

On Complexity-Reduced Implementation of Multi-Dimensional Wiener Interpolation Filtering

Huijun Li

Fraunhofer Institute for Telecommunications
Heinrich-Hertz-Institut
Einsteinufer 37, 10587 Berlin, Germany

Andreas Ibing

Heinrich-Hertz-Chair for Mobile Communications
Technische Universität Berlin
Einsteinufer 25, 10587 Berlin, Germany

Abstract—The Wiener Interpolation Filter is commonly used to reconstruct a stochastic process from noisy samples. We focus on the case of a multi-dimensional stochastic process and the practical example of application of the filter for estimation of mobile radio propagation channels at a wireless receiver. We show that computational complexity of the implementation can be considerably reduced by exploiting two properties: first, multidimensional Wiener filtering is in general non-separable, while upsampling for interpolation is separable if the sample structure is a lattice - so it is beneficial to separate the two steps. Second, Wiener filtering can be implemented using spectral shaping of partially overlapping multidimensional blocks (fast convolution, overlap-add or overlap-save method). We discuss performance and complexity of the application to estimate the time-variant channel transfer function in OFDM (2D channel correlation) and MIMO-OFDM (3D channel correlation) transmission, for varying channel autocorrelation values (WSSUS model) and filter kernel sizes.

Index Terms—Wiener filtering, upsampling, interpolation, fast convolution

I. INTRODUCTION

The Wiener filter [1] is the optimal noise reduction filter for a noisy signal (in the minimum mean square error sense) if autocorrelation and noise variance are known. It is used for one-dimensional signals like speech as well as multidimensional signals like images (2D) and video (3D) [2].

Wiener filtering is also commonly applied in wireless receivers for channel estimation. In modern Orthogonal Frequency Division Multiplex (OFDM) transmission the channel is a 2D stochastic process with dimensions of time and frequency (subcarriers). OFDM channel estimation is normally enabled by transmission of pilot symbols which are known to the receiver, so that the channel is sampled at the pilot positions. The receiver needs to reconstruct the channel at the positions of (unknown) data symbols from the received pilot values, which are corrupted by noise. The combination of Wiener filtering and interpolation to solve this problem is known as Wiener Interpolation Filter and has been described for OFDM channel estimation in [3].

The Wiener filter depends on the autocorrelation of the stochastic process and the noise variance. For the mobile radio channel, these parameters depend on location as well as scenario (velocity etc.). Receivers use either a (less complex) static Wiener filter or an adaptive Wiener filter.

For a static Wiener filter the autocorrelation is assumed to be known a priori and normally predefined according to 'worst-case' assumptions (like maximum delay spread and maximum Doppler spread). This means assuming less correlation between pilots than there will be typically, and the approach is referred to as 'robust' filter design – robust against filter mismatch [4]. High-performance receivers on the other hand aim at exploiting as much correlation between pilots as possible for noise reduction, so they track the channel statistics and use adaptive Wiener filtering. One example of parametric tracking of channel statistics and adaptive filtering is [5].

For the more recent case of MIMO transmission over OFDM channels (MIMO-OFDM) it has been pointed out that the channel correlation (under WSSUS assumption) is 3-dimensional [6] with the dimensions time, frequency (subcarriers) and angle (antennas). In [7] the spatial correlation is exploited for improved noise reduction by 3D Wiener filtering (robust static filtering).

In many cases, filtering can be implemented as convolution. For sampled signals, this depends on the sampling grid (which for channel estimation is the pilot grid). [8] showed that convolution of a (multidimensional) sampled signal is possible if the sample grid is a lattice, i.e. if the n -dimensional sample grid is spanned by n basis vectors. In particular, there are lattices corresponding to nonrectangular sampling.

For computationally efficient computation of convolution, fast algorithms based on Fourier transform can be applied. Fast linear convolution can be implemented based on block-wise Fourier transforms (circular convolution) using the 'overlap-save' or 'overlap-add' method [9].

In this paper we present an algorithm which implements the multidimensional Wiener Interpolation Filter with reduced complexity compared to previously published implementations. Performance and complexity are scalable by choice of Wiener filter kernel size.

