Efficient Algorithms Relating OFDM Channel Parameters to Filter Weights in Single-Carrier Systems with Frequency Domain Equalization

Christian Vincent Sinn[1], Frank Hanschmann, Jürgen Götze
University of Dortmund,
Information Processing Lab
D-44221 Dortmund, Germany
{vincent.sinn, juergen.goetze}@uni-dortmund.de

***Abstract*— In high data rate transmissions over wireless channels with long delay spreads inter symbol interference is a major limiting factor. Transmission systems that cancel the interference efficiently are OFDM, cyclic prefix based single carrier (CP-SC) and single carrier with frequency domain equalization (SC-FDE). Although the systems show significant performance differences in terms of necessity of guard periods, peak to average power ratio, performance in fading conditions, and others the algebraic structures of the receiver processing are very similar: FFT → Multiplications → IFFT. OFDM and CP-SC on the one hand and SC-FDE on the other hand differ mainly in the values of the multipliers. Here efficient algorithms are presented that compute one system parameters from the others. The algorithms are assessed in terms of computational complexity and performance. They can be applied during transmission or training mode. Fast switching between the different systems is feasible. Therefore a multi-mode receiver can be adapted to a particular transmission situation.**

***Index Terms*— OFDM, single-carrier, frequency domain equalization, zero forcing algorithm, Schur algorithm.**

I. Introduction

One major problem in broadband transmissions over wireless channels with long delay spreads is the efficient cancellation of inter symbol interference (ISI). The applied signal processing algorithms shall be as efficient as possible to enable high data rate transmissions and the development of mobile terminals with low power consumption and low production costs.

Starting from a mathematical model describing a SISO (single-input single-output) transmission over a wireless channel different transmission systems that meet the requirement of low complexity are derived. These are OFDM and cyclic prefix based single carrier (CP-SC). Single carrier with frequency domain equalization (SC-FDE) represents a further low complexity system with similar receiver structures. If the wireless channel is assumed to be time invariant during one burst the received vector $\mathbf{x}$ can be calculated by convolving the mobile vector channel $\mathbf{h}$ of length L with the transmitted data symbols $\mathbf{d}_{burst}$ of length K and by adding the noise vector $\mathbf{n}$. The time discrete and synchronous model can be expressed in a system of equations:

$$\mathbf{x} = \mathbf{H}\mathbf{d}_{burst} + \mathbf{n}, \quad \mathbf{H} \in \mathbb{C}^{(K-L-1)\times K}.$$

[1]This work was funded by Siemens AG, ICM, Munich, Germany.

The convolution matrix $\mathbf{H}$ that describes the channel's influence is band- and Toeplitz-structured. The data vector $\mathbf{d}_{burst}$ can be detected by solving the system of equations. If all channel parameters $h_1, h_2, \cdots$, and h_L are zero data can not be detected. If exactly one parameter is not zero the system is solvable. Usually more than one parameter is unequal to zero. Then the system is overdetermined, which implies that it does not have a solution. However, for practical systems only the last two cases need to be taken into account. Then the least squares method represents one suitable criteria to calculate estimates $\hat{\mathbf{d}}_{LS}$ of $\mathbf{d}_{burst}$:

$$\min_{\hat{\mathbf{d}}_{LS}} \|\mathbf{x} - \mathbf{H}\hat{\mathbf{d}}_{LS}\|_2 \quad \rightarrow \quad \hat{\mathbf{d}}_{LS} = \underbrace{(\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H}_{\mathbf{H}^\dagger}\mathbf{x}. \tag{1}$$

Multiplying the received vector $\mathbf{x}$ by the pseudo inverse $\mathbf{H}^\dagger$ yields the LS-estimates $\hat{\mathbf{d}}_{LS}$ of the data $\mathbf{d}_{burst}$. Although there are matrix factorization techniques, e.g., the Cholesky decomposition, and FFT based approximation techniques [9] the signal processing complexity to calculate the estimates $\hat{\mathbf{d}}_{LS}$ is high according to the huge size of $\mathbf{H}$ if large values of K are assumed. In a scenario of high data rate transmissions over frequency selective channels the above method to cancel ISI is computational demanding and thus unsuited for practical systems. To enable efficient strategies to cancel ISI a guard period, which need to be at least as long as the channel impulse response, can be frequently inserted into the data vector $\mathbf{d}_{burst}$ so that each data block $\mathbf{d} \in \mathbb{C}^J$ $(J \ll K)$ can be processed independently. This concept of data blocking enables a lower processing complexity and lower storage requirements as the sizes of the processing matrices corresponding to one data block $\mathbf{d}$ are smaller than the one corresponding to the whole data vector $\mathbf{d}_{burst}$. Further simplification is achieved if the guard period between subsequent blocks consists of a cyclic prefix (CP) and the interfering parts between two subsequent blocks are discarded at the receiver. Such a scenario can be described by the following model [8], [11]:

$$\mathbf{x} = \tilde{\mathbf{H}}\mathbf{d} + \mathbf{n}, \quad \tilde{\mathbf{H}} \in \mathbb{C}^{J\times J}.$$

As subsequent blocks do not interfere with each other only one data block $\mathbf{d}$ is considered. The channel matrix $\tilde{\mathbf{H}}$ is cyclic and square. On contrast to the system mentioned previously this one has either exactly one solution or infinitely many, since it can be underdetermined. This is the case if one or more eigenvalues of the cyclic channel matrix are zero which reflects that the channel suffers from deep fading. However, the extension to a cyclic matrix has a major impact on the system's complexity, as the LS-solution, if it exists, is given by inverting the cyclic channel matrix, which is computational inexpensive since it can be performed by one FFT, one IFFT and one division per data symbol:

$$\hat{\mathbf{d}}_{LS} = \tilde{\mathbf{H}}^{-1}\mathbf{x} = \mathbf{F}^{-1}\mathbf{D}^{-1}\mathbf{F}\mathbf{x}.$$

This simple processing is based on the fact that the eigenvalue decomposition (EVD) of any cyclic matrix is given by:

$$\tilde{\mathbf{H}} = \mathbf{F}^{-1}\mathbf{D}\mathbf{F}, \quad \mathbf{D} = \mathrm{diag}(\mathbf{F}\tilde{\mathbf{H}}(:,1)).$$

The cyclic extension of the channel matrix is responsible for the low complexity of the matrix inversion as it enables a simple computation of the EVD. However, the inverse might not exist.

