Unscented Kalman Filter-Trained Recurrent Neural Equalizer for Time-Varying Channels

Jongsoo Choi
Communications Research Lab.
McMaster University, Hamilton, Canada
Email: choi@soma.mcmaster.ca

Antonio C. de C. Lima
Electrical Engineering Department
UFBA, Salvador, Brazil
Email: acdcl@ufba.br

Simon Haykin
Communications Research Lab.
McMaster University, Hamilton, Canada
Email: haykin@mcmaster.ca

***Abstract*— Recurrent neural networks have been successfully applied to communications channel equalization because of their capability of modelling nonlinear dynamic systems. The major problems of gradient descent learning techniques, commonly employed to train recurrent neural networks, are slow convergence rates and long training sequences. This paper presents a decision feedback equalizer using a recurrent neural network trained with the unscented Kalman filter (UKF). The main features of the proposed recurrent neural equalizer are fast convergence and good performance using relatively short training symbols. Experimental results for time-varying channels are presented to evaluate the performance of the proposed approaches over a conventional recurrent neural equalizer.**

I. Introduction

It is well known that when neural networks are incorporated into a decision feedback equalizer (DFE), decision-feedback neural equalizers [1] achieve significantly improved performance in convergence speed and mean-squared error over conventional DFEs or neural equalizers without decision feedback. Neural networks provide good nonlinear mapping of the inverse model of the channel and can handle uncertainty included in the received data. Feedforward neural networks (FNNs), such as multilayer perceptrons or radial basis function networks, are mainly concerned with equalizer design because of their structural simplicity [2], [3]. However, recent research results show that recurrent neural networks (RNNs) [4] are superior to FNNs in modeling nonlinear systems and predicting time-series signals. The RNN has been successfully applied to channel equalization of communication systems [5],[6]. Kechriotis *et al.* [5] showed that an RNN-based equalizer (RNE) with a small number of neurons outperforms linear transversal equalizers (LTEs) and a FNN-based equalizer for linear and nonlinear channels. Ong *et al.* [6] showed the decision-feedback RNN equalizer (DFRNE) outperforms both the LTE and the FNN equalizer and the convergence rate of the DFRNE is faster and more robust than that of the RNE proposed in [5].

Gradient-based learning approaches, back-propagation algorithms and real-time recurrent learning (RTRL) [7], are commonly employed to train FNNs and RNNs. Major disadvantages of gradient-based methods are slow convergence rates and the long training symbols required for satisfactory channel equalization; another disadvantage is the vanishing gradient problem. For rapid channel equalization, Parisi *et al.* [8] exploited the discriminative least squares learning algorithm, minimizing a cost function that is a measure of the classification error. In [9], performance comparison among three RNEs trained with the RTRL indicated that the performance of these equalizers is indistinguishable and the RTRL may not be efficient for those equalizers.

Most of equalization results published over the past few decades have been limited to time-invariant channels. The channels in real-life mobile communications have time-varying characteristics due to fading. Although the classical equalizers perform well over fixed channels, they may not be appropriate for fast fading channels. In [10], adaptive lattice decision-feedback equalizers have been developed for time-varying channels. Recently, various equalizer structures for treating time-varying channels have been reported in [2]. The time-varying nature of fading channels can be interpreted as a dynamic system with uncertainties in its coefficients. Although FNNs have been applied to the equalization of time-varying channels, it is still a static nonlinear model. Therefore, FNN equalizers have implicit difficulty in dealing with time-varying channels. This motivates us to use RNNs rather than static models such as FNNs for time-varying channels.

In this paper we focus on a learning algorithm for an RNE with suitably fast convergence and good tracking performance. The unscented Kalman filter (UKF) is proposed as a training algorithm for the RNE. Experimental results for time-varying channels are used to evaluate the performances of the proposed approach.

