To appear in
*Machine Learning: Proceedings of the 13th International Conference, 1996.*

# On-Line Adaptation of a Signal Predistorter through Dual Reinforcement Learning

**Patrick Goetz**
Computational and Applied Math
The University of Texas at Austin
Austin, TX 78712 USA
pgoetz@math.utexas.edu

**Shailesh Kumar**
Department of Computer Sciences
The University of Texas at Austin
Austin, TX 78712 USA
skumar@cs.utexas.edu

**Risto Miikkulainen**
Department of Computer Sciences
The University of Texas at Austin
Austin, TX 78712 USA
risto@cs.utexas.edu

## Abstract

Several researchers have demonstrated how neural networks can be trained to compensate for nonlinear signal distortion in e.g. digital satellite communications systems. These networks, however, require that both the original signal and its distorted version are known. Therefore, they have to be trained off-line, and they cannot adapt to changing channel characteristics. In this paper, a novel *dual reinforcement learning* approach is proposed that can adapt on-line while the system is performing. Assuming that the channel characteristics are the same in both directions, two predistorters at each end of the communication channel co-adapt using the output of the other predistorter to determine their own reinforcement. Using the common Volterra Series model to simulate the channel, the system is shown to successfully learn to compensate for distortions up to 30%, which is significantly higher than what might be expected in an actual channel.

## 1 Introduction

In a satellite communication system, a transmitter sends a signal to a satellite which then passes it on to a receiver.[1] Typically, the transmitter encodes binary data as complex numbers, which allows for more efficient use of bandwidth and power. The resulting complex pulse train is then converted to a continuous signal, amplitude modulated to a higher frequency, and fed to a high-power amplifier for transmission.

Because of the large distance between the transmitter and the satellite (satellites typically orbit roughly 35,800 km above the equator), the signal is seriously attenuated when it arrives at the satellite. Therefore, the satellite is equipped with a high-power amplifier (HPA), typically a traveling-wave tube (TWT) or a GaAs FET (Benvenuto et al. 1993; Eun 1995). To conserve power, the HPA is operated at near saturation, which introduces nonlinear distortions in both the phase and the amplitude of the transmitted complex signal. These distortions can result in a phenomenon known as "intersymbol interference" (Widrow and Winter 1988), where the symbols received by the receiver do not exactly match those sent by the transmitter.

A compensator is a device that acts as a channel inverse, and thereby can be used to reduce signal distortions. Nonlinear compensators can be classified into two categories depending on their location in the communication system: A predistorter compensates the signal before it is sent, an equalizer compensates the signal after it has been received.

Neural networks have been used as both equalizers and predistorters (Benvenuto et al. 1991, 1993; Bernardini and Fina 1993; Eun 1995; Rao et al. 1993); however, predistorters have some advantages over equalizers. Nonlinearity is compensated before noise is added to the system, thus avoiding the noise enhancement effect which can result when a deterministic function is applied to a noisy signal. As a consequence, our efforts are directed towards designing a neural network-based predistorter.

In previous methods for using neural networks as compensators (Benvenuto et al. 1991, 1993; Bernardini and Fina 1993; Eun 1995; Rao et al. 1993), both the original and the distorted signals must be known in order to train the network. In other words, the predistorter and the channel output, or, alternatively, the equalizer and the channel input, must be at the same location, which, of course, is never the case in an actual digital communications satellite system. As a result, the neural network compensators can only be trained

[1] For the purposes of this paper, everything between the transmitter and the receiver can be seen as a black box communications channel.

off-line using simulated conditions. It would be useful to be able to train the compensator on-line, using the information actually available in the communications system. In such a system, the compensator could continually adapt to changes in channel characteristics which might occur e.g. due to changing atmospheric conditions.