Based on the eigenvalue decomposition of cyclic matrices two transmission systems with equal overall complexity can be derived [8], [11]: OFDM and CP-SC (see figure 1). Beside

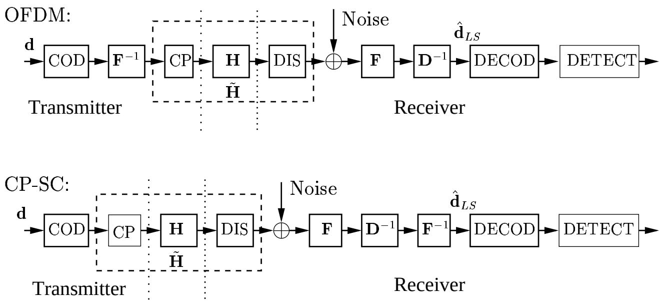

Fig. 1. OFDM and CP-SC: Systems based on inverting a cyclic channel matrix.

the differences of the systems in terms of adaptive modulation, noise enhancement, peak to average ratio, transmitter and receiver complexity, suitability for up- and downlink scenarios, and BER performance there are some similarities [1], [5], [8], [11]. Both systems are based on inverting a cyclic channel matrix $\tilde{\mathbf{H}}$, which is an efficient method to cancel ISI. Therefore both systems are suited for high data rate transmissions over wireless channels with long delay spreads [1], [11]. Major disadvantages of both systems are the necessity of a guard period that drastically reduces the information data rate and the performance loss if the channel suffers from deep fading, which means that $\tilde{\mathbf{H}}^{-1}$ does not exist ($\tilde{\mathbf{H}}$ has zero valued eigenvalues). Then extra coding is enforced.

To circumvent the mentioned disadvantages a SC-FDE might be chosen. The SC-FDE system is depicted in figure 2. Amazingly the processing structure of the frequency domain

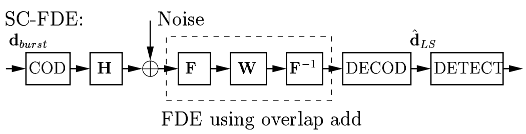

Fig. 2. SC-FDE: Single carrier with frequency domain equalization.

filter is similar to the receiver structure of CP-SC: FFT - diagonal matrix - IFFT. But instead of calculating the inverse cyclic channel matrix the channel is equalized by performing FIR filtering [3]. Main advantages are the assured stability of this filter, even if there are zeros on the FFT grid of the channel transfer function (zero valued eigenvalues) and that the insertion of a guard period at the transmitter becomes dispensable. Although coding is desired it is not necessary to combat frequency selectivity as it is in non-adaptive OFDM. On contrast to CP based systems guard periods can be virtually introduced at the receiver by using the overlap add or save method [7]. The reconstruction of the CP in CP based systems would be computational demanding [4]. Unfortunately the FIR filter equalizes the channel only approximately and causes a small increase in computational complexity. Due to the application of the overlap-add method $J - M$ symbols (J: size of window to which the FFT is applied, M: filter order) are estimated after processing one data block whereas J symbols are estimated in CP based systems. During transmission mode this will result in an increase of computational complexity by a factor of $J/(J - M)$. Therefore J should be chosen significantly larger than M to justify the assumption of about equal complexity of the different systems. The receiver's processing structures in CP-SC and SC-FDE systems differ in the values of the diagonal matrices $\mathbf{D}^{-1}$ and $\mathbf{W}$ and the window to which the FFT is applied. However, from a system point of view there are significant differences, e.g., the necessity of a guard period and performance in deep fading conditions. In this context the question occurs if we can switch from the $\mathbf{D}$ matrix to the $\mathbf{W}$ matrix efficiently, so that receivers belonging to different transmission systems can be implemented by using similar hardware. The switching can be performed during training and transmission mode, respectively.

In this paper algorithms are presented that perform the switching efficiently. They are based on performing the zero forcing (ZF) algorithm that relates time domain channel parameters to time domain filter weights. Three different approaches are used to perform the zero forcing algorithm. One uses a cyclic extension, another the QR decomposition of convolution matrices, a third one a fast inverse QR factorization [6]. A Schur type algorithm is incorporated into the QR based approaches. The algorithms differ in their storage requirements, total number of multiplications, performance in fading conditions, and suitability for parallel implementation. The complexity can be calculated in dependence of the FIR filter order M, the channel length L, and a parameter of free choice in the ZF algorithm, that is the position p of the one in the $\mathbf{p}$ vector (see next chapter for further explanation). The position of the one in the $\mathbf{p}$ vector has an impact on the quality of the estimates and the complexity such that there is a trade off between estimation accuracy and complexity.

In the next chapter different techniques to perform the switching are presented. Chapter 3 compares and contrasts the algorithms in terms of computational complexity and performance. After proposing applications of the algorithms some conclusions are drawn in chapter 4.

