

Parameter Estimation of Recurrent Neural Equalizers Using the Derivative-Free Kalman Filter

Oh-Shin Kwon, *Member, KIMICS*

Abstract— For the last decade, recurrent neural networks (RNNs) have been commonly applied to communications channel equalization. The major problems of gradient-based learning techniques, employed to train recurrent neural networks are slow convergence rates and long training sequences. In high-speed communications system, short training symbols and fast convergence speed are essentially required. In this paper, the derivative-free Kalman filter, so called the unscented Kalman filter (UKF), for training a fully connected RNN is presented in a state-space formulation of the system. The main features of the proposed recurrent neural equalizer are fast convergence speed and good performance using relatively short training symbols without the derivative computation. Through experiments of nonlinear channel equalization, the performance of the RNN with a derivative-free Kalman filter is evaluated.

Index Terms—Derivative-Free Kalman Filter, Parameter Estimation, Channel Equalization.

I. INTRODUCTION

Recurrent neural networks (RNNs) are essentially dynamical systems where the states evolve according to certain nonlinear state equations. Due to their dynamic nature, they have been successfully applied to many problems including modeling and processing of temporal signals, switch as prediction, adaptive control, system identification, and speech recognition[1]. In addition, the sequential nature of inputs and outputs in many field makes RNNs attractive for the general task of sequence prediction, sequence generation, or sequence transduction. In digital communications, channel equalization is an example of sequential data processing, and an adaptive filter used as the equalizer in the communication receiver needs on-line learning to update its free parameters. Recently, RNNs have been successfully applied to channel equalization with a variety of network structures and learning algorithms [2],[3].

Many structures for RNNs have been developed, which are ranging from fully connected to partially (or locally)

connected networks and ranging from single-layered to multi-layered networks. However, common problems to be solved are still remained: analysis of the dynamical behavior of RNNs, and contrivance of learning algorithms to cope with the complexity induced by network's dynamics. Hence intensive research works on dynamical network properties and the corresponding learning techniques have attained a good theoretical grounding of many popular algorithms. We hereby focus on learning algorithms.

Typical learning algorithms for RNNs are real-time recurrent learning (RTRL) [4] for on-line learning and back-propagation through time (BPTT) for off-line learning. These algorithms are totally based on the gradient method using first-order derivative information. The training problem is to update the free parameters of the networks. Since a weight updating affects the states at all times during the course of network evolution, obtaining the error gradient is a complicated procedure. Moreover, due to the first-order derivative information, the RTRL and BPTT many exhibit slow convergence speed relative to learning techniques based on second-order derivative information,. The extended Kalman filter (EKF) forms the basis of a second-order neural network training approach. The essence of recursive EKF procedure is that an approximate covariance matrix that encodes second-order information about the train problem is maintained and evolved during training. However, the EKF is difficult to implement, difficult to tune, and only reliable for systems that are almost linear on the time scale of the update intervals [5]. In addition, the EKF provides first-order approximations to optimal nonlinear parameter estimation and needs the computation of derivative matrix (or Jacobians) in the linearization process of the nonlinear system.

In this paper, the derivative-free Kalman filter, called the unscented Kalman filter [5],[6], is presented for training RNNs. We demonstrate the applicability of the UKF to RNN training. The performance of the UKF algorithm in nonlinear channel equalization applications is evaluated and compared with the RTRL.

Manuscript received May 11, 2010; revised May 18, 2010; accepted May 24, 2010.

Oh-Shin Kwon is with the Department of Control, Robotics and Systems Engineering, Kunsan National University, Kunsan,573-701, Korea(Email: kos@kunsan.ac.kr)

II. FULLY CONNECTED RECURRENT NEURAL NETWORK

The formulation presented here is based on the standard fully connected RNN. The fully connected RNN consists of q neurons with l external inputs, as shown in Fig. 1. Let the q -by- l vector $x(k)$ denotes the state of the network in the form of a nonlinear discrete-time system, the $(l+1)$ -by-1 vector $u(k)$ denotes the input applied the network, and the p -by-1 vector $y(k)$ denotes the output of the network. The dynamic behavior of the network, assumed to be noise free, is described by [7].