The remainder of the paper is structured as follows: the next section describes the system model with focus on OFDM channel estimation application. section III shortly reviews the Wiener Interpolation Filter and published implementations. In section IV we present the proposed implementation. Sections V and VI then evaluate performance

and complexity depending on filter kernel size and channel parameters (multidimensional autocorrelation, noise variance) and also compare performance to other filters like concatenated 1D filters. The last section discusses the results and other possible applications.

II. SYSTEM MODEL

For the OFDM channel estimation application we consider an OFDM transmitter with length- N IFFT. Of the N subcarriers, N_{sub} are used for transmission, the other $(N - N_{sub})$ ones are used as guard band for neighbouring systems. With X_k we denote the complex symbol on the k th subcarrier in the frequency domain. The time domain symbol vector $\mathbf{x} = (x_1 \ x_2 \ \dots \ x_N)^T$ is obtained by IFFT of $\mathbf{X} = (X_1 \ X_2 \ \dots \ X_N)^T$, the time domain OFDM symbol \mathbf{x}^{cp} is formed by copying and prepending the last L time domain symbols as cyclic prefix. Without loss of generality (for lattice structure sample grids) we assume a rectangular pilot pattern illustrated in Fig. 1.

The time-variant multipath channel impulse response (CIR) for a SISO system is described as 2D stochastic process [10]:

$$h(t, \tau) = \sum_n \gamma_n(t) \delta(\tau - \tau_n(t)) \quad (1)$$

The channel transfer function $H(t, f)$ is the Fourier Transform of CIR $h(t, \tau)$ with respect to the variable τ :

$$H(t, f) = \mathcal{FT}_{\tau \rightarrow f} \{h(t, \tau)\} = \sum_n \gamma_n(t) e^{-j2\pi f \tau_n} \quad (2)$$

The received signal is denoted $\mathbf{y}^{cp} = \mathbf{x}^{cp} * h(t, \tau) + z(t)$, where $z(t)$ is additive white Gaussian noise. After removing cyclic prefix and performing N -point FFT, the signal in frequency domain is given as

$$Y[n, k] = X[n, k]H[n, k] + Z[n, k] \quad (3)$$

where $Y[n, k]$ is the received symbol on subcarrier n in OFDM symbol k . As initial step (least squares method), the receiver obtains noisy samples of the time-variant channel transfer function at the locations of the known pilot symbols:

$$\tilde{H}_p[n, k] = \frac{Y_p[n, k]}{X_p[n, k]} = H_p[n, k] + \frac{Z_p[n, k]}{X_p[n, k]} \quad (4)$$

III. REVIEW OF WIENER INTERPOLATION FILTER

Noise suppression filtering based on the initial least squares estimates can be written as

$$\hat{H}_p[n, k] = \sum_{n_0, k_0} \mathbf{c}^*[n, k] \tilde{H}_p[n - n_0, k - k_0] \quad (5)$$

where $\mathbf{c}[n, k]$ are filter coefficients. The estimation Mean Square Error

$$MSE(\mathbf{c}) = E\{\hat{H}_p[n, k] - H_p[n, k]\} \quad (6)$$

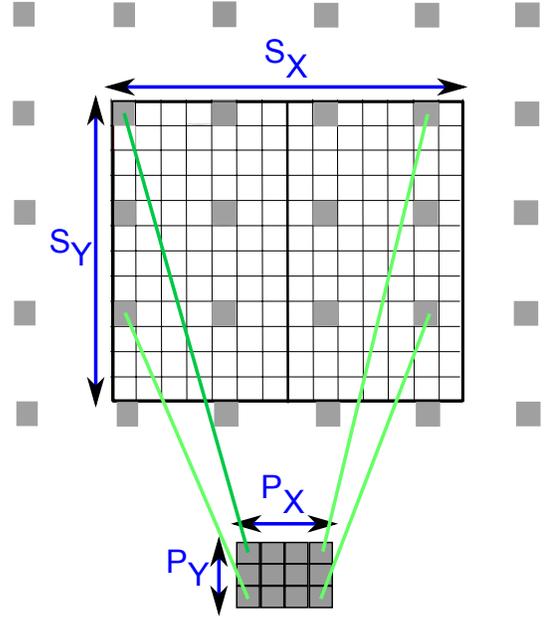


Fig. 1. Example pilot lattice. To estimate an area of $S_x \times S_y$ values, we apply noise reduction filtering only on the $P_x \times P_y$ pilots and upsample afterwards.

