Adaptive Chip Level Equalization for HSDPA

Ahmet Baştuğ
Philips Semiconductors
06560, Sophia Antipolis, FRANCE
tel: 33-492944130 fax: 33-492961280
ahmet.bastug@philips.com

Stefania Sesia
Philips Semiconductors
06560, Sophia Antipolis, FRANCE
tel: 33-492962303 fax: 33-492961280
stefania.sesia@philips.com

Dirk T.M. Slock
Eurecom Institute
06904, Sophia Antipolis, FRANCE
tel: 33-493008106 fax: 33-493008200
slock@eurecom.fr

***Abstract*— We consider a chip level decision-directed NLMS equalization scheme which targets estimating the total transmitted base station chip sequence in a decision-directed manner and using it as the desired response for equalizer adaptation. For this purpose, we explicitly use only the knowledge of the user-assigned HSPDSCH codes in order to obtain reliable signal components by hard decisions. By exploiting the equivalence between the actual multirate transmission in the sense of containing multiple spreading factors and the multicode pseudo-transmission at the single HSDPA spreading level we use also the estimated pseudo-symbols of other codes via LMMSE weightings. In addition to its reasonable complexity and Max-SINR achieving performance in realistic HSDPA working regimes, the proposed scheme also has the advantage of not requiring the channel parameters. We evaluate its performance by extensive simulations vis-à-vis the Griffiths equalizer which requires channel parameters.**

I. INTRODUCTION

High Speed Packet Data Access (HSDPA) service has been standardized in the Release-5 of UMTS FDD standard [1], [2]. In HSDPA, one or more of the High Speed Physical Downlink Shared Channels (HSPDSCHs) at spreading factor level 16 (SF-16), in particular 1,5,10 or 15 of the 16 available codes, are dynamically time multiplexed (scheduled) among users, preferably all allocated to a single user at any time. The goal is to exploit *multiuser diversity*, i.e. the temporal channel quality variance among the users, in order to increase the *sum capacity*, that is the total delivered payload by the BS. The choice for the type of scheduling mechanism is left to the operators. The criterion for the choice is the compromise between throughput and fairness [3], [4].

The initial stages of the UMTS development utilized only the Rake receiver due to its simplicity [5]. Although it is well known that Rake receiver is far from being optimal in multipath-rich environments, mobile vendors and chip manufacturers have been reluctant to switch to a more advanced and hence a more costly solution for the sole benefit of the base station side. In HSDPA, however, such a solution is meaningful due to the fact that the mobile terminal directly benefits from it by not only obtaining more data rate on HSPDSCHs once a connection is established during the scheduling process but also by increasing the probability of getting a connection if fairness is partially sacrificed for throughput in user scheduling. In any case it is inevitable since Rake receiver is not satisfying the 3GPP test performance requirements [6].

One advanced receiver option is the linear minimum mean square error (LMMSE) receiver [7]. However it is quite complex to implement in UMTS FDD mobile terminals since it needs the code and the amplitude knowledge of all the active users. Furthermore LMMSE solution changes for every chip. Max-SINR receiver which is also known as the chip level LMMSE equalizer followed by descrambling and despreading operations is a slightly suboptimal but much simpler alternative derived from the LMMSE receiver by modeling the scrambler as a stationary random sequence [8], [9]. Because of these reasons it is considered as the baseline receiver for HSDPA [6]. Max-SINR receiver can be implemented in several different ways. In this paper we are covering a chip level NLMS implementation which has prominent features specific to HSDPA.

