

A Neural-Network-Based Channel-Equalization Strategy for Chaos-Based Communication Systems

Jiuchao Feng, Chi K. Tse, and Francis C. M. Lau

Abstract—This brief addresses the channel-distortion problem and proposes a technique for channel equalization in chaos-based communication systems. The proposed equalization is realized by a modified recurrent neural network incorporating a specific training (equalizing) algorithm.

Index Terms—Channel equalization, chaos-based communications, recurrent neural networks (RNNs).

I. INTRODUCTION

Chaotic signals, by virtue of their wideband and deterministic nature, are well suited for carrying information in a spread-spectrum communication environment [1]. However, in reality, the performance of a communication system can be seriously impaired by channel effects and noise. Theoretically, if a wideband chaotic signal is transmitted through a band-limited channel, the inevitable loss of spectral components in the received signal may cause the transmitted signal of one symbol to spread over and overlap successive symbol intervals and this effect is commonly termed *intersymbol interference* (ISI) [2]. It has been demonstrated that even simple variations of channel gain or pure phase distortions in the channel may adversely affect the transmitted signal [3], [4]. In addition to linear distortion, the transmitted signal is subject to other contaminations such as thermal noise, impulse noise, and nonlinear distortions arising from the modulation process. The idea of channel equalization is to combat the unfavorable channel effects such that the transmitted signal can be preserved with highest integrity. Recently, some approaches have been proposed for combating the effect of channel distortions in chaotic communication systems. For example, the synchronization-based method [5], [6] takes advantage of synchronization between the transmitter and receiver to estimate channel distortion. However, it is not easy to choose a suitable coupling parameter (or adaptive coupling parameter) to ensure that all of the conditional Lyapunov exponents in the demodulator are less than zero so that the receiver can approximately synchronize with the transmitter. Motivated by the lack of effective channel-equalization methods for chaos-based communication systems, this brief attempts to design a channel equalizer for chaotic signals. Linear and nonlinear distortions are considered, in addition to additive white Gaussian noise (AWGN). Specifically, we will employ a modified recurrent neural network (RNN) to realize the equalization task.

II. PRELIMINARIES

Suppose x is the transmitted signal in a communication system produced by a chaotic modulator and h is the transformation function of the channel. The output of the channel s is corrupted by noise η , which

is usually modeled as an AWGN process with a zero mean. In general, h can be modeled as a nonlinear operator, i.e., $s = h(x)$ [2]. The input to the equalizer is then $y = s + \eta$.

The problem addressed in this brief may be summarized as follows. Given the noisy and distorted sequence y , the problem is to find an equalizer such that the originally transmitted sequence x , or at least a delayed and/or phase shifted version of it, can be reconstructed. Therefore, the ideal equalization requires $\hat{x} = \delta_{tL} e^{j\theta} x$ be achieved, where t is the time instant, L is the time delay, θ is a constant phase shift, and δ is the Kronecker delta function.

The aforementioned equalization problem can be regarded as a nonlinear modeling problem. The nonlinear autoregressive moving average (NARMA), model which is a widely used tool for modeling nonlinear dynamical system, can be used to describe the said system [7]. Typically, we write $x(t) = \varpi(y(t-1), y(t-2), \dots, y(t-M), e(t-1), e(t-2), \dots, e(t-N)) + e(t)$, where $e(t)$ is the error signal at time instant t between the original and the estimated signal, ϖ is an unknown function, M and N are the time delays of the input signal and the error signal, respectively. The conditional mean of x based on the infinite past observations is $\hat{x}(t) = E[\varpi(y(t-1), y(t-2), \dots, y(t-M), e(t-1), e(t-2), \dots, e(t-N)) | y(t-1), y(t-2), \dots]$, where E denotes expectation. Suppose that the NARMA model is invertible in the sense that there exists a function ψ such that $x(t) = \psi(y(t-1), y(t-2), \dots) + e(t)$. Then, given the infinite past observations $y(t-1), y(t-2), \dots$, one can, in principle, use the above equation to estimate $e(t-j)$ in the expression for $x(t)$. In this case, the conditional mean estimate is $\hat{x}(t) = \varpi(y(t-1), y(t-2), \dots, y(t-M), e(t-1), e(t-2), \dots, e(t-N))$. Since, in practice, only a finite observation record is available, one can only perform an approximate computation of $\hat{x}(t)$ by an recursive algorithm [7], i.e., $\hat{x}(t) = \varpi(y(t-1), y(t-2), \dots, y(t-M), \hat{e}(t-1), \hat{e}(t-2), \dots, \hat{e}(t-N))$, where $\hat{e}(j) = x(j) - \hat{x}(j)$, $j = t-1, t-2, \dots, t-N$. This can be approximated by the following recurrent model [7]:

