Kalman Filter-Trained Recurrent Neural Equalizers for Time-Varying Channels

Jongsoo Choi, *Student Member, IEEE*, Antonio C. de C. Lima, *Member, IEEE*, and Simon Haykin, *Fellow, IEEE*

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

***Index Terms*—Channel equalization, extended Kalman filter (EKF), recurrent neural network (RNN), time-varying channel, unscented Kalman filter (UKF).**

I. Introduction

IT IS well known that linear equalizers do not perform well on channels with deep spectral nulls or with nonlinear distortion [1]. However, nonlinear equalizers show better performance than linear equalizers in applications where the channel distortion is severe. A decision-feedback equalizer (DFE) is a nonlinear equalizer. It is widely used in situations where the intersymbol interference (ISI) is large. The DFE outperforms a linear equalizer of equivalent complexity [2]. Furthermore, the DFE can be used to increase the channel bit-rate capacity of next-generation wireless mobile communication systems [3].

When neural networks are incorporated into the DFE, decision-feedback neural equalizers [4] achieve significantly improved performance in convergence speed and mean-squared error (MSE) over the conventional DFE or the neural equalizer 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 (MLPs) or radial basis function networks are mainly concerned with equalizer design because of their structural simplicity [5], [6]. However, recent research results show that recurrent neural networks (RNNs) [7] 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 [8]–[11].

Kechriotis *et al.* [8] showed that the RNN equalizer (RNE) with a small number of neurons outperforms the linear transversal equalizer (LTE) and the FNN equalizer for linear and nonlinear channels. Ong *et al.* [9] also 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. Some researchers developed complex versions of the RNE based on a real-time recurrent learning (RTRL) algorithm to process complex signals whose inputs, weights, and outputs, as well as activation functions, are all complex-valued [12], [13].

Gradient-based learning approaches, back-propagation algorithms, and RTRL [14] for training FNN and RNN are commonly employed. Major disadvantages of gradient-based methods are slow convergence rates and long training symbols required for satisfactory performance of channel equalization; another disadvantage is the vanishing-gradients problem. For rapid channel equalization, Parisi *et al.* [10] exploited the discriminative least-squares learning algorithm, minimizing a cost function that is a measure of the classification error. In [15], performance comparison among three RNEs trained with RTRL indicated that the performance of these equalizers is indistinguishable, and RTRL may not be optimal for those equalizers.

Most equalization results published over the past few decades have been limited to time-invariant channels. The channels in real-life communications, like 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 [16], adaptive lattice DFEs have been developed for time-varying channels. Recently, various equalizer structures for treating time-varying channels have been reported in the literature [5], [17], [18]. The time-varying nature of fading channels can be interpreted as a dynamic system with uncertainties in its coefficients. Although an FNN [5] and fuzzy adaptive filter [17] have been applied to the equalization of time-varying channels, they are still static nonlinear models. Therefore, FNN or fuzzy adaptive filter-based equalizers have implicit difficulty in dealing with

Paper approved by R. A. Kennedy, the Editor for Data Communications Modulation and Signal Design of the IEEE Communications Society. Manuscript received September 17, 2002; revised August 15, 2003. The work of A. C. de C. Lima was supported by the Brazilian Research Council (CNPq). This paper was presented in part at the IEEE International Conference on Communications (ICC), Anchorage, AK, May 2003.

J. Choi was with the Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON, Canada. He is now with the School of Information Technology and Engineering, University of Ottawa, Ottawa, ON K1N 6N5, Canada (e-mail: jongsoo.choi@ieee.org).

A. C. de C. Lima is with the Electrical Engineering Department, The Federal University of Bahia, Salvador BP CEP 40210-630, Brazil (e-mail: acdcl@ufba.br).

S. Haykin is with the Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON L8S 4K1, Canada (e-mail: haykin@mcmaster.ca).

Digital Object Identifier 10.1109/TCOMM.2005.843416

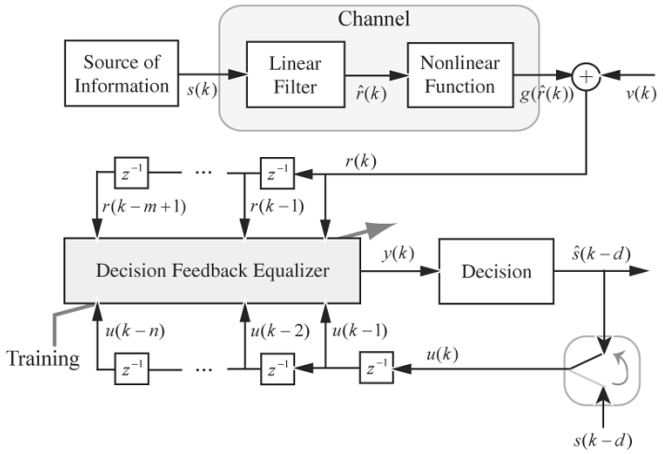


Fig. 1. Digital communications system with DFE.

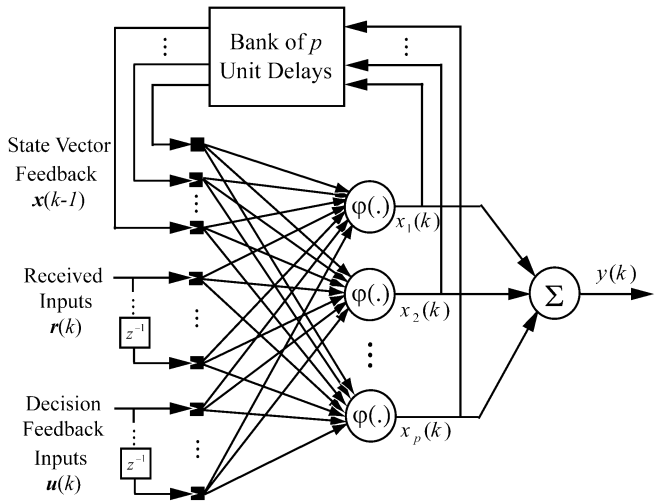


Fig. 2. Structure of the RNE with decision feedback.

time-varying channels. This motivates us to use an RNN, rather than static models such as FNN and fuzzy adaptive filter, for time-varying channels.

In this paper, we focus on learning algorithms for the RNE with suitably fast convergence and good tracking performance. The extended Kalman filter (EKF) and unscented Kalman filter (UKF) are proposed as training algorithms for the RNE. Experimental results for various time-varying channels are used to evaluate the performance of the proposed approaches over a conventional recurrent neural equalizer.

A brief explanation of the DFE and an RNN is given in Section II. Training algorithms for a recurrent neural equalizer based on Kalman filters are described in Sections III and IV. Section V presents experimental results and performance evaluation of equalization for various time-varying channels. Conclusions are given in Section VI.