$$x(k+1) = \varphi(W_x(k)x(k) + W_u(k)u(k)) = \varphi(W(k)z(k)) \tag{1}$$

$$y(k) = Cx(k+1) \tag{2}$$

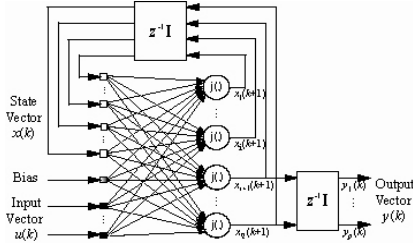


Figure 1. A layout of fully connected recurrent neural network

Where $W_x(k)$ is a q -by- p matrix, $W_u(k)$ is a q -by- $(l+1)$ matrix, C is a p -by- q matrix; and $\varphi : \square^q \rightarrow \square^q$ is a diagonal map. The two separate weight matrices can be merged into a whole weight matrix $W(k)$ with q -by- $(q+l+1)$ dimension, that is,

$$W(k) = [W_x(k)W_u(k)] \tag{3}$$

And the $(q+l+1)$ -by-1 vector $z(k)$ can be defined as

$$z(k) = \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \tag{4}$$

where $x(k)$ is the q -by-1 state vector and $u(k)$ is $(l+1)$ -by-1 input vector. The first element of $u(k)$ is unity, which is the bias input, and in a corresponding way, the first column of $W_u(k)$ is bias terms applied neurons. The dimensionality of the state space, namely q is the order of the system. Therefore the state-space model of Fig. 1 is an l -input, q -output recurrent model of order q . Eq.(1) is the process equation of the model and Eq.(2) is the measurement equation. The process equation (Eq.(1)) in the state-space description of the network is rewritten in the following form:

$$x(k+1) = \begin{bmatrix} \varphi(w_1^T(k)z(k)) \\ \varphi(w_2^T(k)z(k)) \\ \vdots \\ \varphi(w_q^T(k)z(k)) \end{bmatrix} \tag{5}$$

Where $\varphi(\cdot)$ is an activation function, and the $(q+l+1)$ -by-1 weight vector $w_i(k)$, which is connected to the i th neuron in the recurrent network, corresponds to the i th column of the transposed weight matrix $W^T(k)$.

III. A DERIVATIVE-FREE KALMAN FILTER FOR FO PARAMETER ESTIMATION

The Kalman filter is a widely used filtering strategy, over 30 years in the control and signal processing community. The extended Kalman filter(EKF), which simply linearizes all nonlinear models, would be a popular method, when the linear Kalman filter is applied to nonlinear systems. However, the EKF is difficult to implement, difficult to tune, and only reliable for systems that are almost linear on the time scale of the update intervals [5]. In parameter estimation of neural networks, the EKF 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 [8]. The unscented Kalman filter(UKF), first proposed by Julier and Uhlmann [6] and further extended by Wan and van der Merwe [8], is an alternative to the EKF algorithm. The UKF provides third-order approximation of process and measurement errors for Gaussian distributions and at least second-order approximation for non-Gaussian distributions [9]. The UKF may have more accurate estimation features over the EKF in applications [5]. 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.

The unscented transform(UT) is a method for calculating the statistics of a random variable which undergoes a nonlinear transformation [6].

Consider an L -by-1 random variable x that is nonlinearly transformed to yield a random variable y through a nonlinear function, $y=f(x)$. In order to calculate the statistics of y , a matrix χ of $2L+1$ sigma vectors χ_i is formed as the followings:

$$\begin{aligned} \chi_0 &= \bar{x} & i &= 0 \\ \chi_i &= \bar{x} + (\sqrt{(L+\lambda)P_{xx}})_i, & i &= 1, \dots, L \\ \chi_i &= \bar{x} - (\sqrt{(L+\lambda)P_{xx}})_{i-L}, & i &= L+1, \dots, 2L \end{aligned} \tag{6}$$

Where \bar{x} and covariance P_{xx} are the mean and covariance of 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{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

$$\gamma_i = f(\chi_i), \quad i = 0, \dots, 2L. \quad (7)$$

This propagation produces corresponding vector set that can be used to estimate the mean and covariance matrix of the nonlinear transformed vector y . We can approximate the mean and covariance matrix of y using a weighted sample mean and covariance of the posterior sigma points[8].

$$\bar{y} = \sum_{i=0}^{2L} W_i^m \gamma_i \quad (8)$$

$$P_{yy} = \sum_{i=0}^{2L} W_i^c (\gamma_i - \bar{y})(\gamma_i - \bar{y})^T \quad (9)$$

Where the weighting factors are given by

$$\begin{aligned} W_0^m &= \frac{\lambda}{L + \lambda} \\ W_0^c &= \frac{\lambda}{L + \lambda} + (1 - \alpha^2 + \beta) \\ W_i^m &= W_i^c = \frac{1}{2(L + \lambda)}, \quad i = 1, 2, \dots, 2L. \end{aligned} \quad (10)$$

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 x , and $\beta = 2$ is the optimal choice for Gaussian distributions.

