USING THE CONSTANT MODULUS AND KULLBACK-LEIBLER COST FUNCTIONS ON A NONLINEAR PREDICTIVE STRUCTURE FOR BLIND EQUALIZATION

Charles C. Cavalcante$^{(1)}$, *João Cesar M. Mota*$^{(2)}$, *Jugurta R. Montalvão F.*$^{(3)}$ *and Bernadette Dorizzi*$^{(4)}$

(1) Universidade Estadual de Campinas (UNICAMP) - DECOM, charles@decom.fee.unicamp.br
(2) Universidade Federal do Ceará (UFC), mota@dee.ufc.br
(3) Universidade Tiradentes - Aracaju,SE , jugurta.montalvao@netdados.com.br
(4) Institut National des Télécomm.-France, Bernadette.Dorizzi@int-evry.fr

ABSTRACT

The use of a nonlinear structure of filtering for blind equalization is presented. The structure neural network-based is used in order to provide nonlinearity on the filter structure and the learning strategy is then divided in two stages. The Kullback-Leibler divergence is used as the base for the cost function of a self-organized rule and constant modulus criterion for the supervised one. Simulation results illustrate the performance of the strategy compared with classical ones for adaptive equalization. The results show that the proposed strategy outperforms even trained DFE for some cases of channels.

1. INTRODUCTION

It is well known that considering the hypothesis of i.i.d. transmitted symbols, the task of equalization can be done by means of prediction [1]. Regarding the problem of equalization through the use of prediction as a classification one some approaches can be deduced [2, 3, 4]:

- Equalization corresponds mapping on the input space to construct a class separation surface;
- The most suitable mapping is usually nonlinear (nonminimum phase channels case);
- Some channels cannot be equalized by means of linear structures such as transversal filters, using only 2$^{\text{nd}}$ order statistics.

Taking in account the above-referred characteristics the use nonlinearity on filter structure instead of only in the adaptation algorithm seems to be a reasonable choice for such a strategy that performs equalization. Considering the use of prediction, the problem of equalization becomes an interpolation one [4].

Once that we are searching a structure able to perform such an interpolation surface, we have proposed in [3] a predictive equalizer neural network-based which uses a combined learning strategy using a supervised algorithm based on minimization of prediction error and a self-organized algorithm based on minimization of Kullback-Leibler divergence. This strategy has permitted to equalize nonminimum phase (NMP) channels using only 2$^{\text{nd}}$ statistics. This structure have been called *Neural Predictive Structure* (NPS). Due to the use of minimization of prediction error the strategy is called *NPS Minimizing Prediction Error* (NPS-MPE).

Some works have proposed the use of a class of cost functions based on the minimization of the constant modulus criterion (CMC) for predictive equalization strategies [5]. Such good results have inspired us to use the minimization of the CMC on the previous proposed structure. This work investigates and compares the novel strategy, that we called *NPS using Constante Modulus Criterion* (NPS-CMC), with other equalization strategies, including the NPS-MPE.

Before presenting the contribution of this paper, we recall in Section 2 the structure of the NPS underlining the combined learning strategy. Section 3 presents the novel strategy and shows its principals differences from the NPS-MPE. Simulation results are presented in Section 4 in order to evaluate the performance faced to other strategies. Finally, our conclusions are presented in Section 5.

2. RECALLS ABOUT THE NPS

Let $a(n)$ a random variable (r.v.) drawn from a uniform distribution, which denotes the transmitted symbol, and the channel with impulse response given by

$$F(z) = \sum_{n=0}^{N-1} f_n z^{-n}, \tag{1}$$

we can also represent it in the vectorial notation $\mathbf{f} = \begin{bmatrix} f_0 & \cdots & f_{N-1} \end{bmatrix}^T$. Then, the noiseless channel outputs are then written as

$$\overline{\mathbf{x}}(n) = \mathbf{F}^H \mathbf{a}(n), \tag{2}$$

where $\mathbf{F}$ is the *channel convolution matrix* [2, 4] and $\mathbf{a}(n) = \begin{bmatrix} a(n) & a(n-1) & \cdots & a(n-N+1) \end{bmatrix}^T$. Thus, the equalizer inputs are written as:

$$\mathbf{x}(n) = \overline{\mathbf{x}}(n) + \mathbf{n}(n) \tag{3}$$

where $\mathbf{n}(n)$ is a vector of normal r.v. denoting noise.