II. Efficient computation of the zero forcing algorithm

The zero forcing algorithm represents a method to compute either the FIR filter taps from the channel impulse response or the channel impulse response from the filter taps. The convolution of the channel $\mathbf{h}$ and the FIR Filter $\mathbf{g}$ shall result in a vector $\mathbf{p}$ containing a single one at an arbitrary position. The system of equations describing this scenario is either overdetermined or has exactly one solution. If it is overdetermined an error vector $\delta(\mathbf{p})$ is added to the $\mathbf{p}$ vector so that an approximated solution can be found be minimizing the error's energy (LS-criterion):

$$\mathbf{H}\mathbf{g} = \mathbf{p} + \delta(\mathbf{p}), \quad \mathbf{g} \in \mathbb{C}^{M}, \quad \mathbf{H} \in \mathbb{C}^{(M+L-1)\times M} \tag{2}$$
$$\min_{\hat{\mathbf{g}}_{LS}} \|\mathbf{H}\hat{\mathbf{g}}_{LS} - \mathbf{p}\|_2 \quad \rightarrow \quad \hat{\mathbf{g}}_{LS} = \underbrace{(\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H}_{\mathbf{H}^\dagger}\mathbf{p}.$$

Notice that in this case the size of the channel matrix is determined by the order of the equalization filter M and the length of the channel impulse response L. If we assume $M, L \ll K$ this $\mathbf{H}$ matrix is significantly smaller than the one in equation 1. Therefore the computational complexity to estimate filter weights and performing frequency domain FIR filtering is significantly lower than a direct estimation of data symbols as it is done in equation 1. On contrast to the inverse of a cyclic channel matrix the pseudo inverse $\mathbf{H}^\dagger$ does always exist. Thus the filter taps $\hat{\mathbf{g}}_{LS}$ can be computed independently of the channel parameters' values (exception: all channel parameters are zero). Assuming that the FIR filter taps are given, e.g., after performing an LMS algorithm, the channel parameters can be calculated in a similar fashion:

$$\min_{\hat{\mathbf{h}}_{LS}} \|\mathbf{G}\hat{\mathbf{h}}_{LS} - \mathbf{p}\|_2 \quad \rightarrow \quad \hat{\mathbf{h}}_{LS} = \underbrace{(\mathbf{G}^H\mathbf{G})^{-1}\mathbf{G}^H}_{\mathbf{G}^\dagger}\mathbf{p}.$$

After performing an FFT of a zero padded $\mathbf{h}$ or $\mathbf{g}$ the parameters are transformed into the diagonal matrices $\mathbf{D}$ and $\mathbf{W}$, respectively:

$$\mathbf{W} = \mathrm{diag}(\mathbf{F}\mathbf{g}_{ZP}), \quad \mathbf{D} = \mathrm{diag}(\mathbf{F}\mathbf{h}_{ZP}).$$

Therefore we can switch from the $\mathbf{W}$-Matrix to the $\mathbf{D}$-matrix and reverse. Subsequently we focus on transforming $\mathbf{h}$ into $\hat{\mathbf{g}}_{LS}$, bearing in mind that $\hat{\mathbf{h}}_{LS}$ can be estimated from $\mathbf{g}$ by using similar algorithms.

One idea that leads to the first algorithm is to extend the convolution matrix $\mathbf{H}$ cyclically, so that we can take advantage of the eigenvalue decomposition (EVD) of cyclic matrices. If the $\mathbf{g}$-vector is zero padded, any column can be inserted on the right of the $\mathbf{H}$ matrix without changing the right hand side of equation system 2. Therefore zero padding of $\mathbf{g}$ provides a method to create a cyclic channel matrix. As stated above cyclic matrices can be inverted very efficiently but the inverse might not exist:

$$\tilde{\mathbf{H}}\mathbf{g}_{ZP} = \mathbf{p} + \delta(\mathbf{p}), \quad \mathbf{g}_{ZP} = \begin{bmatrix}\mathbf{g}^H & \mathbf{0}^H\end{bmatrix}^H \in \mathbb{C}^{M+L-1},$$
$$\hat{\mathbf{g}}_{ZP} = \tilde{\mathbf{H}}^{-1}\mathbf{p} = \mathbf{F}^{-1}\mathbf{D}^{-1}\mathbf{F}\mathbf{p} = \begin{bmatrix}\hat{\mathbf{g}}^H & \hat{\mathbf{s}}^H\end{bmatrix}^H.$$

After setting $\hat{\mathbf{s}}^H$ to zero the $\mathbf{W}$ matrix is obtained by:

$$\mathbf{W} = \mathrm{diag}(\mathbf{F}\begin{bmatrix}\hat{\mathbf{g}}^H & \mathbf{0}^H\end{bmatrix}^H).$$

This is a very efficient method to compute the filter weights. However, the CP approach will suffer if the inverse $\tilde{\mathbf{H}}$ does not exist, or if we deal with ill-conditioned matrices (deep fading conditions).

Next efficient algorithms to compute the LS-estimates $\hat{\mathbf{g}}_{LS}$ are derived and assessed. It is shown which computational complexity we have to provide additionally to gain the advantage of guaranteed invertibility, independently of the channel parameters. To compute the LS-estimates $\hat{\mathbf{g}}_{LS}$ only one column of the pseudo inverse $\mathbf{H}^\dagger$ need to be computed as $\mathbf{p}$ contains only a single one. The position of the one defines the column which need to be calculated and has therefore an impact on the complexity of computing the filter taps. The QR factorization decomposes the convolution matrix $\mathbf{H}$ into an unitary matrix $\mathbf{Q}$ ($\mathbf{Q}^H\mathbf{Q} = \mathbf{I}$) and an upper triangular matrix $\mathbf{R}$: $\mathbf{H} = \mathbf{QR}$. By using this decomposition two different approaches to compute the LS-estimates $\hat{\mathbf{g}}_{LS}$ are obtained: One computes the $\mathbf{R}$-matrix and a column of $\mathbf{Q}^H$ and solves $\mathbf{R}\hat{\mathbf{g}}_{LS} = \mathbf{Q}^H\mathbf{p}$ via back substitution, another computes $\mathbf{R}^{-1}$ directly and computes $\hat{\mathbf{g}}_{LS} = \mathbf{R}^{-1}\mathbf{Q}^H\mathbf{p}$ via a matrix vector multiplication. Thereby the position of the one in $\mathbf{p}$ selects a column of $\mathbf{Q}^H$. The approaches differ in storage requirements, total number of multiplications and the need of divisions.

