Improved Detected Data Processing for Decision-Directed Tracking of MIMO Channels

Emna Eitel and Joachim Speidel
Institute of Telecommunications, University of Stuttgart, Germany

***Abstract*—This paper addresses the issue of how to process detected data vectors for decision-directed MIMO channel tracking. We differentiate between a serial processing where the decision-directed estimates are computed after each vector has been detected and a parallel one where a block of detected vectors is processed at once which necessitates a matrix inversion. The probability of existence of a matrix inverse is thoroughly investigated. Parallel processing is improved by deriving the optimal block size. A comparison shows that serial processing performs better in case of fast channel variations.**

I. Introduction

MIMO systems with coherent detection can deliver high channel capacity provided that an accurate knowledge of the channel is available at the receiver. The performance can be even increased if the channel state information (CSI) is also known at the transmitter. Algorithms to precisely estimate the CSI are therefore crucial. Often periodical pilot-assisted channel estimation (PACE) is employed. However, in fast time-varying channels, PACE requires more pilots to follow the fast channel variations which decreases the bandwidth efficiency.

A method, that does not require additional pilots is decision-directed channel estimation (DDCE). It considers the recovered data as "new pilots" and can therefore feed the channel estimation module with data that permanently takes account of the actual channel state. The advantage of this method in comparison to simple interpolation, Wiener or Kalman filtering is that DDCE neither introduces a delay nor requires any knowledge about the statistical properties of the channel except for the channel coherence time. Furthermore, DDCE is robust to numerical errors and exhibits a small computational complexity. However, the main drawback of DDCE is its sensitivity to wrongly detected data which may cause error propagation, especially at low SNR. Therefore, we combine DDCE with periodical PACE as also described in [1] and [2].

In [3] we analyze different filtering techniques to suppress the noise contained in the decision-directed channel estimates. In order to avoid numerical instabilities of the optimal maximum a posteriori filter [4][5], a moving-average filter as proposed in [6] is adopted. It is also shown that an optimized filter design with respect to the filter length and appropriate use of PACE information for the filter initialization yields notable performance improvement. In [3], the detected vectors are processed one by one as soon as they are detected, i.e. serially. The blockwise processing is not used as it requires the inversion of a matrix whose invertibility is not always assured depending on the data.

To apply blockwise processing, it is therefore important to investigate the probability of existence of a matrix inverse. This issue is addressed in this paper as, to the best of our knowledge, it has not been yet done in the literature so far. In previous works dealing with DDCE, either the serial processing was adopted ([6], [4]), or the parallel under the assumption that the matrix inversion is guaranteed. However, depending on the transmit (tx) data, the matrix to invert can be rank-deficient which results in numerical instabilities. Simulations show that the probabilities of invertibility built upon detected data vectors remain below the invertibility rates of the tx ones. For the parallel processing, an optimal block size is derived, eventhough we expect that large block sizes stand in favour of high invertibility. The performance of parallel processing is investigated next. If a matrix inversion is not possible, the previous channel estimate is used. A comparison shows that serial processing is more adequate especially for fast channel variations at high SNR, whereas the parallel performs better for low SNR and slow fading.

II. System Model

We consider an $M \times N$ MIMO system. The $N \times 1$ receive signal vector at time instant k is given by:

$$\mathbf{y}(k) = \mathbf{H}(k)\mathbf{s}(k) + \mathbf{n}(k) \tag{1}$$

where $\mathbf{s}(k)$ denotes the $M \times 1$ tx signal vector, $\mathbf{H}(k)$ the $N \times M$ channel matrix and $\mathbf{n}(k)$ the $N \times 1$ additive white Gaussian noise (AWGN) vector whose complex elements are i.i.d and $CN(0, 2\sigma_0^2)$. One element $h_{ij}(k)$ of $\mathbf{H}(k)$ represents the channel coefficient between the *j*th transmit and *i*th receive antenna and is $CN(0,1)$ distributed. $h_{ij}(k)$ realizes a frequency-flat Rayleigh fading in the base-band with the temporal autocorrelation function