becomes minimal for the coefficients satisfying the Wiener-Hopf-Equation (follows from principle of orthogonality [3], [11]):

$$\mathbf{c}[n, k]|_{opt} = (\mathbf{R}[n, k] + \frac{\mathbf{I}}{SNR})^{-1} \mathbf{r}[n, k] \quad (7)$$

where $\mathbf{r}[n, k]$ is the crosscorrelation between the channel transfer value $H[n, k]$ at the position to be computed and the pilots within filter range (values written as vector). $\mathbf{R}[n, k]$ is the autocorrelation (values written as matrix) between $H[n, k]$ at all the pilots within filter range.

The OFDM channel transfer function's 2D correlation function $r_H(\Delta t, \Delta f) = E\{H(t_1, f_1)H^*(t_2, f_2)\}$ can be separated into time and frequency parts:

$$r_H(\Delta t, \Delta f) = r_H(\Delta t)r_H(\Delta f) \quad (8)$$

Static 'robust' filtering normally assumes a uniform multidimensional power density spectrum (for MIMO-OFDM with the dimensions delay spread, Doppler spread and angular spread [4], [7]). From the power spectrum assumption, the channel autocorrelation follows by IFFT [9]. For the uniform spectrum case the correlation in time direction becomes:

$$r_{H,unif}(\Delta t) = \text{sinc}(2\pi f_D \Delta t) \quad (9)$$

with $\text{sinc}(x) = \text{sin}(x)/x$, and f_D being the maximum Doppler shift. $\Delta t = \Delta k t_s (L + N)$ with t_s being the sampling period and $t_s(L + N)$ one OFDM symbol duration. The correlation in frequency direction:

$$r_{H,unif}(\Delta f) = \text{sinc}(\pi T_m \Delta f) e^{j2\pi \tau_{shift} \Delta f} \quad (10)$$

with $T_m = L t_s$ as the maximum delay, $\tau_{shift} = T_m/2$, $\Delta f = \Delta F \Delta l$ with ΔF being the subcarrier distance. With

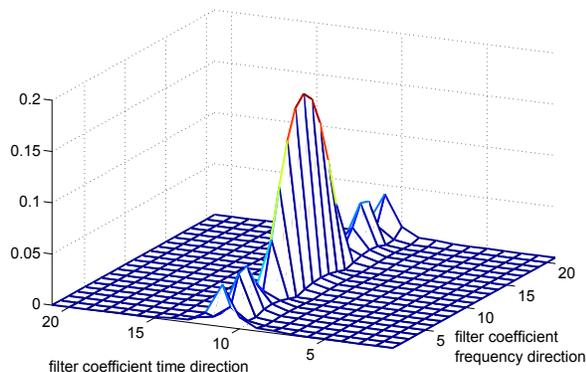


Fig. 2. 2D convolution filter kernel magnitude for filter size 21×21 , maximum Doppler shift 200Hz and 20dB SNR, urban macro channel model [14]. High Doppler spread means little channel correlation in time direction.

the distances $[\Delta k \ \Delta l]$ from pilots inside filter range, the coefficients can be computed from (8) and (7).

Published implementations of the Wiener Interpolation Filter jointly perform noise reduction and interpolation and are based on matrix multiplication [3], [12], [13]. [3] searches for each data symbol position for the n closest pilot locations based on Euclidean distance or a weighted distance measure (to select n pilots with large correlation values). The filter coefficients are not position independent (they differ depending on data symbol position). The estimator is hence shift-variant [3]. For complexity reduction, the number n of pilots involved in filtering can be chosen small. For static filtering, coefficients can be precomputed (to avoid correlation matrix inversion). Also, the 2D estimator can be approximated by a concatenation of two 1D filters [3] (this entails performance loss because while the channel autocorrelation function is separable according to model assumptions, the resulting filter is not).

