Bitrate Maximizing Per Group Equalization for DMT-based Systems

Koen Vanbleu, Geert Ysebaert, Gert Cuypers *, Marc Moonen

Katholieke Universiteit Leuven, Dept. ESAT/SCD-SISTA, Kasteelpark Arenberg 10, 3001 Leuven-Heverlee, Belgium

Abstract

In a previous paper, we proposed a bitrate maximizing design criterion for time-domain equalizers (TEQ) in DMT transceivers to shorten the channel impulse response, as needed in, e.g., ADSL receivers. The proposed criterion truly maximizes the bitrate, as it is based on an exact formulation of the subchannel SNR as a function of the TEQ taps. In this paper, we show how the BM-TEQ design can be used in a bitrate maximizing per group equalization scheme (BM-PGEQ): the active tones are divided into groups and each group is provided with a bitrate maximizing equalizer. This BM-PGEQ design allows for a trade-off between memory requirement and performance, keeping computational complexity during data transmission roughly at the same level. It encompasses the BM-TEQ design and the so-called per tone equalization scheme (PTEQ) as extreme cases. We also present an adaptation algorithm to design the BM-TEQ and BM-PGEQ. Through simulation, we show that the BM-PGEQ scheme outperforms an earlier presented tone grouping scheme where the whole tone group was assigned the PTEQ of the group center tone. The BM-PGEQ scheme appears a useful intermediate between BM-TEQ and PTEQ and closely approaches the PTEQ performance for as few as 4 tone groups in an ADSL scenario, even in harsh environments with narrowband interference and crosstalk.

Key words: DSL, Discrete Multitone, DMT, OFDM, Time-Domain Equalization, Channel Shortening, Per Tone Equalization, Per Group Equalization, Adaptive Equalization

Number of pages: 29

Number of tables: 2

Number of figures: 3

* Corresponding author.
Email address: gert.cuypers@esat.kuleuven.ac.be (Gert Cuypers).

1 Introduction

Multicarrier modulation has regained interest over the last decade. Several all-digital variants have been proposed: discrete multitone (DMT) is adopted as transmission format for asymmetric digital subscriber lines (ADSL) and a strong candidate for very high bit rate digital subscriber lines (VDSL); orthogonal frequency division multiplexing (OFDM) is adopted in wireless local area applications, e.g., HIPERLAN/2.

DMT schemes divide the available bandwidth into parallel subchannels or tones. The incoming bitstream is split into parallel streams that are used to QAM-modulate the different tones. The modulation is done by means of an inverse fast Fourier transform (IFFT). 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 the cyclic prefix length $\nu + 1$, demodulation can be implemented by means of an FFT, followed by a (complex) 1-tap frequency domain equalizer (FEQ) per tone to compensate for channel amplitude and phase effects.

Practical ADSL channel impulse responses can be several hundreds of samples long, hence a large prefix would be required. However, a long prefix results in a large overhead with respect to the data rate. An existing solution for this problem is to insert a (real) T-tap time domain equalizer (TEQ) before demodulation that shortens the channel impulse response to $\nu + 1$ samples. Many algorithms have been developed to initialize this TEQ, but none of them truly optimizes bitrate. They can be roughly divided into three classes.

(1) Minimum mean-square error (MMSE) based methods, such as in [1,2], minimize the time-domain error between the TEQ output and a desired output, i.e., the output of an FIR filter of length $\nu + 1$, called target impulse response (TIR), fed with the transmitted time-domain sequence. The TIR is part of the parameter vector. These methods are improved upon in [3] which uses an eigenapproach for the filter design.
(2) A computationally efficient generalization of the "maximum shortening-SNR" (MSSNR) method is presented in [4], called "minimum-ISI method". The original MSSNR method is described in [5] and aimed at maximizing the ratio of the energy of the channel impulse response inside a target window of length $\nu + 1$ to the energy outside the target window. None of these methods truly optimizes bitrate.
(3) In [6], the authors attempt to maximize the bitrate by maximizing the geometric mean of subchannel SNRs. However, several assumptions render the method suboptimal, e.g., the subchannel SNR model takes neither intercarrier interference (ICI) and intersymbol interference (ISI) of the DMT signal nor DFT leakage of the noise (like thermal noise, crosstalk,

radio frequency interference (RFI), residual echo, etc.) into account. Acknowledging this fact, improved subchannel SNR formulations are given in [4,7,8], but, also there, the signal contribution of interest, the ISI/ICI and/or the DFT leakage are only partially included.

In [9], we presented a nonlinear, nonconvex TEQ design criterion that has as its optimal solution the *bitrate maximizing TEQ* (BM-TEQ) for a given number of taps T.

An alternative bitrate-maximizing structure is offered by the per tone equalizer (PTEQ) in which the equalization is done for each tone separately after the FFT-demodulation [10] (i.e. not using a TEQ+FEQ structure). In [11], a PTEQ scheme with *tone grouping* has been introduced to reduce equalizer design complexity and memory requirement. The idea there was to compute the optimal PTEQ taps for the group center tone only and to reuse these taps for the whole group.

In this paper, we show how the BM-TEQ design can be used in a *bitrate maximizing "per group" equalization scheme (BM-PGEQ)*: the active tones are divided into groups and each group is provided with a (possibly complex) T tap bitrate maximizing equalizer by solving the nonlinear BM-TEQ criterion for that group. This BM-PGEQ design allows for a trade-off between the amount of memory required for storing the equalizer coefficients, and performance, keeping computational complexity during data transmission roughly at the same level. It encompasses the BM-TEQ and the PTEQ as extreme cases, as will be shown: a BM-TEQ only requires one T-tap equalizer (all active tones in just one "group"), whereas the PTEQ scheme can be considered a BM-PGEQ design with "groups" of 1 tone, requiring a memory of N_aT taps (with N_a the number of active tones, e.g., around 220 tones in downstream ADSL).

A summary of the derivation of the BM-TEQ criterion is given in **Section 2**. The BM-PGEQ scheme is introduced in **Section 3**. In **Section 3**, it is shown that the *nonlinear* BM-PGEQ criterion with *single-tone-groups* and complex equalizer coefficients indeed reduces to the PTEQ of [10], which can then be computed by solving a *linear* MMSE problem.

In [4,7,8], the authors resort to (complex) batch optimization procedures for solving the nonlinear TEQ cost function. However, the nonlinear BM-TEQ and BM-PGEQ criteria, presented here, lend themselves to *recursive or adaptive optimization*: the TEQ and PGEQ taps are designed on-the-fly, based on the training sequence transmitted during connection set-up. The resulting adaptive algorithm can then be used for training-based TEQ/PGEQ design, as well as in decision directed mode during data transmission to track changes in channel and noise. **Section 4** deals with this adaptive procedure for solving the nonconvex BM-TEQ and BM-PGEQ criteria.

In the ADSL simulations of **Section 5**, we compare the performance of the BM-TEQ, BM-PGEQ, PTEQ and original tone-grouping scheme and examine the susceptibility to local optima, inherent to nonconvex optimization, by using several initial TEQ settings. We consider a harsh environment that suffers heavily from narrowband interference, such as radio frequency interference (RFI), and crosstalk. The simulations confirm the usefulness of the BM-PGEQ scheme as an intermediate between BM-TEQ and PTEQ since it performs close to PTEQ and reaches local optima with quasi-identical performance for as few as 3 to 4 tone groups.

Notation. We summarize our notation here, which is mostly adopted from [10].

- n is a tone index; g is a tone group index; k is the DMT symbol index; N is the (I)FFT size; ν is the cyclic prefix length.
- $\mathcal{S}_a$ is the set of N_a active tones, e.g., tones 38 to 256 for downstream ADSL; $\mathcal{S}_g$ denotes tone group g; the total number of tone groups is N_g.
- $\mathcal{F}$ is an orthonormal DFT matrix of size N; its n-th row is $\mathcal{F}_n$.
- $\mathbf{w}$ is the time-domain equalizer (TEQ, T taps); $\mathbf{w}_g$ is the per-group equalizer (PGEQ) for the tone group with index g. All notation below is given for the case of a TEQ $\mathbf{w}$. The notation for a PGEQ or PTEQ is then obtained by replacing $\mathbf{w}$ with $\mathbf{w}_g$ or $\mathbf{w}_n$, respectively.
- As detailed below, 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; a variable with subscript $\mathbf{w}$ or $\mathbf{w}_g$ indicates that the data have been filtered with a TEQ or PGEQ, respectively; a selection of elements of a vector or matrix are specified with indices between brackets.
- Define a vector of received samples i to j of the k-th DMT symbol by

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

  where it is tacitly assumed that the samples $y_{k,l}$ depend on the the synchronization delay Δ, a design parameter that estimates the channel group delay [1].
- Then, the time-domain vector of length N at the TEQ output that is fed to the FFT 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 received 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 the matrix-vector product $\mathbf{y}_{k,\mathbf{w}} = \mathbf{Y}_k\mathbf{w}$ corresponds to the convolu-

tion of the k-th received DMT symbol $\mathbf{y}_{k,-T+1:N-1}$ and the TEQ.

- The sliding DFT $\mathcal{F}\mathbf{Y}_k$ (i.e., the DFT of the T columns of $\mathbf{Y}_k$) is denoted as $\tilde{\mathbf{Y}}_k$ ($N \times T$ matrix); the n-th sliding DFT output is a $1 \times T$ row vector $\tilde{\mathbf{y}}_{k,n} = \mathcal{F}_n\mathbf{Y}_k = \tilde{\mathbf{Y}}_k[n,:]$.
- The $N \times 1$ FFT output vector *after* TEQ $\mathbf{w}$ is given by $\tilde{\mathbf{y}}_{k,\mathbf{w}} = \mathcal{F}\mathbf{Y}_k\mathbf{w} = \tilde{\mathbf{Y}}_k\mathbf{w}$; the n-th FFT output (FEQ input) is $\tilde{y}_{k,n,\mathbf{w}} = \tilde{\mathbf{y}}_{k,\mathbf{w}}[n] = \mathcal{F}_n\mathbf{Y}_k\mathbf{w} = \tilde{\mathbf{y}}_{k,n}\mathbf{w}$.
- The $N \times 1$ FFT output vector *without* TEQ or PGEQ is given by $\tilde{\mathbf{y}}_k = \mathcal{F}\mathbf{y}_{k,0:N-1} = \tilde{\mathbf{Y}}_k[:,1]$; the n-th entry of $\tilde{\mathbf{y}}_k$ is denoted $\tilde{y}_{k,n} = \tilde{\mathbf{y}}_k[n] = \mathcal{F}_n\mathbf{y}_{k,0:N-1}$.
- The k-th (frequency-domain) DMT symbol vector that is fed to the modulating IFFT is $\tilde{\mathbf{x}}_k$; the symbol on tone n is $\tilde{x}_{k,n} = \tilde{\mathbf{x}}_k[n]$.
- The symbol estimate for symbol $\tilde{x}_{k,n}$ at the receiver output is given by $\hat{\tilde{x}}_{k,n} = D_n\tilde{y}_{k,n,\mathbf{w}}$ where D_n denotes the complex FEQ for tone n.
- $\mathcal{E}\{\cdot\}$ is the expectation operator.
- $\sigma^2_{\tilde{x},n} = \mathcal{E}\left\{|\tilde{x}_{k,n}|^2\right\}$, $\Sigma^2_{\tilde{\mathbf{y}},n} = \mathcal{E}\left\{\tilde{\mathbf{y}}^H_{k,n}\tilde{\mathbf{y}}_{k,n}\right\}$, $\Sigma_{\tilde{x}\tilde{\mathbf{y}},n} = \mathcal{E}\left\{\tilde{x}^*_{k,n}\tilde{\mathbf{y}}_{k,n}\right\}$ are, respectively, the scalar variance of the transmitted symbol at tone n, $\tilde{x}_{k,n}$, the autocorrelation matrix of the n-th sliding DFT output $\tilde{\mathbf{y}}_{k,n}$ and the cross-correlation vector of the n-th transmitted symbol $\tilde{x}_{k,n}$ and the sliding DFT output $\tilde{\mathbf{y}}_{k,n}$.