The model of the generic predictive equalizer is depicted in Figure 1.

Concerning about the problem of prediction, which is related with the extraction of innovation, we observe that the use of a nonlinear strategy allows the use of ***only one*** input for the predictor, it means, we predict $x(n)$ through $T_{\text{NL}}(x(n-1))$, where $T_{\text{NL}}(\cdot)$ is a nonlinear transformation [3, 4].

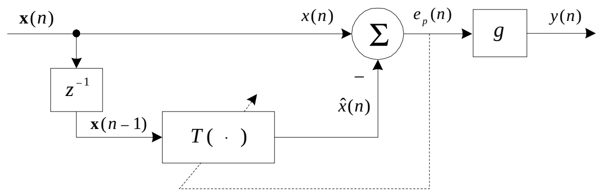

Figure 1: Model of the predictive equalizer.

In classification approach, this corresponds to find a function able to implement the separation between different classes. In digital communication, classes are set from the transmitted alphabet [2]. An example for binary alphabet is shown in Figure 2.

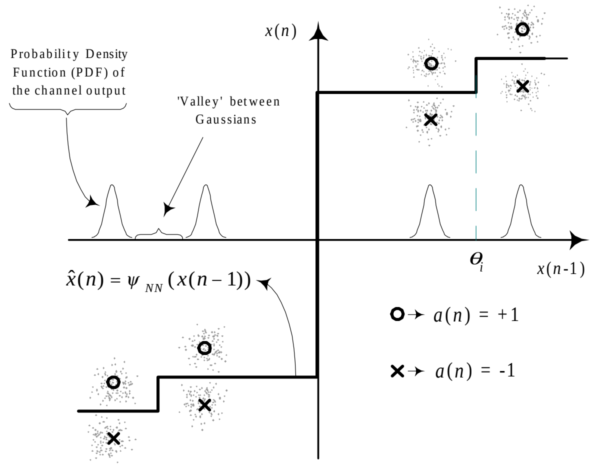

Figure 2: Separation surface for binary signals.

The design of an structure was based on the ability of implementing a function like that one represented in Figure 2 as $\psi_{NN}\left(x(n-1)\right)$.

Based on this model and on the approach of equalization as a classification problem we developed a predictive neural network-based for performing equalization. Figure 3 depicts the structure, where $\boldsymbol{\theta} = [\ \theta_1\ \cdots\ \theta_M\ ]^T$ is the bias of neurons vector, $\varphi_{\text{NPS}}(\cdot)$ is the signum function and $\boldsymbol{\beta} = [\ \beta_1\ \cdots\ \beta_M\ ]^T$ is the output linear coefficient vector.

Once established the structure to use we need to define the learning strategy which is responsible for the right convergence of the equalizer. As we will show, we separated the learning task in two subtasks

1. Learning of $\boldsymbol{\theta}$ parameters;
2. Learning of $\boldsymbol{\beta}$ parameters.

which will be described in summary on next subsections.

2.1. Self-Organized Algorithm

It is know that the assumption of a linear model for the channel and gaussian noise provides a mixture of Gaussians, centered on the channel noiseless outputs (called *channel states*), for the probability density function (pdf) of the received signal [2, 4]. As depicted in Figure 2 we see that fast transitions, for arising at the right function, are placed between the Gaussians of the pdf of received signal. For the first part of learning of NPS is how to find those "valleys" which are related to the neurons bias.

