Turbo-Fuzzy Equalization for Single-Carrier Power Line Channels

Moisés V. Ribeiro
Dept. of Electrical Circuit
Federal University of Juiz de Fora
Juiz de Fora MG, 36036-330, Brazil
mribeiro@ieee.org

Murilo B. Loiola, Renato R. Lopes
Dept. of Communications
School of Electrical and Computer Engineering
University of Campinas, Campinas, Brazil
{mloiola,rlopes}@decom.fee.unicamp.br

Abstract— This paper introduces a *turbo-fuzzy equalizer* (TFEQ) to cope with the inter-symbol interference effects and impulsive noise occurrences in bandlimited digital communication channels, such as the indoor and outdoor power line circuits. We exploit a close relationship between the fuzzy and Bayesian equalizers to derive an adaptive nonsingleton turbo-fuzzy equalization technique. Generally speaking, the TFEQ is obtained by introducing an adaptive decision feedback nonsingleton fuzzy equalizer into the turbo equalization structure. Simulation results reveal the applicability of this approach to overcome the noise impairments in the power line frequency range for broadband applications.

Keywords— Turbo equalization, fuzzy systems, Bayesian equalizer, nonlinear equalization.

I. INTRODUCTION

POWER Line Communications (PLC) have recently emerged as a highly regarded candidate for in-home, local area and rural broadband multimedia transmissions since *power line* (PL) channels present the appealing benefits of low cost access network installation, due to the use of widely deployed power line infrastructure, and theoretical capacities that can surpass 1 Gbps [1].

As a result, a great deal of attention has been devoted to model PL channel properties. These researches have determined that PL channels always present: i) attenuation proportional to frequency and distance; ii) time-varying behavior; iii) high impedance varying with time and frequency; iv) strong impulsive noise presence due to switching on/off of electric loads; and v) frequency selectivity.

The increasing demand for high spectral efficiency modulation has demanded continuous system evolution in order to improve performance. One of the major obstacles that high-rate systems must overcome is the fact that increasing data rates through bandlimited channels introduces *inter-symbol interference* (ISI), which drastically degrades the received signal. For bandlimited PL channels, the occurrence of high power impulsive noise exacerbates the degradation.

The use of powerful channel coding and modulation techniques to cope with the aforementioned problems in PL channels has been investigated in [2]. However, in conventional

receivers, the redundancy added by the codes is not used by the equalizer to mitigate ISI. Consequently, equalization and decoding are carried out separately. The receiver performance could be greatly improved if these two tasks were jointly performed. A successful approach to do this is to iterate between the equalizer and channel decoder several times on the same set of received data. This method, known as turbo equalization, was first proposed by Douillard et al. [3].

Hence, aiming at the development of new signal processing techniques to cope with the PL channel impairments and to achieve a higher spectral efficiency, this contribution presents a turbo-fuzzy equalizer for PL channels. The derived equalizer uses an adaptive nonsingleton fuzzy filter with variable step-size. The advantages offered by the proposed technique are: i) it can deal with the presence of impulsive noise at the output PL channels better than an adaptive and singleton turbo-fuzzy equalizer. Note that this singleton turbo-fuzzy equalizer provides the same results obtained by a turbo equalization technique implemented with a specific adaptive Bayesian equalizer, and ii) it provides improved performance due to the use of variable step-size for each fuzzy equalizer parameters. Comparison results between the proposed turbo-fuzzy equalizer and the one derived from a Bayesian assumption (adaptive version of the turbo equalizer proposed in [4]) verify that the former provides a considerable improvement in terms of BER.

This paper is organized as follows. Section II formulates the PL channel equalization problem. Section III presents some important issues related to fuzzy systems. Section IV discusses the proposed turbo-fuzzy equalizer and computational results are presented in Section V. Finally, concluding remarks are stated in Section VI.

II. PROBLEM FORMULATION

A discrete-time, linear, causal and time-invariant model of baseband PL channels for *broadband* PLC (BoPLC) and *narrowband* PLC (NaPLC) applications is given by

$$y(n) = \tilde{y}(n) + v(n) = \sum_{i=0}^{L_h-1} h(i)x(n-i) + v(n), \quad (1)$$

where $\tilde{y}(n)$ denotes the PL channel output free of noise, $v(n)$ is the additive noise representing the sum of background and

impulsive noises, $\{h(n)\}_{n=0}^{L_h-1}$ is the bandlimited, dispersive, and linear *finite impulse response* (FIR) of the PL channel, and the sequence $\{x(n)\}$ consists of transmitted symbols. For the frequency range from 0.5 to 30.0 MHz, impulse responses based on multipath approach can be derived from [5].