2 A bitrate maximizing TEQ criterion: review [9]

The ultimate benchmark is the total number of bits b_{DMT} transmitted in one DMT symbol,

$$b_{DMT} = \sum_{n\in\mathcal{S}_a} \log_2\left(1 + \frac{\mathrm{SNR}_n}{\Gamma_n}\right) \tag{3}$$

where $\mathcal{S}_a$ is the set of active tones, 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 [6]. It can be proven [9] that the TEQ (BM-TEQ) vector $\mathbf{w}$ maximizing (3) is the solution of the following optimization problem

$$\arg\max_{\mathbf{w}} b_{DMT} = \arg\min_{\mathbf{w}} \sum_{n\in\mathcal{S}_a} \log\left(\frac{\mathbf{w}^T\mathbf{B}_n\mathbf{w}}{\mathbf{w}^T\mathbf{A}_n\mathbf{w}}\right), \tag{4}$$

in which the tone-dependent matrices $\mathbf{A}_n$ and $\mathbf{B}_n$ are independent of $\mathbf{w}$ but depend on the second-order signal statistics $\sigma^2_{\tilde{x},n}$, $\Sigma^2_{\tilde{\mathbf{y}},n}$ and $\Sigma_{\tilde{x}\tilde{\mathbf{y}},n}$:

$$\mathbf{A}_n = \Gamma_n\sigma^2_{\tilde{x},n}\Sigma^2_{\tilde{\mathbf{y}},n} + (1-\Gamma_n)\Sigma^H_{\tilde{x}\tilde{\mathbf{y}},n}\Sigma_{\tilde{x}\tilde{\mathbf{y}},n} \tag{5}$$

$$\mathbf{B}_n = \Gamma_n \left(\sigma^2_{\tilde{x},n} \Sigma^2_{\tilde{\mathbf{y}},n} - \Sigma^H_{\tilde{x}\tilde{\mathbf{y}},n} \Sigma_{\tilde{x}\tilde{\mathbf{y}},n} \right) \tag{6}$$

Note that (4) has a scalar ambiguity: if $\mathbf{w}$ is a solution, then $\gamma\mathbf{w}$ (with γ a scalar) is also a solution. To avoid this parameter ambiguity, one could impose a constraint on $\mathbf{w}$, e.g., $\|\mathbf{w}\| = 1$, which, contrary to the MMSE-based TEQ design, does not influence the optimum bitrate, as b_{DMT} in (4) is independent of γ.

Minimizing (4) is a nonlinear, nonconvex optimization problem. Due to the nonconvex nature of the BM-TEQ cost function, any optimization procedure for solving (4) only assures convergence to a local optimum [12]. Nonlinear *batch optimization methods* such as advanced, *iterative* (quasi-)Newton algorithms and simplex algorithms, are abundantly available in commercial software packages. Therefore, one needs for all N_a tones an estimate of the statistics $\Sigma^2_{\tilde{\mathbf{y}},n}$ and $\Sigma_{\tilde{x}\tilde{\mathbf{y}},n}$, appearing in (5) and (6).

In Section 4, an alternative, *adaptive* design procedure for the BM-TEQ cost function will be given.

3 Per group equalization (PGEQ)

The per tone equalization (PTEQ) scheme was introduced in [10] as an alternative DMT equalization scheme that always performs at least as well as - and usually much better than - a TEQ based receiver while keeping complexity during data transmission at the same level. Equalization is done for each tone separately after the FFT-demodulation, hence the name. In [11], a reduced complexity *tone grouping scheme* (TG) for the PTEQ is introduced. The active tones are divided into groups and only the PTEQ of the center tone of each group is optimized; the other tones in the group are operated with a PTEQ, corresponding to the equivalent TEQ of the center tone. This scheme allows for a considerable reduction in equalizer design complexity and memory requirement, but it does not guarantee better performance than a TEQ, especially for large tone groups.

The BM-TEQ criterion (4) can be used in a new "per group" equalization scheme (PGEQ), which is intermediate (in terms of memory requirement and performance) between BM-TEQ and PTEQ: the active tones are divided into N_g groups and each group is provided with a (possibly complex) T tap bitrate maximizing per-group equalizer (BM-PGEQ) $\mathbf{w}_g$ by solving the nonlinear BM-TEQ criterion for this group:

$$\arg\max_{\mathbf{w}_g} b_{g,DMT} = \arg\min_{\mathbf{w}_g} \sum_{n\in\mathcal{S}_g} \log\left(\frac{\mathbf{w}_g^T\mathbf{B}_n\mathbf{w}_g}{\mathbf{w}_g^T\mathbf{A}_n\mathbf{w}_g}\right) \text{ for } g = 1,\cdots,N_g \qquad (7)$$

As the PGEQ for each tone group optimizes the bitrate for this group, the BM-PGEQ scheme will perform better than the original TG structure and obviously better than a single BM-TEQ. Hence, the BM-PGEQ design encompasses the BM-TEQ and PTEQ design procedure as extreme cases, as will be demonstrated. In Section 3, it is shown that the nonlinear BM-PGEQ criterion with "groups" of 1 tone and complex equalizer coefficients leads to the PTEQ of [10], which can be solved as a linear MMSE problem.

As will be shown in Section 5, the BM-PGEQ scheme is less susceptible than a BM-TEQ to worse performing local optima of the nonlinear cost function, especially in harsh environments with narrowband interference. The BM-PGEQ scheme with as few as 3 to 4 tone groups, requiring only N_gT equalizer taps (e.g., $N_g \times T = 4 \times 32 = 128$), closely approaches the performance of the PTEQ with N_aT taps (e.g., $N_a \times T = 220 \times 32 = 7060$).

Once the BM-PGEQ coefficients have been designed by solving (7), the BM-PGEQ can be implemented in two ways, depending on whether minimum memory or minimum computational complexity is required.

BM-PGEQ with minimum memory requirement Assume we use N_g tone groups and a BM-PGEQ $\mathbf{w}_g$ $(g = 1,\cdots,N_g)$ of T taps per group, then N_gT PGEQ taps plus N_a FEQs D_n need to be stored. One could naively calculate the $N \times 1$ TEQ output $\mathbf{y}_{k,\mathbf{w}_g} = \mathbf{Y}_k\mathbf{w}_g$ and then apply a (partial) FFT $\tilde{\mathbf{y}}_{k,\mathbf{w}_g} = \mathcal{F}\mathbf{Y}_k\mathbf{w}_g$ for each group. However, it is often computationally cheaper to exploit the structure in the sliding DFT $\tilde{\mathbf{y}}_{k,n} = \mathcal{F}_n\mathbf{Y}_k$, as explained in Appendix I:

$$\tilde{\mathbf{y}}_{k,n} = \begin{bmatrix} \tilde{y}_{k,n} & \Delta\mathbf{y}_k \end{bmatrix} \underbrace{\begin{bmatrix} 1 & \alpha_n & \cdots & \alpha_n^{T-1} \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \alpha_n \\ 0 & \cdots & 0 & 1 \end{bmatrix}}_{\mathbf{P}_n} \qquad (8)$$

The BM-PGEQ output can then be obtained efficiently as follows.

(1) Calculate $T-1$ (real) difference terms

$$\Delta\mathbf{y}_k = \Big[(y_{k,-1} - y_{k,N-1}) \cdots (y_{k,-T+1} - y_{k,-N-T+1}) \Big] \tag{9}$$

and one FFT of unequalized received signal samples

$$\tilde{\mathbf{y}}_k = \mathcal{F}\mathbf{y}_{k,0:N-1}. \tag{10}$$

(2) Calculate the sliding DFT outputs $\tilde{\mathbf{y}}_{k,n}$ for all N_a tones of $\mathcal{S}_a$ using the recursion that follows from (8):

$$\begin{aligned} \tilde{\mathbf{y}}_{k,n}[t+1] &= \alpha_n \tilde{\mathbf{y}}_{k,n}[t] + (y_{k,-t} - y_{k,-t+N}),\ 1 \le t \le T-1 \\ &= \tilde{y}_{k,n} = \tilde{\mathbf{y}}_k[n],\ t = 0 \end{aligned} \tag{11}$$

with $\alpha_n = \exp(-j2\pi(n-1)/N)/\sqrt{N}$.

(3) Calculate for all N_a tones of $\mathcal{S}_a$ the inner product of $\tilde{\mathbf{y}}_{k,n}$ with the appropriate PGEQ $\mathbf{w}_g$ and multiply the result with the FEQ D_n to obtain the symbol estimate for tone n, symbol k:

$$\hat{\tilde{x}}_{k,n} = D_n \tilde{y}_{k,n,\mathbf{w}_g} = D_n(\tilde{\mathbf{y}}_{k,n}\mathbf{w}_g). \tag{12}$$

The computational complexity per DMT symbol is then (in case of real PGEQs) $6N_aT$ real multiplications and 1 FFT operation.

