This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2009 proceedings

Improved Decision-Directed Recursive Least Squares MIMO Channel Tracking

Emna Eitel, Rana H. Ahmed Salem, and Joachim Speidel
Institute of Telecommunications, University of Stuttgart, Germany

***Abstract*—This paper presents a comparative study of different Recursive Least Squares algorithms to track a Rayleigh flat fading MIMO channel. We investigate the effect of the initialization and training of these algorithms on their performance. We propose a new training scheme which can deliver a lower mean squared estimation error without loss of bandwidth efficiency.**

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 even be enhanced if the channel state information (CSI) is also available at the transmitter. Algorithms to precisely estimate the CSI are therefore of paramount importance. Often periodical pilot-assisted channel estimation (PACE) is employed. However, in fast time-varying channels, PACE does not only decrease the bandwidth efficiency but is also incapable of detecting fast variations of the channel. Therefore, additional tracking techniques have to be applied. A method that does not require pilots is decision-directed channel estimation (DDCE). It uses previously detected symbols and can therefore feed the channel estimation module with data that permanently takes account of the almost actual channel state. DDCE was recently investigated in [1] where the authors use infinite coding with infinite delay that allows them to make the assumption that the symbols are always correctly detected. In contrast, we consider realistic conditions and perform a symbol-by-symbol tracking that does not introduce any delay.

Adaptive filtering techniques such as Kalman, least mean squares (LMS) or recursive least squares (RLS) filtering can also be used for channel tracking and are decision-directed as well since they make use of previously detected data. In combination with a high-order autoregressive channel modelling, the Kalman filter shows the best performance among them. However, its main drawback is its high complexity. In contrast, the RLS algorithm has a relatively low complexity and the advantage of being independent of the channel model, i.e. it does not require any knowledge about the SNR or the Doppler frequency. Furthermore, the RLS algorithm does not make any assumptions about the statistics of input data unlike e.g. Wiener filters. That is why RLS is known in the literature to have a very good convergence rate and steady state performance in stationary applications [2]. Its main drawback is the significant performance degradation if non-stationary statistics are present [2].

We therefore enhance the tracking capability of the RLS algorithm for fast time-varying MIMO channels through optimizing the effective memory of the algorithm. This is done by either applying a data window which limits the number of considered data samples [3], [4], [5] or by optimizing the forgetting factor with each iteration of the algorithm. We also investigate the use of an optimal value for the forgetting factor from [6] which has the advantage of simplicity. In addition, we consider the effect of initialization of the RLS variables on its performance. In [7] a non-sequential form of the algorithm was used, which does not allow the channel matrix at the start of the training period to be initialized except by the zero matrix. Here we extend the non-sequential standard algorithm to a sequential form allowing the channel matrix estimate to be initialized by a better value than the zero matrix, namely the PACE matrix. This brings us to the important issue of how to deal with the pilots when applying RLS tracking. In the literature two solutions are proposed. First, all pilots are used for a good PACE estimate that represents an adequate starting point for the tracking algorithm. Alternatively, all pilots can be used to train the algorithm as done in [7]. We propose a new training scheme where one part of the pilots is attributed to the PACE block and the rest for training the algorithm. We show that this hybrid training scheme can be optimized for a certain bandwidth (BW) efficiency to attain faster convergence of the RLS algorithm. Simulation results for different Doppler frequencies show the effectiveness of the proposed scheme.

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:

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

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 $CN(0, 2\sigma_0^2)$. One element $h_{ij}(n)$ of $\mathbf{H}(n)$ represents the channel coefficient between the *j*th transmit and *i*th receive antenna and is assumed to be $CN(0, 1)$. $h_{ij}(n)$ realizes a discrete-time Rayleigh flat fading process in the equivalent base-band with the temporal autocorrelation function:

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

where T_s stands for the symbol period, f_d is the Doppler frequency and J_0 is the Bessel function of first kind and order zero. We define the normalized Doppler frequency $f_{dnorm} = f_d \cdot T_s$. In order to estimate the channel at the

