COMPARATIVE STUDY OF TECHNIQUES TO COMPUTE FIR FILTER WEIGHTS IN ADAPTIVE CHANNEL EQUALIZATION

Christian Vincent Sinn * *and Jürgen Götze*

University of Dortmund, Information Processing Laboratory, D-44221 Dortmund, Germany
{vincent.sinn, juergen.goetze}@uni-dortmund.de

ABSTRACT

In this article the computation of FIR filter weights in adaptive channel equalization tasks in quasi stationary environments is considered. The problem is formulated as a system of equations. It can be solved via direct matrix inversion (DMI) or iteratively via the LMS or RLS algorithm. Thereby suitable criteria such as least squares (LS) or mean square error (MSE) are minimized. By using these techniques the filter weights are estimated. Another technique is to estimate the channel impulse response (CIR) by exploiting the eigenvalue decomposition (EVD) of cyclic matrices as done in orthogonal frequency domain multiplex (OFDM) systems and computing the FIR filter weights from the CIR via solving the zero forcing matrix equation. Different techniques to solve this equation are presented: One uses a cyclic prefix (CP) based approach, another a QR decomposition. After describing the different techniques they are assessed in terms of ensemble-averaged square error and computational complexity. The EVD based techniques can render the lowest square error and require significant fewer multiplications than iterative methods or the DMI technique.

1. INTRODUCTION

The transmission of digital data through a linear communication channel is limited by inter symbol interference and noise. Often the channel is time variant as e.g. in mobile communications. Two basic concepts of modeling time varying channels are known. One assumes a change of the channel impulse response (CIR) at every time sample. Another assumes that the channel is time invariant during a short period of time in which one data burst is transmitted. The CIR is assumed to change only from burst to burst. This scenario is adopted in this article. The task of an adaptive equalizer can be subdivided in three parts. First the filter weights need to be estimated. Then the filter process of distorted data is performed. Finally, the filter parameter are adapted to a changed environment. In this article it is focused on the estimation of the filter parameter based on training data that is known at the receiver. The equalization filter is modeled as tapped-delay line filter.

Three techniques to estimate the filter weights are presented. The well known LMS-algorithm is chosen as the reference. It computes the filter weights from the filter input and the desired response iteratively. After an adequate number of iterations, the filter weights converge against the Wiener solution apart from a small deviation which is known as misadjustment [2]. The Wiener solution can be obtained via direct matrix inversion (DMI). Another technique is to estimate the CIR first and then computing the filter weights from the CIR via solving the zero forcing matrix equation. In this article efficient techniques for performing channel estimation and solving the zero forcing matrix equation are presented. The efficiency of the channel estimation is based on exploiting the eigenvalue decomposition (EVD) of cyclic matrices as done in orthogonal frequency division multiplexing (OFDM) based transmission systems. The efficiency of solving the zero forcing matrix equation is either based on the insertion of a cyclic prefix or on solving a Cholesky down dating problem. Thereby a QR decomposition is computed by using hyperbolic rotations. With it the particular structure of the problem is exploited.

The different techniques are assessed in terms of computational complexity and an ensemble-averaged square estimation error. It turns out that the EVD approach for channel estimation in combination with efficient solutions of the zero forcing equation outperforms LMS and DMI techniques in terms of computational complexity and quality of the estimates. However, the tracking techniques and computational structures are more complex.

The paper is organized as follows: The different methods to compute filter weights from a training sequence are presented in the next chapter. Experimental results that contrast the methods in terms of an ensemble-averaged square error and computational complexity are presented in chapter 3. Conclusions are drawn in chapter 4.

2. COMPUTATION OF FIR FILTER WEIGHTS

In the following it is focused on adaptive equalization of linear time dispersive channels. The channel is assumed to be time invariant during one burst. The burst consists of training and data symbols as depicted in figure 1. During the training mode the FIR filter weights are adapted. Then information data is transmitted, whereby the filter weights are kept constant. The channel is only allowed to change in the next burst. There are two different adaptation processes: One adapts the filter weights during training to a constant CIR, another adapts the filter weights to a new CIR burst by burst. This corresponds to a quasi stationary model of the channel to be equalized.