BM-PGEQ with minimum computational complexity Alternatively, one can resort to a BM-PGEQ implementation with the same complexity and memory requirement as the PTEQ [10]. Exploiting (8), the per-group equalizer output for tone n, $\tilde{y}_{k,n,\mathbf{w}_g}$, can be written as

$$\tilde{y}_{k,n,\mathbf{w}_g} = \mathcal{F}_n \mathbf{Y}_k \mathbf{w}_g = \Big[\tilde{y}_{k,n}\ \Delta\mathbf{y}_k \Big] \mathbf{v}_n \tag{13}$$

where $\tilde{y}_{k,n}$ and $\Delta\mathbf{y}_k$ have been defined in (10) and (9), respectively, and where

$$\mathbf{v}_n = \underbrace{\begin{bmatrix} 1 & \alpha_n & \cdots & \alpha_n^{T-1} \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \alpha_n \\ 0 & \cdots & 0 & 1 \end{bmatrix}}_{\mathbf{P}_n} (D_n \mathbf{w}_g) \tag{14}$$

with $\alpha_n = \exp(-j2\pi(n-1)/N)/\sqrt{N}$. Note that the tone-dependent equalizers $\mathbf{v}_n$ can be obtained efficiently by exploiting the recursion in (14), which is very similar to the recursion in (11). Instead of the N_g per-group equalizers

$\mathbf{w}_g$, the N_a transformed, tone-dependent equalizers $\mathbf{v}_n$ are to be stored. The BM-PGEQ outputs are then obtained efficiently by means of the right-hand equation in (13). As with the PTEQ, $\mathbf{v}_n$ is a $T\times 1$ vector with complex equalizer taps, hence this implementation requires memory for N_aT complex taps and has a computational complexity of 1 FFT and $(2T+2)N_a$ real multiplications per DMT symbol.

Table 1 summarizes the memory requirement and overall computational complexity per DMT symbol of a (real) TEQ, a (real) PGEQ and a (real and complex) PTEQ-based receivers. All receivers need 1 FFT per DMT symbol. Note that, in ADSL, the number of active tones N_a is at most equal to $N/2$, so $N \geq 2N_a$.

Here comes Table 1.

Relation with per tone equalization The PTEQ forms an upper performance bound for any TEQ-based design with the same number of taps while maintaining a comparable complexity. This last statement can easily be understood: while a TEQ-based approach implements only one N-taps filter as opposed to a PTEQ having such a filter for each individual tone, it runs at sample rate whereas the PTEQ filters run at the symbol rate. However, the PTEQ has the advantage of each individual tone having its own filter, thereby offering more tuning freedom and a higher bit rate. It is now intuitively clear that the case of *single-tone-groups* (i.e., "groups" of 1 tone, $g = 1, \cdots, N_a$) should offer a performance identical to an N-taps PTEQ and this is indeed true. For a comparison between the BM-TEQ and the PTEQ we refer to [13]

If one allows the BM-PGEQ coefficients to take complex values an additional FEQ will not be needed and the result is exactly equal to a (complex) PTEQ. Alternatively, the BM-PGEQ coefficients can be constrained to take real values only. In this case a FEQ will be needed. This structure can then be implemented in the same way as the "BM-PGEQ with minimum memory requirement", discussed in Section 3. Its memory requirement and computational complexity are included in Table 1 ('real PTEQ').

4 BM-PGEQ adaptation

In Section 2, we proposed nonlinear *batch optimization* routines to design the BM-TEQ. With the BM-PGEQ scheme, these optimization methods are to be applied for each tone group. Also in [4,7,8], the authors resort to (complex) batch optimization procedures for designing the TEQ.

As we make use of a subchannel SNR model at the FEQ output, the here presented nonlinear BM-TEQ and BM-PGEQ criteria (4) and (7) are expressed

in terms of the data statistics $\Sigma^2_{\tilde{\mathbf{y}},n}$ and $\Sigma_{\tilde{x}\tilde{\mathbf{y}},n}$. These statistics can be estimated online using, e.g., an exponentially weighted window. Therefore, the nonlinear BM-TEQ and BM-PGEQ cost function are suited for *recursive or adaptive optimization*: the TEQ and PGEQ taps are then designed on-the-fly, based on the training sequence with cyclic prefix transmitted during connection set-up. The resulting adaptation algorithm can be used for training-based TEQ/PGEQ design, as well as in decision directed mode during data transmission to track changes in channel and noise.

We devise a recursive updating algorithm for $\mathbf{w}_g$ based on a stochastic Newton algorithm with Gauss-Newton-like search direction [12]; stochastic gradient-based alternatives have been found to converge to slowly. The updating algorithm is called "recursive" as it updates the previous parameter estimate $\mathbf{w}_{k-1,g}$, based on data up to time $k-1$, using the newly available data at time k. The new PGEQ estimate $\mathbf{w}_{k,g}$ at time k is an approximation of the (locally) optimal $\mathbf{w}^{opt}_{k,g}$ that minimizes the BM-PGEQ cost function (7) based on all data up to time k; the approximation becomes more accurate for larger k. The estimate $\mathbf{w}_{k,g}$ is obtained from the previous estimate $\mathbf{w}_{k-1,g}$ as:

$$\mathbf{w}_{k,g} \leftarrow \mathbf{w}_{k-1,g} - \mathbf{R}^{\dagger}_{k,g}(\mathbf{w}_{k-1,g})\mathbf{g}_{k,g}(\mathbf{w}_{k-1,g}) \tag{15}$$

$$\mathbf{w}_{k,g} \leftarrow \mathbf{w}_{k,g} / \left\|\mathbf{w}_{k,g}\right\| \tag{16}$$

where $\mathbf{g}_{k,g}(\mathbf{w}_{k-1,g})$ and $\mathbf{R}_{k,g}(\mathbf{w}_{k-1,g})$ are, respectively, the gradient and a positive semidefinite approximation of the Hessian of the BM-PGEQ cost function, based on all data up to time k and evaluated in $\mathbf{w}_{k-1,g}$. The Hessian of a nonlinear cost function is not necessarily positive (semi)definite, hence its use could result in divergence. Therefore, the Hessian is typically replaced by an accurate positive (semi)definite approximation [12]. $\mathbf{R}^{\dagger}_{k,g}(\mathbf{w}_{k-1,g})$ is the pseudo-inverse of the Hessian approximation $\mathbf{R}_{k,g}(\mathbf{w}_{k-1,g})$ ([14], p. 257). Finally, after each update, the new estimate is normalized by (16) to solve the aforementioned parameter ambiguity. In Appendix II, it is shown that the gradient and a positive semidefinite approximation of the Hessian of the BM-PGEQ cost function (7) are given by

$$\mathbf{g}_g(\mathbf{w}_g) = -\sum_{n\in\mathcal{S}_g} 2\alpha_{n,\mathbf{w}_g}\Re\left\{\frac{\Sigma^H_{\tilde{x}\tilde{\mathbf{y}},n}}{\mathbf{w}_g^T\Sigma^H_{\tilde{x}\tilde{\mathbf{y}},n}} - \frac{\Sigma^2_{\tilde{\mathbf{y}},n}\mathbf{w}_g}{\mathbf{w}_g^T\Sigma^2_{\tilde{\mathbf{y}},n}\mathbf{w}_g}\right\} \tag{17}$$

$$= -\sum_{n\in\mathcal{S}_g} 2\tilde{\alpha}_{n,\mathbf{w}_g}\Re\left\{\mathcal{E}\left\{\tilde{\mathbf{y}}^H_{k,n}(D^{MMSE}_n)^*(\tilde{x}_{k,n} - D^{MMSE}_n\tilde{\mathbf{y}}_{k,n}\mathbf{w}_g)\right\}\right\} \tag{18}$$

and

$$\mathbf{R}_g(\mathbf{w}_g) = \sum_{n\in\mathcal{S}_g} 2\alpha_{n,\mathbf{w}_g}\Re\left\{\frac{\Sigma^2_{\tilde{\mathbf{y}},n}}{\mathbf{w}_g^T\Sigma^2_{\tilde{\mathbf{y}},n}\mathbf{w}_g} - \frac{\Sigma^2_{\tilde{\mathbf{y}},n}\mathbf{w}_g\mathbf{w}_g^T\Sigma^2_{\tilde{\mathbf{y}},n}}{\left(\mathbf{w}_g^T\Sigma^2_{\tilde{\mathbf{y}},n}\mathbf{w}_g\right)^2}\right\} \tag{19}$$

$$= \sum_{n\in\mathcal{S}_g} 2\tilde{\alpha}_{n,\mathbf{w}_g}\Re\left\{\underbrace{\mathcal{E}\left\{\left[\tilde{\mathbf{y}}^H_{k,n}(D_n^{MMSE})^*\right]\left[D_n^{MMSE}\tilde{\mathbf{y}}_{k,n}\right]\right\}}_{[1]}\right. \tag{20}$$

$$\left. - \underbrace{\frac{\mathcal{E}\left\{\left[\tilde{\mathbf{y}}^H_{k,n}(D_n^{MMSE})^*\right]\tilde{\mathbf{y}}_{k,n}\right\}\mathbf{w}_g\mathbf{w}_g^T\mathcal{E}\left\{\tilde{\mathbf{y}}^H_{k,n}\left[D_n^{MMSE}\tilde{\mathbf{y}}_{k,n}\right]\right\}}{\mathbf{w}_g^T\Sigma^2_{\tilde{\mathbf{y}},n}\mathbf{w}_g}}_{[2]}\right\}$$

Focusing first on (17) and (19), both the gradient and the Hessian approximation are a weighted sum of tone-dependent contributions. The weights, which depend on $\mathbf{w}_g$, are given by

$$\alpha_{n,\mathbf{w}_g} = \frac{\rho^2_{n,\mathbf{w}_g}}{(\Gamma_n + (1-\Gamma_n)\rho^2_{n,\mathbf{w}_g})(1-\rho^2_{n,\mathbf{w}_g})} \tag{21}$$

where $\rho^2_{n,\mathbf{w}_g}$is the squared normalized correlation for tone n,

$$\rho^2_{n,\mathbf{w}} = \frac{\mathbf{w}^T\Sigma^H_{\tilde{x}\tilde{\mathbf{y}},n}\Sigma_{\tilde{x}\tilde{\mathbf{y}},n}\mathbf{w}}{\sigma^2_{\tilde{x},n}(\mathbf{w}^T\Sigma^2_{\tilde{\mathbf{y}},n}\mathbf{w})} \tag{22}$$

. $\Re\{\cdot\}$ takes the real part and ensures a real PGEQ; by omitting $\Re\{\cdot\}$, a complex PGEQ is obtained. The gradient estimate $\mathbf{g}_{k,g}(\mathbf{w}_{k-1,g})$ and Hessian approximation estimation $\mathbf{R}_{k,g}(\mathbf{w}_{k-1,g})$, needed in (15), are then obtained from (17) and (19) by

- recursively estimating the autocorrelation matrices $\Sigma^2_{\tilde{\mathbf{y}},n}$ and autocorrelation vectors $\Sigma_{\tilde{x}\tilde{\mathbf{y}},n}$, i.e., the estimates $\hat{\Sigma}_{\tilde{\mathbf{y}},k,n}$ and $\hat{\Sigma}_{\tilde{x}\tilde{\mathbf{y}},k,n}$ at time k are obtained by updating the previous estimates using the newly available data at time k:

$$\hat{\Sigma}_{\tilde{\mathbf{y}},k,n} \leftarrow \lambda\hat{\Sigma}_{\tilde{\mathbf{y}},k-1,n} + \tilde{\mathbf{y}}^H_{k,n}\tilde{\mathbf{y}}_{k,n} \tag{23}$$

$$\hat{\Sigma}_{\tilde{x}\tilde{\mathbf{y}},k,n} \leftarrow \lambda\hat{\Sigma}_{\tilde{x}\tilde{\mathbf{y}},k-1,n} + \tilde{x}^*_{k,n}\tilde{\mathbf{y}}_{k,n} \tag{24}$$

  An exponentially weighted window with factor λ (close to 1) has been included to allow for tracking changes in a non-stationary environment;
- evaluating (17) and (19) at $\mathbf{w}_g = \mathbf{w}_{k-1,g}$.