978-1-4244-3435-0/09/$25.00 ©2009 IEEE

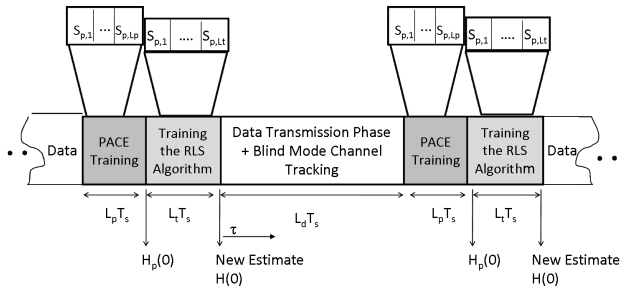

Fig. 1. Alternating training and data phases in the hybrid training scheme

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 $\hat{\mathbf{H}}$ is computed by means of the received pilots according to the maximum likelihood or the minimum mean squared error principle. 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 the coherent detection of data symbols during the subsequent L_d symbol periods. In case tracking is employed, the PACE block can be used as good initial estimate for the tracking algorithm at the start of every training interval, provided a sequential tracking algorithm is applied as will be derived in Section III-A.

Doing so, the question that arises is how good the initialization has to be. The more pilots the better the PACE estimate and we expect therefore that RLS converges faster. On the other hand and to the best of our knowledge, RLS training in the literature is only applied for the algorithm itself. As proposed in [7], [8], using a non-sequential form of the algorithm, the matrix estimate is initialized with zeros. The periodically sent pilots are used as input to the algorithm to train its statistical variables. We combine both training methods to a so-called hybrid scheme as can be seen in Fig. 1, where the periodically sent pilots are devided into two sequences. One sequence of length L_p that is attributed to the PACE block provides the tracking algorithm with a good initial estimate. The second sequence trains the algorithm and takes L_tT_s. For a fair comparison of the new training scheme with the previously established ones, the optimal L_p and L_t are chosen such that $(L_p+L_t)/L_d$ is kept constant. We introduce the discrete time index τ to denote the time elapsed between the last training phase and the end of the data transmission phase, i.e. $0 \leq \tau \leq L_d$.

III. RLS Algorithm for MIMO Channel Tracking

A. Sequential and Non-Sequential MIMO RLS Algorithm

In this section, a sequential form of the RLS algorithm is presented. Sequential means that $\hat{\mathbf{H}}(n)$ can be directly computed from $\hat{\mathbf{H}}(n-1)$. A sequential algorithm is derived in [2] to estimate an $M \times 1$ channel vector. We extend this method to the estimation of the $N \times M$ channel matrix. In [7] the non-sequential algorithm is described starting with the basic cost function which is the weighted sum of the error squares as follows:

$$J(n)_{LSE} = \sum_{i=0}^{n} \lambda^{n-i} \left\| \mathbf{r}(i) - \hat{\mathbf{H}}(i)\hat{\mathbf{s}}(i) \right\|^2 \tag{3}$$

where $\hat{\mathbf{s}}(i)$ denotes the detected symbol vector and λ the forgetting factor. Starting from the non-sequential algorithm as described in Table I, a sequential form is derived in the appendix and summarized in Table II.

TABLE I
NON-SEQUENTIAL RLS ALGORITHM

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

TABLE II
SEQUENTIAL RLS ALGORITHM

Variable	Equation
Gain vector $\mathbf{k}(n)$	$\frac{\mathbf{P}(n-1)\hat{\mathbf{s}}(n)}{\lambda+\hat{\mathbf{s}}^H(n)\mathbf{P}(n-1)\hat{\mathbf{s}}(n)}$
A priori error vector $\mathbf{e}(n)$	$\mathbf{r}(n)-\hat{\mathbf{H}}(n-1)\hat{\mathbf{s}}(n)$
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 $\hat{\mathbf{H}}(n)$	$\hat{\mathbf{H}}(n-1)+\mathbf{e}(n)\mathbf{k}^H(n)$