II. RNE With Decision Feedback

A. Decision-Feedback Equalizer

Fig. 1 shows a general model of a digital communications system with a DFE. It includes both linear and nonlinear distortions. A sequence $\{s(k)\}$, extracted from a source of information, is transmitted, and the transmitted symbols are then corrupted by channel distortion and buried in additive white Gaussian noise (AWGN).

The channel with nonlinear distortion is modeled as

$$\begin{aligned} r(k) &= g(\hat{r}(k)) + v(k) \\ &= g\left(\sum_{i=0}^{N-1} c_i s(k-i)\right) + v(k) \end{aligned} \tag{1}$$

where $g(\cdot)$ is a nonlinear distortion, c 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. If no nonlinear distortion exists, the channel model simplifies to

$$r(k) = \hat{r}(k) + v(k). \tag{2}$$

The DFE is characterized by the three integers, m, n, and d, known as the feedforward order, feedback order, and decision delay, respectively. The inputs to the DFE therefore consist of the forward inputs $\mathbf{r}(k) = [r(k), r(k-1), \ldots, r(k-m+1)]^T$ and feedback inputs $\mathbf{u}(k) = [u(k-1), \ldots, u(k-n)]^T$. 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 n [2], [19]

$$n = N + m - d - 2 \tag{3}$$

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 $m = d+1$ [2].

B. Recurrent Neural Equalizer

RNNs have been successfully applied to channel equalization of communication systems with a fully connected recurrent structure [8], [9], [11], or with one hidden layer [10]. We consider an RNN model, the Elman network representing a simplified RNN [7], [10], [20], that can present the standard state-space representation for a dynamic system as an RNE. An architecture of the RNE using the Elman network is shown in Fig. 2, where the bias inputs to the hidden and output layers are omitted for simplicity. As depicted in Fig. 2, the RNE with decision feedback constitutes a fully connected recurrent network in cascade to a feedforward network. The discrete state-space equation of the RNE follows this form

$$\mathbf{x}(k) = \varphi(\mathbf{r}(k), \mathbf{u}(k), \mathbf{x}(k-1), \mathbf{W}_h) \tag{4}$$

$$y(k) = f(\mathbf{x}(k), \mathbf{w}_o). \tag{5}$$

In the above equations, $\mathbf{r}(k) \in \mathcal{R}^m$ and $\mathbf{u}(k) \in \mathcal{R}^n$ represent the received input vector and the decision-feedback vector, respectively. $\mathbf{x}(k-1) \in \mathcal{R}^p$ is the recurrent state vector coming from the hidden layer, and $y(k) \in \mathcal{R}^1$ is the network output. The weight matrix connected to the hidden layer is defined as $\mathbf{W}_h = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_p]^T$, where $\mathbf{w}_i = [w_{i0}, w_{i1}, \ldots, w_{ij}]^T$ ($i = 1, 2, \ldots, p$ and $j = m + n + p$). The weight vector linked to the output layer is defined as $\mathbf{w}_o = [w_0, w_1, \ldots, w_p,]^T$. The nonlinear activation function $\varphi(\cdot)$, 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

$$\mathbf{x}(k) = \varphi(\mathbf{W}_h[1 \;\; \mathbf{r}^T(k) \;\; \mathbf{u}^T(k) \;\; \mathbf{x}^T(k-1)]^T) \quad (6)$$

$$y(k) = \mathbf{w}_o^T[1 \;\; \mathbf{x}^T(k)]^T \quad (7)$$

$$\hat{s}(k-d) = \mathcal{S}(y(k)) \quad (8)$$

where unity is the bias input, and $\mathcal{S}(\cdot)$ and $\hat{s}(k-d)$ are the decision device and the symbol estimated, respectively. The weight parameters of the RNE are updated by a training algorithm, to be described next.

III. Training Algorithm Using EKF

For the training of RNNs, RTRL [14] and back-propagation through time (BPTT) [21] are most commonly used in the literature. A major disfavor of these gradient-based approaches, such as RTRL and BPTT, is slow convergence rate. In high-rate digital communication, fast convergence procedures are essential.

The training algorithms of the RNE can be regarded as parameter estimation for a nonlinear system. Among them, the EKF is the best-known method, which is derived by linearizing the system equations at each time step and applying the Kalman-filter technique to the linearized system [22]. A key property of the EKF is that it is the minimum-variance estimator for the state of a nonlinear dynamical system. The EKF applied to neural-network training leads to faster convergence than gradient-based algorithms; also, it overcomes the vanishing-gradient problem [7].

Behavior of the RNE depicted in Fig. 2 is described by the following nonlinear discrete-time equations suitable for EKF formulation:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \boldsymbol{\omega}(k) \quad (9)$$

$$y_d(k) = \mathbf{h}(\mathbf{w}(k), \mathbf{z}(k)) + \nu(k) \quad (10)$$

where the vector $\mathbf{z}(k)$ is aggregated from the vectors, which play a role as node inputs of the RNE

$$\mathbf{z}(k) = (\mathbf{r}(k), \mathbf{u}(k), \mathbf{x}(k-1), \mathbf{x}(k)) \quad (11)$$

and the weights in (6) and (7) are reformulated as the weight vector

$$\mathbf{w}(k) = \left[\mathbf{w}_1^T(k) \cdots \mathbf{w}_p^T(k) \mathbf{w}_o^T(k)\right]^T. \quad (12)$$

Equation (9), 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, $\mathbf{w}(k)$. Equation (10), the measurement equation, represents the RNE's desired output $y_d(k)$ as a nonlinear function of the weight vector $\mathbf{w}(k)$ and the input vector to the nodes $\mathbf{z}(k)$; this equation is augmented by a random measurement noise $\nu(k)$. The desired output $y_d(k)$ corresponds to $s(k-d)$ in training mode, and to $\hat{s}(k-d)$ in decision-directed mode. The learning problem of the RNE using the EKF (RNE-EKF) is to find a weight vector that can minimize the MSE of error vector $\mathbf{e}(k)$, using all the observed data.

