Efficient Training of Adaptive MIMO Channel Tracking Algorithms

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

***Abstract*—In this paper, improved training schemes for adaptive decision-directed tracking filters are proposed and applied to time-varying flat fading MIMO channels. In case of periodic training with a limited amount of pilots, it is shown that a trade-off between investing pilots for good initialization and exclusive training of the algorithm leads to a lower BER and a higher spectral efficiency than conventional training. In [1], we introduced a novel aperiodic training scheme for the Kalman filter. We extend this scheme to the RLS and LMS algorithms and develop appropriate metrics for filter divergence detection for each algorithm. The metric thresholds are derived analytically. We show that the proposed training reduces the BER significantly. The impact of feedback delay caused by pilot request and transmission is also investigated. Simulation results show that pilot on request training outperforms periodic training even for large mean pilot delays.**

I. Introduction

To achieve high bit rates with wireless MIMO systems, algorithms for precise channel estimation are of paramount importance. Often periodic pilot-assisted channel estimation (PACE) is employed under the assumption that the channel is almost constant during a data block length. However, with the next generation wireless standards putting greater demands on high mobility support, the radio link varies more rapidly. For such fast channel variations, PACE is not a resource-efficient method for channel tracking. In order to preserve the spectral efficiency decision-directed (DD) channel tracking can be applied. By exploiting detected symbols that reflect the current channel state information (CSI), adaptive filtering techniques such as Kalman filter (KF), least mean squares (LMS) or recursive least squares (RLS) filters can be used for channel tracking. In combination with a high-order autoregressive (AR) channel model, KF shows the best performance among them but at the expense of higher complexity. A drawback of DD tracking filters (TF) is their sensitivity to wrongly detected data. To cope with this problem, periodic pilot patterns can be inserted to stop a potential filter divergence. Other solutions exploit reliability information on the detected data and select only reliable data for channel tracking. These approaches require however iterative receiver structures which introduce significant delay and complexity [2]. Alternatively forward error correction (FEC) can be used in the data detection loop which also results in undesirable complexity and latency.

In this paper, we perform channel tracking by means of hard detected data. Since we neither collect reliability information nor make use of FEC, our tracking algorithms are simple and applicable independently of the decoder, yet showing good performance. Whilst the idea of using adaptive algorithms for channel tracking is already well-known in the literature, the contribution of this paper relies on optimizing training for adaptive channel tracking, an issue which has been barely considered so far. We show that in case of periodic training, investing a fraction of the available training data to initialize the TF appropriately speeds up the filter convergence. Furthermore, our analysis revisit a widespread assumption implying that more training of an adaptive algorithm leads to better performance. This assumption is based on steady state analysis where transient effects are not considered. Instead, we show that in conjunction with frame-wise data transmission, transient effects can be exploited in order to speed up the convergence rate, which is especially beneficial for fast fading channels. Exploiting the existence of local minima in the MSE curves, it turns out that in some cases, less training leads to better channel estimates. To the best of our knowledge, transient effects occuring until the tracking algorithm reaches its steady state have not received much attention in the literature so far. Training is only investigated by means of steady state analysis serving as an appropriate mathematical framework as in [3]. Although the authors in [3] mentioned the importance of the rate of convergence for packetized data-transmission, in contrast to continuous data-transmission, they did not pay profound attention to the issue. We show in this paper, that the training can benefit from local minima in the MSE curves which occur before the settling of steady state. As confirmed by simulations, this leads to a simulataneous BER reduction and bandwidth saving.

Furthermore, an aperiodic training scheme, originally introduced for KF in [1], is extended to the RLS and LMS filters. In this innovative scheme, training is only applied on request. A request for pilots is initiated if the filter diverges as a consequence of several successive detection errors. Appropriate metrics for error propagation detection are designed for each TF. Under the assumption of delay-free pilot request and transmission, we show that pilot on request training (PRQT) leads to significant BER performance gains compared to conventional periodic training. Interestingly, a simple tracking algorithm such as LMS, which is typically not appropriate for tracking fast fading channels, achieves with PRQT a similar performance as the computationally heavy-loaded KF. However, the effect of delays due to request and transmission of training data can not be neglected in realistic wireless networks. We address this problem by modeling the time of arrival of the receive pilot information as a random

variable with different probability density functions (pdf). We show that PRQT outperforms periodic training even for large mean pilot feedback delays. However, the delay tolerance depends strongly on the tracking algorithm.

II. System Model

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

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

where $\mathbf{s}(n)$ denotes the $M \times 1$ sent signal vector, $\mathbf{H}(n)$ the $N \times M$ MIMO flat fading channel matrix and $\mathbf{w}(n)$ the $N \times 1$ additive white Gaussian noise (AWGN) vector whose complex elements are i.i.d and $CN(0, 2\sigma_0^2)$ distributed. Without loss of generality, we assume a spatially uncorrelated MIMO Rayleigh fading channel. An entry $h_{ij}(n)$ of $\mathbf{H}(n)$ is $CN(0,1)$ distributed and satisfies:

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

where f_d stands for the normalized Doppler frequency and J_0 is the Bessel function of first kind and order zero. In order to estimate the channel at the receiver, orthogonal pilot symbol vectors $\mathbf{s}_p$ are periodically sent during the training period that takes L_p symbol periods T_s. At the end of the training phase, a channel estimate $\hat{\mathbf{H}}_p$ is computed by means of the received pilots. The training phase is followed by a data transmission phase where L_d symbol vectors are sent. In the absence of tracking, the PACE estimate is used for coherent detection during the subsequent L_d symbol periods.