II. Recurrent Neural Equalizer (RNE)

A. Decision Feedback Equalizer (DFE)

A general model of a digital communications system with a DFE is depicted in Fig.1. A sequence, $\{s(k)\}$, extracted from a source of information is transmitted, and the transmitted symbols are then buried in additive white Gaussian noise (AWGN). The channel is modelled as

$$r(k) = \sum_{i=0}^{N-1} h_i s(k-i) + v(k) \tag{1}$$

where h is the linear, finite impulse response of the channel with length N, $s(k)$ is the sequence of transmitted symbols, and $v(k)$ is the AWGN with zero mean and variance σ_0^2.

The DFE is characterized by the three integers, m, n and d, known as the feedforward order, feedback order, and decision

0-7803-7802-4/03/$17.00 © 2003 IEEE

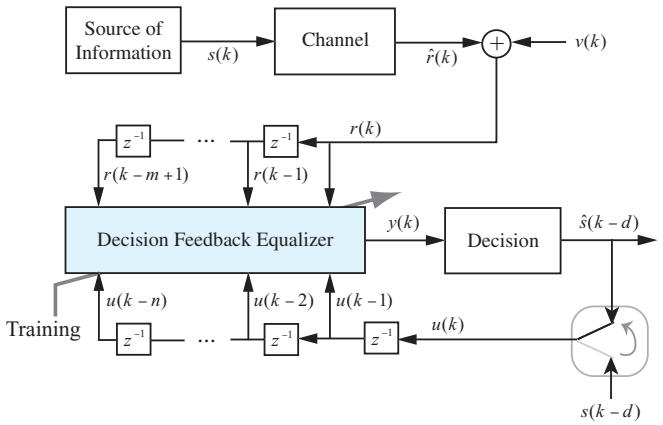

Fig. 1. A digital communications system with decision feedback equalizer

delay, respectively. The inputs to the DFE consist of the forward inputs $\mathbf{r}(k) = [r(k-1), \cdots, r(k-m+1)]^T$ and feedback inputs $\mathbf{u}(k) = [u(k-1), \cdots, u(k-n)]^T$. The output of the DFE is $y(k)$ and it is passed through a decision device to determine the estimated symbol $\hat{s}(k-d)$. It is sufficient to use feedback order, since the transmitted symbols contributing to decision of the equalizer at time k are given by $\mathbf{s}(k) = [s(k), s(k-1), \ldots, s(k-m-N+2)]^T$ for the feedforward order, $m = d+1$ [11].

B. Description of RNE

We consider a RNN model, the *Elman* network representing a simplified RNN [4],[8], that can present the standard state-space representation for a dynamic system as an RNE. The discrete state-space equation of the Elman network with a single output neuron follows the form:

$$\mathbf{x}(k) = \psi(\mathbf{r}(k), \mathbf{x}(k-1), \mathbf{W}_h) \quad (2)$$
$$y(k) = f(\mathbf{x}(k), \mathbf{w}_o) \quad (3)$$

where $r(k) \in \mathcal{R}^m$, $x(k) \in \mathcal{R}^p$ and $y(k) \in \mathcal{R}$ represent the external input vector, the state vector and the network output, respectively. $\mathbf{W}_h$ and $\mathbf{w}_o$ are the weight matrix and vector connected to the hidden layer and the output layer, respectively. The nonlinear activation function, $\psi(\cdot)$, applied to the hidden layer is the hyperbolic tangent function and $f(\cdot)$ is a linear activation function. The matrix form of the RNE with decision feedback input $\mathbf{u}(k)$, can be represented as:

$$\mathbf{x}(k) = \psi\left(\mathbf{W}_h \left[1 \;\; \mathbf{r}^T(k) \;\; \mathbf{u}^T(k) \;\; \mathbf{x}^T(k-1)\right]^T\right) \quad (4)$$
$$y(k) = \mathbf{w}_o \left[1 \quad \mathbf{x}^T(k)\right]^T \quad (5)$$
$$\hat{s}(k-d) = S(y(k)) \quad (6)$$

where unity is the bias input and $S(.)$ is the decision device. The weights of the RNE are updated by a parameter estimation algorithm to be described next.