B. Sliding Exponential Windowed Recursive Least Squares Algorithm (SEW-RLS)

The first approach to enhance the tracking capability of the RLS algorithm is to limit the number of samples in its effective memory. Therefore a modified cost function is introduced:

$$J(n)_{LSE} = \sum_{i=n-L+1}^{n} \lambda^{n-i} \left\| \mathbf{r}(i) - \hat{\mathbf{H}}(i)\hat{\mathbf{s}}(i) \right\|^2 \tag{4}$$

where L denotes the window size. SEW-RLS is composed of two main processes: updating and downdating. In the update equations the effect of the new data samples $\mathbf{r}(n)$ and $\hat{\mathbf{s}}(n)$ is incorporated into the memory of the algorithm, whereas in the downdate equations the effect of the previous $(L+1)$ old data sample is eliminated such that the window size is again L. This implies that two buffers, each storing L data samples, are needed for the implementation of SEW-RLS, which is added to the complexity and storage requirements of the algorithm. The complete algorithm can be summarized by the update equations in (5)-(8), and the downdate equations in (9)-(12):

$$\mathbf{e}(n) = \mathbf{r}(n) - \hat{\mathbf{H}}(n-1)\hat{\mathbf{s}}(n) \tag{5}$$

$$\mathbf{k}(n) = \frac{\mathbf{P}(n-1)\hat{\mathbf{s}}(n)}{\lambda + \mathbf{s}^H(n)\mathbf{P}(n-1)\hat{\mathbf{s}}(n)} \tag{6}$$

$$\tilde{\mathbf{P}}(n) = \lambda^{-1}[\mathbf{P}(n-1) - \mathbf{k}(n)\hat{\mathbf{s}}^H(n)\mathbf{P}(n-1)] \tag{7}$$

$$\tilde{\mathbf{H}}(n) = \hat{\mathbf{H}}(n-1) + \mathbf{e}(n)\mathbf{k}^H(n). \tag{8}$$

$$\tilde{\mathbf{e}}(n-L|n) = \mathbf{r}(n-L) - \tilde{\mathbf{H}}(n)\hat{\mathbf{s}}(n-L) \tag{9}$$

TABLE III
RLS ALGORITHM WITH VARIABLE FORGETTING FACTOR

Variable	Equation
Forgetting factor $\lambda(n)$	$\left[\lambda(n-1) - \alpha Re\left\{\mathbf{e}^H(n)\right.\right.$ $\left.\left.\mathbf{D}(n-1)\hat{\mathbf{s}}(n)\right\}\right]_{\lambda_{min}}^{\lambda_{max}}$
Gradient of the autocorrelation matrix inverse w.r.t. λ: $\mathbf{M}(n)$	$\lambda^{-1}\left[\mathbf{I}_{M\times M} - \mathbf{k}(n)\hat{\mathbf{s}}^H(n)\right]\cdot$ $\mathbf{M}(n-1)\left[\mathbf{I}_{M\times M} - \hat{\mathbf{s}}(n)\mathbf{k}^H(n)\right]$ $-\lambda^{-1}\mathbf{P}(n) + \lambda^{-1}\mathbf{k}(n)\mathbf{k}^H(n)$
Channel matrix gradient w.r.t. λ: $\mathbf{D}(n)$	$\mathbf{D}(n-1)\left[\mathbf{I}_{M\times M} -\right.$ $\left.\hat{\mathbf{s}}(n)\mathbf{k}^H(n)\right] + \mathbf{e}(n)\hat{\mathbf{s}}^H(n)\mathbf{M}(n)$

$$\tilde{\mathbf{k}}(n) = \frac{\tilde{\mathbf{P}}(n)\hat{\mathbf{s}}(n-L)}{-\lambda^{-L} + \hat{\mathbf{s}}^H(n-L)\tilde{\mathbf{P}}(n)\hat{\mathbf{s}}(n-L)} \tag{10}$$

$$\mathbf{P}(n) = \tilde{\mathbf{P}}(n) - \tilde{\mathbf{k}}(n)\hat{\mathbf{s}}^H(n-L)\tilde{\mathbf{P}}(n) \tag{11}$$

$$\hat{\mathbf{H}}(n) = \tilde{\mathbf{H}}(n) + \tilde{\mathbf{e}}(n-L|n)\tilde{\mathbf{k}}^H(n) \tag{12}$$

where $\tilde{\mathbf{P}}(n)$ and $\tilde{\mathbf{H}}(n)$ are intermediate variables which help for the transition in the memory of the algorithm. $\tilde{\mathbf{e}}(n-L|n)$ is the a posteriori error vector at time index $(n-L)$ given n data samples.