In case of channel tracking, the PACE estimate can be used as good initial value for the tracking algorithm at the start of every training interval. However, given a limited amount of training, how much training should be applied for PACE to serve as appropriate initialization and how much should be reserved to train the TF? In order to answer these questions, we first study the TF convergence behavior by means of a full training analysis. Then, the hybrid periodic training (HPT), first suggested in [4] for the RLS filter and in [1] for KF, is applied to LMS and further optimized for all considered filters under a new constraint. A basic prerequisite for the application of HPT is the recursive nature of the tracking algorithms, which will be introduced in the next section.

III. The Adaptive Tracking Algorithms

We briefly introduce the considered TFs. For detailed derivations and algorithm variants please refer e.g. to [5]. It is noteworthy that in the DD mode, the TF operate on the detected symbol vector defined as $\hat{\mathbf{s}}(n)$.

A. Kalman Filter

If the fading channel can be modeled as an autoregressive process of order p (AR(p)), the KF is the optimal MMSE estimator. Since the first few channel correlation terms in (2) are basically important for symbolwise tracking, AR(2) modeling is adopted as in [6]. The Kalman algorithm relies on a state-space formulation composed of the observation equation (3) and the process equation (4).

TABLE I
Kalman filter algorithm

Variable	Equation
Predicted channel state $\hat{\mathbf{z}}(n\|n-1)$	$\mathbf{F}\hat{\mathbf{z}}(n-1\|n-1)$
Predicted MSE $\mathbf{P}(n\|n-1)$	$\mathbf{FP}(n-1\|n-1)\mathbf{F}^H + \mathbf{BB}^H$
Innovation $\mathbf{e}(n)$	$\mathbf{y}(n) - \mathbf{X}(n)\hat{\mathbf{z}}(n\|n-1)$
Autocorrelation of innovation $\mathbf{R}_{ee}(n)$	$\mathbf{X}(n)\mathbf{P}(n\|n-1)\mathbf{X}^H(n) + \mathbf{R}_{ww}$
Kalman gain $\mathbf{K}(n)$	$\mathbf{P}(n\|n-1)\mathbf{X}^H(n)\mathbf{R}_{ee}(n)^{-1}$
Corrected channel state $\hat{\mathbf{z}}(n\|n)$	$\hat{\mathbf{z}}(n\|n-1) + \mathbf{K}(n)\mathbf{e}(n)$
Corrected MSE $\mathbf{P}(n\|n)$	$(\mathbf{I} - \mathbf{K}(n)\mathbf{X}(n))\mathbf{P}(n\|n-1)$

TABLE II
RLS algorithm

Variable	Equation
Gain vector $\mathbf{k}(n)$	$\frac{\mathbf{P}(n-1)\hat{\mathbf{s}}(n)}{\lambda + (\hat{\mathbf{s}}(n))^H\mathbf{P}(n-1)\hat{\mathbf{s}}(n)}$
Innovation $\mathbf{e}(n)$	$\mathbf{y}(n) - \hat{\mathbf{H}}(n-1)\hat{\mathbf{s}}(n)$
Autocorrelation matrix inverse $\mathbf{P}(n)$	$\lambda^{-1}\mathbf{P}(n-1) - \lambda^{-1}\mathbf{k}(n)\hat{\mathbf{s}}^H(n)\mathbf{P}(n-1)$
Estimated channel matrix $\hat{\mathbf{H}}(n)$	$\hat{\mathbf{H}}(n-1) + \mathbf{e}(n)\mathbf{k}^H(n)$

$$\mathbf{y}(n) = \mathbf{X}(n)\cdot\mathbf{z}(n) + \mathbf{w}(n) \tag{3}$$

$$\mathbf{z}(n) = \mathbf{F}\mathbf{z}(n-1) + \mathbf{B}\mathbf{u}(n) \tag{4}$$

where $\mathbf{z}(n) = [\mathbf{h}^T(n) \quad \mathbf{h}^T(n-1) \quad \cdots \quad \mathbf{h}^T(n-p+1)]^T$ with $\mathbf{h}(n) = vec(\mathbf{H}(n))$ and $\mathbf{F}$ is the state transition matrix. $\mathbf{X}(n)$ contains the detected symbol vector $\hat{\mathbf{s}}(n)$ according to $\mathbf{X}(n) = [\hat{\mathbf{s}}^T(n) \otimes \mathbf{I}_N \quad \mathbf{O}_{N\times NM(p-1)}]$. $\mathbf{u}$ is the driving noise with $E[\mathbf{u}(n)\mathbf{u}(n)^H] = \mathbf{I}_{MN}$ [1]. The key equations of the KF are listed in Table I.

