PREDISTORTION OF NON-LINEAR SATELLITE CHANNELS USING NEURAL NETWORKS: ARCHITECTURE, ALGORITHM AND IMPLEMENTATION

F. Langlet, H. Abdulkader and D. Roviras

ENSEEIHT-TeSA, 2 rue C. Camichel 31071 Toulouse cedex France

ABSTRACT

This paper presents the adaptive linearisation of a non-linear digital satellite communication down link. That link is made up a 16-QAM modulator, followed by a non-linear High Power Amplifier, on board the satellite. When using the amplifier with low input back-off for a maximum power efficiency, two kinds of distortions occur on the input signal: amplitude (AM/AM conversion) and phase (AM/PM conversion). The satellite payload is regenerative. So, we use a predistortion on board to linearize the amplifier. We present the predistortion architecture realized with Multi-Layer Perceptron (MLP) Neural Networks (NN). Two algorithms associated to that predistorter are shown and compared: the ordinary and the natural gradient. The major problem to implement that predistorter is to get enough bandwidth (100 Mbits/s data rate). A mixed analog/digital implementation is one solution to solve it. We analyze the implementation imperfections effects in comparison with the theoretical algorithm.

1. INTRODUCTION

In order to comply to the consumer needs, the third generation of mobile communication systems will have to combine spectral efficiency and allow high data rates and power efficiency to allow transmission to small user terminals. To get power efficiency, the on board amplifier such as Solid State Power Amplifier (SSPA) is used with low input back-off, i.e. the amplification is non-linear. To get spectral efficiency, non constant envelope modulation (16-QAM) is necessary. This combination involves problems such as spectrum spreading, constellation distortion, inter-modulation products, etc.. To solve those problems, two solutions are available: equalization [2] and predistortion[5][6]. New satellite generations have regenerative payloads. Baseband signals I and Q are available on board. So, it is possible to put a predistorter on board, between the baseband signal and the down link input (i.e. the modulator input). Therefor, the regenerative payload make the predistortion attractive to counteract the non-linear distortions caused by satellite amplifiers. In this paper, we present predistortion for regenerative payloads. Part two will describe the architecture of the satellite payload with the non-linear amplifier together with the predistorter. Part three presents algorithms for training the MLP NN. Part four deals with implementation issues.

2. ARCHITECTURE

Fig. 1 presents the predistortion. The predistortion baseband input signal (I+j.Q) is the digital signal to transmit, mapped in 16 QAM, then filtered by a Square Root Raised Cosine Filter (SRRCF). To overcome the SSPA non-linearity, that signal is predistorted by a NN (I_E+j.Q_E), further modulated and amplified by the SSPA. In the back loop, the signal is demodulated (I_R+j.Q_R) . We use the gradient algorithm in order to adapt NN predistorter weights. Finally, the overall system transfer function is closed to identity (I+j.Q≈ I_R+j.Q_R).

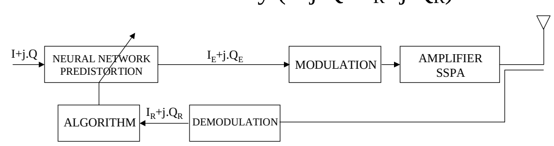

Fig. 1. System architecture

2.1. Memoryless SSPA

This amplifier gives amplitude (Fig. 8a) and phase (Fig. 9) distortions on the signal complex envelope (I_E+j.Q_E). The predistortion system is made up three parts (Fig. 2): a cartesian/polar conversion, the predistortion and the polar/cartesian conversion. The cartesian/polar conversion is realized with two MLP NN. One for the modulus (NN3) and one for the inverse modulus (NN4). For the predistortion, there are two MLP NNs. One for the inversion of the AM/AM conversion (NN1) and one for the AM/PM conversion (NN2). The Fig. 3 zoom the predistortion part of the predistorsion system. The three NNs (NN1, NN2 and NN4) have the same architecture described in Fig. 4: it is a MLP with one input and one output, one hidden layer (10 neurons) and one linear neuron in the output layer. The fourth NN (NN3) has the same architecture as the others except there are two inputs (I and Q). The polar/cartesian conversion consists in approximating the two functions (cos and sin) with their first order limited development.