II. TRANSMISSION MODEL

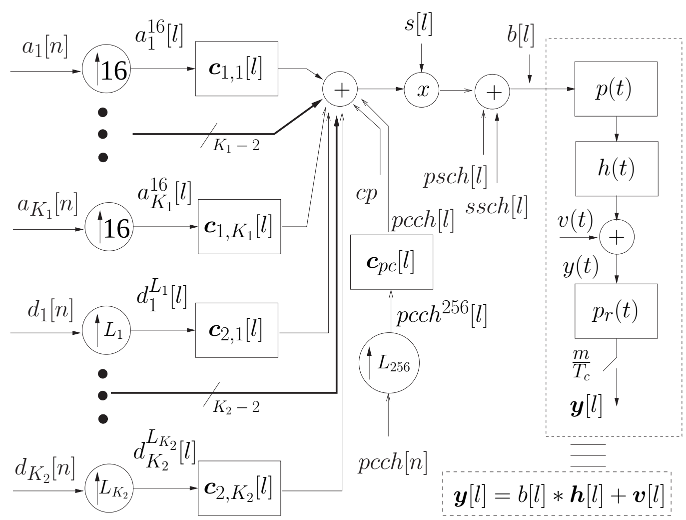

Fig. 1. Baseband downlink transmission and reception model

The baseband downlink transmission model of the UMTS-FDD mode system with HSDPA support is given in Fig. 1. K_1 HSPDSCH codes ($c_{1,j}[l], j \in \{1, \ldots, K_1\}; 1 \leq K_1 \leq 15$) are allocated at SF=16 for the transmission of linearly modulated $a_j[n]$ HSDPA symbols of user of interest. $d_j[n], j \in \{1, \ldots, K_2\}$ represent the symbols of possibly existing multirate downlink channels like dedicated channels (DCHs), HSSCHs and others except the primary common control physical channel (PCCPCH), constant valued primary common pilot channel (PCIPCH), primary and secondary synchronization channels (PSCH and SSCH) whose chip sequences are demonstrated as $pcch[l]$, cp, $psch[l]$ and $ssch[l]$ respectively in Fig. 1. At the base station (BS) transmitter, symbols to be carried on a certain channel are first upsampled ($\uparrow$ operator) by their SF and convolved with the corresponding unit-energy Walsh-Hadamard periodic channelization code before summed up with the chip sequence of other channels and multiplied with the unit magnitude BS-specific aperiodic scrambling code $s[l]$. PSCH and SSCH are exceptions, multiplexed late after the scrambler, since as a first-step task in the reception, they are actually utilized for determining (searching) the scrambling sequence of the BS.

When there is no beamforming, the chip sequences of all the users connected to the same BS, i.e $b[l] = \sum_k b_k[l]$, pass through a common chip-rate finite impulse response (FIR) channel $\boldsymbol{h}$ of size $1 \times P$ which turns into a *multi-channel* (a single input multi output (SIMO) system with memory) of size $mq \times P$ in the presence of multiple (q) antennas or oversampling (by an integer factor m) at the mobile side. Continuous time counterpart of this discrete channel is the overall convolution of the root-raised-cosine pulse shape $p(t)$ that has 0.22 roll-off factor, the propagation channel $h(t)$ and the front end receiver $p_r(t)$. In case of oversampling, if one wants to preserve the whiteness of the additive white Gaussian noise (AWGN), $p_r(t)$ must be a low pass filter (LPF) with BW equal to the sampling rate. At chip rate sampling, however, $p_r(t)$ should be the pulse shape matched filter in order to satisfy the Nyquist criterion.

The received signal is the superposition of the channel-distorted BS signal $b[l] * \boldsymbol{h}[l]$ and an additional *noise plus intercell interference* term $\boldsymbol{v}[l]$ which in practice is assumed to be approximately *white*: $\boldsymbol{y}[l] = b[l] * \boldsymbol{h}[l] + \boldsymbol{v}[l]$. When we model the scramblers as unknown, i.i.d, aperiodic sequences and the symbol sequences as i.i.d., stationary, white sequences, then the transmitted total BS chip sequence $b[l]$ is also stationary and white. Therefore, the two components of $\boldsymbol{y}[l]$ are vector stationary processes the continuous-time counterparts of which are cyclostationary with chip period. Hence, $\boldsymbol{y}[l]$ is vector stationary (with the assumption that channel is not changing, we will visit this issue later), making chip rate LMMSE filtering feasible as will be explained next.

III. Max-SINR (Chip Level LMMSE) Receiver

$1 \times mqN$ *polyphase* LMMSE filter $\boldsymbol{f}_{mmse}$ with the length $N \geq P$ that maximizes the SINR (i.e. that minimizes the MSE) for $\hat{b}[l - l_d]$ by jointly suppressing (compromising) intracell interference and white noise can be derived as [9]

$$\boldsymbol{f}_{mmse} = \boldsymbol{R}_{b\boldsymbol{y}} \boldsymbol{R}_{\boldsymbol{yy}}^{-1} = \sigma_b^2 \acute{\boldsymbol{h}} \boldsymbol{R}_{\boldsymbol{yy}}^{-1} \quad (1)$$

$$\boldsymbol{R}_{\boldsymbol{yy}} = E\left[(\boldsymbol{\mathcal{T}}(\boldsymbol{h})\boldsymbol{b} + \boldsymbol{v})(\boldsymbol{b}^H \boldsymbol{\mathcal{T}}(\boldsymbol{h})^H + \boldsymbol{v}^H)\right] \quad (2)$$
$$= \sigma_b^2 \boldsymbol{\mathcal{T}}(\boldsymbol{h}) \boldsymbol{\mathcal{T}}(\boldsymbol{h})^H + \sigma_v^2 \boldsymbol{I}$$

$$\Rightarrow \boldsymbol{f}_{mmse} = \sigma_b^2 \acute{\boldsymbol{h}} (\sigma_b^2 \boldsymbol{\mathcal{T}}(\boldsymbol{h}) \boldsymbol{\mathcal{T}}(\boldsymbol{h})^H + \sigma_v^2 \boldsymbol{I})^{-1} \quad (3)$$

where $\boldsymbol{\mathcal{T}}(\boldsymbol{h})$ is the $mqN \times (N+P-1)$ block Toeplitz channel convolution matrix with block size $mq \times 1$, $\boldsymbol{b}$ is the $(N+P-1) \times 1$ block of transmitted $b[l]$ sequence, σ_b^2 is the variance of $b[l]$, $\boldsymbol{R}_{\boldsymbol{yy}}$ is the $mqN \times mqN$ autocorrelation matrix of the received polyphase signal
$\boldsymbol{y_l} = [y_1[l], y_2[l], \ldots, y_{mq}[l], y_1[l-1], \ldots, y_{mq}[l-N+1]]^T$, $\acute{\boldsymbol{h}}$ is the zero-padded and conjugated polyphase form of $\boldsymbol{h}$ as $[\boldsymbol{0}_{1 \times mq(l_d-P)}, h_{1,1}, h_{2,1}, \ldots, h_{mq,1}, h_{1,2}, \ldots, h_{mq,P}$ $, \boldsymbol{0}_{1 \times mq(N-l_d)}]^*$, l_d is the filter delay and $\boldsymbol{R}_{b\boldsymbol{y}}$ is the $1 \times mqN$ crosscovariance vector between the received signal and the BS chips.