Dynamic behavior of the RNE without the decision device is described by the following nonlinear discrete-time equations suitable for Kalman filter formulation:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \boldsymbol{\omega}(k) \quad (7)$$
$$\mathbf{y}_d(k) = \mathbf{h}(\mathbf{w}(k), \mathbf{r}(k), \mathbf{u}(k), \mathbf{x}(k-1)) + \boldsymbol{\nu}(k) \quad (8)$$

where the weights in equations (4) and (5) are reformulated as the vector $\mathbf{w}$ for convenience. The first equation, known as the process equation, specifies the state of the RNE when characterized as a stationary process corrupted by process noise $\boldsymbol{\omega}(k)$. The state of the system is given by the weight parameters of the RNE, $\mathbf{w}(k)$. The second equation, the measurement equation, represents the RNE's desired output vector $\mathbf{y}_d(k)$ as a nonlinear function of the weight vector $\mathbf{w}(k)$, the input vector $\mathbf{r}(k)$, the feedback input vector $\mathbf{u}(k)$, the recurrent node activations $\mathbf{x}(k)$, and a random measurement noise $\boldsymbol{\nu}(k)$. The desired output vector $y_d(k)$ corresponds to $s(k-d)$ in training mode and to $\hat{s}(k-d)$ in decision-directed mode.

III. Unscented Kalman Filter for RNE

The extended Kalman filter (EKF), widely used for parameter estimation of neural networks, provides first-order approximations to optimal nonlinear estimation through the linearization of the nonlinear system. These approximations can include large errors in the true posterior mean and covariance of the transformed (Gaussian) random variable, which may lead to suboptimal performance and sometimes filter divergence [12]. The unscented Kalman filter (UKF), first proposed by Julier and Uhlmann [13] and further extended by Wan and van der Merwe [12],[14], is an alternative to the EKF algorithm [15]. The UKF provides third-order approximation of process and measurement errors for Gaussian distributions and at least second-order approximation for non-Gaussian distributions. Consequently, The UKF may have better performance than the EKF. In addition, the UKF does not require the computation of Jacobians, for linearizing the process and measurement equations. This leads to a simpler implementation devoid of inverse matrix errors, but it requires more computational time than the EKF.

Foundation to the UKF is the unscented transform (UT). The UT is a method for calculating the statistics of a random variable which undergoes a nonlinear transformation [13]. Consider an L-by-1 random variable $\mathbf{x}$ that is nonlinearly transformed to yield a random variable $\mathbf{y}$ through a nonlinear function, $\mathbf{y} = f(\mathbf{x})$. In order to calculate the statistics of $\mathbf{y}$, a matrix $\boldsymbol{\chi}$ of $2L+1$ sigma vectors χ_i is formed as the followings:

$$\begin{aligned} \chi_0 &= \bar{\mathbf{x}} \\ \chi_i &= \bar{\mathbf{x}} + (\sqrt{(L+\lambda)\mathbf{P}_{\mathbf{xx}}})_i, \qquad i = 1, \ldots, L \\ \chi_i &= \bar{\mathbf{x}} - (\sqrt{(L+\lambda)\mathbf{P}_{\mathbf{xx}}})_{i-L}, \quad i = L+1, \ldots, 2L \end{aligned} \quad (9)$$

where $\bar{\mathbf{x}}$ and covariance $\mathbf{P}_{\mathbf{xx}}$ are the mean and covariance of $\mathbf{x}$, respectively, and $\lambda = \alpha^2(L+\kappa) - L$ is a scaling factor. The constant α determines the spread of the sigma points around $\bar{\mathbf{x}}$; it is set to a small positive value, typically in the range $0.001 < \alpha < 1$. The constant κ is a secondary scaling factor that is usually set to $3 - L$. The sigma points $\{\chi_i\}_{i=0}^{2L}$ are propagated through the nonlinear function

$$\mathcal{Y}_i = f(\chi_i), \qquad i = 0, \ldots, 2L. \quad (10)$$

This propagation produces a corresponding vector set that can be used to estimate the mean and covariance matrix of the nonlinear transformed vector $\mathbf{y}$. We can approximate the

mean and covariance matrix of $\mathbf{y}$ using a weighted sample mean and covariance of the posterior sigma points [12],