B. RLS Filter

The RLS algorithm aims at minimizing a sum of mean squared errors scaled by the forgetting factor λ. Many enhancements of the algorithm exist which basically adjust its effective memory (limit the number of observations by a sliding window or optimize the forgetting factor). We list the equations of the basic RLS filter in Table II. As described in [7], a correspondence exists between RLS and KF, if we assume that $\mathbf{F} = \lambda^{-\frac{1}{2}}\mathbf{I}$, $\mathbf{B} = \mathbf{O}$, and $\mathbf{R}_{ww} = \mathbf{I}$. Plugging these assumptions into the KF equations in Table I, we obtain an equivalent formulation of the RLS algorithm, where $\mathbf{z}(n) = vec(\mathbf{H}(n))$ instead of the matrix $\mathbf{H}(n)$ in Table II is estimated. Relying on these assumptions, we define $r_{ee}(n) = E\left[\mathbf{e}(n)^H\mathbf{e}(n)\right]$ as

$$r_{ee}(n) = \lambda + \hat{\mathbf{s}}(n)^H\mathbf{P}(n-1)\hat{\mathbf{s}}(n) \tag{5}$$

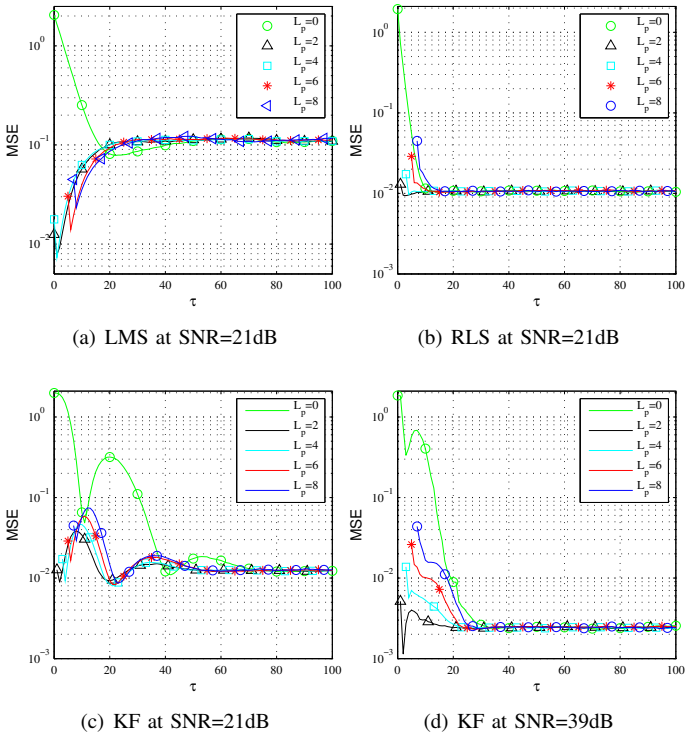
which will be important for the upcoming parts. If we further define $\mathbf{R}_{ee}(n) = E\left[\mathbf{e}(n)\mathbf{e}(n)^H\right]$ as already done for KF in Table I, mathematical manipulations lead to

$$\mathbf{R}_{ee}(n) = r_{ee}(n)\mathbf{I}_N \tag{6}$$

[1] $\mathbf{F}$ and $\mathbf{B}$ are assumed to be known at the receiver. Please refer to [6] for explicit definition.

TABLE III
LMS ALGORITHM

Variable	Equation
Innovation $\mathbf{e}(n)$	$\mathbf{y}(n) - \hat{\mathbf{H}}(n-1)\hat{\mathbf{s}}(n)$
Estimated channel matrix $\hat{\mathbf{H}}(n)$	$\hat{\mathbf{H}}(n-1) + \mu \mathbf{e}(n)\hat{\mathbf{s}}(n)^H$



(a) LMS at SNR=21dB (b) RLS at SNR=21dB

(c) KF at SNR=21dB (d) KF at SNR=39dB

Fig. 1. MSE against the time τ with full training for different L_p at $f_d = 0.01$

C. LMS Filter

The LMS algorithm finds a widespread application in adaptive filtering because of its simplicity and model-independency. Its speed of convergence is tightly related to the proper choice of the involved step size μ. This fact led to many variants of the algorithm which are not considered here. The basic algorithm consists of the two equations in Table III.

IV. PERIODIC TRAINING

To lauch the previously presented recursive algorithms, initial values for some variables must be chosen. It is common in the literature that the starting conditions are set to the mean value of the corresponding variables if known [5]. The filters in [2] and [8] are exclusively trained and marginal attention is paid to the initialization. In [4], we have shown the importance of an appropriate initialization for the RLS algorithm. In order to study the convergence behavior of the TFs, we simulate the full training case, which means that pilots are sent not only during the training but also throughout the data transmission phase. The corresponding MSE results against the time τ for different initialization lengths L_p are plotted in Fig. 1. The full training analysis reveals that independently of the TF, the convergence speed is drastically increased if we initialize the algorithm with the PACE estimate. Therefore, we propose to

TABLE IV
OPTIMAL (L_p, L_t) FOR 2×2 AND 2×4 MIMO SYSTEMS SUBJECT TO (8)