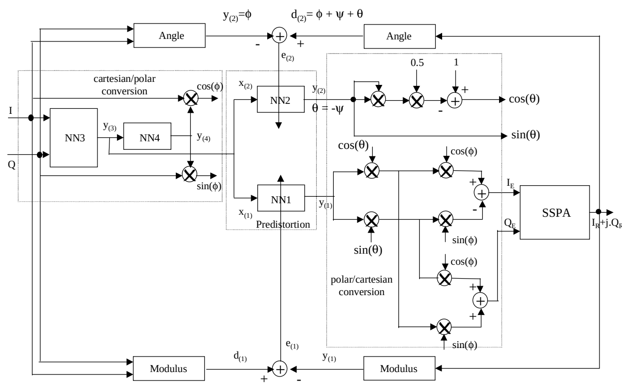

Fig. 2. Predistortion system

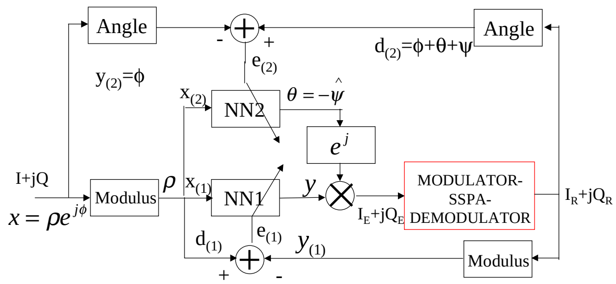

Fig. 3. Predistortion

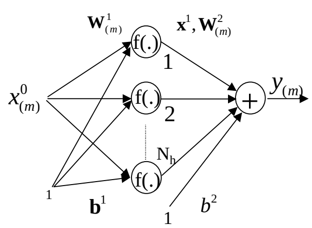

Fig. 4. MLP Neural Network structure

2.2. SSPA with memory

The SSPA with memory is modeled by a memoryless SSPA followed by a filter (the memory). To predistort it, one solution is to identify separately the memoryless SSPA and the memory to come down to the SSPA memoryless predistortion case. That is done by the SSPA with memory identification with a MLP NN mimetic structure made up two parts [5]. The first one has the same architecture than the memoryless SSPA, it is non-linear and computes the AM/AM and AM/PM conversion with two sub-networks. The second part presents the architecture of a linear filter. The predistortion performance is directly linked to the identification precision. More particularly, it depends of the associated algorithm. Two different associated algorithms are used: the ordinary and the natural gradient [4]. In the next section, we present and compare those algorithms (Fig. 5).
Details of predistorter architecture design can be found in [5].

3. ALGORITHMS

3.1. Ordinary gradient

The transfer function for one NN can be written as:

$$y_{(m)}(n) = \mathbf{W}^2_{(m)}(n).f\left[\ \mathbf{W}^1_{(m)}(n).\mathbf{x}^0_{(m)}(n) + \mathbf{b}^1_{(m)}(n)\ \right] + \mathrm{b}^2_{(m)}(n),$$

where $\mathbf{W}^k_{(m)}$ and $\mathbf{b}^k_{(m)}$ correspond respectively to the k^{th} layer vector and bias vector of m^{th} NN (m=1,2,3,4 in Fig. 2). f [.] is an activation function (a sigmoid). The ordinary gradient can be found in [3]. It is summarized bellow.

1) For NN2, NN3 and NN4 (m=2,3,4):

$$d_{(2)}(n) = \phi(n) + \psi(n) + \theta(n)$$

$$d_{(3)}(n) = \sqrt{I^2(n) + Q^2(n)}$$

$$d_{(4)}(n) = 1/d_{(3)}(n)$$

Error calculation : $e_{(m)}(n) = d_{(m)}(n) - y_{(m)}(n)$.

Update of the 2nd layer synaptic weights and biases:

$$w^2_{1k(m)}(n+1) = w^2_{1k(m)}(n) + \mu.e_{(m)}(n).x^1_{k(m)}(n)$$

$$b^2_{11(m)}(n+1) = b^2_{11(m)}(n) + \mu.e_{(m)}(n)$$

Update of the 1st layer synaptic weights and biases:

$$E_{k(m)}(n) = w^2_{k(m)}(n).e_{(m)}(n)$$

$$\Delta_{k(m)}(n) = E_{k(m)}(n).f'(w^1_{k(m)}(n).x^0_{(m)}(n) + b^1_{k(m)}(n))$$

$$w^1_{k1(m)}(n+1) = w^1_{k1(m)}(n) + \mu.\Delta_{k(m)}(n).x^0_{(m)}(n)$$

$$b^1_{k(m)}(n+1) = b^1_{k(m)}(n) + \mu.\Delta_{k(m)}(n)$$

$1 \le k \le N_h$, N_h is the number of neurons in the hidden layer and μ is the learning rate.

2) For NN1:
The ordinary gradient can be computed only if we have the derivative function of the AM/AM Conversion. One solution is to identify this conversion with an other NN associated to the ordinary gradient algorithm. In that case, we can use the identified model in order to get an estimation of the derivative function. So, the AM/AM conversion identification makes the predistortion technique adaptive [6].