C. Optimal Fixed and Variable Forgetting Factor

In this section, we present two methods for optimizing the forgetting factor for MIMO channel tracking:

1) Optimal Fixed Forgetting Factor RLS Algorithm (OFF-RLS): OFF-RLS was derived in [6] in order to minimize the mean square error and exhibits the same complexity as the standard RLS. It needs to be fed directly with the actual values of the Doppler shift and SNR as can be seen in (13):

$$\lambda_{opt} = \begin{cases} \lambda_{min} & \text{if } \lambda_0 < \lambda_{min} \\ \lambda_{max} & \text{if } \lambda_0 > \lambda_{max} \\ \lambda_0 & \text{elsewhere} \end{cases} \tag{13}$$

with $\lambda_0 = 1 - \left[\frac{(2\pi f_d T_s)^2 \cdot E_s}{\sigma_0^2}\right]^{1/3}$.

λ_{min} and λ_{max} are upper and lower limits on λ_{opt} and E_s is the symbol energy. Thus, the former advantage of being model-independent is lost. In addition, OFF-RLS is very sensitive to the value of λ_{min} due to the noise variance in the denominator of λ_0.

2) Variable Forgetting Factor RLS Algorithm (VFF-RLS): Another approach is the use of a VFF which is optimized at each iteration step of the algorithm. This is done by minimizing the instantaneous squared error $\mathbf{e}(n) = \mathbf{r}(n) - \hat{\mathbf{H}}(n-1)\mathbf{s}(n)$ as in [2] and [9]. The complete algorithm is summarized in Table III, where α is the learning parameter and can be optimized empirically.

IV. SIMULATION RESULTS

A. Performance of the Different RLS Algorithms

We present the bit error ratio (BER) and mean square error (MSE) results for the different RLS variants. A 2×4 MIMO system with BPSK modulation and zero-forcing receiver is considered. Two channels with $f_{dnorm} = 0.01$ and $f_{dnorm} = 0.001$ are used. For a fair comparison, the complexity of the different algorithms should also be taken into account. Whereas most of the variants are $O(M^2)$, VFF-RLS exhibits the highest compexity with $O(M^3)$. As shown in Fig. 2 and Fig. 3, the best performance at the high Doppler frequency is provided by VFF-RLS and OFF-RLS. SEW-RLS does enhance the performance compared to the fixed forgetting factor case, but still the window size and the forgetting factor need to be optimized empirically. In Fig. 3 we see how the MSE of the SEW-RLS deviates from the fixed forgetting factor curve exactly at $\tau = L$, i.e. at the point where the downdating starts.

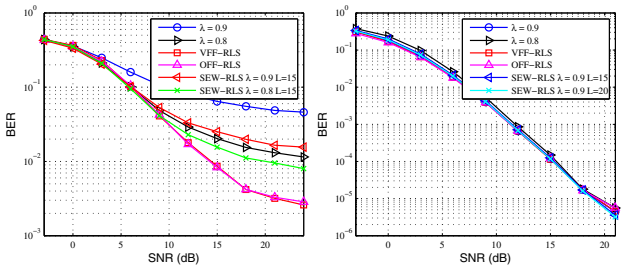

Fig. 2. BER of various RLS algorithms as function of the SNR for $f_{dnorm} = 0.01$ (left) and $f_{dnorm} = 0.001$ (right)

