TURBO EQUALIZATION OF NON-LINEAR COMMUNICATION CHANNELS[*]

L. Szczeciński, R. Elharti, S. Affes
INRS-Telecommunications
Place Bonaventure, 800, de la Gauchetière O Suite 6900
Montréal, QC H5A 1K6, CANADA
{leszek, elharti, affes}@inrs-telecom.uquebec.ca

***Abstract* - The demand for inexpensive and reliable communications stimulates development of digital signal processing techniques which may improve the quality of communication links. A significant interest has aroused recently in the area of turbo equalization. This technique combines the decoding and equalization blocks in a structure which outperforms a traditional approach treating them separately. This paper will extend the known results to the domain of equalization of channels with non-linear distortions; these effects may be important if power-limited and/or inexpensive RF amplifiers are used. Reduced complexity MAP equalizer and max-log MAP channel decoder are employed in this study based on the synthetic data generated using non linear models. The preliminary result show that a significant gain may be obtained if non-linear channel models are used.**

I. INTRODUCTION

Communication systems employ various functional blocks in order to ensure reliable data transmission. In particular the decoder aims at removing errors using the redundant information introduced at the transmitter via channel coding, and the equalizer employs signal processing techniques to diminish the intersymbol interference (ISI), caused by propagation in multipath channel and/or by the partial-response signaling. For simplicity, coding and equalization techniques are usually considered separately leading to receiver's sub-optimality.

Recently both techniques were considered for combined processing in the so-called turbo equalization [1]. This iterative technique allows the equalizer and decoder to exchange information and improves the overall performance in terms of Bit-Error- or Block-Error Rates (BER or BLER). Turbo-processing received considerable attention from the research community and many variants and simplifications of turbo-equalizers have been presented in the literature [2][3][4][5].

Majority of signal processing techniques were developed for linear channel models due to their simplicity and tractability. However, if strong non-linear distortions are present - e.g. due to high-power amplifiers (HPAs) in satellite or mobile communication transmitters [6] - they should be incorporated into the channel model. Various methods has been proposed to mitigate the non-linear distortions introduced by HPAs. They may be applied at the transmitter for signal pre-distortion [7] or at the receiver for non-linear channel equalization. The advantage of non-linear modeling has been shown for example in [8], however results in the context of turbo-equalization of non-linear channels have not yet been reported.

This paper analyzes joint decoding and non-linear equalization, i.e. non-linear turbo equalization. The main objective is to identify the possible gain achievable by a non-linear turbo equalizer (called herein NTEQ), when compared to the approach based on a linear channel model (LEQ). The first is based the non-linear model of the channel, the latter supposes that linear model is adequate and may be identified using standard LS method.

This paper has the following structure: first, the system model is presented and next the turbo-equalization algorithms are defined along with the proposition of an algorithm for non-linear channel identification. The simulation setup and the results are described and the conclusions are drawn.

II. SYSTEM MODEL

The simplified digital transmission model is shown in Fig. 1.

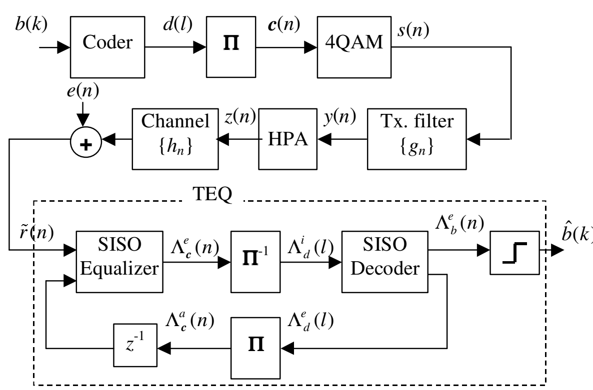

Fig. 1 Model of a digital transmission and turbo-equalization.