$$E\left\{h_{ij}(k)h_{ij}(k')^*\right\} = J_0(2\pi f_d(k-k')), \tag{2}$$

where f_d is the normalized maximum Doppler frequency and J_0 is the Bessel function of first kind and order zero. As shown in 1, in order to estimate the channel at the receiver, orthogonal pilot symbol vectors $\mathbf{s}_p$ are periodically transmitted during the training period that takes L_p symbol periods T_s. The training phase is followed by a data transmission phase of duration $L_d T_s$. At the end of the training phase, a channel estimate $\widehat{\mathbf{H}}$ is computed by means of the received pilots. In the absence of tracking, PACE is used for coherent detection of data during the subsequent data phase. We introduce the parameter τ to denote the time elapsed since the last PACE. In the following,

we use the minimal required number of pilots, i.e. $L_p = M$, in order to keep high bandwidth efficiency.

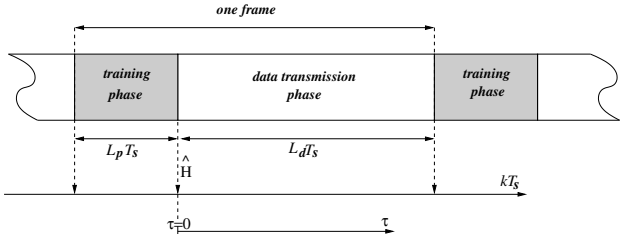


Fig. 1. Periodically alternating training and data transmission phases

III. Computing the Raw Channel Estimates

In the tracking phase, new channel coefficients are estimated based on detected symbol vectors $\hat{\mathbf{s}}$. These coefficients are denoted as raw channel estimates, since they have not been filtered yet[1]. We distinguish between serial and parallel processing as depicted in 2. In case of serial processing, the computation of the raw estimates is done at each time instant as soon as one symbol vector is detected. For the parallel processing, a block of detected vectors is processed at once.

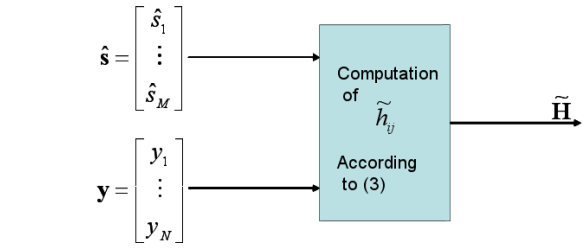
A. Serial Processing

The raw estimates are computed according to:

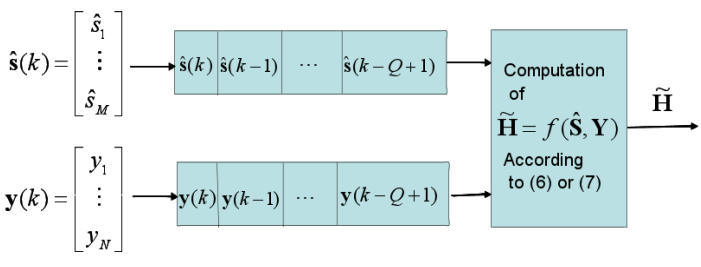
$$\tilde{h}_{ij}(k+1) = \frac{y_i(k) - \sum_{q=1,q\neq j}^{M} \tilde{h}_{iq}(k)\hat{s}_q(k)}{\hat{s}_j(k)} \tag{3}$$

At a given time instant k, applying (3) for all $j = 1, \ldots, M$ simultaneously induces the assumption that some previous channel estimates are still valid since it is otherwise not possible to compute $M \cdot N$ unknown variables upon one channel observation. Note that this assumption requires that the channel does not change significantly within one symbol period, which is a less severe assumption in comparison with the one required for parallel processing (see next section).

[1]Optimized filtering of the raw channel estimates comprising better use of the PACE estimates is considered in [3] and will not be described here.