$$\hat{x}(t) = \sum_{i=1}^N u_i \varphi \left(\sum_{j=1}^M \tilde{w}_{ij} y(t-j) + \sum_{j=1}^N \tilde{w}'_{ij} (\hat{x}(t-j) - \hat{x}(t-j)) + \theta_i \right) \quad (1)$$

which is actually a special case of a general RNN to be described in the following. Here, u_i , \tilde{w}_{ij} , and \tilde{w}'_{ij} are coefficients, θ_i is a parameter, and φ is a nonlinear function.

The RNN we use in this brief is a three-layer network consisting of the input layer, the hidden layer (processing layer), and the output layer. The input vector of the input layer at time instant t is $\mathbf{v}(t)$, which is defined as $\mathbf{v}(t) = [v_1(t), v_2(t), \dots, v_M(t), v_{M+1}(t), \dots, v_{M+N+1}(t)]^T$, where $v_i(t)$, $2 \leq i \leq M+1$, is the external input which is the delayed version of y , i.e., $v_i(t) = y[t - (i-1)]$, and v_i , $M+2 \leq i \leq M+N+1$, is the feedback input of the i th input unit at time instant t . Also, N is the number of hidden-layer units and v_1 is the bias input which has been fixed at “+1” in this brief.

The internal activity of the i th hidden unit at time instant t is given by $r_i(t) = \sum_{j=1}^{M+N+1} w_{ij}(t) v_j(t)$, where $w_{ij}(t)$ is the connection weight between the i th hidden unit and the j th input unit at time instant t . At the next time step $t+1$, the output of the i th neuron $q_i(t+1)$ is computed using a nonlinear activation function, $\varphi(\cdot)$, yielding $q_i(t+1) = \varphi(r_i(t))$. In this study, we choose $\varphi(x) = \tanh(cx)$, where c is constant. Let $u_i(t)$ be the connection weight between the i th hidden unit and the output unit. The output of the output unit is given by

$$\hat{x}(t+1) = \sum_{i=1}^N [u_i(t) q_i(t+1)]. \quad (2)$$

Manuscript received April 26, 2002; revised January 12, 2003. This work was supported by the Hong Kong Polytechnic University under Research Grant G-T418. This paper was recommended by Associate Editor O. Feely.

J. Feng was with Hong Kong Polytechnic University, Hong Kong. He is now with the Department of Electronic and Information Engineering, Southwest China Normal University, Chongqing 400715, China (e-mail: fengjc@swnu.edu.cn).

C. K. Tse and F. C. M. Lau are with the Department of Electronic and Information Engineering, Hong Kong Polytechnic University, Hong Kong (e-mail: enccktse@polyu.edu.hk; encmlau@polyu.edu.hk).

Digital Object Identifier 10.1109/TCSI.2003.813966

For $1 \leq i \leq N$, and letting

$$w_{ij} = -\check{w}_{ij}u_i, \text{ for } M+2 \leq j \leq M+N+1 \quad (3)$$

and

$$w_{ij}(\text{if } M > N) = \begin{cases} \check{w}_{ij} + \check{w}'_{ij}, & \text{for } 2 \leq j \leq N+1 \\ \check{w}_{ij}, & \text{for } N+2 \leq j \leq M+1 \end{cases} \quad (4)$$

we can easily see that (1) is a special case of (2) for $M > N$. Similarly, for the cases where $M < N$ and $M = N$, the same conclusion holds.

