# Complex-Valued Neural Networks with Adaptive Spline Activation Function for Digital Radio Links Nonlinear Equalization

Aurelio Uncini, *Member, IEEE*, Lorenzo Vecci, *Member, IEEE*, Paolo Campolucci, *Student Member, IEEE*, and Francesco Piazza, *Member, IEEE*

***Abstract*—In this paper, a new complex-valued neural network based on adaptive activation functions is proposed. By varying the control points of a pair of Catmull–Rom cubic splines, which are used as an adaptable activation function, this new kind of neural network can be implemented as a very simple structure that is able to improve the generalization capabilities using few training samples. Due to its low architectural complexity (low overhead with respect to a simple FIR filter), this network can be used to cope with several nonlinear DSP problems at a high symbol rate.**

**In particular, this work addresses the problem of nonlinear channel equalization. In fact, although several authors have already recognized the usefulness of a neural network as a channel equalizer, one problem has not yet been addressed: the high complexity and the very long data sequence needed to train the network. Several experimental results using a realistic channel model are reported that prove the effectiveness of the proposed network on equalizing a digital satellite radio link in the presence of noise, nonlinearities, and intersymbol interference (ISI).**

## I. Introduction

SINCE ON-BOARD satellite power is a precious resource, in order to maximize efficiency of digital radio links to have a high efficiency, the transmitter high-power amplifier (HPA) operates near the saturation point; this introduces nonlinearities that can, in turn, cause serious degradation of the received signal [1].

The nonlinearity of a typical HPA, a traveling-wave tube (TWT), or a GaAs FET amplifier affects both amplitude (AM/AM conversion) and phase (AM/PM conversion) of the amplified signal and can be considered as memoryless, i.e., the HPA is a nonlinear system without memory under a wide range of operational conditions [2]. However, in practice, the transmitter contains a pulse-shaping circuit (modulator) at the baseband or at the intermediate frequency (IF) stage in virtually all digital radio systems. Therefore, the overall baseband-equivalent system (the cascade of transmitter, HPA nonlinearity, and receiver) is a nonlinear system with memory.

It is well known that the 16- and 64-QAM links are very sensitive to nonlinear distortion, although they can exhibit better bit-error rate (BER) performance than equivalent phase-shift keying (PSK) systems on additive white Gaussian noise channels. The effects of the nonlinear channel with memory on the QAM signals are manifold, but three of them are of particular importance.

Spectral Spreading: The spectrum of the amplified signal after the HPA is much wider than that of the signal before the HPA, due to the presence of nonlinearity. The output signal may not comply with the design limit on the transmitted power spectral density, as prescribed by the FCC mask for common carrier radio channels. This may cause intolerable interchannel interference, which must be removed by a radio frequency (RF) filter with the consequent loss in signal level and increased distortion.

Intersymbol Interference (ISI): Since the overall system has memory, each symbol of the QAM alphabet, which is usually referred to as a constellation point, is received as a cluster of points due to the interference among symbols at the sampling instants.

Constellation Warping: The respective centers of gravity of the clusters caused by the ISI are no longer on a rectangular grid as in the original constellation.

Two different approaches have been followed to cope with the nonlinearity of the link. The first is based on the idea of accepting the nonlinear channel as it is, without modifying its behavior, and designing a receiver that minimizes the effects of ISI, nonlinearities, and noise. An optimum technique based on this approach is the maximum likelihood sequence estimation (MLSE) by the Viterbi algorithm [3]. However, its processing complexity has suggested several less-demanding suboptimal equalization schemes that are usually based on the Volterra series expansion of the nonlinear channels [4], [5], whose major limitation is its high sensitivity to noise. In fact, the channel noise tends to be enhanced by any receiver-side nonlinear inverse filter that compensates the channel distortion [6], [7].

The second approach works on the compensation of the nonlinearity before the addition of noise, i.e., at the transmitter side of the channel. The techniques based on this approach, which are generally referred to as predistortion techniques, try to linearize the HPA characteristic by predistorting the input signal in such a way that the cascade predistorter–amplifier–receiver resembles as closely as possible an ISI-free channel [6]–[10].

Manuscript received July 23, 1997; revised April 6, 1998. The associate editor coordinating the review of this paper and approving it for publication was Prof. Yu-Hen Hu.

The authors are with the Dipartimento di Elettronica e Automatica, Università di Ancona, Ancona, Italy (e-mail: aurel@eealab.unian.it).

Publisher Item Identifier S 1053-587X(99)00146-4.

Although the second technique is more general than the equalization at the receiver side, it requires a processing complexity that makes implementation using digital signal processors (DSP's) unsuitable for very high data rates [36].

In this paper, we have only considered the first approach to evaluate the performance of the new proposed neural network architecture because the predistorter could be analyzed under a similar approach.

Nonlinear communication channel equalization using real or complex-valued neural networks (NN's) has been deeply studied in the last few years, and several authors have recognized the usefulness of recurrent [11] or multilayer perceptron (MLP) NN's architectures [12]–[15]. One of the main problems involved in using these techniques is the very long data sequence required in the learning phase that, in turn, leads to an increase in the adaptation errors when tracking a time-varying channel.

Other NN models for nonlinear channel equalization have also been studied. In [16], a structure based on a polynomial-perceptron structure is proposed, whereas in [17], the authors have proposed a functional-link approach. The main problem of such solutions is in the high number of free coefficients required when the degree of nonlinearity becomes very high.

Recently, a new family of NN's based on adaptive activation functions adaptive spline neural networks (ASNN's) was proposed in [18], whereas in [37], the authors show the universal approximation capabilities and the regularization properties. They also show that an intelligent use of the adaptive activation function can reduce the number of synaptic interconnections [35].

In [19] and [20], it is shown that for binary classification problems, the expected value of the generalization error has an asymptotic upper bound of $W/M$, where $W$ is the total number of free parameters (in our case, weights and activation function adaptable parameters), and $M$ represents the number of training examples. The generalization error capabilities are mainly related to the total number of free parameters. As a consequence, it is possible to move some free parameters from the connection weights to the activation function and, therefore, the computation resources since splines have a low overhead compared with standard activation functions.

The main advantages of this innovative structure, which are very useful for nonlinear adaptive signal processing, are as follows:

1) The training sequence may be shorter than that required by the classical MLP.
2) The architecture is general and, unlike other approaches, it does not require a specific design.
3) The low hardware complexity (low overhead with respect to a simple adaptive linear combiner) makes it suitable for high-speed data transmissions using a DSP device.

In this paper, we propose a complex extension of the ASNN and investigate the application of this new neural network model to the adaptive channel equalization problem. Our goal is to design a receiver that compensates HPA nonlinearities in digital radio links and extracts the symbols from the received data (demodulation process) when a QAM is used.

## II. The Complex Adaptive Spline Neural Networks

### A. Basic Considerations

The classical neuron computes the weighted sum of its inputs and feeds it into a nonlinear function called activation function [21]. The behavior of a neural network built with such neurons, as in the MLP, thus depends on the chosen activation functions. Sigmoids are commonly used for this purpose [22]. Different classes of nonlinear activation functions, depending on some free parameters, have also been widely studied and applied (see, for example, [23] and [24]).