The coded bits $c(k)$ are obtained by convolutional coding and interleaving of N_b=500 information bits $b(k)$. $N_{payload}$ symbols $s(n)$ from power-normalized ($E\left[|s(n)|^2\right]=1$) 4QAM constellation (each defined by two bits $\boldsymbol{c}(n)=[c_0(n),c_1(n)]$), are sent in bursts, whose structure is shown in Fig. 2;

guard	training	payload	guard

Fig. 2 Burst structure of the signal $s(n)$ in Fig. 1.

guard symbols are introduced to eliminate the effect of adjacent frames; N_{train} training symbols are used for channel

[*] This work was supported by the INRS-Telecommunications startup fund.

estimation. The burst is transmitted through a pulse-shaping filter with a power-normalized impulse response

$$\{g_k\}=\alpha_0\{\alpha_g,1,\alpha_g\}/(1+2\alpha_g^2),\quad \alpha_g=0.5 \tag{1}$$

where α_g controls the amount of ISI introduced and α_0 fixes the operating point of the HPA. The resulting low-power signal $y(n)$ is amplified by a memoryless High-Power Amplifier (HPA), characterized by the following AM-AM and AM-PM conversions [6] :

$$\begin{aligned} &z(n)=f_{HPA}(y(n))=A(|y(n)|)\cdot\exp\{j\Phi(|y(n)|)+j\arg\{y(n)\}\} \\ &A(r)=\frac{\alpha_A r}{1+\beta_A r^2},\ \Phi(r)=\frac{\alpha_\Phi r^2}{1+\beta_\Phi r^2} \end{aligned} \tag{2}$$

where $\alpha_\Phi=\pi/3$ and $\beta_\Phi=\beta_A=1$ are chosen following suggestion of [6]. Depending on the value of α_0, the parameter α_A is chosen to normalize the amplifier output power:

$$E\left[|z(n)|^2\right]=1 \tag{3}$$

The parameter called input backoff (*IBO*):

$$IBO=10\log_{10}\left(\frac{A_{sat}^2}{E\left[|y(n)|^2\right]}\right) \tag{4}$$

characterizes the efficiency of the amplifier and its operating point. Because $A_{sat}=1$ (the amplitude of the input signal for which the HPA saturates) and taking into account (1), the relationship between IBO and α_0 is :

$$IBO=-20\log_{10}(\alpha_0) \tag{5}$$

Note that for high *IBO* values, the amplifier is functioning in a quasi-linear zone; this effect is well illustrated in Fig. 3.

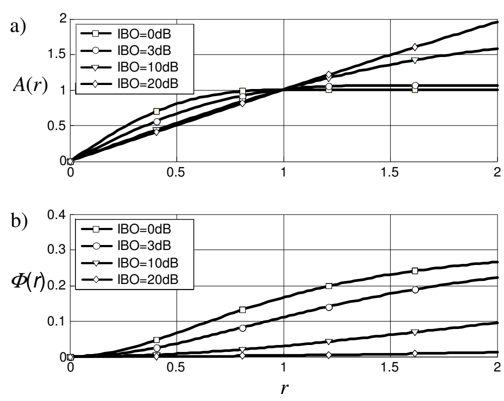

Fig. 3 Characteristics of the HPA for IBO varying from 0dB (highly non-linear system) to 20dB (almost linear system) a) AM-AM and b) AM-PM.

The signal $z(n)$ is sent through a multi-path fading channel, modeled by $\{h_k\}$ composed of two independent Rayleigh variables with equal powers, normalized so as :

$$\sum_k E\left[|h_k|^2\right]=1 \tag{6}$$

A pseudo-random i.i.d. gaussian sequence with variance σ_e^2 is added to simulate the environmental noise, resulting in the Signal to Noise Ratio per bit evaluated for $\tilde{r}(n)$:

$$\frac{E_b}{N_0}=-20\log_{10}(\sigma_e)-3 \qquad [\text{dB}] \tag{7}$$

The decision about transmitted symbol $s(n\text{-}D)$ will be taken on the basis of the observation vector $\tilde{\boldsymbol{r}}(n)=[\tilde{r}(n),\ldots,\tilde{r}(n-M+1)]^T$. Since the channel is well modeled as a FIR system $\mathcal{H}[\ ]$ with memory length M_h-1 then:

$$\tilde{\boldsymbol{r}}(n)=\mathcal{H}[\boldsymbol{s}_{FF}(n),\boldsymbol{s}_{FB}(n)]+\boldsymbol{e}(n) \tag{8}$$

where $\boldsymbol{s}_{FF}(n)=[s(n),\ldots,s(n-D)]$ is the vector of undetected symbols and $\boldsymbol{s}_{FB}(n)=[s(n-D-1),\ldots,s(n-M_s+1)]$ is the vector of symbols already detected, M is the detector order, D the decision delay, $\boldsymbol{e}(n)$ is the vector composed of errors and $M_s=M_h+M$-1 [4][9].