To prepare the application of the EKF to the state-space model, the nonlinear function, for example, $\mathbf{h}(\cdot)$ given in (10), must be linearized, and the linearized model has the $L \times M$ derivative (or *Jacobian*) matrix $\mathbf{H}(k)$ for the M outputs and the L weights of the system. The Jacobian $\mathbf{H}(k)$ is defined as the partial derivatives of the M outputs with respect to the L weights, as given by

$$\mathbf{H}(k) = \frac{\partial \mathbf{h}(k, \hat{\mathbf{w}}, \mathbf{z})}{\partial \mathbf{w}(k)} = [\nabla_{\mathbf{w}} \mathbf{h}_1(k) \nabla_{\mathbf{w}} \mathbf{h}_2(k) \cdots \nabla_{\mathbf{w}} \mathbf{h}_M(k)] \quad (13)$$

where the ith derivative vector can be defined as

$$\nabla_{\mathbf{w}} \mathbf{h}_i(k) = \left[\frac{\partial h_i(k)}{\partial w_1} \frac{\partial h_i(k)}{\partial w_2} \cdots \frac{\partial h_i(k)}{\partial w_L}\right]^T. \quad (14)$$

The Jacobian $\mathbf{H}(k)$ is evaluated at $\mathbf{w}(k) = \hat{\mathbf{w}}(k)$, where $\hat{\mathbf{w}}(k)$ is the estimate of the weight vector $\mathbf{w}(k)$ computed by the EKF at time k, given the observed data up to time $k-1$ [28]. In general, the EKF solution to the parameter-estimation problem is given by the following recursion:

$$\mathbf{A}(k) = [\mathbf{R}(k) + \mathbf{H}^T(k)\mathbf{P}(k)\mathbf{H}(k)]^{-1} \quad (15)$$

$$\mathbf{K}(k) = \mathbf{P}(k)\mathbf{H}(k)\mathbf{A}(k) \quad (16)$$

$$\hat{\mathbf{w}}(k+1) = \hat{\mathbf{w}}(k) + \mathbf{K}(k)\mathbf{e}(k) \quad (17)$$

$$\mathbf{P}(k+1) = \mathbf{P}(k) - \mathbf{K}(k)\mathbf{H}^T(k)\mathbf{P}(k) + \mathbf{Q}(k). \quad (18)$$

The parameter vectors and signal vectors in (15)–(18) are described as follows:

$\mathbf{A}(k)$ M-by-M global scaling matrix;
$\mathbf{K}(k)$ L-by-M Kalman gain matrix;
$\mathbf{e}(k)$ M-by-1 error vector;
$\hat{\mathbf{w}}(k)$ estimate of the L-by-1 weight vector $\mathbf{w}(k)$;
$\mathbf{R}(k)$ M-by-M measurement covariance matrix;
$\mathbf{Q}(k)$ L-by-L process noise covariance matrix;
$\mathbf{P}(k)$ L-by-L approximate error covariance matrix.

The estimate $\hat{\mathbf{w}}(k)$, is a function of the Kalman gain matrix $\mathbf{K}(k)$. In the EKF algorithm, the measurement and process-noise covariance matrices, $\mathbf{R}(k)$ and $\mathbf{Q}(k)$, are specified for all training instances, and the approximate error-covariance matrix $\mathbf{P}(k)$ is initialized at the beginning of training as follows:

$$\mathbf{R}(0) = \eta^{-1}\mathbf{I} \quad (19)$$

$$\mathbf{Q}(0) = q\mathbf{I} \quad (20)$$

$$\mathbf{P}(0) = \epsilon^{-1}\mathbf{I}. \quad (21)$$

Here the BPTT algorithm, which is an extension of the standard back-propagation algorithm [7], can be applied to compute the Jacobian $\mathbf{H}(k)$. According to the *chain rule* of calculus, the partial derivative of the jth output with respect to the weights may computed as follows.

- At the output layer, for the weight w_{ji} linked between the jth output node and the ith node of the hidden layer

$$\frac{\partial h_j(k)}{\partial w_{ji}(k)} = \frac{\partial y_j(k)}{\partial w_{ji}(k)} = f_j'(v_j(k))x_i(k) \quad (22)$$

where the induced local field $v_j(k) = \sum_{i=0}^{p} w_{ji}(k)x_i(k)$ and $x_i(k)$ is p inputs (excluding bias) applied to output neuron j.

- At the hidden layer, for the weight w_{ih} linked between the ith node of the hidden layer and the hth input node

$$\frac{\partial h_j(k)}{\partial w_{ih}(k)} = \varphi'(v_i(k)) \sum_{j}^{M} f_j'(v_j(k)) w_{ji}(k) x_h(k). \quad (23)$$

To use BPTT in real-time fashion, the truncated BPTT with a truncation depth of t steps, denoted by BPTT(t), is available [23]. For BPTT(t), we can recast the derivative matrix as follows:

$$\mathbf{H}(k) = \sum_{i=k-t+1}^{k} \mathbf{H}_i(k) \quad (24)$$

where $\mathbf{H}_i(k)$ is the contribution from the ith step of back-propagation to the computation of the whole derivative matrix. Execution of BPTT(t) process requires storages of node inputs and outputs, as well as the weights, for $t+1$ time steps [24]. In practice, this may be handled efficiently by means of a circular buffer. For the RNE-EKF, we choose $t = 1$ for computational efficiency.

IV. Training Algorithm Using UKF

The EKF algorithm 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 [25]. The UKF, first proposed by Julier and Uhlmann [26] and further extended by Wan and van der Merwe [25], [27], is an alternative to the EKF algorithm [28]. 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, needed 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.

The unscented transform (UT) is a method for calculating the statistics of a random variable which undergoes a nonlinear transformation [26]. 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 following:

$$\begin{aligned}
\chi_0 &= \bar{\mathbf{x}} \\
\chi_i &= \bar{\mathbf{x}} + (\sqrt{(L+\lambda)\mathbf{P}_{\mathbf{xx}}})_i, \quad i = 1, \ldots, L \\
\chi_i &= \bar{\mathbf{x}} - (\sqrt{(L+\lambda)\mathbf{P}_{\mathbf{xx}}})_{i-L}, \quad i = L+1, \ldots, 2L
\end{aligned} \quad (25)$$

where $\bar{\mathbf{x}}$ and covariance $\mathbf{P}_{xx}$ 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 equal 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

$$\mathcal{Y}_i = f(\chi_i), \quad i = 0, \ldots, 2L. \quad (26)$$

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 [25]

$$\bar{\mathbf{y}} = \sum_{i=0}^{2L} W_i^m \mathcal{Y}_i \quad (27)$$

$$\mathbf{P}_{\mathbf{yy}} = \sum_{i=0}^{2L} W_i^c (\mathcal{Y}_i - \bar{\mathbf{y}})(\mathcal{Y}_i - \bar{\mathbf{y}})^T \quad (28)$$

where the weighting factors are given by

$$\begin{aligned}
W_0^m &= \frac{\lambda}{L+\lambda} \\
W_0^c &= \frac{\lambda}{L+\lambda} + (1 - \alpha^2 + \beta) \\
W_i^m &= W_i^c = \frac{1}{2(L+\lambda)}, \quad i = 1, 2, \ldots, 2L.
\end{aligned} \quad (29)$$

In the above equations, the superscripts m and c refer to the mean and covariance, respectively. β is used to take account of 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 RNN given in (9) and (10), the cost function to be minimized in the MSE sense is

$$\begin{aligned}
J(\mathbf{w}(k)) = [\mathbf{y}_d(k) - \mathbf{h}(\mathbf{w}(k), \mathbf{z}(k))]^T \mathbf{R}^{-1} \\
\times [\mathbf{y}_d(k) - \mathbf{h}(\mathbf{w}(k), \mathbf{z}(k))].
\end{aligned} \quad (30)$$

If the measurement-noise covariance $\mathbf{R}$ 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 convergence rate and the tracking performance. We define $\mathbf{Q}$ as

$$\mathbf{Q}(k) = \left(\lambda_{\mathrm{RLS}}^{-1} - 1\right) \mathbf{P}(k) \quad (31)$$

where $\lambda_{\mathrm{RLS}} \in (0, 1]$ is often referred to as the *forgetting factor*, in recursive least-squares (RLS) algorithms [28].