Using the later approach the LS-estimates can thus be computed as shown in figure 3. The outer computational structure

$\hat{\mathbf{g}}_{LS} = [\mathbf{R}^{-1}(1:1,1)\mathbf{q}(1); \mathrm{zeros}(M-1,1)];$
for $m = 1 : M-1$
  $\hat{\mathbf{g}}_{LS} = \hat{\mathbf{g}}_{LS} + [\mathbf{R}^{-1}(1:m+1, m+1)\mathbf{q}(m+1);$
  $\cdots \mathrm{zeros}(M-m-1,1)];$
end

Fig. 3. Computation of the filter taps from the inverse QR factorization of the channel matrix $\mathbf{H}$.

can be subdivided into three parts: computing $\mathbf{R}^{-1}$ column by column, computing $\mathbf{q} = \mathbf{Q}^H\mathbf{p}$ row by row and updating the the filter vector $\hat{\mathbf{g}}_{LS}$ by using the computed column of $\mathbf{R}^{-1}$ and the corresponding row of $\mathbf{q}$. Storage requirements are low since only vectors need to be stored instead of matrices. As the vectors $\mathbf{R}^{-1}(1:m,m)$ are multiplied by the scalars $\mathbf{q}(m)$ the multiplication can be performed in a parallel fashion.

After describing how to obtain $\mathbf{R}$ from the channel parameters row by row it is described how $\mathbf{R}^{-1}$ can be computed column by column. It will turn out that similar hyperbolic rotations can be used to obtain $\mathbf{R}$, $\mathbf{R}^{-1}$ and $\mathbf{Q}^H\mathbf{p}$, respectively [6]. The convolution matrix $\mathbf{H}$ and its hermitian $\mathbf{H}^H$ can be partitioned as follows:

$$\begin{aligned}
\mathbf{H} &= \begin{bmatrix} h_1 & \mathbf{0}^H \\ \mathbf{v} & \mathbf{H}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{H}_1 & \mathbf{u} \\ \mathbf{0}^H & h_L \end{bmatrix} \in \mathbb{C}^{(M+L-1)\times M}, \\
\mathbf{H}^H &= \begin{bmatrix} h_0^H & \mathbf{v}^H \\ \mathbf{0} & \mathbf{H}_1^H \end{bmatrix} = \begin{bmatrix} \mathbf{H}_1^H & \mathbf{0} \\ \mathbf{u}^H & h_L^H \end{bmatrix}, \\
\mathbf{v}^H &= \begin{bmatrix} h_2^H & \cdots & h_L^H & 0^H \end{bmatrix} \in \mathbb{C}^{M+L-2}, \\
\mathbf{u}^H &= \begin{bmatrix} \mathbf{0}^H & h_1 & \cdots & h_{L-1} \end{bmatrix} \in \mathbb{C}^{M+L-2}.
\end{aligned}$$

Since $\mathbf{h}$ shall have at least one non zero entry, $\mathbf{H}$ has full column rank. Utilizing the partitions the correlation matrix $\mathbf{H}^H\mathbf{H}$ can be computed as follows:

$$\begin{aligned}\mathbf{H}^H\mathbf{H} &= \begin{bmatrix} h_1^H h_1 + \mathbf{v}^H\mathbf{v} & \mathbf{v}^H\mathbf{H}_1 \\ \mathbf{H}_1^H\mathbf{v} & \mathbf{H}_1^H\mathbf{H}_1 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{H}_1^H\mathbf{H}_1 & \mathbf{H}_1^H\mathbf{u} \\ \mathbf{u}\mathbf{H}_1 & \mathbf{u}^H\mathbf{u} + h_L^H h_L \end{bmatrix}.\end{aligned}$$

As $\mathbf{H}^H\mathbf{H}$ is hermitian and positive definite the Cholesky decomposition exists: $\mathbf{H}^H\mathbf{H} = \mathbf{R}^H\mathbf{Q}^H\mathbf{Q}\mathbf{R} = \mathbf{R}^H\mathbf{R}$. The upper triangular matrix $\mathbf{R}$ (Cholesky factor) and its hermitian $\mathbf{R}^H$ can be partitioned similarly to $\mathbf{H}$:

$$\begin{aligned}\mathbf{R} &= \begin{bmatrix} r_{11} & \mathbf{z}^H \\ \mathbf{0} & \mathbf{R}_b \end{bmatrix} = \begin{bmatrix} \mathbf{R}_t & \mathbf{w} \\ \mathbf{0} & r_{MM} \end{bmatrix} \in \mathbb{C}^{M\times M}, \\ \mathbf{R}^H &= \begin{bmatrix} r_{11}^H & \mathbf{0}^H \\ \mathbf{z} & \mathbf{R}_b^N \end{bmatrix} = \begin{bmatrix} \mathbf{R}_t^H & 0 \\ \mathbf{w}^H & r_{MM}^H \end{bmatrix}. \end{aligned} \quad (3)$$