There are several ways to implement the Max-SINR equalizer. One group of parametric methods target reliably estimating all the $\boldsymbol{h}$, σ_v^2 and σ_b^2 ingredients once every predetermined time period and calculating the $\boldsymbol{f}_{mmse}$ filter from (3). The update time period depends on the rate of change of the channel, i.e. on the Doppler frequency, and depends on the birth and the death rate of the users. These methods have the advantage of precisely modeling the BS signal component $\sigma_b^2 \boldsymbol{\mathcal{T}}(\boldsymbol{h}) \boldsymbol{\mathcal{T}}(\boldsymbol{h})^H$ in $\boldsymbol{R}_{yy}$. However they ignore the color of other-cells' interference by modeling it as white noise [8], [10]. Another group of semi-parametric methods might aim to avoid this drawback by calculating the $\boldsymbol{R}_{yy}$ statistics directly from the received data, see for example such a technique in a different receiver context in [11]. Although at first sight it looks attractive, the short term sample support is not sufficient to obtain this statistics precisely, especially in highly time-varying channel conditions. In addition to being too parametric, solutions from these two groups also require matrix-matrix multiplication and and matrix inversion operations which both have $O(m^3 q^3 N^3)$ complexity if done in standard ways. One might argue that this filter update can be done at a low rate to decrease the complexity. In that case the receiver would not be able to track the fast varying channels. Even if only the low speed scenarios are considered, they are still not attractive for implementation neither with ASIC hardware nor with programmable vector processors. In hardware, they would occupy a lot of chip space. In software, they would put imbalanced load, making the processor MIPS scheduling troublesome. Due to all these reasons, implementations based on direct computations from the equalizer expression in (3) are not preferable. An alternative approach for equalizer implementation is adaptive filtering within which also there are several techniques associated with different optimization criteria [12], [13], [14], [15]. The two well-known adaptive techniques are RLS and LMS. RLS is not a very suitable method since it also has a high complexity of $O(m^2 q^2 N^2)$. Moreover, in general it is not numerically stable, it cannot cope with non-stationary signals and it is negatively impacted by colored noise at its input since it inherently solves the deterministic least squares problem which requires white noise for convergence to Wiener (MMSE) solution [16], [17]. Unfortunately in wireless channels neither the received signal is stationary (due to time-varying channel) nor the noise, i.e. the additive interference, is white. LMS is on the contrary advantageous regarding all the mentioned aspects: It has low complexity of $O(mqN)$, it is numerically stable and most importantly it is robust to modeling errors, disturbance variations and nonstationarities [16]. Due to these reasons, we restrict our focus to two chip level equalizers derived from the standard LMS algorithm.

IV. Chip Level Adaptive Equalizers

In this section we look at chip-spaced implementations. Extensions to the poly-phase implementations of adaptive equalizers in the case of Rx-diversity (multiple antennas) and/or fractional sampling is straightforward, see for example a poly-phase symbol level LMS implementation in [12].

$\boldsymbol{R_{yy}}$ and $\boldsymbol{R_{by}}$ statistics are useful for LMMSE filtering purpose only for stationary $\boldsymbol{y}$, which unfortunately is not the case in time-varying channels. Still the LMMSE filtering equation is a good starting point for formulating LMS recursions. One can compute $\boldsymbol{f}_{mmse}$ exactly either directly from (3) or do it by the steepest descent method [18]. In the latter case, by starting from an initial filter weights assignment $\boldsymbol{f}_0$, one approaches to $\boldsymbol{f}_{mmse}$ iteratively as

$$\boldsymbol{f}_{l+1} = \boldsymbol{f}_l - \mu \nabla \boldsymbol{f}_l \tag{4}$$

by going in the opposite direction of the instantaneous MSE gradient vector

$$\nabla \boldsymbol{f}_l = \boldsymbol{f}_l \boldsymbol{R_{yy}} - \boldsymbol{R_{by}} \tag{5}$$

which is obtained from the standard Wiener filtering MSE expression [18].