III. TURBO EQUALIZATION ALGORITHM

A turbo-equalizer consists of two blocks: a soft-input soft-output (SISO) equalizer and a SISO decoder. The information between these two blocks is exchanged under the form of log-likelihood ratios (LLR) (cf. Fig. 1) [10]. The decoder produces extrinsic LLRs $\Lambda_b^e(n)$ and $\Lambda_d^e(n)$ on information- and coded bits respectively, taking as the input $\Lambda_d^i(n)$ which is a deinterleaved version of $\Lambda_{c_k}^e(n)$ obtained from the SISO equalizer. The algorithm max-log MAP is used for SISO channel decoding. We refrain from giving the details of this algorithm referring readers to [10][11][12]. The equalizer may also be implemented using the algorithm MAP (or max-log MAP), but they are characterized by prohibitive complexity for higher order modulation (e.g. 4QAM) and significant channel length. For this reason, the algorithm MAP-DFE advocated in [4] was chosen for implementation.

A. Algorithm max-log MAP-DFE

The MAP-DFE equalizer should produce the extrinsic information on the coded bits:

$$\Lambda_{c_k}^e(n-D)=\log\left\{\frac{p_{\tilde{r}}(\tilde{\boldsymbol{r}}(n)|c_k(n-D)=1)}{p_{\tilde{r}}(\tilde{\boldsymbol{r}}(n)|c_k(n-D)=0)}\right\} \tag{9}$$

using a priori information about the coded bits:

$$\Lambda_{c_k}^a(n)=\log\left\{\frac{\Pr(c_k(n)=1)}{\Pr(c_k(n)=0)}\right\} \tag{10}$$

The expression in (9) may be calculated using the following formulas under the assumption that the past symbols were detected correctly (feedback without errors i.e. $\hat{\boldsymbol{s}}_{FB}(n)=\boldsymbol{s}_{FB}(n)$):

$$
\begin{aligned}
&p_{\tilde{r}}\left(\tilde{\boldsymbol r}(n)|c_k(n-D)=q\right)=\\
&=\sum_{\boldsymbol v(n)\in S_{FF}^{k,q}} p_{\tilde{r}}\left(\tilde{\boldsymbol r}(n)|\boldsymbol v(n),\hat{\boldsymbol s}_{FB}(n)\right)\Pr\left(\boldsymbol s(n)=\boldsymbol v(n)\right)\\
&\sim\sum_{\boldsymbol v(n)\in S_{FF}^{k,q}} \exp\left(-\tfrac{1}{\hat\sigma_e^2}\left\|\tilde{\boldsymbol r}(n)-\mathcal{H}\left[\boldsymbol v(n),\hat{\boldsymbol s}_{FB}(n)\right]\right\|^2+\Lambda(\boldsymbol v(n))-q\Lambda_{c_k}^a(n-D)\right)
\end{aligned}
\tag{11}
$$

where q=0,1, $\hat\sigma_e^2$ is the estimate of the variance of the gaussian noise and :

$$
\begin{aligned}
&S_{FF}^{k,q}=\left\{\boldsymbol s_{FF}(n):c_k(n-D)=q\right\}\\
&\Lambda(\boldsymbol v(n))=\sum_{m=0}^{D}\sum_{k=0}^{1}\Lambda_{c_k}^a(n-m)\left[v(n-m)\right]_k
\end{aligned}
\tag{12}
$$

where $\left[v(n-m)\right]_k$ is the k-th bit of the symbol $v(n\text{-}m)$.