To enable the Kalman filter for training the RNN, the network's behavior can be recast as the following nonlinear discrete-time system:

$$w(k+1) = w(k) + \omega(k) \quad (11)$$

$$y(k) = h(w(k), z(k)) + v(k) \quad (12)$$

where all vectors and matrix are complex numbers, nonlinear function $h(\cdot)$ is given by

$$h(w(k), z(k)) = C\varphi(w(k)z(k)) \quad (13)$$

And the weight vector $w(k)$ is defined by

$$w(k) = \begin{bmatrix} w_1(k) \\ w_2(k) \\ \vdots \\ w_q(k) \end{bmatrix} \quad (14)$$

Where $w_i(k) (i=1, 2, \dots, q)$ is the i th column of the transposed weight matrix $W^T(k)$. Eq.(18), known as the process equation, specifies that the state of the system is given by the network's weight parameter values $w(k)$ and is characterized as a stationary process corrupted by process noise $\omega(k)$. The measurement equation, given in Eq.(19), represents the network's output vector $y(k)$ as the nonlinear function $\varphi(\cdot)$ of the weight vector $w(k)$ and the vector $z(k)$ which include both the input vector $u(k)$ and the recurrent node activations $x(k)$. This equation is buried by random measurement noise $v(k)$. The process noise $\omega(k)$ is typically characterized as zero-mean, white noise with covariance given by $E[\omega_i \omega_j^T] = \delta_{ij} Q(k)$. Similarly, the measurement noise $v(k)$ is also characterized as zero-mean, white noise with covariance given by $E[v_i v_j^T] = \delta_{ij} R(k)$. From the state-space model of the RNN given in equations (11) and (12), the cost function to be minimized in the mean-squared error (MSE) sense is:

$$J(w) = e(k)^T R^{-1}(k) e(k) \quad (15)$$

Where the error vector $e(k)$ is defined by using the measurement equation (Eq. (2)):

$$e(k) = \hat{y}(k) - y(k) \quad (16)$$

Where $\hat{y}(k)$ denotes the desired output vector.

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

$$R(k) = \mu^{-1} I \quad (17)$$

$$Q(k) = (\lambda_{RLS}^{-1} - 1)P(k) \quad (18)$$

Where $\lambda \in (0, 1]$ is often referred to as the *forgetting factor*, in recursive least-squares (RLS) algorithms[9].

The UKF effectively evaluates the Jacobian through its sigma-point propagation, without the need to perform any analytical derivative calculation. Specific equations for the RNN using the UKF algorithm are summarized below. The weight vector in the network and the covariance matrix are initialized with

$$\hat{w}(0) = E[w] \quad (19)$$

$$P(0) = E[(w - \hat{w}(0))(w - \hat{w}(0))^T]. \quad (20)$$

The sigma-point calculation is given by

$$\Gamma(k) = (L + \lambda)(P(k) + Q(k)) \quad (21)$$

$$N(k) = [\hat{w}(k), \hat{w}(k) + \sqrt{\Gamma(k)}, \hat{w}(k) - \sqrt{\Gamma(k)}] \quad (22)$$

$$M(k) = h(N(k), z(k)) \quad (23)$$

$$y(k) = h(\hat{w}(k), z(k)). \quad (24)$$

The measurement-update equations are

$$P_{yy}(k) = \sum_{i=0}^{2L} W_i^c (M_i(k) - y(k))(M_i(k) - y(k))^T + R(k) \quad (25)$$

$$P_{wy}(k) = \sum_{i=0}^{2L} W_i^c (N_i(k) - \hat{w}(k))(N_i(k))^T \quad (26)$$

$$\rho(k) = P_{wy}(k)P_{yy}^{-1}(k) \quad (27)$$

$$\hat{w}(k+1) = \hat{w}(k) + \rho(k)e(k) \quad (28)$$

$$P(k+1) = P(k) - \rho(k)P_{yy}(k)\rho^T(k). \quad (29)$$

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

IV. NONLINEAR CHANNEL EQUATION

A. Communication System Model

A general model of a digital communications system with a decision feedback equalizer (DFE) is displayed in Fig.2. 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

$$\begin{aligned} r(k) &= g(\hat{r}(k)) + v(k) \\ &= g\left(\sum_{i=0}^{N-1} h_i s(k-i)\right) + v(k) \end{aligned} \quad (30)$$