Thus, thinking about the measure of similarities between functions we arise at the Kullback-Leibler divergence cost function, which is well known in information theory field for measuring similarities between two positive defined functions. So, we have used the function $|x-\theta_i|$ to measure similarities with the pdf of received signal. This is a self-organized algorithm once the neural network does not need any *teacher* and is a kind of a Anti-Hebbian rule [7]. We called this algorithm SOFVA (*Self-Organized for Finding Valleys Algorithms*), and for further information about deducing the algorithm see [3, 4]. The cost function is then given by:

$$J_{\text{SOFVA}}(\boldsymbol{\theta}) = -\mathbb{E}\left\{\ln\left(|x-\boldsymbol{\theta}|\right)\right\}, \tag{4}$$

where κ is a term inserted to guarantee that the function is positive defined. The stochastic gradient of the algorithm for updating $\boldsymbol{\theta}$ parameters is in the form

$$\begin{aligned}\nabla J_{\text{SOFVA}}(\boldsymbol{\theta}) &= -\frac{\text{sgn}\left(x(n-1)-\boldsymbol{\theta}(n)\right)}{|x(n-1)-\boldsymbol{\theta}(n)|+\kappa} \\ \boldsymbol{\theta}(n+1) &= \boldsymbol{\theta}(n) - \lambda\cdot\nabla J_{\text{SOFVA}}(\boldsymbol{\theta}),\end{aligned} \tag{5}$$

where λ is the convergence step size.

2.2. Supervised Algorithm

For the optimization of the $\boldsymbol{\beta}$ parameters, we have chosen a classical LMS algorithm for prediction, the minimization of prediction error that corresponds to a whitening process on predictor output [6]. Naming this algorithm as supervised, we are referencing the neural network, that has a *teacher* (the prediction error) and not the equalizer once it is self-adapted (or blind).

Than, cost function for the supervised algorithm is

$$J_{\text{SUP-MPE}}(\boldsymbol{\beta}) = \mathbb{E}\left\{|e_p(n)|^2\right\} \tag{6}$$

and the stochastic version of the adaptation algorithm is given by:

$$\begin{aligned}e_p(n) &= x(n) - \psi\left(x(n-1), \boldsymbol{\theta}(n), \boldsymbol{\beta}(n-1)\right) \\ \nabla J_{\text{SUP-MPE}}(\boldsymbol{\beta}) &= e_p(n)\cdot\mathbf{r}(n) \\ \boldsymbol{\beta}(n+1) &= \boldsymbol{\beta}(n) + \mu\cdot\nabla J_{\text{SUP-MPE}}(\boldsymbol{\beta})\end{aligned} \tag{7}$$

where $\psi(\cdot)$ is the function implemented by the NPS, μ is the convergence step factor and $\mathbf{r}(n) = [\ r_1(n)\ \cdots\ r_M(n)\ ]^T$ are the neurons outputs (see Figure 3).

Then, the cost function for the NPS-MPE is

$$\begin{aligned}J_{\text{NPS-MPE}}\left(\boldsymbol{\theta},\boldsymbol{\beta}\right) &= J_{\text{SUP-MPE}}\left(\boldsymbol{\beta}\right) + J_{\text{SOFVA}}\left(\boldsymbol{\theta}\right) \\ J_{\text{NPS-MPE}}\left(\boldsymbol{\theta},\boldsymbol{\beta}\right) &= \mathbb{E}\left\{|e_p(n)|^2\right\} - \mathbb{E}\left\{\ln\left(|x-\boldsymbol{\theta}|\right)\right\}\end{aligned} \tag{8}$$

3. STRATEGY NPS-CMC

Inspired by the good results in [5] that provide faster convergence for predictive equalizers using the CMC than other strategies, we have investigated the use of another criterion for optimizing the $\boldsymbol{\beta}$ coefficients.