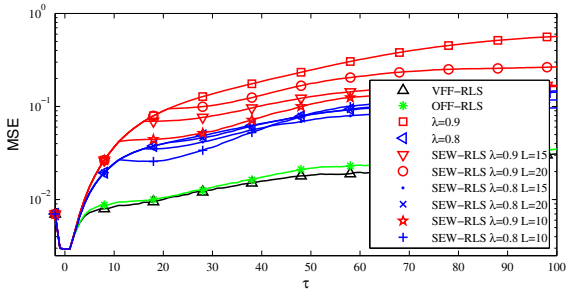

Fig. 3. MSE of various RLS algorithms as a function of the discrete time index τ for SNR=30dB and $f_{dnorm} = 0.01$

B. Optimizing the Training Scheme

In order to optimize the training scheme in Fig. 1, we simulate the full training case, which means that pilots are sent not only during the training but also throughout the data phase. The corresponding MSE is presented in Fig. 4 and Fig. 5. We use OFF-RLS but similar results are obtained with the other RLS algorithms. We compare the case when the channel matrix estimate is initialized by zeros to the case where the initialization is done by means of the PACE estimate. We can see that both approaches converge to the same steady state MSE for $\tau > 10$ at SNR=42dB and $\tau > 20$ at SNR=3dB. We can conclude that for real applications without full training, 20 pilots are required to start the tracking with minimal MSE. We also find out that for SNRs smaller than 6dB, zero initialization outperforms the PACE one. As the SNR increases the PACE estimate quality improves significantly. We can also notice the existence of an MSE minimum in the PACE-initialized curve at high SNRs and high Doppler shift, implying that there exists an optimal choice for L_p and L_t according to

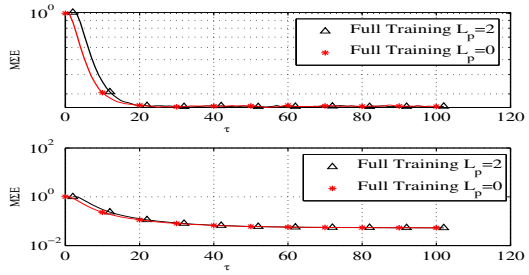

Fig. 4. MSE as function of the discrete time τ with full training at $f_{dnorm} = 0.01$ (top) and $f_{dnorm} = 0.001$ (bottom) for SNR = 3dB

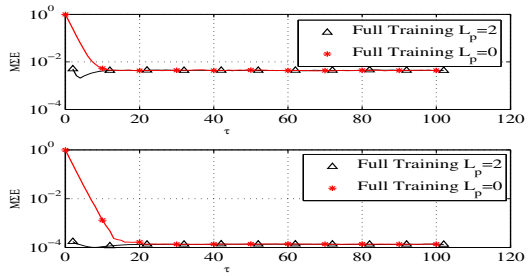

Fig. 5. MSE as function of the discrete time τ with full training at $f_{dnorm} = 0.01$ (top) and $f_{dnorm} = 0.001$ (bottom) for SNR = 42 dB

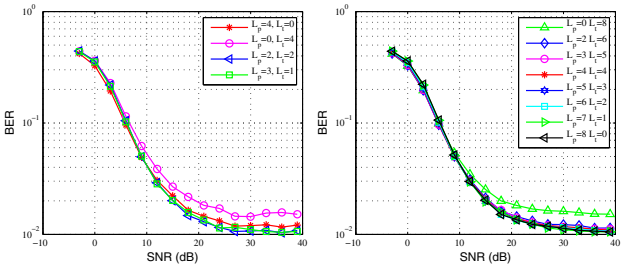

Fig. 6. BER as function of the SNR for $f_{dnorm} = 0.01$ and $L_d = 100$ for fixed $L_p + L_t = 4$ (left) and $L_p + L_t = 8$ (right)