By replacing the $\boldsymbol{R_{yy}}$ and $\boldsymbol{R_{by}}$ statistics by their respective instantaneous values $\boldsymbol{y}_l \boldsymbol{y}_l^H$ and $b[l]\boldsymbol{y}_l^H$, the standard LMS algorithm turns the explained *iterative* LMMSE scheme into a *data recursive* adaptive scheme which is adapted with every incoming sample (chip) as

$$\boldsymbol{f}_{l+1} = \boldsymbol{f}_l - \mu_l(\boldsymbol{f}_l \boldsymbol{y}_l - b[l-l_d])\boldsymbol{y}_l^H = \boldsymbol{f}_l + \mu_l e_l \boldsymbol{y}_l^H \tag{6}$$

where l_d is the filter delay, μ_l is the step size (adapted as well), $b[l-l_d]$ is the desired response which we denote also as $d[l]$ and $\boldsymbol{y}_l = [y[l], y[l-1], \ldots, y[l-N+1]]^T$ is the input regression vector [18]. As seen from (6), the desired response $d[l]$ is the total BS transmitted signal, which unfortunately is not known.

Griffiths algorithm is a preferred method when a training sequence $d[l]$ is not available or is not reliable [19]. Similar to LMS, in equalizer context it also uses the total BS power and it is derived from the LMMSE equations (4) and (5), this time by replacing only the $\boldsymbol{R_{yy}}$ statistics with $\boldsymbol{y}_l \boldsymbol{y}_l^H$

$$\boldsymbol{f}_{l+1} = \boldsymbol{f}_l - \mu_l(\boldsymbol{f}_l \boldsymbol{y}_l \boldsymbol{y}_l^H - \sigma_b^2 \acute{\boldsymbol{h}}) \tag{7}$$

Normalized forms of LMS and Griffiths are obtained by normalizing the update terms with the input signal power:

$$\boldsymbol{f}_{l+1} = \boldsymbol{f}_l + \frac{\mu_l e_l \boldsymbol{y}_l^H}{\boldsymbol{y}_l^H \boldsymbol{y}_l} \text{ (NLMS)} \tag{8}$$

$$\boldsymbol{f}_{l+1} = \boldsymbol{f}_l - \frac{\mu_l(\boldsymbol{f}_l \boldsymbol{y}_l \boldsymbol{y}_l^H - \sigma_b^2 \acute{\boldsymbol{h}})}{\boldsymbol{y}_l^H \boldsymbol{y}_l} \text{ (N-Griffiths)} \tag{9}$$

N-Griffiths equalizer can be implemented directly from (9), however it requires the channel parameters. Implementation of NLMS equalizer is not that trivial since at first sight it seems that there is the single possibility of using the PCPICH as the desired signal. This conventional NLMS algorithms does not work well since the 10% of the BS signal power given to the PCPICH is too small compared to the interference level [15]. If one wants to obtain a more satisfactory performance from NLMS, one must find a way of exploiting more components from the transmitted total BS signal as the desired response.

V. Decision Directed HSDPA Equalizer

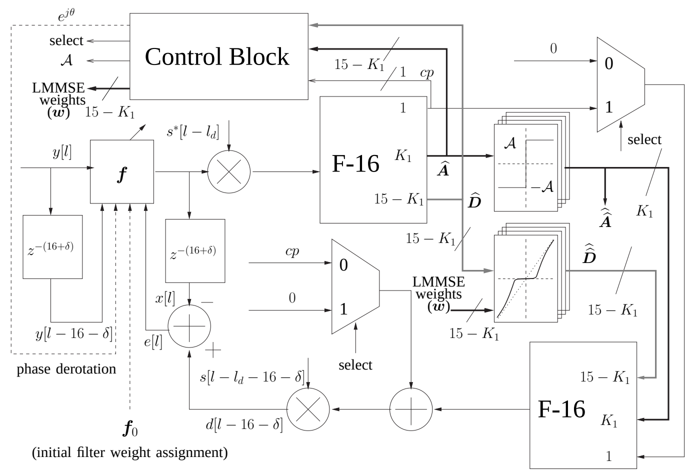

Fig. 2. HSDPA-specific decision directed NLMS (HDD-NLMS) equalizer

Fig. 2 shows a schematic block diagram of our HDD-NLMS adaptive chip level equalizer. The scheme is motivated from the decision directed LMS equalization principle for single user ISI channels, which dates back to 1966 [20]. Since, in the context of CDMA downlink chip equalization, the decisions should be done on the transmitted total BS chip sequence, this first obliges estimating all the user symbols by despreading and symbol post-processing such as hard decisions or LMMSE weightings and then returning back to chip level by respreading. These two cascade operations incur a delay of one HSPDSCH symbol period plus δ-chips-lasting feedback processing, i.e. $16+\delta$ chips, when the symbol decisions are constrained to SF-16. Therefore the same amount of delay should be applied to the input signal and the filter output. The goal is to adapt the NLMS equalizer by the desired sequence $d[l-16-\delta]$ fedback as the brute force estimate of the transmitted total BS chip sequence in the preceding symbol period. Although the data flows more than one symbol period ahead of the adaptation process, we can safely use the obtained filter weights due to the fact that this delay is a negligible time compared to the coherence time of typical wireless channels and hence the associated optimal equalizer weights do not change much during this period.

