Katholieke Universiteit Leuven

Departement Elektrotechniek

ESAT-SISTA/TR 2003-168

Adaptive Bitrate Maximizing Time-Domain Equalizer Design for DMT-based Systems [1]

Koen Vanbleu, Geert Ysebaert, Gert Cuypers and Marc Moonen [2]

October 2004. Accepted for publication with minor changes.

[1] This report is available by anonymous ftp from *ftp.esat.kuleuven.ac.be* in the directory *pub/sista/vanbleu/reports/03-168.pdf*

[2] K.U.Leuven, Dept. of Electrical Engineering (ESAT), Research group SISTA, Kasteelpark Arenberg 10, 3001 Leuven-Heverlee, Belgium, Tel. 32/16/32 18 41, Fax 32/16/32 19 70, WWW: *http://www.esat.kuleuven.ac.be/sista*. E-mail: *koen.vanbleu@esat.kuleuven.ac.be*. This research work was carried out at the ESAT laboratory of the K.U. Leuven, in the frame of IUAP P5/22 and P5/11, GOA-MEFISTO-666, Research Project FWO nr.G.0196.02 and was partially sponsored by Alcatel-Bell. The scientific responsibility is assumed by its authors.

Adaptive Bitrate Maximizing Time-Domain Equalizer Design for DMT-based Systems

Koen Vanbleu*, *IEEE Student Member,* Geert Ysebaert, *IEEE Student Member,* Gert Cuypers, *IEEE Student Member,* Marc Moonen, *IEEE Member*

Address:
Katholieke Universiteit Leuven, Dept. Elektrotechniek, ESAT-SCD, Kasteelpark Arenberg 10,
B-3001 Leuven-Heverlee, Belgium
Phone +32 16 32 18 41 - *Fax* +32 16 32 19 70
E-mail: {vanbleu,ysebaert,cuypers,moonen}@esat.kuleuven.ac.be

EDICS: 3-TDSL Telephone Networks and Digital Subscriber Loops, 3-COMM Signal Processing for Communications

Abstract

The classical discrete multitone receiver as used in, e.g., DSL modems, combines a channel shortening time-domain equalizer (TEQ) with one-tap frequency-domain equalizers (FEQ). In a previous paper, we proposed a non-linear bitrate maximizing (BM) TEQ design criterion and we have shown that the resulting BM-TEQ and the closely related BM per-group equalizers (PGEQ) approach closely the performance of the so-called per-tone equalizer (PTEQ). The PTEQ is an attractive alternative that provides a separate complex-valued equalizer for each active tone. In this paper, we show that the BM-TEQ and BM-PGEQ, despite their non-linear cost criterion, can be designed adaptively, based on a recursive Levenberg-Marquardt algorithm. This adaptive BM-TEQ/BM-PGEQ makes use of the same second-order statistics as the earlier presented recursive least-squares (RLS) based adaptive PTEQ. A complete range of adaptive BM equalizers then opens up: the RLS-based adaptive PTEQ design is computationally efficient but involves a large number of equalizer taps; the adaptive BM-TEQ has a minimal number of equalizer taps at the expense of a larger design complexity; the adaptive BM-PGEQ has a similar design complexity as the BM-TEQ and an intermediate number of equalizer taps between the BM-TEQ and the PTEQ. These adaptive equalizers allow to track variations of transmission channel and noise, which are typical of a DSL environment.

Index Terms

Discrete Multitone, Digital Subscriber Lines, Adaptive Equalization, Levenberg-Marquardt, Time-Domain Equalization, Iteratively Reweighted Separable non-linear Least-Squares Problems

I. INTRODUCTION

Discrete multitone (DMT) modulation and orthogonal frequency division multiplexing (OFDM) are all-digital multicarrier modulation schemes. DMT is adopted as the transmission format for asymmetric digital subscriber lines (ADSL [1], ADSL2 [2], ADSL2+ [3]) and very high bit rate digital subscriber lines (VDSL) [4]; OFDM is proposed for wireless local area applications, e.g., HiperLAN/2 [5].

DMT schemes divide the available bandwidth into parallel subchannels or tones. The incoming bit-stream is split into parallel streams of symbols that are used to QAM-modulate the different tones. The modulation is done by means of an inverse discrete Fourier transform (IDFT). Before transmission of a DMT symbol, a cyclic prefix of ν samples is added. If the channel impulse response length is smaller than or equal to $\nu+1$, demodulation can be implemented by means of a DFT, followed by a (complex-valued) 1-tap frequency domain equalizer (FEQ) per tone to compensate for channel amplitude and phase effects.

Practical ADSL channel impulse responses can be very long, hence a long prefix would be required. However, a long prefix results in a large overhead with respect to the bitrate. An existing solution for this problem is to insert a (real-valued) T-tap time domain equalizer (TEQ) before demodulation that shortens the channel impulse response to $\nu+1$ samples. Many algorithms have been developed to design this TEQ [6], [7], [8], [9], [10], [11], [12], [13], but none of them truly optimizes bitrate.

In [14], the authors propose an attractive alternative equalizer that always performs at least as well as - and usually better than - a TEQ while keeping complexity during data transmission at the same level. A complex-valued bitrate maximizing (BM) equalizer is then designed for each tone separately, hence the term *per-tone equalizer* (PTEQ). The drawback is the memory cost: N_aT complex-valued PTEQ taps (with N_a the number of active tones, e.g., around 220 tones in downstream ADSL, and T the number of taps per tone, e.g., $T=32$) need to be stored, instead of T real-valued TEQ taps plus N_a complex-valued FEQ taps in the standard approach.

In [15], we presented a *bitrate maximizing (BM) TEQ* design criterion based on an exact subchannel signal-to-noise ratio (SNR) model. The global optimum of this non-linear criterion corresponds to a TEQ that achieves the maximum bitrate for a given number of taps. An alternative, but fully equivalent BM-TEQ design criterion has been presented in [16], based on a so-called *iteratively-reweighted separable non-linear least-squares (IR-SNLLS)* formulation. The BM-TEQ is found to closely approach the PTEQ performance. Instead of a BM-TEQ, a *BM per-group equalizer (PGEQ)* can be devised [17]: the active tones are then divided into N_g groups; the solution of the (downscaled) BM-TEQ design criterion for each group results in a BM-PGEQ for the group. This BM-PGEQ allows to trade off memory cost (N_gT taps) and performance, as it encompasses the BM-TEQ and the PTEQ as extreme cases. A BM-PGEQ with as few as 4 tone groups showed reduced susceptibility, compared to the BM-TEQ, to poorly performing local optima in harsh environments with radio-frequency interference (RFI) and crosstalk [17].

In current ADSL modems, a TEQ is typically designed *offline*: the TEQ is computed during connection set-up and is then kept fixed during data transmission. E.g., a minimum mean-square error (MMSE) TEQ with unit norm constraint [6] is obtained as the solution of an eigenvalue problem. The non-linear TEQ designs in [7], [8], [11], on the other hand, ask for complex offline batch optimization routines. A typical ADSL environment, however, varies with time (e.g., due to changing weather conditions, temperature, crosstalk and/or RFI) [18]. Moreover, the newest ADSL2 and ADSL2+ standards support "seamless rate adaptation": the ADSL line conditions are observed continuously and the bitrate and bit allocation are adapted accordingly, e.g. when an RFI is switched off [18]. As a consequence, it is desirable that the TEQ tracks the changing conditions to keep the bitrate as high as possible. *Tracking* can be achieved with a data-driven *adaptive equalizer*. An adaptive equalizer can also be used for TEQ *design*: during ADSL connection set-up, a training signal, the so-called medley signal, is transmitted [1], [2], [3], consisting of 16384 DMT symbols for ADSL and up to 32256 DMT symbols for ADSL2+. So far, few adaptive TEQ designs have been presented in the literature.

- The *MMSE-TEQ* [6] can be designed adaptively, but least-mean-square (LMS) adaptation is cited as converging slowly [19]. Alternatively, Chow's algorithm is faster but is not guaranteed to reach the optimal MMSE-TEQ [19].
- Recently, two cheap, *blind*, adaptive TEQ algorithms have been presented: the SAM algorithm [13] minimizes an estimate of the sum-squared autocorrelation of the shortened channel impulse response outside a window of length $\nu + 1$; the MERRY algorithm [12] minimizes a design criterion related to the maximum shortening SNR method [10].
- Fast and reasonably cheap adaptive *PTEQs*, based on the recursive least-squares (RLS) algorithm or a hybrid RLS/LMS algorithm (both with so-called inverse updating), have been presented in [20] and [21], respectively. An RLS-based algorithm requires second-order information; for the PTEQ, autocorrelation matrices of the sliding DFT of the receive (RX) signal are needed. In [20], it is shown that most of the RLS processing for storing and updating these second-order statistics (SOS) can be shared over all active tones. The required number of coefficients to store the SOS is $\mathcal{O}(N_a T)$; the equalizer design then requires $\mathcal{O}(N_a T)$ operations per update (including the SOS update). A further cost reduction, both in terms of SOS memory and equalizer design, is obtained by resorting to a hybrid RLS/LMS scheme [21].

In this paper, we show that the BM-TEQ can be designed *recursively* or *adaptively*, despite the non-linear and non-convex cost criterion. This may not seem obvious from the original BM-TEQ design criterion in [15]. However, the alternative IR-SNLLS-based BM-TEQ criterion of [16] is an iteratively reweighted sum of frequency-domain squared errors between the symbol estimates at the FEQ output and the true transmit (TX) symbols. The adaptive BM-TEQ algorithm is then driven by these FEQ output errors.

Section II gives an overview of our adopted notation and introduces two important key observations that will allow for an efficient adaptive BM-TEQ design method. The original BM-TEQ design criterion and the alternative IR-SNLLS-based BM-TEQ design criterion are reviewed in **Section III**. The adaptive BM-TEQ design method is developed in **Section IV**, based on the IR-SNLLS-based BM-TEQ design criterion. First, we show how this IR-SNLLS-based criterion, which depends on both the TEQ and FEQ parameters, can be optimized with an elegant, *iterative* Levenberg-Marquardt (LM) algorithm that delivers the BM-TEQ and the corresponding FEQs. Then, we introduce approximations and modifications to optimize the IR-SNLLS-based criterion *adaptively* by updating the joint TEQ-FEQ parameter vector based on a *recursive Levenberg-Marquardt* (RLM) algorithm [22], supplemented with an efficiently adapted regularization parameter to guarantee stability. This RLM algorithm has the same SOS memory cost ($\mathcal{O}(N_aT)$) as the RLS-based adaptive PTEQ design. An adaptive BM-PGEQ solves several downscaled adaptive BM-TEQ problems in parallel. A complete range of *adaptive BM equalizers* then opens up: from a single T-tap adaptive TEQ over an intermediate N_gT-tap adaptive PGEQ to a (complex-valued) N_aT-tap adaptive PTEQ. All designs have the same SOS memory cost, but a varying equalizer design cost. The adaptive BM-TEQ and BM-PGEQ require $\mathcal{O}(N_aT^2)$ computations per update, i.e., roughly T times more than a PTEQ update. Similar to the hybrid RLS/LMS-based adaptive PTEQ of [21], the SOS memory and equalizer design cost can then be further reduced. In **Section V**, the overall memory cost and computational complexity of the adaptive BM equalizers is summarized. A clear distinction is made between memory and computational cost related to *equalizer filtering* on the one hand versus *equalizer updating or design* on the other hand. Complexity figures for typical ADSL specifications are included to show that the adaptive BM-TEQ and BM-PGEQ are a viable alternative for the adaptive PTEQ, despite their larger design complexity. To support the adequacy of the adaptive BM-TEQ, **Section VI** shows extensive simulations for various typical ADSL scenarios and parameter settings. This includes simulations for the adaptive BM-TEQ, BM-PGEQ and PTEQ, scenarios with and without RFI, optimization with different starting points and tracking simulations. **Section VII** concludes the paper.

II. Notation and key observations

In this section, we introduce our notation as well as two key observations that we will frequently exploit.

- $\mathcal{S}_a$ is the set of N_a active tones, e.g., tones 33 to 256 for downstream ADSL; n is the tone index; in the case of a PGEQ, the total numer of tone groups is N_g.
- N is the (I)DFT size; $\mathcal{F}_N$ is a unitary N-point DFT matrix; $\mathcal{F}_{\mathcal{S}_a}$ is a submatrix of $\mathcal{F}_N$ with the N_a rows corresponding to the active tones $\mathcal{S}_a$; the n-th row of $\mathcal{F}_N$ is $\mathcal{F}_n$ with $n \in [0, N-1]$.
- $\mathbf{w}$ is the time-domain equalizer (TEQ, T taps). The notation below is given for a TEQ $\mathbf{w}$, but easily adapted for a PGEQ $\mathbf{w}_g$ or PTEQ $\mathbf{w}_n$. In the derivations, we mostly assume a complex-valued

TEQ for reasons of conciseness and add comments on the real-valued TEQ case. $\tilde{\mathbf{d}}$ is the $N_a \times 1$ vector of FEQs; $\tilde{d}_n$ is the FEQ for tone n. The joint TEQ-FEQ parameter vector is denoted as $\boldsymbol{\theta} = \begin{bmatrix} \mathbf{w}^H & \tilde{\mathbf{d}}^H \end{bmatrix}^H$.

- A tilde over a variable distinguishes frequency-domain symbols from time-domain symbols; vectors are typeset in bold lowercase while matrices are in bold uppercase. In general, the entry of a frequency-domain vector $\tilde{\mathbf{a}}$ corresponding to tone n is denoted with a subscript n, e.g., $\tilde{a}_n$.
- k is the DMT symbol index. The $N_a \times 1$ k-th TX symbol vector that is fed to the modulating IDFT is $\tilde{\mathbf{x}}_k$; the TX symbol on tone n is $\tilde{x}_{k,n}$.
- $\mathcal{E}\{\cdot\}$ is the expectation operator; $\mathbf{I}_m$ is the $m \times m$ identity matrix; $(\cdot)^T$, $(\cdot)^H$, $(\cdot)^*$ denote the transpose, Hermitian and complex conjugate operator, respectively; $\Re\{\cdot\}$ is the real operator; $\mathbf{a} \odot \mathbf{b}$ is the pointwise multiplication of the elements of the vectors $\mathbf{a}$ and $\mathbf{b}$; $\mathrm{diag}(\mathbf{a})$ is a diagonal matrix with $\mathbf{a}$ along the diagonal; $\mathbf{a}[l]$ is the l-th element of $\mathbf{a}$, where the index starts at zero; $\mathbf{A}[k,l]$ is the element of $\mathbf{A}$ on position (k,l).

