LOW-COMPLEXITY EQUALIZERS—RANK VERSUS ORDER REDUCTION

Guido Dietl and Wolfgang Utschick

Institute for Circuit Theory and Signal Processing
Munich University of Technology, 80333 Munich, Germany
E-mail: `dietl@tum.de`

ABSTRACT

Reduced-rank approximations of *finite impulse response* equalizers in *Krylov* subspaces, e. g., the *conjugate gradient* algorithm, can be used to decrease computational complexity involved in calculating the filter coefficients. However, an alternative approach would be to reduce the order of the corresponding full-rank filter or to even combine rank and order reduction. In this paper, we compare both reduction methods based on (G, D)-charts where we analyze the *mean square error* of the reduced-rank equalizers on complexity isosets, i. e., for tuples of the filter length G and its rank D resulting in a certain number of *floating point operations*. The application of (G, D)-charts to a coded system with an iterative receiver (*turbo equalization*) reveals the superiority of rank reduction, especially, if one is interested in low-complexity implementations.

1. INTRODUCTION

The *Wiener Filter* (WF) [1] is the optimal linear equalizer based on the *minimum mean square error* criterion. Nevertheless, to compute its filter coefficients, one has to solve the *Wiener-Hopf* equation which involves a computational complexity of $\mathrm{O}(G^3)$ if G is the dimension of the system of linear equations. Reduced-rank approximations of the WF in either an *eigen* or *Krylov* subspace can be used to reduce this computational burden. In the sequel, we concentrate on Krylov methods since it was shown for several applications [2, 3, 4, 5] that they outperform eigen based approaches.

Goldstein et al. [3] introduced the rank *D Multi-Stage WF* (MSWF) which was later proven [4] to be a WF approximation in a D-dimensional Krylov subspace. Based on this fact, it was shown in [5] that the *Conjugate Gradient* (CG) algorithm when stopped after D iterations, can be used to implement the rank D MSWF very efficiently. Both the original rank D MSWF as well as its CG implementation has a computational complexity of $\mathrm{O}(DG^2)$, i. e., the filter order G affects the computational complexity quadratically whereas the rank D has only a linear influence. Therefore, both D and G can be reduced to obtain low-complexity equalizers as long as the performance meets the system requirements.

Firstly, we compare the exact number of complex *FLoating point OPerations* (FLOPs) required to compute the full-rank WF solution with the number of FLOPs needed for the CG implementation of the reduced-rank MSWF. Here, we implement the WF using the already very efficient *Cholesky* factorization [6] which exploits the fact that the Wiener-Hopf equation is a Hermitian system. The main contribution of this paper is the introduction of the (G, D)-charts to find the optimal combination of the filter length G and the rank D which results in the best performance for a given number of FLOPs. One observation will be that the CG implementation of the reduced-rank MSWF is only computationally cheaper than the Cholesky implementation of the WF if its rank is reduced below a certain fraction of the filter length G. Finally, the application of the reduced-rank equalizers to an iterative (turbo) receiver in a coded system shows that decreasing the rank leads to a better performance than reducing the filter order, especially, if the number of available FLOPs is very small.

The organization of the paper is as follows. Before deriving the low-complexity implementations with the required number of FLOPs in Section 3, we briefly review the WF in Section 2. Finally, Section 4 introduces the (G, D)-charts which are exemplarily presented in Section 6 for the communication system defined in Section 5.