Training symbols	Data symbols

Fig. 1. Structure of a data burst to be transmitted.

2.1. Direct Matrix Inversion and Wiener Solution

Subsequently the training mode is considered. With reference to figure 2 the filter output $\mathbf{y} \in \mathbb{C}^N$ is obtained by multiplying the

*This work was funded by Siemens AG, ICM, Munich, Germany.

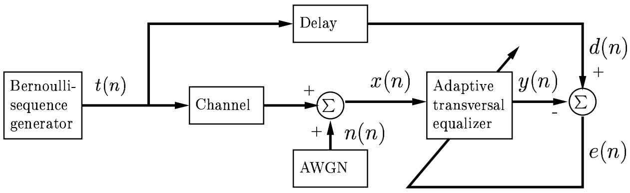

Fig. 2. Structure of adaptive filter during training mode.

convolution matrix $\mathbf{X} \in \mathbb{C}^{N\times M}$ that contains the input data blocks of the filter in its rows by the filter weight vector $\mathbf{w} \in \mathbb{C}^{M}$. The filter output is equal to the delayed training data $\mathbf{d} \in \mathbb{C}^{N}$ (desired response) plus an error vector $\mathbf{e} \in \mathbb{C}^{N}$:

$$\mathbf{y} = \mathbf{X}\mathbf{w} = \mathbf{d} + \mathbf{e}.$$

This matrix equation has the following structure:

$$\begin{bmatrix} x_1 & & & \\ x_2 & x_1 & & \\ \vdots & x_2 & \ddots & \\ x_{R-1} & \vdots & \ddots & x_1 \\ x_R & x_{R-1} & & x_2 \\ & x_R & \ddots & \vdots \\ & & \ddots & x_{R-1} \\ & & & x_R \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_M \end{bmatrix} = \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_T \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ \vdots \\ e_N \end{bmatrix}. \quad (1)$$

One colum of $\mathbf{X}$ is obtained by convolving the training data $\mathbf{t} \in \mathbb{C}^{T}$ with the channel impulse response $\mathbf{h} \in \mathbb{C}^{L}$ and by adding a white Gaussian noise vector $\mathbf{n} \in \mathbb{C}^{R}$:

$$\mathbf{X}(1:R,1) = \mathbf{t} * \mathbf{h} + \mathbf{n} \in \mathbb{C}^{R}, \quad R = T + L - 1.$$

The parameter L defines the channel length, T the length of the training sequence, M the filter order and N the total number of considered time steps. Equation 1 represents an overdetermined system of equations. Usually it has no solution. An approximated solution can be found by minimizing the error's energy in the least squares sense. If the data is free of noise this optimization criterion leads to the pseudo inverse of the data matrix $\mathbf{X}$:

$$\min_{\hat{\mathbf{w}}_{LS}} \|\mathbf{d} - \mathbf{X}\hat{\mathbf{w}}_{LS}\|_2 \quad \rightarrow \quad \hat{\mathbf{w}}_{LS} = \underbrace{(\mathbf{X}^H\mathbf{X})^{-1}\mathbf{X}^H}_{\mathbf{X}^\dagger} \mathbf{d} \in \mathbb{C}^{M}.$$

The least squares (LS) estimates of the filter weights $\hat{\mathbf{w}}_{LS}$ are obtained by multiplying the pseudo inverse $\mathbf{X}^\dagger$ by the desired response $\mathbf{d}$. This technique is called direct matrix inversion. It renders the Wiener solution. However, it is usually not adopted as the complexity to compute the filter weights is enormous: matrix products and matrix inversions of comparatively large matrices need to be performed.