Note that the Hessian approximation $\mathbf{R}_g(\mathbf{w}_g)$ is positive semidefinite, as required to ensure convergence: each tone-dependent contribution in (19), hence

also $\mathbf{R}_g(\mathbf{w}_g)$, is positive semidefinite with one zero-eigenvalue corresponding to the eigenvector $\mathbf{w}_g$. This singularity comes from the scalar ambiguity of (7), which causes all vectors $\gamma\mathbf{w}_g$, with γ an arbitrary real number, to result in the same cost, hence the curvature of the cost function along the direction of $\mathbf{w}_g$ is zero.

If the n-th FEQ output error is given by

$$\tilde{e}_{k,n,\mathbf{w}_g} = \tilde{x}_{k,n} - D_n\tilde{\mathbf{y}}_{k,n}\mathbf{w}_g \tag{25}$$

then the *gradient* (18) is a weighted sum over all tones in $\mathcal{S}_g$ of terms (26):

$$\frac{\partial\mathcal{E}\left\{\left|\tilde{e}_{k,n,\mathbf{w}_g}\right|^2\right\}}{\partial\mathbf{w}_g} = -2\Re\left\{\mathcal{E}\left\{\tilde{\mathbf{y}}_{k,n}^H(D_n^{MMSE})^*(\tilde{x}_{k,n} - D_n^{MMSE}\tilde{\mathbf{y}}_{k,n}\mathbf{w}_g)\right\}\right\} \tag{26}$$

. The tone-dependent weights $\tilde{\alpha}_n$ depend on the PGEQ estimate $\mathbf{w}_{k,g}$ through $\rho^2_{n,\mathbf{w}_g}$. The tone-dependent terms of the *Hessian approximation* (20) are the sum of two contributions. The first contribution, denoted with [1], is expected as it stems from the obvious linear dependence of the FEQ output error (25) on $\mathbf{w}_g$. The second contribution [2] arises from the (nonlinear) dependence of the MMSE FEQs D_n^{MMSE} in (25) on $\mathbf{w}_g$.

This recursive updating algorithm converges very fast (a good estimate $\mathbf{w}_{k,g}$ is obtained after less than 100 updates) and it gives bitrates that are in line with the results obtained with batch optimization of the BM-PGEQ cost function.

A detailed description and complexity analysis of a computationally efficient and numerically stable version of this recursive algorithm are beyond the scope of this paper. Below, we give some rough guidelines on how to reduce its complexity, based on the aforementioned recursive structure (8) and (11) present in the sliding DFT (as explained in detail in the Appendix I, see (.5) and (.6)) *:

$$\tilde{\mathbf{y}}_{k,n} = \left[\begin{array}{c} \tilde{y}_{k,n} \; \Delta\mathbf{y}_k \end{array}\right]\mathbf{P}_n \tag{27}$$

The matrix $\mathbf{P}_n$ is defined in (8). We also give the main steps in computing a BM-PGEQ update (15) and summarize their computational complexity in number of multiplications per update in Table 2.

Here comes Table 2.

* Note that this recursive structure also led to the efficient recursive-least-squares based adaptive per tone equalizer design in [15].

(1) Instead of recursively estimating the autocorrelation matrices $\Sigma^2_{\tilde{\mathbf{y}},n}$ as shown in (23), it is wiser to store the Cholesky factor $\mathbf{Z}_n$ of the autocorrelation matrix of the vector $\left[\begin{array}{cc} \tilde{y}_{k,n} & \Delta\mathbf{y}_k \end{array}\right]$ with its elements in reverse order (see (27)). Because of the reverse ordering and as the $T-1$ real difference terms $\Delta\mathbf{y}_k$ are tone-independent, i.e., common for all tones, the first $T-1$ rows of $\mathbf{Z}_n$ constitute an upper-triangular submatrix that is tone-independent; an update requires $\mathcal{O}((T-1)^2)$ operations. An update of the last (tone-dependent) row of $\mathbf{Z}_n$ for all tones requires roughly $\mathcal{O}(N_aT)$ operations. Instead of estimating the autocorrelation vectors $\Sigma_{\tilde{x}\tilde{\mathbf{y}},n}$ as given in (24), one can recursively estimate $\mathcal{E}\left\{\tilde{x}^*_{k,n}\tilde{y}_{k,n}\right\}$ and $\mathcal{E}\left\{\tilde{x}^*_{k,n}\Delta\mathbf{y}_k\right\}$ for all N_a tones at a cost of $\mathcal{O}(N_a)$ and $\mathcal{O}(N_a(T-1))$ per update, respectively. A *single FFT* per DMT symbol as in (10) is then needed for this efficient updating.

(2) An estimate of $\Sigma^2_{\tilde{\mathbf{y}},n}$, $\Sigma^2_{\tilde{\mathbf{y}},n}\mathbf{w}_g$ and $\mathbf{w}_g^T\Sigma^2_{\tilde{\mathbf{y}},n}\mathbf{w}_g$ for all tones, used for the gradient and the Hessian approximation, is obtained based on the recursion in (27) and the Cholesky factors $\mathbf{Z}_n$, constructed above, requiring roughly $\mathcal{O}(N_aT^2)$ operations:

$$\Sigma^2_{\tilde{\mathbf{y}},n} = \mathbf{P}_n^H\mathbf{J}_n\mathbf{Z}_n^H\mathbf{Z}_n\mathbf{J}_n\mathbf{P}_n \tag{28}$$

where $\mathbf{J}_n$ is the antidiagonal identity matrix that undoes the reverse ordering of $\left[\begin{array}{cc} \tilde{y}_{k,n} & \Delta\mathbf{y}_k \end{array}\right]$ and $\mathbf{P}_n$ is defined in (27). Similarly, a new estimate of $\Sigma_{\tilde{x}\tilde{\mathbf{y}},n}$ (and $\mathbf{w}_g^T\Sigma^H_{\tilde{x}\tilde{\mathbf{y}},n}$), needed for the gradient, is obtained from the statistics $\mathcal{E}\left\{\tilde{x}^*_{k,n}\tilde{y}_{k,n}\right\}$ and $\mathcal{E}\left\{\tilde{x}^*_{k,n}\Delta\mathbf{y}_k\right\}$, obtained in the first step, using the recursion in

$$\Sigma_{\tilde{x}\tilde{\mathbf{y}},n} = \left[\begin{array}{cc} \mathcal{E}\left\{\tilde{x}^*_{k,n}\tilde{y}_{k,n}\right\} & \mathcal{E}\left\{\tilde{x}^*_{k,n}\Delta\mathbf{y}_k\right\} \end{array}\right]\mathbf{P}_n \tag{29}$$

and costs $\mathcal{O}(N_aT)$. Based on these estimates, the tone-dependent scalars $\rho^2_{n,\mathbf{w}_g}$ and α_n are computed, requiring $\mathcal{O}(N_a)$ operations.

(3) Based on the intermediate computations above, a gradient $\mathbf{g}_g(\mathbf{w}_g)$ is computed per group; this gives a cost of $\mathcal{O}(N_aT)$.

(4) Building the estimate of the approximate Hessian $\mathbf{R}_g(\mathbf{w}_g)$ per group in (19) costs $\mathcal{O}(N_aT^2)$ computations. Computing $\mathbf{R}_g^\dagger(\mathbf{w}_g)\mathbf{g}_g(\mathbf{w}_g)$ per group in (15), e.g., based on a Cholesky-like decompositon of the positive semi-definite $\mathbf{R}_g(\mathbf{w}_g)$ [14], requires $\mathcal{O}(N_gT^3)$ operations.

The complexity is independent of the number of groups, except for the pseudo-inverse $\mathbf{R}_g^\dagger(\mathbf{w}_g)$ which is needed for each tone group, hence the cost of $\mathcal{O}(N_gT^3)$. Depending on the FFT size N, the number of taps T and the number of tone groups N_g, the complexity is dominated by either $\mathcal{O}(N_gT^3)$ or $\mathcal{O}(N_aT^2)$. The computational complexity of the RLS-based adaptive PTEQ design of [15] is dominated by a term on the order of $\mathcal{O}(N_aT)$ operations.

A total number of $4N_aT + \frac{(T-1)^2}{2}$ real coefficients need to be stored for the BM-PGEQ adaptation scheme:

- the Cholesky factors $\mathbf{Z}_n$: the tone-independent triangular submatrix requires $\frac{(T-1)^2}{2}$ real memory coefficients; the tone-dependent rows have N_aT complex coefficients;
- the crosscorrelations $\mathcal{E}\left\{\tilde{x}_{k,n}^*\tilde{y}_{k,n}\right\}$ and $\mathcal{E}\left\{\tilde{x}_{k,n}^*\Delta\mathbf{y}_k\right\}$ requiring N_a and $N_a(T-1)$ complex coefficients, respectively.

The RLS-based adaptive PTEQ design of [15] requires a memory of $2N_aT + \frac{(T-1)^2}{2}$ real coefficients.

Whereas the memory requirement of the BM-PGEQ and the PTEQ adaptation scheme are comparable (the BM-PGEQ adaptation scheme requires roughly twice as much memory), the computational load of the BM-PGEQ scheme is considerably larger. As the time-variations in a DSL environment are typically slow, adaptation for tracking purposes during data transmission can be done at a rate that is much slower than the DMT symbol rate, rendering the computational load of the BM-PGEQ adaptation acceptable during data transmission. During connection set-up, equalizer design should be finished within the designated amount of time. In ADSL, a training sequence with cyclic prefix of around 15000 DMT symbols or 3.7 sec is available. When adopting parameter settings for a typical ADSL scenario as suggested in Section 5, i.e., $N_a = 224$ active tones, $N_g = 4$ tone groups and $T = 32$ taps per group or tone, the BM-PGEQ adaptation requires roughly 2.1×10^6 real multiplications per update (the terms $\mathcal{O}(N_aT^2)$ are dominating), whereas the RLS-based PTEQ adaptation [15] requires only 128×10^3 real multiplications. The adaptive BM-PGEQ design complexity is an issue as it is on the order of a factor T larger than the adaptive PTEQ design complexity. Both algorithms need on the order of 100 updates to converge, see Section 5. For the BM-PGEQ, this corresponds to a load of around 2.1×10^8 real multiplications in the given amount of 3.7 sec, hence a feasible figure of about 60×10^6 real multiplications per second during connection set-up.

