstein, I. S. Reed, and L. L. Scharf, "A Multistage Representation of the Wiener Filter Based on Orthogonal Projections," *IEEE Transactions on Information Theory*, vol. 44, no. 7, pp. 2943–2959, November 1998.

[5] M. L. Honig and W. Xiao, "Performance of Reduced-Rank Linear Interference Suppression for DS-CDMA," submitted to *IEEE Transactions on Information Theory*, December 1999.

[6] M. Joham and M. D. Zoltowski, "Interpretation of the Multi-Stage Nested Wiener Filter in the Krylov Subspace Framework," Tech. Rep. TUM-LNS-TR-00-6, Munich University of Technology, November 2000, Also: Technical Report TR-ECE-00-51, Purdue University.

[7] M. R. Hestenes and E. Stiefel, "Methods of Conjugate Gradients for Solving Linear Systems," *Journal of Research of the National Bureau of Standards*, vol. 49, no. 6, pp. 409–436, December 1952.

[8] L. N. Trefethen and D. Bau III, *Numerical Linear Algebra*, SIAM, 1997.

[9] Guido Dietl, "Conjugate Gradient Implementation of Multi-Stage Nested Wiener Filter for Reduced Dimension Processing," Diploma thesis, Munich University of Technology, May 2001.

[10] G. Golub and C. V. Loan, *Matrix Computations*, Johns Hopkins University Press, 1996.

[11] G. Dietl, M. D. Zoltowski, and M. Joham, "Recursive Reduced-Rank Adaptive Equalization for Wireless Communications," in *Proc. of SPIE*, April 2001, vol. 4395.

[12] P. S. Chang and A. N. Willson, Jr., "Analysis of Conjugate Gradient Algorithms for Adaptive Filtering," *IEEE Transactions on Signal Processing*, vol. 48, no. 2, pp. 409–418, February 2000.

[13] European Telecommunications Standards Institute, "Digital cellular telecommunications system (Phase 2+); Modulation," EN 300 959 V7.1.1, ETSI, http://www.etsi.org, June 2000.