Using the approximation $\ln(\sum_k\exp(\beta_k))\approx\max_k\{\beta_k\}$, the algorithm in (11) may be simplified, resulting in the algorithm called max-log MAP DFE:

$$
\begin{aligned}
&\Lambda_{c_k}(n-D)=\Sigma_{\min}^{k,1}-\Sigma_{\min}^{k,0}-\Lambda_{c_k}^e(n-D)\\
&\Sigma_{\min}^{k,q}=\min_{\boldsymbol v(n)\in S_{FF}^{k,q}}\left\{\tfrac{1}{\hat\sigma_e^2}\left\|\tilde{\boldsymbol r}(n)-\mathcal{H}\left[\boldsymbol v(n),\hat{\boldsymbol s}_{FB}(n)\right]\right\|^2-\Lambda(\boldsymbol v(n))\right\}
\end{aligned}
\tag{13}
$$

Clearly all possible vectors $\boldsymbol s_{FF}(n)$ have to be enumerated in the algorithm (13) so its complexity per symbol is proportional to 4^{D+1} independently of the structure of $\mathcal{H}[\ ]$.

There is however a difference whether the operator $\mathcal{H}[\ ]$ is linear or non-linear. Supposing a quasi-static channel behavior, all possible channel states $\mathcal{H}\left[\boldsymbol s_{FF}(n),\boldsymbol s_{FB}(n)\right]$ may be found after the channel is identified and prior to equalization. This initialization process has a complexity proportional to 4^{M_s}. It may be diminished for linear operator $\mathcal{H}[\ ]$ noting that:

$$
\mathcal{H}\left[\boldsymbol s_{FF}(n),\boldsymbol s_{FB}(n)\right]=\mathcal{H}_{FF}\left[\boldsymbol s_{FF}(n)\right]+\mathcal{H}_{FB}\left[\boldsymbol s_{FB}(n)\right]\tag{14}
$$

In this case the complexity is proportional to the number of all possible states $\boldsymbol s_{FF}(n)$ and $\boldsymbol s_{FB}(n)$, i.e. to $4^{D+1}+4^{M_h-1}$.

B. Linear channel identification

The channel estimate $\{\hat h_k\}$ used by the equalizer in LTEQ is obtained using Least Squares (LS):

$$
\begin{aligned}
&\hat{\boldsymbol h}=\left(\boldsymbol S_{train}^H\boldsymbol S_{train}\right)^{-1}\boldsymbol S_{train}^H\tilde{\boldsymbol r}_{train}\\
&\boldsymbol S_{train}=\begin{bmatrix} s_{train}(M_h-1) & \cdots & s_{train}(0)\\ \vdots & & \vdots\\ s_{train}(N_{train}-1) & & s_{train}(N_{train}-M_h)\end{bmatrix}\\
&\tilde{\boldsymbol r}_{train}=\left[\tilde r_{train}(M_h-1),\ldots,\tilde r_{train}(N_{train}-1)\right]
\end{aligned}
\tag{15}
$$

where $\tilde r_{train}(n)$ and $s_{train}(n)$ n=0,…,N_{train} are the received and sent signals corresponding to the training periods in the burst (c.f. Fig. 2).

C. Non-Linear channel identification

In order to obtain the estimate of the non-linear channel we will make the assumption that the structure of the model is known and that we know the impulse response $\{g_k\}$ (pulse shaping is introduced by the system). Note also that the element α_A contributes only to the amplitude change in the signal $z(n)$ and may be as well included in the impulse response $\{\hat h_k\}$, so it was fixed to $\hat\alpha_A=1$. The remaining parameters are then identified using LS method, i.e. minimizing the criterion:

$$
\begin{aligned}
&J\left[\boldsymbol p_{NL},\{h_k\}\right]=\left\|\tilde{\boldsymbol r}_{train}-\mathcal{H}\left[\boldsymbol s_{train};\boldsymbol p_{NL},\{h_k\}\right]\right\|_2^2\\
&\mathcal{H}\left[\boldsymbol s_{train};\boldsymbol p_{NL},\{h_k\}\right]=\boldsymbol H\boldsymbol z_{train},\ \boldsymbol z_{train}=\boldsymbol f_{HPA}(\boldsymbol y_{train};\boldsymbol p_{NL})
\end{aligned}
\tag{16}
$$