The partitioned matrices are used to compute the correlation:

$$\begin{aligned}\mathbf{R}^H\mathbf{R} &= \begin{bmatrix} r_{11}^H r_{11} & r_{11}^H\mathbf{z}^H \\ \mathbf{z}r_{11} & \mathbf{z}\mathbf{z}^H + \mathbf{R}_b^H\mathbf{R}_b \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{R}_t^H\mathbf{R}_t & \mathbf{R}_t^H\mathbf{w} \\ \mathbf{w}^H\mathbf{R}_t & \mathbf{w}^H\mathbf{w} + r_{MM}^H r_{MM} \end{bmatrix}.\end{aligned}$$

If corresponding matrix blocks of $\mathbf{H}^H\mathbf{H} = \mathbf{R}^H\mathbf{R}$ are compared the following equations are found:

$$r_{11}^H r_{11} = \mathbf{h}^H\mathbf{h} \Rightarrow r_{11} = \sqrt{\mathbf{h}^H\mathbf{h}}, \quad \mathbf{z}^H = \frac{1}{r_{11}^H}\mathbf{v}^H\mathbf{H}_0.$$

By using these two equations the first row of $\mathbf{R}$ can be computed from the channel parameters. The vector $\mathbf{z}$ has $L-1$ non zero entries. Further comparisons yield an equation known as Cholesky down dating problem [2]:

$$\left.\begin{matrix} \mathbf{z}\mathbf{z}^H + \mathbf{R}_b^H\mathbf{R}_b = \mathbf{H}_0^H\mathbf{H}_0 \\ \mathbf{H}_0^H\mathbf{H}_0 = \mathbf{R}_t^H\mathbf{R}_t \end{matrix}\right\} \Rightarrow \mathbf{R}_b^H\mathbf{R}_b = \mathbf{R}_t^H\mathbf{R}_t - \mathbf{z}\mathbf{z}^H$$

A matrix $\mathbf{Y}$ is called J-orthogonal if the following conditions are kept:

$$\mathbf{Y^H J Y} = \mathbf{J}, \quad \mathbf{J} = \begin{bmatrix} \mathbf{I} & \\ & -1 \end{bmatrix}.$$

By using the J-orthogonal matrix $\mathbf{Y}$ the down dating problem can be rewritten:

$$\begin{bmatrix} \mathbf{R}_t^H & \mathbf{z} \end{bmatrix} \mathbf{Y^H J Y} \begin{bmatrix} \mathbf{R}_t \\ \mathbf{z}^H \end{bmatrix} = \begin{bmatrix} \mathbf{R}_b^H & \mathbf{0} \end{bmatrix} \mathbf{J} \begin{bmatrix} \mathbf{R}_b \\ \mathbf{0} \end{bmatrix}.$$

Therefore a J-orthogonal transformation $\mathbf{Y}$ can be found so that

$$\mathbf{Y}\begin{bmatrix} \mathbf{R}_t \\ \mathbf{z}^H \end{bmatrix} = \begin{bmatrix} \mathbf{R}_b \\ \mathbf{0} \end{bmatrix}. \quad (4)$$

The transformation $\mathbf{Y}$ has to create zeros in the last row of the left hand side of equation 4.

An elementary 2×2 hyperbolic rotation is J-orthogonal and can create a zero in a matrix. For complex data it is defined by the conditions:

$$\mathbf{T}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ \gamma \end{bmatrix}, \quad \mathbf{T} = \begin{bmatrix} c & -s \\ -s^* & c^* \end{bmatrix} \in \mathbb{C}^{2\times 2}, \quad (5)$$

$$\mathbf{T}^H\mathbf{J}\mathbf{T} = \mathbf{J} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \Leftrightarrow c^*c - s^*s = 1.$$

The parameters s and c that define the elementary transformations $\mathbf{T}$ can be computed from the data x_1 and x_2 as shown in figure 4. There are some numerical problems if x_1 and x_2 are of similar values. The Matrix $\mathbf{Y}$ can be expressed as a concatenation of elementary hyperbolic transformations $\mathbf{T}(i, j, \Phi)$ that is an $M \times M$ identity matrix except $t_{ii} = c$, $t_{jj} = c^*$, $t_{ij} = s^*$ and $t_{ji} = s$.

function $[c, s] = \text{hyp}(x_1, x_2);$
if $x_2 = 0$
  $s = 0; c = 1;$
elseif $\|x_2\|_2 < \|x_1\|_2$
  $\tau = x_2/x_1; c = 1/\sqrt{1-\tau^*\tau}; s = c\tau;$
elseif $\|x_1\|_2 < \|x_2\|_2$
  $\tau = x_1/x_2; s = 1/\sqrt{1-\tau^*\tau}; c = s\tau;$
end
return

Fig. 4. Computation of the complex hyperbolic rotation.

Referring to equation 4 the following computation technique to obtain $\mathbf{R}$ is adopted: Since the first row of $\mathbf{R}_t$ and $\mathbf{z}^H$ is known the first hyperbolic rotation, that creates a zero in the first element in the last row can be computed and thus the first row of $\mathbf{R}_b$. The first row of $\mathbf{R}_b$ without the last element equals the second row of $\mathbf{R}_t$ (see equation. 3). Thus the second row of $\mathbf{R}_b$ can be computed and so on. This is a Schur type algorithm to overwrite $\mathbf{R}_t$ with $\mathbf{R}_b$ row by row. Finally the Cholesky factor $\mathbf{R}$ is found as it can be composed of r_{11}, $\mathbf{z}^H$ and $\mathbf{R}_b$. This algorithm is depicted in figure 5. This provides an efficient method to compute the Cholesky factor $\mathbf{R}$ row by row by using a concatenation of hyperbolic rotations. The filter coefficients can be computed from $\mathbf{R}\hat{\mathbf{g}}_{LS} = \mathbf{Q}^H\mathbf{p}$ by using a back substitution. However, one division per estimated coefficient is required and the whole matrix $\mathbf{R}$ need to be stored.