We use Fast Walsh Hadamard Transformation (FWHT) for efficiently implementing multiple despreading operations. If one wants to despread K codes with spreading factor L, using FWHT instead of K independent correlators decreases the complexity from KL units to $Llog_2(L)$ units. As long as $K > log_2(L)$, FWHT is advantageous. The crossover K value for SF-16 is $log_2(16) = 4$. Since in our system we are interested in despreading with all the 16 codes at SF-16 we use FWHT of length 16. We call it F-16 on the figure. F-16 outputs $\widehat{\boldsymbol{A}}$ associated with HSPDSCHs of the user of interest are passed through hard decision blocks and fedforward as $\hat{\hat{\boldsymbol{A}}}$ to the channel decoder and other post processing units.

We use three different means for feeding back the F-16 correlator outputs:

i. K_1 soft HSPDSCH symbol estimates $\widehat{\boldsymbol{A}}$ are passed through slicers, i.e. hard detected, and hence the resultant $\widehat{\widehat{\boldsymbol{A}}}$ are supposed to carry the most reliable components of the desired signal estimate. This is actually the case as long as mostly correct detections are made and as long as HSPDSCH symbol amplitudes are estimated precisely.

ii. The remaining correlator outputs $\widehat{\boldsymbol{D}}$, except the first one, are fedback as $\widehat{\widehat{\boldsymbol{D}}}$ scaled by separate LMMSE weights. In fact we do not know a priori the active spreading codes in the OVSF code space which are spanned by the spreading codes of these $15 - K_1$ correlators. However, as long as hard decisions or other nonlinear operations which definitely require the symbol constellations and the symbol amplitudes are not considered, one does not need to know the actual channelization (spreading) codes and one does not need to estimate their symbols. In this case it is equally sufficient to get pseudo-symbol estimates reflected from the actual symbols residing at various possible different places in the OVSF hierarchy to the SF-16 level and to apply LMMSE weights on these pseudo-symbols.

Say the instantaneous power on any correlator output with index u is $|\hat{a}_u|^2$ and the noise-plus-interference variance is $\sigma^2_{n_u}$. Then the instantaneous LMMSE weight for that output will be $w_u = \frac{|\hat{a}_u|^2 - \sigma^2_{n_u}}{|\hat{a}_u|^2}$. The numerator term corresponds to the useful signal power and the denominator term corresponds to the sum of the useful signal power and the noise-plus-interference power. If the estimated LMMSE weight on any particular branch is negative, then it is replaced by zero. This latter situation is equivalent to excluding those outputs from the feedback operation and it occurs when the power at that particular branch is below $\sigma^2_{n_u}$.

iii. The first correlator output, i.e. the cp output from despreading with the 16-ones code $c_{16,0}$, partially despreads PCPICH, PCCPCH and all the other active codes under the OVSF subtree rooted from $c_{16,0}$. There are two possible approaches here. The first option is excluding the output of this branch from the feedback operation but instead adding the PCPICH chip sequence in a hard manner since the PCPICH sequence is a known sequence. The advantage is that the added term is not noisy. However it has two disadvantages. First of all one needs to also estimate the PCPICH amplitude. Secondly and more importantly, once PCPICH is explicitly fedback, one cannot exploit the signal contributions from any other code under the OVSF subtree rooted from $c_{16,0}$. The second option is feeding back by LMMSE scaling as is done for the other remaining branches.

The multiplexing mechanism between the hard PCPICH addition and scaled linear feedback of the first correlator output requires defining a threshold value for the first correlator output power.

Lemma V.1 *Let PCPICH power be P_{cp}, the instantaneous power at the first correlator output be $|\hat{a}_1|^2$ and the LMMSE weight be $w_1 = \frac{|\hat{a}_1|^2 - \sigma^2_{n_1}}{|\hat{a}_1|^2}$. Then the optimal $|\hat{a}_1|^2$ threshold value for multiplexing PCPICH and the LMMSE weighted first correlator output is $P_{thr} = \frac{P_{cp}}{w_1^2} + 2\sigma^2_{n_1}$.*

Proof. The useful signal power at the LMMSE weighting output is $\left(|\hat{a}_1|^2 - \sigma^2_{n_1}\right) w_1^2$. The noise-plus-interference power at the LMMSE weighting output is $\sigma^2_{n_1} w_1^2$. We take the useful signal power as a reference. Then selection of pure PCPICH signal can be considered as an estimation with error variance equal to the power difference between the useful signal power and P_{cp}. That is to say if PCPCH is selected then this selection has a constant *sample variance*. In order for the two options to have the same variance, this variance should be equal to the noise-plus-interference power at the LMMSE weighting output

$$\left(|\hat{a}_1|^2 - \sigma^2_{n_1}\right) w_1^2 - P_{cp} = \sigma^2_{n_1} w_1^2 \Rightarrow P_{thr} = \frac{P_{cp}}{w_1^2} + 2\sigma^2_{n_1} \tag{10}$$

□

We add the pilot signal to the feedback path in hard manner if the power on the first branch is smaller than P_{thr} obtained in (10) which is estimated and updated once every CPICH symbol period in the *Control Block* in Fig. 2.

The overall feedback strategy improves the energy of the desired signal and allows for a better tracking of the channel. Moreover, this recursive process can be interpreted as a learning process also for the desired signal. With each recursion, the quality of filter weights and thus the detected or estimated feedback signal, i.e. the desired signal, is improved.