Recently, a simple solution involving the adaptation of the gain and the slope of the sigmoidal activation function during the learning process was proposed [25]. Comparisons with classical MLP in modeling static and dynamic systems have shown that an adaptive sigmoid in a single hidden layer structure leads to an improved data modeling.

In [26], the use of adaptive polynomial functions was proposed; in particular, it has been shown that networks composed by polynomial neurons are isomorphic to conventional polynomial discriminant classifiers or Volterra filters, thus retaining their approximation capabilities. Moreover, such a solution makes it possible to reduce the size of the network, trading connection complexity for activation function complexity. When digitally implemented, the overall complexity of the resulting MLP can be reduced since the computation of the activation function does not depend heavily on the shape of the function itself because it is performed through a look-up table (LUT).

Any polynomial function, however, is not a squashing function (it is not bounded) and can have problems with spurious minima and maxima when its coefficients are adapted during the learning phase [26]. In [27], the direct adaptation of the LUT coefficients is proposed; whereas the reported experiments show a large reduction of the network size, nevertheless, the adaptation process can be difficult mainly because of the large number of adaptive parameters and the loss of smoothness in the resulting functions.

In [18], the ASNN is introduced. This spline-based NN is designed using a neuron called the generalized sigmoidal (GS) neuron, which contains an adaptive parametric spline activation function with several interesting features:

1) It is easy to adapt.
2) It retains the squashing property of the sigmoid.
3) It has the necessary smoothing characteristics.
4) It is easy to implement both in hardware and in software simulations.

Multilayer networks built with such neurons are still universal approximators and usually have a smaller structural complexity, together with good generalization capabilities.

### B. Complex-Valued Activation Function

The advantage of using complex-valued NN's instead of a real-valued NN counterpart fed with a pair of real values is

well known [28], [29]. In complex-valued neural networks, one of the main problems is the complex domain activation function, whose most important features are suggested in [30]. Let $F(S)$ be the complex activation function with $S \in \mathbf{C}$ defined as the complex linear combiner output; the main constraints that $F(S)$ should satisfy are the following:

1) $F(S)$ should be nonlinear and bounded.
2) In order to derive the backpropagation algorithm, the partial derivatives of $F(S)$ should exist and be bounded.
3) Because of *Liouville's theorem* $F(S)$ should not be an analytic function (see [31] for more details).

According to the previous properties, one possible choice, which was proposed in [14] and [32], consists of the superimposition of a real and an imaginary activation function

$$F(S) = f_{\mathrm{Re}}(\mathrm{Re}[S]) + j f_{\mathrm{Im}}(\mathrm{Im}[S]) \tag{1}$$

where functions $f_{\mathrm{Re}}(\bullet)$ and $f_{\mathrm{Im}}(\bullet)$ can be simple real-valued sigmoids or more sophisticated adaptive functions.

### C. Spline-Based Activation Function

Since the activation function is the same both for the real and the imaginary part, in this and the following paragraphs, we will simply write $f(\bullet)$ without the index Re or Im.

Let the LUT be defined as a $N+1$ points array $\{Q_0, \cdots, Q_N\}$, where each point is represented by coordinates $Q_i = [q_{x,i} \;\; q_{y,i}]^T$, and $T$ is the transpose operator. We constrain the LUT points as

$$q_{x,0} < q_{x,1} < \cdots < q_{x,N}.$$

This is because we do not want the curve to generate loops (reverse ordering of abscissas) or multiple output values for a single abscissa value (overlapping abscissas). This constraint is necessary as it prevents us from representing noninjective functions.

The choice of an interpolation scheme is not trivial in spite of the fact that it has been demonstrated that even piecewise linear activation functions retain the universal approximation property [23]; a wrong choice can lead to problems in the development of the learning algorithm. We have found that a zero-order interpolation [27] (just like a sample-and-hold system) suffers from numerical instability because the derivative, which is computed as a finite difference of the activation function, is a noncontinuous approximation.

The interpolation scheme to be used should guarantee a continuous first derivative as well as the ability to locally adapt the curve. Such properties are met by the so-called piecewise polynomial spline interpolation schemes [33].

A planar spline curve is a 2-D vector whose components are piecewise polynomial univariate functions of the same degree. Its mathematical formulation ensures both its continuity and the existence of its derivatives along the curve and in correspondence with the joining points between the various curve spans. Given an LUT with $N+1$ sample points, a general spline expression for the interpolated curve would be

$$f(u) = [f_x(u) \quad f_y(u)]^T = \mathbf{C}_{i=0}^{N-3} f_i(u) \tag{2}$$

where $\mathbf{C}$ is the concatenation operator along the $u$ axis and $f_i(u)$ the function on the $i$th curve span. The limits in the $\mathbf{C}$ operator in (2) are valid for cubic polynomials only. Cubic polynomials have been chosen because of the tradeoff between the requested properties and the computational complexity.

The parameter $u$ has the property of being *local*, and its domain is $0 \le u \le 1$ for every curve span. Therefore, there must be a unique mapping that allows us to calculate, from the activation function abscissa *global* parameter (i.e., the output of the linear combiner), the local parameter $u$, as well as the proper curve span $i$. In this way, we can represent any point lying on the spline curve $f(u)$ as a point belonging to a single $f_i(u)$ curve span, which is described as

$$f_i(u) = [f_{xi}(u) \quad f_{yi}(u)]^T = \sum_{j=0}^{3} Q_{i+j} c_j(u) \tag{3}$$

where $c_j(u)$ are the spline polynomials. As in (2), each spatial coordinate is described by a univariate function, namely, a cubic polynomial of the variable $u$.

Let $s$ be the real or the imaginary part of the neuron's linear combiner output that is usually called "activation"; there is a direct dependency between $s$ and one of the two components of $f_i(u)$, that is

$$s = f_{xi}(u). \tag{4}$$

Equation (4) has a key role in finding a suitable activation function architecture as it is the main bottleneck of the overall structure; in fact, it is the only link between the linear combiner output $s$ and the nonlinear neuron output $f_{yi}(u)$, which is a function of the $i \in [0, N]$ value (which represents the LUT index) and of the $u \in [0, 1]$ value (which represents the offset of the $i$th curve span). Let $y$ be the real or the imaginary part of the complex neuron output; a two-step procedure should be introduced to calculate $y$ of a single neuron, given the output $s$ of the linear combiner. Calculate $u$ and $i$ from $s$ by inverting (4), and substitute these values of $u$ and $i$ in $y = f_{yi}(u)$.

Equation (3) resolves the link between the curve and the sample points $Q_i$, which are called *control points* in the literature as they control the shape of the curve. The cubic polynomial functions $c_i(u)$, which here are called *spline basis functions* or *blending functions*, characterize the way the curve moves along the path made up by the control points. The curve can interpolate or just approximate its control points, depending on which set of blending functions is used.