In this paper a design for such a neural network predistorter (with an accompanying equalizer to facilitate training) is introduced. The utility of the approach is based on a number of reasonable assumptions about the stochastic regularity of the flow of symbols between transmitter and receiver units. The design is evaluated in a simulated communication system where the nonlinear channel distortion is modeled using Volterra series. The results are encouraging, showing that the system can learn to compensate for distortions up to 30%. These results suggest that such a system could be effectively used to track continuously changing distortions in real satellite communication systems.

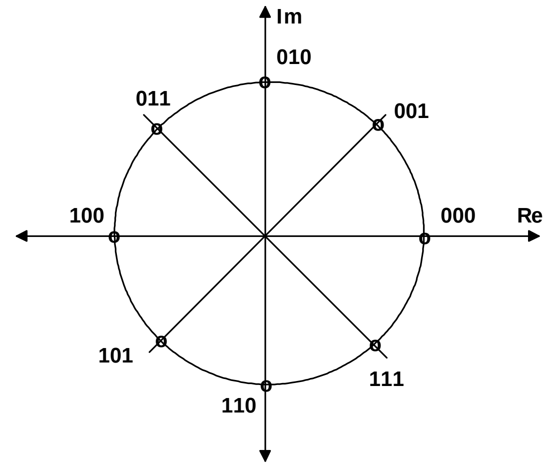


Figure 1: **Symbol Mapping.** The symbols (three-bit binary numbers) are represented as complex numbers located on the unit circle.

## 2 Signal and Channel Characteristics

Since real-valued signals have a symmetric spectral distribution with respect to the DC value, with modulation it is difficult to use the frequency band efficiently (Eun 1995). Consequently, binary data is often encoded as complex symbols prior to transmission. A number of different complex signal formats are used in telecommunications. In the simulations described in this paper, phase shift keying with 8 symbols (8-PSK; Figure 1) is used. In PSK format, the information to be transmitted is encoded into the phase of a complex number while the amplitude is held constant. The number of distinct phases corresponds to the number of symbols to be represented.

Volterra Series models have been used extensively to simulate nonlinear channels (Bellafemina and Benedetto 1985; Benedetto and Biglieri 1983; Biglieri et al. 1988). Given the current input symbol $x(0)$ and a sequence of previous input symbols $x(1), x(2), x(3), \ldots$, the current output $y(n)$ of the communication channel is given by

$$y(n) = \sum_{k=0}^{N_1} h_k^{(1)} x(n-k) + \sum_{k_1=0}^{N_2} \sum_{k_2=0}^{N_2} h_{k_1,k_2}^{(2)} x(n-k_1)x(n-k_2) + \sum_{k_1=0}^{N_3} \sum_{k_2=0}^{N_3} \sum_{k_3=0}^{N_3} h_{k_1,k_2,k_3}^{(3)} x(n-k_1)x(n-k_2)x(n-k_3) + \ldots$$

where $N_1, N_2, N_3, \ldots$ are the memory durations of the first order, second order, third order, and so on (Eun 1995; Schetzen 1980). In the simulations discussed below, only first order memory is used, that is, the output of the channel depends on the current symbol and only one previous symbol.

## 3 The Reinforcement Learning System

### 3.1 The Model/Decision-Maker approach

The reinforcement learning setup for the channel predistortion task consists of training a Model neural network to predict how the environment (i.e. the channel) will react to the decisions made by a Decision-Maker neural network (i.e. the predistorter; Hertz et al. 1991; Munro 1987; Williams 1988). The general architecture is outlined in Figure 2.

The Decision-Maker receives input from the environment and generates an output signal. The environment delivers a reinforcement signal $r$ in response to this output. The Model takes as its input the Decision-Maker's output signal together with the input from the environment, and generates a single real-valued output $R$, which is an estimate of the reinforcement signal $r$.