5 Simulations

In [9], ADSL simulations showed that the BM-TEQ approaches the performance of the PTEQ very closely, even with crosstalk noise and strong front-end filters introducing a significant amount of ISI/ICI. Moreover, the attained local minima of the nonconvex BM-TEQ cost function corresponded to bitrates with a negligible difference. In this paper, we consider even harsher conditions, where the noise also contains narrowband radio frequency interference

(RFI). RFI, especially ingress from AM radio stations, can be an important interferer in ADSL [16]. There exist specific solutions to deal with RFI, e.g., based on *receiver windowing* combined with a cyclic prefix extension [17]. It was shown in [18] that receiver windowing has an equivalent, implicit implementation by means of the per tone equalizer: by using a sufficient number of PTEQ taps, one obtains a receiver that optimizes the equalizer *and* the window in a per tone fashion. In this section, we investigate by simulation whether the same holds true for the BM-TEQ and BM-PGEQ, and compare their performance with the PTEQ, the original tone-grouping (TG) scheme of [11] and the MMSE TEQ. For comparisons between the BM-TEQ and the other methods mentioned in the introduction (e.g. [2], [4],...) we refer to [9].

The simulations below are done for the Carrier-Serving-Area 4 (CSA4) downstream loop (tones 33 to 256) and are representative of simulations with all 8 CSA loops [2,4]. The included front-end filters are moderate and introduce only a small amount of ISI/ICI. The synchronization delay Δ is fixed to 46 for all equalizer designs and lengths, which corresponds to the first sample index of a channel impulse response window of $\nu+1$ samples with maximum energy. The noise includes additive white Gaussian background noise with a power of -140 dBm/Hz, residual echo and near-end crosstalk (NEXT) coming from 24 ADSL disturbers [16]. We consider an RFI scenario with 7 RFIs, taken from [16] where we omitted the out-of-band RFIs, which fold back into the ADSL band through aliasing. The carrier frequencies are 540, 650, 680, 760, 790, 840 and 1080kHz. The first two RFIs have a power of -30dBm, the remaining five have -50dBm. We use the Matlab batch routine `fminunc` (with appropriate parameter settings) from the Optimization Toolbox, and supplement it with gradient (17) and Hessian information (19) to ensure convergence. The BM-TEQ and BM-PGEQ design algorithms are run with 100 random starting points; the worst and best performance over these 100 runs are considered.

Figure 1 shows the bitrate as a function of the number of equalizer taps[**] for the MMSE-TEQ, the BM-TEQ, the original TG scheme and the new BM-PGEQ scheme, both with 7 groups (of 32 tones) and the PTEQ. We also include the PTEQ performance if there is no RFI present (dashed line with asterisks). The latter curve and simulations suggest that only a small number of taps (far less than 24) is needed to equalize the channel; the resulting bitrate without RFI is 8.47 MBps. If RFI is present, the extra taps clearly help to mitigate the RFI: with 16 taps (not shown), the PTEQ only reaches 4.04 MBps, whereas 64 taps per tone result in 7.87 MBps, corresponding to a loss of less than 10 % w.r.t. the bitrate without RFI. Increasing the number of taps even more gives no significant further increase. The presence of AM RFI has

[**]Using T taps in case of the BM-PGEQ and PTEQ means that T taps are used per group or tone. As was shown in Section 3, Table 1, all equalizers (TEQ, PGEQ and PTEQ) have a complexity that is roughly proportional to $O(NT)$.

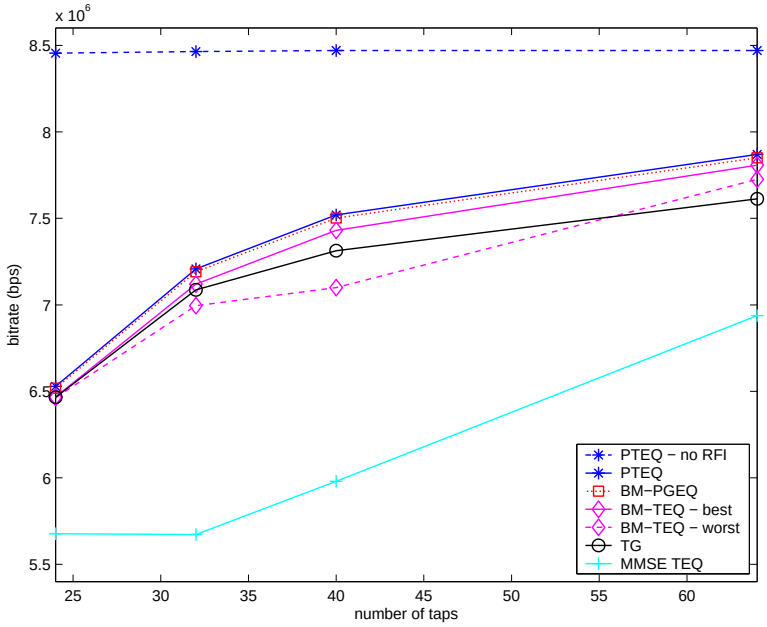

Figure 1. Bitrate as a function of the number of TEQ taps for the downstream CSA4 loop with RFI (the PTEQ performance without RFI is also depicted). The BM-PGEQ and TG scheme use 7 groups of 32 tones. The BM-PGEQ and PTEQ performance (in case of RFI) almost coincide. Depicted number of taps: 24, 32, 40 and 64.

indeed a drastic impact on the achieved bitrate. It increases the performance gap between different equalization solutions, e.g., between the MMSE-TEQ, BM-TEQ and PTEQ. The MMSE-TEQ is clearly outperformed by the other schemes, which was not the case in the absence of RFI [9]. The TG scheme loses up to 257 kbps at 64 taps w.r.t. the PTEQ. Note that this performance would be even worse if an RFI (almost) coincides with a group center tone, as this would destroy the equalization of a whole tone group. Due to the RFIs, the BM-TEQ is more susceptible to less performant local optima: there is a performance gap between the worst and best performing BM-TEQ of 125 kbps for 32 taps, 400 kbps for 40 taps and 85 kbps for 64 taps; without RFIs, this gap was found to be only around 6 kbps. The BM-PGEQ performance almost coincides with the PTEQ performance.

In scenarios with RFI, it is useful to apply the newly presented BM-PGEQ, as this fills the performance gap between BM-TEQ and PTEQ. **Figure 2** shows the bitrate as a function of the number of tone groups (with an equal number of tones and a typical 32 taps per group) for the TG and BM-PGEQ scheme. Note that "1 group" corresponds to the BM-TEQ whereas the (not depicted) PTEQ corresponds to the maximum number of groups (i.e., 224 groups for this downstream loop). This PTEQ performance is 7.203 Mbps, i.e., roughly equal to the depicted BM-PGEQ performance with 14 groups.

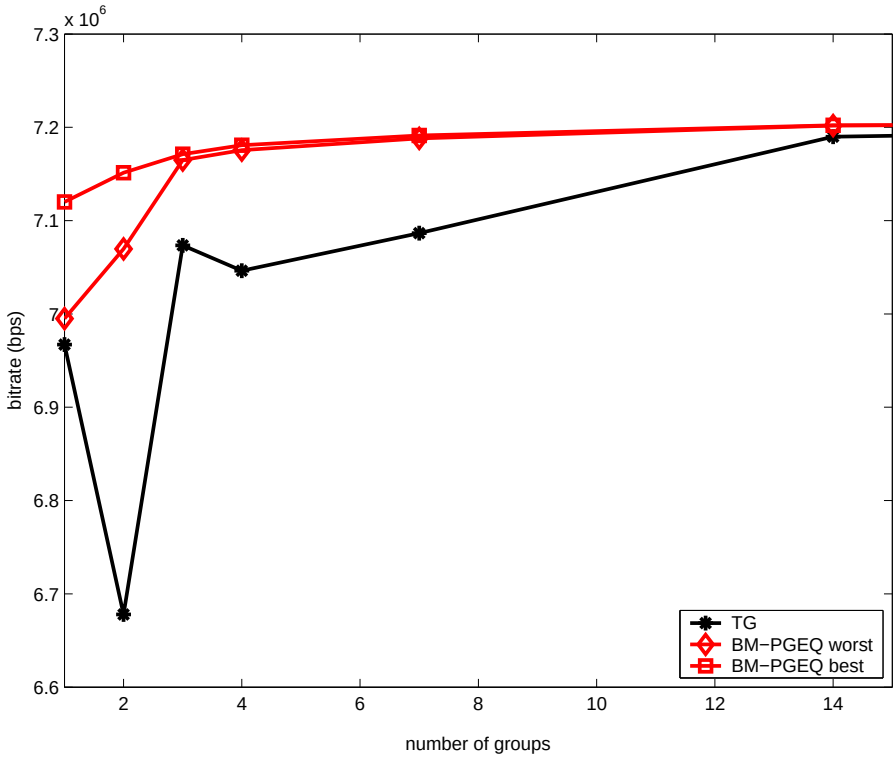

Figure 2. Bitrate as a function of the number of tone groups for the downstream CSA4 loop. A BM-PGEQ with 1 group corresponds to a BM-TEQ. The PTEQ corresponds to a BM-PGEQ with 224 groups and is not depicted; its performance roughly coincides with the depicted BM-PGEQ performance for 14 groups. Depicted number of groups: 1, 2, 3, 4, 7 and 14.

The TG scheme does not guarantee better performance for a larger number of groups: when going from 1 to 2 groups, the bitrate drops significantly with almost 300 kbps. The TG performance depends on the quality of the PTEQ of the center tone as an equalizer for the whole group. In contrast, the BM-PGEQ bitrate does increase with the number of groups. Moreover, as can be seen in Figure 2, the susceptibility to worse performing local optima, reduces with an increasing number of groups (i.e., with smaller groups). Apparently, different equalizers designed for smaller groups correspond to local optima that do not differ much in bitrate. Figure 2 suggests to use at least 3 groups. A BM-PGEQ with 3 groups performs only 50 kbps worse than the PTEQ; a BM-PGEQ with 7 groups further reduces this loss to only 20 kbps.