The above estimation procedures, together with the training algorithm described in the following section, can be used to realize the equalization task.

III. TRAINING ALGORITHM

Let $d(t+1)$ be the desired output of the output unit at time instant $t+1$. The error signal $e(t+1)$ is $e(t+1) = d(t+1) - \hat{x}(t+1)$. The weight between the hidden layer and the output unit is then updated by a least-mean-square (LMS) algorithm, i.e., $u_i(t+1) = u_i(t) + \beta_1 e(t+1)q_i(t+1)$, where β_1 is the learning rate. The instantaneous sum-of-squares errors of the network is defined as $\varepsilon^o(t+1) = (1/2)e^2(t+1)$. Also, we define the local gradient of the i th hidden unit at time instant $t+1$, $\gamma_i(t+1)$, as

$$\begin{aligned} \gamma_i(t+1) &= -\frac{\partial \varepsilon^o(t+1)}{\partial r_i(t)} \\ &= e(t+1)u_i(t)\varphi'(r_i(t)) \end{aligned} \quad (5)$$

where $\varphi'(\cdot)$ is the derivative of φ with respect to its argument. According to the delta learning law, the weight w_{ij} ($i = 1, 2, \dots, N$, $j = 1, 2, \dots, M+N+1$) can be updated using $w_{ij}^-(t+1) = w_{ij}(t) + \beta_2 \gamma_i(t+1)v_j(t)$, where β_2 is the learning rate. Now, define the instantaneous sum of squared errors for the hidden-layer units as

$$\varepsilon(t) = \frac{1}{2} \sum_{k=1}^N e_k^2(t) \quad (6)$$

where $e_k(t)$ is the difference (error) in the output of the k th hidden unit before and after the weight w_{ij} is updated. Then, the instantaneous weight is updated as $w_{ij}(t+1) = w_{ij}^-(t+1) - \beta_3 [\partial \varepsilon(t) / \partial w_{ij}^-(t+1)]$, where β_3 is learning rate. Since $q_i(t+1) = \varphi(r_i(t))$ and from (6), we have

$$\frac{\partial \varepsilon(t)}{\partial w_{ij}^-(t+1)} = -\sum_{k=1}^N e_k(t) \frac{\partial q_k^+(t+1)}{\partial w_{ij}^-(t+1)} \quad (7)$$

where $q_k^+(t+1)$ is the output of the k th hidden unit after the weight w_{ij} is updated to $w_{ij}^-(t+1)$. From the above equation, we can obtain the update equation of $w_{ij}(t+1)$ as

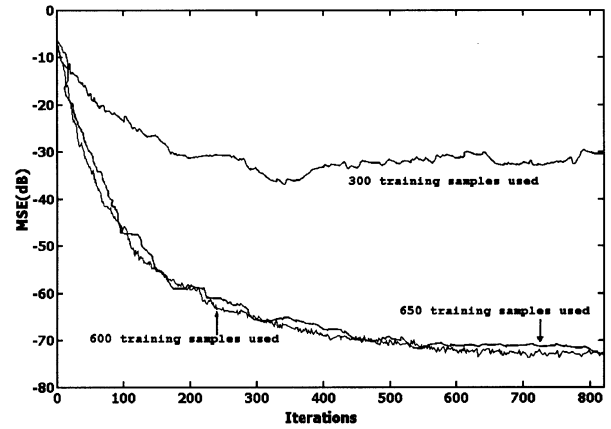
$$w_{ij}(t+1) = w_{ij}^-(t+1) + \beta_3 \sum_{k=1}^N e_k(t+1)\Omega_{ij}^k(t+1). \quad (8)$$

The above procedure is repeatedly applied to all input sample pairs during the training stage.