2.2. Least Mean Square Algorithm

Another optimization criterion is the least mean square (LMS) error. Thereby the mean square error $\mathrm{E}\{e_n e_n^*\}$ is minimized at every time instant. That is for each row in equation 1. The LMS-algorithm that is based on an approximated stochastic gradient computes an estimate of the LMS solution at every time step. The computation can be divided into three steps: Computing the filter output $y(n) = \mathbf{X}(n,:)\hat{\mathbf{w}}^H(n)$, computing the error $e(n) = d(n) - y(n)$ and adapting filter weights $\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) - \mu\mathbf{X}^T(n,:)e^*(n)$. The step size parameter is denoted by μ. To compute estimated filter weights by using either the DMI or the iterative LMS method the desired response $\mathbf{d}$ and the filter input data $\mathbf{X}$ need to be known. Another approach that leads to a similarly structured system of equations is the zero forcing matrix equation. On contrast to the methods mentioned so far the filter weights are computed from the CIR, which therefore need to be estimated first.

2.3. Zero Forcing Approach

The convolution of the CIR $\mathbf{h} \in \mathbb{C}^{L}$ and the FIR filter $\mathbf{w} \in \mathbb{C}^{M}$ shall result in a vector $\mathbf{p} \in \mathbb{C}^{S}$ containing a single one at an arbitrary position. Since the system is overdetermined an error vector $\delta(\mathbf{p}) \in \mathbb{C}^{S}$ is added. The position of the one defines the delay of the filter output (compare with figure 2):

$$\mathbf{H}\mathbf{w} = \mathbf{p} + \delta(\mathbf{p}), \quad \mathbf{H} \in \mathbb{C}^{S\times M}, \quad S = M + L - 1.$$

This matrix equation has the following structure:

$$\begin{bmatrix} h_1 & & & \\ h_2 & h_1 & & \\ \vdots & h_2 & \ddots & \\ h_{L-1} & \vdots & \ddots & h_1 \\ h_L & h_{L-1} & & h_2 \\ & h_L & \ddots & \vdots \\ & & \ddots & h_{L-1} \\ & & & h_L \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_M \end{bmatrix} = \begin{bmatrix} \\ 1 \\ \\ \end{bmatrix} + \begin{bmatrix} \delta(p_1) \\ \delta(p_2) \\ \delta(p_3) \\ \vdots \\ \delta(p_S) \end{bmatrix}. \quad (2)$$

The LS-solution is again obtained by minimizing the error's energy:

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

If the CIR is free of noise the filter weights $\hat{\mathbf{w}}_{LS}$ are obtained by multiplying the pseudo inverse $\mathbf{H}^\dagger$ by $\mathbf{p}$. By comparing the system matrices $\mathbf{X}$ (eq. 1) and $\mathbf{H}$ (eq. 2) it turns out that $\mathbf{H}$ is significantly smaller than $\mathbf{X}$ if $L \ll R$, which is usually the case (L: channel length, T length of training symbols, $R = T + L - 1$). Furthermore the $\mathbf{p}$-vector contains on contrast to the $\mathbf{d}$-vector only a single one. Therefore only one column of the pseudo inverse $\mathbf{H}^\dagger$ need to be computed to obtain the LS-estimates of the filter weights $\hat{\mathbf{w}}_{LS}$. Both facts result in significantly lower computational complexity to solve equation 2 than equation 1. However, the CIR need to be estimated first. This estimation process is considered in chapter 2.4. Next it is focused on efficient techniques to solve this system of equations.

2.3.1. *Cyclic Prefix Based Approach*

If the convolution matrix $\mathbf{H}$ in equation 2 is extended cyclically we can take advantage of the EVD of cyclic matrices which 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 matrices $\mathbf{F}$ and $\mathbf{F}^{-1}$ are the DFT and IDFT matrices, respectively. They can be implemented by using the FFT. The Matrix $\tilde{\mathbf{H}}$ denotes the cyclic channel [6, 4]. The diagonal matrix $\mathbf{D}$ contains the CIR in the frequency domain. If the $\mathbf{w}$-vector is zero padded, any column can be inserted on the right of the matrix $\mathbf{H}$