3.2. Natural gradient

NN training procedure is a stochastic process in general. A NN with a certain structure can be viewed as a Riemannian manifold spanned by its coefficient vector. Each point of the manifold represents a probability distribution p(x,y,θ), where *x* is the input vector, *y* is the target vector and θ is the vector of NN coefficients considered as the manifold coordinates. The NN manifold is a submanifold of the manifold of all the distributions p(x,y). From a geometrical point of view, the ordinary gradient is a covariant vector. To apply the ordinary gradient for training the NN, it must be transformed by a contravariant tensor (the metric of the manifold) to form a contravariant vector. The matrix metric of the manifold is the Fisher information matrix given by the relation [1]:

$$g_{ij} = E\left[\frac{\partial \log p(x,y,\theta)}{\partial \theta_i}\frac{\partial \log p(x,y,\theta)}{\partial \theta_j}\right]$$

Hence the updating rule of the coefficient vector becomes:

$$\theta(n+1) = \theta(n) - \mu G \nabla l \,,$$

where μ is the learning rate and ∇l is the ordinary gradient vector. Details of the natural gradient algorithm can be found in [8]. We compare both algorithms applied to the considered down link channel identification, for a SSPA with memory. Fig. 5 presents the Mean Square Error (MSE) evolution. With the natural gradient, the identification performance is about 25 dB better in comparison with the ordinary gradient [4].

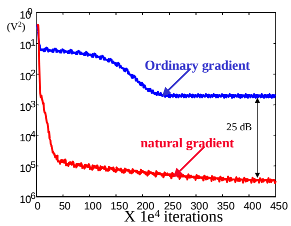

Fig. 5. MSE evolution with ordinary and natural gradient

4. IMPLEMENTATION

4.1. Introduction

The MLP NN implementation approach is determined by the high data rate. In fact, with a 100 Mbits/s data rate and a 16-QAM modulation, the maximum bandwidth of the input signal (I+j.Q) is 25 MHz (when the SRRCF roll off is 1). Moreover, the MLP NN transfer function digital implementation (30 multiplication, 11 additions and 10 hyperbolic tangents) must be computed in 40 ns (for one sample by symbol). Actually, it is impossible to realize this computation in such a time. So, the MLP implementation must be analog. For the algorithm, the down link variations (which depend of the temperature and the ageing process), are very slow in comparison with the data rate. To get the good down link predistortion pursuit (for an adaptive predistortion), the updating weights and biases cycle is 0.25 hour. In that case, we have the choice between an analog or digital implementation. We chose the digital solution for two reasons: The precision of computation and the flexibility [7]. As the implementation of the system is mixed, we decided to implement it in a CMOS technology. Fig. 6 presents the mixed implementation of the MLP NN associated to the ordinary gradient algorithm. In order to model the most important imperfections witch are linked to the implementation, we consider each part of the algorithm in the next sub sections [6].

4.2. Weights and biases loading (n^{th} iteration)

Each weight and bias is analog (capacitive storage). New values are loaded via N bits Digital Analog Converter (DAC). The DAC imperfections are modeled by two parts: weights and biases coding in signed fixed point format and the quantification. The algorithm robustness is tested by varying N [6].

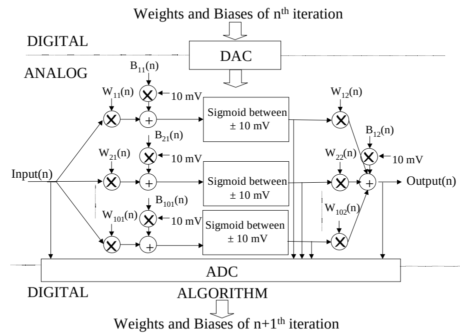

Fig. 6. NN mixed implementation

4.3. Forward propagation and storage

The multiplication function (used for the weights and the biases) is implemented with a Gilbert multiplier MOS version (Fig. 7a). The sigmoid function is implemented with a simple MOS differential stage (Fig. 7b). Both of these imperfections are modeled. Then, we test the algorithm robustness by varying the multiplier non-linearity and the sigmoid linear slope [6]. Finally, we store (capacitive storage) all the values in order to compute the algorithm: the NN input and output, the sigmoid outputs. To close this phase, we define the technology. To get the NN bandwidth, a transistor with small channel length is needed. The NN voltage dynamic range is around 1 V. We choose a 0.18 μm transistor (the maximum dynamic is 1.8 V).

4.4. Back propagation

This part of the algorithm is digital. The weights and biases update are computed with the stored values we get in the previous phase. To convert those values in digital, we model a N bits Analog Digital Converter (ADC). The ADC imperfections are: the quantification step and the signed fixed point coding. We also test the algorithm robustness by varying N [6]. At the end of this part we get the new weights and biases for the n+1th iteration.