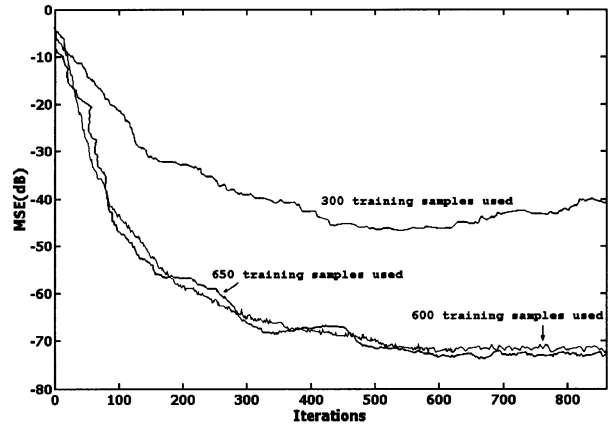
IV. SIMULATION STUDY

In this section, we simulate a chaotic communication system which is subject to channel distortion and AWGN. Our purpose is to test the ability of the proposed equalizer in combating the channel effects and noise.

Two chaotic systems will be used to evaluate the performance of the proposed equalizer in this brief. The first system is based on the Hénon map [8] and the second one is the Chua's system [9].



(a)



(b)

Fig. 1. MSE of the equalized samples for Channel I, averaged over 40 independent realizations, versus the number of iterations for (a) the Hénonmap systems; (b) the Chua's system, with different numbers of training samples.

Three channel models will be used to test the performance of the proposed equalizer in this brief. The first two channels are linear channels which can be described in the z domain by the following transformation functions:

$$H_1(z) = 1 + 0.5z^{-1} \quad (\text{Channel I}) \quad (9)$$

$$H_2(z) = 0.3 + 0.5z^{-1} + 0.3z^{-2} \quad (\text{Channel II}). \quad (10)$$

These two channel models are widely used to evaluate the performance of equalizers in communication systems [10]. Note that Channel II has a deep spectrum null at a normalized angular frequency of 2.56, which is difficult to equalize by the usual LTE [2].

The third channel to be studied is a nonlinear channel, which can be described by

$$y = \check{s} + a_1 \check{s}^2 + a_2 \check{s}^3 + \eta \quad (11)$$

where a_1 and a_2 are channel parameters which are fixed at 0.2 and -0.1 , respectively, and \check{s} is the output of the linear part of the channel, which is given by $\check{s}(t) = 0.3482x(t) + 0.8704x(t-1) + 0.3482x(t-2)$. Thus, the transformation function of the linear component can be expressed as $H(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2}$. It has been shown that these channels distort the transmitted signal drastically [11]. Furthermore, without an equalizer, the simple inverse system approach will give unacceptable performance even when the channel, besides AWGN, is an ideal allpass filter, i.e., $h = 1$.

It should be noted that the size of training sample sets is important. If the sets are too small, the equalizer cannot experience all states of the

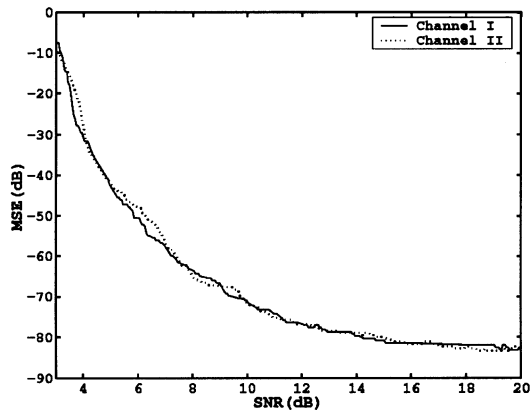


Fig. 2. MSE performance of the recovered \hat{x}_2 of the Hénonmap system versus SNR for Channel I (solid line) and Channel II (dotted line).