Where $g(\cdot)$ is a nonlinear distortion, h_i 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 delay, respectively. The inputs to the DFE therefore consist of the feedforward inputs $r(k) = [r(k), r(k-1), \dots, r(k-m+1)]^T$ and feedback inputs $u(k) = [u(k-1), \dots, u(k-n)]^T$. 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[11],

$$n = N + m - d - 2 \quad (31)$$

Since the transmitted symbols contributing to decision of the equalizer at time k are given by $s(k) = [s(k), s(k-1), \dots, s(k-m-N+2)]^T$ for the feedforward order $m = d + 1$. The decision-feedback recurrent neural equalizer (DFRNE) using the fully

connected RNN is used as the DFE in the following experiments. When the RNN is used as the DFRNE, the input vector $u(k)$ includes the received signals from the channel and the decision feedback inputs, as well as the bias input.

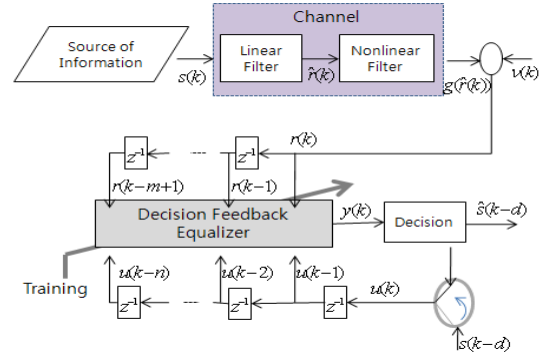


Figure 2. A communications system with decision feedback equalizer

B. Experiment 1: Convergence Rate

Channel Model 1: A linear channel model with a non-minimum phase has the transfer function:

$$H_1(z) = b_0 + b_1 z^{-1} + b_2 z^{-2}$$

Where the channel impulse response is $b = [b_0 \ b_1 \ b_2]^T$. We use $b = [0.3482 \ 0.8704 \ 0.3482]^T$, which is generally used for channel equalization in the literature[2][3]. The nonlinear channel is modeled as

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

Where a nonlinearity is applied to the output of the linear channel. This non linear distortion of the channel may take into account saturation effects due to transmission amplifiers. The learning rate of the RTRL is chosen empirically as $\eta = 0.1$ and this value ensures a stable convergence. The parameters for the UKF are chosen empirically as $\alpha = 0.1$ and $\lambda = 0.99$. The decision delay is $d = 2$.

Convergence properties of the DFRNEs used in the simulations are depicted in Fig.3, with log and linear scales of MSE values. These results are ensemble-averaged over 200 independent runs. Each run has a different BPSK random sequence and random initial weights for all DFRNEs and is performed at a SNR of 14 dB. The UKF outperforms the RTRL in terms of convergence speed. For instance, MSE value of the UKF reaches around -55 dB after 10^2 training symbols, while MSE value of the RTRL reaches -23 dB.

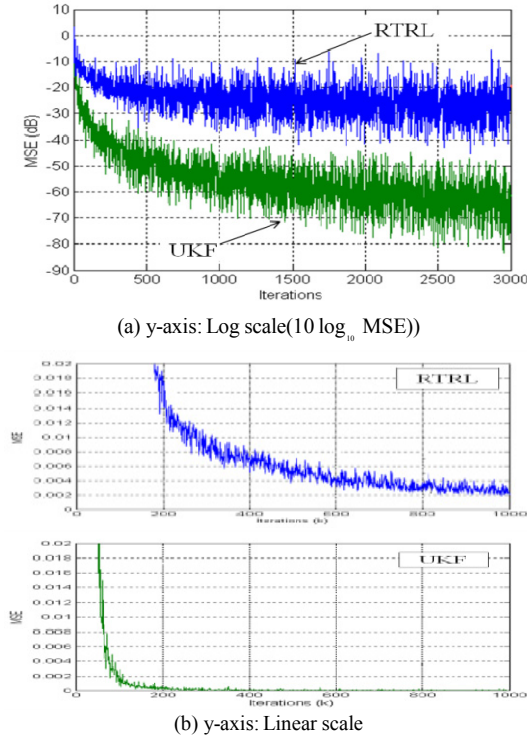


Figure 3. Convergence properties for Channel Model 1

As shown in Fig. 3(b), the UKF reaches steady state after 100 iterations. These results confirm that UKF algorithm provides an improvement with regard to both the convergence speed and the steady-state MSE. Fig.4 shows the bit-error rate (BER) performance, averaged over 100 independent trails. In each trail, the first 100 symbols are used for training and the next 10^4 symbols are used for testing.