[12] considers 1D filtering (in frequency direction) and achieves complexity reduction (with performance loss) by 'Optimal Low Rank' filtering (OLR-MMSE), which is a dimension reduction by projection of pilot values onto a subspace found by singular value decomposition (SVD), before filtering is performed by matrix multiplication. For static filtering, the SVD can be precomputed.

[13] applies 1D Wiener Interpolation filtering (frequency direction) by matrix multiplication in a block-based way to reduce complexity (with performance loss): the matrix for one small block is precomputed (static filtering), and to reduce edge distortion the blocks of pilots overlap.

IV. PROPOSED IMPLEMENTATION

We separate noise reduction filtering from interpolation and perform them sequentially. Since the sample grid is assumed to be a lattice, multidimensional Wiener filtering only on pilot samples becomes shift-invariant and can be implemented by convolution (compare Fig. 1). We apply fast linear convolution by overlap-save (or overlap-add) method,

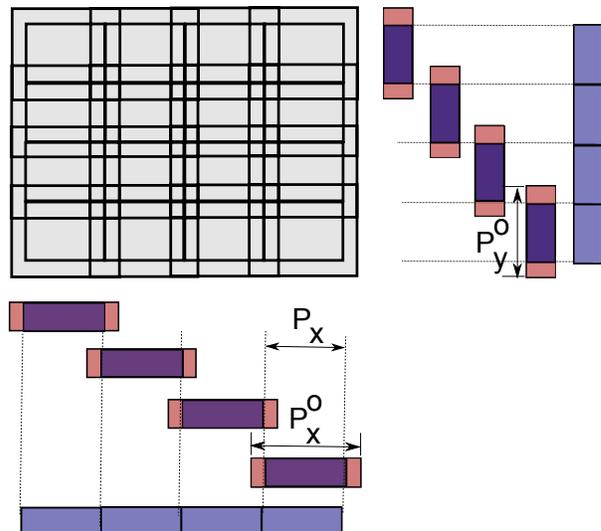


Fig. 3. Illustration of 2D application of overlap-save method.

which is based on a block-wise application of the circular convolution theorem [9]:

$$b[n] * c[n] = \mathcal{F}^{-1}\{\mathcal{F}(b[n]) \cdot \mathcal{F}(c[n])\} \quad (11)$$

For computation of linear convolution by this method, zero-padding is used to guard against the overlap error compared to cyclic convolution. Two-dimensional application of the overlap-save method is illustrated in Fig. 3. Magnitude of a 2D complex Wiener filter kernel is illustrated in Fig. 2. For static filtering, the multidimensional FFT of the filter kernel can be precomputed.

In a second step, multidimensional upsampling interpolation is performed. This operation is separable [2], so it can be performed sequentially for each dimension. Implementation consists of upsampling (zero stuffing) followed by 1D lowpass filtering, where we again use fast convolution and a precomputed FFT of the filter kernel. Upsampling is:

$$H_{up}(n) = \begin{cases} H(\frac{n}{L_{up}}) & \text{when } L_{up} \text{ divides } n \\ 0 & \text{else} \end{cases} \quad (12)$$

For an upsampling factor of L_{up} (in Fig. 1 L_{up} is 4), the cutoff frequency of the lowpass filter is π/L_{up} .

V. PERFORMANCE

We evaluate filter performance simulatively by measuring channel estimation mean square error for different channel autocorrelation and noise parameters. The underlying channel model we use is the 'urban macro' type from [14]. Parameters of OFDM transmission are FFT length 2048, of which 1200 subcarriers are used, and 15kHz subcarrier distance (20MHz bandwidth). The assumed periodically extended pilot grid is shown in Fig. 1. The gain of increasing filter kernel size is illustrated in Fig. 4, where the resulting MSE is shown over channel SNR. Fig.5 shows a comparison of a 2D filter with with different filter types, namely a 1D