Both the Model and the Decision-Maker are feedforward backpropagation neural networks using sigmoid neurons. The system is trained by alternating between two phases: Model learning and Decision-Maker training. In the Model learning mode, no changes are made to the Decision-Maker, but the Model uses backpropagation to minimize $(r-R)^2$. In essence, the Model is learning to predict how the environment will

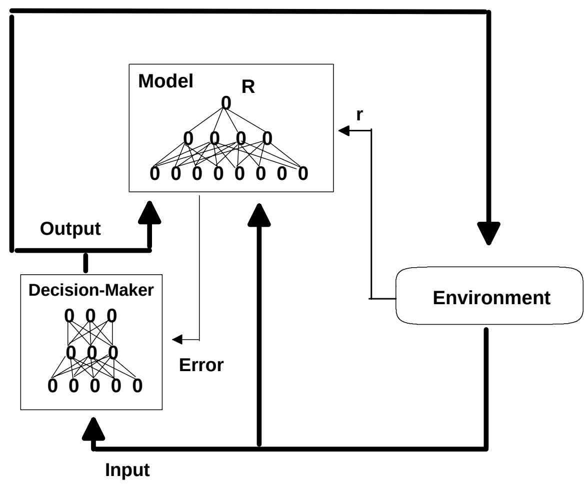


Figure 2: **The Reinforcement Learning System.** The system consists of two neural networks: the Model learns to predict the reinforcement resulting from the Decision-Maker's actions, and the Decision-Maker learns actions that maximize the reinforcement.

respond to the decisions made by the Decision-Maker. If the Model learns its task well, $R = r$ for every input situation.

During Decision-Maker training, the Model is assumed to be a good estimator for the reinforcement signal $r$. The goal is to maximize $R$ by using backpropagation through both networks. The target of the Model network is set to maximum reinforcement 1.0. The error signal at the output is $e = 1.0 - R$. The error signal is propagated through the Model to the Decision-Maker, but the weights of the Model network are not changed. The error signals at the output of the Decision-Maker then indicate how the Decision-Maker should be changed to obtain higher reinforcement.

### 3.2 Dual Reinforcement Learning

With the above reinforcement learning system, it would be possible to train the Decision-Maker to be a good predistorter for the communications channel. However, the central problem remains: how can one accurately train the compensator given that the input to the predistorter (i.e. the original signal) and the reinforcement signal (that describes the quality of the transmitted signal) are located at physically distinct locations?

The problem is overcome by the *dual reinforcement environment* approach. The channel is assumed to be bidirectional, with similar characteristics in both directions. Both end points (nodes) of the channel have transmitters and receivers, and both of them have a reinforcement learning system consisting of a Model and a Decision-Maker network. The quality of the signal that the node receives is used as the reinforcement signal for the Model network, which is then used to train the local Decision-Maker/predistorter. In other words, when the node acts as a transmitter, its predistorter is adapted to the prevalent channel characteristics as indicated by the distortion of the incoming signal. This way, the Model/Decision-Maker pairs are simultaneously learning at both nodes by making use of signals generated by the other node. Assuming that they start with same initial configuration, and the channel characteristics are the same in both directions, they both learn the same predistorter function.

### 3.3 Application to Satellite Communications

Below, the term node is used to refer to the entire digital communications system, consisting of all the modulators, amplifiers, and other components required for signal transmission and reception, as well as the compensator unit. The compensator, referred to as the "unit", acts as the interface between the physical transmitter/receiver and the channel. Each unit is composed of three components: a Predistorter, a (reinforcement learning) Model, and a Slicer (Figure 3). The Predistorter and the Model are three-layer feedforward neural networks. The job of the Slicer is to map the complex number $c$ that the node receives from the channel to the closest valid symbol (i.e. to that complex number representing a valid symbol whose Euclidean distance is closest to $c$).

Depending on whether the node is currently receiving signals, sending symbols, or idle, the compensation unit is in one of two modes: Model learning or Decision-Maker training.

When the node is receiving data, the Model is learning about the channel characteristics. As its input, it needs the original, exact symbol representation $s$ and its predistorted version $p$, and as its output $R$, it computes how close to $s$ the signal would be after it goes through the channel. Assuming that the Predistorter is performing well enough so that the Slicer can recover the original symbol from $c$, $s$ is obtained from the output $s_r$ of the Slicer and $p$ from the output $p_r$ of the Predistorter, with the Slicer output as the input to the Predistorter (dotted line paths in Figure 3). The target $r$ is obtained based on the distance between the input and the output of the Slicer (that is, as $r = -1.0 + 2e^{|c-s_r|}$, which results in a reinforcement signal within $(-1, 1)$). The error $e = r - R$, is backpropagated, and the weights of the Model are updated.