Throughout the paper, vectors and matrices are denoted by lower and upper case bold letters, and random variables are written using *sans serif* font. The matrix $\boldsymbol{I}_n$ is the $n \times n$ identity matrix, $\boldsymbol{e}_\nu$ its ν-th column, $\mathbf{0}_{m\times n}$ the $m \times n$ zero matrix, and $\mathbf{0}_n$ the n-dimensional zero vector. The operation $\mathrm{E}\{\cdot\}$ denotes expectation, $(\cdot)^*$ conjugate complex, $(\cdot)^T$ transpose, $(\cdot)^H$ Hermitian, i. e., conjugate transpose, $\|\cdot\|_2$ the Euclidean norm, and $\lceil\cdot\rceil$ rounding to the next integer greater than the argument. The matrix $\boldsymbol{S}_{(\ell,M,N)} = [\mathbf{0}_{M\times\ell}, \boldsymbol{I}_M, \mathbf{0}_{M\times(N-\ell)}] \in \{0,1\}^{M\times(M+N)}$ is used for selection. The auto-correlation matrix of the stationary and ergodic random vector process $\boldsymbol{\mathsf{u}}[n] \in \mathbb{C}^m$ is $\boldsymbol{R}_{\boldsymbol{\mathsf{u}}} = \mathrm{E}\{\boldsymbol{\mathsf{u}}[n]\boldsymbol{\mathsf{u}}^H[n]\} \in \mathbb{C}^{m\times m}$, the cross-correlation vector between $\boldsymbol{\mathsf{u}}[n]$ and the stationary and ergodic random scalar process $\mathsf{v}[n] \in \mathbb{C}$ is $\boldsymbol{r}_{\boldsymbol{\mathsf{u}},\mathsf{v}}[\nu] = \mathrm{E}\{\boldsymbol{\mathsf{u}}[n]\mathsf{v}^*[n-\nu]\} \in \mathbb{C}^m$, and the variance of $\mathsf{v}[n]$ is σ_{v}^2. The entry in the i-th row and j-th column of a matrix $\boldsymbol{A}$ is denoted as $a_{i,j}$, the row vector in the i-th row from columns j to k of

$\boldsymbol{A}$ as $\boldsymbol{a}_{i,j:k}$, the i-th element of the vector $\boldsymbol{a}$ as a_i, and the subvector from row i to j of $\boldsymbol{a}$ as $\boldsymbol{a}_{i:j}$.

2. WIENER FILTER

The WF [1] $\boldsymbol{w} \in \mathbb{C}^G$ estimates the unknown signal $x[n] \in \mathbb{C}$ from the observation vector $\boldsymbol{y}[n] \in \mathbb{C}^G$ by minimizing the *Mean Square Error* (MSE) $\xi(\boldsymbol{\omega}) \in \mathbb{R}_{0,+}$ between $x[n-\nu]$ and its estimate $\hat{x}[n] = \boldsymbol{\omega}^H \boldsymbol{y}[n]$, i. e.,

$$\boldsymbol{w} = \underset{\boldsymbol{\omega}}{\operatorname{argmin}}\, \xi(\boldsymbol{\omega}), \quad \text{with} \tag{1}$$

$$\begin{aligned}\xi(\boldsymbol{\omega}) &= \mathrm{E}\left\{|x[n-\nu] - \hat{x}[n]|^2\right\} \\ &= \sigma_x^2 - \boldsymbol{\omega}^H \boldsymbol{r}_{\boldsymbol{y},x}[\nu] - \boldsymbol{r}_{\boldsymbol{y},x}^H[\nu]\boldsymbol{\omega} + \boldsymbol{\omega}^H \boldsymbol{R}_{\boldsymbol{y}} \boldsymbol{\omega},\end{aligned} \tag{2}$$

where ν is the latency time. The optimization leads to the Wiener-Hopf equation

$$\boldsymbol{R}_{\boldsymbol{y}} \boldsymbol{w} = \boldsymbol{r}_{\boldsymbol{y},x}[\nu] \in \mathbb{C}^G, \tag{3}$$

which has to be solved according to $\boldsymbol{w}$. If the second order statistics are given, solving Equation (3) involves the highest computational complexity in calculating the WF coefficients comprised in $\boldsymbol{w}$.

3. LOW-COMPLEXITY IMPLEMENTATIONS

In this section, we present the full-rank WF implementation based on the Cholesky factorization and the CG implementation of the reduced-rank MSWF. Additionally, we investigate the number of FLOPs needed for their implementation. Here, a FLOP is either an addition, subtraction, multiplication, division, or root extraction, no matter if operating on real or complex values.