SNR [dB]	6	9	12	15	18	21	⋯	39
2×2	(4,0)	(2,0)	(2,0)	(2,8)	(2,8)	(2,8)	⋯	(2,8)
2×4	(4,0)	(4,0)	(2,2)	(2,2)	(2,2)	(2,8)	⋯	(2,8)

devide the periodically sent pilots into two sequences. The first sequence of length L_p provides the tracking algorithm with a PACE initial estimate. The second one trains the tracking algorithm and takes L_t samples. The optimal training length which minimizes the PACE MSE was derived in [1] as

$$L_{p,opt} = \begin{cases} \left\lfloor \sqrt{\frac{1}{2} + \frac{3\sigma_0^2}{\pi^2 f_d^2 M}} \right\rfloor & \text{if } \sqrt{\frac{1}{2} + \frac{3\sigma_0^2}{\pi^2 f_d^2 M}} > M \\ M & \text{otherwise} \end{cases} \quad (7)$$

where $\lfloor \cdot \rfloor$ refers to the floor operation. (7) shows that $L_{p,opt}$ increases with increasing σ_0^2 and decreases with increasing f_d or M. Keeping in mind that $L_p \geq M$ must be satisfied in case of maximum likelihood PACE estimation, (7) suggests that for high SNR and/or fast fading rate f_d, we should intuitively invest only M pilots for the PACE initialization and spend the remaining pilots at hand for training the algorithm. This was confirmed by simlulation results for RLS in [4] and KF in [1]. Therein, the parameter pair (L_p, L_t) was optimized to minimize the BER under the assumption of a constant spectral efficiency, i.e a fixed training amount $L_{tot} = L_p + L_t$,

The MSE results in Fig. 1 reveal however that if for example 30 pilots are available, then it is more beneficial to use only 20 of them since the MSE at $\tau = 20$ is smaller than the MSE at $\tau = 30$ (this holds especially for LMS in (a) and KF in (c)). In light of these remarks, we optimize (L_p, L_t) in this paper under the modified relaxed constraint

$$L_p + L_t \leq L_{tot} \quad (8)$$

In so doing, the existence of local minima in the full training MSE curves is exploited in order to not only reduce the channel estimation MSE but also to save bandwidth. Due to the more oscillative transient behavior of KF in comparison to RLS and LMS, we focus our findings on KF in the following. By intensive simulations, the optimal pairs (L_p, L_t) were determined for each SNR value in order to minimize the BER subject to (8) with $L_{tot} = 10$. The results are listed in Table IV for 2×2 and 2×4 MIMO systems and $L_d = 100$. Table IV shows that the relaxed constraint (8) becomes especially advantageous at low SNR. Whereas $(L_p, L_t) = (2, 8)$ is most adequate for high SNR, at SNR=12dB and with the 2×2 system for example, it is more beneficial to use $(L_p, L_t) = (2, 0)$. This means an 8% spectral efficiency improvement if we can transmit data instead of training during the freed 8 samples.

The corresponding BER results illustrated in Fig. 2 show that training the algorithm exclusively performs worse than allocating an amount of the training to supply the algorithm with a PACE initialization. As suggested by Table IV, we notice that HPT with $L_p = 2, L_t = 8$ performs best on a large SNR range. Though evident in terms of MSE, the advantage of the relaxed contraint (8) for low SNRs becomes less significant

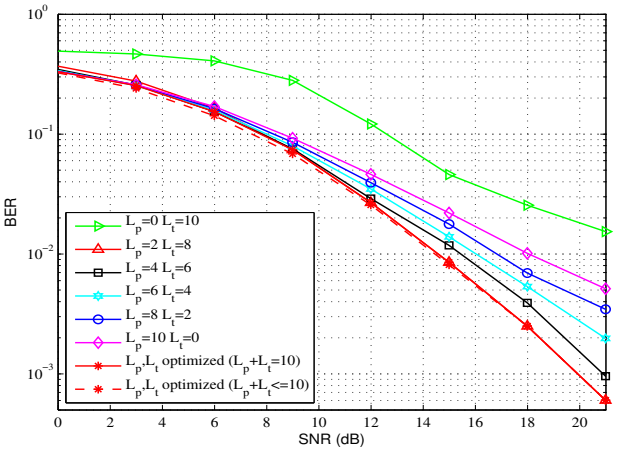


Fig. 2. BER against the SNR for KF trained by HPT and different (L_p, L_t)

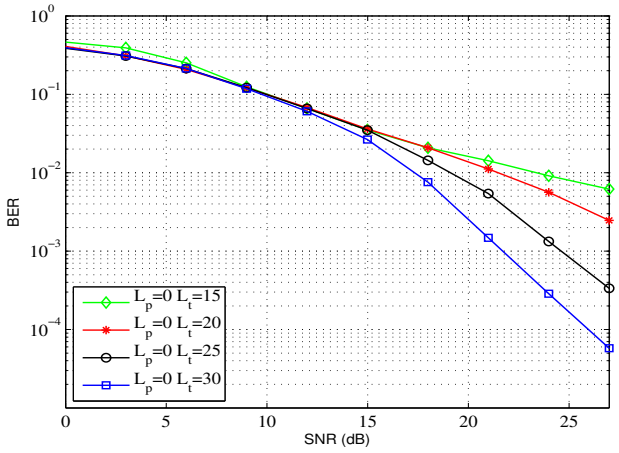
in terms of BER and only brings a 0.2dB gain at 10^{-2}. This is plausible since in the low SNR regime, the BER is governed by the high noise level which is more dominant than the channel estimation errors.