When the node is transmitting data or is idle, the Predistorter (i.e. the Decision-Maker) is trained to form predistorted signals. As its input, it receives the sym-

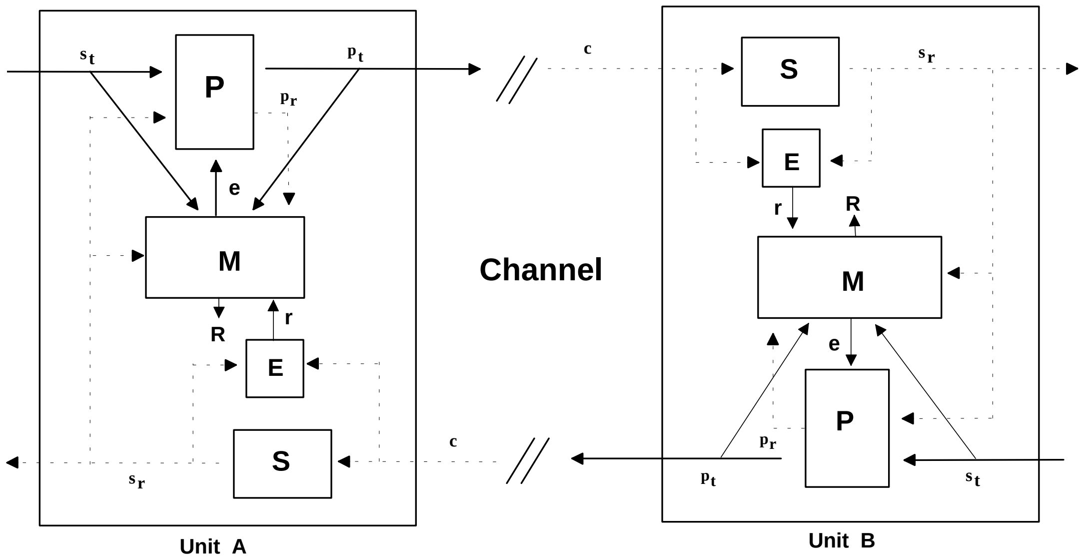


Figure 3: **Dual Reinforcement Learning in the Satellite Communications Task.** Each unit consists of a Predistorter (P), Model (M), and a Slicer (S). The Slicer maps the incoming distorted signal $c$ to the closest symbol $s_r$. The Reinforcement Signal Generator (E) forms the reinforcement signal $r$ for the Model based on how close $c$ is to $s_r$. During Model learning, based on the symbol $s_r$ and its predistorted version $p_r$, the Model learns to predict $r$. During Predistorter learning, based on the symbol $s_t$ and its predistorted version $p_t$, the Model forms the error signal $e$ for the Predistorter so that it learns to maximize the reinforcement, that is, to compensate for channel distortion.

bol $s_t$ to be transmitted. Its output $p_t$ is passed on to the Model together with $s_t$ (solid line paths in Figure 3), and the Model produces the estimate $R$ of how close to $s_t$ the signal would be after it goes through the channel. The difference between $R$ and maximum reinforcement 1.0 (indicating $c = s_t$) is then used as the error, and propagated back through the Model to its input layers (without updating weights). The error values in the model's input neurons, which are connected to the output of the Predistorter, are then back-propagated through the Predistorter, whose weights are updated. Hence, by alternating Model and Predistorter learning, the Model adapts to the channel characteristics, and in turn, trains the Predistorter to compensate for them.