Figure 3 shows the bitrate convergence when designing a PTEQ using RLS-based adaptation, and a BM-PGEQ with 4 tone groups and a BM-TEQ, both using the adaptation algorithm of Section 4. The exponential weighting factor has been set to $\lambda = 0.999$; the number of taps T is set to 32. Both schemes converge very fast (within a few hundreds of updates): the BM-PGEQ/BM-TEQ adaptation appears faster than the PTEQ adaptation; this is probably due to the fact that the BM-PGEQ/BM-TEQ adaptation estimates the cross-

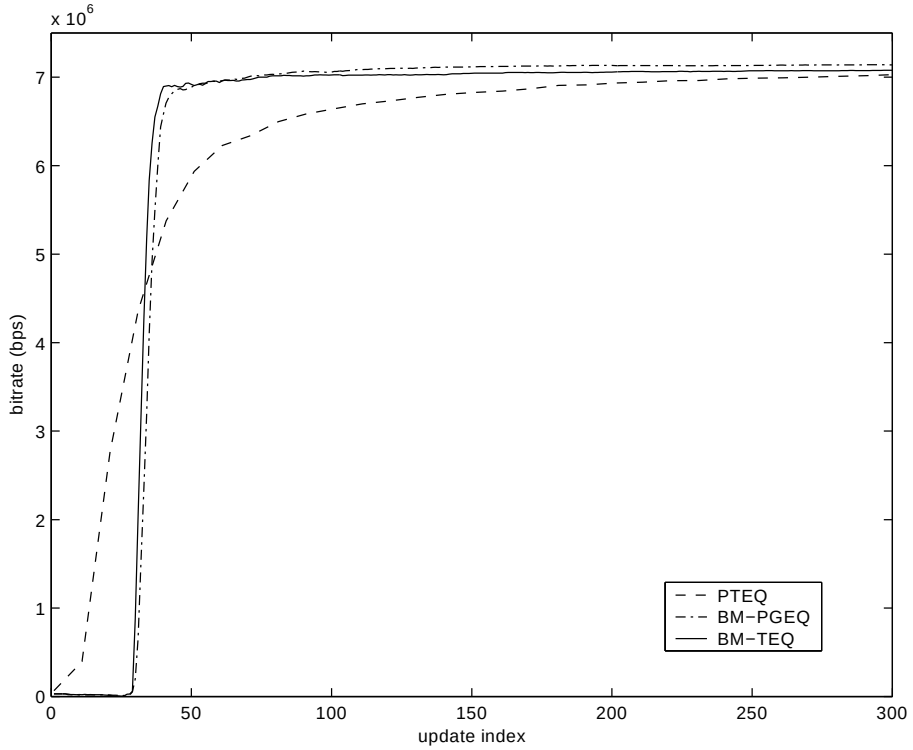

Figure 3. Bitrate convergence curve for the adaptive RLS-based PTEQ design and the adaptive Gauss-Newton based BM-PGEQ (4 tone groups) and BM-TEQ design. $T = 32; \lambda = 0.999$.

correlation vectors $\Sigma_{\tilde{x}\tilde{\mathbf{y}},n}$ recursively, whereas the PTEQ adaptation uses an instantaneous estimate. After complete convergence (e.g., 1000 updates, which is not shown on the plot), the PTEQ (slightly) outperforms the BM-PGEQ and BM-TEQ, as it is expected. Both the BM-PGEQ/BM-TEQ adaptation algorithm and the batch routine result in the same performance. Indeed, a large number of simulation runs showed that the batch routine and adaptation algorithm typically attain the same local optima and bitrates.

Conclusions

In an earlier paper [9], we proposed a *bitrate maximizing* design criterion for time-domain equalizers (BM-TEQ) in DMT transceivers to shorten the channel impulse response. In this paper, we have shown how the BM-TEQ design can be used in a bitrate maximizing "per group" equalization scheme (BM-PGEQ): the active tones are divided in groups and each group is provided with a bitrate maximizing equalizer. This BM-PGEQ design allows for a trade-off between memory requirement and performance, keeping computational complexity during data transmission roughly at the same level. We have shown that the BM-PGEQ scheme encompasses the BM-TEQ design and the so-called per tone equalization scheme (PTEQ) as extreme cases. We developed an adaptive BM-PGEQ training algorithm, which can also be

used to track slow changes in channel and noise. Through simulation, we have shown that the BM-PGEQ scheme outperforms an earlier presented tone grouping scheme where the whole tone group was assigned the PTEQ of the group center tone. The BM-PGEQ scheme is a useful intermediate between BM-TEQ and PTEQ in an ADSL scenario, even in harsh environments with narrowband interference: the BM-PGEQ scheme is less susceptible than a BM-TEQ to worse performing local optima that are inherently present in a nonlinear, nonconvex optimization context. The BM-PGEQ scheme with as few as 3 to 4 tone groups, requiring only N_gT equalizer taps (e.g., $N_g \times T = 4 \times 32 = 128$), closely approaches the performance of the PTEQ with N_aT taps (e.g., $N_a \times T = 220 \times 32 = 7060$). The drawback of using an adaptive BM-PGEQ scheme instead of a PTEQ is its larger equalizer design complexity. However, we pointed out that the adaptive BM-PGEQ equalizer design requires a feasible amount of around 60×10^6 real multiplications per second during connection set-up.

Algorithm		Computational complexity	Memory requirements
real TEQ	TEQ	NT	T
TEQ	FEQ	$4N_a$	$2N_a$
TEQ	Total	$NT + N_a$	$T + 2N_a$
real PGEQ (min. memory)	PGEQ	$(6T - 4)N_a$	$N_g T$
	FEQ	$4N_a$	$2N_a T$
	Total	$6N_a T$	$N_g T + 2N_a$
real PGEQ (min. computations)	Total	$(2T + 2)N_a$	$2N_a T$
real PTEQ (min. memory)	PTEQ	$(6T - 4)N_a$	$N_a T$
(see Section 3)	FEQ	$4N_a$	$2N_a$
	Total	$6N_a T$	$(T + 2)N_a$
complex PTEQ	Total	$(2T + 2)N_a$	$2N_a T$

Table 1
Memory requirement (number of real coefficients) and computational complexity (number of real multiplications per DMT symbol) of TEQ, BM-PGEQ and PTEQ-based receivers. All receivers need 1 FFT per DMT symbol.

	Step	Complexity
1.	Update of the statistics for all tones	
	Cholesky factor $\mathbf{Z}_n$	1 FFT + $\mathcal{O}((T-1)^2) + \mathcal{O}(N_a T)$
	$\mathcal{E}\left\{\tilde{x}_{k,n}^* \tilde{y}_{k,n}\right\}$ and $\mathcal{E}\left\{\tilde{x}_{k,n}^* \Delta\mathbf{y}_k\right\}$	$\mathcal{O}(N_a) + \mathcal{O}(N_a(T-1))$
2.	Intermediate computations for all tones	
	$\Sigma^2_{\tilde{\mathbf{y}},n}$, $\Sigma^2_{\tilde{\mathbf{y}},n}\mathbf{w}_g$ and $\mathbf{w}_g^T \Sigma^2_{\tilde{\mathbf{y}},n}\mathbf{w}_g$	$\mathcal{O}(N_a T^2)$+$\mathcal{O}(N_a T)$
	$\Sigma^H_{\tilde{x}\tilde{\mathbf{y}},n}$ and $\mathbf{w}_g^T \Sigma^H_{\tilde{x}\tilde{\mathbf{y}},n}$	both $\mathcal{O}(N_a T)$
	$\rho^2_{n,\mathbf{w}_g}$and α_n	both $\mathcal{O}(N_a)$
3.	Gradient per group $\mathbf{g}_g(\mathbf{w}_g)$	$\mathcal{O}(N_a T)$
4.	Hessian approximation per group $\mathbf{R}_g(\mathbf{w}_g)$	$\mathcal{O}(N_a T^2)$
	$\mathbf{R}_g^\dagger(\mathbf{w}_g)\mathbf{g}_g(\mathbf{w}_g)$	$\mathcal{O}(N_g T^3)$

Table 2
Computational complexity in number of multiplications of an update (15) in the PGEQ adaptation.

Koen Vanbleu is a Research Assistant with the F.W.O. Vlaanderen. Geert Ysebaert and Gert Cuypers are Research Assistants with the I.W.T. This research work was carried out at the ESAT laboratory of the Katholieke Universiteit Leuven, in the frame of the Belgian State, Prime Minister's Office - Federal Office for Scientific, Technical and Cultural Affairs - Interuniversity Poles of Attraction Programme (2002-2007) - IUAP P5/22 ('Dynamical Systems and Control: Computation, Identification and Modelling') and P5/11 ('Mobile multimedia communication systems and networks'), the Concerted Research Action GOA-MEFISTO-666 (Mathematical Engineering for Information and Communication Systems Technology) of the Flemish Government, Research Project FWO nr.G.0196.02 ('Design of efficient communication techniques for wireless time-dispersive multi-user MIMO systems') and was partially sponsored by Alcatel-Bell and Alcatel-MicroElectronics. The scientific responsibility is assumed by its authors.

Appendix I

This appendix describes an efficient alternative computation of the n-th DFT output. This alternative is used in the derivation of the BM-TEQ cost function in Section 2, as well as in the efficient implementation of the BM-PGEQ scheme presented in Section 3 and the BM-PGEQ adaptation algorithm of Section 4. Note that this alternative has also led to the efficient PTEQ structure, described in [10].

The n-th DFT output can be obtained in 2 ways:

$$\textbf{[1]}\ \tilde{y}_{k,n,\mathbf{w}} = \mathcal{F}_n(\mathbf{Y}_k\mathbf{w}) \tag{.1}$$

$$\textbf{[2]}\ \tilde{y}_{k,n,\mathbf{w}} = (\mathcal{F}_n\mathbf{Y}_k)\mathbf{w} = \tilde{\mathbf{y}}_{k,n}\mathbf{w} \tag{.2}$$

i.e.,

[1] as the n-th coefficient of the DFT of the TEQ output $\mathbf{Y}_k\mathbf{w}$ (which is the convolution of the TEQ $\mathbf{w}$ and the k-th received DMT symbol $\mathbf{y}_{k,-T+1:N-1}$),

$$\mathbf{Y}_k\mathbf{w} = \begin{bmatrix} y_{k,0} & \cdots & y_{k,-T+1} \\ \vdots & \ddots & \vdots \\ y_{k,N-1} & \cdots & y_{k,N-T} \end{bmatrix} \mathbf{w} \tag{.3}$$

[2] or, alternatively, as a linear combination $\mathbf{w}$ of $\tilde{\mathbf{y}}_{k,n} = \mathcal{F}_n\mathbf{Y}_k$, i.e., the outputs on tone n of a sliding DFT applied to the k-th unequalized, received DMT symbol $\mathbf{y}_{k,-T+1:N-1}$:

$$(\bar{\tilde{\mathbf{y}}}_{k,n})^T = \underbrace{\begin{bmatrix} \boxed{\mathcal{F}_n} & 0 & \cdots & 0 \\ 0 & \boxed{\mathcal{F}_n} & \cdots & 0 \\ & \ddots & \ddots & \ddots \\ 0 & \cdots & 0 & \boxed{\mathcal{F}_n} \end{bmatrix}}_{\mathbf{F}_n} \begin{bmatrix} y_{k,-T+1} \\ \vdots \\ y_{k,N-1} \end{bmatrix} \tag{.4}$$

where $\bar{\tilde{\mathbf{y}}}_{k,n}$ (with upper bar) is a $1 \times T$ vector with the elements of the sliding DFT output $\tilde{\mathbf{y}}_{k,n}$ in reverse order.