In the literature, many different blending functions have been proposed (see, for example, [34]), but there are actually few that interpolate their control points while maintaining a low computational overhead. One of these is the so-called *Catmull–Rom cubic spline basis* (CR) [34], which is described

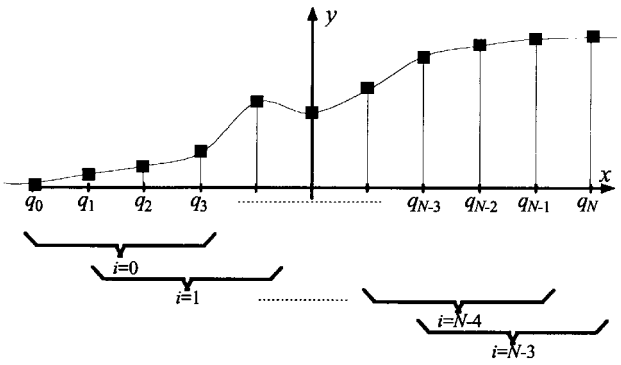



Fig. 1. Spline activation function with its uniformly spaced control points.

by the following polynomials:

$$\begin{aligned} c_0(u) &= \frac{1}{2}(-u^3+2u^2-u) \\ c_1(u) &= \frac{1}{2}(3u^3-5u^2+2) \\ c_2(u) &= \frac{1}{2}(-3u^3+4u^2+u) \\ c_3(u) &= \frac{1}{2}(u^3-u^2). \end{aligned} \tag{5}$$

In matrix form [with reference to (3)], we have

$$f_i(u) = [u^3 \quad u^2 \quad u \quad 1]\frac{1}{2}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix}\begin{bmatrix} Q_i \\ Q_{i+1} \\ Q_{i+2} \\ Q_{i+3} \end{bmatrix} \tag{6}$$

which makes the array structure of a curve span clearer: a *parameter vector*, a basis *matrix*, and a *control points vector*, combined using matrix multiplication. Fig. 1 shows an example of the proposed spline activation function.

## III. The Complex-Valued Backpropagation for ASNN

As in [18], in order to simplify the learning algorithm, we force the control point abscissas to be equidistant and fixed. We also impose another constraint on the control points, centering the sampling interval on the $x$-axis origin. It is then possible to represent the abscissa of every point of the activation function with two parameters: the span index $a$ and the local parameter $u$. All we need to know is the number of control points the curve has and the $x$-axis step $\Delta x$.

Using a notation similar to that introduced by Widrow and Lehr in [22], we define the following symbols:

| | |
|---|---|
| $X_k^{(l)}$ | complex output of the $k$th neuron in the $l$th layer ($l = 0$ stands for the input layer); |
| $D_k$ | complex desired output of the $k$th neuron in the output layer; |
| $W_{kj}^{(l)}$ | complex weight between the $k$th neuron in the $l$th layer and the $j$th neuron in the previous layer; |
| $S_k^{(l)}$ | complex net output (i.e., the linear combiner output) of the $k$th neuron in the $l$th layer; |
| $N+1$ | number of control points for each neuron in the network; |
| $a_{k\mathrm{Re}}^{(l)}$, $a_{k\mathrm{Im}}^{(l)}$ | curve span index of the real or imaginary part of the activation function for the $k$th neuron in the $l$th layer ($0 \le a_k^{(l)} \le N-2$); |
| $u_{k\mathrm{Re}}^{(l)}$, $u_{k\mathrm{Im}}^{(l)}$ | local parameter of the spline span of the $k$th neuron in the $l$th layer ($0 \le u_k^{(l)} \le 1$); |
| $q_{k,m\mathrm{Re}}^{(l)}$, $q_{k,m\mathrm{Im}}^{(l)}$ | ordinates of the $m$th control point of the $k$th neuron in the $l$th layer ($0 \le m \le N$) (we here assume $q_y$ index coordinate as default because the $x$-axes is uniformly sampled and then the control points $q_x$ are analytically determined); |
| $F_{k,m}^{(l)}(\cdot)$ | $m$th spline patch of the activation function for the $k$th neuron in the $l$th layer; |
| $c_{k,m\mathrm{Re}}^{(l)}(\cdot)$, $c_{k,m\mathrm{Im}}^{(l)}(\cdot)$ | $m$th CR polynomial (blending function) for the $k$th neuron in the $l$th layer ($0 \le m \le 3$). |

Considering $M$ total layers and indicating each of them with the index $l$, $l = 1, \ldots, M$, we can find the span and the local variable by

$$\begin{aligned} z_{k\mathrm{Re}}^{(l)} &= \frac{\mathrm{Re}\left[S_k^{(l)}\right]}{\Delta x} + \frac{N-2}{2}\mathbb{P} z_{k\mathrm{Im}}^{(l)} = \frac{\mathrm{Im}\left[S_k^{(l)}\right]}{\Delta x} + \frac{N-2}{2} \\ a_{k\mathrm{Re}}^{(l)} &= \left\lfloor z_{k\mathrm{Re}}^{(l)} \right\rfloor \mathbb{P} a_{k\mathrm{Im}}^{(l)} = \left\lfloor z_{k\mathrm{Im}}^{(l)} \right\rfloor \\ u_{k\mathrm{Re}}^{(l)} &= z_{k\mathrm{Re}}^{(l)} - a_{k\mathrm{Re}}^{(l)} \mathbb{P} u_{k\mathrm{Im}}^{(l)} = z_{k\mathrm{Im}}^{(l)} - a_{k\mathrm{Im}}^{(l)} \end{aligned} \tag{7}$$

where $\lfloor\ \rfloor$ is the floor operator, and $z_k^{(l)}$ is an internal dummy variable. These expressions lead to the neuron structure reported in Fig. 2.

*Backpropagation:* Following a development similar to [29], [31], and [32] for the synaptic weights, the learning algorithm is now extended to the spline control points.

$$\begin{aligned} E_k^{(l)} &= \begin{cases} D_k - X_k^{(l)} & l = M \\ \sum_{m=1}^{N_{l+1}} \Delta_m^{(l+1)} W_{mk}^{*(l+1)} & l = M-1, \ldots, 1 \end{cases} \\ \Delta_k^{(l)} &= \mathrm{Re}\left[E_k^{(l)}\right] \left( \frac{d\mathrm{Re}\left[F_{k,a_k^{(l)}}^{(l)}(u)\right]}{du}\bigg|_{u=u_{k\mathrm{Re}}^{(l)}} \right) \frac{1}{\Delta x} \\ &\quad + j\mathrm{Im}\left[E_k^{(l)}\right] \left( \frac{d\mathrm{Im}\left[F_{k,a_k^{(l)}}^{(l)}(u)\right]}{du}\bigg|_{u=u_{k\mathrm{Im}}^{(l)}} \right) \frac{1}{\Delta x} \\ W_{kj}^{(l)}[p+1] &= W_{kj}^{(l)}[p] + 2\mu_w \Delta_k^{(l)} X_j^{(l-1)} \end{aligned} \tag{8}$$

with $0 \le k \le N_l$, $0 \le j \le N_{l-1}$, and the symbol $^*$ indicating complex conjugation. The adaptation of the control points is