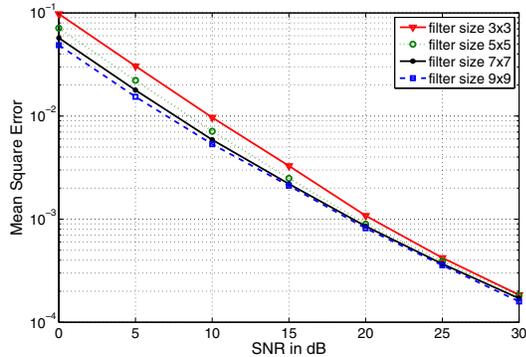


Fig. 4. Channel estimation performance for varying 2D filter size (urban macro channel, Jakes Doppler spectrum with 72Hz max. Doppler shift).

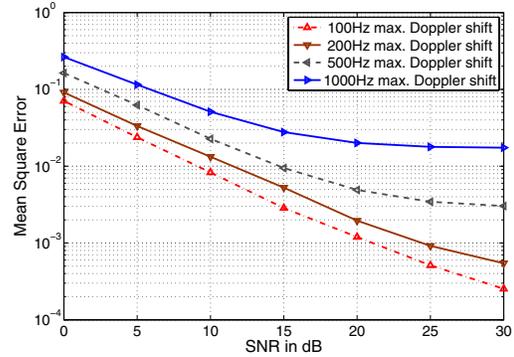


Fig. 6. Influence of different Doppler spread (Jakes spectrum) on performance of 2D filter with 7×7 coefficients.

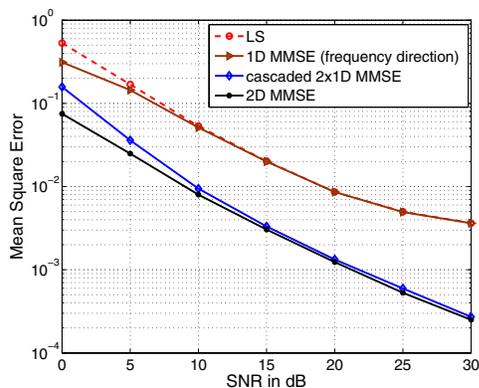


Fig. 5. Performance of different filter types. All filters except LS have the same number of 49 filter coefficients (urban macro channel, max. Doppler shift 100Hz).

filter in frequency direction and a cascaded 2x 1D filter (separable, applied first in frequency direction and then in time direction). All three filters use the same amount of pilots for Wiener filtering. The effect of degrading performance with increasing terminal velocity (different max. Doppler shift) is illustrated in Fig. 6. The filter size has been chosen odd in the examples for the convenience of having a symmetric filter kernel. The filter size does not equal the processing block size, which is detailed in the next section.

VI. COMPLEXITY

We compare the complexity of the proposed implementation for two dimensions with an implementation following [3], [13], where a precomputed filter matrix is used (static filter or adaptive selection of one of several precomputed filters). We assume block-wise filtering, where we estimate the channel coefficients for a block of $S_x \times S_y$ symbols, which contains $P_x \times P_y$ pilots. We compare the number of complex multiplications necessary per estimated sample, because in signal processors additions tied to multiplications are most often computed 'for free' by means of *multiply-accumulate* instructions.

The normal implementation is a linear mapping (matrix multiplication) of the block's pilot values onto all symbols of the block. Following [13], the mapping includes a few pilots from neighbouring blocks, to avoid poor performance at the edges. Thus, the effective 2D blocks overlap at the edges: instead of using only the $P_x \times P_y$ pilots, the matrix is a mapping from $P_x^o \times P_y^o$ pilots onto the $S_x \times S_y$ symbols and therefore needs

$$C_{\text{WIF}}^{\text{normal}} = P_x^o P_y^o \quad (13)$$

multiplications per estimated sample.