### 3.4 Assumptions

Dual reinforcement learning is possible only if the communications channel meets a certain set of assumptions:

1. The number of symbols transmitted and the distribution of symbols is similar in both directions.
2. Any significant changes in channel characteristics due to environmental factors occur at a rate much smaller than the rate of symbol transmission over the channel.
3. Approximate channel characteristics are known a priori, so that the process can begin with a Predistorter and a Model that perform within bearable limits. The initial training can be done using a channel simulator (such as those based on Volterra Series methods).
4. The channel characteristics are the same in both directions, that is, on average, the distortions introduced by the channel in transmitting signals from a node A to node B are the same as those introduced by transmitting from B to A. Hence the Models at A and B learn the same task and may be assumed to be similar.
5. Predistorters at the two nodes remain approximately equivalent as time progresses. The justification for this assumption is that, initially, both Predistorters are copies of the same network, obtained via simulation (assumption 3), and further, by assumption 4 and 1, both Models learn the same channel characteristics. Since the Predistorters are trained based on the Models, they can be expected to remain similar.

# 4 Experiments

## 4.1 Simulations

To test the validity of the approach, the communications system described in section 3.3 was simulated computationally, using the Volterra series method (with memory one) as the model for the communications channel. The Predistorters are implemented as three-layer feedforward neural networks with two input, eight hidden, and two output neurons. The input and output neurons represent the real and imaginary components of the signal. The Models are also three-layer feedforward networks with four input, nine hidden, and one output neurons. The input neurons represent the real and imaginary parts of the two complex numbers $s$ and $p$, and the output is the scalar estimate of the reinforcement $R$.

The training proceeded in epochs, each consisting of a presentation of all eight symbols in random order. Backpropagation (in batch mode) with a fixed learning rate of 0.5 and momentum of 0.0 was used for all networks. Before the actual adaptation simulation, the Predistorters were trained to behave as identity functions, copying the input symbol directly to the output. This way, the communication system initially starts with approximately adequate performance. The weights of the Model were initially set randomly. The same Model and Predistorter networks were instantiated at both ends of the channel.

Distortion in the channel is modeled by passing a complex number representing the symbol through the Volterra equation modeling the channel and adding noise to the result. The memory one Volterra equation introduces a second order non-linearity, and the noise is simulated by adding a fixed number to both the real and imaginary parts of the resulting signal. At the beginning of the simulation, the noise is set to 0.0. At regular intervals (500 epochs of Model learning followed by 500 epochs of Predistorter learning), the noise was increased by a value of 0.05 until it reached 0.45, at which point the simulation was terminated.

## 4.2 Results

Figure 4 plots the error at the output of each Model network during the Model learning phases of the simulation, and Figure 5 plots the same error during the Predistorter learning phases. For each epoch, the errors were averaged over all 8 symbols, and the resulting errors were averaged over 10 different simulations.

During Predistorter learning, the Predistorter and the model are tightly coupled, and the Model error (Figure 5) serves as a measure of Predistorter performance: Provided that the Model learns the channel characteristics correctly, the closer its output is to 1.0 (representing maximum reinforcement), the closer the Predistorter is to perfect compensation. In other words, the graph in Figure 5 also represents the actual performance of the compensation unit.