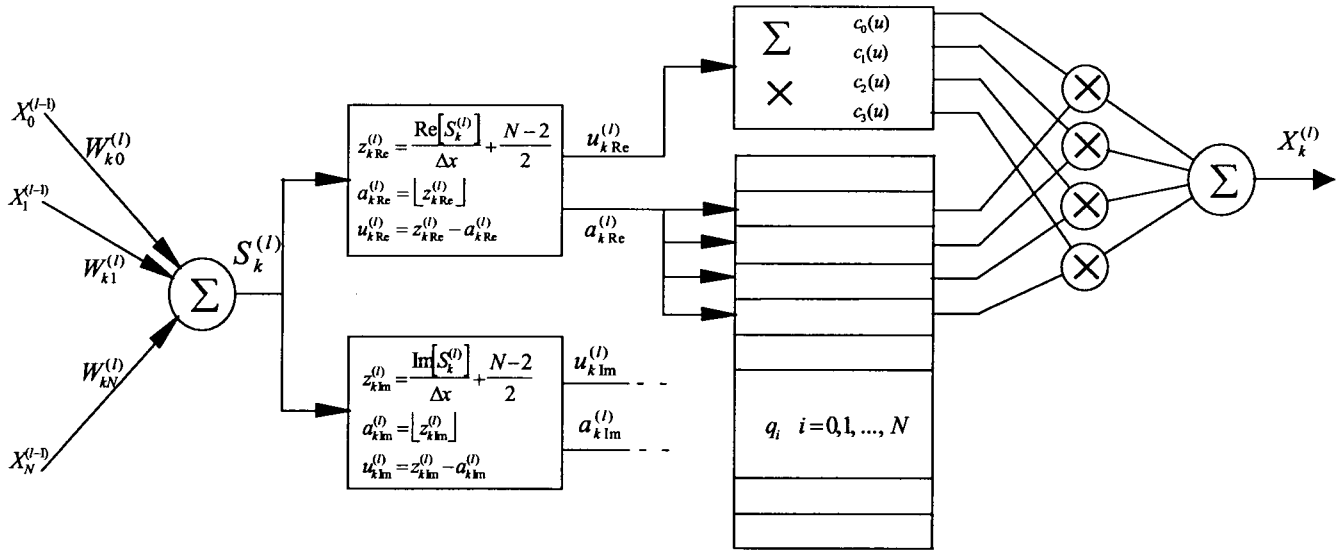



Fig. 2. Complex-valued generalized sigmoid (GS) neuron structure (the structure of the imaginary part is omitted because it is identical to that of the real part).

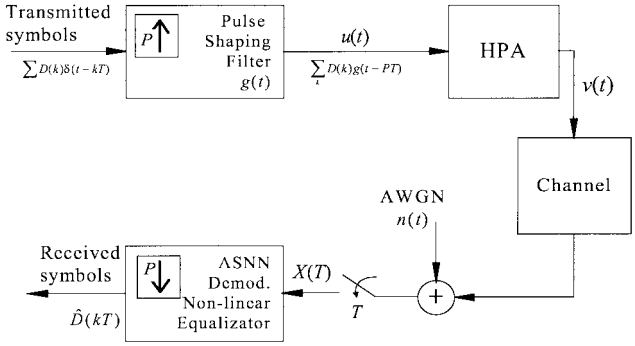
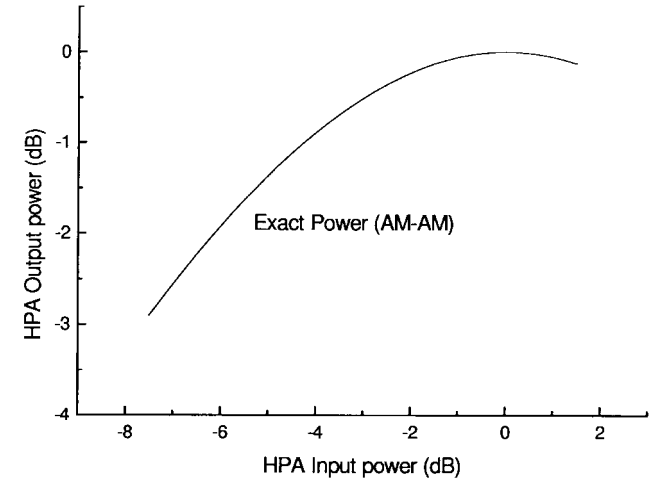



Fig. 3. Block diagram of the digital radio link using a complex-valued ASNN as a nonlinear equalizer.

ruled by

$$
\begin{aligned}
q_{k,(a_k^{(l)}+m)\mathrm{Re}}^{(l)}[p+1] &= q_{k,(a_k^{(l)}+m)\mathrm{Re}}^{(l)}[p] \\
&\quad + 2\mu_q \mathrm{Re}\left[E_k^{(l)}\right] \left( \frac{\partial \mathrm{Re}\left[F_{k,a_k^{(l)}}^{(l)}\right]}{\partial q_{k,(a_k^{(l)}+m)\mathrm{Re}}^{(l)}} \right) \\
&= q_{k,(a_k^{(l)}+m)\mathrm{Re}}^{(l)}[p] + 2\mu_q e_k^{(l)} c_{k,m\mathrm{Re}}^{(l)}\left(u_k^{(l)}\right) \\
q_{k,(a_k^{(l)}+m)\mathrm{Im}}^{(l)}[p+1] &= q_{k,(a_k^{(l)}+m)\mathrm{Im}}^{(l)}[p] \\
&\quad + 2\mu_q \mathrm{Im}\left[E_k^{(l)}\right] \left( \frac{\partial \mathrm{Im}\left[F_{k,a_k^{(l)}}^{(l)}\right]}{\partial q_{k,(a_k^{(l)}+m)\mathrm{Im}}^{(l)}} \right) \\
&= q_{k,(a_k^{(l)}+m)\mathrm{Im}}^{(l)}[p] + 2\mu_q e_k^{(l)} c_{k,m\mathrm{Im}}^{(l)}\left(u_k^{(l)}\right)
\end{aligned}
\tag{9}
$$

where the patch index $m = 0, \ldots, 3$. The adaptation rates are $\mu_w$ for the connection weights and biases and $\mu_q$ for the control points. In order to have a squashing behavior, the left and right extreme control points $q$ with the indexes $m = 0, 1, (N-1)$ and $N$ are kept fixed.



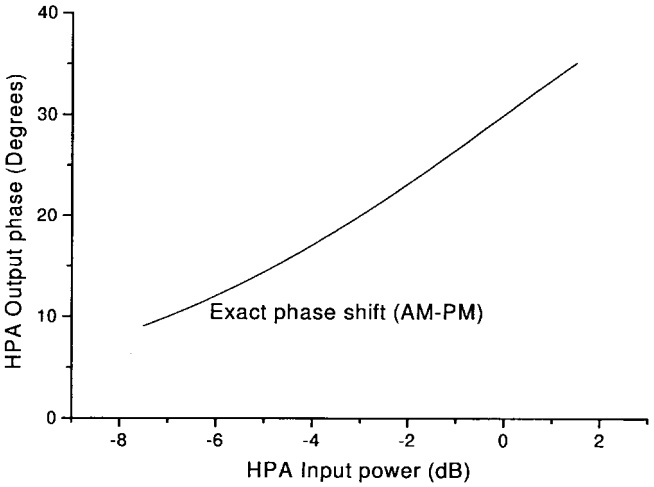
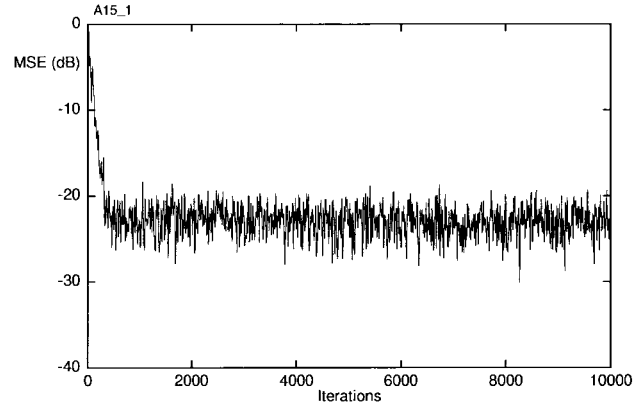
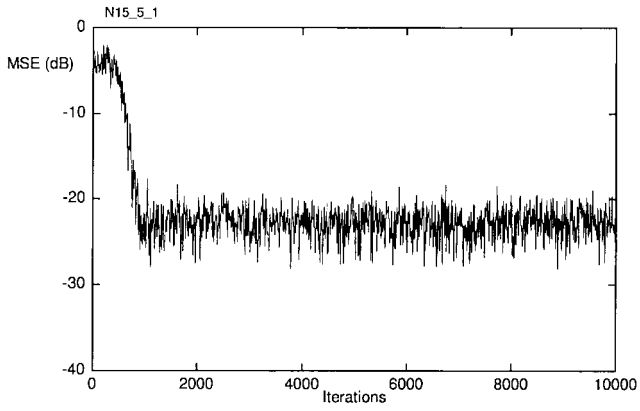