3.1. WF Based on Cholesky Factorization

Since the auto-correlation matrix $\boldsymbol{R}_{\boldsymbol{y}}$ is Hermitian, the Cholesky factorization [6] can be used to solve the Wiener-Hopf equation $\boldsymbol{R}_{\boldsymbol{y}} \boldsymbol{w} = \boldsymbol{r}_{\boldsymbol{y},x}[\nu]$. The procedure will be as follows. With $\boldsymbol{R}_{\boldsymbol{y}} = \boldsymbol{U}^H \boldsymbol{U} \in \mathbb{C}^{G \times G}$ where $\boldsymbol{U} \in \mathbb{C}^{G \times G}$ is an upper triangular matrix, firstly, the system $\boldsymbol{U}^H \boldsymbol{v} = \boldsymbol{r}_{\boldsymbol{y},x}[\nu]$ of linear equations is solved with respect to $\boldsymbol{v} \in \mathbb{C}^G$ (*forward substitution*), and subsequently, the system $\boldsymbol{U}\boldsymbol{w} = \boldsymbol{v}$ is solved according to $\boldsymbol{w}$ (*backward substitution*).

Algorithm 1 depicts the *outer product* version [6] of the Cholesky factorization where the number in brackets are the FLOPs required to calculate the corresponding line of the algorithm. After summing up, the number of FLOPs required to perform Algorithm 1 is

$$\zeta_{\mathrm{C}}(G) = \frac{G^3}{3} + \frac{G^2}{2} + \frac{G}{6}. \tag{4}$$

Algorithm 1 Cholesky factorization

```
Choose $\boldsymbol{U}$ as the upper triangular part[1] of $\boldsymbol{R}_{\boldsymbol{y}}$
for $i = 1, 2, \ldots, G$ do
    $u_{i,i} = \sqrt{u_{i,i}}$                                                    $\langle 1 \rangle$
    $\boldsymbol{u}_{i,i+1:G} \leftarrow \boldsymbol{u}_{i,i+1:G} \,/\, u_{i,i}$                     $\langle G - i \rangle$
    for $j = i+1, i+2, \ldots, G \,\wedge\, j \leq G$ do
        $\boldsymbol{u}_{j,j:G} \leftarrow \boldsymbol{u}_{j,j:G} - \boldsymbol{u}_{i,j:G} u_{i,j}^*$         $\langle 2(G - j + 1) \rangle$
    end for
end for
```

The solution of the system $\boldsymbol{U}^H \boldsymbol{v} = \boldsymbol{L}\boldsymbol{v} = \boldsymbol{r}_{\boldsymbol{y},x}[\nu] = \boldsymbol{b}$ of linear equations is obtained based on the forward substitution given by Algorithm 2. Finally, $\boldsymbol{U}\boldsymbol{w} = \boldsymbol{v}$ is solved based on the backward substitution described in Algorithm 3. Note that the forward as well as the backward substitution requires

$$\zeta_{\mathrm{FS}}(G) = \zeta_{\mathrm{BS}}(G) = G^2 \quad \text{FLOPs}. \tag{5}$$

Thus, the total computational complexity of the WF implementation based on the Cholesky factorization computes finally as

$$\zeta_{\mathrm{WF}}(G) = \zeta_{\mathrm{C}}(G) + \zeta_{\mathrm{FS}}(G) + \zeta_{\mathrm{BS}}(G) = \frac{G^3}{3} + \frac{5G^2}{2} + \frac{G}{6}. \tag{6}$$

Algorithm 2 Forward substitution

```
$v_1 \leftarrow b_1 / l_{1,1},$                                               $\langle 1 \rangle$
for $i = 2, 3, \ldots, G$ do
    $v_i \leftarrow (b_i - \boldsymbol{l}_{i,1:i-1} \boldsymbol{v}_{1:i-1}) \,/\, l_{i,i}$                  $\langle 2i - 1 \rangle$
end for
```

Algorithm 3 Backward substitution

```
$w_G \leftarrow v_G / u_{G,G}$                                              $\langle 1 \rangle$
for $i = G-1, G-2, \ldots, 1$ do
    $w_i \leftarrow (v_i - \boldsymbol{u}_{i,i+1:G} \boldsymbol{w}_{i+1:G}) \,/\, u_{i,i}$      $\langle 2(G - i) + 1 \rangle$
end for
```

3.2. Reduced-Rank MSWF Based on CG Algorithm

In [5], it was shown that the rank D MSWF $\boldsymbol{w}_D$ as an approximation of the WF in the D-dimensional Krylov subspace [4], can be implemented using the CG algorithm [7, 6] which is stopped after D iterations.The resulting procedure is depicted in Algorithm 4 where again, the number of FLOPs per line are given in brackets.