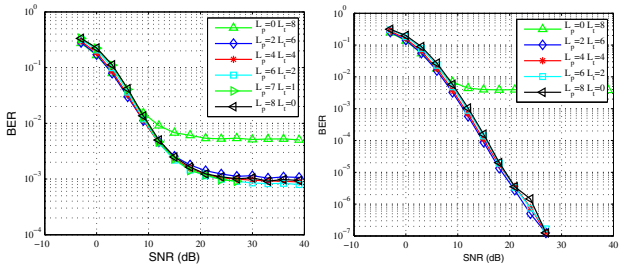

Fig. 7. BER as function of the SNR for $L_d = 20$ and $L_p + L_t = 8$ at $f_{dnorm} = 0.01$ (left) and $f_{dnorm} = 0.001$ (right)

the SNR and Doppler shift. From the full training results we can conclude that initializing with PACE results in a faster convergence of the tracking algorithm beginning from a certain SNR value. Thus for a training period smaller than the time needed for settling the steady state, PACE initialization can be expected to be more beneficial. These observations are confirmed by simulations with real data transmission phases where the training takes $L_p + L_t = 4$ and $L_p + L_t = 8$. Fig. 6 shows that for $f_{dnorm} = 0.01$ in the low SNR range only training the algorithm leads to lower BER. In the high SNR range, the hybrid training scheme performs significantly better than pure training of the algorithm as in [7] and slightly better than the case where all the pilots are exclusively attributed to PACE. For $f_{dnorm} = 0.001$ the BER of the proposed training is lower than the pure training of the algorithm on many decades (see Fig. 8). For these simulations we set $L_d = 100$. The advantage of the hybrid training scheme with equal BW efficiency becomes even clearer for smaller data transmission phases, as can be seen in Fig. 7 with $L_d = 20$. It is worth to mention that random BPSK pilots were used to train the RLS algorithm as in [7]. Thus the results represent an averaging over all possible random pilot sequences. However, for applications where only one fixed pilot sequence is used, the results strictly depend on the applied training sequence.

C. Initialization

In this section we inspect the impact of the initialization of the autocorrelation matrix inverse $\mathbf{P}(n)$ with either the value attained at the end of the preceeding data transmission phase or as in [2], [7] with $\delta^{-1}\mathbf{I}_M$, where $\delta << 1$. We initialize $\hat{\mathbf{H}}(0)$ with the PACE estimate. As can be seen in Fig. 9, at high Doppler shift it is better to reset $\mathbf{P}(\mathbf{n})$ to $\delta^{-1}\mathbf{I}_{N\times M}$. At low Doppler shift, however, using the value attained at the end of the preceeding data transmission phase leads to 0.5dB gain. Another advantage of resetting $\mathbf{P}(\mathbf{n})$, besides enhancing the performance at high Doppler shift, is that OFF-RLS and VFF-RLS become less sensitive to the minimum value of λ. For instance, to avoid instabilities in the algorithm, λ_{min} is set empirically to 0.4 for OFF-RLS and 0.2 for VFF-RLS when $\mathbf{P}(\mathbf{n})$ is reset, whereas λ_{min} takes the values 0.68 and 0.6 respectively when $\mathbf{P}(n)$ is set to the value attained at the preceeding data transmission phase. Smaller values of λ_{min} lead to lower BER since this way we can make benefit of the adaptive forgetting factor on a larger definition interval. Same behaviour is noticed for lower Doppler shifts.

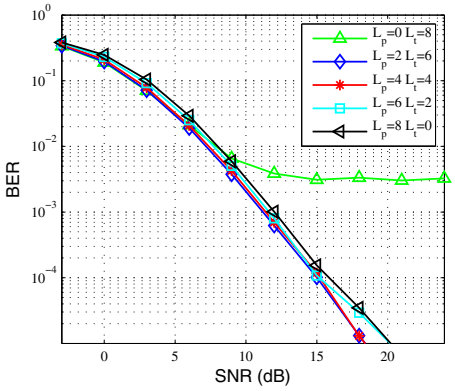

Fig. 8. BER as function of the SNR for $L_d = 100$ and $L_p + L_t = 8$ at $f_{dnorm} = 0.001$