Fig. 4. Typical traveling-wave tube (TWT) HPA memoryless input–output characteristic.

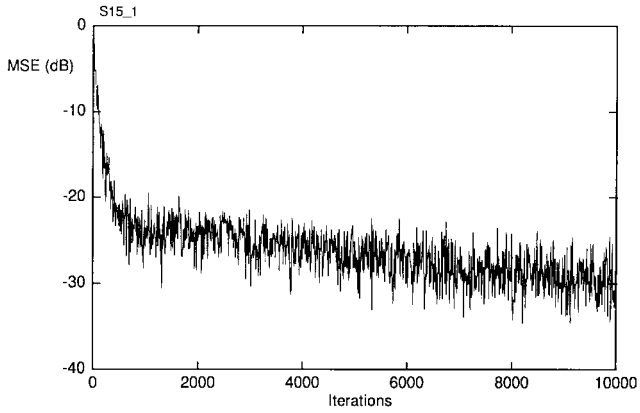
The spline curve control points are initialized by sampling a bipolar sigmoid (or other well-known activation functions); outside the sampling interval, the asymptotic shape of the sigmoid is mantained.




(a)



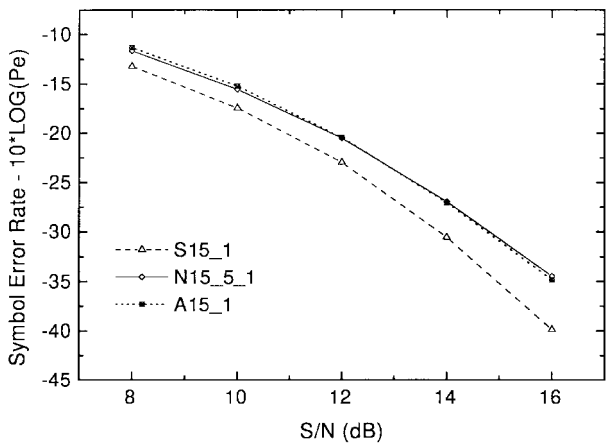

(b)



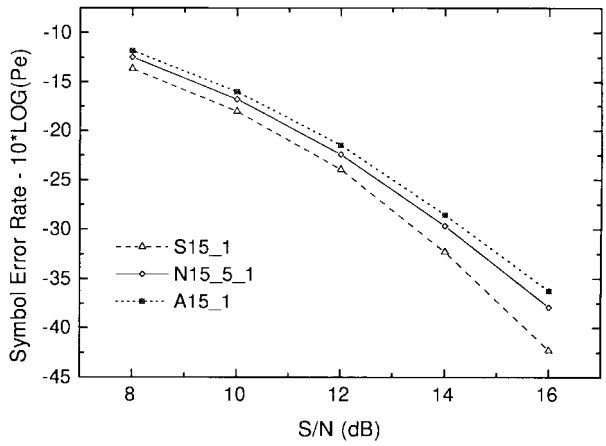

(c)

Fig. 5. MSE versus training iterations. (a) Linear equalizer A15_1. (b) MLP neural network N15_5_1 with five hidden neurons. (c) Proposed ASNN S15_1.

A brief analysis of computational complexity for a single GS neuron can be carried out as follows. The first block of Fig. 2 that computes the index $a_k^{(l)}$ and the offset $u_k^{(l)}$ of the $k$th spline span performs two additions, one rounding operation, and one multiplication [see (7)]. For the second block, using a smart implementation based on the structure of (5), we just need nine additions, two products, and two shifts for the forward phase. Moreover, for the backward phase, the four control points of the $k$th span have to be adapted. Obviously, these calculations must be applied to both the real and imaginary part of the neuron and added to the complexity of the linear combiner.




(a)




(b)

Fig. 6. Symbols error probability $(P_e)$ plotted versus signal-to-noise ratio $(S/N)$ at the output of the receiver using various equalization schemes. (a) $10^4$ learning iterations. (b) $10^5$ learning iterations.

## IV. Experimental Results

The block diagram of the radio link used in our experiments is depicted in Fig. 3. The complex input data sequence $D(k)$, represents the points of the QAM constellation.

The function $g(t)$ represents the modulator filter impulse response. In the simulation, we use the square-root of a raised-cosine having roll-off factor $\alpha$ equal to 0.5; the oversampling factor $P$ is chosen equal to 3. The $g(t)$ filter length is equal to $5P$ taps.

The model for the HPA, which is described in [2], is characterized by the expression

$$v(t) = \frac{2u(t)}{1+|u(t)|^2} \exp\left[j\Phi_0 \frac{2|u(t)|^2}{1+|u(t)|^2}\right]. \qquad (10)$$

The input–output HPA (memoryless) response is described by the AM–AM response, which is represented by the module, and the AM–PM response, which is represented by the phase. This description assumes, for convenience, that the maximum possible HPA input power $W_{\text{in}} = |u(t)|^2$ is equal to 1 W, and the maximum shift is $\Phi_0 = \frac{\pi}{6}$, which are typical values [1]. In