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

$$\hat{\mathbf{w}}(0) = E[\mathbf{w}] \quad (32)$$

$$\mathbf{P}(0) = E[(\mathbf{w} - \hat{\mathbf{w}}(0))(\mathbf{w} - \hat{\mathbf{w}}(0)^T]. \quad (33)$$

The sigma-point calculation is given by

$$\boldsymbol{\Gamma}(k) = (L+\lambda)(\mathbf{P}(k) + \mathbf{Q}(k)) \quad (34)$$

$$\mathcal{W}(k) = [\hat{\mathbf{w}}(k), \hat{\mathbf{w}}(k) + \sqrt{\boldsymbol{\Gamma}(k)}, \hat{\mathbf{w}}(k) - \sqrt{\boldsymbol{\Gamma}(k)}] \quad (35)$$

$$\mathcal{D}(k) = \mathbf{h}(\mathcal{W}(k), \mathbf{r}(k), \mathbf{u}(k), \mathbf{x}(k-1), \mathbf{x}(k)) \quad (36)$$

$$\mathbf{y}(k) = \mathbf{h}(\hat{\mathbf{w}}(k), \mathbf{r}(k), \mathbf{u}(k), \mathbf{x}(k-1), \mathbf{x}(k)). \quad (37)$$

The measurement-update equations are

$$\mathbf{P}_{\mathbf{yy}}(k) = \sum_{i=0}^{2L} W_i^c(\mathcal{D}_i(k) - \mathbf{y}(k))(\mathcal{D}_i(k) - \mathbf{y}(k))^T + \mathbf{R}(k) \tag{38}$$

$$\mathbf{P}_{\mathbf{wy}}(k) = \sum_{i=0}^{2L} W_i^c(\mathcal{W}_i(k) - \hat{\mathbf{w}}(k)) \times (\mathcal{D}_i(k) - \mathbf{y}(k))^T \tag{39}$$

$$\boldsymbol{\Upsilon}(k) = \mathbf{P}_{\mathbf{wy}}(k)\mathbf{P}_{\mathbf{yy}}^{-1}(k) \tag{40}$$

$$\hat{\mathbf{w}}(k+1) = \hat{\mathbf{w}}(k) + \boldsymbol{\Upsilon}(k)\mathbf{e}(k) \tag{41}$$

$$\mathbf{P}(k+1) = \mathbf{P}(k) - \boldsymbol{\Upsilon}(k)\mathbf{P}_{\mathbf{yy}}(k)\boldsymbol{\Upsilon}^T(k). \tag{42}$$

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

V. Performance Evaluation for Time-Varying Channels

A. Time-Varying Channel Models

For equalization simulations, we consider three types of time-varying channel models: linear and nonlinear time-varying channels, as well as a fading channel.

Channel Model 1: A linear time-invariant channel model is a nonminimum-phase channel with transfer function

$$C(z) = c_0 + c_1 z^{-1} + c_2 z^{-2}$$

where the channel impulse response is $\mathbf{c} = [c_0\ c_1\ c_2]^T$. We use $\mathbf{c} = [0.3482\ 0.8704\ 0.3482]^T$, which is generally used for channel equalization in the literature [8], [10], [29], because this type of channel is encountered in real communication systems. In these simulations, our interests also includes time-varying channels. For the purpose of a time-varying channel, the transfer function above is modified as

$$C(z) = (c_0 + a_0(k)) + (c_1 + a_1(k))z^{-1} + (c_2 + a_2(k))z^{-2}$$

where the added channel coefficients $a_i(k)(i = 0, 1, 2)$ are varying with time k. 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 lowpass filter, as found in [5] and [16]–[18]. 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. In the following, time-varying coefficients have Gaussian distribution with c_i mean and variance dependent on γ. Readers can find the source code for this kind of time-varying coefficients in [17].

Channel Model 2: The nonlinear channel used in [8] and [10] is modeled as

$$\begin{aligned} r(k) &= g(\hat{r}(k)) + v(k) \\ &= \hat{r}(k) + 0.2(\hat{r}(k))^2 + v(k) \end{aligned}$$

where a nonlinearity is applied to the output of a linear filter, whose transfer function and time-varying coefficients are the same as Channel Model 1. This channel is often encountered in satellite communication links, as mentioned in [8].

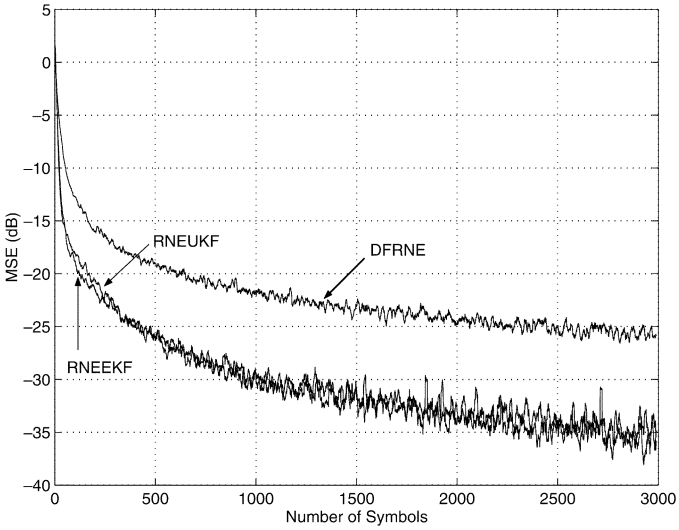


Fig. 3. Convergence properties of equalizers with Channel Model 1 under SNR = 16 dB.

Channel Model 3: The transfer function of a discrete-time channel model is described by

$$C(z) = a_0(k) + a_1(k)z^{-1} + a_2(k)z^{-2}.$$

This channel model represents a fading channel with $a_i(k)$ varying 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. We assume that we have a nominal 2 kHz channel, 2400 symbols/s sampling rate, and a second-order Butterworth filter having a 3 dB bandwidth of 0.5 or 1.0 Hz. This time-varying scenario for fading channels was used in [5], [16], and [18].