By considering a *binary phase shift keying* (BPSK) constellation for a single-carrier or a *code division multiple access* (CDMA) system, the transmitted sequence $\{x(n)\}$ is taken from $\{+1, -1\}$, and it is assumed to be an equiprobable and independent sequence. The channel outputs observed by the equalizer can be written as $\mathbf{y}(n) = [y(n) \cdots y(n - L_w + 1)]^{\mathrm{T}}$, while the equalizer impulse response can be written as $\mathbf{w}(n) = [w(n) \cdots w(n - L_w + 1)]^{\mathrm{T}}$. The vector of the transmitted symbols that influence the equalizer decision is expressed by $\mathbf{x}(n) = [\hat{x}(n) \cdots \hat{x}(n - L_h - L_w + 1)]^{\mathrm{T}}$. As a result, there are $n_s = 2^{L_h + L_w}$ possible combinations of the channel input sequence, and n_s different values of the noise-free channel output vector $\widetilde{\mathbf{y}}(n) = [\widetilde{y}(n) \cdots \widetilde{y}(n - L_w + 1)]^{\mathrm{T}}$ are possible. Each of these noise-free channel output vector is called channel output state vector $\widetilde{\mathbf{y}}_j(n)$, $j = 1, \cdots, n_s$ given by

$$\widetilde{\mathbf{y}}_j(n) = \mathbf{H}\mathbf{x}_j(n), \tag{2}$$

where $\mathbf{x}_j(n) = [x(n) \cdots x(n - L_h + L_w + 1)]^{\mathrm{T}}$ denotes the *j*th possible input vector and $\mathbf{H}$ is the channel convolution matrix given by

$$\mathbf{H} = \begin{bmatrix} h_0 & h_1 & \cdots & h_{L_h-1} & \cdots & 0 \\ 0 & h_0 & \cdots & h_{L_h-1} & \cdots & 0 \\ \vdots & \vdots & & & & \vdots \\ 0 & 0 & h_0 & \cdots & h_{L_h-2} & h_{L_h-1} \end{bmatrix}, \tag{3}$$

The equalizer output $\hat{x}(n-d)$ is an estimate of the transmitted symbol, delayed by d samples.

If the additive noise is modeled by a random process with normal distribution as $\mathcal{N}(0, \sigma_v^2)$, the *normalized optimal bayesian equalizer* (NOBE) given the observation vector $\mathbf{y}(n)$ is defined by [6]

$$f_b(\mathbf{y}(n)) = \frac{1}{\sum\limits_{\widetilde{\mathbf{y}}_k \in \mathcal{Y}_d} \exp\left(\frac{-\|\mathbf{y}(n) - \widetilde{\mathbf{y}}_k\|^2}{2\sigma_v^2}\right)}$$

$$\left\{ \sum\limits_{\widetilde{\mathbf{y}}_i \in \mathcal{Y}_d^+} \exp\left(\frac{-\|\mathbf{y}(n) - \widetilde{\mathbf{y}}_i\|^2}{2\sigma_v^2}\right) - \sum\limits_{\widetilde{\mathbf{y}}_j \in \mathcal{Y}_d^-} \exp\left(\frac{-\|\mathbf{y}(n) - \widetilde{\mathbf{y}}_j\|^2}{2\sigma_v^2}\right) \right\} \tag{4}$$

where $\mathcal{Y}_d^+ = \{\widetilde{\mathbf{y}}_i, \ i = 1, \cdots, 2^{L_h + L_w - 1} | x(n-d) = +1\}$ and $\mathcal{Y}_d^- = \{\widetilde{\mathbf{y}}_j, \ j = 2^{L_h + L_w - 1} + 1, \cdots, 2^{L_h + L_w} | x(n-d) = -1\}$ denote the state space composed of L_w-length vectors $\mathbf{y}_i$ given that $x(n-d) = +1$ and $x(n-d) = -1$, respectively, and $\mathcal{Y}_d = \mathcal{Y}_d^+ \cup \mathcal{Y}_d^- = \{\widetilde{\mathbf{y}}_k\}$, $1 \leq k \leq n_s = 2^{L_h + L_w}$ denotes the set of all possible channel output state vectors. A decision feedback version of NOBE is obtained assuming that the equalizer input vector is composed of output channel samples along with past equalizer decisions [7].

In the following section, we present some basic ideas on fuzzy systems as well as the fuzzy equalizers we used in our simulations.
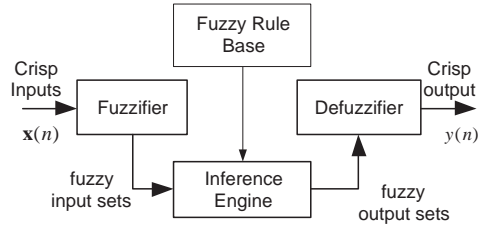

Fig. 1. Basic components of a Fuzzy system.