This sliding DFT (.4) can be computed efficiently (without requiring T consecutive DFTs). The t-th entry of $\tilde{\mathbf{y}}_{k,n}[t] = \mathcal{F}_n \mathbf{Y}_k[:,t]$ (where $\mathbf{Y}_k[:,t]$ is the t-th column of $\mathbf{Y}_k$), obeys the following recursion, by exploiting the Toeplitz structure of $\mathbf{Y}_k$ in (.3) or, equivalently, the Toeplitz structure of $\mathbf{F}_n$ in (.4):

$$\tilde{\mathbf{y}}_{k,n}[t+1] = \alpha_n \tilde{\mathbf{y}}_{k,n}[t] + \underbrace{(y_{k,-t} - y_{k,N-t})}_{\Delta y_{k,-t}}, \; t = 1, \cdots, T-1 \tag{.5}$$

where $\alpha_n = \exp(-j2\pi(n-1)/N)/\sqrt{N}$. From (.5), it follows that

$$\tilde{\mathbf{y}}_{k,n} = \begin{bmatrix} \tilde{y}_{k,n} & \Delta \mathbf{y}_k \end{bmatrix} \underbrace{\begin{bmatrix} 1 & \alpha_n & \cdots & \alpha_n^{T-1} \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \alpha_n \\ 0 & \cdots & 0 & 1 \end{bmatrix}}_{\mathbf{P}_n} \tag{.6}$$

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

$$\Delta \mathbf{y}_k = \begin{bmatrix} (y_{k,-1} - y_{k,N-1}) & \cdots & (y_{k,-T+1} - y_{k,-T+N+1}) \end{bmatrix} \tag{.7}$$

and the n-th output of the DFT/FFT of signal samples $\mathbf{y}_{k,0:N-1}$:

$$\tilde{y}_{k,n} = \tilde{\mathbf{y}}_{k,n}[1] = \mathcal{F}_n \mathbf{y}_{k,0:N-1}. \tag{.8}$$

An efficient computation of $\tilde{\mathbf{y}}_{k,n}$ for *all* N_a active tones n in $\mathcal{S}_a$ then requires 1 FFT (.8), $T-1$ differences (.7) and $N_a(T-1)$ recursions (.5).

Appendix II

The Gauss-Newton-like recursive updating algorithm we devise for solving the BM-PGEQ cost function is based on an estimate of the gradient and the Hessian of the cost function. In this appendix, we derive this gradient and Hessian. Both derivations start from the BM-PGEQ cost function (7), rewritten as a function of the squared normalized correlation $\rho^2_{n,\mathbf{w}_g}$ using (22):

$$\begin{aligned}\arg\max_{\mathbf{w}_g} b_{g,DMT} &= \arg\min_{\mathbf{w}_g} \sum_{n\in\mathcal{S}_g} \log\left(\frac{\mathbf{w}_g^T\mathbf{B}_n\mathbf{w}_g}{\mathbf{w}_g^T\mathbf{A}_n\mathbf{w}_g}\right) \\ &\arg\min_{\mathbf{w}_g} \sum_{n\in\mathcal{S}_g} \log\left(\frac{\Gamma_n(1-\rho^2_{n,\mathbf{w}_g})}{\Gamma_n+(1-\Gamma_n)\rho^2_{n,\mathbf{w}_g}}\right) \end{aligned} \tag{.9}$$

The *gradient* (w.r.t. $\mathbf{w}_g$) of the BM-PGEQ cost function (.9) can be written compactly as

$$\frac{\partial b_{g,DMT}}{\partial \mathbf{w}_g} = -\sum_{n\in\mathcal{S}_g} \beta_{n,\mathbf{w}_g} \frac{\partial \rho^2_{n,\mathbf{w}_g}}{\partial \mathbf{w}_g} \tag{.10}$$

where $\beta_{n,\mathbf{w}_g} = \frac{1}{(\Gamma_n+(1-\Gamma_n)\rho^2_{n,\mathbf{w}_g})(1-\rho^2_{n,\mathbf{w}_g})}$ is a function of $\rho^2_{n,\mathbf{w}_g}$ and where the gradient of $\rho^2_{n,\mathbf{w}_g}$ is easily shown to be equal to:

$$\begin{aligned}\frac{\partial \rho^2_{n,\mathbf{w}_g}}{\partial \mathbf{w}_g} &= 2\Re\left\{\frac{\Sigma^H_{\tilde{x}\tilde{\mathbf{y}},n}\Sigma_{\tilde{x}\tilde{\mathbf{y}},n}\mathbf{w}_g}{\sigma^2_{\tilde{x},n}(\mathbf{w}_g^T\Sigma^2_{\tilde{\mathbf{y}},n}\mathbf{w}_g)} - \frac{\Sigma^2_{\tilde{\mathbf{y}},n}\mathbf{w}_g\left|\Sigma_{\tilde{x}\tilde{\mathbf{y}},n}\mathbf{w}_g\right|^2}{\left(\mathbf{w}_g^T\Sigma^2_{\tilde{\mathbf{y}},n}\mathbf{w}_g\right)^2\sigma^2_{\tilde{x},n}}\right\} \\ &= 2\rho^2_{n,\mathbf{w}_g}\Re\left\{\frac{\Sigma^H_{\tilde{x}\tilde{\mathbf{y}},n}}{\mathbf{w}_g^T\Sigma^H_{\tilde{x}\tilde{\mathbf{y}},n}} - \frac{\Sigma^2_{\tilde{\mathbf{y}},n}\mathbf{w}_g}{\mathbf{w}_g^T\Sigma^2_{\tilde{\mathbf{y}},n}\mathbf{w}_g}\right\}\end{aligned} \tag{.11}$$

Combining (.10), (.11) and $\alpha_{n,\mathbf{w}_g} \triangleq \beta_{n,\mathbf{w}_g}\rho^2_{n,\mathbf{w}_g}$ gives the gradient in (17):

$$\mathbf{g}_g(\mathbf{w}_g) = -\sum_{n\in\mathcal{S}_g} 2\alpha_{n,\mathbf{w}_g}\Re\left\{\frac{\Sigma^H_{\tilde{x}\tilde{\mathbf{y}},n}}{\mathbf{w}_g^T\Sigma^H_{\tilde{x}\tilde{\mathbf{y}},n}} - \frac{\Sigma^2_{\tilde{\mathbf{y}},n}\mathbf{w}_g}{\mathbf{w}_g^T\Sigma^2_{\tilde{\mathbf{y}},n}\mathbf{w}_g}\right\} \tag{.12}$$

Introducing the MMSE FEQ in (.12)

$$D_n^{MMSE} = \frac{\mathcal{E}\left\{\tilde{y}^*_{k,n,\mathbf{w}_g}\tilde{x}_{k,n}\right\}}{\mathcal{E}\left\{\left|\tilde{y}_{k,n,\mathbf{w}_g}\right|^2\right\}} = \frac{\mathbf{w}_g\Sigma^H_{\tilde{x}\tilde{\mathbf{y}},n}}{\mathbf{w}_g^T\Sigma^2_{\tilde{\mathbf{y}},n}\mathbf{w}_g} \tag{.13}$$

and $\tilde{\alpha}_{n,\mathbf{w}_g} \triangleq \alpha_{n,\mathbf{w}_g}/(\rho^2_{n,\mathbf{w}_g}\sigma^2_{\tilde{x},n})$ gives the alternative expression (18) for the

gradient

$$\mathbf{g}_g(\mathbf{w}_g) = -\sum_{n\in\mathcal{S}_g} 2\tilde{\alpha}_{n,\mathbf{w}_g}\Re\left\{\mathcal{E}\left\{\tilde{\mathbf{y}}_{k,n}^H (D_n^{MMSE})^*(\tilde{x}_{k,n} - D_n^{MMSE}\tilde{\mathbf{y}}_{k,n}\mathbf{w}_g)\right\}\right\} \quad (.14)$$

The *Hessian* of the BM-PGEQ cost function (.9) can also be written as a function of $\rho^2_{n,\mathbf{w}_g}$ and its partial derivatives. From (.10), it follows that

$$\mathbf{H}_g(\mathbf{w}_g) = \frac{\partial^2 b_{g,DMT}}{\partial \mathbf{w}_g^2} = -\sum_{n\in\mathcal{S}_g}\beta_n\frac{\partial^2\rho^2_{n,\mathbf{w}_g}}{\partial\mathbf{w}_g^2} - \sum_{n\in\mathcal{S}_g}\gamma_n\left(\frac{\partial\rho^2_{n,\mathbf{w}_g}}{\partial\mathbf{w}_g}\right)\left(\frac{\partial\rho^2_{n,\mathbf{w}_g}}{\partial\mathbf{w}_g^T}\right) \quad (.15)$$

where $\beta_{n,\mathbf{w}_g}$ is defined as above and $\gamma_{n,\mathbf{w}_g} \triangleq \frac{\partial\beta_{n,\mathbf{w}_g}}{\partial\rho^2_{n,\mathbf{w}_g}}$. The gradient and Hessian determine the *Newton direction* $-\mathbf{H}_g^{-1}(\mathbf{w}_g)\mathbf{g}_g(\mathbf{w}_g)$. So called "Newton methods" are optimization algorithms, based on the Newton direction, that typically give much better performance than gradient-based methods. However, in order to assure convergence, the Hessian $\mathbf{H}_g(\mathbf{w}_g)$ should be positive (semi)definite, which does not necessarily hold true. Therefore, it is often approximated by a guaranteed positive (semi)definite matrix. An often applied approximation results in the so-called Gauss-Newton method [12], which is also our method of choice. When rearranging $\frac{\partial\rho^2_{n,\mathbf{w}_g}}{\partial\mathbf{w}_g}$ in (.11):