For the NPS-CMC the cost function for the supervised part is

$$J_{\text{SUP-CMC}}(\boldsymbol{\beta}) = \mathbb{E}\left\{\left(|y(k)|^2 - R_2\right)^2\right\}, \tag{9}$$

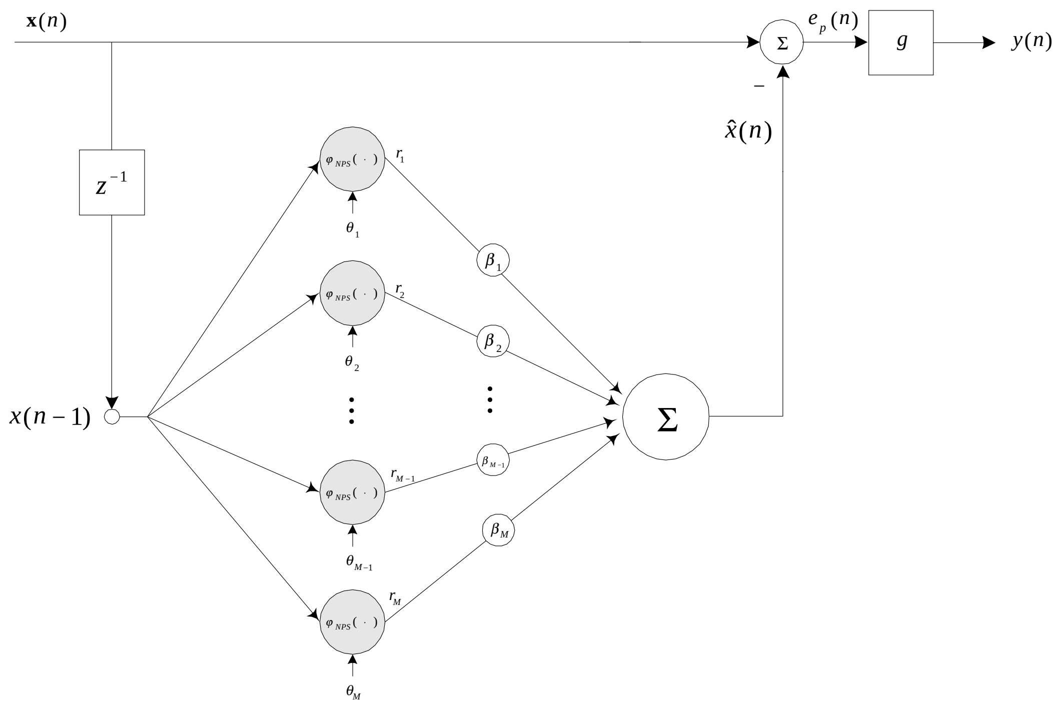

Figure 3: Neural Predictive Structure (NPS).

where $R_2 = \frac{\mathbb{E}\{|a^4(k)|\}}{\mathbb{E}\{|a^2(k)|\}}$ is the *equalization radius.* Equation 9 has a corresponding stochastic version given by:

$$\begin{aligned} e_p(n) &= x(n) - \psi\left(x(n-1), \boldsymbol{\theta}(n), \boldsymbol{\beta}(n-1)\right) \\ \nabla J_{\text{SUP-CMC}}(\boldsymbol{\beta}) &= g \cdot e_p(n) \cdot \left(R_2 - |g \cdot e_p(n)|^2\right) \\ \boldsymbol{\beta}(n+1) &= \boldsymbol{\beta}(n) + \mu \cdot \mathbf{r}(n) \nabla J_{\text{SUP-CMC}}(\boldsymbol{\beta}) \end{aligned} \tag{10}$$

where g is the automatic gain control (AGC) which is adapted by the algorithm [1],

$$\begin{aligned} G(n+1) &= G(n) + \mu_{\text{CAG}} \cdot \left(\sigma_a^2 - |y(n)|^2\right) \\ g(n) &= \sqrt{|G(n)|}, \end{aligned} \tag{11}$$

where the initialization is $G(0) = g(0) = 1$.