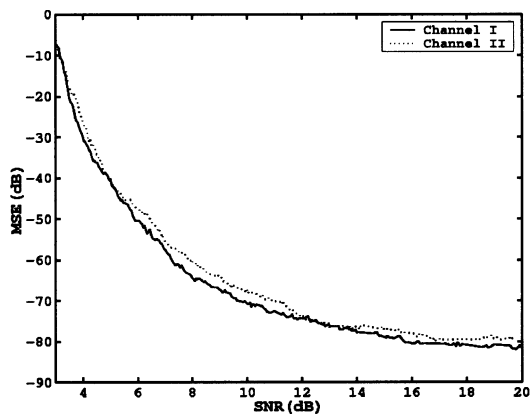
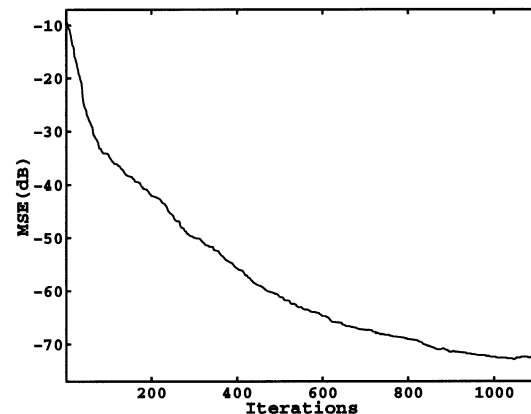


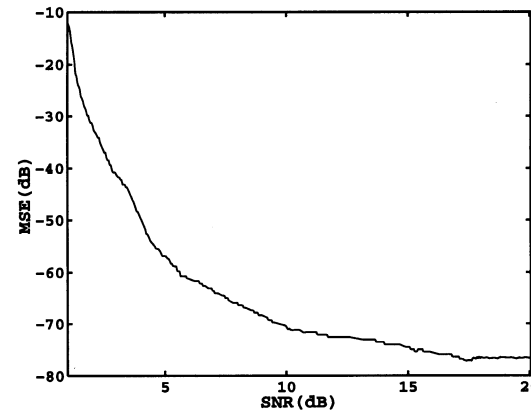
Fig. 3. MSE performance of the recovered \hat{x}_3 of the Chua's system versus SNR for Channel I (solid line) and Channel II (dotted line).

system in the given communication environment, leading to poor reconstruction of the signal. However, if the sets are too large, the training duration will be excessively long. In our simulations, we first consider different sizes of the training sample sets, and examine the results in terms of the mean-square error (MSE) as a function of the number of iterations (here, one iteration corresponds to the duration for the equalizer to go through the training once using all training samples), where the MSE is defined as $\langle (\hat{x} - x)^2 \rangle$, in which " $\langle \cdot \rangle$ " is an averaging operator. According to the requirement for the embedding dimension in Takens' theory [12], we assign each training sample pair with eight elements (seven transmitted signals and one known signal at the receiver). Also, noise is added to the training samples at a signal-to-noise ratio (SNR) of 10 dB during the training stage, and the RNN is assigned with $M = 7$ and $N = 6$. Fig. 1(a) and (b) shows the MSE of the equalized samples, averaged over 40 independent realizations, versus the number of iterations for the Hénonmap system and for the Chua's system. We can see that the smaller the training size, the more uncertain the equalizer will be and when the size of the training sets is about 600, the MSE is below -70 dB after some iterations. According to the above results, we select the size of the training sets as 600 in this study and stop the training when the MSE for all samples is below 10^{-7} .

1) When the RNN-based equalizer is applied to equalize Channels I and II, the following results are obtained. For the Hénonmap system, it is found that the equalizer completes the training in approximately 500 iterations for Channel I and 600 iterations for Channel II (i.e., $\text{MSE} < -70$ dB). The trained equalizers are then used to test their performances when actual communication takes place. Fig. 2 shows the MSE versus SNR for Channels I



(a)



(b)

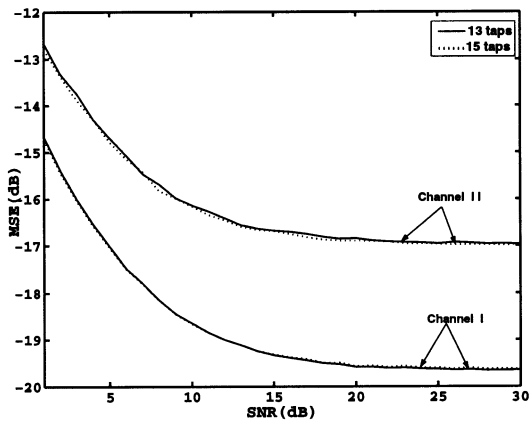
Fig. 4. Equalization of the nonlinear channel for the Hénonmap system in (a) training stage and (b) test stage.