A. *Misconvergence Problem*

Any decision directed scheme is prone to misconvergence problem. This is a phenomenon which occurs when the equalizer locks to a rotated constellation state, does systematic errors all the time and cannot recover from there [14]. A common remedy is to have a backup solution such as a pure pilot-aided method or a constant modulus algorithm (CMA) [21], [22] which takes the turn when the equalizer diverges and gives the turn back to the decision-directed scheme when SINR conditions are again above an acceptable level [14], [23], [24], [25], [26]. Since we do not want to have any backup solution and we want instead to *avoid* misconvergence, in the *Control Block* we obtain a Super-PCPICH-Symbol, i.e. sum of a block of PCPICH-symbols, every 5 or 10 PCPICH symbol periods and derotate the equalizer filter weights by an angle θ which is equal to the difference between the phase of the estimated Super-PCPICH Symbol and 45 degrees, which is the correct phase of the pilot signal. This, interestingly enough, adds a local zero-forcing (ZF) dimension to the global

MMSE equalization problem. The PCPICH tone is a significant element, not only for avoiding misconvergence most of the time but also bringing the filter back to convergence state if misconvergence cannot be avoided in deep fades. Consider the very initialization of the adaptation, for example. We are using the channel matched filter (CMF), i.e. the Rake receiver in its FIR form for the initialization of the filter weights. This is a nice-to-have but not a strictly essential feature. Even if we start with the all-zeros filter weights, by the aid of the CPICH zero-forcing mechanism, the filter passes the transient phase and the filter output locks to the correct constellation. This of course takes longer than starting with the CMF.

VI. Simulations and Conclusions

For simulations, we consider 3GPP-RAN4 compatible HSDPA service scenarios in the UMTS FDD downlink for mobile terminals from Category 7 and Category 8 [6]. Table I shows the simulation settings. On Figures 3 to 5 $\{\hat{I}_{or}, I_{oc}, E_c\}$

TABLE I

Simulation Settings

Parameters	**Settings**
Chip rate	3.84 Mcps
Number of HSPDSCH codes	10 (All belonging to the user)
Modulation scheme	QPSK
PCPICH power	10% of the BS power (-10dB)
PCCPCH power	6.3% of the BS power (-12dB)
Total power in the first subtree	25% of the the BS power
Total HSPDSCHs power (E_c)	25% or 50% of the BS power
$\hat{I}_{or}/I_{oc}$	6dB or 10dB
OCNS power	Remaining BS power randomly distributed to the 5 remaining codes
Equalizer tap spacing	1 chip
Number of receive antennas	1
Equalizer length	24
Equalizer adaptation rate	Once every 2 chips
Transmission pulse shape	rrc with roll-off factor 0.22
Channel model	Jakes fading model
Channel power delay profile	ITU Vehicular A
Relative Path Delays [ns]	[0 310 710 1090 1730 2510]
Relative Mean Power [dB]	[0 -1 -9 -10 -15 -20]
Mobile speeds	30km/h and 120km/h
Channel update rate	Once every 16 chips

TABLE II

Channel Estimation NMSE Simulation Settings for Griffiths Equalizer

Channel Estimation Quality	$\hat{I}_{or}/I_{oc} = 10dB$		$\hat{I}_{or}/I_{oc} = 6dB$	
	30km/h	120km/h	30km/h	120km/h
Low Quality	-8	-7	-6	-5
Medium Quality	-12	-11	-10	-9
High Quality	-18	-16	-15	-13

respectively denote {chip level received BS signal power, additive white noise power modeling also the intercell interference, total power assigned to HSPDSCH codes} and {D, C, GL, GM, GH, M} respectively denote {HDD-NLMS, CMF, Griffiths with a low quality channel estimator, Griffiths with a medium quality channel estimator, Griffiths with a high quality channel estimator, Max-SINR}. CMF serves as the SNR bound and Max-SINR receiver serves as the SINR bound. Therefore, for CMF and Max-SINR we assumed that we have ideal channel information. For Griffiths, however, to have a reasonably fair comparison with HDD-NLMS scheme, we perturbed the correct channel parameters by adding random Gaussian noise complying with the normalized MSE values set in Table II. These are some judiciously chosen values taking into account the performance of channel estimation methods in [27]. They can be modified to reflect the performance of any particular channel estimation technique. For plotting convenience, we obtain BER results for each TTI, i.e. 3 UMTS slots, and we sample the instantaneously obtained SINR every slot. Both HDD-NLMS and Griffiths perform much better than the CMF. HDD-NLMS scheme performs significantly better than Griffiths in several conditions. Furthermore at high SNR regions, i.e. when far from deep fades, HDD-NLMS performance comes very close to the Max-SINR performance. The impact of channel estimation quality is clearly seen when we compare the GL, GM and GH performances. Still the differences among them are not very significant especially at 30km/h mobile speed. Therefore we conclude that Griffiths is a very robust scheme against channel estimation errors. HDD-NLMS performance is sometimes getting worse than other schemes in very deep fades. As explained in the text this is mostly attributable to not assigning a backup solution for HDD-NLMS in such cases. However one must at the same time note that during deep fades, the mobile terminal will most probably not be scheduled by the BS. Taking into account such realistic HSDPA schedulers we can exclude deep fades from comparisons. In all the other cases HDD-NLMS is a better performing solution and it does not require channel parameters. However it has also some limitations. It cannot be directly used during inactive periods when the mobile does not receive any HSDPA service. However, since UMTS downlink is a code limited system, when the mobile is not scheduled Node-B will most probably assign the same codes to the user who gets the service. Even if this is not the case it is easy to detect the active HSPDSCH codes since the code search space is only limited to 15 codes, the HSPDSCH codes are placed consecutively and the constellation is limited to only QPSK and 16-QAM modulations. Decision directed schemes are also known to have convergence problems with QAM modulations. Therefore Griffiths alone is perhaps a more proper solution for 16-QAM. However a better performing 16-QAM solution would be to multiplex the two, Griffiths serving as an eye-opener for HDD-NLMS.