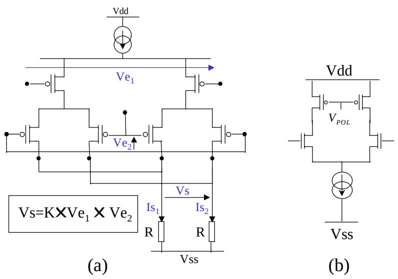

Fig. 7. (a) multiplier and (b) sigmoid implementation

4.5. Global simulation

In this simulation, we present the down link linearization, for a memoryless SSPA, by predistortion. All the mixed implementation predistorter imperfections are modeled. For the DAC and the ADC, the number of bits N is set to 12. For the Gilbert multiplier, the maximum non linearity error is set to 5 %. For the MOS sigmoid, the maximum slope variation is set to 5%. After 2.10^6 iterations, the NNs have converged. We present respectively on Fig. 8a, 8b the down link AM/AM conversion without and with predistortion. We also present on Fig. 9 the down link AM/PM conversion with and without predistortion. We define the Signal to Error Ratio SER_I (SER_Q) for the I (Q) component, where the signal is the input predistortion I (Q) Component. The error is the difference between the input predistortion I (Q) and the output down link I_R (Q_R) (see Fig. 1). The SER for the I and Q components at the end of convergence is:

$$SER_I = SER_Q = 38dB \cdot$$

Without predistortion we get

$$SER_I = SER_Q = 20dB \cdot$$

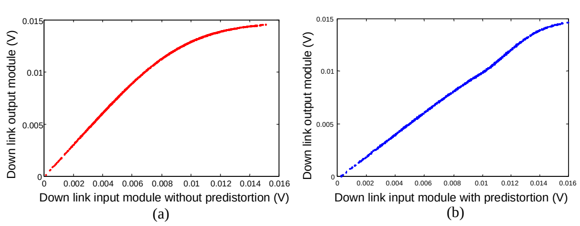

Fig. 8. (a) Down link AM/AM conversion and (b) linearized down link AM/AM conversion

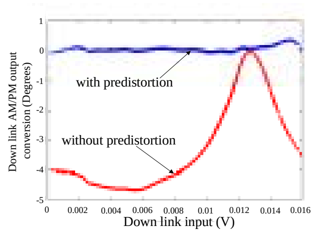

Fig. 9. Linearized down link AM/PM conversion

5. CONCLUSION

We have Presented a new adaptive predistortion for a down link digital communication with 100 Mbits/s data rate.

For the algorithm part, we shown that a NN associated to the ordinary gradient algorithm is a good approach to linearize a memoryless down link. For a down link with memory, the predistortion performance depends on the precision of the down link identification. To get enough precision, one solution consists to identify that link with a NN mimetic structure associated to the natural gradient algorithm. Moreover, this algorithm increases the precision of the identification of 25 dB in comparison with the ordinary gradient.

As regards to implementation, we have shown that the mixed analog/digital approach is forced by the 100 Mbits/s data rate. We have modeled the majors implementation imperfections i.e. the DAC, ADC, multiplier and sigmoid. Simulation analysis shows that the algorithm is enough robust to support all the implementations imperfections. Moreover, the predistortion increase the SER by 18 dB.

6. REFERENCES

[1] S. I. Amari, “Natural Gradient Works Efficiently in Learning”, Neural Computing 10,251-276 (1998).

[2]. S. Bouchired, M. Ibnkahla, D. Roviras and F. Castanié, '”Equalization of satellite mobile channels with neural network techniques”, Space Communications 15, IOS Press, pp. 209-220, 1998/99.

[3] Simon Haykin, '”*Neural Network: A Comprehensive Foundation*'”, Prentice Hall, 1994.

[4] F. Langlet, H. Abdulkader, D. Roviras and F. Castanié, “Comparison of Neural Network Natural and Ordinary Gradient Algorithms for satellite Down link Identification”, Proc. of ICASSP 2001, Salt lake City.

[5] F. Langlet, H. Abdulkader, D. Roviras, A. Mallet and F. Castanié, “Comparison of Neural Network Adaptive Predistortion Techniques for Satellite Down Links”, Proc. of IJCNN 2001, Washington DC, USA, July 2001.

[6] F. Langlet, D. Roviras, A. Mallet and F. Castanié, “Mixed Analog/Digital implementation of MLP NN for Predistortion” ”, Proc. of IJCNN 2002, Hawaii, USA, May 2002.

[7] F. Langlet, M. Ibnkahla, and F. Castanié, “Neural Network Hardware Implementation : Overview and applications to satellite Communications”, Proc. of DSP’98, ESA, Nordwick, Holland, September 1998.

[8] H.H. Yang and S.I. Amari, “Training Multi-Layer Perceptrons by Natural Gradient descent”, In ICONIP’97 Proceeding, new Zealand.