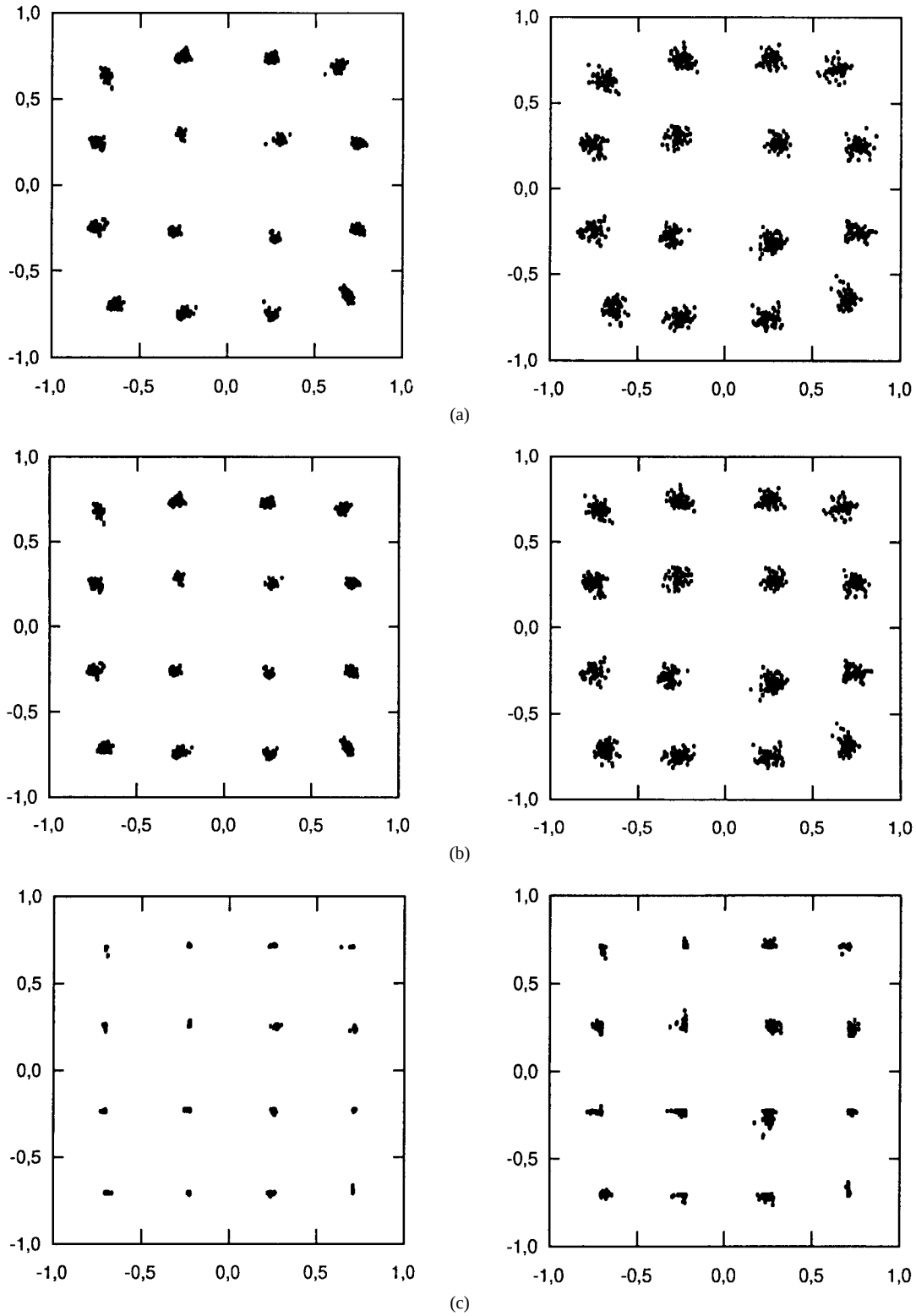



Fig. 7. Received 16-QAM signals (2048 symbols after $10^5$ learning iterations) for various equalizers. The input HPA maximum power is $P_{\text{inMAX}} = -1$ dB. (left) No additive noise. (Right) Additive noise (AWGN) $S/N = 20$ dB. (a) Adaptive linear combiner (A15_1). (b) Sigmoidal MLP (N15_15_1). (c) Adaptive spline neural network (S15_1).

Fig. 4, the AM–AM and AM–PM characteristics of a typical HPA are reported.

In the simulations, the transmission of QAM signals is considered and, for the sake of brevity, only the case of 16 QAM has been reported. The maximum input power to the HPA $W_{\text{inMAX}}$ is very close to the saturation point and fixed at $-1$ dB (see Fig. 4).

Several tests with different architectures have been performed, although the most significant experiments are based on a comparison among the following three complex-valued

equalizers.

1) Complex-valued linear combiner (A15_1) (15 is related to the tapped delay line length).
2) Complex-valued standard multilayer neural network with one hidden layer composed of five sigmoidal neurons and a linear output (N15_5_1).
3) Complex-valued ASNN composed of only one complex GS neuron (S15_1).

In order to evaluate the performance with respect to the learning iterations (a forward and backward step for one symbol), we have trained the equalizers in two different conditions, that is, using $10^4$ and $10^5$ different training symbols (or training iterations), which correspond to $10^4 \times P$ and $10^5 \times P$ modulated input samples, where $P$ is the modulation oversampling rate. Each modulated sample is corrupted by adding various $S/N$ values of white zero-mean Gaussian noise [$v(t) + n(t)$ in the scheme Fig. 3]. Note that the network also performs the downsampling conversion.

For all the networks, a search of the optimal adaptation rates have been carried out. $\mu_q$ and $\mu_w$ for the S15_1 are chosen equal to 0.001 and 0.1, respectively; a value of $\mu_w$ equal to 0.01 is used for the MLP N15_5_1, whereas for the adaptive linear combiner, A15_1 $\mu_w$ is equal to 0.001.

Several tests using different network initialization weights have been carried out. Fig. 5 shows the plot of the MSE expressed in decibels versus the training iterations for the three architectures using $10^4$ training symbols.

In order to evaluate the radio link performances in a more realistic way, the symbol error probability $(P_e)$ is estimated. Fig. 6 reports the $P_e$ values versus the $S/N$, both expressed in decibels.

Fig. 6(a) shows the symbol error probability after $10^4$ iterations, whereas in Fig. 6(b), the same data are plotted after $10^5$ iterations. From these figures, we can observe that in both cases, the proposed approach leads to significant improvements not only with respect to the classical linear adaptive filter but also with respect to the conventional sigmoidal MLP-based equalization technique. On the other hand, the common sigmoidal MLP needs more training to reach its best performance. In fact, if only $10^4$ iterations are used, the MLP has a behavior similar to the linear equalizer. Although there is not a strict correlation between MSE and symbol error rate, the MSE plots in Fig. 5 confirm our previous results.

After the learning, a forward phase test is performed; the received 16 QAM constellation is depicted in Fig. 7(a) for the adaptive linear combiner, Fig. 7(b) for the sigmoid MLP and Fig. 7(c) for the proposed ASNN. Two different cases are reported: case 1 without additive noise at the receiver input, and case 2 with signal-to-noise ratio of 20 dB. It is worth noting that the constellation of the received symbols when using ASNN is less distorted than in the other two cases (in Fig. 7(c1) the points are exactly fixed on the original square grid).

The module of the complex activation function is plotted in Fig. 8. The symmetry of the problem is reflected in the plot. The effect that such a nonlinearity has on the constellation points is evident.

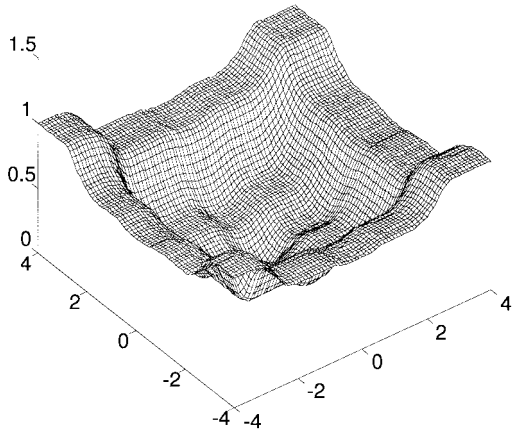



Fig. 8. Module of the complex activation function after the learning for the networks S15_1 used in the experiments.