(a) serial processing



(b) parallel processing

Fig. 2. Flow diagram for the serial and parallel processing

B. Parallel Processing

At the beginning, we wait until a block of Q data vectors is detected, such that Q observations are available. Afterwards, we adopt a sliding scheme. To be able to determine all $M \cdot N$ unknown variables, Q must satisfy $Q \geq M$. Assuming that the channel matrix does not change in the interval of Q symbol periods and the detected symbols are identical to the tx ones, we can write the following:

$$\mathbf{Y} = \mathbf{H}\hat{\mathbf{S}} + \mathbf{N} \tag{4}$$

where $\hat{\mathbf{S}} = \begin{bmatrix} \hat{\mathbf{s}}_0 & \hat{\mathbf{s}}_1 & \cdots & \hat{\mathbf{s}}_{Q-1} \end{bmatrix}$ and $\mathbf{Y} = \begin{bmatrix} \mathbf{y}_0 & \mathbf{y}_1 & \cdots & \mathbf{y}_{Q-1} \end{bmatrix}$ are the matrices of the last Q detected and received vectors, respectively. $\mathbf{N} = \begin{bmatrix} \mathbf{n}_0 & \mathbf{n}_1 & \cdots & \mathbf{n}_{Q-1} \end{bmatrix}$ is the corresponding AWGN matrix.

To compute the raw decision-directed estimate $\tilde{\mathbf{H}}$, we can search for the maximum a posteriori (MAP) channel estimate $\tilde{\mathbf{H}}_{MAP}$ that maximizes the probability density function (pdf) $p(\tilde{\mathbf{H}}|\mathbf{Y}, \hat{\mathbf{S}})$. Therefore, we rewrite (4) in vector form to apply standard results from estimation theory:

$$\underbrace{vec(\mathbf{Y})}_{\tilde{\mathbf{y}}} = \underbrace{\left(\hat{\mathbf{S}}^T \otimes \mathbf{I}\right)}_{\mathbf{X}} \cdot \underbrace{vec(\mathbf{H})}_{\mathbf{h}} + \underbrace{vec(\mathbf{N})}_{\tilde{\mathbf{n}}} \tag{5}$$

where $\otimes$ is the Kronecker product. The MAP estimate is given by the well-known equation [7]:

$$\tilde{\mathbf{h}}_{MAP} = \mathbf{R}_{\mathbf{hh}}\mathbf{X}^H \left(\mathbf{X}\mathbf{R}_{\mathbf{hh}}\mathbf{X}^H + \mathbf{R}_{\tilde{\mathbf{n}}\tilde{\mathbf{n}}}\right)^{-1} \tilde{\mathbf{y}} \tag{6}$$

where $\tilde{\mathbf{h}}_{MAP} = vec\left\{\tilde{\mathbf{H}}_{MAP}\right\}$, $(.)^H$ refers to the Hermitian of a matrix, $\mathbf{R}_{\mathbf{hh}} = E\left\{\mathbf{h}\mathbf{h}^H\right\}$ and $\mathbf{R}_{\tilde{\mathbf{n}}\tilde{\mathbf{n}}} = E\left\{\tilde{\mathbf{n}}\tilde{\mathbf{n}}^H\right\}$. Alternatively, we can compute the least minimum square (LMS) estimate $\tilde{\mathbf{H}}_{LMS}$ that minimizes the mean square error $\left|\tilde{\mathbf{H}} - \mathbf{H}\right|^2$.

$$\tilde{\mathbf{H}}_{LMS} = \mathbf{Y} \cdot \hat{\mathbf{S}}^H \cdot \left(\hat{\mathbf{S}}\hat{\mathbf{S}}^H\right)^{-1} \tag{7}$$

As can be seen from (6) and (7), both solutions require a matrix inversion. This results in two disadvantages. First, since in (6) and in (7) the dimensions of the matrices to invert are $QN \times QN$ and $M \times M$ respectively, the complexity of parallel processing is $O((QN)^3)$ and $O(M^3)$ and therefore higher than that of serial processing. Second, the invertibility of these matrices depends strongly on the tx data, which is very undesirable.