Then the NPS-CMC strategy is summarized by the following equation:

$$\begin{aligned} J\left(\boldsymbol{\theta}, \boldsymbol{\beta}\right) &= J_{\text{SUP-CMC}}\left(\boldsymbol{\beta}\right) + J_{\text{SOFVA}}\left(\boldsymbol{\theta}\right) \\ J\left(\boldsymbol{\theta}, \boldsymbol{\beta}\right) &= \mathbb{E}\left\{\left(|y(n)|^2 - R_p\right)^2\right\} - \mathbb{E}\left\{\ln\left(|x - \boldsymbol{\theta}|\right)\right\}. \end{aligned} \tag{12}$$

4. SIMULATION RESULTS

Simulations were done in order to compare the performance of the proposed nonlinear strategy with other equalization ones. To evaluate them, we have simulated over 100 Monte-Carlo trials the decision squared error (DSE), $\left((\text{Dec}((y(n)) - y(n))^2\right)$, where $\text{Dec}(\cdot)$ is the decision device, for a signal-to-noise ratio (SNR) defined in dB as $\text{SNR} = 10.\log_{10}\left(\frac{\sigma_a^2 \cdot \sum_{i=0}^{N-1} |f_i|^2 + \sigma_b^2}{\sigma_n^2}\right)$ where σ_a^2 and σ_n^2 are the power of transmitted symbols and noise respectively. In order to smooth the curves we have applied a lowpass filter with $\omega_c = 10^{-2}$.

In order to define the *status* of the channel, it means if it is in the “open eye” or “closed eye” situation, we will use the maximum distortion (MD) criterion [2, 4] given by:

$$\text{MD}(\mathbf{f}) = \frac{\sum_{k=0}^{N-1} |f_k| - \max_k |f_k|}{\max_k |f_k|}. \tag{13}$$

In this way, for $\text{MD}(\mathbf{f}) < 1$ we have the “open eye” situation and for $\text{MD}(\mathbf{f}) \geq 1$ we have the “closed eye” situation.

Other strategies were also simulated for comparing the performance, namely the constant modulus algorithm (CMA) with a transversal filter, linear prediction filter usinf 2nd order statistics and the trained decision feedback equalizer (DFE). As a reference for the linear structures the Wiener solution [6] is shown.

An important point to be discussed is relative to the initialization of the NPS-MPE and NPS-CMC. As the goal of the SOFVA is to find the “valleys” between the Gaussians on the pdf of received signal, the initialization plays an important role in this task once that, if the neurons are initialized far of the channel states, the algorithm will have an ill convergence. In order to solve that, we initialize a great number of neurons for cover all possible channel states values.

The first used channel has the following impulse response represented in the vectorial way $\mathbf{f}_1 = \begin{bmatrix} 1 & 0.6 & 0.2 \end{bmatrix}^T$ and respects the condition of “open eye”. Figures 4 and 5 show the evaluation of the DSE for different values of SNR.

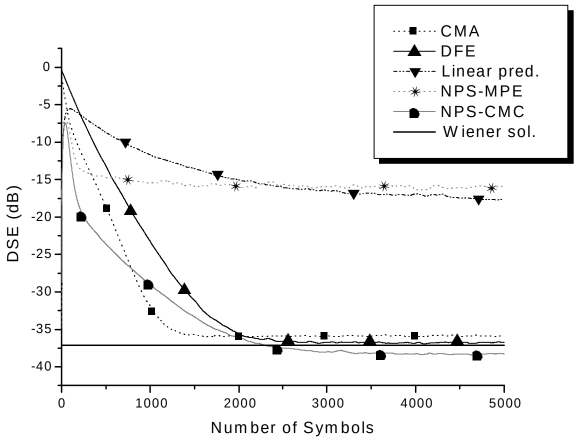

Figure 4: Comparison of DSE for equalization strategies using channel $\mathbf{f}_1$ - SNR $= 40$dB.