It is noteworthy that the results in Fig. 2 can be explained by the first local minimum in the MSE full training curves. If we allow for more training the second minimum comes into play. Fig. 3 shows that the intuitive expectation *"the more to train, the better"* only holds for conventional training with mean value initialization. However, using an appropriate initialization which takes $L_p = 2$ to build the PACE estimate, $L_t = 20$ performs better than $L_t = 25$ or even $L_t = 30$. Furthermore, Fig. 3 shows that at a BER of 10^{-4}, $(L_p, L_t) = (2, 20)$ necessitates 22dB, whereas $(L_p, L_t) = (0, 30)$ necessitates 26dB though taking 8 more samples for training. Therefore, we can conclude that HPT under the new relaxed constraint (8) results in simultaneously reducing the BER and increasing the bandwidth efficiency.

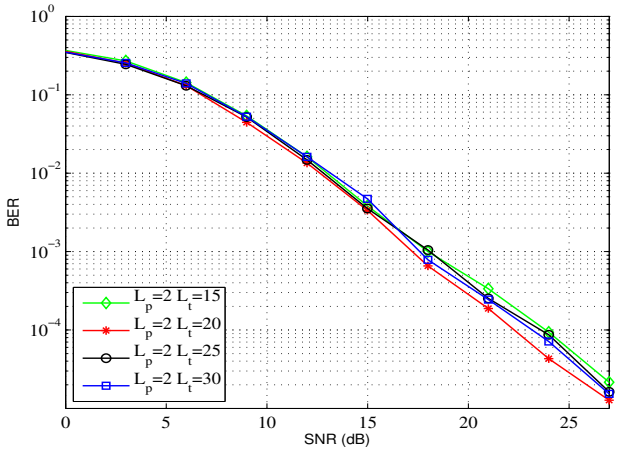
When applying HPT, we noticed that the MSE at the end of a frame can be smaller than the MSE at the beginning of a frame. This happens for example at high SNR when the detected data is mostly correct. In this case, reinitialization is disadvantageous. This gave birth to the idea of applying training only when needed, which is described next.

V. Aperiodic Training: Pilots on Request

We briefly explain the principle of PRQT as first suggested in [1] and extend it to RLS and LMS. In the DD mode, TFs work robustly as long as almost all detected symbols are correct. In case of misdetections, we have a model mismatch and the channel estimation quality deteriorates resulting in more misdetections in the following steps and to error propagation. Accurate detection of error propagation is a key issue for PRQT in order to preserve the spectral efficiency. Therefore, we design and discuss appropriate metrics for the detection of error propagation for each TF. If the metric exceeds a certain threshold M_d an error propagation is signaled and pilots are requested to stop it. The threshold optimization minimizes the BER satisfying a certain spectral efficiency constraint.



(a) HPT with $L_p = 0$



(b) HPT with $L_p = 2$

Fig. 3. BER against the SNR for KF at $f_d = 0.01$ showing the impact of the second local minimum in the MSE full training curve

A. Metric Design for the Kalman and RLS Filter

Because of afore-mentioned correspondences, KF and RLS are considered together in this subsection. Closer analysis of different statistical quantities involved in the algorithms suggests that a filter divergence occurs in most of the cases just after a steep peak has appeared in their progress. This is for example the case for the innovation process norm $|\mathbf{e}(n)|^2$. Fig. 4 shows some variables involved in the KF tracking process e.g. for $f_d = 0.004$, $L_t = 2$ and $L_d = 200$. We can see that a large $|\mathbf{e}(n)|^2$ due to an instantaneous high noise value gives birth to a series of wrong detections resulting in a filter divergence. The error propagates until the beginning of the next frame where the estimate is set to the PACE value.

Armed with these observations, we develop a first metric m_1 to detect a filter divergence defined as

$$m_1 = |\mathbf{e}(n)|^2 = \mathbf{e}(n)^H \mathbf{e}(n) \tag{9}$$

Intuitively, we expect the threshold M_d on m_1 to depend on the SNR which is proved in the following for $N = 2$ receive antennas. Under the assumption of correctly detected data and negligible channel estimation errors, m_1 follows an Erlang pdf with the rate parameter $\frac{1}{2\sigma_0^2}$ and N degrees of freedom.