without changing the right hand side of equation 2. Therefore zero padding of $\mathbf{w}$ provides a method to create a cyclic channel matrix. Cyclic matrices can be inverted very efficiently. However, the inverse might not exist. The concept of cyclic matrices is also used in some single and multi carrier systems [6, 4]. By using this technique equation 2 can be rewritten:

$$\tilde{\mathbf{H}}\mathbf{w}_{ZP} = \mathbf{p} + \delta(\mathbf{p}), \quad \mathbf{w}_{ZP} = \begin{bmatrix}\mathbf{w}^H & \mathbf{0}^H\end{bmatrix}^H \in \mathbb{C}^S.$$

Cyclic matrices are square. Therefore estimates of the zero padded filter weights $\hat{\mathbf{w}}_{ZP}$ are obtained by multiplying the inverse cyclic channel matrix $\tilde{\mathbf{H}}^{-1}$ by $\mathbf{p}$. Thereby one column of $\tilde{\mathbf{H}}^{-1}$ is selected:

$$\hat{\mathbf{w}}_{ZP} = \tilde{\mathbf{H}}^{-1}\mathbf{p} = \mathbf{F}^{-1}\mathbf{D}^{-1}\mathbf{F}\mathbf{p} = \begin{bmatrix}\hat{\mathbf{w}}^H & \hat{\mathbf{s}}^H\end{bmatrix}^H.$$

This is a very efficient method to compute the filter weights. However, the CP approach will suffer if the inverse $\tilde{\mathbf{H}}^{-1}$ does not exist, or if we deal with ill-conditioned matrices.

Next an efficient algorithm to compute the LS-estimates $\hat{\mathbf{w}}_{LS}$ is described and assessed. On contrast to the inverse cyclic channel matrix $\tilde{\mathbf{H}}^{-1}$ the pseudo inverse $\mathbf{H}^{\dagger}$ does always exist. It is shown which computational complexity we have to provide additionally to gain the advantage of guaranteed invertability, independently of the values of the CIR.

2.3.2. *QR Decomposition Based Approach*

To compute the LS-estimates $\hat{\mathbf{w}}_{LS}$ only one column of the pseudo inverse $\mathbf{H}^{\dagger}$ need to be computed. The position of the one in the $\mathbf{p}$-vector defines the column which need to be computed and has therefore an impact on the complexity of the filter weight computation. 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 the LS-estimates $\hat{\mathbf{w}}_{LS}$ are obtained by computing the $\mathbf{R}$-matrix and a column of $\mathbf{Q}^H$ and by solving $\mathbf{R}\hat{\mathbf{g}}_{LS} = \mathbf{Q}^H\mathbf{p}$ via back substitution. By exploiting the Toeplitz structure of $\mathbf{H}$ the matrix $\mathbf{R}$ and a column of $\mathbf{Q}^H$ can be computed from the CIR by using hyperbolic rotations which solve a Cholesky down dating problem. For further explanation of the algorithm see [3, 1, 5].

2.4. Channel Estimation Based on EVD of Cyclic Matrices

Now the estimation of the CIR is considered. The systems depicted in figures 2 and 3 are based on training sequences. Pseudo noise (PN) sequences are suited in both systems due to their auto and cross correlation properties. Next the channel estimation that is performed in OFDM systems is reviewed, since it provides a very efficient procedure for estimating the CIR in the frequency domain. The estimation structure is depicted in figure 3.

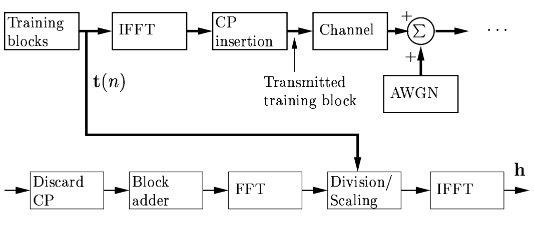

Fig. 3. Structure of channel estimation exploiting the eigenvalue decomposition of cyclic matrices.