and II. When the SNR exceeds 14 dB, the MSE of the equalized signal for the two channels is less than -80 dB. For the Chua's system, it is found that the equalizer completes the training after approximately 500 iterations for Channel I and 700 iterations for Channel II. Again, the trained equalizers are then used to test their performances when actual communication takes place. Fig. 3 shows the MSE versus SNR for Channels I and II. When the SNR is more than 14 dB, the MSE of the equalized signal for the two channels is less than -76.3 dB.

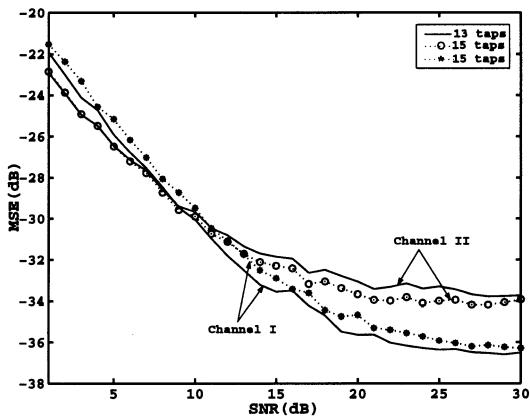
2) When the RNN-based equalizer is applied to equalize the nonlinear channel described earlier, the following results are obtained. For the Hénonmap system, Fig. 4(a) shows the MSE versus the number of iterations during training, and Fig. 4(b) shows the equalization performance of the trained equalizer. We can see from Fig. 4 that the equalizer completes its training after about 800 iterations in the training stage, and the MSE is -73.8 dB when SNR is equal to 14 dB. For the Chua's system, similarly (figures omitted for the sake of conciseness), it is found that the equalizer completes its training after about 900 iterations, and the MSE decreases with the increased SNR in the equalizing stage. When the SNR is equal to 14 dB in the equalizing stage, the MSE of the equalized signal is approximately -76 dB.

V. COMPARISONS AND DISCUSSIONS

First, we compare the proposed RNN-based equalizer with conventional 13-tap and 15-tap LTEs. The results are summarized as follows.



(a)



(b)

Fig. 5. MSE performance of linear transversal equalizers (LTE's) versus SNR for (a) the Hénon map system and (b) the Chua's system. 13-tap equalizer (solid line) and 15-tap equalizer (dotted line).

Fig. 5(a) and (b) shows the MSE performance of the equalized signal versus the SNR of the channel for the communication systems based on the Hénon map and the Chua's system, respectively. When the SNR is 14 dB, the MSE is -19.2 and -16.7 dB for Channels I and II, respectively, for the Hénon map system. Here, the RNN-based equalizer outperforms these LTEs by 60 and 63.3 dB for Channels I and II, respectively. Likewise, for the Chua's system, the MSE is -32.8 and -31.8 dB for Channels I and II, respectively. In this case, the RNN-based equalizer outperforms the LTEs by 45.5 and 44.5 dB for Channels I and II, respectively. Also, the LTEs are found completely inadequate for equalizing the nonlinear channel, and results are omitted here.