References

[1] "Physical layer aspects of HSDPA," 3GPP, [Online]. Available: http://www.3gpp.org/ftp/Specs, Tech. Rep. TS 25.848.

[2] "UTRA high speed downlink packet access; overall UTRAN description," 3GPP, [Online]. Available: http://www.3gpp.org/ftp/Specs, Tech. Rep. TS25.855.

[3] H. Holma and A. Toskala, *WCDMA for UMTS*. John Wiley and Sons, 2000.

[4] P. Frenger, S. Parkvall, and E. Dahlman, “The evolution of WCDMA towards higher speed downlink packet data access,” in *Proc. of the Vehicular Technology Conf.*, October 2001.

[5] R. Price and P. E. Green, “A communication technique for multipath channels,” *Proc. of the IRE*, vol. 46, pp. 555–570, March 1958.

[6] “New work task proposal: Improved performance requirements for HSDPA UE categories 7 and 8,” 3GPP, [Online]. Available: http://www.3gpp.org, Tech. Rep. RP040375.

[7] U. Madhow and M. Honig, “MMSE interference suppression for direct-sequence spread-spectrum CDMA,” *IEEE Trans. on Communications Theory*, vol. 42, pp. 3178–3188, December 1994.

[8] M. Lenardi and D. T. M. Slock, “SINR maximizing equalizer receiver for DS-CDMA,” in *Proc. of the EUSIPCO Conf.*, Tampere, Finland, September 2000.

[9] ——, “A RAKE structured SINR maximizing mobile receiver for the WCDMA downlink,” in *Proc. of the Asilomar Conf. on Signals, Systems and Computers*, Pacific Grove, California, USA, November 2001.

[10] L. Mailaender, “Low complexity implementation of CDMA downlink equalization,” in *Proc. of the Int. Conf on 3G Mobile Communications Technologies*, March 2001, pp. 396–400.

[11] W. Chen and U. Mitra, “An improved blind adaptive MMSE receiver for fast fading DS-CDMA channels,” in *Proc. of the Globecom Conf.*, San Antonio, November 2001, pp. 758–762.

[12] F. Petre, M. Moonen, M. Engels, B. Gyselinckx, and H.D.Man, “Pilot-aided adaptive chip equalizer receiver for interference suppression in DS-CDMA forward link,” in *Proc. of the Vehicular Technology Conf.*, Sept 2000, pp. 303–308.

[13] M. Hadef, S. Weiss, and M. Rupp, “Adaptive blind multiuser DS-CDMA,” *Electronics Letters*, vol. 41, pp. 1184–1186, October 2005.

[14] A. Margetts and P. Schniter, “Adaptive chip-rate equalization of downlink multirate wideband CDMA,” *IEEE Trans. on Signal Processing*, vol. 53, no.6, June 2005.

[15] K. Hooli, M. Juntti, M. Heikkila, P. Komulainen, M. Latva-aho, and J. Lilleberg, “Chip-level channel equalization in WCDMA downlink,” *Eurasip Journal on Applied Signal Processing*, no. no.8, pp. 757–770, August 2002.

[16] B. Hassibi, *On the Robustness of LMS Filters*, least-mean-square adaptive filters ed. John Wiley and Sons, 2003, ch. 4.

[17] O. Macchi, *Adaptive Processing : The Least Mean Squares Approach with Applications in Transmission*. John Wiley and Sons, 1995.

[18] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice Hall, 1985.

[19] M. Heikkila, P.Komulainen, and J. Lilleberg, “Interference suppression in CDMA downlink through adaptive channel equalization,” in *Proc. of the Vehicular Technology Conf.*, September 1999.

[20] R. Lucky, “Techniques for adaptive equalization of digital communication systems,” *Bell Sys. Tech, Journal*, vol. 45, pp. 255–286, February 1966.

[21] Y. Sato, “A method of self-recovering equalization for multilevel amplitude-modulations systems,” *IEEE Trans. on Communications*, pp. 679–682, June 1975.