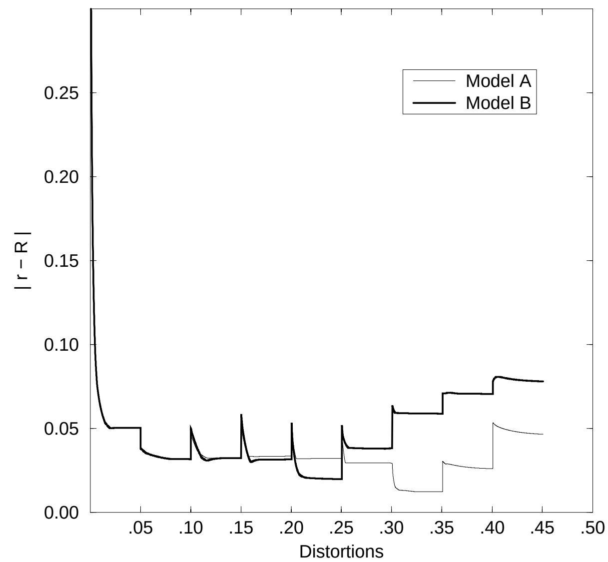


Figure 4: **Model Error during Model Learning.** The error $|r-R|$ at the output of each Model network during the Model learning phases is shown for each epoch, averaged over all 8 symbols and 10 different simulations. The simulation consisted of increasing the level of distortion at fixed increments every 500 epochs (interleaved with Predistorter training). The Models adapt to increasing distortions of up to 30%, at which point their learning starts to diverge and their predictions become inaccurate.

As can be seen from these graphs, the error increases significantly each time more distortion is added to the channel. The system catches up quickly (usually in about 150 epochs), and successfully adapts to increasing distortions of up to 0.30, or 30%, at which point its performance starts to break down. Note that in an actual communications channel, the noise to signal ratio is rarely higher than 20%. Hence, the simulated compensator successfully filters noise at levels that are 50% higher than what might be expected in an actual channel.

# 5 Discussion and Future Directions

The computational simulation results presented above suggest that the dual reinforcement learning model is a viable alternative to existing channel distortion compensator systems. The advantage of this approach is that the compensator adapts to channel distortion online, while the system is performing, hence allowing

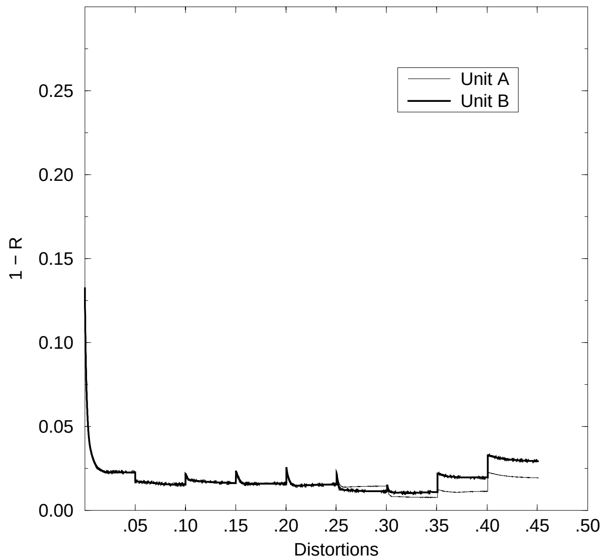


Figure 5: **Model Error during Predistorter Learning.** The error $1 - R$ at the output of each Model network is shown during the Predistorter learning phases, again averaged over all 8 symbols and 10 simulations. Because the Predistorters are trained to maximize reinforcement, this error can be interpreted as a measure of the overall performance of the compensator unit. The performance is accurate up to 0.30 distortion level, but after that the inaccurate error signals from the Model networks cause the Predistorters to diverge and become unreliable.

the communication system to continuously adapt to changing changing channel characteristics.

One potential problem with dual reinforcement learning is that the learning in the two nodes may diverge, leading into unreliable reinforcement signals. As long as the assumptions outlined in section 3.4 hold this should not happen. Indeed, Figures 4 and 5 show that the Model and the Predistorter networks in the two nodes perform almost exactly the same until quite high distortion levels. In other words, as long as the two Model and the two Predistorter networks start from the same initial configuration and the distortions are tractable, the networks will remain nearly identical throughout adaptation.

One potential problem in implementing the system in real satellite communication systems is that it uses two compensators: an equalizer (slicer) and a neural network predistorter, which appears to increase complexity and cost. However, signal filtering systems based on equalization alone suffer from noise amplification and those employing only an adaptable predisorter must be trained a priori, as was discussed in Section 1. We believe that combining a relatively simple equalizer with a continously adaptable predistorter could prove to be both inexpensive and effective.