Obviously, the performance of parallel processing depends tightly on the probability P_{inv} that a stable matrix inverse can be computed. We adopt two criteria for the invertibility, the determinant and the condition number. For the LMS estimate, only the determinant is used since it takes on integer values that can either strictly equal zero in case the matrix is rank-deficient, or be different from zero otherwise. This holds also for the reciprocal condition number which takes on integer values. In case of MAP estimation, both determinant and condition number can take on real values depending on the noise variance and the detected data. These values can be arbitrarily small resulting in an ill-conditioning of the matrix to invert. Therefore, to decide on the invertibility of a matrix, we fix a threshold for the determinant or the condition number.

1) LMS Estimation: In the following, we focus on the LMS estimate, i.e. we investigate whether the matrix $\hat{\mathbf{S}}\hat{\mathbf{S}}^H$ in (7) is invertible or not. Intuitively, we expect that the larger the matrix dimension, the higher the probability that the rank is full. This reasoning is confirmed by the matrix invertibility rate results in 3 for all possible $\mathbf{S}\mathbf{S}^H$. Therein, we restrict ourselves to BPSK and two transmit antennas as the number of possible matrices raises exponentially with M, Q, and the order of the applied QAM.

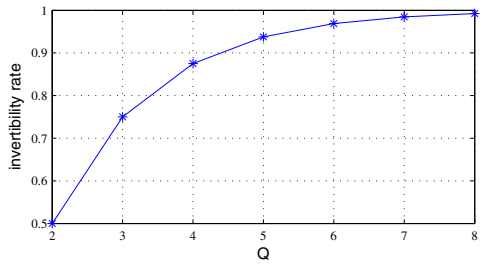


Fig. 3. Invertibility rate of $\mathbf{S}\mathbf{S}^H$ as function of Q for BPSK

To explain this, let us first recall a well-known property of random matrices. Given an $M \times M$ random matrix generated by complex-valued uniformly distributed random variables. The determinant of this random matrix is a random variable with a continuous pdf over a certain domain. Thus the probability that the determinant takes a specific value, namely zero, equals zero.

An element of $\hat{\mathbf{S}}\hat{\mathbf{S}}^H$ is given by $b_{ij} = \sum_{l=1}^{Q} \hat{s}_{il}\hat{s}_{jl}^*$. Let N_a be the cardinality of the set of possible values $\hat{s}_{il}\hat{s}_{jl}^*$ for an arbitrary QAM constellation, then b_{ij} can take on $\begin{pmatrix} N_a + Q - 1 \\ Q \end{pmatrix}$ different values. For an antipodal BPSK, $\hat{s}_{il}\hat{s}_{jl}^* \in \{+1, -1\}$ holds, i.e. $N_a = 2$. If for example $Q = 3$, then $b_{ij} \in \{ +1, \quad -1, \quad +3, \quad -3 \}$, i.e we have 4 different values. For this example, we get $\lim_{Q \to \infty} \begin{pmatrix} N_a + Q - 1 \\ Q \end{pmatrix} = \infty$. Consequently, if we increase Q, we can approximate the pdf of b_{ij} by a continuous uniform function on its definition interval. Referring to the afore-mentioned property of random matrices, we conclude that the matrix $\hat{\mathbf{S}}\hat{\mathbf{S}}^H$ is invertible with probability 1 if Q is large enough.

We also investigate the probability P_{inv} that $\hat{\mathbf{S}}\hat{\mathbf{S}}^H$ is invertible by simulation for BPSK and $M = 2$. The results show that P_{inv} strongly depends on various system parameters, which we discuss in the following.

In order to analyze the impact of the SNR exclusively, we evaluate P_{inv} for a system with perfect CSI. As expected, P_{inv} of $\hat{\mathbf{S}}\hat{\mathbf{S}}^H$ corresponds to that of $\mathbf{S}\mathbf{S}^H$ in 3 for large SNR as the detection errors become scarcer, i.e. $\mathbf{S} \to \hat{\mathbf{S}}$. As shown in 4, at low SNR P_{inv} decreases in a way that depends on the detection scheme, here zero-forcing (ZF) or ordered successive interference cancellation (OSIC) MMSE. The more sophisticated the detection scheme, the higher the invertibility.