Second, we discuss the performance of the proposed equalizer based on the modified RNN in this brief. Basically, artificial neural networks (ANN) can perform complex mapping between its input and output space. Specifically, RNN can perform temporally extended tasks, for which static networks have serious limitations [13]. The salient property of the RNN is that the outputs of the hidden units are fed back at every time step to provide additional inputs. This recurrence enables the filtered data of the previous period to be used as additional inputs in the current period. This additional information of filtered input history acts as an additional guidance to evaluate the current noisy input and its signal component. Here, as an example, we use a feedforward neural network without recurrent input to realize the same equalization task for Channels I and II. In the adaption stage, the back propagation algo-

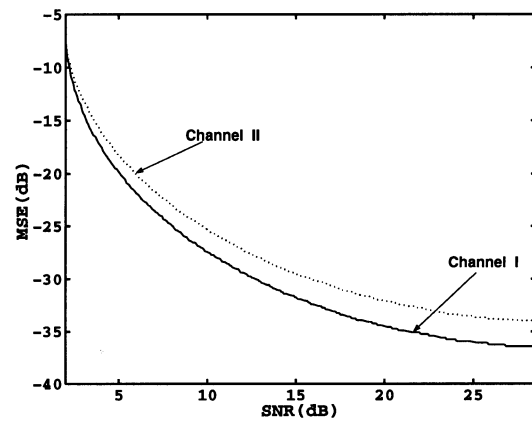


Fig. 6. Equalization performance by using an equalizer based on the feedforward neural network without recurrence using the back propagation learning algorithm in which the Hénon map is employed in the transmitter.

rihm is used to train the equalizer. Fig. 6 shows the MSE performance of the equalized signal versus the SNR of the channel. We can see by comparing Figs. 2 and 6 that when SNR is 14 dB, the RNN-based equalizer outperforms the static feedforward network by 48.9 and 51.8 dB for Channels I and II, respectively.

VI. CONCLUSION

Channel equalization in chaos-based communication systems has been studied in this brief. The main focus is the kind of channel distortion arising from linear delays as well as nonlinearity. The equalizer essentially consists of a modified RNN which incorporates a specific learning algorithm. It has been found that the proposed equalizer can effectively "undo" the channel effects, permitting the chaotic signal to be reconstructed at the receiver.

REFERENCES

- [1] F. C. M. Lau and C. K. Tse, *Chaos-Based Digital Communication Systems*. Heidelberg, Germany: Springer-Verlag, 2003.
- [2] J. G. Proakis, *Digital Communications*, 3rd ed. New York: McGraw-Hill, 1995.
- [3] A. V. Oppenheim, K. M. Cuomo, R. J. Baron, and A. E. Fredman, "Channel equalization for communication with chaotic signals," in *Chaos, Fractals and Nonlinear Signal Processing*, R. A. Katz, Ed. New York: AIP, 1996, pp. 289–301.
- [4] J. Feng and C. K. Tse, "On-line adaptive chaotic demodulator based on radial-basis-function neural networks," *Phys. Rev. E*, vol. 63, p. 026 202-1:10, 2001.
- [5] N. Sharma and E. Ott, "Combating channel distortions in communication with chaotic systems," *Phys. Lett. A*, vol. 248, pp. 347–352, 1998.
- [6] L. O. Chua, T. Yang, G. Q. Zhong, and C. W. Wu, "Synchronization of Chua's circuits with time-varying channels and parameters," *IEEE Trans. Circuits Syst. I*, vol. 43, pp. 862–868, 1996.
- [7] G. E. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 1994.
- [8] M. Hénon, "Two dimensional mapping with strange attractor," *Commun. Math. Phys.*, vol. 50, pp. 69–77, 1976.
- [9] R. Madan, Ed., *Chua's Circuit: A Paradigm for Chaos*, Singapore: World Scientific, 1993.
- [10] A. A. Khalid and S. R. Irving, "The use of neural nets to combine equalization with decoding for severe intersymbol interference channels," *IEEE Trans. Neural Networks*, vol. 5, pp. 982–988, Nov. 1994.
- [11] J. Feng, "Signal reconstruction with applications to chaos-based communications," Ph.D. dissertation, Hong Kong Polytechnic Univ., Hong Kong, 2002.
- [12] F. Takens, "Detecting strange attractors in turbulence," in *Dynamical Systems and Turbulence*, D. Rand and I. Young, Eds. Berlin, Germany: Springer-Verlag, 1981.
- [13] R. Gencay and T. Liu, "Nonlinear modeling and prediction with feedforward and recurrent networks," *Physica D*, vol. 108, pp. 119–134, 1997.