$$\frac{\partial\rho^2_{n,\mathbf{w}_g}}{\partial\mathbf{w}_g} = \frac{2}{\sigma^2_{\tilde{x},n}}\Re\left\{\mathcal{E}\left\{\tilde{\mathbf{y}}_{k,n}^H D_n^{MMSE*}\underbrace{(\tilde{x}_{k,n} - D_n^{MMSE}\tilde{\mathbf{y}}_{k,n}\mathbf{w}_g)}_{\tilde{e}^{MMSE}_{k,n,\mathbf{w}_g}}\right\}\right\} \quad (.16)$$

where D_n^{MMSE} denotes the MMSE FEQ as defined in (.13) and $\tilde{e}^{MMSE}_{k,n,\mathbf{w}_g}$ is the corresponding residual MMSE FEQ output error, and when using the partial derivative $\frac{\partial D_n^{MMSE}}{\partial\mathbf{w}_g}$, the second order partial derivative $\frac{\partial^2\rho^2_{n,\mathbf{w}_g}}{\partial\mathbf{w}_g^2}$ in (.15) can be expressed as:

$$\begin{aligned}\frac{\partial\rho^2_{n,\mathbf{w}_g}}{\partial\mathbf{w}_g^2} = -2\rho^2_{n,\mathbf{w}_g}\Re\Bigg\{&\frac{\Sigma^2_{\tilde{\mathbf{y}},n}}{\mathcal{E}\left\{\left|\tilde{y}_{k,n,\mathbf{w}_g}\right|^2\right\}} - \frac{\Sigma^2_{\tilde{\mathbf{y}},n}\mathbf{w}_g\mathbf{w}_g^T\Sigma^2_{\tilde{\mathbf{y}},n}}{\mathcal{E}\left\{\left|\tilde{y}_{k,n,\mathbf{w}_g}\right|^2\right\}^2} \\ &- \frac{\Sigma^2_{\tilde{\mathbf{y}},n}\mathbf{w}_g\mathcal{E}\left\{\tilde{e}^{MMSE}_{k,n,\mathbf{w}_g}\tilde{\mathbf{y}}^*_{k,n}\right\}}{\mathcal{E}\left\{\left|\tilde{y}_{k,n,\mathbf{w}_g}\right|^2\right\}\mathcal{E}\left\{\tilde{y}^*_{n,k,\mathbf{w}_g}\tilde{x}_n\right\}} - \frac{\mathcal{E}\left\{\tilde{e}^{MMSE}_{k,n,\mathbf{w}_g}\tilde{\mathbf{y}}^H_{k,n}\right\}\mathbf{w}_g^T\Sigma^2_{\tilde{\mathbf{y}},n}}{\mathcal{E}\left\{\left|\tilde{y}_{k,n,\mathbf{w}_g}\right|^2\right\}\mathcal{E}\left\{\tilde{y}^*_{n,k,\mathbf{w}_g}\tilde{x}_n\right\}} \\ &- \frac{\mathcal{E}\left\{\tilde{e}^{MMSE}_{k,n,\mathbf{w}_g}\tilde{\mathbf{y}}^H_{k,n}\right\}\mathcal{E}\left\{\tilde{e}^{MMSE}_{k,n,\mathbf{w}_g}\tilde{\mathbf{y}}^*_{k,n}\right\}}{\left|\mathcal{E}\left\{\tilde{y}^*_{n,k,\mathbf{w}_g}\tilde{x}_n\right\}\right|^2}\Bigg\}\end{aligned} \quad (.17)$$

The Gauss-Newton method exploits the fact that, around the optimum solution $\mathbf{w}_g^{opt}$, the errors $\tilde{e}_{k,n,\mathbf{w}_g}^{MMSE}$ are expected to become small w.r.t. the data entries of $\tilde{\mathbf{y}}_{k,n}$. The last three terms in (.17) as well as the second summation term in (.15) contain correlations with $\tilde{e}_{k,n,\mathbf{w}_g}^{MMSE}$, e.g., $\mathcal{E}\left\{\tilde{e}_{k,n,\mathbf{w}_g}^{MMSE}\tilde{\mathbf{y}}_{k,n}^H\right\}$, whereas the first two terms in (.17) contain the sliding DFT autocorrelation matrix $\Sigma_{\tilde{\mathbf{y}},n}^2$. These two terms in (.17) are expected to be dominant w.r.t. all the other terms. Moreover, it can be shown that those two terms construct a positive semidefinite matrix (with 1 zero eigenvalue and corresponding eigenvector $\mathbf{w}_g$). Hence we propose the following positive semidefinite approximation of $\mathbf{H}_g(\mathbf{w}_g)$ in (.15):

$$\mathbf{R}_g(\mathbf{w}_g) = \sum_{n\in\mathcal{S}_g} 2\alpha_{n,\mathbf{w}_g}\Re\left\{\frac{\Sigma_{\tilde{\mathbf{y}},n}^2}{\mathbf{w}_g^T\Sigma_{\tilde{\mathbf{y}},n}^2\mathbf{w}_g} - \frac{\Sigma_{\tilde{\mathbf{y}},n}^2\mathbf{w}_g\mathbf{w}_g^T\Sigma_{\tilde{\mathbf{y}},n}^2}{\left(\mathbf{w}_g^T\Sigma_{\tilde{\mathbf{y}},n}^2\mathbf{w}_g\right)^2}\right\} \tag{.18}$$

where $\alpha_{n,\mathbf{w}_g} \triangleq \beta_{n,\mathbf{w}_g}\rho_{n,\mathbf{w}_g}^2$. The alternative expression 20, based on the MMSE FEQs (.13) and $\tilde{\alpha}_{n,\mathbf{w}_g} \triangleq \alpha_{n,\mathbf{w}_g}/(\rho_{n,\mathbf{w}_g}^2\sigma_{\tilde{x},n}^2)$ is then:

$$\begin{aligned}\mathbf{R}_g(\mathbf{w}_g) = \sum_{n\in\mathcal{S}_g} 2\tilde{\alpha}_{n,\mathbf{w}_g}\Re\Big\{&\mathcal{E}\left\{\left[\tilde{\mathbf{y}}_{k,n}^H(D_n^{MMSE})^*\right]\left[D_n^{MMSE}\tilde{\mathbf{y}}_{k,n}\right]\right\} \\ &-\frac{\mathcal{E}\left\{\left[\tilde{\mathbf{y}}_{k,n}^H(D_n^{MMSE})^*\right]\tilde{\mathbf{y}}_{k,n}\right\}\mathbf{w}_g\mathbf{w}_g^T\mathcal{E}\left\{\tilde{\mathbf{y}}_{k,n}^H\left[D_n^{MMSE}\tilde{\mathbf{y}}_{k,n}\right]\right\}}{\mathbf{w}_g^T\Sigma_{\tilde{\mathbf{y}},n}^2\mathbf{w}_g}\Big\}\end{aligned} \tag{.19}$$

References

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

[2] N. Al-Dhahir, J. M. Cioffi, Efficiently computed reduced-parameter input-aided MMSE equalizers for ML detection: A unified approach, IEEE Trans. Information Theory 42 (3) (1996) 903–915.

[3] B. M. D. Farhang-Boroujeny, Design methods for time-domain equalizers in dmt transceivers, IEEE Trans. Communications 49 (3) (2001) 554–562.

[4] G. Arslan, B. L. Evans, S. Kiaei, Equalization for discrete multitone transceivers to maximize bit rate, IEEE Trans. Signal Processing 49 (12) (2001) 3123–3135.

[5] P. J. W. Melsa, R. C. Younce, C. E. Rohrs, Impulse response shortening for discrete multitone transceivers, IEEE Trans. Communications 44 (12) (1996) 1662–1672.

[6] N. Al-Dhahir, J. M. Cioffi, Optimum finite-length equalization for multicarrier transceivers, IEEE Trans. Communications 44 (1) (1996) 56–64.

[7] W. Henkel, T. Kessler, Maximizing the channel capacity of multicarrier transmission by suitable adaptation of the time-domain equalizer, IEEE Trans. Communications 48 (12) (2000) 2000–2004.

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

[9] K. Vanbleu, G. Ysebaert, G. Cuypers, M. Moonen, K. Van Acker, Bitrate maximizing time-domain equalizer design for DMT-based systems, IEEE Trans. Communications 52 (6) (2004) 871–876.

[10] K. Van Acker, G. Leus, M. Moonen, O. van de Wiel, T. Pollet, Per-tone equalization for DMT-based systems, IEEE Trans. Communications 49 (1) (2001) 109–119.

[11] K. Van Acker, G. Leus, M. Moonen, T. Pollet, Frequency domain equalization with tone grouping in DMT/ADSL-receivers, in: Proc. Asilomar Conf. on Signals, Systems and Computers, Pacific Grove, California, 1999, pp. 1067–1070.

[12] L. Ljung, T. Söderström, Theory and Practice of Recursive Identification, Cambridge, MA: MIT Press, 1983.

[13] K. Vanbleu, G. Ysebaert, G. Cuypers, M. Moonen, On time-domain and frequency-domain MMSE-based TEQ design for DMT transmission, IEEE Trans. Signal Processing 53 (8) (2005) 3311–3324.

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

[15] K. Van Acker, G. Leus, M. Moonen, T. Pollet, RLS-based initialization for per-tone equalizers in DMT receivers, IEEE Trans. Communications 51 (6) (2003) 885–889.

[16] International Telecommunications Union (ITU), Recommendation G.test.bis. Test procedures dfor digital subscriber line (DSL) transceivers (Apr. 2000).

[17] P. Spruyt, P. Reusens, S. Braet, Performance of improved DMT transceiver for VDSL, ANSI T1E1.4 Committee contribution 96-104, Colorado Springs (Apr. 1996).

[18] K. Van Acker, T. Pollet, G. Leus, M. Moonen, Combination of per tone equalization and windowing in DMT-receivers, Signal Processing 81 (8) (2001) 1571–1579.

List of Figures

1 Bitrate as a function of the number of TEQ taps for the downstream CSA4 loop with RFI (the PTEQ performance without RFI is also depicted). The BM-PGEQ and TG scheme use 7 groups of 32 tones. The BM-PGEQ and PTEQ performance (in case of RFI) almost coincide. Depicted number of taps: 24, 32, 40 and 64. 16

2 Bitrate as a function of the number of tone groups for the downstream CSA4 loop. A BM-PGEQ with 1 group corresponds to a BM-TEQ. The PTEQ corresponds to a BM-PGEQ with 224 groups and is not depicted; its performance roughly coincides with the depicted BM-PGEQ performance for 14 groups. Depicted number of groups: 1, 2, 3, 4, 7 and 14. 17

3 Bitrate convergence curve for the adaptive RLS-based PTEQ design and the adaptive Gauss-Newton based BM-PGEQ (4 tone groups) and BM-TEQ design. $T = 32;\lambda = 0.999$. 18