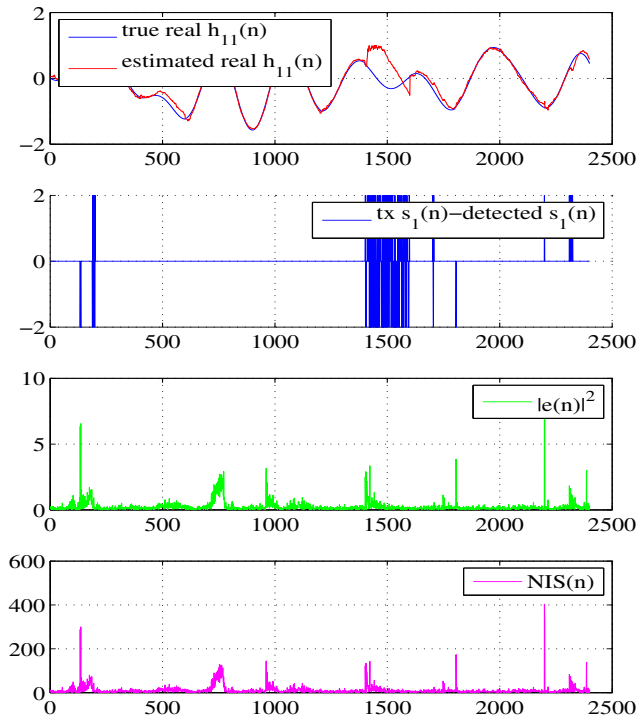


Fig. 4. Analysis of different variables involved in the KF as function of the discrete time n

Relying on the theory of validation gating, the threshold M_d on m_1 can be computed from

$$\frac{p}{100} = \frac{1}{(2\sigma_0)^{2N}(N-1)!}\int_0^{M_d} e^{\frac{-t}{2\sigma_0^2}} t^{N-1} dt \qquad (10)$$

(10) means that $m_1 < M_d$ holds for a probability p%. Further mathematical manipulations on (10) lead to the following equation:

$$\frac{M_d}{2\sigma_0^2} = ln\left(1 + \frac{M_d}{2\sigma_0^2}\right) - ln\left(1 - p\%\right) \qquad (11)$$

Solving (11) for a specific p shows that there exists a constant factor Q such that $M_d = Q \cdot 2\sigma_0^2$, which proves a linear dependency of M_d on the SNR[2]. In [1], we presented an empirical approach which describes M_d as a function of the SNR but this method requires the optimization of two parameters for each SNR, which might be numerically a tedious task.

For this reason, we suggest a second metric m_2 referred to as the normalized innovation squared (NIS) as it does not depend on the SNR. m_2 is also depicted in Fig. 4 and can be written as:

$$m_2 = \mathbf{e}(n)^H \mathbf{R}_{ee}^{-1}(n)\mathbf{e}(n) \qquad (12)$$

In case of the RLS filter, plugging (6) into (12) yields

$$m_2 = \frac{\mathbf{e}(n)^H \mathbf{e}(n)}{r_{ee}(n)} \qquad (13)$$

with $r_{ee}(n)$ from (5). Under the assumption of correctly detected data, m_2 follows a chi-square pdf. Thus,

[2]An explicit real closed-form solution can not be given for arbitrary p

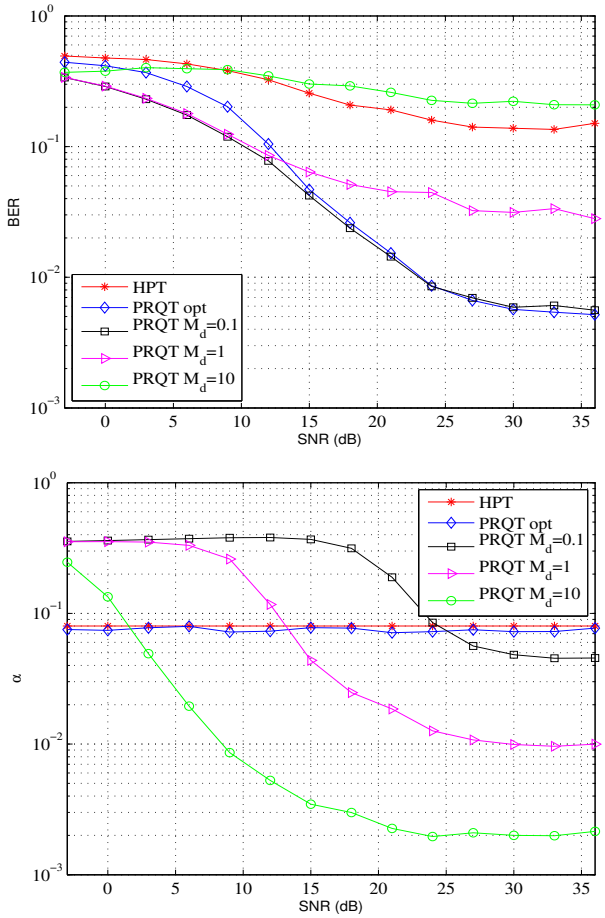


Fig. 5. BER against the SNR for RLS with PRQT and various M_d (top). Corresponding ratio α of pilots over data against the SNR (bottom)

$\mathbf{e}(n)^H \mathbf{R}_{ee}^{-1}\mathbf{e}(n) < M_d$ for a probability p% means that M_d satisfies

$$\frac{p}{100} = \frac{1}{\Gamma(N/2)}\int_0^{M_d/2} e^{-t} t^{N/2-1}\, dt \qquad (14)$$

where Γ is the Gamma function. For $N = 2$, $M_d = -2ln(1 - p\%)$ which is not SNR-dependent. The BER for RLS ($\lambda = 0.7$) with different M_d values and the ratio α of pilots over data are illustrated in Fig. 5. A trade-off between low BER and high spectral efficiency is plain to see. Despite our target of designing an SNR-independent metric, α does vary for small SNRs and becomes constant when the SNR increases. This can be explained by the fact that the chi-square distribution only holds for perfectly detected data, which is increasingly true when the SNR grows.