$$\bar{\mathbf{y}} = \sum_{i=0}^{2L} W_i^c \mathcal{Y}_i \quad (11)$$

$$\mathbf{P_{yy}} = \sum_{i=0}^{2L} W_i^c (\mathcal{Y}_i - \bar{\mathbf{y}})(\mathcal{Y}_i - \bar{\mathbf{y}})^T \quad (12)$$

where the weighting factors are given by

$$\begin{aligned} W_0^m &= \frac{\lambda}{L+\lambda} \\ W_0^c &= \frac{\lambda}{L+\lambda} + (1-\alpha^2+\beta) \\ W_i^m &= W_i^c = \frac{1}{2(L+\lambda)}, \quad i = 1, 2, \ldots, 2M \end{aligned} \quad (13)$$

In the above equations, the superscripts m and c refer to the mean and covariance, respectively. β is used to take account for prior knowledge on the distribution of $\mathbf{x}$, and $\beta = 2$ is the optimal choice for Gaussian distributions.

From the state-space model of the RNE given in (8), the cost function to be minimized in the mean-squared error (MSE) sense is:

$$J(\mathbf{w}) = [\mathbf{y}(k) - \mathbf{h}(\mathbf{z}(k))]^T \mathbf{R}^{-1}(k)[\mathbf{y}(k) - \mathbf{h}(\mathbf{z}(k))]. \quad (14)$$

If the measurement-noise covariance $\mathbf{R}(k)$ is a constant diagonal matrix, it cancels out in the algorithm, and therefore can be set arbitrarily. The process-noise covariance $\mathbf{Q}(k) = E[\boldsymbol{\omega}(k)\boldsymbol{\omega}(k)^T]$ affects the covariance rate and the tracking performance. We define $\mathbf{R}(k)$ and $\mathbf{Q}(k)$ as

$$\mathbf{R}(k) = \eta^{-1}\mathbf{I} \quad (15)$$

$$\mathbf{Q}(k) = (\lambda^{-1} - 1)\mathbf{P}(k) \quad (16)$$

where $\lambda \in (0, 1]$ is often referred to as the *forgetting factor*, in recursive least-squares (RLS) algorithms [15].

The UKF effectively evaluates the Jacobian through its sigma-point propagation, without the need to perform any analytical differentiation. Specific equations for the RNE using the UKF (RNE-UKF) algorithm are summarized below. The weight vector of the RNE-UKF and the covariance matrix are initialized with

$$\hat{\mathbf{w}}(0) = E[\mathbf{w}] \quad (17)$$

$$\mathbf{P}(0) = E[(\mathbf{w} - \hat{\mathbf{w}}(0))(\mathbf{w} - \hat{\mathbf{w}}(0)^T]. \quad (18)$$

The sigma-point calculation is given by

$$\boldsymbol{\Gamma}(k) = (L+\lambda)(\mathbf{P}(k) + \mathbf{Q}(k)) \quad (19)$$

$$\mathcal{W}(k) = [\,\hat{\mathbf{w}}(k),\, \hat{\mathbf{w}}(k) + \sqrt{\boldsymbol{\Gamma}(k)},\, \hat{\mathbf{w}}(k) - \sqrt{\boldsymbol{\Gamma}(k)}] \quad (20)$$

$$\mathcal{D}(k) = \mathbf{h}(\mathcal{W}(k),\, \mathbf{r}(k),\, \mathbf{u}(k),\, \mathbf{x}(k-1)) \quad (21)$$

$$\mathbf{y}(k) = \mathbf{h}(\hat{\mathbf{w}}(k),\, \mathbf{r}(k),\, \mathbf{u}(k),\, \mathbf{x}(k-1)) \quad (22)$$