If we take into account that Lines 8 to 11 have to be no longer executed at the last iteration step, i. e., for $i = D$, the execution of Algorithm 4 requires

$$\zeta_{\mathrm{CG}}(G, D) = 2DG^2 + (9D - 4)G - 1 \quad \text{FLOPs}. \tag{7}$$

[1] Here, the upper triangular part of a matrix includes its main diagonal.

Algorithm 4 Conjugate Gradient (CG) algorithm

```
$\boldsymbol{w}_0 = \boldsymbol{0}_G$
$\boldsymbol{p}_1 = -\boldsymbol{r}_1 = \boldsymbol{r}_{\boldsymbol{y},x}[\nu]$
$\varrho_1 = \boldsymbol{r}_1^H \boldsymbol{r}_1$    $\langle 2G-1 \rangle$
for $i = 1, 2, \ldots, D$ do
    $\boldsymbol{q} = \boldsymbol{R}_{\boldsymbol{y}} \boldsymbol{p}_i$    $\langle G(2G-1) \rangle$
    $\gamma_i = \varrho_i \,/\, (\boldsymbol{p}_i^H \boldsymbol{q})$    $\langle 2G \rangle$
    $\boldsymbol{w}_i = \boldsymbol{w}_{i-1} + \gamma_i \boldsymbol{p}_i$    $\langle 2G \rangle$
    $\boldsymbol{r}_{i+1} = \boldsymbol{r}_i + \gamma_i \boldsymbol{q}$    $\langle 2G \rangle$
    $\varrho_{i+1} = \boldsymbol{r}_{i+1}^H \boldsymbol{r}_{i+1}$    $\langle 2G-1 \rangle$
    $\delta_i = \varrho_{i+1} \,/\, \varrho_i$    $\langle 1 \rangle$
    $\boldsymbol{p}_{i+1} = -\boldsymbol{r}_{i+1} + \delta_i \boldsymbol{p}_i$    $\langle 2G \rangle$
end for
```

Note that Algorithm 4 can be also used to compute the WF $\boldsymbol{w}$, i. e., the exact solution of the Wiener-Hopf equation $\boldsymbol{R}_{\boldsymbol{y}} \boldsymbol{w} = \boldsymbol{r}_{\boldsymbol{y},x}[\nu]$, if the rank is chosen to be equal to the filter length, i. e., $D = G$. However, in this case, its computational complexity is $\zeta_{\mathrm{CG}}(G, G) = 2G^3 + 9G^2 - 4G - 1$ which is greater than $\zeta_{\mathrm{WF}}(G)$ for every $G \in \mathbb{N}$. In fact, D has to be reduced to

$$D < \left\lceil \frac{2G^3 + 15G^2 + 25G + 6}{6G(2G+9)} \right\rceil, \tag{8}$$

to get an implementation with a smaller complexity than the WF realized based on the Cholesky factorization. For $G \to \infty$, this means that D has to be smaller than $\lceil G/6 \rceil$. In Section 6, we compare the performance of the full-rank WF and the CG implementation of the reduced-rank MSWF when they have the same computational complexity. Here, we vary not only the rank D but also the filter length G.