B. Experimental Results

The performance of the RNE-EKF and the RNE-UKF is compared with that of the DFRNE [9] using a fully recurrent network trained with RTRL. 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. Decision delay is set to $d = 2$ for the following simulations. For comparison, the network structure is set to four neurons (32 weights) for the DFRNE, and three hidden neurons and one output neuron (31 weights) for both the RNE-EKF and the RNE-UKF. Information symbols 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 $\lambda_{\mathrm{RLS}} = 0.999$ for the RNE-UKF. All the parameters herein are chosen suboptimally 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. 3. 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

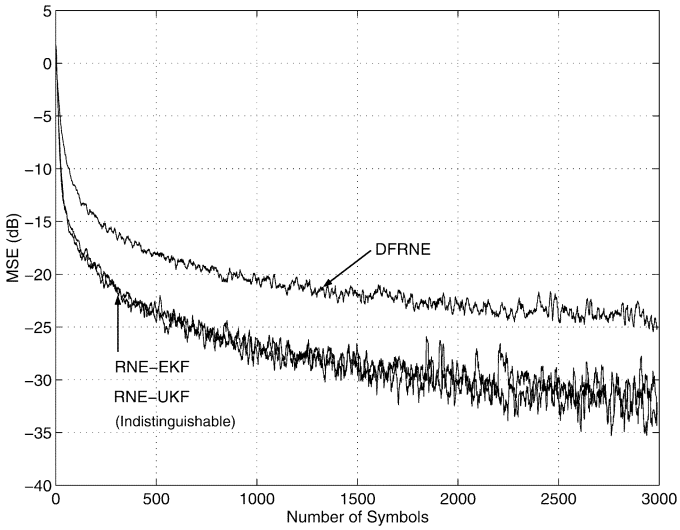


Fig. 4. Convergence properties of equalizers with Channel Model 2 under SNR = 16 dB.

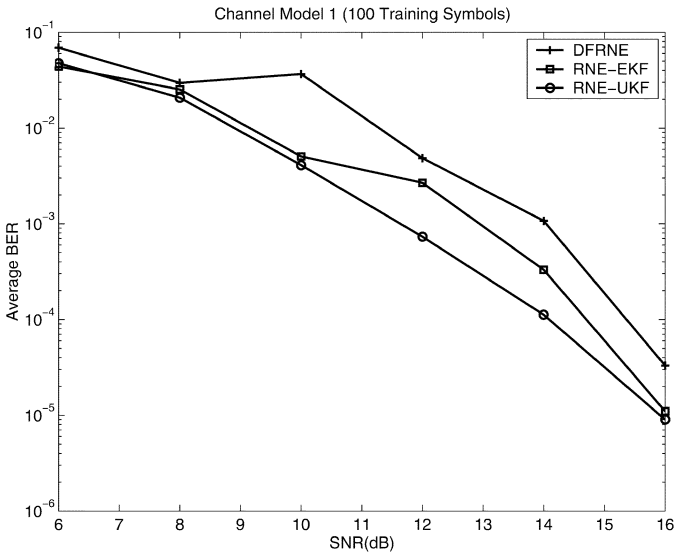


Fig. 5. BER performance of RNEs with Channel Model 1.

MSE curves of the RNE-EKF and the RNE-UKF are not distinguishable, and both the RNE-EKF and the RNE-UKF outperform the DFRNE. This shows that Kalman filter-trained RNEs have an improvement in terms of both the convergence speed and the steady-state MSE. For Channel Model 1, MSE value of the DFRNE reaches around -26 dB after 3000 training symbols. On the other hand, MSE values of the RNE-EKF and the RNE-UKF fall below -26 dB after approximately 500 training symbols. As shown in Fig. 4, Channel Model 2 behaves in a similar way. Fast convergence rates of the equalizers come from the superiority of Kalman-filtering algorithms for parameter estimation over gradient-based algorithms like RTRL.

For bit-error rate (BER) performance of Channel Models 1 and 2, we set $\gamma = 0.1$ and SNR = 6 to 16 dB at 2 dB intervals. Figs. 5 and 6 show the BER performance for the three equalizers for Channel Models 1 and 2, averaged over 10 independent trials. In each trial, the first 100 symbols are used for training, and the next 10^5 symbols are used for testing. The

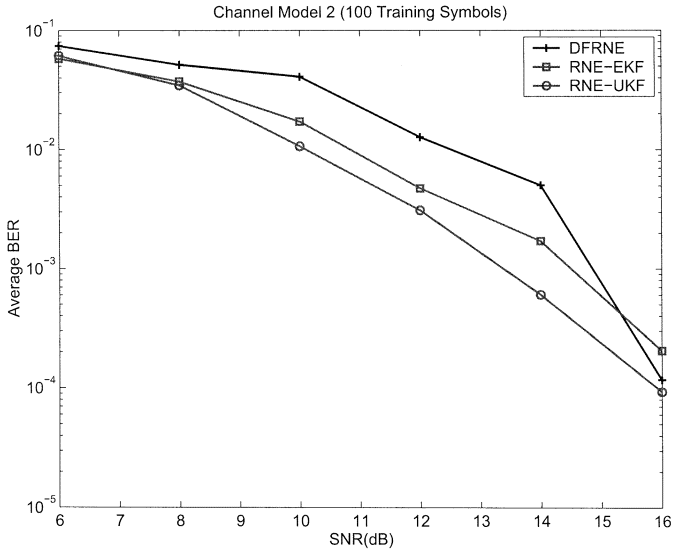


Fig. 6. BER performance of RNEs with Channel Model 2.

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 marginally better than the RNE-EKF for both linear and nonlinear time-varying channels. This is a remarkable performance, because many reported results on conventional equalizers require long training sequences (more than 1000) to achieve a satisfactory BER.

We next set SNR at 16 dB and perform simulations for various γ ranging from 0.05 to 0.3 in order to show BER performance of channels 1 and 2 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 deviation γ are displayed in Fig. 7(a) and (b). From the average and standard deviation values of BER, we observe the following: 1) in terms of average BER, Kalman filter-based equalizers are superior to the DFRNE, and the RNE-UKF performs better than the RNE-EKF; 2) in terms of standard deviation of BER, the Kalman filter-based equalizers are more robust with respect to AWGN than the DFRNE, and performance of the RNE-UKF is better than that of the RNE-EKF.