Define a vector of RX samples i to j of the k-th DMT symbol as

$$\mathbf{y}_{k,i:j} = \begin{bmatrix} y_{k,i} & \cdots & y_{k,j} \end{bmatrix}^T \tag{1}$$

where it is tacitly assumed that the samples $y_{k,l} = y_{(k-1)(N+\nu)+\nu+l}$ (with ν the cyclic prefix length) also depend on a synchronization delay Δ, i.e., a design parameter that estimates the channel group delay [19]. Then, the $N \times 1$ sample vector at the TEQ output that is fed to the DFT is the result of the matrix-vector product $\mathbf{y}_{k,\mathbf{w}} = \mathbf{Y}_k \mathbf{w}$; the matrix $\mathbf{Y}_k$ is Toeplitz (of size $N \times T$) with RX samples $y_{k,l}$:

$$\mathbf{Y}_k = \begin{bmatrix} y_{k,0} & \cdots & y_{k,-T+1} \\ \vdots & \ddots & \vdots \\ y_{k,N-1} & \cdots & y_{k,N-T} \end{bmatrix} \tag{2}$$

hence $\mathbf{y}_{k,\mathbf{w}} = \mathbf{Y}_k \mathbf{w}$ corresponds to the convolution of the k-th RX sample vector $\mathbf{y}_{k,-T+1:N-1}$ and the TEQ. The $N_a \times 1$ DFT output vector and the DFT output on tone n are given by, respectively,

$$\tilde{\mathbf{y}}_{k,\mathbf{w}} = \mathcal{F}_{\mathcal{S}_a} \mathbf{Y}_k \mathbf{w} \text{ and } \tilde{y}_{k,n,\mathbf{w}} = \mathcal{F}_n \mathbf{Y}_k \mathbf{w} \tag{3}$$

A **first key observation** [14], underlying the PTEQ and BM-TEQ, is that the DFT output $\tilde{\mathbf{y}}_{k,\mathbf{w}}$ can also be obtained as a linear combination $\mathbf{w}$ of the T outputs of an N-point sliding DFT, $\tilde{\mathbf{Y}}_k = \mathcal{F}_{\mathcal{S}_a} \mathbf{Y}_k$, applied to the unequalized RX sample vector $\mathbf{y}_{k,-T+1:N-1}$. This can be summarized mathematically by means of the associativity property in the following equalities [14]:

$$\boxed{\tilde{\mathbf{y}}_{k,\mathbf{w}} = \mathcal{F}_{\mathcal{S}_a} \underbrace{(\mathbf{Y}_k \mathbf{w})}_{\mathbf{y}_{k,\mathbf{w}}} = \underbrace{(\mathcal{F}_{\mathcal{S}_a} \mathbf{Y}_k)}_{\tilde{\mathbf{Y}}_k} \mathbf{w} \text{ and } \tilde{y}_{k,n,\mathbf{w}} = \mathcal{F}_n \underbrace{(\mathbf{Y}_k \mathbf{w})}_{\mathbf{y}_{k,\mathbf{w}}} = \underbrace{(\mathcal{F}_n \mathbf{Y}_k)}_{\tilde{\mathbf{y}}_{k,n}} \mathbf{w}} \tag{4}$$

for the $N_a \times 1$ DFT output vector and for a particular tone n, respectively. The sliding DFT $\mathcal{F}_{\mathcal{S}_a}\mathbf{Y}_k$ (i.e., the DFT of the T columns of $\mathbf{Y}_k$) is denoted as $\tilde{\mathbf{Y}}_k$ ($N_a \times T$ matrix) and the n-th sliding DFT output is a $1 \times T$ row vector $\tilde{\mathbf{y}}_{k,n} = \mathcal{F}_n\mathbf{Y}_k$. This first key observation allows to decouple the SOS and the parameter vectors, e.g., see (11). The estimate of $\tilde{\mathbf{x}}_k$ at the FEQ outputs and the estimate of $\tilde{x}_{k,n}$ at the n-th FEQ output are then given by

$$\hat{\tilde{\mathbf{x}}}_k = \mathrm{diag}(\tilde{\mathbf{d}})\tilde{\mathbf{y}}_{k,\mathbf{w}} \text{ and } \hat{\tilde{x}}_{k,n} = \tilde{d}_n\tilde{y}_{k,n,\mathbf{w}} \tag{5}$$

A **second key observation** [14] will result in a reduction of the computational and memory cost. The n-th sliding DFT output $\tilde{\mathbf{y}}_{k,n}$ in (4) can be computed efficiently (avoiding the computation of T consecutive DFTs of the Toeplitz matrix $\mathbf{Y}_k$ in (2)), by making use of the following recursion:

$$\tilde{\mathbf{y}}_{k,n}[t] = \alpha_n\tilde{\mathbf{y}}_{k,n}[t-1] + \underbrace{(y_{k,-t} - y_{k,N-t})}_{\Delta\mathbf{y}_k[T-t-1]},\ t = 1, \cdots, T-1 \tag{6}$$

where $\alpha_n = \exp(-j2\pi(n-1)/N)/\sqrt{N}$. From (6), it follows that

$$\boxed{\tilde{\mathbf{y}}_{k,n} = \underbrace{\begin{bmatrix} \Delta\mathbf{y}_k^T & \tilde{y}_{k,n} \end{bmatrix}}_{\mathbf{z}_{k,n}} \underbrace{\begin{bmatrix} 0 & \cdots & 0 & 1 \\ \vdots & \iddots & \iddots & \alpha_n \\ 0 & 1 & \iddots & \vdots \\ 1 & \alpha_n & \cdots & \alpha_n^{T-1} \end{bmatrix}}_{\mathbf{T}_n} = \mathbf{z}_{k,n}\mathbf{T}_n} \tag{7}$$

i.e., $\tilde{\mathbf{y}}_{k,n}$ is a linear combination of $T-1$ tone-independent real-valued difference terms

$$\Delta\mathbf{y}_k \ = \ \begin{bmatrix} (y_{k,-T+1} - y_{k,-T+N+1}) & \cdots & (y_{k,-1} - y_{k,N-1}) \end{bmatrix}^T \tag{8}$$

and the n-th DFT output of the RX sample vector $\mathbf{y}_{k,0:N-1}$:

$$\tilde{\mathbf{y}}_k = \mathcal{F}_{\mathcal{S}_a}\mathbf{y}_{k,0:N-1} \text{ and } \tilde{y}_{k,n} = \mathcal{F}_n\mathbf{y}_{k,0:N-1} = \tilde{\mathbf{y}}_{k,n}[0] \tag{9}$$

An efficient computation of the sliding DFT $\tilde{\mathbf{y}}_{k,n}$ for *all* N_a active tones $\mathcal{S}_a$ then requires 1 DFT (9), $T-1$ differences (8) and $N_a(T-1)$ recursions (6).

Based on the above key observations, we will use the following SOS:

$$\sigma^2_{n,\tilde{x}} = \mathcal{E}\left\{|\tilde{x}_{k,n}|^2\right\},\ \mathbf{\Sigma}^2_{n,\tilde{\mathbf{y}}} = \mathcal{E}\left\{\tilde{\mathbf{y}}^H_{k,n}\tilde{\mathbf{y}}_{k,n}\right\},\ \boldsymbol{\sigma}_{n,\tilde{x}\tilde{\mathbf{y}}} = \mathcal{E}\left\{\tilde{x}^*_{k,n}\tilde{\mathbf{y}}_{k,n}\right\} \tag{10}$$

are the scalar variance of the TX symbol $\tilde{x}_{k,n}$, the autocorrelation matrix of the n-th sliding DFT output $\tilde{\mathbf{y}}_{k,n}$ and the $1 \times T$ crosscorrelation vector between $\tilde{x}_{k,n}$ and $\tilde{\mathbf{y}}_{k,n}$, respectively. E.g., it follows from (4) and (10) that

$$\mathcal{E}\left\{|\tilde{y}_{k,n,\mathbf{w}}|^2\right\} = \mathbf{w}^H\mathbf{\Sigma}^2_{n,\tilde{\mathbf{y}}}\mathbf{w} \text{ and } \mathcal{E}\left\{\tilde{x}^*_{k,n}\tilde{y}_{k,n,\mathbf{w}}\right\} = \boldsymbol{\sigma}_{n,\tilde{x}\tilde{\mathbf{y}}}\mathbf{w} \tag{11}$$

To distinguish between *true* (stochastic) SOS and (deterministic) SOS *estimates*, we denote an estimate obtained by averaging over K DMT symbols by a subscript K, e.g., $\mathbf{\Sigma}^2_{K,n,\tilde{\mathbf{y}}}$, and an exponentially weighted average at time k by a subscript k, e.g., $\boldsymbol{\sigma}_{k,n,\tilde{x}\tilde{\mathbf{y}}}$.

III. BM-TEQ DESIGN CRITERION

In this section, we review the BM-TEQ criterion as originally presented in [15] and the alternative, but fully equivalent IR-SNLLS-based design criterion of [16].

A. Original BM-TEQ criterion

For the derivation of a BM-TEQ design criterion in [15], we start from the same bitrate expression as in [7]. The total number of bits transmitted in one DMT symbol is given by

$$b_{DMT} = \sum_{n\in\mathcal{S}_a} \log_2\left(1+\frac{\mathrm{SNR}_n}{\Gamma_n}\right) \tag{12}$$

where SNR_n represents the SNR on tone n and Γ_n is the SNR gap between SNR_n and the SNR required to achieve Shannon capacity. Γ_n is a function of the desired probability of error, the coding gain and the system margin, and is typically assumed to be independent of the equalizer [7].

When deriving a BM-TEQ design criterion, it is crucial that the dependence of SNR_n on the TEQ $\mathbf{w}$ is accurately taken into account. Whereas in [7], [8], [9], [11], the authors define SNR_n at the DFT output, we prefer to consider the signal and residual error energies *at the FEQ output*. This is equally valid as the 1-tap FEQs do not alter SNR_n in the sense that they equally scale both the desired signal and residual error signal, hence both the numerator and denominator of SNR_n. This effectively makes the FEQs appear in the subchannel SNR model. As explained in [15], unbiased MMSE (uMMSE) FEQs are the natural choice in DMT. They depend in an elegant way on the TEQ, hence the FEQs can be written as a function of $\mathbf{w}$, using the first key observation (4) and the SOS in (10) and (11):

$$\tilde{d}_n = \tilde{d}_n^{\mathrm{uMMSE}} = \frac{\mathcal{E}\left\{|\tilde{x}_{k,n}|^2\right\}}{\mathcal{E}\left\{\tilde{x}^*_{k,n}\tilde{y}_{k,n,\mathbf{w}}\right\}} = \frac{\sigma^2_{n,\tilde{x}}}{\boldsymbol{\sigma}_{n,\tilde{x}\tilde{\mathbf{y}}\mathbf{w}}} \tag{13}$$

The uMMSE FEQs (13), together with the FEQ output (5) and the first key observation (4), provide a convenient way of modelling the signal and residual error energy and lead to an *exact* SNR model:

$$\mathrm{SNR}_n = \frac{\sigma^2_{n,\tilde{x}}}{\mathcal{E}\left\{\left|\tilde{e}_{k,n}^{\mathrm{uMMSE}}\right|^2\right\}} = \frac{\sigma^2_{n,\tilde{x}}}{\mathcal{E}\left\{\left|\tilde{d}_n^{\mathrm{uMMSE}}\tilde{\mathbf{y}}_{k,n}\mathbf{w}-\tilde{x}_{k,n}\right|^2\right\}} \tag{14}$$

where the denominator is the mean-square error (MSE) at the n-th FEQ output. Combining the bitrate expression (12), the SNR model (14) and the uMMSE FEQs (13) and making use of the SOS (10) and

(11), the following non-linear **BM-TEQ criterion** is obtained:

$$\boxed{\arg\max_{\mathbf{w}} b_{DMT} = \arg\min_{\mathbf{w}} \sum_{n\in\mathcal{S}_a} \log\left(\frac{\mathbf{w}^H\mathbf{B}_n\mathbf{w}}{\mathbf{w}^H\mathbf{A}_n\mathbf{w}}\right)} \tag{15}$$

where the tone-dependent matrices $\mathbf{A}_n$ and $\mathbf{B}_n$ are independent of $\mathbf{w}$ and given by:

$$\mathbf{A}_n = \Gamma_n\sigma^2_{n,\tilde{x}}\mathbf{\Sigma}^2_{n,\tilde{\mathbf{y}}} + (1-\Gamma_n)\boldsymbol{\sigma}^H_{n,\tilde{x}\tilde{\mathbf{y}}}\boldsymbol{\sigma}_{n,\tilde{x}\tilde{\mathbf{y}}} \tag{16}$$

$$\mathbf{B}_n = \Gamma_n\left(\sigma^2_{n,\tilde{x}}\mathbf{\Sigma}^2_{n,\tilde{\mathbf{y}}} - \boldsymbol{\sigma}^H_{n,\tilde{x}\tilde{\mathbf{y}}}\boldsymbol{\sigma}_{n,\tilde{x}\tilde{\mathbf{y}}}\right) \tag{17}$$

Minimizing (15) is a non-linear, non-convex optimization problem with local optima, for which non-linear, offline, batch optimization methods are abundantly available.

The BM-TEQ criterion (15) can also be used for the bitrate maximizing per-group equalizer (BM-PGEQ) [17], which is intermediate (in terms of memory cost and performance) between the BM-TEQ and the PTEQ: the active tones are divided into N_g groups and each group is provided with a (possibly complex-valued) T-tap BM-PGEQ $\mathbf{w}_g$ by solving the non-linear BM-TEQ criterion (15) for this group. As the PGEQ for each tone group optimizes the bitrate for the group, the BM-PGEQ performance lies between the BM-TEQ and the PTEQ performance. The BM-PGEQ is less susceptible than the BM-TEQ to poorly performing local optima of the non-linear cost function, especially in harsh environments with RFI. A BM-PGEQ with as few as $N_g = 4$ tone groups closely approaches the PTEQ performance [17], while requiring only N_gT equalizer taps. In the derivations below, we mainly focus on the BM-TEQ, but all results are easily ported to the BM-PGEQ.

B. IR-SNLLS-based BM-TEQ design criterion

It follows from Section III-A that the BM-TEQ design criterion (15) corresponds to the following constrained non-linear optimization problem in the joint TEQ-FEQ parameter vector $\boldsymbol{\theta}$:

$$\max_{\boldsymbol{\theta}} \sum_{n\in\mathcal{S}_a} \log_2\left(1+\frac{\mathrm{SNR}_n}{\Gamma_n}\right) \tag{18}$$

$$\text{with } \mathrm{SNR}_n = \frac{\sigma^2_{n,\tilde{x}}}{\mathcal{E}\left\{\left|\tilde{e}^{\mathrm{uMMSE}}_{k,n}\right|^2\right\}} = \frac{\sigma^2_{n,\tilde{x}}}{\mathcal{E}\left\{\left|\tilde{d}^{\mathrm{uMMSE}}_n\tilde{\mathbf{y}}_{k,n}\mathbf{w} - \tilde{x}_{k,n}\right|^2\right\}} \tag{19}$$

$$\text{subject to } \tilde{d}^{\mathrm{uMMSE}}_n = \frac{\sigma^2_{n,\tilde{x}}}{\boldsymbol{\sigma}_{n,\tilde{x}\tilde{\mathbf{y}}}\mathbf{w}},\ \forall n\in\mathcal{S}_a \tag{20}$$

By turning the constrained optimization problem into a Lagrangian cost function, it is shown in [16] that the original (stochastic) BM-TEQ criterion (15) or its equivalent (18-20) lead to the following

(deterministic) **iteratively reweighted (IR) separable non-linear least-squares (SNLLS) based bitrate maximizing TEQ design criterion** (explained below):

$$\min_{\boldsymbol{\theta}} \frac{1}{K} \sum_{k=1}^{K} \left\| \operatorname{diag}\left(\sqrt{\boldsymbol{\gamma}_{K,\mathbf{w}_{\text{prev}}}}\right) \underbrace{\left(\operatorname{diag}(\tilde{\mathbf{d}})\tilde{\mathbf{Y}}_k\mathbf{w} - \tilde{\mathbf{x}}_k\right)}_{\tilde{\mathbf{e}}_k} \right\|^2 \tag{21}$$

The cost criterion (21) is clearly a *least-squares* (LS) criterion as it averages the squared norm of an error vector over K data points (i.e., K DMT symbols). It follows from (4) and (5) that the LS error vector $\tilde{\mathbf{e}}_k = \operatorname{diag}(\tilde{\mathbf{d}})\tilde{\mathbf{Y}}_k\mathbf{w} - \tilde{\mathbf{x}}_k$ is the k-th $N_a \times 1$ vector of FEQ output errors.

In addition, (21) is a so-called *iteratively reweighted* (IR) LS criterion. An IR-LS criterion forms a representation of a non-linear cost criterion that can be written as a weighted LS criterion where the weights $\boldsymbol{\gamma}_{K,\mathbf{w}_{\text{prev}}}$ depend on the LS errors $\tilde{\mathbf{e}}_k$, $k = 1, \ldots, K$, and hence on the optimization parameters $\boldsymbol{\theta}$ [23]. IR-LS criteria are typically optimized as an iterative sequence of weighted LS criteria with fixed weights in each iteration, computed with the parameter estimates from the previous iteration, $\boldsymbol{\theta}_{\text{prev}}$. According to [23], convergence occurs provided that the weights are bounded and non-increasing in the (absolute value) of the LS errors. For a non-convex cost function, the IR-LS algorithm leads to a local optimum. *For $K \to \infty$ and under the usual assumption of stationarity, the iterative sequence of (deterministic) weighted LS criteria (21) is equivalent to the original (stochastic) BM-TEQ criterion (15).*
The weights in (21) are determined by the $N_a \times 1$ weight vector $\boldsymbol{\gamma}_{K,\mathbf{w}}$, with as entry for tone n [16]:

$$\gamma_{K,n,\mathbf{w}} = \frac{(\text{SNR}_{K,n} + 1)^2}{\sigma^2_{K,n,\tilde{x}} \left(\text{SNR}_{K,n} + \Gamma_n\right)} \tag{22}$$

$$\text{with } \text{SNR}_{K,n} = \frac{\sigma^2_{K,n,\tilde{x}}}{\frac{1}{K}\sum_{k=1}^{K} |\tilde{e}_{k,n}|^2} = \frac{1}{\rho^{-2}_{K,n,\mathbf{w}} - 1} \tag{23}$$

$$\text{and } \rho^2_{K,n,\mathbf{w}} = \frac{\left|\boldsymbol{\sigma}_{K,n,\tilde{x}\tilde{\mathbf{y}}}\mathbf{w}\right|^2}{\sigma^2_{K,n,\tilde{x}} \left(\mathbf{w}^H \boldsymbol{\Sigma}^2_{K,n,\tilde{\mathbf{y}}}\mathbf{w}\right)} \tag{24}$$

The subscripts K indicate that $\gamma_{K,n,\mathbf{w}}$, $\text{SNR}_{K,n}$ and $\rho^2_{K,n,\mathbf{w}}$ are based on (deterministic) SOS *estimates* $\sigma^2_{K,n,\tilde{x}} = \frac{1}{K}\sum_{k=1}^{K} |\tilde{x}_{k,n}|^2$, $\boldsymbol{\sigma}_{K,n,\tilde{x}\tilde{\mathbf{y}}} = \frac{1}{K}\sum_{k=1}^{K} \tilde{x}^*_{k,n}\tilde{\mathbf{y}}_{k,n}$ and $\boldsymbol{\Sigma}^2_{K,n,\tilde{\mathbf{y}}} = \frac{1}{K}\sum_{k=1}^{K} \tilde{\mathbf{y}}^H_{k,n}\tilde{\mathbf{y}}_{k,n}$. Note that $\rho^2_{K,n,\mathbf{w}}$ is an estimate of the squared normalized correlation coefficient of the TX symbol $\tilde{x}_{k,n}$ and the DFT output $\tilde{y}_{k,n,\mathbf{w}}$, $\rho^2_{n,\mathbf{w}} = \frac{\left|\mathcal{E}\{\tilde{x}^*_{k,n}\tilde{y}_{k,n,\mathbf{w}}\}\right|^2}{\mathcal{E}\{|\tilde{x}_{k,n}|^2\}\mathcal{E}\{|\tilde{y}_{k,n,\mathbf{w}}|^2\}}$ [15].