$\mathbf{R}(1,1) = \sqrt{\mathbf{h}^H\mathbf{h}};$
$\mathbf{z}^H = \mathbf{v}^H\mathbf{H}_1/\mathbf{R}(1,1);$
$\mathbf{R}(1, 2:M) = \mathbf{z}^H;$
for $m = 1 : M-1;$
  $[c, s] = \text{hyp}(\mathbf{R}(m,m), \mathbf{z}^H(m));$
  $\mathbf{R}(m+1, m+1:M) =$
  $c^*\mathbf{R}(m, m:M-1) - s^*\mathbf{z}^H(m:M-1);$
  $\mathbf{z}^H(m:M-1) =$
  $-s\mathbf{R}(m, m:M-1) + c\mathbf{z}^H(m:M-1);$
end

Fig. 5. Computation of the Cholesky factor $\mathbf{R}$.

Subsequently we focus on computing the inverse $\mathbf{R}^{-1}$ directly. The inverse $\mathbf{R}^{-H}$ is partitioned similar to $\mathbf{R}$:

$$\mathbf{R}^{-H} = \begin{bmatrix} \frac{1}{r_{11}^H} & \mathbf{0}^H \\ -\mathbf{R}_b^{-H}\frac{\mathbf{z}}{r_{11}^H} & \mathbf{R}_b^{-H} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_t^{-H} & \mathbf{0}^H \\ -w^H\mathbf{R}_t^{-H}/r_{MM}^H & \frac{1}{r_{MM}^H} \end{bmatrix}. \quad (6)$$

We want to compute $\mathbf{R}_b^{-H}$ directly from $\mathbf{R}_t^{-H}$. Since only square matrices can have an inverse the vector $[\mathbf{0}^H 1^H]^H$ is inserted at the left hand side of equation 4. Then a vector $\mathbf{s}$ and a scalar κ exist, so that the following equation applies:

$$\mathbf{Y}\begin{bmatrix} \mathbf{R}_t & \mathbf{0} \\ \mathbf{z}^H & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_b & \mathbf{s} \\ \mathbf{0}^H & \kappa \end{bmatrix}.$$

By utilizing the J-orthogonality of $\mathbf{Y}$ a vector $\mathbf{t}$ and a scalar γ can be found so that:

$$\begin{bmatrix} \mathbf{R}_t^H & \mathbf{z} \\ \mathbf{0}^H & 1^H \end{bmatrix}\underbrace{\mathbf{Y^H J Y}}_{=\mathbf{J}}\begin{bmatrix} \mathbf{R}_t^{-H} & \mathbf{R}_t^{-H}\mathbf{z} \\ \mathbf{0}^H & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_b^H & \mathbf{0}^H \\ \mathbf{s}^H & \kappa^H \end{bmatrix}\mathbf{J}\begin{bmatrix} \mathbf{R}_b^{-H} & \mathbf{0} \\ -\mathbf{t}^H & \gamma \end{bmatrix}.$$

Therfore a similar transformation $\mathbf{Y}$ used to compute $\mathbf{R}_b$ from $\mathbf{R}_t$ can be used to compute $\mathbf{R}_b^{-H}$ from $\mathbf{R}_t^{-1}$:

$$\mathbf{Y}\begin{bmatrix} \mathbf{R}_t^{-H} & \mathbf{R}_t^{-H}\mathbf{z} \\ \mathbf{0}^H & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_b^{-H} & \mathbf{0} \\ -\mathbf{t}^H & \gamma \end{bmatrix}.$$

With $r_{11}, \mathbf{z}$ and $\mathbf{R}_b^{-H}$ the hermitian, inverse Cholesky factor $\mathbf{R}^{-H}$ can be composed (see equation 6). The vector $\left[(\mathbf{R}_t^{-H}\mathbf{z})^H \quad 1^H\right]^H$ is rotated to $\left[\mathbf{0}^H \quad \gamma\right]^H$ in a hyperbolic fashion, which means that

$$\gamma = \sqrt{1 - \|\mathbf{R}_t^{-H}\mathbf{z}\|_2^2}.$$

This leads to a further method to find the elementary hyperbolic rotations:

$$\mathbf{Y}\begin{bmatrix} \mathbf{R}_t^{-H}\mathbf{z} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \gamma \end{bmatrix}.$$

By using the elementary hyperbolic rotations computed in this manner the rows of the matrix $\mathbf{R}_t^{-H}$ can be overwritten by the rows of $\mathbf{R}_b^{-H}$:

$$\mathbf{Y}\begin{bmatrix} \mathbf{R}_t^{-H} \\ \mathbf{0}^H \end{bmatrix} = \begin{bmatrix} \mathbf{R}_b^{-H} \\ -\mathbf{t} \end{bmatrix}.$$

The resulting algorithm computes $\mathbf{R}^{-H}$ row by row, which implies that $\mathbf{R}^{-1}$ is computed column by column. The algorithm is depicted in figure 6.