III. FUZZY SYSTEMS: IMPORTANT ISSUES

The PL channels introduce strong ISI. Also, the deep frequency notches present in the channel transfer function prevent the use of linear equalizers, as the noise enhancement they cause is a serious drawback on noisy channels. In these cases, *decision feedback equalizers* (DFEs) are an attractive alternative. Against the use of DFEs is the fact that PL channels have long and varying impulse responses and strong noise bursts, concentrated in both time and frequency. The noise bursts may cause catastrophic error propagation, and the fast time variations may not allow channel parameter tracking when adaptation cannot be fast enough. In this regard, the use of other kinds of nonlinear equalizers capable of coping with strong noise bursts as well as long and fast-varying impulse responses is of interest. Among the nonlinear equalizers, *fuzzy systems* (FS) have been widely applied in communication channels presenting the aforementioned impairments [8].

Generally speaking, fuzzy systems are nonlinear systems capable of inferring complex nonlinear relationships between input and output variables. They do not make any assumptions regarding the structure of the process, nor do they invoke any kind of probabilistic distribution model, i.e., they belong to the general family of model-free, data-driven, nonparametric methods. The basic building blocks of a fuzzy system, illustrated in Fig. 1, are the fuzzifier, the fuzzy rule base, the inference engine, and the defuzzifier. The fuzzy rule base is a group of rules that are combined in the inference engine to produce a fuzzy output. In essence, the inference engine produces mappings from fuzzy sets to fuzzy sets whereas the fuzzifier maps the inputs into fuzzy sets, which are subsequently used as inputs to the inference engine. Finally, the defuzzifier is responsible for mapping the fuzzy sets yielded by the inference engine into crisp numbers.

The exact mathematical structure of FSs depends mainly on four choices namely, type of fuzzification, membership functions, inference mechanism, and defuzzification strategy. Assuming singleton fuzzification and a proper choice of the other parameters, it is possible to show [8] that the singleton fuzzy system, operating as a transverse or feedback symbol-decision equalizer, results in the normalized Bayesian solution that achieves a probability of misclassification very close to the one attained with Bayesian equalizer derived from the *maximum likelihood* (ML) criterion. Then, we can affirm that (4) can be surprisingly seen as a fuzzy system if some restrictions are taken into account. In fact, if one assumes

singleton fuzzification, Gaussian membership functions, max-product composition, product implication, and height defuzzification, then the output of the normalized singleton type-1 fuzzy equalizer is given by [9]

$$f_s(\mathbf{y}(n)) = \cfrac{1}{\sum\limits_{\mathbf{y}_k \in \mathcal{Y}_d} \sum\limits_{i=0}^{L_w-1} \exp\left(\frac{-(y(n-i)-y_k(i))^2}{2\sigma_{F_i^k}^2}\right)}$$
$$\left\{ \sum\limits_{\mathbf{y}_k \in \mathcal{Y}_d^+} \sum\limits_{i=0}^{L_w-1} \exp\left(\frac{-(y(n-i)-y_k(i))^2}{2\sigma_{F_i^k}^2}\right) - \right.$$
$$\left. \sum\limits_{\mathbf{y}_k \in \mathcal{Y}_d^-} \sum\limits_{i=0}^{L_w-1} \exp\left(\frac{-(y(n-i)-y_k(i))^2}{\sigma_{F_i^k}^2}\right) \right\} \quad (5)$$

where $y(n-i)$ is PL channel output samples, $\sigma_{F_i^k}^2$ and $y_k(i)$ are the variance and the mean of the ith Gaussian membership function of the kth rule of the fuzzy system. Now, assuming that $\sigma_{F_i^k} = \sigma_v^2$ and $\mathbf{y}_k = [y_k(0) \; y_k(1) \; \cdots \; y_k(L_w-1)]^T = \tilde{\mathbf{y}}_k$, where $\tilde{\mathbf{y}}_k$ is the PL channel vector state defined in Section II. These assumptions mean that the variances of all membership functions are the same and equal to the noise variance at the PL channel output and the centers of these membership functions in the kth rule are equal to the elements of the kth output PL channel vector state, $k = 1, \cdots, n_s$, which was defined in Section II. As a result, we can conclude that (5) and (4) are equal under the aforementioned constraints. In other words, NOBE can be seen as a particular case of a fuzzy equalizer. Also, we can note that this singleton fuzzy equalizer is equal to normalized version of the the equalizer used in [4].