where $\hat{\boldsymbol H}$ is the convolution matrix composed of elements $\{\hat h_k\}$, $\boldsymbol p_{NL}=\left[\beta_A,\alpha_\Phi,\beta_\Phi\right]$, $\boldsymbol z_{train}$ and $\boldsymbol y_{train}$ are vectors composed of signals $z(n)$ and $y(n)$ (c.f. Fig. 1) corresponding to the training period in the burst (c.f. Fig. 2), and $\boldsymbol f_{HPA}()$ is the vector-extended scalar function defined in (2). Note that for fixed parameters $\boldsymbol p_{NL}$, the vector $\boldsymbol z_{train}$ is known so that $\{\hat h_k\}$ may be obtained using the linear channel identification method defined in section III.B. No closed-form solution of (16) exists so iterative minimization was used to obtain the estimates of the parameters. Satisfactory convergence (faster then the steepest-descent method) was obtained using a gradient-based variable-step minimization inspired from VLMS adaptive algorithm [13]:

$$
\left.\begin{aligned}
&\boldsymbol\mu(i)=\boldsymbol\mu(i-1)+\mu\nabla J\left[\boldsymbol p_{NL}(i)\right]\odot\nabla J\left[\boldsymbol p_{NL}(i-1)\right]\\
&\boldsymbol p_{NL}(i+1)=\boldsymbol p_{NL}(i)-\boldsymbol\mu(i)\odot\nabla J\left[\boldsymbol p_{NL}(i)\right]
\end{aligned}\right\}\quad i=1,\ldots,I_{MAX}\tag{17}
$$

where $\odot$ denotes element-by element multiplication, i is the iteration number, I_{MAX}=30 was used in this study, and the gradient

$$
\nabla J\left[\boldsymbol p_{NL}\right]=\left[\frac{\partial}{\partial\beta_A}J\left[\boldsymbol p_{NL}\right],\frac{\partial}{\partial\alpha_\Phi}J\left[\boldsymbol p_{NL}\right],\frac{\partial}{\partial\beta_\Phi}J\left[\boldsymbol p_{NL}\right]\right]^T\tag{18}
$$

is defined by the following formula :

$$
\begin{aligned}
&\frac{\partial}{\partial\gamma}J\left[\boldsymbol p_{NL}\right]=2\mathrm{Re}\left\{\left[\boldsymbol H\frac{\partial}{\partial\gamma}\boldsymbol z_{train}\right]^H\left(\boldsymbol H\boldsymbol z_{train}-\tilde{\boldsymbol r}_{train}\right)\right\}.\\
&\frac{\partial}{\partial\gamma}\boldsymbol z_{train}=\left[\frac{\partial}{\partial\gamma}z_{train}(0),\ldots,\frac{\partial}{\partial\gamma}z_{train}(N_{train}-1)\right]
\end{aligned}
\tag{19}
$$

Then partial derivatives in (19) are calculated as follows:

$$
\begin{aligned}
&\frac{\partial}{\partial\beta_A}z(n)=\frac{-|y(n)|^2}{1+\beta_A|y(n)|^2}z(n)\\
&\frac{\partial}{\partial\alpha_\Phi}z(n)=\frac{j|y(n)|^2}{1+\beta_\Phi|y(n)|^2}z(n)\\
&\frac{\partial}{\partial\beta_\Phi}z(n)=\frac{-j|y(n)|^4}{\left(1+\beta_\Phi|y(n)|^2\right)^2}z(n)
\end{aligned}
\tag{20}
$$

The algorithm was initialized (for i=0 in (17)) supposing the linear model of HPA i.e.

$$
\boldsymbol p_{NL}(0)=\left[\hat\beta_A(0),\hat\alpha_\Phi(0),\hat\beta_\Phi(0)\right]=[0,0,0]\tag{21}
$$

The estimate of the variance of the errors $\hat\sigma_e^2$ required by the equalization algorithms (cf. (11)) is calculated as the average energy of the residual errors (for both linear and non-linear channel estimation):

$$\hat{\sigma}_e^2 = \frac{1}{N_{train}} J\left[\boldsymbol{p}_{NL}, \{h_k\}\right] \tag{22}$$

The iterative adaptation is compared in Fig. 4 with the results obtained by means of linear channel identification for the same realization of data at E_b/N_0=10dB for *IBO*=0dB. It may be appreciated that the modeling errors (the difference between $\hat{\sigma}_e^2$ and σ_e^2) are only. 10% of σ_e^2 for NTEQ while they are 60% of σ_e^2 for LTEQ, which explains well the differences in performance of both algorithms described further on.