Impact of the normalized Doppler frequency f_d on P_{inv} can be seen in 5 for $L_d = 30$ and ZF receiver. Again, we see that P_{inv} increases for large SNR and $f_d = 0.001$ which is reversed in case of $f_d = 0.02$. This can be explained as

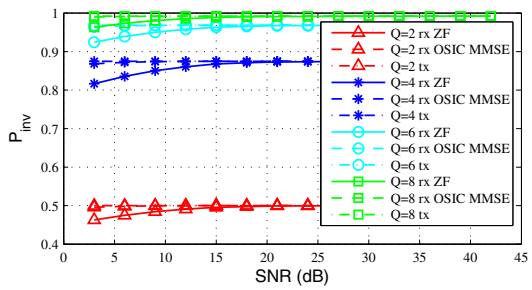


Fig. 4. P_{inv} as a function of SNR for tx data as well as detected data (rx) with ZF and OSIC MMSE receivers

follows. For low Doppler frequencies the BER decreases with increasing SNR and the detected matrices match the tx ones more frequently such that the corresponding P_{inv} become identical. If f_d is high, the channel estimation errors are large and the BER does not improve significantly with an increasing SNR (high BER error floor). $\hat{\mathbf{S}}\hat{\mathbf{S}}^H$ differs from $\mathbf{S}\mathbf{S}^H$. Because of AWGN, the occurring matrices $\hat{\mathbf{S}}\hat{\mathbf{S}}^H$ are still stochastic such that their invertibility is high. This effect is reversed in the high SNR regime, where the constant channel estimation errors are dominant.

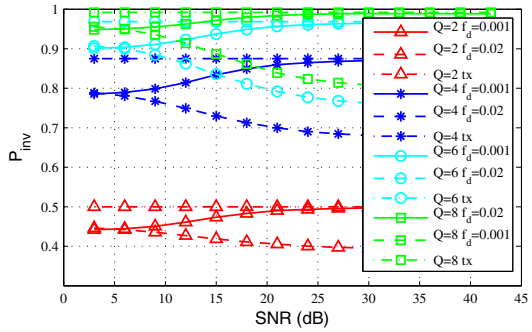


Fig. 5. P_{inv} as function of SNR for different Doppler frequencies f_d

Impact of the data interval length L_d on P_{inv} is shown in 6. P_{inv} increases with a decreasing L_d in the high SNR range, since in this case more correct detections are achieved. At low SNR, the large channel estimation error for longer L_d in combination with a high noise level result in a higher invertibility. In general, we see that P_{inv} is higher for $\mathbf{S}\mathbf{S}^H$

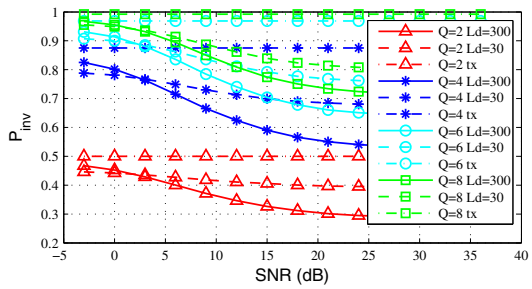


Fig. 6. P_{inv} as function of SNR at $f_d = 10^{-3}$ for different L_d

than for $\hat{\mathbf{S}}\hat{\mathbf{S}}^H$. We conclude first that detection errors generally decrease P_{inv}. P_{inv} of the tx matrices, as for example in 3

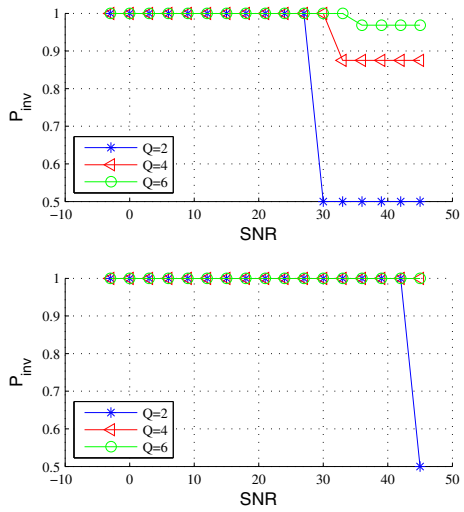


Fig. 7. Impact of determinant threshold on probability of invertibility. threshold = 10^{-4} (top), threshold = 10^{-7} (bottom)

for BPSK and $M = 2$, can be viewed as an upper bound for the probability of invertibility of $\hat{\mathbf{S}}\hat{\mathbf{S}}^H$.