Although the LS error vector $\tilde{\mathbf{e}}_k$ is non-linear in $\boldsymbol{\theta}$, the TEQ $\mathbf{w}$ and FEQs $\tilde{\mathbf{d}}$ appear linearly, which renders (21) a *separable non-linear least-squares (SNLLS)* criterion. As a consequence, the solution for $\tilde{\mathbf{d}}$ follows from considering a linear MMSE problem in $\tilde{\mathbf{d}}$, obtained by keeping $\mathbf{w}$ in (24) fixed. This results in an LS estimate of the biased MMSE (bMMSE) FEQs for the given $\mathbf{w}$:

$$\tilde{d}^{\text{bMMSE}}_n = \frac{\mathbf{w}^H \boldsymbol{\sigma}^H_{K,n,\tilde{x}\tilde{\mathbf{y}}}}{\mathbf{w}^H \boldsymbol{\Sigma}^2_{K,n,\tilde{\mathbf{y}}}\mathbf{w}} \tag{25}$$

which is independent of the weights $\boldsymbol{\gamma}_{K,\mathbf{w}_{\text{prev}}}$. The LS estimate of the uMMSE FEQs (13):

$$\tilde{d}_n^{\text{uMMSE}} = \frac{\sigma^2_{K,n,\tilde{x}}}{\boldsymbol{\sigma}_{K,n,\tilde{x}\tilde{\mathbf{y}}}\mathbf{w}} \tag{26}$$

which are the natural choice in DMT, can then be obtained from $\tilde{d}_n^{\text{bMMSE}}$ and $\mathbf{w}$ via $\rho^2_{K,n,\mathbf{w}}$ (24), for which the following holds true:

$$\rho^2_{K,n,\mathbf{w}} = \frac{\tilde{d}_n^{\text{bMMSE}}}{\tilde{d}_n^{\text{uMMSE}}} \tag{27}$$

Alternatively, (21) can be slightly modified using (27) so that the solution for the parameters $\tilde{\mathbf{d}}$ immediately provides an LS estimate of the uMMSE FEQs, rather than the bMMSE FEQs:

$$\boxed{\min_{\boldsymbol{\theta}} \frac{1}{K}\sum_{k=1}^{K} \left\| \text{diag}\left(\sqrt{\tilde{\boldsymbol{\gamma}}_{K,\mathbf{w}_{\text{prev}}}}\right) \underbrace{\left(\text{diag}(\tilde{\mathbf{d}})\tilde{\mathbf{Y}}_k\mathbf{w} - \boldsymbol{\rho}^{-2}_{K,\mathbf{w}_{\text{prev}}} \odot \tilde{\mathbf{x}}_k\right)}_{\check{\mathbf{e}}_k} \right\|^2} \tag{28}$$

This is also an **IR-SNLLS-based BM-TEQ design criterion** where $\check{\boldsymbol{\gamma}}_{K,\mathbf{w}_{\text{prev}}}$ and $\check{\mathbf{e}}_k$ are related to $\boldsymbol{\gamma}_{K,\mathbf{w}_{\text{prev}}}$ and $\tilde{\mathbf{e}}_k$ in (21), respectively, by:

$$\check{\gamma}_{K,n,\mathbf{w}_{\text{prev}}} = \left(\rho^2_{K,n,\mathbf{w}_{\text{prev}}}\right)^2 \gamma_{K,n,\mathbf{w}_{\text{prev}}} = \frac{\text{SNR}^2_{K,n}}{\sigma^2_{K,n,\tilde{x}}\left(\text{SNR}_{K,n}+\Gamma_n\right)} \tag{29}$$

$$\check{\mathbf{e}}_k = \boldsymbol{\rho}^{-2}_{K,\mathbf{w}_{\text{prev}}} \odot \tilde{\mathbf{e}}_k \tag{30}$$

The vector $\boldsymbol{\rho}^{-2}_{K,\mathbf{w}_{\text{prev}}}$ is short-hand notation for an $N_a \times 1$ vector with as entries the inverses of $\rho^2_{K,n,\mathbf{w}_{\text{prev}}}$ (24). The vector $\check{\mathbf{e}}_k$ (30) contains the errors between the uMMSE FEQ outputs and the virtual symbols $\left(\rho^2_{K,n,\mathbf{w}_{\text{prev}}}\right)^{-1}\tilde{x}_{k,n}$. Each error $\check{e}_{k,n}$ is a scaled version of the *biased* MMSE FEQ output error, i.e., $\tilde{e}_{k,n} = \rho^2_{K,n,\mathbf{w}_{\text{prev}}}\check{e}_{k,n}$. Note that for tones with a high SNR, $\rho^2_{K,n,\mathbf{w}_{\text{prev}}}$ is very close to 1.

According to [24], [25], SNLLS criteria are optimized iteratively. A naive algorithm could consist of alternately updating $\mathbf{w}$ and $\tilde{\mathbf{d}}$. However, this algorithm is cited to converge only linearly. The convergence speed becomes quadratic if the separability property is exploited in an appropriate way, e.g., by means of an iterative Gauss-Newton (GN) based or Levenberg-Marquardt (LM) based method. Two approaches may be pursued.

- By eliminating the parameters $\tilde{\mathbf{d}}$ in (28) using (26), a non-linear design criterion in $\mathbf{w}$ is obtained. The BM-TEQ and corresponding FEQs are then computed by alternating between a GN-based update of $\mathbf{w}$ to solve the non-linear design criterion in $\mathbf{w}$ and an update of $\tilde{\mathbf{d}}$ using (26). This approach has been taken in [16].
- Alternatively, an iterative GN-based or LM-based method can be applied to the original IR-SNLLS-based criterion in $\boldsymbol{\theta}$ (28), i.e., without explicit exploitation of (26). An *adaptive* algorithm is developed in Section IV, based on this approach.

Both alternatives have the same asymptotic convergence, but the former has the fastest initial convergence, hence also the fastest overall convergence, and is cited to be better conditioned [24], [25]. However, an adaptive algorithm that includes an updating of $\tilde{\mathbf{d}}$ based on (26), as in the former approach, requires the storage and updating of the $1 \times T$ crosscorrelation vectors $\boldsymbol{\sigma}_{K,n,\tilde{x}\tilde{\mathbf{y}}}$, $n \in \mathcal{S}_a$. As will be shown in Section IV, this can be avoided with the latter approach where the update of $\tilde{\mathbf{d}}$ is incorporated in the update of $\boldsymbol{\theta}$. Therefore, the second approach is better suited for the low-cost adaptive algorithm we are aiming for. To remedy the possible ill-conditioning of the second approach, we resort to an LM-type of algorithm. As is shown in the simulations of Section VI, the resulting adaptive algorithm has a sufficiently fast convergence, comparable to the adaptive PTEQ [20].

IV. ADAPTIVE BM-TEQ DESIGN METHOD

In Section IV-A, we show how the IR-SNLLS-based criterion in $\boldsymbol{\theta}$ can be optimized with an elegant, *iterative* Levenberg-Marquardt (LM) based algorithm that delivers the BM-TEQ and the corresponding FEQs. In Section IV-B, we introduce approximations and modifications to the iterative LM-based algorithm to turn it into a stable and efficient *adaptive* algorithm. In essence, the result is a *recursive Levenberg-Marquardt* (RLM) algorithm [22] that updates the joint TEQ-FEQ parameter vector $\boldsymbol{\theta}$ at each time instant k. In Section IV-C, it is shown that the RLM algorithm makes use of exactly the same SOS as the RLS-based adaptive PTEQ, based on the autocorrelation matrices of the sliding DFT of the RX signal. Similar to the RLS and RLS/LMS-based adaptive PTEQ, the SOS memory cost can then be further reduced by exploiting the sliding DFT structure. In Section IV-D, the RLM algorithm is further refined. The regularization parameter, which ensures stability, is adapted in an efficient way, based on the method presented in [26] that takes the changing condition of the Hessian over time into account.

A. Iterative LM-based algorithm

Based on the comments in Section III-B and on the reasoning in [24], the following 2-step iterative algorithm for solving the IR-SNLLS BM-TEQ criterion in $\boldsymbol{\theta}$ (28) can be proposed. The i-th iteration step consists of the computation of

1) the weights, $\check{\boldsymbol{\gamma}}_{K,\mathbf{w}_{\mathrm{prev}}}$, using (29), (23) and (24), with $\mathbf{w}_{\mathrm{prev}} = \mathbf{w}_{i-1}$;
2) a new estimate $\boldsymbol{\theta}_i$ as explained below.

The new estimate $\boldsymbol{\theta}_i$ is obtained from the previous estimate $\boldsymbol{\theta}_{i-1}$ based on the following LM update:

$$\boldsymbol{\theta}_i \leftarrow \boldsymbol{\theta}_{i-1} - \alpha_i \mathbf{R}^{-1}_{\boldsymbol{\theta}_{i-1},\delta_i} \mathbf{g}_{\boldsymbol{\theta}_{i-1}} \tag{31}$$

where $\boldsymbol{\theta}_{i-1} = \left[\begin{array}{cc} \mathbf{w}_{i-1}^H & \left(\tilde{\mathbf{d}}_{i-1}^{\mathrm{uMMSE}}\right)^H \end{array} \right]^H$, where $\mathbf{g}_{\boldsymbol{\theta}_{i-1}}$ and $\mathbf{R}_{\boldsymbol{\theta}_{i-1},\delta_i}$ are the gradient and a positive-definite regularized Hessian approximation of the IR-SNLLS-based BM-TEQ design criterion (28), eval-

uated in $\boldsymbol{\theta}_{i-1}$, and where α_i is a step size, obtained through a line search algorithm [24]. The gradient $\mathbf{g}_{\boldsymbol{\theta}}$ and regularized approximate Hessian $\mathbf{R}_{\boldsymbol{\theta},\delta}$ of (28) for a *complex-valued* TEQ are given by:

$$\mathbf{g}_{\boldsymbol{\theta}} = \begin{bmatrix} \mathbf{g}_{1,\boldsymbol{\theta}} \\ \mathbf{g}_{2,\boldsymbol{\theta}} \end{bmatrix} = \frac{1}{K}\sum_{k=1}^{K} \check{\mathbf{Y}}_{k,\boldsymbol{\theta}}^{H} \mathrm{diag}\left(\check{\boldsymbol{\gamma}}_{K,\mathbf{w}_{\mathrm{prev}}}\right) \check{\mathbf{e}}_k \tag{32}$$

$$\mathbf{R}_{\boldsymbol{\theta},\delta} = \frac{1}{K}\sum_{k=1}^{K} \check{\mathbf{Y}}_{k,\boldsymbol{\theta}}^{H} \mathrm{diag}\left(\check{\boldsymbol{\gamma}}_{K,\mathbf{w}_{\mathrm{prev}}}\right) \check{\mathbf{Y}}_{k,\boldsymbol{\theta}} + \delta \mathbf{I}_{T+N_a} \tag{33}$$

where

$$\check{\mathbf{Y}}_{k,\boldsymbol{\theta}} = \left[\begin{array}{c|c} \mathrm{diag}\left(\tilde{\mathbf{d}}\right)\tilde{\mathbf{Y}}_k & \mathrm{diag}(\tilde{\mathbf{Y}}_k\mathbf{w}) \end{array} \right] \tag{34}$$

where $\mathbf{g}_{1,\boldsymbol{\theta}}$ and $\mathbf{g}_{2,\boldsymbol{\theta}}$ are $T\times 1$ and $N_a \times 1$ subvectors of $\mathbf{g}_{\boldsymbol{\theta}}$, respectively, and where δ is a regularization parameter. The expressions for a real-valued TEQ are similar, but more cumbersome, because $\boldsymbol{\theta}$ then contains a real-valued $\mathbf{w}$ and a complex-valued $\tilde{\mathbf{d}}$. Based on (33), we define the following block description for the approximate Hessian:

$$\mathbf{R}_{\boldsymbol{\theta},\delta} = \begin{bmatrix} \mathbf{E} & \mathbf{F} \\ \mathbf{F}^H & \mathbf{G} \end{bmatrix} \tag{35}$$

with submatrices:

$$\begin{aligned} \mathbf{E} &= \frac{1}{K}\sum_{k=1}^{K} \tilde{\mathbf{Y}}_k^H \mathrm{diag}\left(\check{\boldsymbol{\gamma}}_{K,\mathbf{w}_{\mathrm{prev}}} \odot \left|\tilde{\mathbf{d}}\right|^2\right)\tilde{\mathbf{Y}}_k + \delta\mathbf{I}_T \\ &= \sum_{n\in\mathcal{S}_a} \check{\gamma}_{K,n,\mathbf{w}_{\mathrm{prev}}} \left|\tilde{d}_n\right|^2 \boldsymbol{\Sigma}^2_{K,n,\tilde{\mathbf{y}}} + \delta\mathbf{I}_T \end{aligned} \tag{36}$$

$$\begin{aligned} \mathbf{F}^H &= \frac{1}{K}\sum_{k=1}^{K} \mathrm{diag}\left(\tilde{\mathbf{Y}}_k\mathbf{w}\right)^* \mathrm{diag}\left(\check{\boldsymbol{\gamma}}_{K,\mathbf{w}_{\mathrm{prev}}} \odot \tilde{\mathbf{d}}\right)\tilde{\mathbf{Y}}_k \\ &= \left[\begin{array}{c} \vdots \\ \left[\check{\gamma}_{K,n,\mathbf{w}_{\mathrm{prev}}} \tilde{d}_n \left(\mathbf{w}^H \boldsymbol{\Sigma}^2_{K,n,\tilde{\mathbf{y}}}\right)\right] \\ \vdots \end{array} \right] \Big\updownarrow N_a \text{ rows} \end{aligned} \tag{37}$$

$$\begin{aligned} \mathbf{G} &= \frac{1}{K}\sum_{k=1}^{K} \mathrm{diag}\left(\check{\boldsymbol{\gamma}}_{K,\mathbf{w}_{\mathrm{prev}}} \odot \left|\tilde{\mathbf{Y}}_k\mathbf{w}\right|^2\right) + \delta\mathbf{I}_{N_a} \\ &= \mathrm{diag}\left(\left[\begin{array}{ccc} \cdots & \check{\gamma}_{K,n,\mathbf{w}_{\mathrm{prev}}} \left(\mathbf{w}^H \Sigma^2_{K,n,\tilde{\mathbf{y}}}\mathbf{w}\right) + \delta & \cdots \end{array} \right]\right) \end{aligned} \tag{38}$$

Each of the block matrices above has been written explicitly in terms of the $T \times T$ sliding DFT autocorrelation matrix estimates $\boldsymbol{\Sigma}^2_{K,n,\tilde{\mathbf{y}}}$, $n \in \mathcal{S}_a$ in (11). The SOS $\boldsymbol{\Sigma}^2_{K,n,\tilde{\mathbf{y}}}$ and the TEQ $\mathbf{w}$ are again

decoupled, based on (4). The approximate Hessian inverse is then given by:

$$\mathbf{R}_{\boldsymbol{\theta},\delta}^{-1} = \begin{bmatrix} \mathbf{Q}^{-1} & -\mathbf{Q}^{-1}\mathbf{F}\mathbf{G}^{-1} \\ -\mathbf{G}^{-1}\mathbf{F}^{H}\mathbf{Q}^{-1} & \mathbf{G}^{-1}\left(\mathbf{I}_{N_a} + \mathbf{F}^{H}\mathbf{Q}^{-1}\mathbf{F}\mathbf{G}^{-1}\right) \end{bmatrix} \tag{39}$$

with

$$\mathbf{Q} = \mathbf{E} - \mathbf{F}\mathbf{G}^{-1}\mathbf{F}^{H} \tag{40}$$

B. Basic RLM algorithm

In this section, several approximations and modifications are applied to the iterative LM method of the previous section, which then lead to an RLM-based adaptive BM-TEQ design method. *Iterative* GN and LM methods are classical and well-studied strategies for solving non-linear batch optimization problems. The *recursive* LM (RLM) algorithm developed here is inspired by earlier presented algorithms for adaptively solving non-linear problems in, e.g., recursive system identification [22], [27] and neural-network-based adaptive filter training [26], [28]. The RLM algorithm is a modification of the recursive GN algorithm, e.g., it regularizes the Hessian to remedy its ill-conditioning, which is often encountered when solving non-linear problems. We refer to [22] for a detailed derivation and analysis of the RLM algorithm.