The measurement-update equations are

$$\mathbf{P_{yy}}(k) = \sum_{i=0}^{2L} W_i^c (\mathcal{D}_i(k) - \hat{\mathbf{y}}(k))(\mathcal{D}_i(k) - \hat{\mathbf{y}}(k))^T + \mathbf{R}(k) \quad (23)$$

$$\mathbf{P_{wy}}(k) = \sum_{i=0}^{2L} W_i^c (\mathcal{W}_i(k) - \hat{\mathbf{w}}(k))(\mathcal{W}_i(k) - \hat{\mathbf{w}}(k))^T \quad (24)$$

$$\boldsymbol{\Upsilon}(k) = \mathbf{P_{wy}}(k)\mathbf{P_{yy}}^{-1}(k) \quad (25)$$

$$\hat{\mathbf{w}}(k+1) = \hat{\mathbf{w}}(k) + \boldsymbol{\Upsilon}(k)\mathbf{e}(k) \quad (26)$$

$$\mathbf{P}(k+1) = \mathbf{P}(k) - \boldsymbol{\Upsilon}(k)\mathbf{P_{yy}}(k)\boldsymbol{\Upsilon}^T(k) \quad (27)$$

The weight vector of the RNE-UKF is updated on-line with the above equations.

IV. Performance Evaluation

A. Time-Varying Channel Models

For equalization simulations, we consider two types of time-varying channel models: a nonlinear time-varying channel and a fading channel.

Channel Model 1: The nonlinear time-invariant channel model considered has the following transfer function,

$$H(z) = (h_0 + a_0(k)) + (h_1 + a_1(k))z^{-1} + (h_2 + a_2(k))z^{-2}. \quad (28)$$

In above equation the fixed channel impulse response is $\mathbf{h} = [0.3482, 0.8704, 0.3482]^T$, which is a nonminimum-phase channel [5],[8]. The channel coefficients added, $a_i(k)(i = 0, 1, 2)$, are varying with time k. A nonlinearity applied to the output of a linear filter is

$$r(k) = \hat{r}(k) + 0.2(\hat{r}(k))^2 + v(k). \quad (29)$$

The time-varying coefficients are generated by the application of a second-order Markov model in which a white Gaussian noise source drives a second-order Butterworth low-pass filter, as found in [2],[10]. In our simulations, a second-order Butterworth filter with cutoff frequency 0.1 is used. The colored Gaussian sequences used as time-varying coefficients a_i are independently generated with various standard deviations γ.

Channel Model 2: The transfer function describing a fading channel is

$$H(z) = a_0(k) + a_1(k)z^{-1} + a_2(k)z^{-2} \quad (30)$$

where channel coefficients $a_i(k)(i = 0, 1, 3)$ vary with time k. These time-varying coefficients are generated by convolving white Gaussian noise and a Butterworth filter; the same as Channel Model 1. The bandwidth of the Butterworth filter determines the relative bandwidth (fading rate) of the channel. In this simulation, we assume that the channel parameters have a nominal 2 kHz channel, 2400 symbols/s sampling rate, and a second-order Butterworth filter having a 3 dB bandwidth of 1 Hz. This time-varying scenario for fading channels was utilized in [2] and [10].

B. Experimental Results

The performance of the RNE-UKF is compared to those of both the decision feedback recurrent neural equalizer (DFRNE) [6] and the RNE trained with the EKF (RNE-EKF). In our simulations all the equalizers have three forward inputs ($m = 3$), and two decision feedback inputs ($n = 2$) for the RNE-EKF and the RNE-UKF. For comparison, the network structure is set to 4 neurons (32 weights) for the DFRNE, and 3 hidden neurons and 1 output neuron (31 weights) for both the RNE-EKF and the RNE-UKF. Information symbols are from uniformly distributed binary phase shift keying (BPSK) signals

in presence of ISI and AWGN is used in the simulations. In the DFRNE, the learning rate is 0.1 and this value ensures a stable convergence. The parameters are chosen as $\eta = 0.1$, $\epsilon = 0.01$, and $q = 0.01$ for the RNE-EKF, and $\eta = 2$ and $\lambda = 0.999$ for the RNE-UKF. All the parameters herein are chosen sub-optimally through trial-and-error.