2) MAP Estimation: Applying MAP estimation, the dependency of P_{inv} on different system parameters shows similar behaviour as for LMS estimation. It is furthermore well-known that MAP and LMS channel estimation perform comparably for uncorrelated MIMO channels and high SNRs. We prefer therefore to focus on the impact of the fixed threshold for evaluating the matrix invertibility on P_{inv}. From 7, we can see that the larger the threshold, the smaller the SNR beyond which P_{inv} reaches P_{inv} of the tx matrices. Second, the consideration of P_{inv} confirms the problem of invertibility we are confronted with in case of parallel processing. In case of $Q = 3$, we can compute a matrix inverse in only at most 75% of the cases.

C. Optimal Block Size for Parallel Processing

Now, Q shall be optimized. Although we expect that we have to choose Q as large as possible, the simulation results show that there exists numerically an optimum. This can be explained by deriving the channel estimation MSE. Due to space constraints, we only consider the MSE of the LMS estimator in (7), defined as follows

$$\zeta(\tau) = \frac{1}{MN} E\left\{tr\left\{vec(\mathbf{E}(\tau)) \cdot (vec(\mathbf{E}(\tau)))^H\right\}\right\} \quad (8)$$

where $\mathbf{E}(\tau) = \mathbf{H}(\tau) - \tilde{\mathbf{H}}(\tau)$ and the expectation is taken over channel, noise, and data.

Under the assumption of correctly detected symbol vectors $\hat{\mathbf{s}}_i$, spatially uncorrelated MIMO channel and the channel temporal correlation in (2), we first evaluate the MSE taking the expectation over channel and noise for a given data sequence $\hat{\mathbf{S}}$. Therefore, we obtain a data-dependent MSE $\zeta(\tau, \hat{\mathbf{S}})$. When $\zeta(\tau, \hat{\mathbf{S}})$ is averaged over all possible detected matrices $\hat{\mathbf{S}}$, it equals $\zeta(\tau)$. The following can be derived:

$$\zeta(\tau, \hat{\mathbf{S}}) = \zeta_1(\tau, \hat{\mathbf{S}}) + \zeta_2(\tau, \hat{\mathbf{S}}) \quad (9)$$

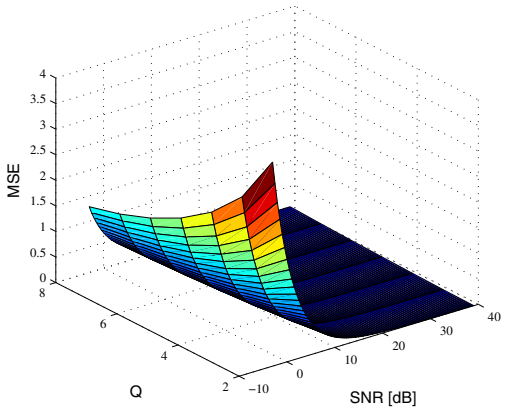


Fig. 8. Channel estimation MSE $\zeta(0)$ for $f_d = 0.02$ as function of Q and the SNR

where

$$\zeta_1(\tau, \hat{\mathbf{S}}) = \frac{1}{MN} tr[(\mathbf{B}^T\mathbf{B}^* \otimes \mathbf{I}) \cdot \quad (10)$$

$$\sum_{i=1}^{Q}\sum_{j=1}^{Q} \xi(i,j,\tau)(\hat{\mathbf{s}}_i^*\hat{\mathbf{s}}_i^T\hat{\mathbf{s}}_j^*\hat{\mathbf{s}}_j^T \otimes \mathbf{I})] \quad (11)$$

with $\mathbf{B} = (\hat{\mathbf{S}}\hat{\mathbf{S}}^H)^{-1}$, $\xi(i,j,\tau) = 1 + J_0(2\pi f_d(i-j)) - J_0(2\pi f_d(\tau + Q - i)) - J_0(2\pi f_d(\tau + Q - j))$ and

$$\zeta_2(\tau, \hat{\mathbf{S}}) = \frac{2\sigma_0^2}{MN} tr\left[\mathbf{B}^T \otimes \mathbf{I}\right] \quad (12)$$