The **RLM-based algorithm** to adapt the joint TEQ-FEQ parameter vector $\boldsymbol{\theta}$ at each time instant k is given by:

$$\boldsymbol{\theta}_k \leftarrow \boldsymbol{\theta}_{k-1} \underbrace{-\mathbf{R}_{k,\boldsymbol{\theta}_{k-1},\delta_k}^{-1}\mathbf{g}_{k,\boldsymbol{\theta}_{k-1}}}_{\Delta\boldsymbol{\theta}_k} \tag{41}$$

$$\tilde{\mathbf{d}}_k \leftarrow \tilde{\mathbf{d}}_k \cdot \|\mathbf{w}_k\| \tag{42}$$

$$\mathbf{w}_k \leftarrow \mathbf{w}_k / \|\mathbf{w}_k\| \tag{43}$$

with

$$\mathbf{g}_{k,\boldsymbol{\theta}} = \begin{bmatrix} \mathbf{g}_{k,1,\boldsymbol{\theta}} \\ \mathbf{g}_{k,2,\boldsymbol{\theta}} \end{bmatrix} = \check{\mathbf{Y}}_{k,\boldsymbol{\theta}}^{H}\mathrm{diag}\left(\hat{\boldsymbol{\gamma}}_{k,\mathbf{w}_{k-1}}\right) \underbrace{\left(\mathrm{diag}(\tilde{\mathbf{d}})\tilde{\mathbf{Y}}_k\mathbf{w} - \boldsymbol{\rho}_{k,\mathbf{w}_{k-1}}^{-2} \odot \tilde{\mathbf{x}}_k\right)}_{\tilde{\mathbf{e}}_k} \tag{44}$$

$$\mathbf{R}_{k,\boldsymbol{\theta},\delta_k} = \underbrace{\sum_{\kappa=1}^{k} \lambda^{k-\kappa}\check{\mathbf{Y}}_{\kappa,\boldsymbol{\theta}}^{H}\mathrm{diag}\left(\hat{\boldsymbol{\gamma}}_{k,\mathbf{w}_{k-1}}\right)\check{\mathbf{Y}}_{\kappa,\boldsymbol{\theta}}}_{\mathbf{R}_{k,\boldsymbol{\theta}}} + \delta_k\mathbf{I}_{T+N_a} \tag{45}$$

Compared to the iterative LM update (31), several approximations and modifications have been introduced. It will be illustrated in Section VI that, nevertheless, the adaptive algorithm converges sufficiently fast to a BM-TEQ with corresponding FEQs. The convergence speed turns out to be comparable to the convergence speed of the adaptive PTEQ [20].

- The weights $\check{\gamma}_{K,n,\mathbf{w}_{\text{prev}}}$ (29) assume knowledge of and depend on $\rho^2_{K,n,\mathbf{w}_{\text{prev}}}$ (24) via $\text{SNR}_{K,n}$ (23). This requires knowledge of $\mathbf{\Sigma}^2_{K,n,\tilde{\mathbf{y}}}$ and $\boldsymbol{\sigma}_{K,n,\tilde{x}\tilde{\mathbf{y}}}$, $n \in \mathcal{S}_a$, i.e., based on all data up to time K. Adopting the ideas from [27], [29], the optimal weights $\check{\gamma}_{K,n,\mathbf{w}}$ and $\text{SNR}_{K,n}$ are replaced by *instantaneous a priori* estimates at time k, i.e., based on the $\mathbf{w}_{k-1}$ and $\tilde{d}_{k-1,n}$:

$$\hat{\gamma}_{k,n,\mathbf{w}_{k-1}} = \frac{\widehat{\text{SNR}}^2_{k,n}}{\sigma^2_{k,n,\tilde{x}}\left(\widehat{\text{SNR}}_{k,n} + \Gamma_n\right)} \text{ with } \widehat{\text{SNR}}_{k,n} = \frac{\sigma^2_{k,n,\tilde{x}}}{\left|\tilde{d}_{k-1,n} - \tilde{\mathbf{y}}_{k,n}\mathbf{w}_{k-1} - \tilde{x}_{k,n}\right|^2} \tag{46}$$

  This instantaneous estimate $\widehat{\text{SNR}}_{k,n}$ is also used in (44) to approximate the virtual symbols in $\check{e}_{k,n}$: $\rho^{-2}_{K,n,\mathbf{w}_{\text{prev}}}\tilde{x}_{k,n} \approx \hat{\rho}^{-2}_{k,n,\mathbf{w}_{k-1}}\tilde{x}_{k,n} = \left(1 + \frac{1}{\widehat{\text{SNR}}_{k,n}}\right)\tilde{x}_{k,n}$[1]. This avoids the need for storing and updating the crosscorrelation vectors $\boldsymbol{\sigma}_{K,n,\tilde{x}\tilde{\mathbf{y}}}$, $n \in \mathcal{S}_a$.
- The gradient $\mathbf{g}_{\boldsymbol{\theta}}$ (32) is replaced by an *instantaneous* gradient estimate $\mathbf{g}_{k,\boldsymbol{\theta}}$ (44). This is common practice when developing adaptive algorithms and is mainly done for complexity reasons.
- The regularized approximate Hessian $\mathbf{R}_{\boldsymbol{\theta},\delta}$ (33) is replaced by an exponentially weighted estimate $\mathbf{R}_{k,\boldsymbol{\theta},\delta_k}$. As $\mathbf{R}_{\boldsymbol{\theta},\delta}$ depends on the SOS $\mathbf{\Sigma}^2_{n,\tilde{\mathbf{y}}}$, it suffices to update the SOS estimates $\mathbf{\Sigma}^2_{k,n,\tilde{\mathbf{y}}}$ using:

$$\mathbf{\Sigma}^2_{k,n,\tilde{\mathbf{y}}} = \lambda\mathbf{\Sigma}^2_{k-1,n,\tilde{\mathbf{y}}} + \tilde{\mathbf{y}}^H_{k,n}\tilde{\mathbf{y}}_{k,n} \tag{47}$$

  An exponential weighting factor $\lambda < 1$ allows to track the SOS in a non-stationary environment.
- For stability reasons, the approximate Hessian estimate $\mathbf{R}_{k,\boldsymbol{\theta}}$ is regularized by the second term in (45). An adaptation rule for the regularization parameter δ_k is given in Section IV-D taking into account the varying condition of $\mathbf{R}_{k,\boldsymbol{\theta}}$. The regularization also solves the inherent singularity of $\mathbf{R}_{k,\boldsymbol{\theta}}$: it can easily be shown that the vector $\begin{bmatrix} \mathbf{w}^H & -\tilde{\mathbf{d}}^H \end{bmatrix}^H$ lies in the right null space of $\check{\mathbf{Y}}_{k,\boldsymbol{\theta}}$ and, hence, of $\mathbf{R}_{k,\boldsymbol{\theta}}$. The inverse of $\mathbf{R}_{k,\boldsymbol{\theta},\delta_k}$ can be computed in a similar way as the inverse of $\mathbf{R}_{\boldsymbol{\theta},\delta}$ in (39).
- By imposing a normalized TEQ in (43) and scaling the FEQs accordingly in (42), the scalar parameter ambiguity of (28) is removed, namely that if $\boldsymbol{\theta} = \begin{bmatrix} \mathbf{w}^H & \tilde{\mathbf{d}}^H \end{bmatrix}^H$ is a solution of (28), then $\boldsymbol{\theta} = \begin{bmatrix} \eta\mathbf{w}^H & \eta^{-1}\tilde{\mathbf{d}}^H \end{bmatrix}^H$ is also a solution.

It is clear from (39) that the inversion of the regularized approximate Hessian $\mathbf{R}_{k,\boldsymbol{\theta},\delta_k}$ does not require $\mathcal{O}((T+N_a)^3)$ operations, thanks to its block structure $\mathbf{R}_{k,\boldsymbol{\theta},\delta_k} = \begin{bmatrix} \mathbf{E}_k & \mathbf{F}_k \\ \mathbf{F}^H_k & \mathbf{G}_k \end{bmatrix}$. The diagonal submatrix $\mathbf{G}_k$ is easily inverted; only the inverse of the full $T \times T$ matrix $\mathbf{Q}_k$ ($\mathcal{O}(T^3)$ operations) needs to be computed. Exploiting the block structure as in (39), the updating equation of $\boldsymbol{\theta}$ (41) can be split into an

[1]To avoid an overly pessimistic (i.e., large) a priori error $\check{e}_{k,n}$, we impose that $\hat{\rho}^2_{k,n,\mathbf{w}_{k-1}}$ is bounded between 0.5 and 1. This corresponds to a realistic lower bound on $\widehat{\text{SNR}}_{k,n}$ of 1 (or 0dB), as tones with a lower SNR can not reliably carry a single bit.

update of $\mathbf{w}$ and an update of $\tilde{\mathbf{d}}$. With the definitions of $\mathbf{Q}$ (40) and $\check{\mathbf{Y}}_{k,\boldsymbol{\theta}}$ (34), we obtain a GN-like update rule for $\mathbf{w}$:

$$\boxed{\mathbf{w}_k \leftarrow \mathbf{w}_{k-1} \underbrace{-\mathbf{Q}_k^{-1}\acute{\mathbf{g}}_k}_{\Delta\mathbf{w}_k}} \tag{48}$$

with

$$\acute{\mathbf{g}}_k = \sum_{n\in\mathcal{S}_a} \hat{\gamma}_{k,n,\mathbf{w}_{k-1}} \left(\tilde{\mathbf{y}}_{k,n}^H \tilde{d}_{k-1,n}^* - \frac{\left(\boldsymbol{\Sigma}_{k,n,\tilde{\mathbf{y}}}^2 \mathbf{w}_{k-1}\right) \tilde{d}_{k-1,n}^* \tilde{y}_{k,n,\mathbf{w}_{k-1}}^*}{\mathbf{w}_{k-1}^H \boldsymbol{\Sigma}_{k,n,\tilde{\mathbf{y}}}^2 \mathbf{w}_{k-1} + \delta_k} \right) \left(\tilde{d}_{k-1,n} \tilde{\mathbf{y}}_{k,n} \mathbf{w}_{k-1} - \frac{\tilde{x}_{k,n}}{\hat{\rho}_{k,n,\mathbf{w}_{k-1}}^2} \right) \tag{49}$$

$$\mathbf{Q}_k = \sum_{n\in\mathcal{S}_a} \hat{\gamma}_{k,n,\mathbf{w}_{k-1}} \left(\left|\tilde{d}_{k-1,n}\right|^2 \boldsymbol{\Sigma}_{k,n,\tilde{\mathbf{y}}}^2 - \frac{\left|\tilde{d}_{k-1,n}\right|^2 \boldsymbol{\Sigma}_{k,n,\tilde{\mathbf{y}}}^2 \mathbf{w}_{k-1} \mathbf{w}_{k-1}^H \boldsymbol{\Sigma}_{k,n,\tilde{\mathbf{y}}}^2}{\mathbf{w}_{k-1}^H \boldsymbol{\Sigma}_{k,n,\tilde{\mathbf{y}}}^2 \mathbf{w}_{k-1} + \delta_k} \right) + \delta_k \mathbf{I}_T \tag{50}$$

The n-th FEQ $\tilde{d}_n$ is updated as follows:

$$\boxed{\tilde{d}_{k,n} \leftarrow \tilde{d}_{k-1,n} \underbrace{- \left(\mathbf{w}_{k-1}^H \Sigma_{k,n,\tilde{\mathbf{y}}}^2 \mathbf{w}_{k-1} + \delta_k\right)^{-1} \left(\tilde{y}_{k,n,\mathbf{w}_{k-1}}^* \check{e}_{k,n} + \tilde{d}_{k-1,n} \left(\mathbf{w}_{k-1}^H \boldsymbol{\Sigma}_{k,n,\tilde{\mathbf{y}}}^2\right) \Delta\mathbf{w}_k \right)}_{\Delta\tilde{d}_{k,n}}} \tag{51}$$

where $\Delta\mathbf{w}_k$ has already been computed in (48). It will be shown in Section V that the computational cost of the equalizer update is dominated by $\mathcal{O}(N_a T^2)$ computations to construct $\mathbf{Q}_k$. One can show that for a real-valued TEQ (of course maintaining complex-valued FEQs), $\Re\{\acute{\mathbf{g}}_k\}$ and $\Re\{\mathbf{Q}_k\}$ are used in (48).

C. Reducing the SOS memory cost

Thanks to the block structure of the regularized approximate Hessian $\mathbf{R}_{k,\boldsymbol{\theta},\delta_k}$, which has size $(T + N_a) \times (T + N_a)$ (in the case of a complex-valued T-tap TEQ and N_a complex-valued 1-tap FEQs), the RLM algorithm only requires to store and update the N_a SOS estimates $\boldsymbol{\Sigma}_{k,n,\tilde{\mathbf{y}}}^2$ (size $T \times T$). Based on the sliding DFT shift structure, which is summarized by the second key observation (7), the memory cost can be further reduced from $N_a T^2$ complex-valued coefficients to $\mathcal{O}(N_a T)$ coefficients, as will be demonstrated.

It is instructive to mention that the same key observation (7) has led to the efficient RLS-based adaptive PTEQ. The PTEQ $\mathbf{w}_n$, which provides a separate equalizer for each tone n, minimizes the denominator of SNR_n (14), corresponding to the following linear MSE problem per tone [14]:

$$\min_{\mathbf{w}_n} \mathcal{E}\left\{ \left|\tilde{\mathbf{y}}_{k,n}\mathbf{w}_n - \tilde{x}_{k,n}\right|^2 \right\} \tag{52}$$

Here the PTEQ coefficients implicitly combine the FEQ and TEQ: $\mathbf{w}_n = \tilde{d}_n \mathbf{w}$. The solution of (52) is given by

$$\mathbf{w}_n = \left(\boldsymbol{\Sigma}_{n,\tilde{\mathbf{y}}}^2\right)^{-1} \boldsymbol{\sigma}_{n,\tilde{x}\tilde{\mathbf{y}}}^H \tag{53}$$

An RLS-based adaptive solution of (53) would then require recursive estimates of $\mathbf{\Sigma}^2_{n,\tilde{\mathbf{y}}}$, $n \in \mathcal{S}_a$. A computationally efficient solution with reduced SOS memory cost has been presented in [20]. Rather than recursively solving the linear PTEQ MSE problem (52), we then solve:

$$\min_{\mathbf{v}_n} \mathcal{E}\left\{ \left| \mathbf{z}_{k,n} \underbrace{\mathbf{T}_n \mathbf{w}_n}_{\mathbf{v}_n} - \tilde{x}_{k,n} \right|^2 \right\} \tag{54}$$

where we adopt the notation $\mathbf{z}_{k,n}$ and $\mathbf{T}_n$ from (7). The transformed PTEQ $\mathbf{v}_n$ linearly combines the modified PTEQ input vector $\mathbf{z}_{k,n} = \begin{bmatrix} \Delta\mathbf{y}_k^T & \tilde{y}_{k,n} \end{bmatrix}$, which consists of $T-1$ real-valued tone-independent difference terms (8) and one tone-dependent DFT output $\tilde{y}_{k,n}$ (9). The $T \times T$ autocorrelation matrix of $\mathbf{z}_{k,n}$ is given (in terms of submatrices) by

$$\mathbf{\Sigma}^2_{n,\mathbf{z}} = \mathcal{E}\left\{\mathbf{z}^H_{k,n}\mathbf{z}_{k,n}\right\} = \begin{bmatrix} \mathcal{E}\left\{\Delta\mathbf{y}_k \Delta\mathbf{y}_k^T\right\} & \mathcal{E}\left\{\tilde{y}_{k,n}\Delta\mathbf{y}_k\right\} \\ \mathcal{E}\left\{\tilde{y}^*_{k,n}\Delta\mathbf{y}_k^T\right\} & \mathcal{E}\left\{|\tilde{y}_{k,n}|^2\right\} \end{bmatrix} = \begin{bmatrix} \mathbf{\Sigma}^2_{\Delta\mathbf{y}} & \boldsymbol{\sigma}^H_{n,\tilde{y}\Delta\mathbf{y}} \\ \boldsymbol{\sigma}_{n,\tilde{y}\Delta\mathbf{y}} & \sigma^2_{n,\tilde{y}} \end{bmatrix} = \mathbf{U}^H_n \mathbf{U}_n \tag{55}$$

The square-root RLS PTEQ algorithm with inverse updating, presented in [20], stores and updates the inverse transpose $\mathbf{S}_{k,n}$ of the (upper triangular) Cholesky factor $\mathbf{U}_{k,n}$ of the SOS estimate $\mathbf{\Sigma}^2_{k,n,\mathbf{z}}$. It is easy to see from the block description that the first $T-1$ rows $\mathbf{U}_{k,\Delta\mathbf{y}}$ of $\mathbf{U}_{k,n}$, and, hence, the upper $T-1$ rows of $\mathbf{S}_{k,n} = \mathbf{U}^{-T}_{k,n}$ are real-valued, tone-independent and determined by $\mathbf{\Sigma}^2_{k,\Delta\mathbf{y}}$. The SOS memory cost is then reduced to $\frac{T(T-1)}{2}$ real-valued coefficients for storing $\mathbf{U}^{-T}_{k,\Delta\mathbf{y}}$ plus $2T-1$ real-valued coefficients per tone for the $1 \times T$ tone-dependent, complex-valued, last row of $\mathbf{S}_{k,n}$, i.e., an overall number of $(2T-1)N_a + \frac{T(T-1)}{2}$ real-valued coefficients. The SOS update cost is dominated by a term of $18N_aT$ real multiplications, mainly required for updating the tone-dependent last row of $\mathbf{S}_{k,n}$ [20].