If we now assume nonsingleton fuzzification, Gaussian membership functions, max-product composition, product implication, height defuzzification, Gaussian membership functions modeling the uncertainty of the input data and fuzzy rules and $\star$, a t-norm named product of a nonsingleton type-1 FS [10], then the output of this *normalized nonsigleton fuzzy equalizer* (NONFE) is given by [6]

$$f_{ns}(\mathbf{y}(n)) = \cfrac{1}{\sum\limits_{\tilde{\mathbf{y}}_k \in \mathcal{Y}_d} \sum\limits_{i=0}^{L_w-1} \exp\left(\frac{-(y(n-i)-\tilde{y}_k(i))^2}{2(\sigma_y^2+\sigma_{F_i^k}^2)}\right)}$$
$$\left\{ \sum\limits_{\tilde{\mathbf{y}}_k \in \mathcal{Y}_d^+} \sum\limits_{i=0}^{L_w-1} \exp\left(\frac{-(y(n-i)-\tilde{y}_k(i))^2}{2(\sigma_y^2+\sigma_{F_i^k}^2)}\right) - \right.$$
$$\left. \sum\limits_{\tilde{\mathbf{y}}_k \in \mathcal{Y}_d^-} \sum\limits_{i=0}^{L_w-1} \exp\left(\frac{-(y(n-i)-\tilde{y}_k(i))^2}{2(\sigma_y^2+\sigma_{F_i^k}^2)}\right) \right\} \quad (6)$$

where $y(n-i)$ is PL channel output samples, $\sigma_{F_i^k}^2$ and $y_k(i)$ are the variance and the mean of the ith Gaussian membership function of the kth rule of the fuzzy system. σ_y is the standard deviation of the membership function that is used to model the uncertainty at the input data.

If one compares (5) and (6), it possible to see that the only difference between both of them is the introduction of the term σ_y^2. In this case, if $\sigma_y = 0$, then $f_{ns}(\mathbf{y}(n)) = f_s(\mathbf{y}(n))$. As a result, the NONFE can be reduced to the NOBE. Now, the big question is: Why should we replace NOBE by NONFE?

The main advantage offered by NONFE is that nonsingleton fuzzy system are capable of dealing with uncertainties in the input data. From the telecommunication point of view, it sounds quite difficult to understand because the used constraints to obtain NOBE from the NONFE.

However, when the singleton fuzzy system is derived we do not take into account the uncertainties on the input data into the fuzzy system formulation because singleton fuzzy system is unable to do it. Although, some authors have derived some relations between singleton fuzzy systems and Bayesian equalizers, we can state that those relations are not based on a fuzzy perspective. As a result, few people have noted the strength that nonsingleton fuzzy system can offer for digital communication. Overall, we can say that the nonsingleton fuzzy systems are the only fuzzy system capable of dealing with uncertainties in the input data and, as a result, can improve the equalization performance. Two advantages offered by nonsingleton fuzzy equalizers are: i) faster convergence rate than those obtained with singleton fuzzy equalizers because nonsingleton fuzzy equalizer can handle uncertainties in the input data and ii) lower *mean square error* MSE at the equalizer output than the MSE value attained with singleton fuzzy equalizers if the input data is corrupted by nonGaussian noise because nonsingleton fuzzy equalizers can model the input uncertainties.

As it was well-posed in [10], nonsingleton fuzzy systems are especially useful in cases where the available training data, or the input data to the fuzzy logic system, are corrupted by noise. Conceptually, the nonsingleton fuzzifier implies that the given input value is the most likely value to be the correct one from all the values in its immediate neighborhood; however, because the input is corrupted by noise, neighboring points are also likely to be the correct values, but to a lesser degree.

A type-1 fuzzy system-based decision feedback equalizer can be easily derived. Essentially, the idea is to employ a feedback filter of order $L_b = L_h + L_w - 2 - d$, so that the feedback vector has $n_b = 2^{L_b}$ states. Thus, the channel states, $\mathcal{Y}_d$, can be partitioned into n_b subsets. The new positive and negative channel states are given by

$$\mathcal{Y}_d^{++} = \{\mathbf{y}(n)|x(n-d) = +1 \cap \hat{x}(n-d) = +1\} \quad (7)$$

and

$$\mathcal{Y}_d^{--} = \{\mathbf{y}(n)|x(n-d) = -1 \cap \hat{x}(n-d) = -1\} \quad (8)$$

The related number of states in $\mathcal{Y}_d^{++}$ and $\mathcal{Y}_d^{--}$ becomes

$$n_{ns} = \frac{n_s}{n_b} = 2^{d+1}. \quad (9)$$

It is obvious that the feedback vector reduces the number of channel states required for decision purposes, as originally proved for a Bayesian equalizer implemented by using a RBF network [7].