Averaging (9) over all possible realizations of $\hat{\mathbf{S}}$ such that $\mathbf{B}$ exists yields $\zeta(\tau)$. However, this approach is limited to small Q since the number of different $\hat{\mathbf{S}}$ grows exponentially with Q. For large Q a practical alternative is to limit the number of $\hat{\mathbf{S}}$ to a smaller number of random realizations. For example, in case of BPSK and $M = 2$, we obtained the convergence of the channel estimation MSE summing up over 20000 random realizations of $\hat{\mathbf{S}}$ that are generated with 10 different random number generator seeds. The corresponding $\zeta(0)$ is illustrated in 8. The simulations are limited to $Q = 8$. We denote Q that minimizes $\zeta(0)$ by Q_{opt}. The results of the minimization are shown in 9 for two f_d. We can see that Q_{opt} is larger for the smaller f_d as the channel variations during QT_s become less significant.

Drawing these conclusions, we have to keep in mind that (9) was derived taking the invertibility of $\hat{\mathbf{S}}\hat{\mathbf{S}}^H$ for granted, which is only true for large Q, and that the inversion is made at each discrete time instant, i.e. no use of last estimate while waiting untill $\mathbf{B}$ is computable. This explains why the BER results for determining Q_{opt} might differ from the theoretical MSE ones as will be shown in the next section.

IV. BER Results

In this section, we compare between the serial and parallel processing by means of the BER. The simulation results depend strongly on the temporal variance of the channel and the SNR. If not otherwise stated, we use 2 tx antennas, BPSK, and a ZF receiver. We consider two normalized Doppler frequencies, $f_d = 0.02$ in combination with a data transmission

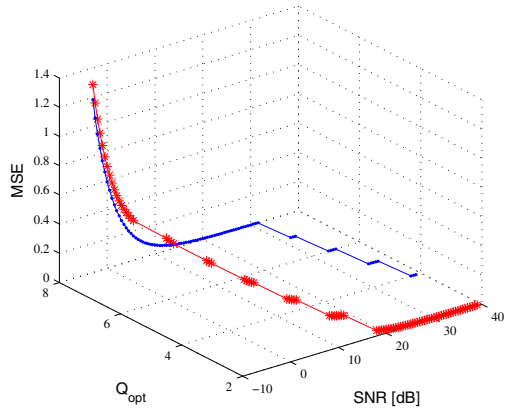


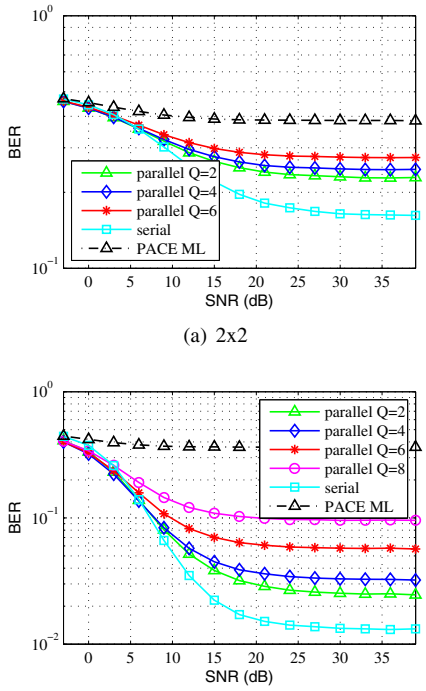
Fig. 9. Optimal block size Q_{opt} as function of the SNR for $f_d = 0.02$ (stars) and $f_d = 0.001$ (dots)

phase of $L_d = 30$ representing fast time-varying channels and $f_d = 0.001$ with $L_d = 300$ that models comparatively a rather slow time-varying channel.