In [21], it has been shown that a further complexity reduction is possible with a so-called hybrid RLS/LMS scheme. The idea is to ignore the two off-diagonal tone-dependent blocks of $\mathbf{\Sigma}^2_{n,\mathbf{z}}$ in (55):

$$\check{\mathbf{\Sigma}}^2_{n,\mathbf{z}} = \begin{bmatrix} \mathbf{\Sigma}^2_{\Delta\mathbf{y}} & \mathbf{0} \\ \mathbf{0} & \sigma^2_{n,\tilde{y}} \end{bmatrix} = \check{\mathbf{U}}^H_n \check{\mathbf{U}}_n \tag{56}$$

The first $T-1$ columns of the Cholesky factors $\check{\mathbf{U}}_n$ and $\mathbf{U}_n$ are the same and equal to the tone-independent $\mathbf{U}_{\Delta\mathbf{y}}$. In the last column of $\check{\mathbf{U}}_n$ only the last element is non-zero and equal to $\sqrt{\sigma^2_{n,\tilde{y}}}$. Similarly, the inverse transpose $\check{\mathbf{S}}_n = \check{\mathbf{U}}^{-T}_n$ has the same upper $T-1$ rows as $\mathbf{S}_n$ and a last tone-dependent row with one non-zero last element equal to $\sqrt{\sigma^{-2}_{n,\tilde{y}}}$. The tone-independent part $\mathbf{U}^{-T}_{k,\Delta\mathbf{y}}$ of $\check{\mathbf{S}}_{k,n}$ whitens the difference terms $\Delta\mathbf{y}_k$; the non-zero element $\sqrt{\sigma^{-2}_{k,n,\tilde{y}}}$ normalizes the tone-dependent DFT output $\tilde{y}_{k,n}$ in a similar way as the normalized LMS-algorithm. The hybrid RLS/LMS algorithm has an SOS memory cost of $N_a + \frac{T(T-1)}{2}$ real-valued coefficients and an SOS update cost that is dominated by $3T^2$ multiplications for the tone-independent part [21].

A similar approach can be taken for the SOS estimates of the adaptive BM-TEQ algorithm. Rather than storing and updating the inverse transpose Cholesky factor $\mathbf{S}_{k,n}$ as in the RLS PTEQ algorithm,

now the Cholesky factor itself $\mathbf{U}_{k,n}$ is updated and stored and then used to construct the inverse of the regularized approximate Hessian (45). The same SOS memory cost and a similar SOS updating cost as for the RLS-based adaptive PTEQ is obtained. Inspired by the hybrid RLS/LMS algorithm, the complexity can be further reduced by storing and updating $\check{\mathbf{U}}_{k,n}$ in (56). For the details on updating $\mathbf{U}_{k,n}$ and $\check{\mathbf{U}}_{k,n}$, e.g., based on a recursive QR update with Givens rotations, we refer to [30], [31].

As will be shown in Section V and in contrast to the adaptive PTEQ, the SOS updating is computationally not the most demanding part of the adaptive BM-TEQ. All PTEQ operations (the equalizer filtering as well as the SOS and equalizer updating) are based on the *transformed* inputs $\mathbf{z}_{k,n}$ and coefficients $\mathbf{v}_n$ in (54). The adaptive BM-TEQ, on the other hand, requires that the SOS estimates $\mathbf{\Sigma}^2_{k,n,\mathbf{z}}$ or $\check{\mathbf{\Sigma}}^2_{k,n,\mathbf{z}}$ are *back-transformed* after each update, based on:

$$\mathbf{\Sigma}^2_{n,\tilde{\mathbf{y}}} = \mathbf{T}_n^H \mathbf{\Sigma}^2_{n,\mathbf{z}} \mathbf{T}_n \text{ and } \check{\mathbf{\Sigma}}^2_{n,\tilde{\mathbf{y}}} = \mathbf{T}_n^H \check{\mathbf{\Sigma}}^2_{n,\mathbf{z}} \mathbf{T}_n \tag{57}$$

which follow from (7). Based on the definition of $\mathbf{T}_n$, a recursive computation can be derived as follows:

$$\begin{aligned}
\mathbf{\Sigma}^2_{n,\tilde{\mathbf{y}}}[s,t] &= \mathbf{\Sigma}^2_{n,\tilde{\mathbf{y}}}[s-1,t-1] + \alpha_n^* \mathbf{\Sigma}^2_{n,\tilde{\mathbf{y}}}[s-1,t] \\
&\quad + \alpha_n \mathbf{\Sigma}^2_{n,\tilde{\mathbf{y}}}[s,t-1] + \mathbf{\Sigma}^2_{n,\mathbf{z}}[T-s-1,T-t-1], \text{ if } 1 \le s \le T-1,\ 1 \le t \le T-1 \\
\mathbf{\Sigma}^2_{n,\tilde{\mathbf{y}}}[0,t] &= \alpha_n \mathbf{\Sigma}^2_{n,\tilde{\mathbf{y}}}[0,t-1] + \mathbf{\Sigma}^2_{n,\mathbf{z}}[T-1,T-t-1] \text{ if } 1 \le t \le T-1 \\
\mathbf{\Sigma}^2_{n,\tilde{\mathbf{y}}}[s,0] &= \alpha_n^* \mathbf{\Sigma}^2_{n,\tilde{\mathbf{y}}}[s-1,0] + \mathbf{\Sigma}^2_{n,\mathbf{z}}[T-s-1,T-1] \text{ if } 1 \le s \le T-1 \\
\mathbf{\Sigma}^2_{n,\tilde{\mathbf{y}}}[0,0] &= \mathbf{\Sigma}^2_{n,\mathbf{z}}[T-1,T-1]
\end{aligned} \tag{58}$$

where $\alpha_n = \exp(-j2\pi(n-1)/N)/\sqrt{N}$. This requires $\mathcal{O}(N_a T^2)$ operations. The matrices $\mathbf{\Sigma}^2_{k,n,\tilde{\mathbf{y}}}$ or $\check{\mathbf{\Sigma}}^2_{k,n,\tilde{\mathbf{y}}}$ are needed for computing the $T \times 1$ vectors $\mathbf{\Sigma}^2_{k,n,\tilde{\mathbf{y}}}\mathbf{w}$ ($\mathcal{O}(N_a T^2)$ operations) and scalars $\mathbf{w}^H \mathbf{\Sigma}^2_{k,n,\tilde{\mathbf{y}}}\mathbf{w}$ ($\mathcal{O}(N_a T)$ operations) in (49), (50) and (51). Note that the back-transformation (58) being unnecessary is an exclusive advantage of the adaptive PTEQ with *complex-valued* taps $\mathbf{v}_n$ over the adaptive (real-valued or complex-valued) BM-TEQ and BM-PGEQ and even the adaptive PTEQ with *real-valued* taps, which is basically a real BM-PGEQ with single-tone-groups: an adaptive BM-TEQ update requires $\mathcal{O}(N_a T^2)$ operations, while $\mathcal{O}(N_a T)$ operations suffice for the RLS-based adaptive PTEQ update.

D. Further refinements to the RLM algorithm

1) Exponential weighting factor λ: As suggested by (47), SOS like $\mathbf{\Sigma}^2_{n,\mathbf{z}}$ can be estimated recursively using an exponential window:

$$\mathbf{\Sigma}^2_{k,n,\mathbf{z}} = \lambda \mathbf{\Sigma}^2_{k-1,n,\mathbf{z}} + \mathbf{z}^H_{k,n} \mathbf{z}_{k,n} \tag{59}$$

An exponential window with factor λ corresponds to an equivalent window length of $\frac{1}{1-\lambda}$. When adaptively solving a *linear* problem, as with the RLS-based adaptive PTEQ, λ trades off tracking of time-variations versus accuracy of the SOS estimates and the adapted parameters: when λ approaches 1, the equivalent window is larger, hence the tracking performance is slower, but also the parameter estimates are more accurate, i.e., the excess MSE is smaller [30]. When adaptively solving a *non-linear* problem, an extra effect should be considered: when λ approaches 1, the adaptive algorithm converges slower [22]. During the first updates, the instantaneous gradient estimate (44) is inaccurate, due to poor initial parameter estimates. Hence, it is advised to depreciate these first estimates by using a smaller λ. In [22], λ is smoothly increased during convergence, e.g., from 0.95 to 0.99. In the simulations of Section VI, fast and accurate convergence is observed when starting with $\lambda = 0.9$ during the first 400 updates, switching to $\lambda = 0.95$ until update 800 and finishing with 200 updates at $\lambda = 0.99$.

2) Equalizer input power normalization: In contrast with the RLS algorithm, the stability of the RLM algorithm is not guaranteed by a nonsingular $\mathbf{R}_{k,\boldsymbol{\theta},\delta_k}$. The convergence speed and stability of the RLM method are to a large extent determined by the condition number or eigenvalue spread of the regularized approximate Hessian estimate (45). It is good practice to make sure that the eigenvalue spread only depends on the correlation among the equalizer inputs and not on the power spread of the equalizer inputs. In the case of a tapped delay-line equalizer/filter, the input autocorrelation matrix is Toeplitz, hence the diagonal elements are equal and so there is no input power spread. This does not necessarily hold true for the TEQ-FEQ-based receiver. We use the term "equalizer inputs" in a loose and general way here, as we are referring to the $N_a \times (N_a + T)$ matrix $\left[\mathrm{diag}\left(\sqrt{\hat{\boldsymbol{\gamma}}_{k,\mathbf{w}_{k-1}}}\right)\check{\mathbf{Y}}_{k,\boldsymbol{\theta}_{k-1}}\right]$, with $\check{\mathbf{Y}}_{k,\boldsymbol{\theta}}$ defined in (34), which appears in the gradient (44). Its autocorrelation matrix is estimated by the first term $\mathbf{R}_{k,\boldsymbol{\theta}}$ in (45). The diagonal of $\mathbf{R}_{k,\boldsymbol{\theta}}$, denoted with $\boldsymbol{\epsilon}_k$, is a measure for the "equalizer input powers" . The first T entries, corresponding to the diagonal elements of $\mathbf{E}_k$ (36), are almost equal to one another as they form the diagonal of a weighted sum of "almost-Toeplitz" matrices $\boldsymbol{\Sigma}^2_{k,n,\tilde{\mathbf{y}}}$. The last N_a entries, corresponding to the diagonal elements of $\mathbf{G}_k$ however can have a large dynamic range, depending on the frequency-dependent channel attenuation. Therefore, we normalize the input powers before each update by applying a diagonal transformation to the "equalizer input" and to the approximate Hessian in (45):

$$\left[\mathrm{diag}\left(\sqrt{\hat{\boldsymbol{\gamma}}_{k,\mathbf{w}_{k-1}}}\right)\check{\mathbf{Y}}_{k,\boldsymbol{\theta}_{k-1}}\right] \leftarrow \left[\mathrm{diag}\left(\sqrt{\hat{\boldsymbol{\gamma}}_{k,\mathbf{w}_{k-1}}}\right)\check{\mathbf{Y}}_{k,\boldsymbol{\theta}_{k-1}}\right]\mathrm{diag}\left(\sqrt{\boldsymbol{\epsilon}_k}\right)^{-1} \quad (60)$$

$$\mathbf{R}_{k,\boldsymbol{\theta}} \leftarrow \mathrm{diag}\left(\sqrt{\boldsymbol{\epsilon}_k}\right)^{-1}\mathbf{R}_{k,\boldsymbol{\theta}}\mathrm{diag}\left(\sqrt{\boldsymbol{\epsilon}_k}\right)^{-1} \quad (61)$$

The joint TEQ-FEQ vector $\boldsymbol{\theta}$ should be transformed accordingly with $\mathrm{diag}\left(\sqrt{\boldsymbol{\epsilon}_k}\right)$ before and $\mathrm{diag}\left(\sqrt{\boldsymbol{\epsilon}_k}\right)^{-1}$ after the update (41). The diagonal of $\mathbf{R}_{k,\boldsymbol{\theta}}$ (61) is then reduced to an identity matrix $\mathbf{I}_{T+N_a}$. In the next section, we consider the regularization of this transformed $\mathbf{R}_{k,\boldsymbol{\theta}}$ (61).

3) Adaptive regularization: Both the convergence speed and stability are affected by a suitable choice of the regularization parameter δ_k in (45): a too small δ_k could cause the RLM algorithm to run unstable, while a too large δ_k could induce slow convergence in directions in the parameter space that correspond to small eigenvalues of $\mathbf{R}_{k,\boldsymbol{\theta}_{k-1}}$. The parameter δ_k should be adapted, as the condition of $\mathbf{R}_{k,\boldsymbol{\theta}_{k-1}}$ depends on the estimates $\boldsymbol{\theta}_{k-1}$ and hence changes during convergence. There exists a vast literature on adaptive regularization for *iterative,* batch optimization methods where the condition changes while iterating. An *adaptive* algorithm, based on instantaneous estimates of the actual and predicted cost criterion reduction, is proposed in [26]. We adapt this algorithm for use with the adaptive BM-TEQ as follows.

For each update, the RLM algorithm (41) optimizes a local, instantaneous, quadratic approximation $J_{k,\mathrm{approx}}(\boldsymbol{\theta})$ of the IR-SNLLS-based design criterion $J_{k,\mathrm{IR-SNLLS}}(\boldsymbol{\theta})$ in (28):

$$J_{k,\mathrm{approx}}(\boldsymbol{\theta}) = J_{k,\mathrm{IR-SNLLS}}(\boldsymbol{\theta}_{k-1}) + \Delta\boldsymbol{\theta}^H \mathbf{g}_{k,\boldsymbol{\theta}_{k-1}} + \Delta\boldsymbol{\theta}^T \mathbf{g}^*_{k,\boldsymbol{\theta}_{k-1}} + \Delta\boldsymbol{\theta}^H \mathbf{R}_{k,\boldsymbol{\theta}_{k-1},\delta_k} \Delta\boldsymbol{\theta} \tag{62}$$

with $\Delta\boldsymbol{\theta} = \boldsymbol{\theta} - \boldsymbol{\theta}_{k-1}$, $\mathbf{g}_{k,\boldsymbol{\theta}_{k-1}}$ and $\mathbf{R}_{k,\boldsymbol{\theta}_{k-1},\delta_k}$ defined in (41-45). Minimizing $J_{k,\mathrm{approx}}(\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$ indeed gives the RLM update (41). The *predicted* instantaneous cost reduction estimate is obtained by substituting $\Delta\boldsymbol{\theta}_k$ (41) in (62):

$$\begin{aligned}
\varepsilon_{k,\mathrm{p}} &= J_{k,\mathrm{IR-SNLLS}}(\boldsymbol{\theta}_{k-1}) - J_{k,\mathrm{approx}}(\boldsymbol{\theta}_k) \\
&= \mathbf{g}^H_{k,\boldsymbol{\theta}_{k-1}} \mathbf{R}^{-1}_{k,\boldsymbol{\theta}_{k-1},\delta_k} \mathbf{g}_{k,\boldsymbol{\theta}_{k-1}} &(63)\\
&= -\mathbf{g}^H_{k,\boldsymbol{\theta}_{k-1}} \Delta\boldsymbol{\theta}_k &(64)
\end{aligned}$$

The predicted cost reduction estimate $\varepsilon_{k,\mathrm{p}}$ is cheaply computed with (64). Note that in the case of a real-valued TEQ, (64) is given by $\varepsilon_{k,\mathrm{p}} = -\mathbf{g}^T_{k,1,\boldsymbol{\theta}_{k-1}} \Delta\mathbf{w}_k - 2\Re\{\mathbf{g}^H_{k,2,\boldsymbol{\theta}_{k-1}} \Delta\tilde{\mathbf{d}}_k\}$, where $\mathbf{g}_{k,1,\boldsymbol{\theta}}$ and $\mathbf{g}_{k,2,\boldsymbol{\theta}}$ has been defined in (44) and where $\Delta\mathbf{w}_k$ and $\Delta\tilde{\mathbf{d}}_k$ have been defined in (48) and (51), respectively.