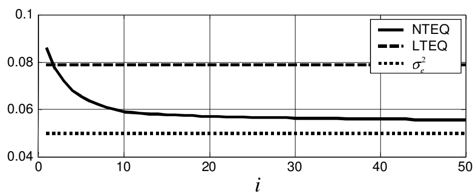

Fig. 4 Comparison of variance estimators $\hat{\sigma}_e^2$ obtained by means of the algorithms for linear and non-linear channel estimation (LTEQ and NTEQ respectively) for E_b/N_0=10dB and *IBO*=0dB. The actual variance of errors σ_e^2 is shown for reference.

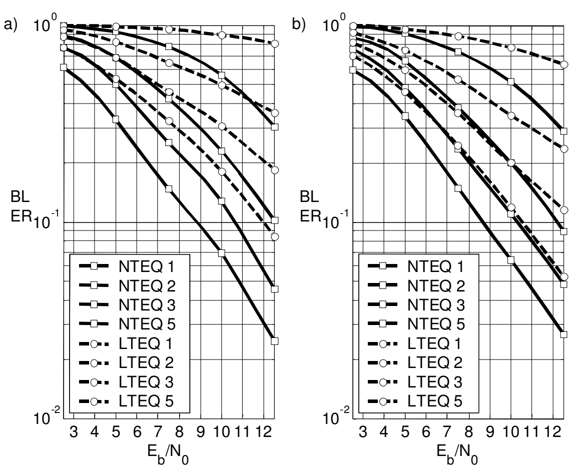

Fig. 5 Results obtained by means of turbo-equalization algorithms for a) *IBO*=0dB and b) *IBO*=3dB.

IV. Simulation Results

The algorithms of turbo-equalization were compared using computer simulation. The results were obtained using 3000 independent transmissions for each value of *IBO* varying from 0dB (highly non-linear system) to 20dB (system with relatively linear amplitude behavior) (cf. Fig. 3). The training sequences and interleaver were obtained using pseudo-random number generator but were fixed for all the simulations. The convolutional code with rate ½ of constraint length 3 with generators [5 7] (in octal) was used. The equalizer's parameters were: *D*=2 and *M*=3. The estimated channel length M_h was supposed to be known *a priori*., i.e. was set to M_h=4 for linear channel identification (prior to applying the algorithm LTEQ) and to M_h=2 for the estimation of the channel $\{h_k\}$ (being a part of non-linear channel identification required by the algorithm NTEQ). No increase in performance was observed beyond the fifth turbo-iteration, so it was taken as the ultimately achievable performance. The performance of the studied algorithms is presented in terms of BLER in Fig. 5, Fig. 6, Fig. 7, and Fig. 8; this criterion may be calculated in actual system (for example using CRC) and it is related to the theoretical outage capacity of the link.

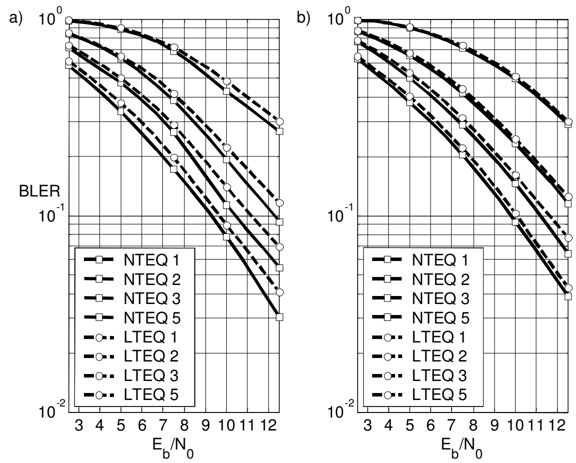

Fig. 6 Results obtained by means of turbo-equalization algorithms for a) *IBO*=10dB and b) *IBO*=20dB.