The typical HPA, a traveling-wave tube or a GaAs FET amplifier, nonlinearly distorts both the amplitude and the phase of the amplified signal, but can be regarded as memoryless over wide range of operating conditions (Saleh 1981). However, the combination of the pulse shaping circuit (modulator) in the transmitter with the nonlinear HPA and the demodulator in the receiver results in a nonlinear system with memory (Benvenuto et al. 1993). Consequently, the next step in the development of the model is to add memory to the Predistorter to compensate for the memory of the channel. This can be accomplished by introducing a pair of Predistorter input neurons for each memory element, and shifting these inputs as new signals arrive. The simulation and evaluation of a compensator system designed for channels with memory will be addressed in future work.

Another direction for future effort involves extending the model to accomodate other complex signal formats such as quadrature amplitude modulation (QAM) and 16-PSK.

Finally, we have only demonstrated the ability of our model to adapt to incremental changes in signal distortion. It would be interesting to investigate the model's response to highly randomized and more severely discontinuous changes in the distortion, as might occur under less than optimal transmission conditions.

## 6 Conclusion

Experiments on a simulated communication channel indicate that dual reinforcement learning is a powerful new approach to on-line adaptation of signal compensators. The system learned to compensate for distortion levels up to 30% which is more than sufficient for current satellite communication systems. Moreover, the training of the two compensators did not diverge even in prolonged training, suggesting that the approach is robust and could potentially be applied to other types of communication channels as well.

## References

Bellafemina, M., and Benedetto, S. (1985). Identification and equalization of nonlinear channels for digital transmission. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, 1477–1480. Piscataway, NJ: IEEE.

Benedetto, S., and Biglieri, E. (1983). Nonlinear equalization of digital satellite channels. *IEEE Journal on Selected Areas of Communications*, 1:57–62.

Benvenuto, N., Marchesi, M., Piazza, F., and Uncini, A. (1991). Nonlinear satellite radio links equalized using neural networks. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, 1521–1524. Piscataway, NJ: IEEE.

Benvenuto, N., Piazza, F., and Uncini, A. (1993). A neural network approach to data predistortion with memory in digital radio systems. In *Proceedings of the IEEE International Conference on Communications*, 232–236. Piscataway, NJ: IEEE.

Bernardini, A., and Fina, S. D. (1993). A new predistortion technique using neural nets. *Signal Processing*, 34:231–243.

Biglieri, E., Barberis, S., and Catena, M. (1988). Analysis and compensation of nonlinearities in digital transmission systems. *IEEE Journal on Selected Areas in Communications*, 6:42–52.

Eun, C.-S. (1995). *Design and Comparison of Nonlinear Compensators*. PhD thesis, Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX.

Hertz, J., Krogh, A., and Palmer, R. G. (1991). *Introduction to the Theory of Neural Computation*. Reading, MA: Addison-Wesley.

Munro, P. W. (1987). A dual back-propagation scheme for scalar reward learning. In *Proceedings of the Ninth Annual Conference of the Cognitive Science Society*, 165–176. Hillsdale, NJ: Erlbaum.

Rao, D. H., Nikiforuk, P. N., Gupta, M. M., and Wood, H. C. (1993). Neural equalization of communication channels. In *Proceedings of the IEEE Western Canada Conference on Communications, Computers and Power in the Modern Environment*, 313–319. Piscataway, NJ: IEEE.

Saleh, A. A. M. (1981). Frequency-independent and frequency-dependent nonlinear models of TWT amplifiers. *IEEE Transactions on Communications*, 29:1715–1720.

Schetzen, M. (1980). *The Volterra and Wiener Theories of Nonlinear Systems*. New York: Wiley.

Widrow, B., and Winter, R. (1988). Neural nets for adaptive filtering and adaptive pattern recognition. *IEEE Computer*, 21:25–39.

Williams, R. J. (1988). On the use of back-propagation in associative reinforcement learning. In *Proceedings of the IEEE International Conference on Neural Networks* (San Diego, CA), vol. I, 263–270–84. Piscataway, NJ: IEEE.