For Channel Model 3, we test channel-tracking performance for the three RNEs, because tracking is a *steady-state phenomenon*, to be contrasted with convergence, which is a transient phenomenon [28]. As we expected, the RNE-EKF and the RNE-UKF provide faster channel-tracking capabilities than the corresponding DFRNE. Fig. 8(a) shows a fading-channel prototype drawn at fading rate of 0.5 Hz (upper figure) and the absolute values of the root of $a_0(k)x^2 + a_1(k)x + a_2(k)$ (lower figure), so that a burst of errors may be associated with rapid changes of these roots. The equalizers are in training phase until $k = 2500$, and then the equalizers are switched to tracking phase at $k = 2501$. In Fig. 8(a), channel-tracking properties for three equalizers are evaluated in both the training phase and tracking phase at an SNR of 16 dB. This result verifies that the convergence properties of the RNE-EKF and the RKE-UKF are much lower than that of the DFRNE for both training and

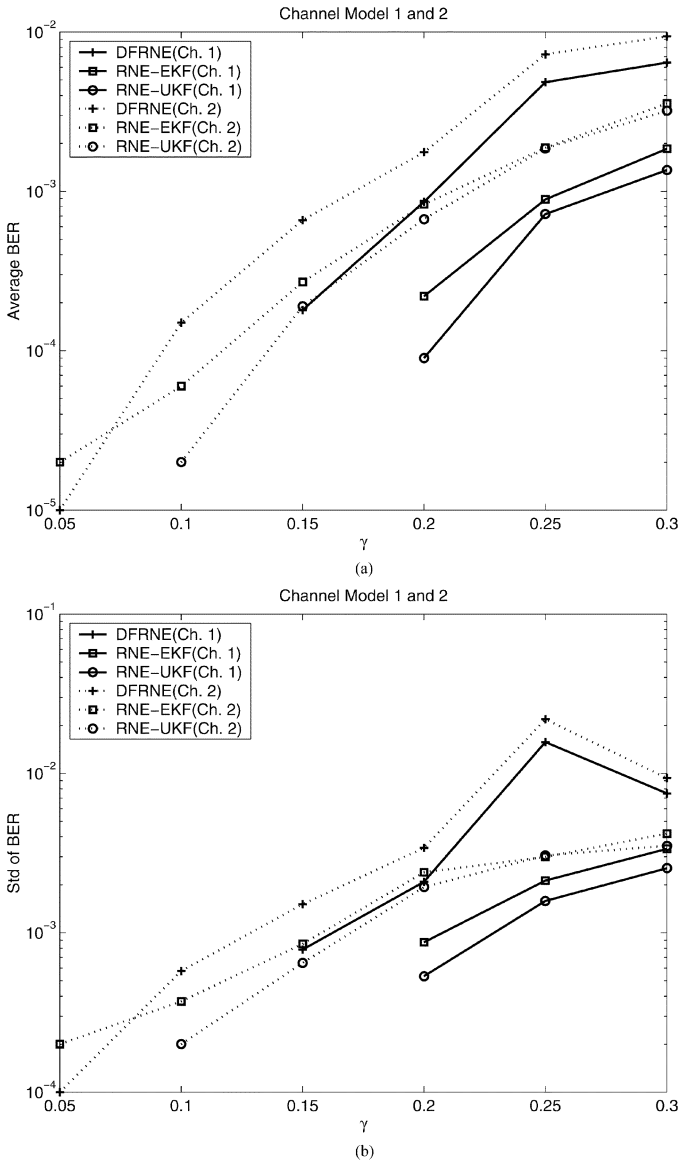


Fig. 7. BER performance comparison with changing standard deviation of time-varying channels. (a) Average BER versus standard deviation. (b) Standard deviation of BER versus standard deviation.

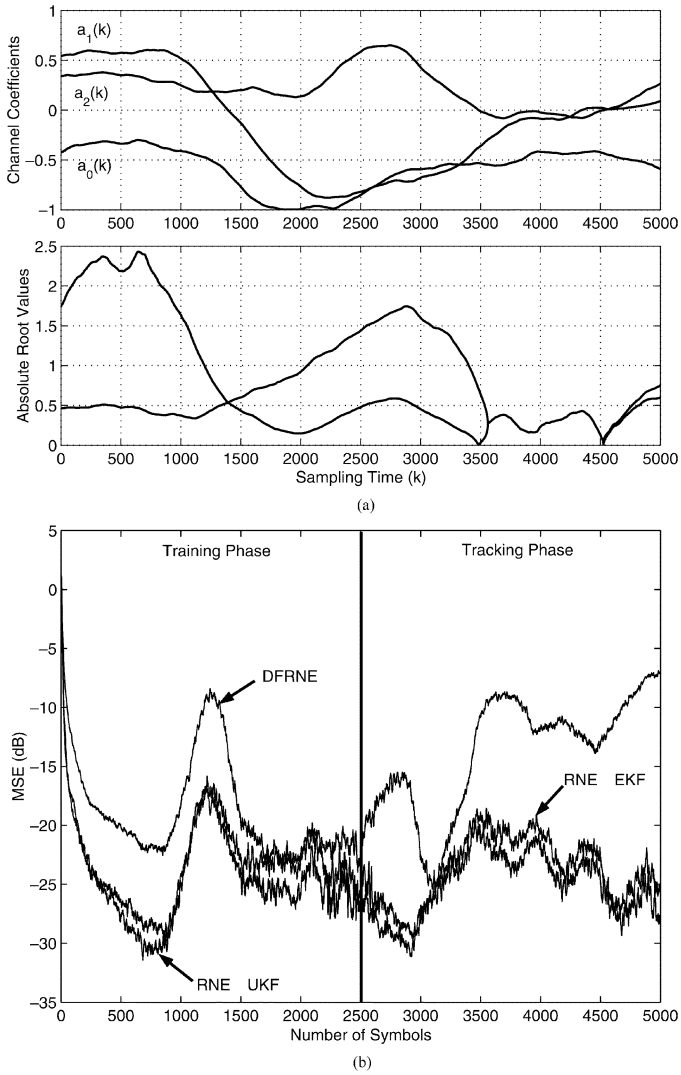


Fig. 8. Channel tracking performance for Channel Model 3. (a) Channel characteristics drawn over fading rate 0.5 Hz. (b) Tracking properties under SNR $=$ 16 dB.

tracking phases. Moreover, the RNE-UKF presents a marginally faster recovery than the RNE-EKF. The BER performance, with fading rates of 0.5 and 1.0 Hz, is illustrated in Fig. 9. BER performances are averaged over 100 independent trials, where 100 training symbols and 10^3 test symbols are employed. Unlike simulations for Channel Models 1 and 2, all the equalizers still update their weight vectors to track fading characteristics of the channel. BER performance reveals that the DFRNE is not appropriate for fast-fading channel equalization, since the DFRNE failed to equalize this channel. On the other hand, the Kalman filter-based equalizers show good channel-tracking performance. Like previous results for Channel Models 1 and 2, the results depict the superiority of the RNE-UKF compared with the RNE-EKF, with respect to both fading rates of 0.5 and 1.0 Hz. It can be noted that the superiority of the RNE-EKF and the RNE-UKF compared with the DFRNE has consistency in both channel-tracking performance and BER performance.

From simulation results of different time-varying channels, we may conclude that the RNE-EKF and the RNE-UKF are more suitable for time-varying communication environments than the DFRNE [9], which outperformed conventional DFEs based on the LMS or RLS algorithms, as well as conventional neural equalizers for both linear and nonlinear fixed channels.