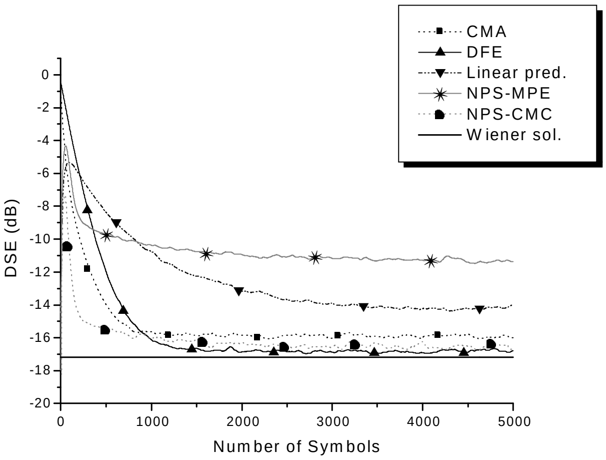

Figure 5: Comparison of DSE for equalization strategies using channel $\mathbf{f}_1$ - SNR $= 20$dB.

One can easily see that the NPS-CMC outperforms the other strategies, even the trained DFE in the case of a higher SNR. There is also a high significant increase of performance by the NPS-CMC over the NPS-MPE. This important result is due to the fact that the *teacher* for the NPS in the case of the CMC provides an error that tends to zero, what is not observed in the NPS-MPE case where the error tends to the alphabet symbols. Simulation parameters are given in Table 1.

The second used channel is given by $\mathbf{f}_2 = \begin{bmatrix} 1 & 0.8 & 0.4 \end{bmatrix}^T$ that presents the “closed eye” status and Figures 6 and 7 show the evaluation of the DSE for different values of SNR. Simulation parameters are given in Table 2.

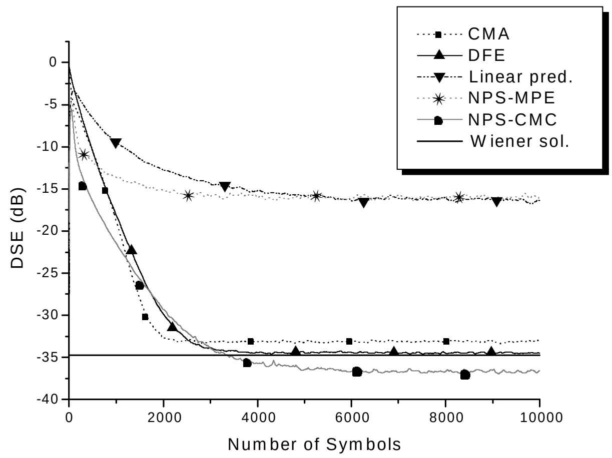

Figure 6: Comparison of DSE for equalization strategies using channel $\mathbf{f}_2$ - SNR $= 40$dB.

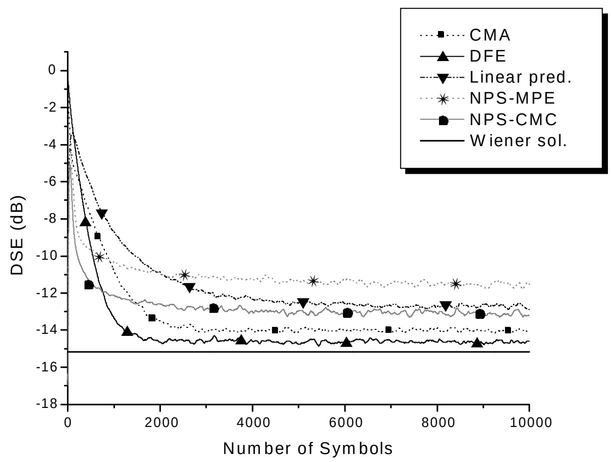

Figure 7: Comparison of DSE for equalization strategies using channel $\mathbf{f}_2$ - SNR $= 20$dB.

This case presents the same behavior than the previous one. The fact of the “closed eye” does not affect the performance of the NPS-CMC.

The two channels presented are minimum phase. For evaluating NMP channels we have used a channel with an impulse response $\mathbf{f}_3 = \begin{bmatrix} 0.5 & 1 & -0.6 \end{bmatrix}^T$. Figure 8 shows the evaluation for that channel and simulation parameters are given in Table 3.

In this case we observe no significant gain using NPS-CMC instead of NPS-MPE. The behavior of such a kind of channels using only one input limits the performance of the structure [4].