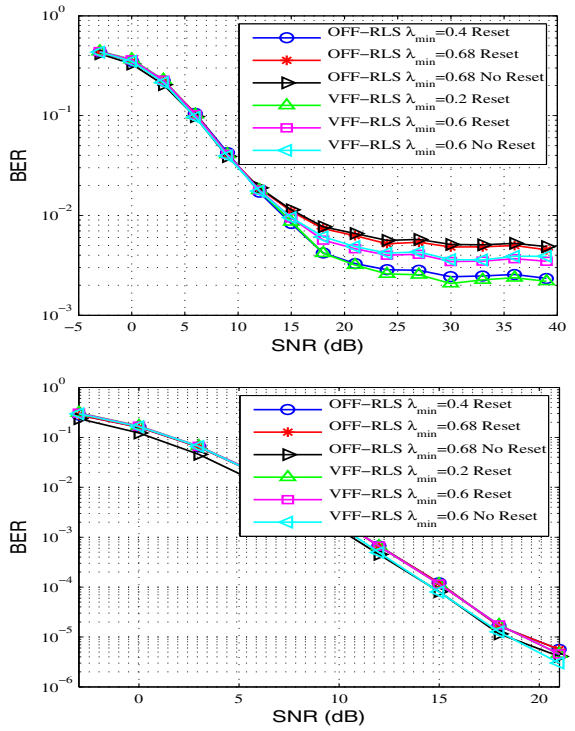

Fig. 9. Impact of initialization of the autocorrelation matrix inverse on OFF-RLS and VFF-RLS at $f_{dnorm} = 0.01$ (top) and $f_{dnorm} = 0.001$ (bottom)

V. Conclusion

In this paper, we presented three different methods to enhance the tracking capability of the RLS algorithm. The sliding window RLS algorithm can enhance the performance at the expense of optimizing the involved window size and forgetting factor. RLS with an optimal forgetting factor suffers from sensitivity to small limit values of the forgetting factor and the need for channel model knowledge. The RLS algorithm with a variable forgetting factor shows the best performance at the expense of the highest complexity. We investigated the effect of the initialization of the autocorrelation matrix inverse and the channel matrix estimate involved in the algorithm. At high Doppler shifts it is better to reset the autocorrelation matrix inverse, whereas at low Doppler shifts it is advantageous to keep the matrix that is obtained at the end of the previous data transmission phase. We introduced a new training scheme which can decrease the BER floor by many orders of magnitude in comparison to state of the art while maintaining the same bandwidth efficiency.

VI. Appendix

We present the derivation of the sequential MIMO tracking RLS algorithm. First, we define $\hat{\mathbf{X}}(n)$ as:

$$\hat{\mathbf{X}}(n) = \hat{\mathbf{H}}^H(n). \tag{14}$$

Recalling the equations of the non-sequential algorithm in Table I, we have

$$\begin{aligned} \hat{\mathbf{X}}(n) &= \mathbf{P}^H(n)\mathbf{R}^H(n) && (15)\\ &= \mathbf{P}^H(n)\mathbf{Z}(n), && (16) \end{aligned}$$

with

$$\mathbf{Z}(n) = \mathbf{R}^H(n) = \lambda\mathbf{Z}(n-1) + \hat{\mathbf{s}}(n)\mathbf{r}^H(n). \tag{17}$$

Using the fact that $\mathbf{P}(n) = \mathbf{P}^H(n)$, we get:

$$\begin{aligned} \hat{\mathbf{X}}(n) &= \mathbf{P}(n)\mathbf{Z}(n) && (18)\\ &\overset{(17)}{=} \lambda\mathbf{P}(n)\mathbf{Z}(n-1) + \mathbf{P}(n)\hat{\mathbf{s}}(n)\mathbf{r}^H(n). && (19) \end{aligned}$$

Now substituting $\mathbf{P}(n)$ from Table I into (19):

$$\begin{aligned} \hat{\mathbf{X}}(n) &= \lambda(\lambda^{-1}\mathbf{P}(n-1))\mathbf{Z}(n-1)\\ &\quad -\lambda(\lambda^{-1}\mathbf{k}(n)\hat{\mathbf{s}}^H(n)\mathbf{P}(n-1)\mathbf{Z}(n-1))\\ &\quad +\mathbf{P}(n)\mathbf{s}(n)\mathbf{r}^H(n) && (20)\\ &\overset{(19)}{=} \hat{\mathbf{X}}(n-1) - \mathbf{k}(n)\hat{\mathbf{s}}^H(n)\hat{\mathbf{X}}(n-1)\\ &\quad +\mathbf{P}(n)\hat{\mathbf{s}}(n)\mathbf{r}^H(n). && (21) \end{aligned}$$