C. Comparison of Computational Complexity

The computational complexity is one of the important issues in implementing channel equalizers. In this paper, we represent the computational complexity in terms of the number of states (S) and weights (L). The computational time of the RTRL increases in the order $\mathcal{O}(L+S)$, and that of the EKF and the UKF increases in the order $\mathcal{O}(L^2)$ and $\mathcal{O}(L^3)$, respectively [7], [25]. Although the EKF and the UKF are more expensive than the RTRL in computational complexity, they lead to better convergence rate, MSE level, and BER, compared with the RTRL. There is an implementation versus complexity tradeoff in using EKF and UKF algorithms. As the network size grows, the computational expense required to train transmitted symbols also

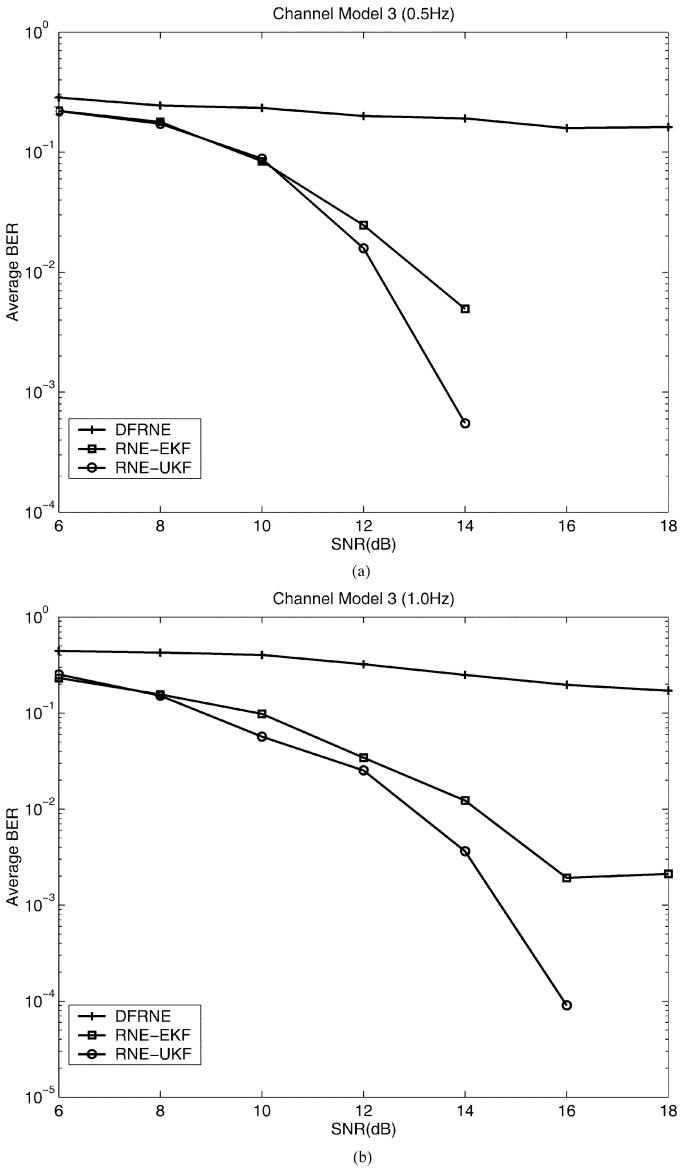


Fig. 9. BER performance for Channel Model 3. (a) Fading rate 0.5 Hz. (b) Fading rate 1 Hz.

increases. Fortunately, the RNE employing the EKF and the UKF uses only a small number of neurons, and also needs relatively short training symbols at the training stage. One notes that the computational complexity of the UKF can be reduced by using the square-root UKF algorithm [25]. Its computational complexity can be reduced to $\mathcal{O}(L^2)$, i.e., the same level as the EKF algorithm.

VI. Conclusions

We have presented RNEs with decision feedback trained with Kalman filters, called the RNE-EKF and the RNE-UKF, for channel equalization over BPSK signals. The performance of the equalizers presented in this paper has been assessed for equalization of linear and nonlinear time-varying channels, and has been compared with that of the DFRNE. Simulation results showed that the RNE-EKF and 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 equalizers required relatively short training sets (100 or 200 symbols) to attain good performance, because of the Kalman filter algorithms, by virtue of their 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 RNEs, better performance is expected. In short, we conclude that the RNE-EKF and the RNE-UKF are more suitable for time-varying communication environments than existing conventional equalizers or feedforward neural equalizers.

References

[1] S. Qureshi, "Adaptive equalization," *Proc. IEEE*, vol. 73, pp. 1349–1387, Sep. 1985.
[2] S. Chen, B. Mulgrew, and S. McLaughlin, "Adaptive Bayesian equalizer with decision feedback," *IEEE Trans. Signal Process.*, vol. 41, pp. 2918–2927, Sep. 1993.
[3] S. Ariyavisitakul, N. R. Sollenberger, and L. J. Greenstein, "Tap-selectable decision-feedback equalization," *IEEE Trans. Commun.*, vol. 45, pp. 1497–1500, Dec. 1997.
[4] S. Siu, G. J. Gibson, and C. F. N. Cowan, "Decision feedback equalization using neural network structures and performance comparision with standard architecture," *IEE Proc.*, pt. 1, vol. 137, no. 4, pp. 221–225, 1990.
[5] A. Zerguine, A. Shafi, and M. Bettayeb, "Multilayer perceptron-based DFE with lattice structure," *IEEE Trans. Neural Netw.*, vol. 12, pp. 532–545, May 2001.
[6] B. Mulgrew, "Applying radial basis function networks," *IEEE Signal Process. Mag.*, pp. 50–65, Mar. 1996.
[7] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 1999.
[8] G. Kechriotis, E. Zervas, and E. S. Manolakos, "Using recurrent neural networks for adaptive communication channel equalizations," *IEEE Trans. Neural Netw.*, vol. 5, pp. 267–278, Mar. 1994.
[9] S. Ong, C. You, S. Choi, and D. Hong, "A decision feedback recurrent neural equalizer as an infinite impulse response filter," *IEEE Trans. Signal Process.*, vol. 45, pp. 2851–2858, Nov. 1997.
[10] R. Parisi, E. D. D. Claudio, G. Orlandi, and B. D. Rao, "Fast adaptive digital equalization by recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, pp. 2731–2739, Nov. 1997.
[11] K. Hacioglu, "An improved recurrent neural network for M-PAM symbol detection," *IEEE Trans. Neural Netw.*, vol. 8, pp. 779–783, May 1997.
[12] S. Ong, S. Choi, C. You, and D. Hong, "A complex version of a decision feedback recurrent neural equalizer as an infinite impulse filter," in *Proc. GLOBECOM*, 1997, pp. 57–61.
[13] H. R. Jiang and K. S. Kwak, "On modified complex recurrent neural network adaptive equalizer," *J. Circuits, Syst., Computers*, vol. 11, no. 1, pp. 93–101, 2002.
[14] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Computat.*, vol. 1, pp. 270–280, 1989.
[15] J. D. Ortiz-Fuentes and M. L. Forcada, "A comparison between recurrent neural network architectures for digital equalization," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1997, pp. 3281–3284.
[16] F. Ling and J. G. Proakis, "Adaptive lattice decision-feedback equalizers—Their performance and application to time-variant multipath channels," *IEEE Trans. Commun.*, vol. COM-33, pp. 348–356, Apr. 1985.
[17] Q. Liang and J. M. Mendel, "Equalization of nonlinear time-varying channels using type-2 fuzzy adaptive filters," *IEEE Trans. Fuzzy Syst.*, vol. 8, pp. 551–563, Oct. 2000.
[18] C. Cowan and S. Semnani, "Time-variant equalization using a novel nonlinear adaptive structure," *Int. J. Adaptive Control, Signal Process.*, vol. 12, no. 2, pp. 195–206, 1998.
[19] M. Solazzi, A. Uncini, E. D. D. Claudio, and R. Parisi, "Complex discriminative learning Bayesian neural equalizer," in *Proc. IEEE Int. Symp. Circuits Syst.*, 1999, pp. 343–346.
[20] J. Elman, "Finding structure in time," *Cognitive Sci.*, vol. 14, pp. 179–211, 1990.