Both NPS strategies presents a higher computational complexity than the linear ones due to the number of neurons and also the combined learning strategy. We have observed that we can obtain an structure with a smaller number of neurons if we use the linearly spaced initialization for the $\boldsymbol{\theta}$ parameters.

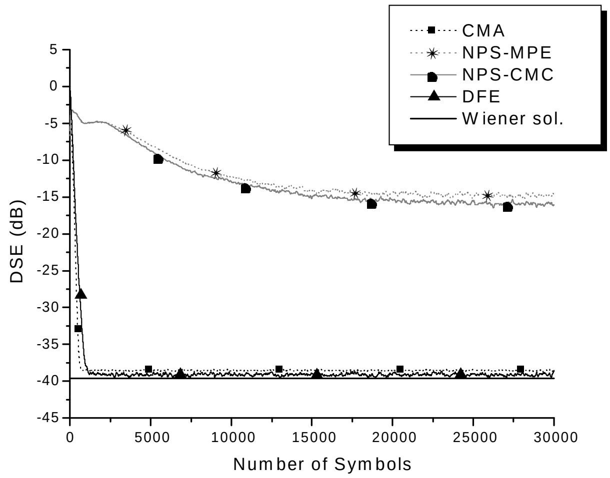

Figure 8: Comparison of DSE for equalization strategies using channel $\mathbf{f}_3$ - SNR $= 40$dB.

5. CONCLUSIONS

In this paper we have presented a neural predictive structure with a combined learning strategy with self-organized and supervised stages. In order to update the linear part of the structure we have used the constant modulus criterion.

The proposed strategy outperforms other equalization ones for minimum phase channels, consisting of an plausible alternative even for trained DFE strategy. For nonminimum phase channels the performance is not increased compared with the NPS-MPE, showing that the strategy using only one input is not good enough for every sort of channel.

The SOFVA algorithm is under more investigation for trying to accelerate it and make possible a regularization of neurons in order to decrease the computational complexity.

We also think about studying the semi-blind strategies using the NPS strategies which could lead good results.

6. REFERENCES

[1] Macchi O. *Adaptive Processing: The Least Squares with Applications in Transmission.* John Wiley & Sons, 1995.

[2] Montalvão, J.R.F., "Égalisation et identification de canaux de communication numérique: une approache par reconnaissance des formes et mélange de Gaussienes", Ph.D. Thesis (in French), Université Paris XI, France, November, 2000.

[3] Cavalcante, C.C., J.R. Montalvão Filho, B. Dorizzi and J.C.M. Mota, "A Neural Predictor for Blind Equalization of Digital Communication Systems", in Proceedings of Adaptive Systems for Signal Processing, Communications and Control, Lake Louise, Canada, October 2000.

[4] Cavalcante, C.C., "Predição Neural e Estimação de Função de Densidade de Probabilidade Aplicadas à Equalização Cega", Master Thesis (in Portuguese), Universidade Federal do Ceará (UFC), Brazil, February, 2001.

[5] Barbosa, L.J.M., J.C.M. Mota and F.R.P. Cavalcanti, "A Family of Predictive Constant Modulus Algorithms for Blind Equalization", in Proceedings of Adaptive Systems for Signal Processing, Communications and Control, Lake Louise, Canada, October 2000.

[6] Haykin, S., *Adaptive Filter Theory*, 3rd edition, Prentice-Hall, 1996.

[7] Haykin, S., *Neural Networks: A Comprehensive Foundation*, 2nd edition, Prentice Hall, 1998.