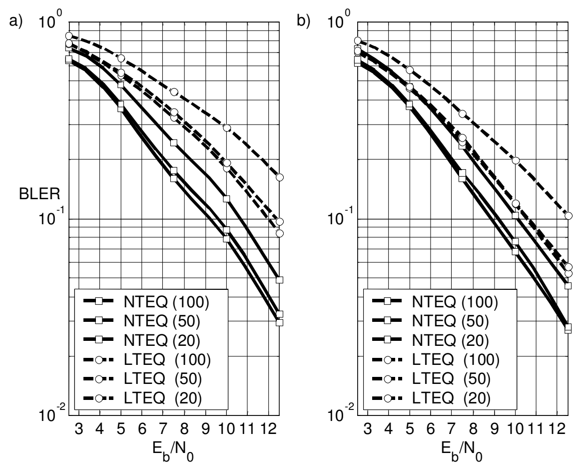

Fig. 7 Results obtained after the fifth iteration of the algorithms NTEQ and LTEQ for different lengths of the training sequence (the parameter shown in parenthesis). The results for a) *IBO*=0dB and b) *IBO*=3dB.

On the basis of the simulations we made the following observations :

- As expected, the algorithm NTEQ outperforms the algorithm LTEQ under the condition that the channel is known exactly. In this case the gain in the fifth iteration evaluated for BLER=10^{-1} drops from 3dB for *IBO*=dB down to 1dB for *IBO*=3dB; it is practically 0dB for quasi-linear channels i.e. IBO=20dB.

- The performance of the algorithm NTEQ is practically insensitive to the channel's non-linearity (the same E_b/N_0 is required at BLER=10^{-1} for different *IBO*s) .
- The channel estimation introduced penalty of ca. 2dB when the number of training symbols decreased from 100 to 20 (cf. Fig. 5 with Fig. 7 and Fig. 6 with Fig. 8). This effect is observed for both algorithms NTEQ and LTEQ since the number of parameters to be estimated is practically the same in both cases, i.e., 4 parameters for LTEQ and 5 parameters for NTEQ; this is because in the latter we take advantage of *a priori* information about the impulse response $\{g_k\}$.

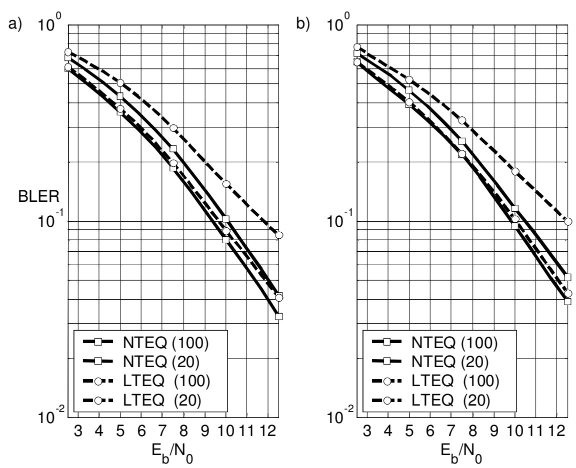

Fig. 8 Results obtained after the fifth iteration of the algorithms NTEQ and LTEQ for different lengths of the training sequence (the parameter shown in parenthesis). The results for a) *IBO*=10dB and b) *IBO*=20dB.

For given parameters M, D and M_h used in this study the complexity of the initialization of the algorithms LTEQ and NTEQ is proportional to $4096=4^6$ and to $128=4^3+4^3$ respectively, so it increases for the latter by the factor of 32. Similar increase in the complexity may be observed comparing non-linear channel identification (cf. III.C) to linear channel identification (cf.III.B); this is mainly due to the iterative algorithm (17).

V. CONCLUSIONS

This paper evaluates the applicability of turbo-equalization algorithms for telecommunication channels characterized by non-linear distortions likely introduced by High Power Amplifiers. A turbo-equalization algorithm based on a non-linear channel model (NTEQ) and one based on a linear model (LTEQ) were compared by means of numerical simulations. The algorithm NTEQ combined with proposed non-linear channel identification yields near-optimal results if sufficiently long training sequence is available.