The last experiment concerns the tracking capabilities of the ASNN. The test has been carried on changing randomly every 1000 symbols the added noise and the HPA input power (back off) [1]. The $S/N$ range is [8 dB, 16 dB], whereas the back-off range is [$-2$ dB, $-0.5$ dB]. The $S/N$ and both the output power and the output phase dramatically change; an action on this parameter produces a greatly different channel model. Fig. 9 shows the robustness in tracking mode of the proposed architecture. The symbol error rates for this experiment are directly reported in the figure caption.

## V. Conclusion

The main problem using standard neural networks as nonlinear adaptive filters is the high degree of overall complexity (which accounts for a very long adaptation time and a high number of interconnections) that can hinder any practical application. In this paper, a new architecture is proposed, based on an adaptive spline activation function approach, characterized by a much lesser complexity. This fact overcomes the previous drawbacks and makes neural structures of effective use in real-world problems.

As for the QAM equalization problem, this advantage is even more evident since the network reduces to a single complex neuron. The reduced complexity is responsible for the shorter adaptation phase in terms of training iterations, as experimentally observed. Comparing our technique with classical linear approaches, we have noted little implementation overhead compared with the adaptive linear filter but with significant improvement in the performance, both in terms of MSE and symbol error rate. As far as nonlinear methods are concerned, the proposed architecture exhibits performance comparable with finite-order inverse Volterra filters [4] and to global compensation [8], [9] but with less complexity, which makes it suitable for an effective implementation.

## Acknowledgment

The authors wish to acknowledge the anonymous reviewers for their useful suggestions and comments that yielded to an improved version of this manuscript.

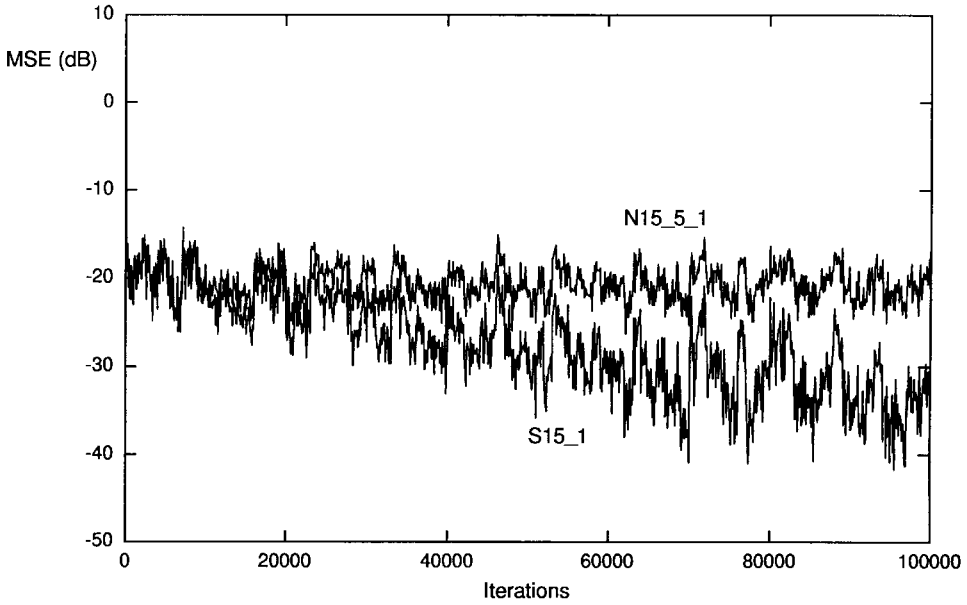



Fig. 9. MSE versus training iterations using a time-varying channel. The $S/N$ and the HPA backoff are randomly varied every 1000 symbols from 8 to 16 dB and from $-2$ to $-0.5$ dB, respectively. The symbol error rates are $P_e = -18.2$ dB for the N15_5_1 and $P_e = -19.1$ dB for the S15_1 net. For plot clarity, the A15_1 the learning curve is not reported, whereas its symbol error rate is $P_e = -17.7$ dB.

## REFERENCES


[1] S. Pupolin and L. J. Greenstein, "Performance analysis of digital radio links with nonlinear transmit amplifiers," *IEEE J. Select. Areas Commun.*, vol. SAC-5, pp. 534–456, Apr. 1987.

[2] A. A. M. Saleh, "Frequency-independent and frequency-dependent nonlinear models of TWT amplifiers," *IEEE Trans. Commun.*, vol. COMM-29, pp. 1715–1720, Nov. 1981.

[3] M. F. Mesiya, P. J. McLane, and L. L. Campbell, "Maximum likelihood receiver for carrier-modulated data transmission systems," *IEEE Trans. Commun.*, vol. COMM-22, pp. 624–636, May 1974.

[4] S. Benedetto and E. Biglieri, "Nonlinear equalization of digital satellite channels," *IEEE J. Select. Areas Commun.*, vol. SAC-1, pp. 57–62, Jan. 1983.

[5] D. D. Falconer, "Adaptive equalization of channel nonlinearities in QAM data transmission," *Bell Syst. Tech. J.*, vol. 57, pp. 2589–2611, Sept. 1978.

[6] E. Biglieri, S. Barberis, and M. Catena, "Analysis and compensation of nonlinearities in digital transmission systems," *IEEE J. Select. Areas Commun.*, vol. 6, pp. 42–51, Jan. 1988.

[7] G. Karam and H. Sari, "Analysis of predistortion, equalization, and ISI cancellation techniques in digital radio systems with non linear transmit amplifiers," *IEEE Trans. Commun.*, vol. 37, pp. 1245–1253, Dec. 1989.

[8] G. Karam and H. Sari, "Data predistortion techniques using intersymbol interpolation," *IEEE Trans. Commun.*, vol. 38, pp. 1716–1723, Oct. 1990.

[9] G. Karam and H. Sari, "Generalized data predistortion using intersymbol interpolation," *Philips J. Res.*, vol. 46, pp. 1–21, 1991.

[10] N. Benvenuto, F. Piazza, and A. Uncini, "A neural network approach to data predistorter with memory in digital radio system," in *Proc. ICC93 Int. Commun. Conf.*, Geneva, Switzerland, May 1993.

[11] G. Kechriotis, E. Zarvas, and E. S. Manolakos, "Using recurrent networks for adaptive communication channel equalization," *IEEE Trans. Neural Networks*, vol. 5, pp. 267–278, Mar. 1994.

[12] S. Chen, G. J. Gibson, C. F. N. Cowan, and P. M. Grant, "Adaptive equalization of finite nonlinear channels using multilayer perceptron," *Signal Process.*, vol. 20, pp. 107–109, 1990.

[13] G. J. Gibson, S. Siu, and C. F. N. Cowan, "Multilayer perceptrons structures applied to adaptive equalisers for data communications," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. ICASSP*, Glasgow, U.K., May 1989.

[14] N. Benvenuto, M. Marchesi, F. Piazza, and A. Uncini, "Non linear satellite radio links equalized using blind neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. ICASSP*, Toronto, Ont., Canada, May 14–17, 1991.