The proposed implementation filters on blocks of pilot symbols with the same block size $P_x^o \times P_y^o$ (including overlap, but not including data positions). Multidimensional Fourier transformation is separable [2]. For a block area (with overlap) including $2^m \times 2^k$ pilots, the 2D-FFT of the block's pilots needs $P_y^o P_x^o \log_2 P_x^o + P_x^o P_y^o \log_2 P_y^o$ multiplications (Cooley-Tukey implementation [15]). If the block dimensions are not chosen as power-of-2 values, the prime factor FFT can be applied [16]. The FFT of the filter kernel is assumed to be precomputed, where zeros were stuffed at the edges to fill the block size and to avoid wrap-around. After transformation, $P_x^o P_y^o$ multiplications are needed in Fourier domain. IFFT needs the same number of multiplications as FFT, so for pilot filtering complexity we have in sum:

$$C_{\text{filter}} = \frac{2(P_y^o P_x^o \log_2 P_x^o + P_x^o P_y^o \log_2 P_y^o) + P_x^o P_y^o}{S_x S_y} \quad (14)$$

Upsampling interpolation by zero stuffing and lowpass filtering is performed sequentially for the dimensions. The lowpass can be implemented by 1D fast convolution. Depending on the size of the lowpass filter kernel, the overlap can be different than for previous Wiener filtering. We assume that an area of $P_x^u \times P_y^u$ pilots (including overlap for convolution) is interpolated to a resolution of $S_x^u \times S_y^u$ samples (also including overlap), first in y-direction with

$$C_{\text{upsample-y}} = \frac{2P_x^u S_y^u \log_2 S_y^u + P_x^u S_y^u}{S_x S_y} \quad (15)$$

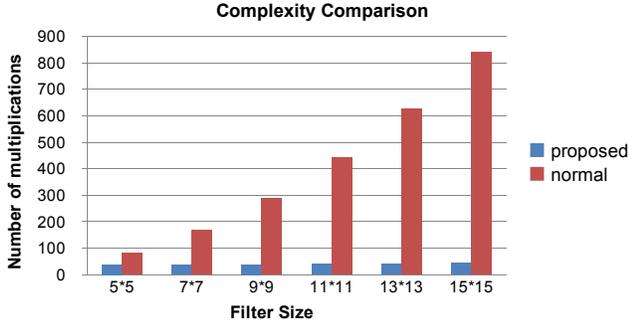


Fig. 7. Complexity comparison for the example pilot grid (Fig. 1). The proposed implementation reduces absolute complexity as well as complexity growth order with respect to filter size.

multiplications, then in x-direction with

$$C_{\text{upsample-x}} = \frac{2S_y S_x^u \log 2S_x^u + S_y S_x^u}{S_x S_y} \quad (16)$$

multiplications. For the proposed Wiener interpolation filter implementation we yield as multiplications per output sample:

$$C_{\text{WIF}}^{\text{proposed}} = C_{\text{filter}} + C_{\text{upsample-y}} + C_{\text{upsample-x}} \quad (17)$$

Using the same pilot grid as in Sec. V, a comparison of the complexity of normal and proposed implementation for different filter sizes is shown in Fig. 7. Due to the different complexity growth orders, the gain of the proposed implementation quickly grows with larger filter size. A breakdown of the complexity of the proposed implementation into the three contributing parts from Eq. (17) is shown in Fig. 8 for different filter sizes.

VII. DISCUSSION

Modern wireless communication systems supporting broadband transmission at high velocity with low pilot overhead clearly need high-performance channel estimation. This means exploiting as much pilot correlation as possible with the most adequate noise reduction filter. For MIMO-OFDM transmission, pilot correlation is three-dimensional. With a growing number of antennas, a 3D fast convolution and upsampling based implementation becomes interesting. The complexity reduction gain would be even higher than illustrated in the examples in this paper, due to the larger difference in complexity growth orders. The choice of block length in time direction needs to consider channel estimation delay (for both the normal and the proposed implementation). In the presented examples, the complexity of the proposed WIF implementation was mainly determined by upsampling interpolation (Fig. 8). This is due to the fact that we chose an upsampling interpolation filter suited for general-purpose signal processors. In case of hardware acceleration by a specialized co-processor, further complexity reduction is possible by using a different lowpass filter. With a cascaded-integrator-comb (CIC) filter, upsampling interpolation is possible without any multiplications, only