A PN sequence $\mathbf{t} \in \mathbb{C}^L$ (training block) is modulated on different sub carriers by using the IFFT. Then a cyclic prefix (CP) of size $L-1$ is inserted [6, 4]. It has to be at least as long as the CIR minus one to eliminate block interference. After passing through the channel overlapping parts of subsequent training blocks are discarded. The remaining parts are feed to an FFT. In the absence of noise the CIR in the frequency domain is obtained by dividing the distorted training block by the initial training block $\mathbf{t}$. In the presence of noise a noisy estimate is obtained. Noise reduction is achieved by linear interpolation of several training blocks. Performing an IFFT of the noise reduced data block reveals the CIR $\mathbf{h} \in \mathbb{C}^L$. The size of one transmitted training block in the EVD based approach is $2L-1$ whereas the training sequence $t(n)$ of the adaptive filter is of length T. Usually T is significantly larger than $2L-1$ even if several training blocks $\mathbf{t}$ are transmitted: $T \gg J(2L-1)$, (J: number of training blocks).

2.5. Complexity Consideration

One coarse technique to assess the computational complexity of the algorithms is to count the necessary multiplications. Thereby complex multiplications are weighted by 4. In table 1 the total number of real multiplications of the algorithms is given as a function of the used parameters. The number of iterations I necessary for the LMS to converge is usually significantly larger than the number of training blocks J, the channel length L and the filter order M. For a coarse comparison of complexity we may assume $I \gg M > J > L$. Then the complexity of the LMS algorithm is significantly higher than performing both channel estimation and computation of the FIR filter weights via solving the zero forcing matrix equation. Computing a column of the pseudo inverse (PI) $\mathbf{H}^{\dagger}$ needs more computations than computing a column of the inverse cyclic channel matrix $\tilde{\mathbf{H}}^{-1}$. The position of the one in the $\mathbf{p}$-vector has an impact on the complexity [5] of the PI-method. Table 1 contains the upper bound.

Table 1. Real multiplications of different algorithms: Channel estimation via EVD (CH EST EVD), CP based approach (CP) and QR decomposition based approach (PI) to solve the zero forcing equation. The filter order is denoted by M, the number of iterations of the LMS algorithm by I and the channel length by L. The parameter $L_p(M_p)$ is a power of 2 that is larger than $L(M)$.

	Real multiplications
LMS	$8MI + I$
CH EST EVD	$4L_p \ln(L_p/2)$
CP	$4M_p \ln(M_p/2) + 4M_p$
PI	$28ML - 12L^2 - 4M - 8$

2.6. Discussion

One main difference between the iterative method (LMS) and those based on direct matrix inversions is the capability of tracking. In iterative methods an estimate of the filter weights is computed at every time instant whereas in direct methods the estimates are only computed after a block of time steps. However an update of the filter weights at every time step is usually not required. In iterative methods tracking does not work arbitrarily fast. Therefore a block of time steps is necessary to adapt to a knew transmission situation. In the next chapter the different techniques are assessed in terms of ensemble-averaged square error and number of real multiplications.

3. EXPERIMENTAL RESULTS

The channel impulse response of length L=3 is set to

$$h_l = \begin{cases} \frac{1}{2}\left[1+\cos\left(\frac{2\pi}{W}(l-1)\right)\right] & : \quad l=0,1,2 \\ 0 & : \quad \text{otherwise} \end{cases} .$$

The amplitude distortion of this channel increases with W. The noise generator (AWGN) produces a normally distributed noise signal with zero mean and a variance of $\sigma_n^2 = 0.005$. The training signal for the LMS algorithm is given by a Bernoulli sequence $t_n = \pm 1$. The random variable t_n has zero mean and unit variance. The step size parameter is set to $\mu = 0.04$. The filter order is $M = 12$. The size of one data block $\mathbf{t}$ in the EVD based channel estimation is defined by the channel length $L = 3$. Then the size of the CP is given by $L - 1 = 2$. Next the square error $e_n^* e_n$ of the LMS algorithm at every time instant averaged over 500 trails is depicted for different distortions of the channel: $W = 2.9$, $W = 3.8$. These results are compared to the error produced by the FIR filter computed via channel estimation in combination with the zero forcing approach. The number of training blocks J is either 1 or 16.