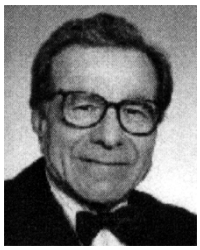
[21] P. J. Werbos, "Backpropagation through time: What it does and how to do it," *Proc. IEEE*, vol. 78, pp. 1550–1560, Oct. 1990.
[22] Y. Iiguni, H. Sakai, and H. Tokumaru, "A real-time learning algorithm for a multilayered neural network based on the extended Kalman filter," *IEEE Trans. Signal Process.*, vol. 40, pp. 959–966, May 1992.
[23] R. J. Williams and J. Peng, "An efficient gradient-based algorithm for on-line training of recurrent trajectories," *Neural Computat.*, vol. 2, pp. 490–501, 1990.
[24] L. A. Feldkamp and G. V. Puskorius, "A signal processing framework based on dynamic neural networks with application to problems in adaptation, filtering and classification," *Proc. IEEE*, vol. 86, pp. 2259–2277, Sep. 1998.
[25] E. A. Wan and R. van der Merwe, "The unscented Kalman filter," in *Kalman Filtering and Neural Networks*, S. Haykin, Ed. New York: Wiley, 2001.
[26] S. J. Julier and J. K. Uhlmann. A new extension of the Kalman filter to nonlinear systems. presented at *AeroSence: 11th Int. Symp. Aerospace/Defence Sensing, Simulation, Controls*. [Online]. Available: http://citeseer.ist.psu.edu/julier97new.html
[27] E. A. Wan and R. van der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proc. IEEE Adaptive Syst. Signal Process., Commun., Control Symp.*, 2000, pp. 153–158.
[28] S. Haykin, *Adaptive Filter Theory*, 4th ed. Upper Saddle River, NJ: Prentice-Hall, 2002.
[29] S. Chen, B. Mulgrew, and S. McLaughlin, "Adaptive equalization of finite nonlilear channels using multilayer perceptrons," *Signal Process.*, vol. 20, pp. 107–119, 1990.

Jongsoo Choi (M'98–S'03) was born in Chungnam, Korea, in 1969. He received the Ph.D. degree in control engineering from Chonbuk National University, Chonju, Korea, in 1996. Currently, he is working toward a second Ph.D. degree in the School of Information Technology and Engineering, University of Ottawa, Ottawa, ON, Canada, after receiving the M.A.Sc. degree from McMaster University, Hamilton, ON, Canada.

He was a Visiting Scholar at University of Waterloo, Waterloo, ON, Canada, in 2000, and was an Adjunct Associate Professor at Uiduk University, Gyeongju, Korea, in 1999. From 1994 to 1999, as a Senior Researcher, he was with the Automation Research Division, Research Institute of Industrial Science and Technology (RIST), Pohang, Korea. His research interests lie in adaptive signal processing for telecommunications and control, wireless communications, and intelligent systems.

Antonio C. de C. Lima (S'88–M'95) was born in Bahia, Brazil, in 1963. He received the B.Sc. degree in electrical engineering from the Federal University of Bahia (UFBA), Bahia, Brazil, in 1988, the M.Sc. degree from the State University of Campinas (UNICAMP), Campinas, Brazil, in 1990, and the Ph.D. degree in electronic engineering from the University of Kent, Canterbury, U.K., in 1994.

He joined the UFBA Electrical Engineering Department in 1994 as a Professor of Applied Electromagnetism. He recently spent one year at the Adaptive Systems Laboratory, McMaster University, Hamilton, ON, Canada, doing postdoctoral work in signal processing applied to communication systems.



Simon Haykin (F'86) received the B.S. degree with First-Class Honors in 1953, the Ph.D. degree in 1956, and the D.Sc. degree in 1967, all in electrical engineering, from the University of Birmingham, Birmingham, U.K. He also received the Honorary Doctor of Technical Sciences degree from ETH, Zurich, Switzerland, in 1999.

He is the founding director of Communications Research Laboratory, McMaster University, Hamilton, ON, Canada, where he became a Professor in 1996. He is the author/editor of over 500 technical papers and over 40 books, including the popular undergraduate and graduate textbooks *Adaptive Filter Theory* (Englewood Cliffs, NJ: Prentice-Hall, 4th ed., 2002), *Neural Networks: A Comprehensive Foundation* (Englewood Cliffs, NJ: Prentice-Hall, 2nd ed., 1999), *Communication Systems* (New York: Wiley, 4th ed., 2001), and *Signal and Systems* (with B. Van Veen, New York: Wiley, 2nd ed., 2003). He is the series editor of the Wiley-Interscience Series on Adaptive and Learning Systems for Signal Processing, Communications, and Control.

Dr. Haykin was the recipient of the Ross Medal from the Engineering Institute of Canada in 1978. In 1980, he was elected a Fellow of the Royal Society of Canada. In 1986, he received the IEEE McNaughton Gold Medal (Region 7). He was a recipient of the Canadian Telecommunications Award from Queen's University in 1992. In 1996, he was awarded the McGraw-Hill/Jacob Millman Award from the IEEE Education Society. He is also the recipient of the 2002 IEEE Signal Processing Society Education Award and the inaugural 2002 Booker Gold Medal from USRI.