4. (G, D)-CHARTS

In this section, we describe a new chart which can be used to find out which combination of the filter length G and the rank D resulting in a given computational complexity, yields the best performance according to a chosen criterion. Remember from the previous subsection that the CG algorithm for $D = G$ is computationally more expensive than the WF based on the Cholesky factorization (cf. Subsection 3.1). If we use the CG based WF approximation for $D < G$ but the Cholesky based WF for $D = G$, calculating the rank D MSWF for an arbitrary $D \in \{1, 2, \ldots, G\}$, involves the computational complexity

$$\zeta(G, D) = \begin{cases} \zeta_{\mathrm{CG}}(G, D), & G > D, \\ \zeta_{\mathrm{WF}}(G), & G = D. \end{cases} \tag{9}$$

Next, we define the β-ζ_0-*complexity isoset*[1] $\mathcal{I}_{\zeta_0}^{\beta}$ as the set of (G, D)-tuples leading to the computational complexity ζ_0 with a tolerance of $\pm\beta\zeta_0$, i. e.,

$$\mathcal{I}_{\zeta_0}^{\beta} = \left\{ (G, D) \in \mathbb{N} \times \{1, 2, \ldots, G\} : \left| \frac{\zeta(G, D) - \zeta_0}{\zeta_0} \right| \leq \beta \right\}.$$

The (G, D)-chart depicts the β-ζ_0-*complexity isoset* in the G-D-plane for a given computational complexity ζ_0. Finally, the performance for each element of the β-ζ_0-*complexity isoset* is investigated in order to find the best combination of the filter length G and the rank D with respect to the chosen performance criterion and for a given computational complexity ζ_0. In Section 6, we present the (G, D)-charts for the communication system which will be introduced in the next section.

5. SYSTEM MODEL

We consider a system where the block $\boldsymbol{b}$ of binary data bits is convolutionally encoded with rate $1/2$, interleaved, and mapped to the block $\boldsymbol{x}$ of QPSK symbols $x[n] \in \mathbb{C}$ based on a *Gray code*, before they are transmitted over the channel as described in Subsection 5.1.

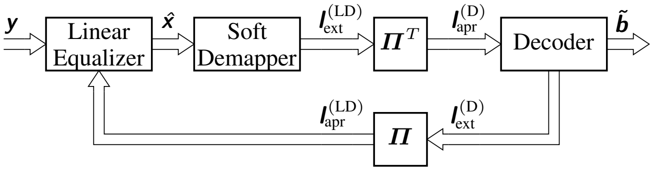
$\boldsymbol{y}$
Linear Equalizer
$\hat{\boldsymbol{x}}$
Soft Demapper
$\boldsymbol{l}_{\mathrm{ext}}^{(\mathrm{LD})}$
$\boldsymbol{\Pi}^T$
$\boldsymbol{l}_{\mathrm{apr}}^{(\mathrm{D})}$
Decoder
$\tilde{\boldsymbol{b}}$
$\boldsymbol{l}_{\mathrm{apr}}^{(\mathrm{LD})}$
$\boldsymbol{\Pi}$
$\boldsymbol{l}_{\mathrm{ext}}^{(\mathrm{D})}$

Figure 1. Iterative Receiver (Turbo Equalization)

Figure 1 depicts the iterative receiver. After equalization as described in the previous sections, *soft demapping*, and deinterleaving ($\boldsymbol{\Pi}^T$), the BCJR algorithm [8] as a *soft-input soft-output Maximum A Posteriori* (MAP) decoder provides the block $\tilde{\boldsymbol{b}}$ of detected data bits. In the case of iterative (*turbo*) detection, the soft information at the output of the MAP decoder, i. e., the block $\boldsymbol{l}_{\mathrm{ext}}^{(\mathrm{D})}$ of *Log-Likelihood Ratios* (LLRs), is fed back as soft *a priori* information ($\boldsymbol{l}_{\mathrm{apr}}^{(\mathrm{LD})}$) to the equalizer whose design has to be slightly adjusted according to [9, 10, 11] which is not presented in this paper due to space limitations. In Section 6, we will see that the *a priori* information from the decoder can improve drastically the equalizer performance, i. e., decrease the MSE which is estimated according to the method described in [12], at the next turbo iteration.