The *actual* instantaneous cost reduction estimate is determined by the a priori error vector $\check{\mathbf{e}}_{k,\boldsymbol{\theta}_{k-1}}$ and a posteriori error vector $\check{\mathbf{e}}_{k,\boldsymbol{\theta}_k}$:

$$\varepsilon_{k,\mathrm{a}} = \left\| \mathrm{diag}\left(\sqrt{\hat{\boldsymbol{\gamma}}_{k,\mathbf{w}_{k-1}}}\right) \check{\mathbf{e}}_{k,\boldsymbol{\theta}_{k-1}} \right\|^2 - \left\| \mathrm{diag}\left(\sqrt{\hat{\boldsymbol{\gamma}}_{k,\mathbf{w}_{k-1}}}\right) \check{\mathbf{e}}_{k,\boldsymbol{\theta}_k} \right\|^2 \tag{65}$$

The a posteriori error vector is efficiently computed from the a priori error vector as:

$$\begin{aligned}
\check{\mathbf{e}}_{k,\boldsymbol{\theta}_k} &= \mathrm{diag}\left(\tilde{\mathbf{d}}_k\right) \tilde{\mathbf{Y}}_k \mathbf{w}_k - \boldsymbol{\rho}^{-2}_{k,\mathbf{w}_{k-1}} \odot \tilde{\mathbf{x}}_k &(66)\\
&= \mathrm{diag}\left(\Delta\tilde{\mathbf{d}}_k\right) \left(\tilde{\mathbf{Y}}_k \mathbf{w}_{k-1} + \tilde{\mathbf{Y}}_k \Delta\mathbf{w}_k\right) + \mathrm{diag}\left(\tilde{\mathbf{d}}_{k-1}\right) \tilde{\mathbf{Y}}_k \Delta\mathbf{w}_k \\
&\quad + \underbrace{\mathrm{diag}\left(\tilde{\mathbf{d}}_{k-1}\right) \tilde{\mathbf{Y}}_k \mathbf{w}_{k-1} - \boldsymbol{\rho}^{-2}_{k,\mathbf{w}_{k-1}} \odot \tilde{\mathbf{x}}_k}_{\check{\mathbf{e}}_{k,\,\boldsymbol{\theta}_{k-1}}} &(67)
\end{aligned}$$

where $\tilde{\mathbf{Y}}_k \Delta\mathbf{w}_k$ in the first two terms of (67) dominates the computation cost ($4N_aT$ multiplications).

Both the actual and predicted instantaneous cost reduction estimates are averaged using an exponential window with $\lambda_\varepsilon = 0.9$: $\bar{\varepsilon}_{k,\mathrm{a}} = \sum_{\kappa=1}^{k} \lambda_\varepsilon^{k-\kappa} \varepsilon_{\kappa,\mathrm{a}}$ and $\bar{\varepsilon}_{k,\mathrm{p}} = \sum_{\kappa=1}^{k} \lambda_\varepsilon^{k-\kappa} \varepsilon_{\kappa,\mathrm{p}}$. The following adaptation rule is then applied, once per 10 BM-TEQ updates, based on the ratio $r_k = \frac{\bar{\varepsilon}_{k,\mathrm{a}}}{\bar{\varepsilon}_{k,\mathrm{p}}}$, which is a measure for the match between the local quadratic approximation (62) and the actual design criterion (28) [26]:

$$\delta_{k+1} \leftarrow \begin{cases} 2\delta_k & \text{if } r_k < 0.25 \\ \max(\delta_k/2, \delta_{\min}) & \text{if } r_k > 0.75 \\ \delta_k & \text{otherwise} \end{cases} \tag{68}$$

A lower bound of, e.g., $\delta_{\min} = 10^{-12}$, prevents a numerically singular $\mathbf{R}_{k,\boldsymbol{\theta}_{k-1},\delta_k}$. If $\bar{\varepsilon}_{k,\mathrm{a}}$ happens to drop below zero (due to the stochastic nature of the incoming data), the update (41) is ignored. In addition, if this occurs twice in a row, the regularization is increased: $\delta_{k+1} \leftarrow 2\delta_k$. The larger the confidence in the initial parameter estimate, the smaller the initial δ_0 should be chosen. This adaptive regularization algorithm introduces a minor extra computational cost and a memory cost of 2 real-valued coefficients $\bar{\varepsilon}_{k,\mathrm{a}}$ and $\bar{\varepsilon}_{k,\mathrm{p}}$.

E. Summary of the adaptive BM-TEQ and BM-PGEQ algorithm

The adaptive BM-TEQ algorithm (for the case of RLS-based SOS and a complex-valued TEQ) is summarized in **Table I**. The presented algorithm is easily extended to an adaptive BM-PGEQ by solving several downscaled adaptive BM-TEQ problems in parallel. The steps 1 to 5 are then identical, as they are independent of the use of tone groups. Step 6 to 8 need to be executed for each tone group. A complete range of *adaptive BM equalizers* then opens up: from a single (real-valued or complex-valued) T-tap adaptive TEQ over an intermediate $N_g T$-tap adaptive PGEQ to a complex-valued $N_a T$-tap adaptive PTEQ. All designs have the same SOS memory cost, but a varying equalizer design cost. As explained in Section IV-C and detailed in Section V, the adaptive BM-TEQ and BM-PGEQ require $\mathcal{O}(N_a T^2)$ computations per update, i.e., roughly T times more than an adaptive PTEQ update. The adaptive BM-TEQ and BM-PGEQ have roughly the same complexity, see Section V.

V. MEMORY COST AND COMPUTATIONAL COMPLEXITY OF THE ADAPTIVE BM EQUALIZERS

In this section, the overall memory cost and computational complexity of the adaptive BM equalizers is summarized. A clear distinction is made between memory and computational cost related to *equalizer "filtering"*[2], i.e., computing the equalizer output, versus *equalizer updating*, i.e. adapting the equalizer coefficients.

[2]The term "filtering" is used in a loose sense here: e.g., the PTEQ is a *linear combiner*, rather than a *filter*.

TABLE I

ADAPTIVE BM-TEQ ALGORITHM.

Initialize $\mathbf{w}$, $\tilde{\mathbf{d}}$, $\bar{\varepsilon}_{k,\mathrm{a}}$ and $\bar{\varepsilon}_{k,\mathrm{p}}$ as well as $\mathbf{U}_{k,n}$ for each tone $n \in \mathcal{S}_a$.
For $k = 1, \ldots, \infty$

- *Filtering*:

  1. Compute the sliding DFT output, $\tilde{\mathbf{y}}_{k,n}$, for each tone $n \in \mathcal{S}_a$:
  $$\tilde{\mathbf{y}}_{k,n}[0] = \mathcal{F}_n \mathbf{y}_{k,0:N-1} \text{ and } \tilde{\mathbf{y}}_{k,n}[t] = \alpha_n \tilde{\mathbf{y}}_{k,n}[t-1] + (y_{k,-t} - y_{k,N-t}),\ t = 1, \cdots, T-1 \tag{69}$$

  2. Compute the FEQ output $\hat{\tilde{x}}_{k,n}$, for each tone $n \in \mathcal{S}_a$:
  $$\hat{\tilde{x}}_{k,n} = \tilde{d}_{k-1,n} \tilde{y}_{k,n,\mathbf{w}_{k-1}} \text{ with } \tilde{y}_{k,n,\mathbf{w}_{k-1}} = \tilde{\mathbf{y}}_{k,n} \mathbf{w}_{k-1} \tag{70}$$

- *Updating*:

  1. Compute the FEQ output error $\tilde{e}_{k,n}$, for each tone $n \in \mathcal{S}_a$:
  $$\tilde{e}_{k,n} = \hat{\tilde{x}}_{k,n} - \tilde{x}_{k,n} \tag{71}$$

  2. Compute $\widehat{\mathrm{SNR}}_{k,n}$ and $\hat{\gamma}_{k,n,\mathbf{w}_{k-1}}$, for each tone $n \in \mathcal{S}_a$:
  $$\hat{\gamma}_{k,n,\mathbf{w}_{k-1}} = \frac{\widehat{\mathrm{SNR}}_{k,n}^2}{\sigma_{k,n,\tilde{x}}^2 \left(\widehat{\mathrm{SNR}}_{k,n} + \Gamma_n \right)} \text{ with } \widehat{\mathrm{SNR}}_{k,n} = \frac{\sigma_{k,n,\tilde{x}}^2}{\left| \tilde{e}_{k,n} \right|^2} \tag{72}$$

  3. Compute the "virtual error" $\check{e}_{k,n}$, for each tone $n \in \mathcal{S}_a$:
  $$\check{e}_{k,n} = \tilde{e}_{k,n} - \frac{\tilde{x}_{k,n}}{\max\left(\widehat{\mathrm{SNR}}_{k,n}, 1 \right)} \tag{73}$$

  4. Update the SOS $\mathbf{\Sigma}_{k,n,\tilde{\mathbf{y}}}^2$, for each tone $n \in \mathcal{S}_a$:

  1) Use Givens rotations [30], [31] to update the Cholesky factor $\mathbf{U}_{k,n}$ (i.e., $\mathbf{U}_{k,\Delta\mathbf{y}}$, which is common for all tones, as well as the last column $\mathbf{u}_{k,n}$ of $\mathbf{U}_{k,n}$ for each tone $n \in \mathcal{S}_a$) with $\mathbf{z}_{k,n} = \begin{bmatrix} \Delta\mathbf{y}_k^T & \tilde{y}_{k,n} \end{bmatrix}$:
  $$\mathbf{U}_{k,n} = \mathrm{givens}(\mathbf{U}_{k-1,n}, \lambda, \mathbf{z}_{k,n}) \tag{74}$$

  2) Compute $\mathbf{\Sigma}_{k,n,\mathbf{z}}^2$, i.e., $\mathbf{\Sigma}_{k,\Delta\mathbf{y}}^2$ as well as $\boldsymbol{\sigma}_{k,n,\tilde{y}\Delta\mathbf{y}}$ and $\sigma_{k,n,\tilde{y}}^2$ for each tone $n \in \mathcal{S}_a$:
  $$\mathbf{\Sigma}_{k,n,\Delta\mathbf{y}}^2 = \mathbf{U}_{k,\Delta\mathbf{y}}^H \mathbf{U}_{k,\Delta\mathbf{y}},\ \boldsymbol{\sigma}_{k,n,\tilde{y}\Delta\mathbf{y}} = \mathbf{u}_{k,n}^H \mathbf{U}_{k,\Delta\mathbf{y}} \text{ and } \sigma_{k,n,\tilde{y}}^2 = \mathbf{u}_{k,n}^H \mathbf{u}_{k,n} \tag{75}$$

  3) Compute $\mathbf{\Sigma}_{k,n,\tilde{\mathbf{y}}}^2$ recursively, for each tone $n \in \mathcal{S}_a$:
  $$\begin{aligned} \mathbf{\Sigma}_{k,n,\tilde{\mathbf{y}}}^2[s,t] &= \mathbf{\Sigma}_{k,n,\tilde{\mathbf{y}}}^2[s-1,t-1] + \alpha_n^* \mathbf{\Sigma}_{k,n,\tilde{\mathbf{y}}}^2[s-1,t] \\ &\quad + \alpha_n \mathbf{\Sigma}_{k,n,\tilde{\mathbf{y}}}^2[s,t-1] + \mathbf{\Sigma}_{k,n,\mathbf{z}}^2[T-s-1,T-t-1],\ 0 \le s \le T-1,\ 0 \le t \le T-1 \end{aligned} \tag{76}$$

  5. Compute for each tone $n \in \mathcal{S}_a$:
  $$\mathbf{b}_{k,n} = \mathbf{w}_{k-1}^H \mathbf{\Sigma}_{k,n,\tilde{\mathbf{y}}}^2 \text{ and } b_{k,n} = \mathbf{w}_{k-1}^H \left(\mathbf{\Sigma}_{k,n,\tilde{\mathbf{y}}}^2 \mathbf{w}_{k-1} \right) \tag{77}$$

  6. Compute the update $\Delta\mathbf{w}_k$ (the equalizer input power normalization is omitted, see Section IV-D.2):
  $$\mathbf{w}_k \leftarrow \mathbf{w}_{k-1} \underbrace{-\mathbf{Q}_k^{-1} \hat{\mathbf{g}}_k}_{\Delta\mathbf{w}_k} \tag{78}$$

  with
  $$\mathbf{g}_{k,1} = \sum_{n \in \mathcal{S}_a} \hat{\gamma}_{k,n,\mathbf{w}_{k-1}} \tilde{\mathbf{y}}_{k,n}^H \tilde{d}_{k-1,n}^* \check{e}_{k,n} \text{ and } g_{k,n,2} = \hat{\gamma}_{k,n,\mathbf{w}_{k-1}} \tilde{d}_{k-1,n}^* \tilde{y}_{k,n,\mathbf{w}_{k-1}}^* \check{e}_{k,n} \tag{79}$$
  $$\hat{\mathbf{g}}_k = \mathbf{g}_{k,1} - \sum_{n \in \mathcal{S}_a} \frac{\mathbf{b}_{k,n}^H g_{k,n,2}}{b_{k,n} + \delta_k} \tag{80}$$
  $$\mathbf{Q}_k = \sum_{n \in \mathcal{S}_a} \hat{\gamma}_{k,n,\mathbf{w}_{k-1}} \left(\left| \tilde{d}_{k-1,n} \right|^2 \mathbf{\Sigma}_{k,n,\tilde{\mathbf{y}}}^2 - \frac{\left| \tilde{d}_{k-1,n} \right|^2 \mathbf{b}_{k,n}^H \mathbf{b}_{k,n}}{b_{k,n} + \delta_k} \right) + \delta_k \mathbf{I}_T \tag{81}$$

  7. Compute the update $\Delta\tilde{d}_{k,n}$, for each tone $n \in \mathcal{S}_a$:
  $$\tilde{d}_{k,n} \leftarrow \tilde{d}_{k-1,n} \underbrace{- \left(b_{k,n} + \delta_k \right)^{-1} \left(\tilde{y}_{k,n,\mathbf{w}_{k-1}}^* \check{e}_{k,n} + \tilde{d}_{k-1,n} \mathbf{b}_n \Delta\mathbf{w}_k \right)}_{\Delta\tilde{d}_{k,n}} \tag{82}$$