Adaptive version of these equalizers are also easily derived with backpropagation algorithm. However, we have noted that improved convergence rate is attained with a variable step-size method. In this regard, the *delta-bar-delta* (DBD) method [11]

is a good candidate for updating the step-size associated with each equalizer's parameters. DBD provides high convergence rates as it tries to find the proper learning rate to compensate for small magnitude of the gradient in the flat regions and to dampen the large free parameter changes in deep regions. The DBD consists of a parameter vector-updating rule performed by a modified backpropagation procedure and a learning rate rule defined by

$$\Delta \mathbf{w}(n+1) = -(1-\alpha) \operatorname{diag}\{\mu_0(n)....\mu_{P-1}(n)\} \\ \nabla J(\mathbf{w}(n)) + \alpha \Delta \mathbf{w}(n) \tag{10}$$

and

$$\mu_i(n+1) = \begin{cases} \kappa, & \text{if } \bar{\lambda}_i(n-1)\lambda_i(n) > 0, \\ -\phi \mu_i(n), & \text{if } \bar{\lambda}_i(n-1)\lambda_i(n) < 0 \\ 0, & \text{otherwise}, \end{cases} \tag{11}$$

respectively, where $i = 0, \ldots P - 1$, $\mathbf{w}(n) = [\begin{array}{ccc} w_0(n) & \cdots & w_{P-1}(n) \end{array}]^{\mathrm{T}}$ denotes the vector constituted by all parameters of a fuzzy equalizer, $\mu(n) = [\begin{array}{ccc} \mu_0(n) & \cdots & \mu_{P-1}(n) \end{array}]^{\mathrm{T}}$ is the learning rate vector; $\Delta \mathbf{w}(n+1) = \mathbf{w}(n+1) - \mathbf{w}(n)$; α is the momentum rate, $\lambda_i(n) = \partial J(\mathbf{w}(n))/\partial w_i(n)$ is the partial derivative of a considered cost function with respect to $w_i(n)$ at the nth iteration, $\bar{\lambda}_i(n) = (1 - \delta)\lambda_i(n) + \delta \bar{\lambda}_i(n - 1)$ is an exponential average of the current and past derivatives. $J(\mathbf{w}(n)) = \frac{1}{2}[x(n + d) - \hat{x}(n)]^2$, where $x(n + d)$ and $\hat{x}(n)$ are the transmitted symbol delayed by d samples and the estimated symbol at the receiver side, respectively.

In Section IV, a turbo-fuzzy technique making use of decision-feedback nonsingleton fuzzy system, whose parameters are updated by DBD rule is discussed.

IV. TURBO-FUZZY EQUALIZER

In the turbo approach, it is assumed that the information is encoded by an error-control code before transmission. The ISI channel is modeled by a discrete-time linear filter as in (1), which can be regarded as a rate R = 1, non-systematic, possibly time-varying, convolutional code with complex valued coded symbols. From this point of view, the channel encoder and the channel form a serially concatenated scheme, which can be decoded by an iterative ("turbo") algorithm. So, the equalizer and the channel decoder act as the inner and the outer decoder respectively. This is shown in Fig. 3, where $y(n)$ represents the signal at the output of the channel corrupted by a zero mean additive white Gaussian noise $v(n)$ with variance σ_v^2.

As in turbo coding, the receiver components are *soft-input soft-output* (SISO) devices that exchange soft information iteratively between them. They are classically implemented using conventional *a posteriori probability* (APP) algorithms, based on the optimal BCJR-MAP decoder [12]. The SISO equalizer must accept not only channel values but also *a priori* information about the symbols to be detected. This *a priori* information is provided by the channel decoder at the previous iteration. The outputs of these SISO components

are usually expressed in terms of *log likelihood ratios* (LLR), whose definition for binary symbols is:

$$L(x_k) = \ln \frac{P(x_k = +1|\mathbf{y})}{P(x_k = -1|\mathbf{y})} \tag{12}$$

The sign of the LLR gives the estimated symbols while their amplitudes indicate the reliability of these decisions.

The interleaver plays a key role in an iterative scheme since it permutes the symbols in a given data block, allowing temporal error sequence distribution to be modified [13]. Moreover, it decorrelates the inputs of both decoder and equalizer and, consequently, enables the *a priori* information to be considered as independent information for the equalization process.

A turbo equalizer operates as follows. In the first iteration, there is no *a priori* knowledge about the coded bits, so the equalizer computes the a posteriori probabilities given only the received symbols. From these *a posteriori* values, *extrinsic* information ($L_e^E(x_n)$) for each bit is calculated by subtracting the *a priori* information from the corresponding equalizer output. The *extrinsic* information is the incremental information about the current bit obtained through the detection/decoding from all the other bits in a data block. The soft output from the equalizer is then de-interleaved and passed to the channel decoder that further generates new LLR's. Finally, a new set of *extrinsic* information ($L_e^D(x_k)$) is computed, interleaved and fed back to the equalizer, which treats it as an *a priori* information for the next iteration. The steps above are repeated until a chosen termination criterion, like a maximum number of iterations, is reached.