5.1. Channel Model Based WF Design

The symbol stream $x[n]$ with variance σ_x^2 is transmitted over the *Finite Impulse Response* (FIR) channel $h[n] \in \mathbb{C}$ of order L and perturbed by *Additive White Gaussian Noise* (AWGN) $\eta[n]$ with the complex-valued normal distribution $\mathcal{N}_{\mathrm{c}}(0, \sigma_\eta^2)$,

[1] Here, we extend the notation *isoset* by introducing a certain tolerance level.

such that the received signal reads as

$$y[n] = h[n]*x[n]+\eta[n] = \sum_{\ell=0}^{L} h[\ell]x[n-\ell]+\eta[n] \in \mathbb{C}. \quad (10)$$

In order to end up with the WF $\boldsymbol{w}$ with FIR structure of order $G-1$, we define the observation vector as

$$\boldsymbol{y}[n] = [y[n], y[n-1], \ldots, y[n-G+1]]^T \in \mathbb{C}^G. \quad (11)$$

With the convolutional channel matrix

$$\boldsymbol{H} = \sum_{\ell=0}^{L} h[\ell]\boldsymbol{S}_{(\ell,G,L)} \in \mathbb{C}^{G\times(G+L)}, \quad (12)$$

the transmit vector $\boldsymbol{x}[n] \in \mathbb{C}^{G+L}$ and the noise vector $\boldsymbol{\eta}[n] \in \mathbb{C}^G$ defined as

$$\boldsymbol{x}[n] = [x[n], x[n-1], \ldots, x[n-G-L+1]]^T, \quad (13)$$
$$\boldsymbol{\eta}[n] = [\eta[n], \eta[n-1], \ldots, \eta[n-G+1]]^T, \quad (14)$$

the transmission over the channel $h[n]$ can be written as the matrix-vector model

$$\boldsymbol{y}[n] = \boldsymbol{H}\boldsymbol{x}[n] + \boldsymbol{\eta}[n]. \quad (15)$$

Thus, the statistics needed to solve the Wiener-Hopf equation $\boldsymbol{R}_{\boldsymbol{y}}\boldsymbol{w} = \boldsymbol{r}_{\boldsymbol{y},x}[\nu]$ compute as

$$\boldsymbol{R}_{\boldsymbol{y}} = \sigma_x^2 \boldsymbol{H}\boldsymbol{H}^H + \sigma_\eta^2 \boldsymbol{I}_G, \quad (16)$$
$$\boldsymbol{r}_{\boldsymbol{y},x}[\nu] = \sigma_x^2 \boldsymbol{H}\boldsymbol{e}_{\nu+1}. \quad (17)$$

5.2. Estimation of Second Order Statistics

In this subsection, we review the *Least Squares* (LS) method [1] to estimate the second order statistics based on the channel model described in Subsection 5.1. First, we assume the transmission of $P+L$ training symbols summarized in the matrix

$$\boldsymbol{X}_\text{p} = [\boldsymbol{x}_\text{p}[0], \boldsymbol{x}_\text{p}[1], \ldots \boldsymbol{x}_\text{p}[L]] \in \mathbb{C}^{P\times(L+1)}, \quad (18)$$
$$\boldsymbol{x}_\text{p} = [x[n], x[n+1], \ldots, x[n+P-1]]^T \in \mathbb{C}^P. \quad (19)$$

With the additional definitions of the vectors

$$\boldsymbol{y}_\text{p} = [y[0], y[1], \ldots, y[P-1]]^T \in \mathbb{C}^P, \quad (20)$$
$$\boldsymbol{\eta}_\text{p} = [\eta[0], \eta[1], \ldots, \eta[P-1]]^T \in \mathbb{C}^P, \quad \text{and} \quad (21)$$
$$\boldsymbol{h} = [h[0], h[1], \ldots, h[L]] \in \mathbb{C}^{L+1}, \quad (22)$$

the transmission of the training symbols over the channel $h[n]$ can be written as (cf. Equation 10)

$$\boldsymbol{y}_\text{p} = \boldsymbol{X}_\text{p}\boldsymbol{h} + \boldsymbol{\eta}_\text{p}, \quad (23)$$

and the LS estimate of the channel vector $\boldsymbol{h}$ computes as

$$\hat{\boldsymbol{h}} = \boldsymbol{X}_\text{p}^\dagger \boldsymbol{y}_\text{p} = \left(\boldsymbol{X}_\text{p}^H \boldsymbol{X}_\text{p}\right)^{-1} \boldsymbol{X}_\text{p}^H \boldsymbol{y}_\text{p}. \quad (24)$$

The channel estimate $\hat{\boldsymbol{h}}$ is then used to create an estimate of the channel matrix $\boldsymbol{H}$ as defined in Equation (12) which is finally taken to compute the estimates of the second order statistics according to Equations (16) and (17). Note that the computational complexity of the LS channel estimation is not taken into account in the considerations of the next section since it is negligible compared to the computational complexity of the WF calculation. This is due to the fact that the computational intense pseudo inversion $\boldsymbol{X}_\text{p}^\dagger$ in Equation (24) has to be performed only once if the training symbols do not change.