TABLE I
ADAPTIVE BM-TEQ ALGORITHM *(continued)*

8. Update the regularization parameter δ_{k+1}:
   1) Compute the predicted cost reduction estimate $\varepsilon_{k,\mathrm{p}}$:
   $$\varepsilon_{k,\mathrm{p}} = -\mathbf{g}_{k,1}^H \Delta\mathbf{w}_k - \sum_{n\in\mathcal{S}_a} g_{k,n,2}^* \Delta\tilde{d}_{k,n} \tag{83}$$
   2) Compute the actual cost reduction estimate $\varepsilon_{k,\mathrm{a}}$:
   $$\varepsilon_{k,\mathrm{a}} = \sum_{n\in\mathcal{S}_a} \hat{\gamma}_{k,n,\mathbf{w}_{k-1}} \left(|\check{e}_{k,n}|^2 - |\check{e}_{k,n,\boldsymbol{\theta}_k}|^2 \right) \tag{84}$$
   with
   $$\check{e}_{k,n,\boldsymbol{\theta}_k} = \Delta\tilde{d}_{k,n} \left(\tilde{y}_{k,n,\mathbf{w}_{k-1}} + \Delta\tilde{y}_{k,n,\mathbf{w}} \right) + \tilde{d}_{k-1,n} \Delta\tilde{y}_{k,n,\mathbf{w}} + \check{e}_{k,n} \text{ with } \Delta\tilde{y}_{k,n,\mathbf{w}} = \tilde{\mathbf{y}}_{k,n} \Delta\mathbf{w}_k \tag{85}$$
   3) Update $\bar{\varepsilon}_{k,\mathrm{a}} = \lambda_\varepsilon \bar{\varepsilon}_{k-1,\mathrm{a}} + \varepsilon_{k,\mathrm{a}}$ and $\bar{\varepsilon}_{k,\mathrm{p}} = \lambda_\varepsilon \bar{\varepsilon}_{k-1,\mathrm{p}} + \varepsilon_{k,\mathrm{p}}$.
   4) If $\bar{\varepsilon}_{k,\mathrm{a}} < 0$, then $\Delta\mathbf{w}_k \leftarrow \mathbf{0}$ and $\Delta\tilde{d}_{k,n} \leftarrow 0$. If $\bar{\varepsilon}_{k-1,\mathrm{a}} < 0$ and $\bar{\varepsilon}_{k,\mathrm{a}} < 0$, then $\delta_{k+1} \leftarrow 2\delta_k$.
   5) If $k = 10, 20, 30, \ldots$, then compute $r_k = \frac{\bar{\varepsilon}_{k,\mathrm{a}}}{\bar{\varepsilon}_{k,\mathrm{p}}}$ and update δ_{k+1}:
   $$\delta_{k+1} \leftarrow \begin{cases} 2\delta_k & \text{if } r_k < 0.25 \\ \max(\delta_k/2, \delta_{\min}) & \text{if } r_k > 0.75 \\ \delta_k & \text{otherwise} \end{cases} \tag{86}$$

End

The computational complexity of the adaptive BM-TEQ algorithm of Table I is summarized in **Table II**. We distinguish between a real-valued and a complex-valued TEQ as well as between RLS-based and RLS/LMS-based SOS. As in [14], the number of real multiplications is given; a division is counted as a multiplication and a multiplication of 2 complex-valued numbers counts as 4 real multiplications. Some remarks are as follows:

- In step 1, we choose to compute $\hat{\tilde{\mathbf{x}}}_k$ based on a linear combination $\mathbf{w}$ of the T sliding DFT outputs $\tilde{\mathbf{Y}}_k$, rather than the classical convolution with the TEQ (see key observations (4) and (7)), because $\tilde{\mathbf{Y}}_k$ is also needed in the update of $\mathbf{w}$ (step 6) and the regularization update (step 8).
- Symmetric or Hermitian matrices are involved in steps 4 and 6, so roughly only half of the elements of these matrices must be computed.
- The inverse of the symmetric positive definite matrix $\mathbf{Q}_k$ in step 6 can be computed via its Cholesky decomposition [31].

The complexity is dominated by the cost of steps 4 and 5, more specifically the computation of the SOS estimates $\boldsymbol{\Sigma}_{k,n,\tilde{\mathbf{y}}}^2$ or $\check{\boldsymbol{\Sigma}}_{k,n,\tilde{\mathbf{y}}}^2$ and $\mathbf{Q}_k$, namely $8.5N_aT^2$ multiplications for a real-valued TEQ and $12N_aT^2$ multiplications for a complex-valued TEQ. The computational advantage of using the RLS/LMS-based SOS of (56) is minor: only around N_aT^2 computations are saved. The cost of the matrix inversion $\mathbf{Q}_k^{-1}$ ($\mathcal{O}(T^3)$ multiplications) in step 6 is considerably smaller than the $\mathcal{O}(N_aT^2)$ terms in steps 4 and 5 and some $\mathcal{O}(N_aT)$ terms for typical values of N_a and T.

For the adaptive BM-PGEQ, all dominant cost terms still hold true, except for the inversion of $\mathbf{Q}_k$:

the inversion is computed for each of the N_g tone groups, causing a dominant cost term $\frac{N_g T^3}{6}$. Still, the $\mathcal{O}(N_a T^2)$ terms dominate the overall cost of the algorithm for typical values of N_a, N_g and T: e.g., for $N_a = 224$, $T = 32$ and real-valued PGEQs, it always holds that $8.5 N_a T^2 > \frac{N_g T^3}{6}$, independent of N_g.

Table III compares the dominant terms in the memory cost and computational complexity of equalizer filtering and updating for the adaptive BM-TEQ, BM-PGEQ and PTEQ. Memory and complexity figure estimates are included for $N_a = 224$, $T = 16 \rightarrow 32$ and $N_g = 4$.

- The computational cost of *filtering* is roughly the same for all presented adaptive BM equalizers: $\mathcal{O}(N_a T)$ + 1 DFT.
- However, there is a large difference in memory cost: from $N_a T$ complex-valued taps for the PTEQ, over $N_g T$ real-valued or complex-valued taps plus N_a complex-valued FEQ taps (with typically $N_g \ll N_a$) for the BM-PGEQ, towards only T real-valued or complex-valued TEQ taps plus N_a complex-valued FEQ taps for a single BM-TEQ.
- All adaptive BM equalizers have the same SOS memory cost for equalizer *updating*, dominated by $\mathbf{U}_{k,n}$ or $\mathbf{S}_{k,n}$ (see Section IV-C). The adaptive BM-TEQ stores two extra coefficients $\bar{\varepsilon}_{k,\mathrm{a}}$ and $\bar{\varepsilon}_{k,\mathrm{p}}$ for the adaptive regularization (see Section IV-D.3). The adaptive BM-PGEQ stores these elements for each tone group. This extra memory cost is negligible.
- The adaptive BM-TEQ and BM-PGEQ have roughly the same *updating* complexity. The PTEQ updating appears to be 8 to 16 times less complex than the BM-TEQ/BM-PGEQ updating. As already pointed out in Section IV-C, the PTEQ has the exclusive advantage that its SOS computations are based on the transformed inputs $\mathbf{z}_{k,n}$ and equalizer taps $\mathbf{v}_n$ (see (54)) and does not require the back-transformation (58).

Assuming a typical number of 4000 DMT symbols per second, the RLS-based adaptive PTEQ updating requires 272×10^6 to 536×10^6 real multiplications per second (rmps), while the BM-TEQ updating requires 2280×10^6 to 8400×10^6 rmps. However, in an application such as ADSL, equalizer design and tracking can typically be done at a rate that is much lower than the equalizer filtering rate of 4 kHz: assuming that 16000 DMT training symbols are available during connection set-up while convergence occurs within 200 to 300 symbols (see the simulations in Section VI), then the updating rate can be decimated by a factor 50 to 80, resulting in a complexity of 3.4×10^6 to 10.7×10^6 rmps for the adaptive PTEQ and 28.5×10^6 to 168×10^6 rmps for the adaptive BM-TEQ.

VI. Simulations

In this section, we provide extensive simulation results for various typical ADSL scenarios and parameter settings. They confirm that the adaptive BM-TEQ converges to a (locally optimal) TEQ with a similar bitrate as the iterative GN-based BM-TEQ of [16]. A similar conclusion is drawn for the adaptive

TABLE II

COMPUTATIONAL COST IN REAL MULTIPLICATIONS OF THE ADAPTIVE BM-TEQ ALGORITHM FOR A REAL-VALUED AND A COMPLEX-VALUED TEQ AS WELL AS FOR RLS-BASED AND RLS/LMS-BASED SOS. N_a IS THE NUMBER OF ACTIVE TONES AND T IS THE NUMBER OF TEQ TAPS.

Step	Equations	Real-valued TEQ	Complex-valued TEQ
Filtering: (69)-(70)		$6N_aT+1$ DFT	$8N_aT+1$ DFT
Updating:			
1	(71)	0	0
2	(72)	$6N_a$	$6N_a$
3	(73)	$2N_a$	$2N_a$
4	1) (74) $\mathbf{U}_{k,\Delta\mathbf{y}}$	$\frac{5T^2}{2}+\frac{5T}{2}-5$	$\frac{5T^2}{2}+\frac{5T}{2}-5$
	1) (74) $\mathbf{u}_{k,n}$ (RLS)	$N_a(10T+8)$	$N_a(10T+8)$
	1) (74) $\sqrt{\sigma^{-2}_{k,n,\tilde{y}}}$ (RLS/LMS)	$3N_a$	$3N_a$
	2) (75) $\mathbf{\Sigma}^2_{k,\Delta\mathbf{y}}$	$\sum_{n=1}^{T-1} nT-n^2$	$\sum_{n=1}^{T-1} nT-n^2$
	2) (75) $\mathbf{\Sigma}^2_{k,n,\mathbf{z}}$ (RLS)	$N_a(T^2+T-1)$	$N_a(T^2+T-1)$
	2) (75) $\check{\mathbf{\Sigma}}^2_{k,n,\mathbf{z}}$ (RLS/LMS)	$2N_a$	$2N_a$
	3) (76)	$N_a(4T^2-12T+4)$	$N_a(4T^2-12T+4)$
5	(77)	$2N_aT^2$	$4N_aT^2$
6	(79)-(80)	$N_a(10T+4)+T$	$N_a(12T+4)+T$
	(81)	$N_a(\frac{3}{2}T^2+\frac{5}{2}T+3)$	$N_a(3T^2+5T+3)$
	(78)	$\frac{T}{6}^3+\frac{T}{2}^2+\frac{T}{2}$	$\frac{2T}{3}^3+2T^2$
7	(82)	$N_a(2T+10)$	$N_a(4T+10)$
8	(83)-(86)	$2N_aT+18N_a+T$	$4N_aT+18N_a+2T$

TABLE III

DOMINANT TERMS OF THE MEMORY COST AND COMPUTATIONAL COMPLEXITY OF EQUALIZER FILTERING AND UPDATING FOR THE ADAPTIVE BM-TEQ, BM-PGEQ AND PTEQ. MEMORY AND COMPLEXITY FIGURE ESTIMATES FOR $N_a=224$, $T=16\rightarrow 32$ AND $N_g=4$.

	Memory cost				Computational cost			
	EQ taps (K $=10^3$)		SOS coeffs ($\times 10^3$)		Filtering (excl. DFT) ($\times 10^3$)		Updating ($\times 10^3$)	
real BM-TEQ								
RLS-based SOS	$T+2N_a$	$464\rightarrow 480$	$2N_aT$	$7\rightarrow 15$	$6N_aT$	$22\rightarrow 43$	$8.5N_aT^2$	$570\rightarrow 2100$
RLS/LMS-based SOS	$T+2N_a$	$464\rightarrow 480$	$N_a+\frac{T^2}{2}$	$0.35\rightarrow 0.50$	$6N_aT$	$22\rightarrow 43$	$7.5N_aT^2$	$470\rightarrow 2070$
real BM-PGEQ	N_gT+2N_a	$512\rightarrow 528$	$2N_aT$	$7\rightarrow 15$	$6N_aT$	$22\rightarrow 43$	$8.5N_aT^2$	$570\rightarrow 2110$
complex PTEQ								
RLS	$2N_aT$	7K $\rightarrow$ 14K	$2N_aT$	$7\rightarrow 15$	$2N_aT$	$8\rightarrow 15$	$18N_aT$	$68\rightarrow 134$
RLS/LMS	$2N_aT$	7K $\rightarrow$ 14K	$N_a+\frac{T^2}{2}$	$0.35\rightarrow 0.50$	$2N_aT$	$8\rightarrow 15$	$2N_aT$	$9\rightarrow 18$

BM-PGEQ, which is simulated with $N_g = 4$ equally sized tone groups. All simulations are done for $T = 32$ taps. The TEQ and PGEQ are initialized with $\mathbf{w}_0 = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^T$ unless stated otherwise. The regularization parameter δ_k is initialized with $\delta_0 = 10^{-5}$. The exponential weighting factor for the SOS of the adaptive BM-TEQ is increased from $\lambda = 0.9$ during the first 400 updates over $\lambda = 0.95$ during the next 400 updates to $\lambda = 0.99$ for the last 200 updates; the RLS-based and RLS/LMS-based adaptive PTEQ simulations use $\lambda = 0.998$ and $\lambda = 0.996$, respectively, which result in a sufficiently small excess MSE. When assessing the simulation results below, one should take into account that the choice of λ affects the convergence speed. We include plots for the downstream CSA #4 loop and summarize the performance for all downstream CSA #1-8 loops [6], [8] (with active tones 33 to 256) in tables. Moderate front-end filters to separate upstream and downstream transmission are included in the channel. The synchronization delay Δ is determined by the first sample index of a channel impulse response window of $\nu+1$ samples with maximum energy. The noise is a superposition of AWG noise at -140 dBm/Hz, residual echo and near-end crosstalk from 24 ADSL disturbers. We also include simulations with severe RFI (7 RFIs with carrier frequencies 540, 650, 680, 760, 790, 840 and 1080 kHz; the first two RFIs have a power of -30 dBm, the remaining five have a power of -50 dBm). RFI, especially ingress from AM radio stations, can be an important interferer in ADSL [17]. There exist specific solutions to deal with RFI, e.g., based on *receiver windowing*, but these are beyond the scope of this paper. It has been shown in [17] that the BM-TEQ and BM-PGEQ, as the PTEQ, can effectively mitigate RFI, and outperform suboptimal TEQ designs such as the MMSE-TEQ design.

In [15], the BM-TEQ was found to approach the PTEQ performance very closely, despite possible local minima of the non-convex BM-TEQ design criterion. This result is confirmed here, when comparing the adaptive PTEQ with the adaptive BM-TEQ. **Figure 1** shows bitrate convergence curves as a function of the number of updates for the case *without RFI*. The RLS-based and RLS/LMS-based adaptive PTEQ are compared with the RLM-based adaptive BM-TEQ (with RLS-based and RLS/LMS-based SOS) and the BM-PGEQ (with RLS-based SOS). They all reach the same bitrate of 8.4 Mbps. The RLS-based PTEQ and BM-TEQ curves almost coincide, while the BM-PGEQ converges the fastest. The RLS/LMS-based adaptive PTEQ has the slowest convergence.

If *RFI is present* (**Figure 2**), the convergence of all methods is slower. The adaptive BM-TEQ achieves 6.87 Mbps, which is only 300 kbps (or 4 %) below the adaptive PTEQ bitrate; the adaptive BM-PGEQ fills the gap in convergence speed and bitrate between the adaptive PTEQ and the BM-TEQ. Note that the RLS/LMS-based adaptive PTEQ is too slow in RFI scenarios. As shown on the same Figure 2, the convergence time can be decreased by initializing the RLM algorithm with a cheaply computed suboptimal TEQ (e.g., an MMSE-TEQ, see the thick dotted line; $\delta_0 = 10^{-8}$).

It has been shown in [17] that the BM-PGEQ is less susceptible to poorly performing *local optima*

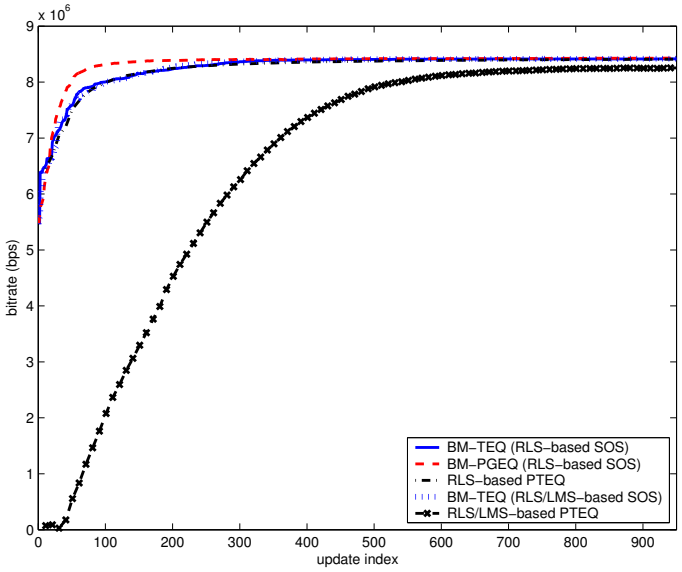

Fig. 1. Bitrate convergence of the adaptive BM-TEQ, BM-PGEQ (4 tone groups) and PTEQ for the CSA #4 loop if no RFI is present. The BM-TEQ and PTEQ are depicted for RLS-based and RLS/LMS-based SOS. $T = 32$.