It is well known that the most suitable equalizers for turbo equalization are implemented by the MAP algorithm. Unfortunately, these algorithms have a computational complexity that grows exponentially with the length of the channel impulse response. In the case of PL channels, these equalizers become unfeasible because PL channels are very long. Hence, simpler linear or nonlinear suboptimal structures, like those proposed in [4], [13]–[15] are devised to deal with this problem.

Additionally, PL channels suffer from high power impulsive noises, which deteriorates the performance of linear equalizers. On the other hand, nonlinear devices, like the fuzzy filters described in section III, are well-suited for these channels.
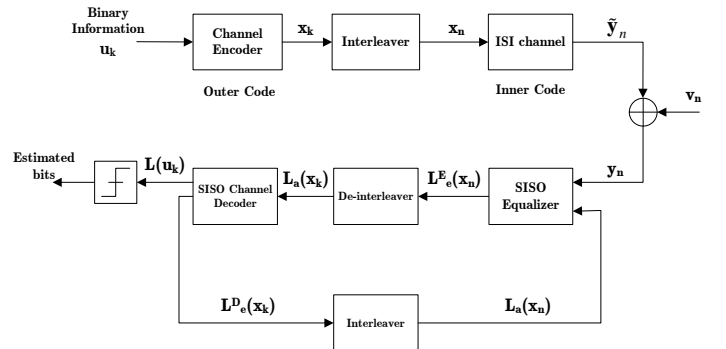

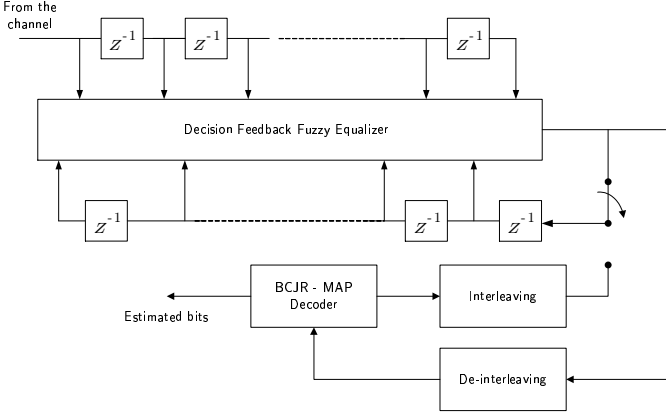Fig. 2. Transmitter and receiver in a turbo equalizer.

Fig. 3. Turbo-Fuzzy Equalizer.

Hence, in order to provide a low complexity solution without performance degradation, this paper proposes the introduction of a decision feedback nonsingleton type 1 fuzzy equalizer, discussed in Section III, into a turbo equalization scheme. The structure of the nonsingleton turbo-fuzzy equalizer is depicted in Fig. 3. One can note that *a priori* information is incorporated into the structure of the fuzzy equalizer as feedback inputs. As in a traditional turbo equalizer, during the first iteration there is no *a priori* knowledge about the received symbols. As a result, the feedback inputs in Fig. 3 are zero. However, to accelerate the convergence and reduce the computational load, a classical decision feedback structure is employed, so that the soft outputs from the equalizer are fed back to its inputs. In the ensuing iterations, the feedback information is provided by the decoder. In the other words, subsequent iterations offers to the fuzzy equalizer *a priori* information about the received symbols in accordance to the turbo principle. As a result, good performance can be attained as will be discussed in Section V.

Turbo-fuzzy equalizers require another change from traditional turbo systems. Indeed, in traditional turbo systems the soft information provided by the decoder is in the form of LLR of the interleaved coded bits, while the fuzzy equalizer expects estimated values of the transmitted symbols. Therefore, a mapping of the soft information in this feedback loop is required [13], [14]. This is possible by computing the expectation of each transmitted symbol. Using (12) and the fact that $P(x_n = +1|\mathbf{y}) + P(x_n = -1|\mathbf{y}) = 1$ for BPSK modulation, we have:

$$\bar{x}_n = \sum_{x \in B} x P\left(x_k = x|\mathbf{y}\right) = \tanh\left(\frac{L_a(x_n)}{2}\right) \qquad (13)$$

where $B = \{+1, -1\}$ is the modulation alphabet and $L_a(x_n)$ is the *a priori* information provided by the decoder. Although the soft outputs from the equalizer are not in LLR form, they contain the same information as the LLR, as there is a one-to-one mapping between both values.