In 10, the serial processing outperforms the parallel one with significant performance differences for $f_d = 0.02$ in comparison to $f_d = 0.001$ in 11. A slight performance improvement of the parallel processing with large Q can be noticed for low SNRs. Furthermore, the discrepancy between bit error floors corresponding to different Q values is especially notable for high f_d. For low f_d, no significant performance differences can be seen because of the small channel time variations during Q symbol periods. It is also worth to mention that $Q_{opt} = 2$ leads to the lowest error floor for $f_d = 0.02$ which is in agreement with the theoretical MSE minimization results in 9. However, as can be seen from the zoomed figures in 11, the error floors for $Q = 4$ and $Q = 2$ almost overlap for $f_d = 0.001$ which suggests that $Q_{opt} = 2$ is more adequate in this case due to the lower numerical complexity.

V. Conclusion

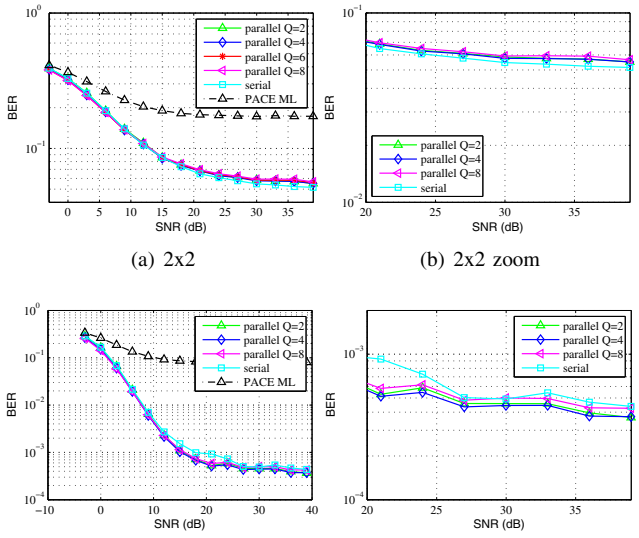
We investigate decision-directed MIMO channel tracking making use of the minimal number of pilots. Two different processing methods for the detected data vectors are compared, namely a parallel and a serial scheme. The parallel scheme requires a blockwise processing of the detected vectors. The corresponding theoretical channel estimation MSE is derived. The involved block size is optimized by minimization of the derived channel estimation error. The BER results for different MIMO systems show that serial processing is more appropriate if the channel time variations are fast. Furthermore the results of the optimal block size on the basis of theoretical estimation error are in agreement with the simulated BER ones.



(a) 2x2

(b) 2x4

Fig. 10. BER as a function of SNR at $f_d = 0.02$



(a) 2x2

(b) 2x2 zoom

(c) 2x4

(d) 2x4 zoom

Fig. 11. BER as a function of SNR at $f_d = 0.001$

References

[1] Q. Sun, D. Cox, H. Huang, and A. Lozano, "Estimation of continuous flat fading MIMO channels," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 549–553, October 2002.

[2] M. Kiessling and J. Speidel, "Statistical transmit processing for enhanced mimo channel estimation in presence of correlation," *IEEE International Conference on Global Communications (GLOBECOM)*, December 2003.

[3] E. Eitel and J. Speidel, "Enhanced decision-directed channel estimation of time-varying flat MIMO channels," *Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007*, pp. 1–5, 3-7 Sept. 2007.

[4] J. Gao and H. Liu, "Decision-directed estimation of MIMO time-varying rayleigh fading channels," *IEEE Transactions on Wireless Communications*, vol. 4, no. 4, pp. 1412–1417, July 2005.

[5] S. Haykin, *Adaptive Filter Theory*. Prentice-Hall, Inc., 1996, ISBN 0-13-322760-X.

[6] A. Grant, "Joint decoding and channel estimation for linear MIMO channels," *Wireless Commun. and Networking Conf.*, vol. 3, pp. 1009–1012, September 2000.

[7] S. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1993, ISBN 0-13-345711-7.