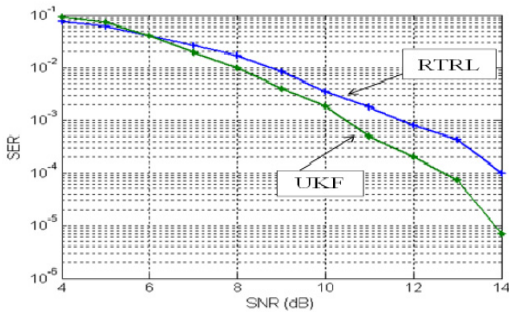


Figure 4. BER performance for Channel Model 1(100 training symbols)

The weight vectors of the DFRNEs are frozen after the training, and the transmission symbols are evaluated at the decision-directed mode. The UKF attains about 1.3 dB of improvement over the RTRL at 10^{-4} of BER. We have observed that the RTRL requires more than 200 training symbols to achieve the same BER performance of the UKF.

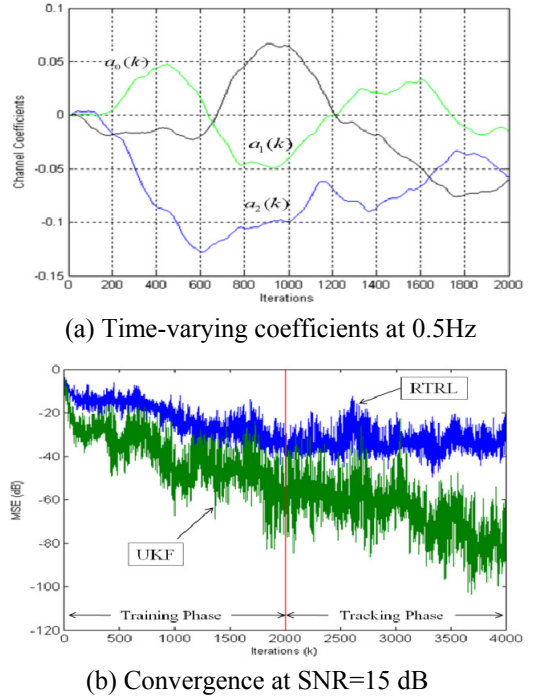


Figure 5. Channel tracking capability for Channel Model 2.

C. Experiment 2: Tracking Capability

Channel tracking performance of the DFRNEs is tested for a time-varying channel, because tracking is a *steady-state phenomenon*, in contrast with convergence which is a transient phenomenon [9]. A nonlinear channel with time-varying coefficients is considered here.

Channel Model 2: A time-varying discrete-time channel is described by

$$H_2(z) = (b_0 + a_0(k) + (b_1 + a_1(k))z^{-1} + (b_2 + a_2(k))z^{-2})$$

The nonlinear distortion employed in Channel Model 1 is applied to this channel. This channel model represents a nonlinear time-varying channel with $a_i(k) (i=0,1,2)$ varying with time k . These time-varying coefficients are generated by convolving white Gaussian noise and a Butterworth filter response. The bandwidth of the Butterworth filter determines the relative bandwidth (fading rate) of the channel. A nominal 2 kHz channel with 2400 symbol/s sampling rate are assumed, and a second-order Butterworth filter having a 3 dB bandwidth of 0.5 Hz is used [10].

The parameters are set to the same values as those used in Channel Model 1. Fig. 5(a) shows the time-varying coefficients $a_0(k), a_1(k)$ and $a_2(k)$ drawn for a fading rate of 0.5 Hz. The DFRNEs are in training phase until $k = 2000$ and then they are switched to tracking phase at $k = 2001$. Unlike simulations for Channel Model 1, the DFRNEs still update their weight vectors during testing (tracking) phase in order to track fading characteristic of the channel. In Fig. 5(b), the channel tracking property is

evaluated in both training phase and decision-directed tracking phase at a SNR of 15 dB. As expected, the UKF provides faster channel tracking capability than the corresponding RTRL. This result verifies that the MSE value of the UKF is much lower than that of the RTRL for both training and tracking phases.