Convergence behaviors of the three neural equalizers averaged over 200 independent trials for Channel Model 1, with $\gamma = 0.1$, are depicted in Fig. 2. Each run has a different BPSK random sequence and random starting weights for all the neural equalizers. An SNR of 16 dB is applied. We observe that the MSE curves of the RNE-EKF and the RNE-UKF are not distinguishable and the RNE-EKF and the RNE-UKF both outperform the DFRNE. This shows that Kalman filter-trained recurrent neural equalizers have an improvement in terms of both the convergence speed and the steady-state MSE. From these, the MSE values of both the RNE-EKF and the RNE-UKF are below the noise level after less than 100 training symbols for Channel Model 1. The MSE value of the DFRNE reaches around -26 dB after more than 3000 training symbols. On the other hand, the MSE values of the RNE-EKF and the RNE-UKF fall below -26 dB after approximately 700 training symbols. Fast convergence rates of the equalizers come from the superiority of Kalman filter algorithms for parameter estimation over gradient-based algorithms like the RTRL.

For bit error rate (BER) performance of Channel Model 1, we set $\gamma = 0.1$ and SNR $= 6$ dB to 16 dB at 2 dB intervals. Fig. 3 shows the BER performance for the three equalizers for Channel Model 1, averaged over 20 independent trials. In each trial, the first 100 symbols are used for training and the next 10^5 symbols are used for testing. The weight vectors of the equalizers are frozen after the training stage, and then the test is continued. It is clear that both the RNE-EKF and the RNE-UKF show better performance than the DFRNE. The RNE-UKF is better than the RNE-EKF for the nonlinear time-varying channel. This is remarkable performance, because many reported results on conventional equalizers require long training symbols (more than a few thousand) to achieve a satisfactory BER.

We next set SNR at 16 dB and perform simulations for various γ ranging from 0.1 to 0.3 in order to show BER performance of Channel Model 1 for different values of γ. For each γ value, 200 independent runs employing 200 training symbols and 10^3 test symbols are performed. Average BER and standard deviation of BER with respect to different standard deviations of γ are displayed in Fig. 4. From the average and standard deviation values of BER, we observe the following: 1) In terms of average BER, the RNE-UKF is superior to the DFRNE, and the RNE-UKF performs better than the RNE-EKF, 2) In terms of standard deviation of BER, Kalman filter-based RNEs are more robust than the DFRNE, and performance of the RNE-UKF is better than that of the RNE-EKF.

For Channel Model 2, BER performance with fading rate 1 Hz, is illustrated in Fig. 5. BER performances are averaged

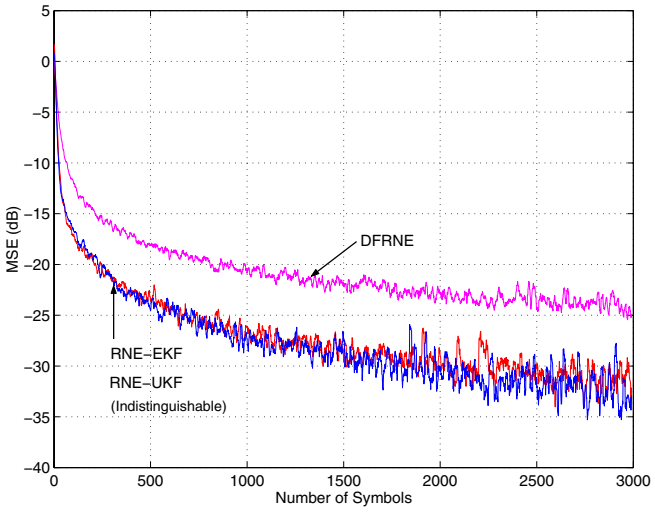

Fig. 2. Convergence properties of the equalizers with Channel Model 1.