[22] J. R. Treichler and B. G. Agee, “A new approach to multipath correction of constant modulus signals,” *IEEE Trans. Acoust. Speech and Signal Processing*, vol. ASSP-31, pp. 459–472, April 1983.

[23] L. Litwin, M. Zoltowski, T. Endres, and S. Hulyalkar, “Blended CMA: smooth, adaptive transfer from CMA to DD-LMS,” in *Proc. of the Wireless Communications and Networking Conference*, New Orleans, LA-USA, September 1999.

[24] F. D. Castro, M. D. Castro, and D. Arantes, “Concurrent blind deconvolution for channel equalization,” in *Proc. of the IEEE International Conf. on Comm.*, Helsinki, Finland, June 2001, pp. 366–371.

[25] S. Chen and E. Chng, “Concurrent constant modulus algorithm and soft decision directed scheme for fractionally-spaced blind equalization,” in *Proc. of the IEEE International Conf. on Comm.*, Paris-France, June 2004, pp. 2342 – 2346.

[26] C. Papadias and D. Slock, “On the decision-directed equalization of constant modulus signals,” in *Proc. of the Asilomar Conf. on Signals, Systems and Computers*, Pacific Grove, CA, November 1994, pp. 1423–1427.

[27] A. Bastug, G. Montalbano, and D. Slock, “Common and dedicated pilot-based channel estimates combining and Kalman filtering for WCDMA terminals,” in *Proc. of the Asilomar Conf. on Signals, Systems and Computers*, November 2005.

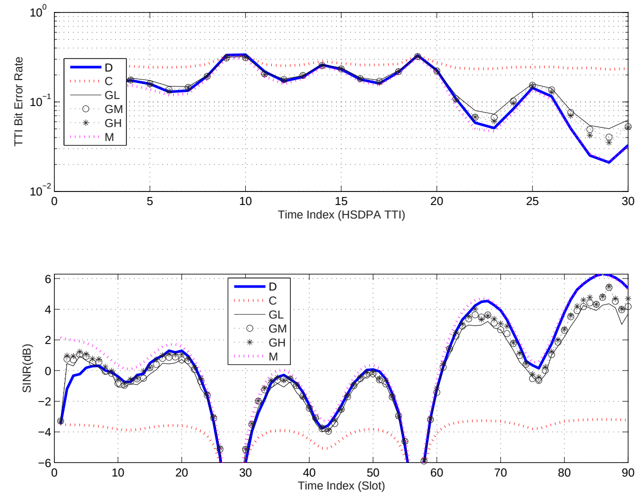

Fig. 3. vA30, 10 codes, $E_c/\hat{I}_{or} = -6dB$, $\hat{I}_{or}/I_{oc} = 10dB$

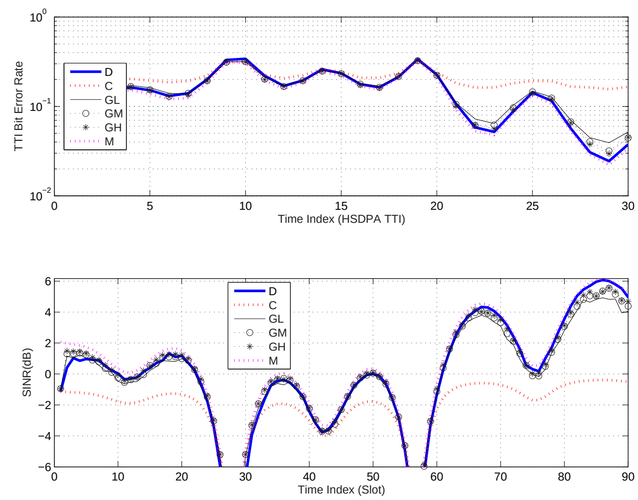

Fig. 4. vA30, 10 codes, $E_c/\hat{I}_{or} = -3dB$, $\hat{I}_{or}/I_{oc} = 6dB$

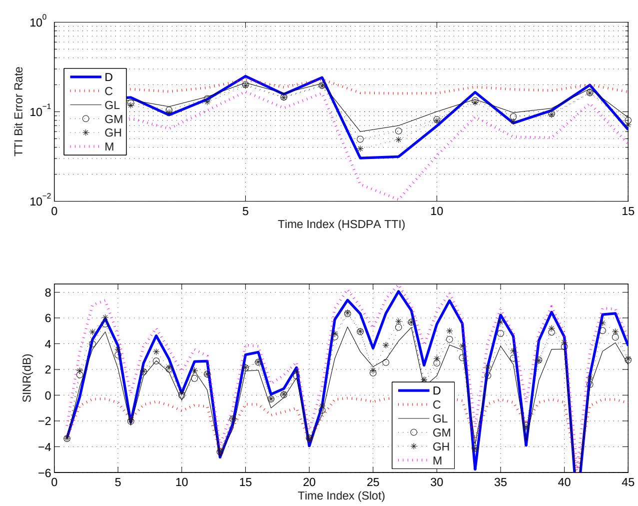

Fig. 5. vA120, 10 codes, $E_c/\hat{I}_{or} = -3dB$, $\hat{I}_{or}/I_{oc} = 10dB$