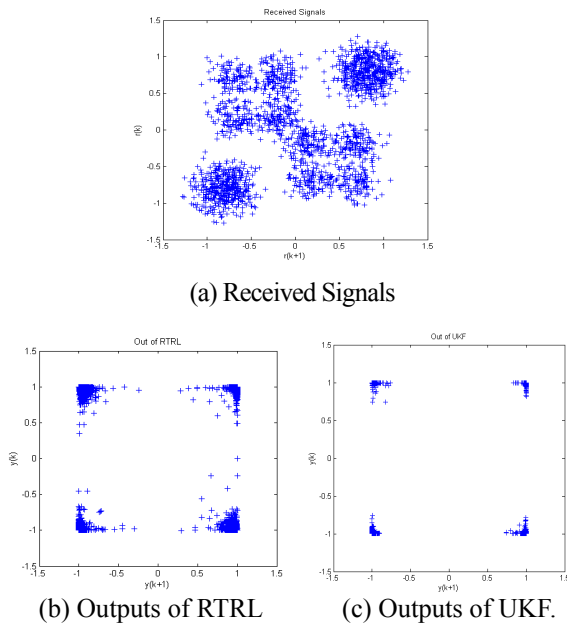


Figure 6. Eye diagrams for Channel Model 2 during tracking mode (SNR=15dB)

Fig. 6 shows eye diagrams during decision-directed tracking mode for 2×10^3 symbols. The equalized outputs of the UKF have no spots near the decision boundary. In contrast, some of the RTRL's equalized outputs are located in the decision boundary, which creates wrong a symbol detections.

ACKNOWLEDGMENT

This paper was supported by research funds of Kunsan National University in 2008.

IV. CONCLUSIONS

We have presented recurrent neural equalizers with decision feedback trained with Kalman filter for channel equalization, and their performance shows that the derivative-free Kalman filter performed better than the real-time recurrent learning (RTRL) in terms of convergence rate, BER performance, and tracking capability. In regard of convergence rate, the derivative-free Kalman filter is superior to the RTRL. This means that the fast convergence rate of the derivative-free Kalman filter (needs) requires

less training symbols and leads to better BER performance than the RTRL technique. This fast convergence rate may be suitable for high-rate channel equalization. In terms of channel tracking capability compared with the RTRL, the derivative-free Kalman filter training algorithm has shown rapid tracking mode. In short, The derivative-free Kalman filter is more suitable for high speed communication environments than RTRL.

REFERENCES

- [1] A. F. Atiya and A. G. Parlos, "New results on recurrent network training: Unifying the algorithms and accelerating convergence," *IEEE Transactions on Neural Networks*, vol.11, pp.697-709, May 2000.
- [2] 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.
- [3] J.S. Choi, M. Bouchard and T. H. Yeap, "Decision Feedback Recurrent Neural Equalization with Fast Convergence Rate," *IEEE transactions on Neural Networks*, vol. 16, No. 3, May 2005.
- [4] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Computation*, vol. 1, pp. 270-280, 1989.
- [5] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Transactions on Automatic Control*, vol. 45, pp. 477-482, March 2000.
- [6] S. J. Julier and J. K. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," in *Proceedings of Aerospace: The 11th International Symposium on Aerospace/Defence Sensing, Simulation and Controls*, 1997.
- [7] S. Haykin, *Neural Networks: a comprehensive Foundation*, 2nd Ed. Upper Saddle River, NJ: Prentice Hall, 1999.
- [8] 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.
- [9] S. Haykin, *Adaptive Filter Theory*, 4th Ed. Upper Saddle River, NJ: Prentice Hall, 2002.
- [10] C. Cowan and S. Semnani, "Time-variant equalization using a novel non-linear adaptive structure," *International Journal of Adaptive Control and Signal Processing*, vol. 12, no. 2, pp. 195-206, 1998.
- [11] O.S. Kwon, J. S. Choi, M. Bouchard and T.H. Yeap, "A Derivative-Free Kalman Filter for Parameter Estimation of Recurrent Neural Networks and Its Applications to Nonlinear Channel Equalization," 4th international ICSC Symposium on EIS 2004.



Oh-Shin Kwon received the M.S. and Ph. D. degrees in department of Electrical Engineering from Chonbuk National University, Chonju, Korea, in 1984 and 1988 respectively. During 2003-2004, he was visit professor at the Electro-Communication University, Osaka, Japan. In year 2003-2004, he also was visit professor at the University of Ottawa, Canada. From 1988 to present, He is a professor, Department of Control, Robotics and Systems Engineering, Kunsan National University in Korea. His research interests include control and neural network, intelligent system.