V. PERFORMANCE EVALUATION

To evaluate the performance of the proposed turbo-fuzzy equalizers in PL channels, blocks of information bits were encoded by a rate $R = 1/2$ *recursive systematic convolutional* (RSC) code, with generator polynomials equal to 23 and 35, expressed in octal. The coded bits were BPSK modulated and permuted by a random interleaving of size $L = 10000$. The BCJR-MAP algorithm was used to implement the decoder. In all simulations, 10 turbo iterations per received data block were performed. We also used fuzzy equalizers with 10 forward and 10 feedback inputs and each input had 20 Gaussian membership functions. We considered that the noise was composed by white gaussian background noise and impulsive noise, modeled as in [16].

We considered data transmission through the channel whose frequency response is shown in Fig. 4. The *bit error rate* (BER) curves, computed as a function of the *signal to noise ratio* of the background noise, for four different turbo-fuzzy equalizers are shown in Fig. 5. In this figure, we can see that, after the last iteration, the nonsingleton turbo-fuzzy equalizer with variable step-size had the best performance, followed by the singleton equalizer with variable step-size and by the nonsingleton equalizer with fixed step-size. Also, we notice that the performance of these three equalizers are much better than that of a singlenton equalizer with fixed step-size that is equal to the normalized version of the turbo equalization technique proposed in [4].

For a BER of 10^{-3}, the nonsingleton DBD equalizer is approximately 2.5 dB better than the equalizers with fixed step-size and for a BER of 10^{-4}, the nonsingleton DBD equalizer is approximately 6 dB better than the singleton equalizer with fixed step-size, 2.5 dB better than the nonsingleton equalizer with fixed step-size and 1 dB better than the singleton DBD equalizer. Hence, we can conclude that the variation of the step-size can greatly improve the performance of fuzzy equalizers.

We plot in Fig. 6 the BER curves at the output of the decoder. We can observe that, except for the singleton equalizer with fixed step-size, the turbo-equalizers could recover the transmitted information without any error for E_b/N_0 greater than 10 dB. This clearly shows the advantage of using nonsingleton equalizers and variable step-size over singleton fixed step-size fuzzy equalizers.

VI. CONCLUSIONS

In this paper, we propose turbo-fuzzy equalizers to mitigate the detrimental effects of PL channels. We develop singleton and nonsingleton fuzzy equalizers with fixed and variable step-sizes and compare them to evaluate their performance when all of them are applied to PLC single-carrier systems. The comparison results reveal that the proposed nonsingleton turbo-fuzzy equalizer with variable step-size provides a considerable improvement compared with the other techniques. In fact, as stated before, nonsingleton fuzzy system is an appropriate technique to deal with problems in which the data are corrupted by noise. Another interesting contribution refers
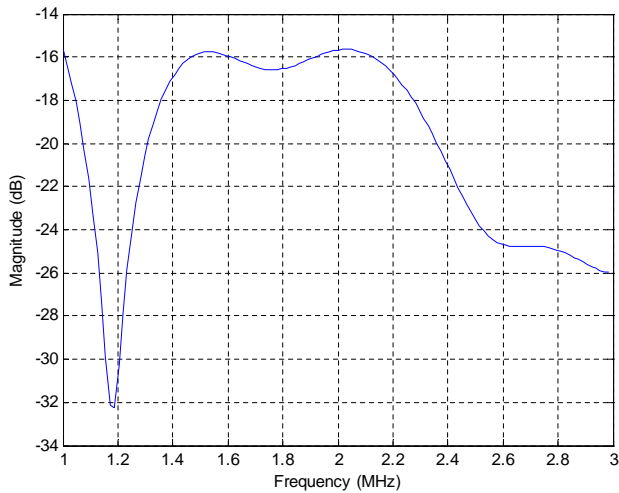
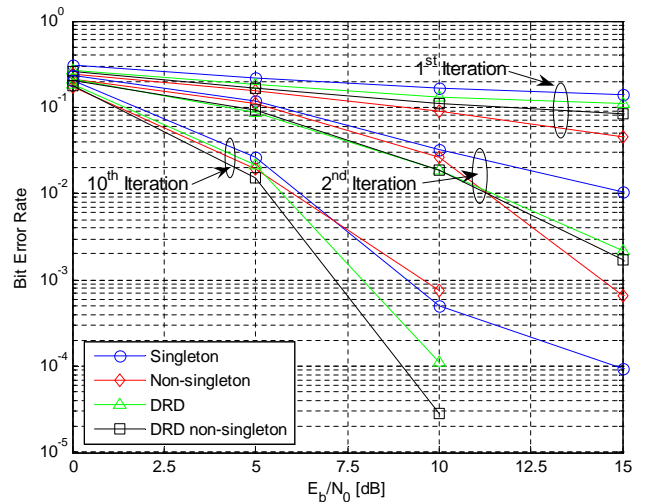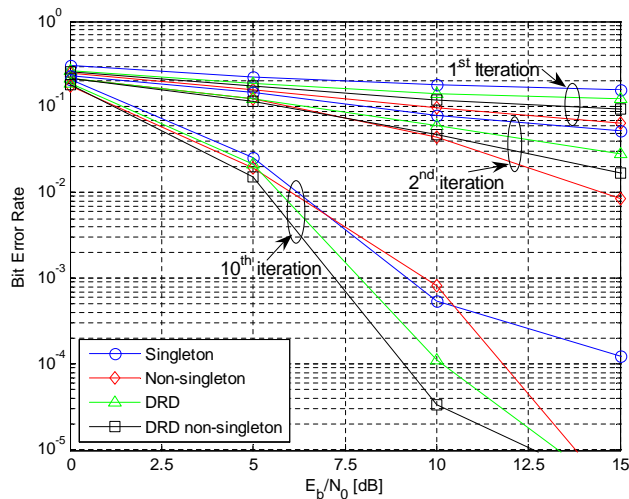Fig. 4.   Channel frequency response.