The results obtained allow us to draw the following conclusions and recommendations which suggest as well the further research directions:

- Turbo-equalization may significantly improve the results when applied to highly-nonlinear channel. It is not recommended for channels with mild and weak non-linearities due to significant additional computation load it requires. Hence it would be advantageous to determine the degree of channel non-linearity before deciding whether the algorithm LTEQ or NTEQ should be applied.
- The non-linear channel may be successfully identified provided that sufficient training data is available. Since the number of training symbols affects seriously the equalization results, it might be beneficial to apply the so called bootstrap technique [14] which uses the preliminary results from the iterations to artificially increase the number of training data.
- The computation load of the proposed SISO equalization algorithm may be alleviated using alternative PIC-type soft- or hard decision linear SISO algorithms e.g [2].

REFERENCES

[1] C. Douillard, M. Jezequel, C. Berrou, A. Picart, P. Didier, A. Glavieux, “Iterative Correction of Intersymbol Inteference : Turbo-Equalization”, *European Trans. Telecommunications*, Vol.6-5, pp. 507-511, 1995.

[2] C. Laot, A. Glavieux, J. Labat, “Turbo equalization: Adaptive Equalization and Channel Decoding Jointly Optimized”, *IEEE J. Sel. Areas Comm.* , Vol. 19, No. 9, pp. 1744-1752, Sept 2001.

[3] M. Tüchler, J. Hagenauer, “Turbo equalization using frequency domain equalizer”, *Proc. Allerton Conference*, Monticello, USA, pp. 144–153, Oct. 2000.

[4] M. S. Yee, B. L. Yeap, L. Hanzo, “Radial Basis Function Assisted Turbo Equalisation”, *Proc. IEEE VTC-Spring*, pp. 640-644, 2000.

[5] B. L. Yeap, C. H. Wong, L. Hanzo, “Reduced Complexity In-phase/quadrature-phase Turbo Equalisation Using Iterative Channel Estimation”, *Proc. ICC'2001*, pp.393-397, 2001.

[6] E. Costa, S. Pupolin, “M-QAM-OFDM System Performance in the Presence of a Nonlinear Amplifier and Phase Noise”, *IEEE Trans. Comm.*, Vol. 50, No. 3, pp. 462–472, March 2002.

[7] H. W. Kang, Y. S. Cho, D. H. Youn, “On Compensating Nonlinear Distortions of an OFDM Systems Using an Efficient Adaptive Predistorter”, *IEEE Trans. Comm.*, Vol. 47, No. 4, pp. 522-526, April 1999.

[8] S.W. Heo, S.B. Gelfand, J.V. Krogmeier, “Performance analysis of combined equalization and decoding in the trellis coded nonlinear satellite channel Communications”, *Proc. IEEE Intern. Conf. on Comm.*, Vol. 7, pp. 2231-2235, June 2001.

[9] S. Chen, B. Mulgrew, S. McLaughlin, "Adaptive Bayesian Equalizer with Decision Feedback", *IEEE Trans. Signal Proc.*, Vol. 41, No. 9, pp. 2918-2927, Sept. 1993.

[10] G. Bauch, V. Franz, “A comparison of soft-in/soft-out algorithms for "Turbo-detection".” Proc. Intern. Conf. Telecom.. (ICT98), Porto Carras, Greece, pp. 259-263, June 1998.

[11] L. R. Bahl, J. Cocke, F. Jelinek, J. Raviv, “Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate”, *IEEE Trans. Inf. Theory*, pp. 284-287, March 1974.

[12] J. Hagenauer “The Turbo Principle - Tutorial Introduction and State of the Art”, *Proc. Intern. Symposium on Turbo-Codes*, Brest, France, Sept. 1997.

[13] B. Farhang-Boroujeny, *Adaptive Filtres*, Wiley & Sons 1998, Ch. 6.7.

[14] C. Tidestav, E. Lidskog, “Bootstrap Equalization”, *IEEE Intern. Conf. on Universal Person. Comm. (ICUPC'98)*, pp. 1221-1225, 1998.