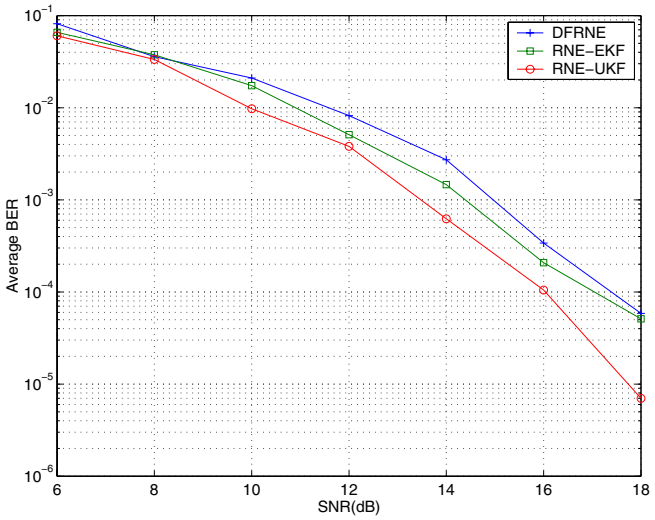

Fig. 3. BER performance of the equalizers with Channel Model 1.

over 100 independent trials where 100 training symbols and 10^3 test symbols are employed. Unlike simulations for Channel Model 1, all the equalizers still update their weight vectors to track fading characteristic of the channel. BER performance reveals that the DFRNE is not appropriate for fast fading channel equalization since it failed to equalize this channel. On the other hand, the RNE-UKF shows good channel tracking performance. Like previous results for Channel Model 1, the results depict the superiority of the RNE-UKF compared to the RNE-EKF with respect to fast fading.

C. Comparison of Computational Complexity

We represent the computational complexity in terms of the number of states (S) and weights (L). The computational time of the RTRL increases on the order $\mathcal{O}(L + S)$, and that of the EKF and the UKF increases on the order $\mathcal{O}(L^2)$ and $\mathcal{O}(L^3)$, respectively [4], [12]. Although the EKF and the UKF are more expensive than the RTRL in computational

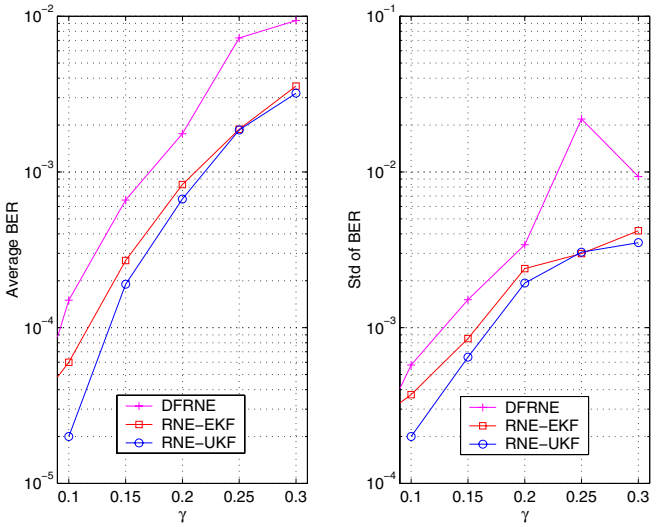

Fig. 4. BER performance comparison with changing standard deviation of Channel Model 1.

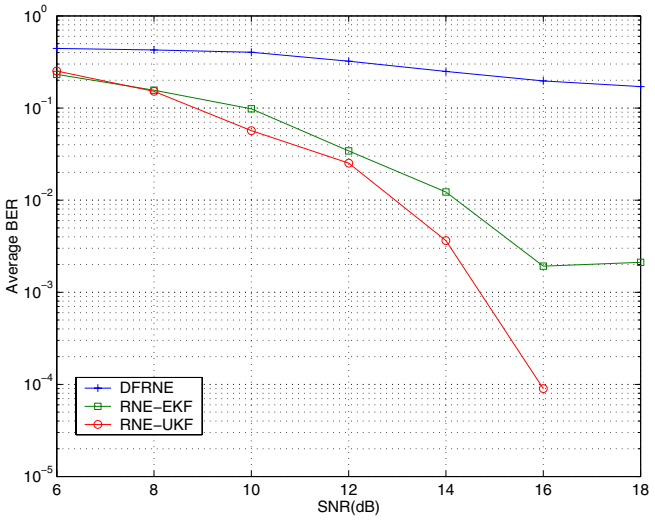

Fig. 5. BER performance for Channel Model 2 at fading rate of 1 Hz.

complexity, they lead to faster convergence rate, lower MSE level, and smaller BER, compared to the RTRL. There is an implementation versus complexity trade-off in using the EKF and the UKF algorithms. As the network size grows, the computational expense required to train transmitted symbols also increases. Fortunately, the RNEs employing the EKF and the UKF use only a small number of neurons, and also need relatively short training symbols. One notes that the computational complexity of the UKF can be reduced by using the square-root version of the UKF [12]. Its computational complexity can be reduced to $\mathcal{O}(L^2)$, i.e., the same level as the EKF algorithm.