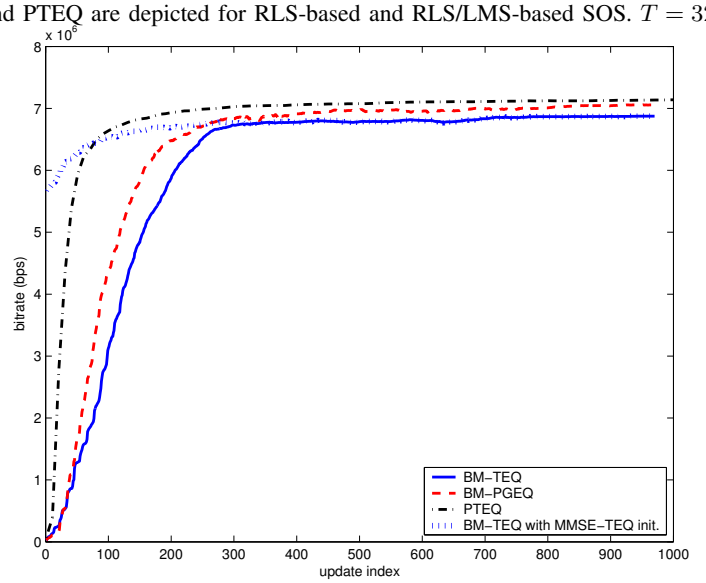

Fig. 2. Bitrate convergence of the adaptive BM-TEQ, BM-PGEQ (4 tone groups) and PTEQ for the CSA#4 loop if RFI is present. $T = 32$. The thick dotted line shows the adaptive BM-TEQ, initialized with an MMSE-TEQ.

than the BM-TEQ, especially if RFI is present. We have run 25 different initializations $\mathbf{w}_0$ to check this property for the adaptive algorithms. **Figure 3** shows convergence plots for the initializations with worst and best performance after 1000 updates. Also here, it is clear that the adaptive BM-PGEQ is less susceptible to poorly performing local optima than the adaptive BM-TEQ: while the two BM-PGEQ curves almost coincide (around 7.06 Mbps), the adaptive BM-TEQ reaches a local optimum between 6.52 and 7.00 Mbps.

As suggested in [18], the seamless rate adaptation functionality of ADSL2 and ADSL2+ allows to adapt the bitrate to new channel conditions, e.g., when one or more RFIs are switched off. **Figure 4**

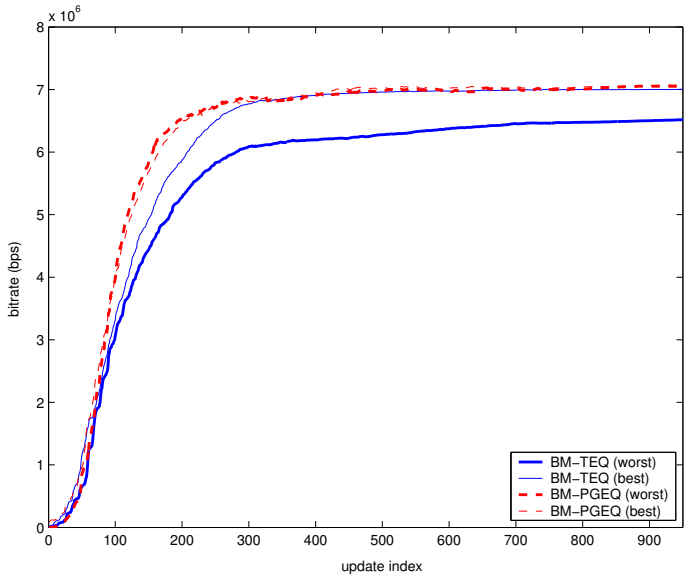

Fig. 3. Bitrate convergence of the adaptive BM-TEQ and BM-PGEQ (4 tone groups) for different initializations if RFI is present. The best and worse performing initializations are depicted. CSA#4 loop. $T = 32$.

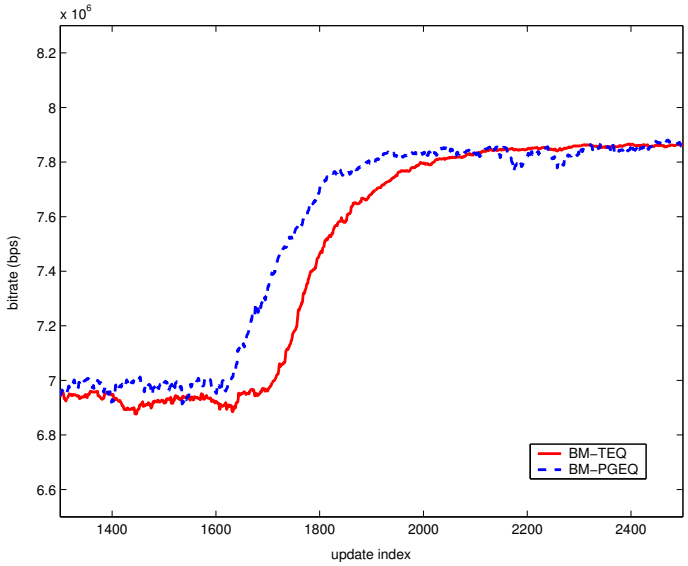

Fig. 4. Tracking performance of the adaptive BM-TEQ and BM-PGEQ (4 tone groups). CSA#4 loop. $T = 32$. $\lambda = 0.9$. Tracking of the disappearance of 2 RFIs at time instant 1500.

shows the tracking capability of the adaptive BM-TEQ and BM-PGEQ (with $\lambda = 0.9$), when the first two RFIs are switched off at time instant 1500. Both algorithms track the change, but the adaptive BM-PGEQ is faster than the adaptive BM-TEQ.

Table IV and **Table V** illustrate the performance for the other CSA loops, with and without RFI, respectively. **Table IV** gives the number of updates to reach 98.5 % of the optimal BM-TEQ, BM-PGEQ and PTEQ bitrate for the adaptive BM-TEQ (with RLS-based and RLS/LMS-based SOS), BM-PGEQ (with RLS-based SOS) and RLS-based and RLS/LMS-based adaptive PTEQ, respectively. The adaptive

TABLE IV

CONVERGENCE OF THE ADAPTIVE BM-TEQ (WITH RLS-BASED AND RLS/LMS-BASED SOS), BM-PGEQ (4 TONE GROUPS, RLS-BASED SOS) AND PTEQ (RLS AND RLS/LMS) FOR THE CSA #1-8 LOOPS IF NO RFI IS PRESENT. NUMBER OF UPDATES TO REACH 98.5 % OF THE OPTIMAL BM-TEQ, BM-PGEQ AND PTEQ RATE.

	#updates to reach 98.5 % of bitrate					optimal bitrate (Mbps)		
loop	BM-TEQ		BM-PGEQ	PTEQ		BM-TEQ	BM-PGEQ	PTEQ
	RLS-based	RLS/LMS-based		RLS	RLS/LMS			
1	128	121	105	349	1025	9.039	9.079	9.133
2	228	306	94	333	1028	9.233	9.258	9.338
3	182	171	124	388	1054	8.292	8.303	8.364
4	235	210	89	382	908	8.414	8.419	8.475
5	99	113	85	339	1066	9.286	9.300	9.356
6	176	188	108	387	1009	8.043	8.039	8.101
7	93	104	100	406	1055	7.422	7.422	7.486
8	139	143	114	436	1158	7.332	7.337	7.394

BM-TEQ performance appears independent of the choice of RLS-based or RLS/LMS-based SOS. The adaptive BM-PGEQ only requires around 100 updates. **Table V** gives the number of updates to reach 95 % and 98.5 % of the optimal BM-TEQ, BM-PGEQ and PTEQ bitrate for the adaptive BM-TEQ, BM-PGEQ (both with RLS-based SOS) and RLS-based adaptive PTEQ, respectively. The adaptive BM-PGEQ performance is more uniform than the adaptive BM-TEQ performance: for all loops, the 95 % threshold is reached after 250-300 updates; the 98.5% threshold requires 400-500 updates. Note that for loop 8, this particular run of the adaptive BM-TEQ reaches a rate of 6.64 Mbps, due to a local optimum, which is below the 98.5 % threshold. The convergence speed of the adaptive BM-PGEQ and PTEQ are comparable.

VII. CONCLUSIONS

In this paper, we have presented an RLM-based adaptive BM-TEQ design method, derived from an IR-SNLLS-based BM-TEQ design criterion. This adaptive BM-TEQ requires the same SOS as the earlier presented RLS-based adaptive PTEQ. A complete range of adaptive BM equalizers then opens up, which all have the same SOS memory cost: from the computationally efficient RLS-based adaptive PTEQ with a large number of equalizer taps, over the adaptive BM-PGEQ, towards a single adaptive BM-TEQ with a larger equalizer update complexity. Through extensive simulations with several ADSL scenarios and parameter settings, it has been shown that the adaptive BM-TEQ exhibits fast convergence, which is comparable to the adaptive PTEQ. Moreover, the adaptive BM-PGEQ closely approaches the

TABLE V

CONVERGENCE OF THE ADAPTIVE BM-TEQ, BM-PGEQ (4 TONE GROUPS) AND PTEQ FOR THE CSA #1-8 LOOPS IF RFI IS PRESENT. NUMBER OF UPDATES TO REACH 95 % AND 98.5 % OF THE OPTIMAL BM-TEQ, BM-PGEQ AND PTEQ RATE.

	#updates to reach 95 % and 98.5 % of bitrate						optimal bitrate (Mbps)		
loop	BM-TEQ		BM-PGEQ		PTEQ		BM-TEQ	BM-PGEQ	PTEQ
	95%	98.5%	95%	98.5%	95%	98.5%			
1	261	315	277	496	128	435	7.915	8.128	8.183
2	277	417	290	492	123	429	8.335	8.403	8.460
3	466	756	226	412	134	488	7.634	7.670	7.716
4	260	703	255	681	157	467	6.945	7.100	7.208
5	255	321	229	409	139	467	7.964	8.102	8.200
6	424	660	225	424	147	488	7.390	7.427	7.495
7	270	325	223	498	146	499	6.582	6.628	6.719
8	591	-	254	512	144	533	6.771	6.793	6.850

performance of the adaptive PTEQ, even in harsh environments with RFI. Besides providing an equalizer design algorithm, adaptive BM equalizers also allow to track variations of the transmission channel and noise characteristics. This makes them a good candidate for integration with the seamless rate adaptation functionality of ADSL2 and ADSL2+ modems.

REFERENCES

[1] International Telecommunications Union (ITU), "Recommendation G.992.1. Asymmetric digital subscriber line (ADSL) transceivers," July 1999.

[2] ——, "Recommendation G.992.3. Asymmetric digital subscriber line (ADSL) transceivers - 2 (ADSL2)," July 2002.

[3] ——, "Recommendation G.992.5. Asymmetrical digital subscriber line (ADSL) transceivers - extended bandwidth ADSL2 (ADSL2+)," May 2003.

[4] The European Telecomm. Standards Inst. (ETSI), "Transmission and multiplexing (TM); Access transmission systems on metallic access cables; Very high speed digital subscriber line (VDSL). TS 101 270 v1.3.1." July 2003.

[5] ——, "Broadband radio access networks (BRAN); HIPERLAN type 2; Physical (PHY) layer. TS 101 475 v1.3.1." Dec. 2001.

[6] N. Al-Dhahir and J. M. Cioffi, "Efficiently computed reduced-parameter input-aided MMSE equalizers for ML detection: A unified approach," *IEEE Trans. Information Theory*, vol. 42, no. 3, pp. 903–915, May 1996.

[7] ——, "Optimum finite-length equalization for multicarrier transceivers," *IEEE Trans. Communications*, vol. 44, no. 1, pp. 56–64, Jan. 1996.

[8] G. Arslan, B. L. Evans, and S. Kiaei, "Equalization for discrete multitone transceivers to maximize bit rate," *IEEE Trans. Signal Processing*, vol. 49, no. 12, pp. 3123–3135, Dec. 2001.

[9] W. Henkel and T. Kessler, "Maximizing the channel capacity of multicarrier transmission by suitable adaptation of the time-domain equalizer," *IEEE Trans. Communications*, vol. 48, no. 12, pp. 2000–2004, Dec. 2000.

[10] P. J. W. Melsa, R. C. Younce, and C. E. Rohrs, "Impulse response shortening for discrete multitone transceivers," *IEEE Trans. Communications*, vol. 44, no. 12, pp. 1662–1672, Dec. 1996.

[11] M. Milosevic, L. F. C. Pessoa, B. L. Evans, and R. Baldick, "Optimal time domain equalization design for maximizing data rate of discrete multi-tone systems," *IEEE Trans. Signal Processing*, Accepted for publication.

[12] R. K. Martin, J. Balakrishnan, W. A. Sethares, and C. R. Johnson, Jr., "A blind, adaptive TEQ for multicarrier systems," *IEEE Signal Processing Lett.*, vol. 9, no. 11, pp. 341–343, Nov. 2002.

[13] J. Balakrishnan, R. K. Martin, and C. R. Johnson, Jr., "Blind, adaptive channel shortening by sum-squared auto-correlation minimization (SAM)," *IEEE Trans. Signal Processing*, vol. 51, no. 12, pp. 3086–3093, Dec. 2003.

[14] K. Van Acker, G. Leus, M. Moonen, O. van de Wiel, and T. Pollet, "Per-tone equalization for DMT-based systems," *IEEE Trans. Communications*, vol. 49, no. 1, pp. 109–119, Jan. 2001.

[15] K. Vanbleu, G. Ysebaert, G. Cuypers, M. Moonen, and K. Van Acker, "Bitrate maximizing time-domain equalizer design for DMT-based systems," *IEEE Trans. Communications*, Nov. 2003. Accepted for publication. Available via ftp://ftp.esat.kuleuven.ac.be/pub/sista/vanbleu/reports/02-118.pdf.

[16] K. Vanbleu, G. Ysebaert, G. Cuypers, and M. Moonen, "On time-domain and frequency-domain MMSE-based TEQ design for DMT transmission," ESAT, K.U.Leuven, Belgium, Tech. Rep. ESAT-SISTA/TR 2004-02. Submitted for publication. Available at ftp://ftp.esat.kuleuven.ac.be/pub/sista/vanbleu/reports/04-02.pdf, Jan. 2004.

[17] ——, "Bitrate maximizing per group equalization for DMT-based systems," Submitted for publication. Available at ftp://ftp.esat.kuleuven.ac.be/pub/sista/vanbleu/reports/03-48.pdf.

[18] A. Inc., "ADSL2 and ADSL2+ - the new ADSL standards," Tech. Rep., Mar. 2003, white paper.

[19] T. Starr, J. M. Cioffi, and P. J. Silvermann, *Understanding Digital Subscriber Line Technology*. Englewood Cliffs, NJ: Prentice Hall, 1999.

[20] K. Van Acker, G. Leus, M. Moonen, and T. Pollet, "RLS-based initialization for per-tone equalizers in DMT receivers," *IEEE Trans. Communications*, vol. 51, no. 6, pp. 885–889, June 2003.

[21] G. Ysebaert, K. Vanbleu, G. Cuypers, M. Moonen, and T. Pollet, "Combined RLS-LMS initialization for per tone equalizers in DMT-receivers," *IEEE Trans. Signal Processing*, vol. 51, no. 7, pp. 1916–1927, July 2003.

[22] L. Ljung and T. Söderström, *Theory and Practice of Recursive Identifi cation* Cambridge, MA: MIT Press, 1983.

[23] R. H. Byrd and D. A. Payne, "Convergence of the iteratively reweighted least squares algorithm for robust regression," The Johns Hopkins University, Baltimore, MD, Tech. Rep. 313, June 1979.

[24] A. Bjorck, *Least-Squares Methods*. Elsevier, 1987.

[25] G. Golub and V. Pereyra, "Separable nonlinear least squares: the variable projection method and its applications," *Institute of Physics, Inverse Problems*, vol. 19, no. 2, pp. R1–R26, Apr. 2003.

[26] L. S. H. Ngia and J. Sjöberg, "Efficient training of neural nets for nonlinear adaptive filtering using a recursive Levenberg-Marquardt algorithm," *IEEE Trans. Signal Processing*, vol. 48, no. 7, pp. 1915–1927, July 2000.

[27] H. Dai and N. K. Sinha, "Robust recursive least-squares method with modified weights for bilinear system identification," *Proc. IEEE*, vol. 136, no. 3, pp. 122–126, May 1989.

[28] S. Haykin, "Adaptive digital communication receivers," *IEEE Commun. Mag.*, vol. 38, no. 12, pp. 106–114, Dec. 2000.

[29] A. T. Georgiadis and B. Mulgrew, "Adaptive Bayesian decision feedback equalizer for alpha-stable noise environments," *Signal Processing*, vol. 81, no. 8, pp. 1603–1623, Aug. 2001.

[30] S. Haykin, *Adaptive Filter Theory*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[31] G. H. Golub and C. F. Van Loan, *Matrix Computations - Third Edition*. The John Hopkins University Press, 1996.