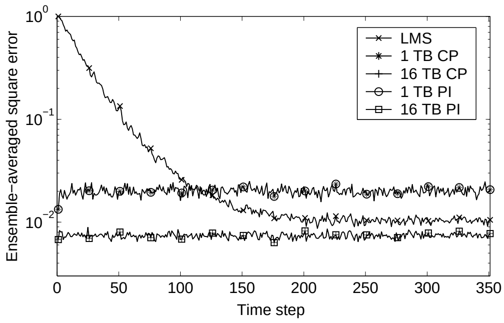

Fig. 4. Ensemble-averaged square error for Bernoulli training sequence at every time step. Channel distortion W=2.9.

In figure 4 the ensemble-averaged square error is plotted as a function of the time step. The pseudo inverse (PI) and the cyclic prefix (CP) based method render similar results. The number of used training blocks (TB) can reduce the average square error. If the number is large enough the result gives a lower bound for the LMS algorithm. In figure 5 the distortion parameter W is set to 3.8. This will result in ill-conditioned matrix equations (eq. 1 and eq. 2). If only one training block is transmitted the CP approach will become unstable. Therefore this case is not plotted. The bad condition of the problem results in a worse adaption capability of the LMS. In figure 6 the number of real multiplications is plotted as a function of the channel length L. Three different techniques to compute filter weights are considered: the LMS, the channel estimation in combination with either the CP based solution or with the QR based solution (PI).

4. CONCLUSIONS

Different techniques to compute FIR filter weights in adaptive channel equalization problems are presented. Some of them compute

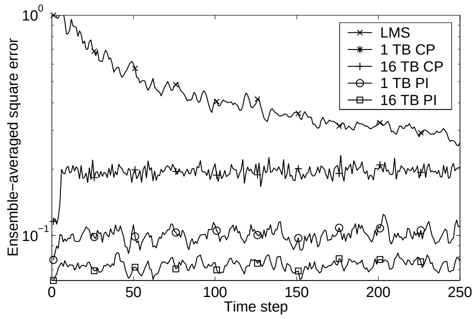

Fig. 5. Ensemble-averaged squared error for Bernoulli training sequence at every time step. Channel distortion W=3.8.

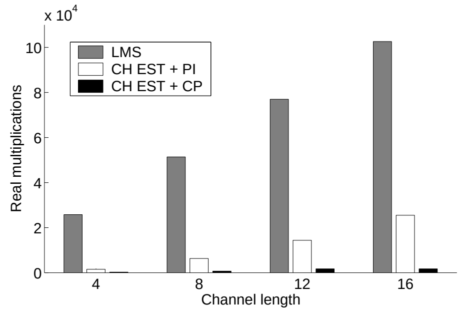

Fig. 6. Real multiplications of the different algorithms, ($I = 200$, $M = 4L$).

the filter weights from training symbols and the received data either directly (DMI) or iteratively (LMS, RLS). Other estimate the CIR in the frequency domain by using a cyclic extension and by exploiting the properties of the EVD of cyclic matrices. Then the CIR is used to compute the filter weights via solving the zero forcing matrix equation. Thereby a CP based approach and a computation of a pseudo inverse via solving a Cholesky down dating problem are considered. The channel estimation in combination with the solution of the zero forcing matrix equation results in significantly lower computational requirements than DMI or LMS techniques. The CP approach results in fewer computations than the pseudo inverse. However this approach might suffer if the cyclic matrix happens to be singular.

5. REFERENCES

[1] 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.

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

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

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

[5] C. V. Sinn, F. Hanschmann, J. Götze, "*Efficient Algorithms Relating OFDM Channel Parameters to Filter Weights in Single-Carrier Systems with Frequency Domain Equalization*," 7^{th} International OFDM Workshop, Hamburg, Germany, September 2002.

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