Fig. 6.   BER after the decoder.

Fig. 5.   BER after the equalizer.

to the fact that the use of variable step-size technique yield a fast convergence rate.

REFERENCES

[1] M. V. Ribeiro, *Telecommunications: Advances and Trends in Transmission, Networking and Applications*, chapter Power Line Communications: A Promising Communication System's Paradigm for Last Miles and Last Meters Applications, pp. 133–156, Fundação Edson Queiroz, 2006.

[2] H. Day and H. V. Poor, "Advanced signal processing for power line communications," *IEEE Communications Magazine*, pp. 100–107, May 2003.

[3] C. Douillard, M. Jézéquel, C. Berrou, A. Picart, P. Didier, and A. Glavieux, "Iterative Correction of intersymbol interference: Turbo-equalization," *European Transactions on Telecommunications*, vol. 6, no. 5, pp. 507–511, Sept.-Oct. 1995.

[4] M. S. Yee, B. L. Yeap, and L. Hanzo, "Radial Basis Function-Assisted Turbo Equalization," *IEEE Trans. on Communications*, vol. 51, no. 4, pp. 664–675, Apr. 2003.

[5] M. Zimmermann and K. Dostert, "A Multipath model for the powerline channel," *IEEE Trans. on Communications*, vol. 50, no. 4, pp. 553–559, Apr. 2002.

[6] M. V. Ribeiro, "Learning rate updating methods applied to adaptive fuzzy equalizers for broadband power line communications," *EURASIP Journal on Applied Signal Processing, Special Issue on Nonlinear and Image Processing*, vol. 2004, no. 16, pp. 2592–2599, 2004.

[7] S. Chen, B. Mulgrew, and S. McLaughlin, "Adaptive bayesian equalizer with decision feedback," *IEEE Trans. on Signal Processing*, vol. 41, no. 9, pp. 2918–2927, Sep. 1993.

[8] S. K. Patra and B. Mulgrew, "Efficient architecture for bayesian equalization using fuzzy filters," *IEEE Trans. on Circuits and Systems - II, Analog and Digital Signal Processing*, vol. 45, no. 7, pp. 812–820, 1998.

[9] L. X. Wang and J. M. Mendel, "Fuzzy adaptive filters, with application to nonlinear channel equalization," *IEEE Trans. on Fuzzy Systems*, vol. 1, pp. 161–170, 1993.

[10] G. C. Mouzouris and J. M. Mendel, "Nonsingleton fuzzy logic systems: Theory and design," *IEEE Trans. on Fuzzy Systems*, vol. 5, no. 1, pp. 56–71, Feb. 1997.

[11] R. A. Jacobs, "Increased rates of convergence through learning rate adaptation," *Neural Networks*, vol. 1, pp. 295–307, Sep. 1988.

[12] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate," *IEEE Trans. on Information Theory*, vol. IT-20, pp. 284–287, Mar. 1974.

[13] C. Laot, A. Glavieux, and J. Labat, "Turbo Equalization: Adaptive Equalization and Channel Decoding Jointly Optimized," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 9, pp. 1744–1752, Sept. 2001.

[14] M. Tüchler, R. Koetter, and A. C. Singer, "Turbo Equalization: Principles and New Results," *IEEE Trans. on Communications*, vol. 50, no. 5, pp. 754–767, May 2002.

[15] R. R. Lopes and J. R. Barry, "Soft-Output Decision-Feedback Equalization with a Priori Information," in *Proceedings IEEE Global Telecommunications Conference, GLOBECOM'03*, December 2003, vol. 3, pp. 1705–1709.

[16] M. Zimmermann and K. Dostert, "Analysis and modeling of impulse noise in broad-band powerline communications," *IEEE Trans. on Electromagnetic Compatibility*, vol. 44, no. 1, pp. 249–258, Mar. 2002.