Next we need to compute $\mathbf{q} = \mathbf{Q}^H\mathbf{p}$. Since $\mathbf{p}$ contains only a single one only one column of $\mathbf{Q}^H$ needs to be calculated. The column that must be computed is defined by the position of the one. In [6] it is derived how the parameters of the elementary hyperbolic rotations c and s used to compute either $\mathbf{R}$ or $\mathbf{R}^{-H}$

$\mathbf{r}_{11} = \sqrt{\mathbf{h}^H\mathbf{h}}; \mathbf{t}^H = 0;$
$\mathbf{z} = \mathbf{H}_1^H\mathbf{v}/\mathbf{r}_{11}; \mathbf{t}^H = 0;$
for $m = 1 : M-1$
  $\beta = \mathbf{R}^{-H}(m, 1 : L-1)\mathbf{z};$
  $[c, s] = \text{hyp}(\gamma, \beta);$
  $\gamma = -s^*\beta + c^*\gamma;$
  $\mathbf{R}_b^{-H}(m) = c\mathbf{R}^{-H}(m, 1 : m) - s * \mathbf{t}^H;$
  $\mathbf{t}^H = s^*\mathbf{R}^{-H}(m, 1 : m) - c^* * \mathbf{t}^H;$
  $\mathbf{t}^H = [\mathbf{t}^H \quad 0];$
  $\mathbf{R}^{-H}(m+1, :) = [-\mathbf{R}_b^{-H}(m, 1 : L-1)z)/r_{11}$
  $\cdots \mathbf{R}_b^{-H}(m, 1 : m)];$
end

Fig. 6. Computation of $\mathbf{R}^{-1}$ column by column from the channel coefficients.

can be used to construct $\mathbf{q} = \mathbf{Q}^H\mathbf{p}$. A transformation matrix $\mathbf{\Theta}$ is obtained by:

$$\begin{aligned} \mathbf{\Theta} &= \hat{\mathbf{T}}_{M-1}\cdots\hat{\mathbf{T}}_2\hat{\mathbf{T}}_1, \\ \hat{\mathbf{T}}_k &= \mathbf{T}(m, 2m-k, \Phi_k)\cdots \\ &\quad \mathbf{T}(k+2, n+2, \Phi_k)\mathbf{T}(k+1, n+1, \Phi_k). \end{aligned}$$

It follows from [6] that the J-orthogonal matrix $\mathbf{\Theta}$ can be used to obtain $\mathbf{Q}^H\mathbf{p}$:

$$\mathbf{\Theta}\begin{bmatrix} U(\mathbf{y}_1) \\ U(\mathbf{y}_2) \end{bmatrix}\mathbf{p} = \begin{bmatrix} \mathbf{q} = \mathbf{Q}^H\mathbf{p} \\ * \end{bmatrix}, \quad \mathbf{y}_1 = \mathbf{h}/(\mathbf{h}^H\mathbf{h}) = \mathbf{y}_2.$$

We only need the first M values of $\mathbf{q}$ to compute the filter taps. The Krylov matrix $U(\mathbf{y}_i)$ is defined by

$$U(\mathbf{y}_i) = \begin{bmatrix} \mathbf{y}_i \\ \mathbf{y}_i\mathbf{Z}^H \\ \vdots \\ \mathbf{y}_i(\mathbf{Z}^{M-1})^H \end{bmatrix}, \quad \mathbf{Z} = \begin{bmatrix} \mathbf{0}^H & 0 \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \in \mathbb{C}^{M\times M}.$$

Then the first M values of $\mathbf{q}$ can be computed by the loop depicted in figure 7. The vector $\mathbf{c}$ is overwritten with $\mathbf{Q}^H\mathbf{p}$ row

$\mathbf{c}_1 = \mathbf{c}_2 = \mathbf{U}(\mathbf{y}_1)\mathbf{p};$
$\mathbf{q} = [\mathbf{c}_1; \mathbf{c}_2];$
for m=1:M-1;
  $\mathbf{q} = \tilde{\mathbf{T}}_m\mathbf{q}$
end

Fig. 7. Computation of one column of $\mathbf{Q}^H$.

by row.

Now three different algorithms can be composed to switch between the $\mathbf{D}$-matrix and the $\mathbf{W}$-matrix. One uses the cyclic extension, another a back substitution after a QR decomposition and a third a matrix vector product after an inverse QR factorization. These algorithms are compared and assessed in the next chapter.

III. Assessment and Applications of the Algorithms

The number of necessary multiplications for the different algorithms can be expressed as a function of the filter order M,

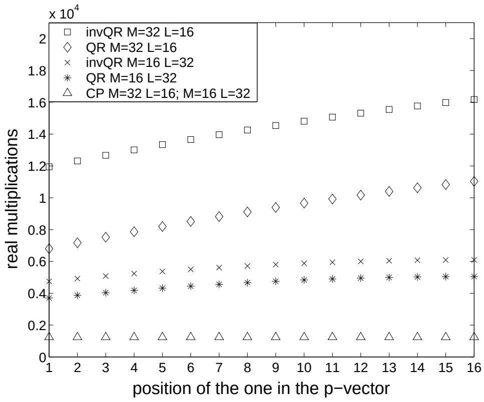

Fig. 8. Real multiplications of three different algorithms as a function of p.

the channel length L and the position of the one p (see table I). The parameter N is the smallest power of two larger than $M+L-1$. Complex-complex multiplications are weighted with 4, whereas complex-real multiplications are weighted with 2. Multiplications by zero are not taken into account. The complexity changes if we switch from OFDM parameters to FIR filter taps or the way round. Either we have to compute many filter coefficients from a few channel parameters ($M > L$) or we have to compute a few channel parameters from many filter coefficients ($M < L$). The number of multiplications of the different algorithms as a function of the position of the one are depicted in figure 8. The algorithm presented in [6] outperforms other algorithms known to compute $\hat{\mathbf{g}}_{LS}$ via an inverse QR factorization in a general Toeplitz system. By exploiting the characteristics of convolution matrices (band-structure), the degree of freedom concerning the position of the one in the $\mathbf{p}$ vector and that only the first M values of $\mathbf{p}$ need to be computed the overall complexity of the basic algorithm using the fast inverse QR factorization [6] is reduced enormously. The main advantage of this approach is the low storage requirements as only vectors need to be stored and that only one division is necessary. Often vectors are multiplied by scalars which can have advantages for implementations. Using a back substitution needs fewer multiplications but the whole matrix $\mathbf{R}$ need to be stored and one division per filter tap becomes necessary. Significant fewer multiplications are needed if the channel matrix is extended cyclically and the corresponding system is solved via the eigenvalue decomposition of cyclic matrices. However, in this approach the property of guaranteed invertability is sacrificed. One technique to reduce error enhancement in the CP case is to set zero valued eigenvalues to a certain numbers. Especially in deep fading conditions the SC-FDE equalization will outperform uncoded, non adaptive OFDM and CP-SC. Therfore it seems to be reasonable to switch between the systems if at least one eigenvalue is zero.