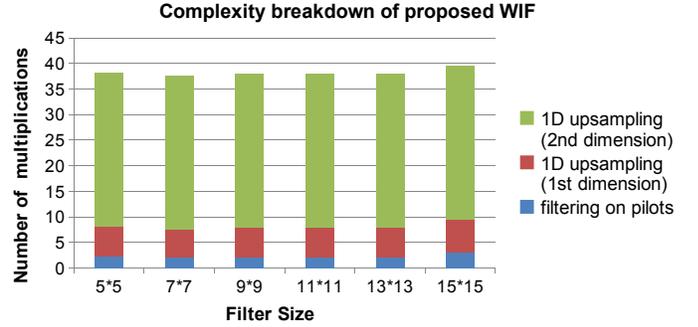


Fig. 8. Complexity of the proposed implementation in the example is mainly determined by upsampling interpolation. Further complexity scaling is possible by choice of the 1D lowpass filter.

using additions [9]. A combination of the proposed implementation with the matrix multiplication based one is also possible: edge blocks where there are no pilots from neighbouring blocks available, could be estimated using a precomputed matrix (at the edges of frequency band or at the edges of resource allocations in frequency direction for OFDMA uplink). The limitation of filter kernel size can employ an appropriately smooth windowing function (e.g. Hamming window [9]). The filter kernel size can be chosen adaptive to delay spread and Doppler spread (adaptive filtering, e.g. rectangular but non-quadratic filter kernel).

REFERENCES

- [1] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*. Wiley, 1949.
- [2] J. Woods, *Multidimensional Signal, Image, and Video Processing and Coding*. Academic Press, 2006.
- [3] P. Hoehner, S. Kaiser, and P. Robertson, "Two-dimensional pilot-symbol-aided channel estimation by wiener filtering," in *International Conference on Acoustics, Speech and Signal Processing*.
- [4] V. Srivastava, C. Ho, P. Fung, and S. Sun, "Robust MMSE channel estimation in OFDM systems with practical timing synchronization," in *IEEE WCNC*, 2004.
- [5] K. Jeong and J. Lee, "Low complexity channel tracking for adaptive MMSE channel estimation in OFDM," in *IEEE CISS*, 2007.
- [6] B. Fleury, "First- and second-order characterization of direction dispersion and space selectivity in the radio channel," *IEEE Trans. Information Theory*, vol. 46, no. 6, pp. 2027–2044, 2000.
- [7] G. Auer, "3d pilot aided channel estimation," in *IEEE WCNC*, 2009.
- [8] R. Mersereau and T. Speake, "The processing of periodically sampled multidimensional signals," *IEEE Trans. ASSP*, 1983.
- [9] A. Oppenheim and R. Schaffer, *Discrete-Time Signal Processing*, 3rd ed. Prentice Hall, 2009.
- [10] P. Bello, "Characterization of randomly time-variant linear channels," *IEEE Trans. Communication Systems*, vol. 11, no. 4, pp. 360–393, 1963.
- [11] S. Haykin, *Adaptive Filter Theory*.
- [12] O. Edfors, M. Sandell, J. Beek, S. Wilson, and P. Börjesson, "Ofdm channel estimation by singular value decomposition," *IEEE Trans. Communications*, vol. 46, no. 7, 1998.
- [13] S. Schiffermüller and V. Jungnickel, "Practical channel interpolation for OFDMA," in *IEEE GLOBECOM*, 2006.
- [14] TSG RAN: TR25.814v7.1.0 Physical layer aspects for Evolved Universal Terrestrial Radio Access, 3GPP, Sept 2006. [Online]. Available: <http://www.3gpp.org/ftp/Specs/html-info/25814.htm>
- [15] J. Cooley and J. Tukey, "An algorithm for the machine calculation of complex fourier series," *Math. Comput.*, vol. 19, pp. 297–301, 1965.
- [16] I. Good, "The interaction algorithm and practical fourier analysis," *J. R. Statist. Soc.*, vol. 20, no. 2, pp. 361–372, 1958.