[15] Q. Zhang, "Adaptive equalization using the back propagation algorithm," *IEEE Trans. Circuits Syst.*, vol. 37, pp. 848–849, June 1990.

[16] S. Chen, G. J. Gibson, and C. F. N. Cowan, "Adaptive channel equalization using a polynomial-perceptron structure," *Proc. Inst. Elect. Eng.*, vol. 137, pt. I, no. 5, Oct. 1990.

[17] W. S. Gan, J. J. Soraghan, and T. S. Durrani, "A new functional-link based equaliser," *Electron. Lett.*, vol. 28, no. 17, pp. 1643–1645, Aug. 1992.

[18] S. Guarnieri, F. Piazza, and A. Uncini, "Multilayer neural networks with adaptive spline-based activation functions," in *Proc. Int. Neural Network Soc. Annu. Meet. WCNN*, Washington, DC, 1995, pp. I695–I699.

[19] S. Amari and N. Murata, "Statistical theory of learning curves under entropic loss criterion," *Neural Comput.*, vol. 5, pp. 140–153, 1993.

[20] S. Amari, "A universal theorem on learning curves," *Neural Networks*, vol. 6, no. 2, pp. 161–166, 1993.

[21] D. E. Rumelhart, J. L. McClelland, and the PDP Res. Group, *Parallel Distributed Processing (PDP): Exploration in the Microstructure of Cognition*. Cambridge, MA: MIT Press, 1986, vol. 1.

[22] M. Widrow and M. Lehr, "30 years of adaptive neural networks: Perceptron, adaline and backpropagation," *Proc. IEEE*, vol. 78, Sept. 1990.

[23] K. Hornik, M. Stinchombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359–366, 1989.

[24] ———, "Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks," *Neural Networks*, vol. 3, pp. 551–560, 1990.

[25] C. T. Chen and W. D. Chang, "A feedforward neural network with function shape autotuning," *Neural Networks*, vol. 9, no. 4, pp. 627–641, June 1996.

[26] F. Piazza, A. Uncini, and M. Zenobi, "Artificial neural networks with adaptive polynomial activation function," in *Proc. IJCNN*, Beijing, China, Nov. 1992, pp. II-343–349.

[27] ———, "Neural networks with digital LUT activation function," in *Proc. IJCNN*, Nagoya, Japan, 1993, pp. II-1401–1404.

[28] N. Benvenuto, M. Marchesi, F. Piazza, and A. Uncini "A comparison between real and complex valued neural networks in communication applications," in *Proc. Int. Conf. Neural Networks ICANN*, Espoo, Finland, June 1991.

[29] H. Leung and S. Haykin, "The Complex backpropagation algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 39, pp. 2101–2104, Sept. 1991.

[30] N. Georgiu and C. Koutsougeras, "Complex domain backpropagation," *IEEE Trans. Circuits Syst.*, vol. 39, pp. 330–334, May 1992.

[31] S. Haykin, *Adaptive Filter Theory*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[32] N. Benvenuto and F. Piazza "On the complex backpropagation algorithm," *IEEE Trans Signal Processing*, vol. 40, pp. 967–969, Apr. 1992.
[33] I. J. Schoenberg, "Contributions to the problem of approximation of equidistant data by analytic functions," *Quart. Appl. Math.*, vol. 4, pp. 45–99, 1946.
[34] E. Catmull and R. Rom, "A class of local interpolating splines," in *Computer Aided Geometric Design*, R. E. Barnhill and R. F. Riesenfeld, Eds. New York: Academic, 1974, pp. 317–326.
[35] L. Vecci, P. Campolucci, F. Piazza, and A. Uncini, "Approximation capabilities of adaptive spline neural networks," in *Proc. Int. Conf. Neural Network ICNN*, Houston TX, June 1997, pp. 260–265.
[36] N. Benvenuto, F. Piazza, A. Uncini, and M. Visintin, "Generalized backpropagation algorithm for training a data predistorter with memory in radio systems," *Electron. Lett.*, vol. 32, no. 20, pp. 1925–1926, Sept. 1996.
[37] L. Vecci, F. Piazza, and A. Uncini, "Learning and approximation capabilities of adaptive spline activation neural networks," *Neural Networks*, vol. 11, no. 2, pp. 259–270, Mar. 1998.

**Aurelio Uncini** (M'95) was born in Cupra Montana, Italy, in 1958. He received the laurea degree in electronic engineering from the University of Ancona, Ancona, Italy, in 1983 and the Ph.D. degree in electrical engineering from University of Bologna, Bologna, Italy in 1994.

From 1984 to 1986, he was with the "Ugo Bordon" Foundation, Rome, Italy, engaged in research on digital processing of speech signals. From 1986 to 1987, he was at Italian Ministry of Communications. Currently, he is an Associate Professor with the University of Ancona in the Electronics and Automatics Department, the Digital Signal Processing, and the Neural Networks Research Group. His research interests are in digital signal processing, neural networks, and circuit theory. His present research interests include adaptive filters, audio and speech processing, optimization algorithms for circuits design, and neural networks for signal processing. He has published more than 70 journal and conference papers in these areas.

**Lorenzo Vecci** (S'94–M'97) was born in Jesi, Italy, in 1971. He received the laurea degree in electronic engineering with honors from the University of Ancona, Ancona, Italy.

From September 1996 through July 1997, he was with the Computer Science Institute, University of Ancona, where he worked on algorithms for image processing; he also worked with the Digital Signal Processing and Neural Networks Research Group in the Electronics and Automatics Department of the same university. Since August 1997, he has been with CSELT, Turin, Italy. His research interests include digital signal processing (DSP), neural networks, pattern recognition, and microelectronics. He is involved with the theoretical properties of neural network architectures; he is also involved in the IC design of digital filters and neural network applications to DSP.

**Paolo Campolucci** (S'94) was born in Forlì, Italy, in 1969. He received the laurea degree in electronic engineering (summa cum laude) in 1994 from the University of Ancona, Ancona, Italy, and the Ph.D. degree in circuit theory in 1998 from the University of Bologna, Bologna, Italy.

He is currently with the Digital Signal Processing and Neural Networks Research Group, Electronics and Automatics Department, University of Ancona. From 1995 to 1996, he was a Visiting Scholar with the Electrical and Computer Engineering Department, University of California, San Diego, and in 1998, he was with the Engineering Department, Brown University, Providence, RI. His current research interests include dynamic and recurrent neural networks, learning theory, digital signal processing, and circuit theory.

**Francesco Piazza** (M'87) was born in Jesi, Italy, in February 1957. He received the degree with honors in electronic engineering from the University of Ancona, Ancona, Italy, in 1981.

From 1981 to 1983, he worked on CCD-image processing at the Physics Department, University of Ancona. In 1983, he worked at the Olivetti OSAI Software Development Center. In 1984, he was a consultant in the field of microprocessor-based systems and radar displays. In 1985, he joined the Department of Electronics and Automatics, University of Ancona, first as Researcher in Electrical Engineering and then as Associate Professor. His current research interests are in the areas of circuit theory and digital signal processing and include adaptive digital signal processing algorithms and circuits, neural networks, and speech and audio processing.