TABLE I

REAL MULTIPLICATIONS OF THREE DIFFERENT ALGORITHMS.

	$M > L$	$M < L$
	$\hat{\mathbf{g}}_{LS} = \mathbf{R}^{-1}\mathbf{Q}^H\mathbf{p}$	
$\hat{\mathbf{g}}_{LS}$	$2M^2 + 2M - 4$	$2M^2 + 2M - 4$
$\mathbf{q}$	$p < L$: $12Mp - 6p^2 - 6M$	$p < M$: $12Mp - 6p^2 - 6M$
$\mathbf{q}_{max}$	$12ML - 6L^2 - 6L$	$12ML - 6L^2 - 6L$
$\mathbf{R}^{-1}$	$6M^2 + 8ML - 2L^2$ $-6M + 8L - 10$	$8M^2 + 4ML$ $+2M - 10$
sum	$8M^2 + 12Mp - 2L^2$ $-6p^2 + 8ML - 10M$ $+8L - 14$	$10M^2 + 4ML$ $+12Mp - 6p^2$ $-2M - 14$
	$\hat{\mathbf{R}}\mathbf{g}_{LS} = \mathbf{Q}^H\mathbf{p}$	
$\mathbf{R}$	$12ML - 4L^2$ $+4L - 8$	$4M^2 + 4LM$ $+4M - 8$
$\hat{\mathbf{g}}_{LS}$	$4ML - 2L^2$ $-4M + 2L$	$2M^2 - 2M$
sum	$16ML + 12Mp$ $-6L^2 - 6p^2 + 6L$ $-10M - 8$	$6M^2 + 4LM - 6p^2$ $+12Mp - 4M - 8$
	$\hat{\mathbf{g}}_{ZP} = \mathbf{F}^{-1}\mathbf{D}^{-1}\mathbf{F}\mathbf{p}$	
sum	$4N\text{ld}(N/2)$	$4N\text{ld}(N/2)$

IV. CONCLUSIONS

The presented algorithms enable an efficient transfer between channel parameters and filter weights. The receivers' algebraic structures of SC-FDE, OFDM and CP-SC systems are very similar and differ manly in their parameters. Although the systems are suited for high data rate transmissions over wireless channels there are significant performance differences between the systems so that it is desirable to adapt the system to a specific transmission scenario. A multi-mode device able to perform receiver processing in SC-FDE, OFDM and CP-SC systems can be implemented by using similar hardware. Beside the switching during transmission mode it might be reasonable to switch during training mode. Either to start the channel estimation in SC-FDE systems, e.g., applying the LMS algorithm, at better starting values or to improve the channel estimation in OFDM systems.

REFERENCES

[1] D. Falconer, S.L. Ariyavisitakul, A. Benyamin-Seeyar and B. Eidson, "*Frequency Domain Equalization for Single-Carrier Broadband Wireless Systems*," IEEE Communication Magazine, April 2002.

[2] G.H. Golub, C.F. van Loan, "*Matrix Computations*," The Johns Hopkins University Press, Baltimore and London, ISBN 0-8018-5413-X, Third Edition 1996.

[3] S. Haykin, "*Adaptive Filter Theory*," Prentice Hall, ISBN 0-13-322760-X, Third Edition 1996.

[4] J. Kim, J. Kang and E. J. Powers, "*A Bandwidth Efficient OFDM Transmission Scheme*," Proceedings ICASSP 2002, Orlando, USA, Mai 2002.

[5] A. Koppler, H. Witschnig, M. Huemer, R. Weigel, "*Comparison of channel estimation techniques for a single carrier system using frequency domain equalization and Hiperlan/2-like OFDM systems*," MC-SS Workshop, DLR, Oberpfaffenhofen, Germany, September 2001.

[6] J.G. Nagy, "*Fast Inverse QR Factorization for Toeplitz Matrices*," SIAM J. Sci. Comput., Vol. 14, No. 5, pp. 1174-1193, September 1993.

[7] A.V. Oppenheim, R.W. Schafer, "*Discrete-Time Signal Processing*," Prentice-Hall, Englewood Cliffs, New Jersey 07632, ISBN 0-13-216292-X, 1989.

[8] C.V. Sinn, J. Götze and M. Haardt, "*Efficient Data Detection Algorithm in Single- and Multi-Carrier Systems without the Necessity of a Guard Period*," Proceedings ICASSP 2002, Orlando, USA, Mai 2002.

[9] M. Vollmer, M. Haardt and J. Götze, "*Comparative Study of Joint-Detection Techniques for TD-CDMA based Mobile Radio Systems*," IEEE J. Select. Areas Commun., vol. 19, no. 8, pp. 1461-1475, August 2001.

[10] M. Vollmer, J. Götze, M. Haardt, "*Schur Algorithms for Joint-Detection in TD-CDMA based Mobile Radio Systems*," Annales des Telecommunications, vol.54, pp. 365-378, special issue on "Multiuser Detection", Sept. 1999.

[11] Z. Wang and G. B. Giannakis, "*Wireless Multicarrier Communications*," IEEE Signal Processing Magazine, vol. 17, no.3, pp. 29-48, May 2000.