B. Metric Design for the LMS Filter

According to the LMS algorithm in Table III, two quantities are conceivable for building a metric: the innovation process, which leads to our afore-defined metric m_1 in (9), and the gradient $\mathbf{G}(n) = \mathbf{e}(n)\hat{\mathbf{s}}(n)^H$. It is well-known in the literature that the gradient provides a good measure for the filter convergence. Here, we exploit this measure to detect a filter divergence. We define the gradient-based metric m_3 as:

$$m_3 = \left|\mathbf{e}(n)\hat{\mathbf{s}}(n)^H\right|_F^2 \qquad (15)$$

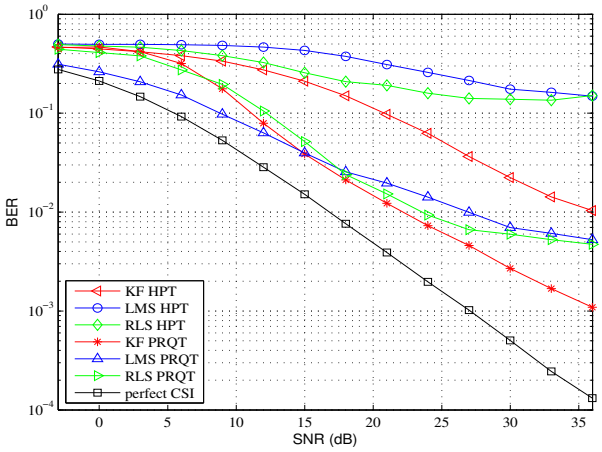


Fig. 6. BER as function of the SNR for HPT and delay-free PRQT with $\alpha = 8\%$ pilot requirement

For constant energy modulations such as M-PSK with symbol energy E_s, m_3 is equivalent to m_1 scaled by a constant factor since $m_3 = tr\left[\mathbf{e}(n)\hat{\mathbf{s}}(n)^H\hat{\mathbf{s}}(n)\mathbf{e}(n)^H\right] = ME_s\left|\mathbf{e}(n)\right|^2 = ME_sm_1$. Due to the lack of information on the innovation process or gradient statistics inherent to the LMS algorithm, the threshold on m_1 or m_3 can not rely on any expectation information as for KF and RLS. Analogously to the afore-presented validation gating, we rely on the metric pdfs, which we determine by simulations.

VI. BER Simulation Results for optimized PRQT with and without Pilot Feedback Delay

We present in Fig. 6 the BER results of PRQT for the three considered filters and delay-free pilot request and transmission. We use a 2×2 MIMO system with BPSK and zero-forcing receiver. For fair comparison with HPT fulfilling $\frac{L_p+L_t}{L_d} = 8\%$, the metric thresholds were optimized under the constraint that the pilot requirement α satisfies $\alpha \leq 8\%$. The significant performance improvement for all three filters is plain to see. We draw the reader's attention to the LMS filter which exhibits with HPT a poor channel tracking performance for $f_d = 0.01$. However, the BER is significantly decreased by adopting PRQT and even outperforms KF with HPT. We can therefore conclude that PRQT allows the LMS filter to have a better tracking performance than the more complicated KF algorithm. This means a significant complexity reduction when tracking fast fading channels ($O(MN)$ for LMS vs. $O((MNP)^3)$ in case of KF with AR(p)).

Whereas the afore-presented results as well as [1] rely on the unrealistic delay-free pilot feedback assumption, the effect of delayed pilot request and transmission is considered in this paper. Feedback delay in general is admittedly an issue that is dependent on upper layers. One of the most important aspects of next generation wireless communication standards is the stricter latency requirement. Putting that into perspective and due to the simple PRQT signal processing in comparison to common CSI feedback for adaptive MIMO transmit schemes, we expect that the delay should not be a critical parameter. Nevertheless, the coming results intend to give a rather qualitative investigation about the impact of pilot feedback delay on realistic PRQT operability.

According to the literature, many distributions for the arrival time of the received pilots are conceivable. We define N_d as the number of samples from the error propagation detection until the arrival of pilot symbols at the receiver. In our simulations, we tested different pdfs for N_d and found out that randomness and pdf nature of the delay are not relevant for the BER performance. This is plausible, since bit errors are likely to happen during the whole pilot feedback delay. While in abscence of observations, KF is able of propagating the estimate by predicting, the RLS and LMS algorithms are not. Therefore, in case of RLS and LMS the last estimate available before pilot request is used for detection during the delay. This estimate is presumably of very poor quality and results in a series of detection errors during the delay time. Thus, the mean delay controls the mean number of erroneous bits and consequently the average BER. The performance degradation by delay consideration can be seen in Fig. 7[3]. From these results, we conclude that PRQT outperforms HPT for large mean delays. LMS and RLS are more delay-tolerant than KF which can be explained by the strong model-dependency of KF in comparison to the other TFs. Applying PRQT to KF without prediction is very delay-sensitive and reaches already at $f_dN_d = 0.1$ a BER comparable to HPT.