6. SIMULATION RESULTS

All following simulations are averaged over several channel realizations with order $L = 29$ and a uniform power delay profile, i. e., $h[n] \sim \mathcal{N}_\text{c}(0, 1/(L+1))$, $n \in \{0, 1, \ldots, L\}$. The *Signal-to-Noise Ratio* (SNR) has been set to 5 dB.

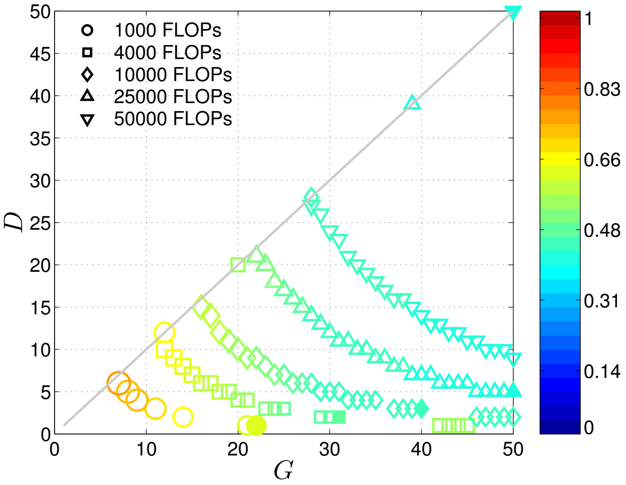

Figure 2. (G, D)-chart for perfect CSI, no turbo iteration

Figures 2 and 3 depict the simulation results for perfect *Channel State Information* (CSI). The tuples of G and D plotted with the same marker correspond to the 10%-ζ_0-complexity isoset $\mathcal{I}_{\zeta_0}^{10\%}$ with the computational complexity ζ_0 as given in the legend. The color as well as the size[2] of a marker determines its MSE which we have chosen as the performance criterion. The filled marker denotes the combination of G and D with the optimal performance for a given ζ_0. It can be seen that the reduced-rank method yields the best performance if we are interested in low-complexity receivers. Only for $\zeta_0 = 50000$ FLOPs, the full-rank WF achieves the lowest MSE. Moreover, if we perform turbo iterations, i. e., we use the soft information from the decoder to adjust the statistics in Equations (16) and (17), and therefore, the reduced-rank WF (see [9, 10, 11]) used to repeat the equalization of the received

[2] The marker size depends linearly on the MSE.

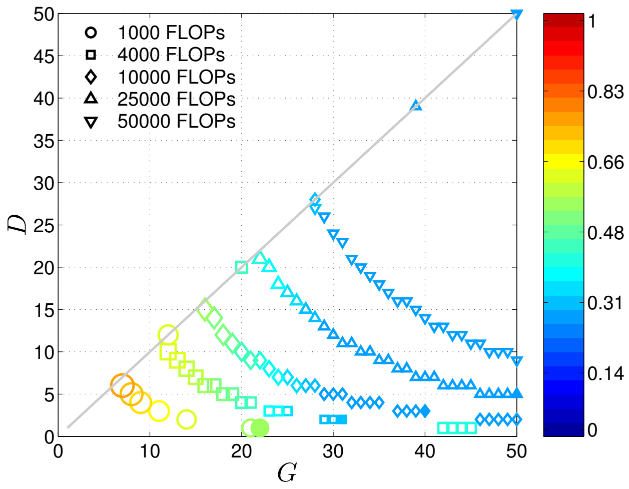

Figure 3. (G, D)-chart for perfect CSI, 2 turbo iterations

signal at the next turbo iteration, we observe that the MSE is drastically increased. In other words, the computational complexity of the linear equalizer can be further reduced without changing its performance.