Strategy	Parameters
CMA	$\mu = 5 \cdot 10^{-3}$ Filter: 25 taps Initialization: $\mathbf{h} = \begin{bmatrix} 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{bmatrix}^T$
DFE	$\mu_h = 5 \cdot 10^{-3}$, $\mu_d = 10^{-4}$ Filter: 15 (foward) e 5 (feedback) Initialization: $\begin{cases} \mathbf{h} = \mathbf{0} \\ \mathbf{d} = \mathbf{0} \end{cases}$
Linear Predictor	$\mu_h = 10^{-3}$, $\mu_{\text{AGC}} = 10^{-3}$ Filter: 25 taps Initialization: $\mathbf{h} = \mathbf{0}$
NPS-MPE	$\lambda = 10^{-4}$, $\mu_\beta = 5 \cdot 10^{-4}$, $\mu_{\text{AGC}} = 10^{-3}$, $\kappa = 10^{-7}$ Number of neurons: 100 Initialization: $\begin{cases} \theta_i = U[-2, 2] \\ \boldsymbol{\beta} = \mathbf{0} \end{cases}$
NPS-CMC	$\lambda = 10^{-4}$, $\mu_\beta = 5 \cdot 10^{-4}$, $\mu_{\text{AGC}} = 10^{-3}$, $\kappa = 10^{-7}$ Number of neurons: 100 Initialization: $\begin{cases} \theta_i = U[-2, 2] \\ \boldsymbol{\beta} = \mathbf{0} \end{cases}$

Table 1: Simulation parameters for channel $\mathbf{f}_1$.

Strategy	**Parameters**
CMA	$\mu = 3 \cdot 10^{-3}$ Filter: 25 taps Initialization: $\mathbf{h} = \begin{bmatrix} 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{bmatrix}^T$
DFE	$\mu_h = 5 \cdot 10^{-3}$, $\mu_d = 10^{-4}$ Filter: 15 (foward) e 5 (feedback) Initialization: $\begin{cases} \mathbf{h} = \mathbf{0} \\ \mathbf{d} = \mathbf{0} \end{cases}$
Linear Predictor	$\mu_h = 10^{-3}$, $\mu_{\text{AGC}} = 10^{-3}$ Filter: 25 taps Initialization: $\mathbf{h} = \mathbf{0}$
NPS-MPE	$\lambda = 10^{-4}$, $\mu_\beta = 5 \cdot 10^{-4}$, $\mu_{\text{AGC}} = 10^{-3}$, $\kappa = 10^{-7}$ Number of neurons: 100 Initialization: $\begin{cases} \theta_i = U[-2, 2] \\ \boldsymbol{\beta} = \mathbf{0} \end{cases}$
NPS-CMC	$\lambda = 10^{-4}$, $\mu_\beta = 5 \cdot 10^{-4}$, $\mu_{\text{AGC}} = 10^{-3}$, $\kappa = 10^{-7}$ Number of neurons: 100 Initialization: $\begin{cases} \theta_i = U[-2, 2] \\ \boldsymbol{\beta} = \mathbf{0} \end{cases}$

Table 2: Simulation parameters for channel $\mathbf{f}_2$.

Strategy	**Parameters**
CMA	$\mu = 5 \cdot 10^{-3}$ Filter: 25 taps Initialization: $\mathbf{h} = \begin{bmatrix} 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{bmatrix}^T$
DFE	$\mu_h = 5 \cdot 10^{-3}$, $\mu_d = 10^{-4}$ Filter: 25 (foward) e 5 (feedback) Initialization: $\begin{cases} \mathbf{h} = \mathbf{0} \\ \mathbf{d} = \mathbf{0} \end{cases}$
NPS-MPE	$\lambda = 10^{-4}$, $\mu_\beta = 5 \cdot 10^{-4}$, $\mu_{\text{AGC}} = 10^{-3}$, $\kappa = 10^{-7}$ Number of neurons: 100 Initialization: $\begin{cases} \theta_i = U[-2.5, 2.5] \\ \boldsymbol{\beta} = \mathbf{0} \end{cases}$
NPS-CMC	$\lambda = 10^{-4}$, $\mu_\beta = 5 \cdot 10^{-4}$, $\mu_{\text{AGC}} = 10^{-3}$, $\kappa = 10^{-7}$ Number of neurons: 100 Initialization: $\begin{cases} \theta_i = U[-2.5, 2.5] \\ \boldsymbol{\beta} = \mathbf{0} \end{cases}$

Table 3: Simulation parameters for channel $\mathbf{f}_3$.