VII. Conclusion

In this paper, we track fast-varying MIMO channels by applying adaptive filters such as Kalman, RLS and LMS algorithms. We show that in conjunction with frame-wise data transmission, special attention should be paid to the filter training. In case of periodic training, a suggested hybrid periodic training performs better than conventional periodic training in terms of BER and spectral efficiency. An aperiodic pilot on request training (PRQT) scheme is generalized to various filters. To this end, appropriate metrics for error propagation detection are designed and a method to analytically derive their thresholds is presented. Simulation results show that PRQT leads to significant performance improvement even for large mean pilot feedback delays.

References

[1] E. Eitel and J. Speidel, "Efficient training of Kalman algorithm for MIMO channel tracking," *Wireless Conference 2011 - Sustainable Wireless Technologies, 11th European Wireless*, Apr. 2011.

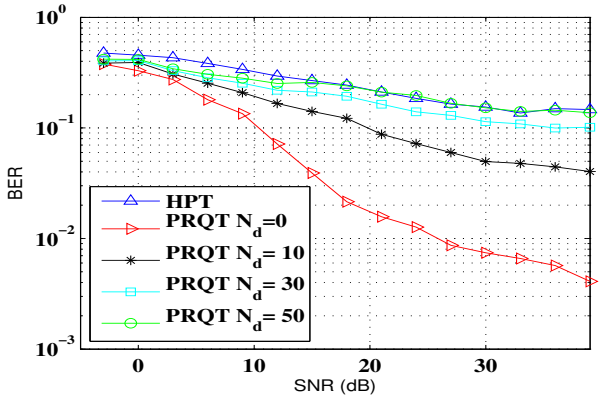
[2] I. Nevat and J. Yuan, "Joint channel tracking and decoding for BICM-OFDM systems using consistency tests and adaptive detection selection," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 8, pp. 4316 –4328, Oct. 2009.

[3] M. Dong, L. Tong, and B. Sadler, "Optimal insertion of pilot symbols for transmissions over time-varying flat fading channels," *Signal Processing, IEEE Transactions on DOI - 10.1109/TSP.2004.826182*, vol. 52, no. 5, pp. 1403–1418, 2004.

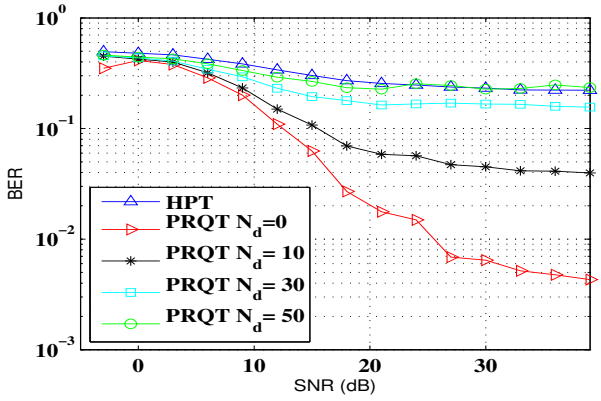
[4] E. Eitel, R. A. Salem, and J. Speidel, "Improved decision-directed recursive least squares MIMO channel tracking," *IEEE International Conference on Communications*, Jun. 2009.

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

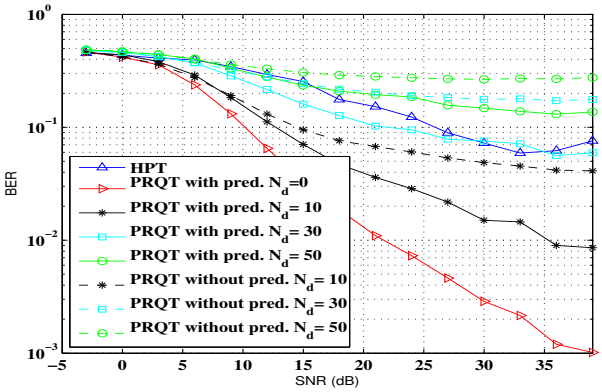
[3] A way to overcome this problem is to buffer the received data while awaiting the pilot information and perform the detection posteriorly when a better channel estimate is available, which is beyond the scope of this paper.



(a) LMS



(b) RLS



(c) KF with and without prediction

Fig. 7. BER as function of the SNR without ($N_d = 0$) and with various pilot feedback delays N_d for $f_d = 0.01$

[6] C. Komninakis, C. Fragouli, A. Sayed, and R. Wesel, "Multi-input multi-output fading channel tracking and equalization using Kalman estimation," *IEEE Transactions on Signal Processing*, vol. 50, no. 5, pp. 1065 –1076, May. 2002.

[7] A. Sayed and T. Kailath, "A state-space approach to adaptive RLS filtering," *IEEE Signal Processing Magazine*, vol. 11, no. 3, pp. 18 –60, Jul. 1994.

[8] E. Karami and M. Shiva, "Decision-directed recursive least squares MIMO channel tracking," *EURASIP Journal on Wireless Communications and Networking*, Dec. 2005.