Figure 4 presents the (G, D)-chart for a system where the channel is estimated using the LS method with $P + L = 100$ pilot symbols. Although the MSE is greater than in the case of perfect CSI, reducing the rank D should again be preferred to a WF with decreased filter order.

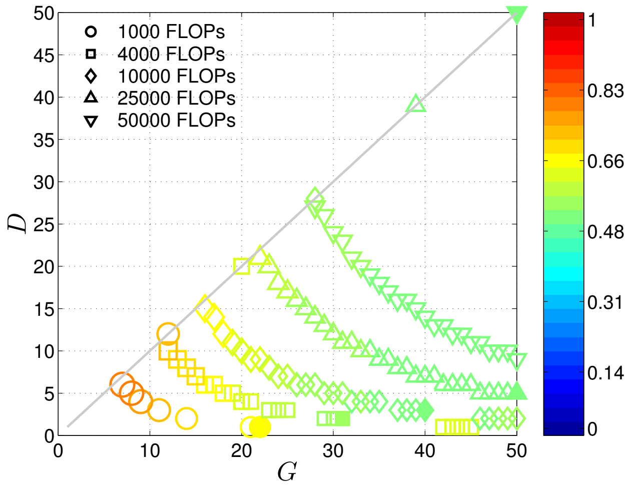

Figure 4. (G, D)-chart for estimated CSI, no turbo iteration

7. CONCLUSIONS

In this paper, we introduced (G, D)-charts in order to analyze the MSE performance of the rank D MSWF for so-called complexity isosets, i. e., combinations of the filter length G and the rank D which result in a given computational complexity. The analysis of these charts has shown that a reduced-rank MSWF performs better than a full-rank WF with the same computational complexity, i. e., with a decreased filter order. The superiority of the reduced-rank MSWF is especially dominant in systems based on turbo equalization.

Acknowledgment

The authors would like to thank Georg Zeitler for his support in simulating the coded system with iterative detection.

8. REFERENCES

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

[2] S. Moshavi, E. G. Kanterakis, and D. L. Schilling, "Multistage Linear Receivers for DS-CDMA Systems," *International Journal of Wireless Information Networks*, vol. 3, no. 1, pp. 1–17, 1996.

[3] 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.

[4] M. L. Honig and W. Xiao, "Performance of Reduced-Rank Linear Interference Suppression," *IEEE Transactions on Information Theory*, vol. 47, no. 5, pp. 1928–1946, July 2001.

[5] G. Dietl, M. D. Zoltowski, and M. Joham, "Reduced-Rank Equalization for EDGE Via Conjugate Gradient Implementation of Multi-Stage Nested Wiener Filter," in *Proceedings of the 2001 IEEE 54th Vehicular Technology Conference (VTC 2001-Fall)*, October 2001, pp. 1912–1916.

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

[7] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, 2003.

[8] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate," *IEEE Transactions on Information Theory*, vol. IT-20, no. 2, pp. 284–287, March 1974.

[9] M. Tüchler, A. Singer, and R. Koetter, "Minimum Mean Squared Error Equalization Using A-priori Information," *IEEE Transactions on Signal Processing*, vol. 50, no. 3, pp. 673–683, March 2002.

[10] G. Dietl, C. Mensing, and W. Utschick, "Iterative Detection Based on Reduced-Rank Equalization," in *Proceedings of the 2004 IEEE 60th Vehicular Technology Conference (VTC 2004-Fall)*, September 2004, vol. 3, pp. 1533–1537.

[11] G. Dietl and W. Utschick, "Complexity Reduction of Iterative Receivers Using Low-Rank Equalization," *Submitted to IEEE Transactions on Signal Processing*, 2005.

[12] G. Zeitler, G. Dietl, and W. Utschick, "MSE Comparison of Reduced-Rank Equalizers for Iterative Multi-User Receivers Based on EXIT Charts," *Accepted for presentation at the 4th International Symposium on Turbo Codes*, April 2006.