V. Conclusions

We have presented a recurrent neural equalizer with decision feedback trained with the UKF for channel equalization over BPSK signals. Simulation results show that the RNE-UKF performed better than the DFRNE in terms of convergence rate, BER performance, and tracking capability. The RNE-UKF showed robust and marginally better performance than the RNE-EKF even though the computational cost of the former was greater. Moreover, the proposed equalizer required short training sets to attain good performance because of the UKF virtue of fast convergence. This fast convergence rate may be suitable for high-rate channel equalization. If techniques such as whitening the received data are applied to recurrent neural equalizers, better performance is expected. In short, we conclude that the RNE-UKF is more suitable for time-varying communication environments.

Acknowledgments

Antonio C. de C. Lima would like to acknowledge the Brazilian Research Council (CNPq) for financial support.

References

[1] S. Siu, G. J. Gibson, and C. F. N. Cowan, "Decision feedback equalisation using neural network structures and performance comparision with standard architecture," *IEE Proceedings: Part I*, vol. 137, no. 4, pp. 221–225, 1990.

[2] A. Zerguine, A. Shafi, and M. Bettayeb, "Multilayer perceptron-based DFE with lattice structure," *IEEE Transactions on Neural Networks*, vol. 12, pp. 532–545, May 2001.

[3] B. Mulgrew, "Applying radial basis function networks," *IEEE Signal Processing Magazine*, pp. 50–65, March 1996.

[4] S. Haykin, *Neural Networks: a Comprehensive Foundation, 2nd Ed.* Upper Saddle River, NJ: Prentice Hall, 1999.

[5] G. Kechriotis, E. Zervas, and E. S. Manolakos, "Using recurrent neural networks for adaptive communication channel equalizations," *IEEE Transactions on Neural Networks*, vol. 5, pp. 267–278, March 1994.

[6] S. Ong, C. You, S. Choi, and D. Hong, "A decision feedback recurrent neural equalizer as an infinite impulse response filter," *IEEE Transactions on Signal Processing*, vol. 45, pp. 2851–2858, November 1997.

[7] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Computation*, vol. 1, pp. 270–280, 1989.

[8] R. Parisi, E. D. D. Claudio, G. Orlandi, and B. D. Rao, "Fast adaptive digital equalization by recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, pp. 2731–2739, November 1997.

[9] J. D. Ortiz-Fuentes and M. L. Forcada, "A comparison between recurrent neural network architectures for digital equalization," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 3281–3284, 1997.

[10] F. Ling and J. G. Proakis, "Adaptive lattice decision-feedback equalizers–Their performance and application to time-variant multipath channels," *IEEE Transactions on Communications*, vol. 33, pp. 348–356, April 1985.

[11] S. Chen, B. Mulgrew, and S. McLaughlin, "Adaptive Bayesian equalizer with decision feedback," *IEEE Transactions on Signal Processing*, vol. 41, pp. 2918–2927, September 1993.

[12] E. A. Wan and R. van der Merwe, *"The unscented Kalman filter," in Kalman Filtering and Neural Networks, Edited by S. Haykin*. John Wiley and Sons, Inc., 2001.

[13] S. J. Julier and J. K. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," in *Proceedings of AeroSence: The 11th International Symposium on Aerospace/Defence Sensing, Simulation and Controls*, 1997.

[14] E. A. Wan and R. van der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications and Control Symposium (AS-SPCC)*, pp. 153–158, 2000.

[15] S. Haykin, *Adaptive Filter Theory, 4th Ed.* Upper Saddle River, NJ: Prentice Hall, 2002.