Rearranging the gain vector equation in Table I:

$$\begin{aligned} \mathbf{k}(n) &= \lambda^{-1}(\mathbf{P}(n-1) - \mathbf{k}(n)\hat{\mathbf{s}}^H(n)\mathbf{P}(n-1))\hat{\mathbf{s}}(n) && (22)\\ &= \mathbf{P}(n)\hat{\mathbf{s}}(n). && (23) \end{aligned}$$

Using (23), we may formulate (21) as follows:

$$\begin{aligned} \hat{\mathbf{X}}(n) &= \hat{\mathbf{X}}(n-1) - \mathbf{k}(n)\hat{\mathbf{s}}^H(n)\hat{\mathbf{X}}(n-1)\\ &\quad +\mathbf{k}(n)\mathbf{r}^H(n) && (24)\\ &= \hat{\mathbf{X}}(n-1) + \mathbf{k}(n)[\mathbf{r}^H(n) - \hat{\mathbf{s}}^H(n)\hat{\mathbf{X}}(n-1)] && (25) \end{aligned}$$

The expression $[\mathbf{r}^H(n) - \hat{\mathbf{s}}^H(n)\hat{\mathbf{X}}(n-1)]$ is the Hermitian of the a priori error vector $\mathbf{e}(n) = \mathbf{r}(n) - \hat{\mathbf{H}}(n-1)\hat{\mathbf{s}}(n)$. Therefore we can rewrite (25) in the form:

$$\hat{\mathbf{X}}(n) = \hat{\mathbf{X}}(n-1) + \mathbf{k}(n)\mathbf{e}^H(n). \tag{26}$$

Finally, with (14), the sequential expression for $\hat{\mathbf{H}}(n)$ reads:

$$\hat{\mathbf{H}}(n) = \hat{\mathbf{H}}(n-1) + \mathbf{e}(n)\mathbf{k}^H(n). \tag{27}$$

References

[1] M. Coldrey and P. Bohlin, "Training-based MIMO systems: Part II improvements using detected symbol information," *IEEE Transactions on Signal Processing*, vol. 56, no. 1, pp. 296–303, Jan. 2008.

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

[3] H. Liu and Z. He, "A sliding-exponential window RLS adaptive filtering algorithm: properties and applications," *Signal Processing*, pp. 357 – 366, Sept. 1995.

[4] M. Bu and J. Chen, "Sliding-windowed Recursive Least-Square algorithm and its behaviors in stationary and non-stationary environments," *Proc. of the International Conf. on Circuits and Systems*, June 1991.

[5] K. Maouche and D. T. M. Slock, "The generalized sliding window recursive least-square (GSW RLS)," *IWAENC'95, 4th International Workshop on Acoustic Echo and Noise Control - 21-23, Roros, Norway*, Jun 1995.

[6] J. Lin, J. G. Proakis, F. Ling, and H. Lev-Ari, "Optimal tracking of time varying channels: A frequency domain approach for known and new algorithms," *IEEE Transactions on Selected Areas in Communications*, vol. 13, no. 1, Jan. 1995.

[7] E. Karami and M. Shiva, "Decision-directed Recursive Least Squares MIMO Channel Tracking," *EURASIP Journal on Wireless Communications and Networking*, Dec. 2005.

[8] E. Karami, "Tracking Performance of Least Squares MIMO Channel Estimation Algorithm," *IEEE Transactions on Communications*, vol. 55, no. 11, pp. 2201–2209, Nov. 2007.

[9] S. Song, J. Lim, S. Baeck, and K. Sung, "Gauss Newton variable forgetting factor Recursive Least Squares for time varying parameter tracking," *Electronics letters*, vol. 36, no. 11, May 2000.