

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.

For more information visit www.intechopen.com



Complex Extended Kalman Filters for Training Recurrent Neural Network Channel Equalizers

Coelho Pedro H G and Biondi Neto Luiz
*State University of Rio de Janeiro (UERJ)-DETEL
Brazil*

1. Introduction

The Kalman filter was named after Rudolph E. Kalman published in 1960 his famous paper (Kalman, 1960) describing a recursive solution to the discrete-data linear filtering problem. There are several tutorial papers and books dealing with the subject for a great variety of applications in many areas from engineering to finance (Grewal & Andrews, 2001; Sorenson, 1970; Haykin, 2001; Bar-Shalom & Li, 1993). All applications involve, in some way, stochastic estimation from noisy sensor measurements. This book chapter deals with applications of Complex Valued Extended Kalman Filters for training Recurrent Neural Networks particularly RTRL (Real Time Recurrent Learning) neural networks. Gradient-based learning techniques are usually used in back-propagation and Real-Time Recurrent Learning algorithms for training feed forward Neural Networks and Recurrent Neural Network Equalizers. Known disadvantages of gradient-based methods are slow convergence rates and long training symbols necessary for suitable performance of equalizers. In order to overcome such problems Kalman filter trained neural networks has been considered in the literature. The applications are related to mobile channel equalizers using realistic channel responses based on WSSUS (Wide-Sense Stationary Uncorrelated Scattering) models. The chapter begins with a detailed description showing the application of Extended Kalman Filters to RTRL (Real Time Recurrent Learning) neural networks. The main equations are derived in a state space framework in connection to RTRL training. Then applications are envisioned for mobile channel equalizers where WSSUS models are adequate for handling equalization in presence of time-varying channels. This chapter proposes a fully recurrent neural network trained by an extended Kalman filtering including covariance matrices adjusted for better filter tuning in training the recurrent neural network equalizer. Several structures for the Extended Kalman Filter trained equalizer are described in detail, and simulation results are shown comparing the proposed equalizers with traditional equalizers and other recurrent neural networks structures. Conclusions are drawn in the end of the chapter and future work is also discussed.

2. Training a complex RTRL neural network using EKF

This chapter deals with the training of Recurrent Neural Networks that are characterized by one or more feedback loops. These feedback loops enable those neural networks to acquire

Source: Kalman Filter, Book edited by: Vedran Kordić,
ISBN 978-953-307-094-0, pp. 390, May 2010, INTECH, Croatia, downloaded from SCIYO.COM

state representations making them appropriate devices for several applications in engineering such as adaptive equalization of communication channels, speech processing and plant control. In many real time applications fast training is required in order to make the application successful. This chapter extends the EKF (Extended Kalman Filter) learning strategy considered by Haykin (Haykin, 2001) for recurrent neural networks to the one using Real Time Recurrent Learning (RTRL) training algorithm for complex valued inputs and outputs. For instance, in the adaptive channel equalization problem for modulated signals, complex envelope signals are used, so a complex RTRL recurrent neural network could be useful in such equalization application. Rao, (Rao et. al., 2000) used EKF techniques for training a complex backpropagation neural network for adaptive equalization. The complex RTRL neural network training was also considered by Kechriotis and Manolakos (Kechriotis & Manolakos, 1994) and their training algorithm is also revisited in section 3 of this chapter with the use of a state space representation. Results indicate the feasibility of the proposed complex EKF trained RTRL neural network for tracking slow time varying signals but also shows the proposed structure does not suit scenarios where fast time varying signals are concerned. So, better time tracking mechanisms are needed in the proposed neural network structure. The authors are currently pursuing enhanced mechanisms in the complex RTRL neural network so to incorporate more information in the RTRL neural network in order improve fast time tracking. Next sections show details on how the EKF training is performed for a complex RTRL neural network. First the structure of a recurrent neural network is described then how is usually trained.

3. Recurrent neural networks

The structure of the neural network considered in this chapter is that of a fully connected recurrent network as depicted in figure 1. The usual training algorithm for that neural network is known as RTRL and was derived by Williams and Zipser (Williams & Zipser, 1989). For complex valued signals the corresponding training algorithm is called Complex EKF-RTRL or EKF-CRTRL in this chapter. Usually CRTRL training algorithms use gradient techniques for updating the weights such as the training scheme proposed by and Kechriotis and Manolakos (Kechriotis & Manolakos, 1994). Their training algorithm can be rewritten in terms of a state space representation extending Haykin's analysis (Haykin, 1999) for complex signals. So, in the noise free case, the dynamic behavior of the recurrent neural network in figure 1 can be described by the nonlinear equations.

$$\begin{aligned} \underline{x}(n+1) &= \varphi^C(W_a \underline{x}(n) + W_b \underline{u}(n)) = \\ &= \varphi(\text{real}(W_a \underline{x}(n) + W_b \underline{u}(n)) + i \varphi(\text{imag}(W_a \underline{x}(n) + W_b \underline{u}(n))) = \\ &= \underline{x}^R(n+1) + i \underline{x}^I(n+1) = \end{aligned} \quad (1)$$

$$\underline{y}(n) = C \underline{x}(n) \quad (2)$$

where W_a is a q -by- q matrix, W_b is a q -by- m matrix, C is a p -by- q matrix and $\varphi: \mathbb{R}^q \rightarrow \mathbb{R}^q$ is a diagonal map described by

$$\Phi: \begin{pmatrix} x_1^{\text{R or I}} \\ x_2^{\text{R or I}} \\ \vdots \\ x_q^{\text{R or I}} \end{pmatrix} \rightarrow \begin{pmatrix} \varphi(x_1^{\text{R or I}}) \\ \varphi(x_2^{\text{R or I}}) \\ \vdots \\ \varphi(x_q^{\text{R or I}}) \end{pmatrix} \quad (3)$$

for some memoryless component-wise nonlinearity $\varphi^c: \mathbb{C} \rightarrow \mathbb{C}$. The spaces \mathbb{C}^m , \mathbb{C}^q , and \mathbb{C}^p are named the input space, state space, and output space, respectively. It can be said that q , that represents the dimensionality of the state space, is the order of the system. So the state space model of the neural network depicted in figure 1 is an m -input, p -output recurrent model of order q . Equation (1) is the process equation and equation (2) is the measurement equation. Moreover, \mathbf{W}_a contains the synaptic weights of the q processing neurons that are connected to the feedback nodes in the input layer. Besides, \mathbf{W}_b contains the synaptic weights of each one of the q neurons that are connected to the input neurons, and matrix \mathbf{C} defines the combination of neurons that will characterize the output. The nonlinear function $\varphi^c(\cdot)$ represents the sigmoid activation function of each one of the q neurons supposed to have the form

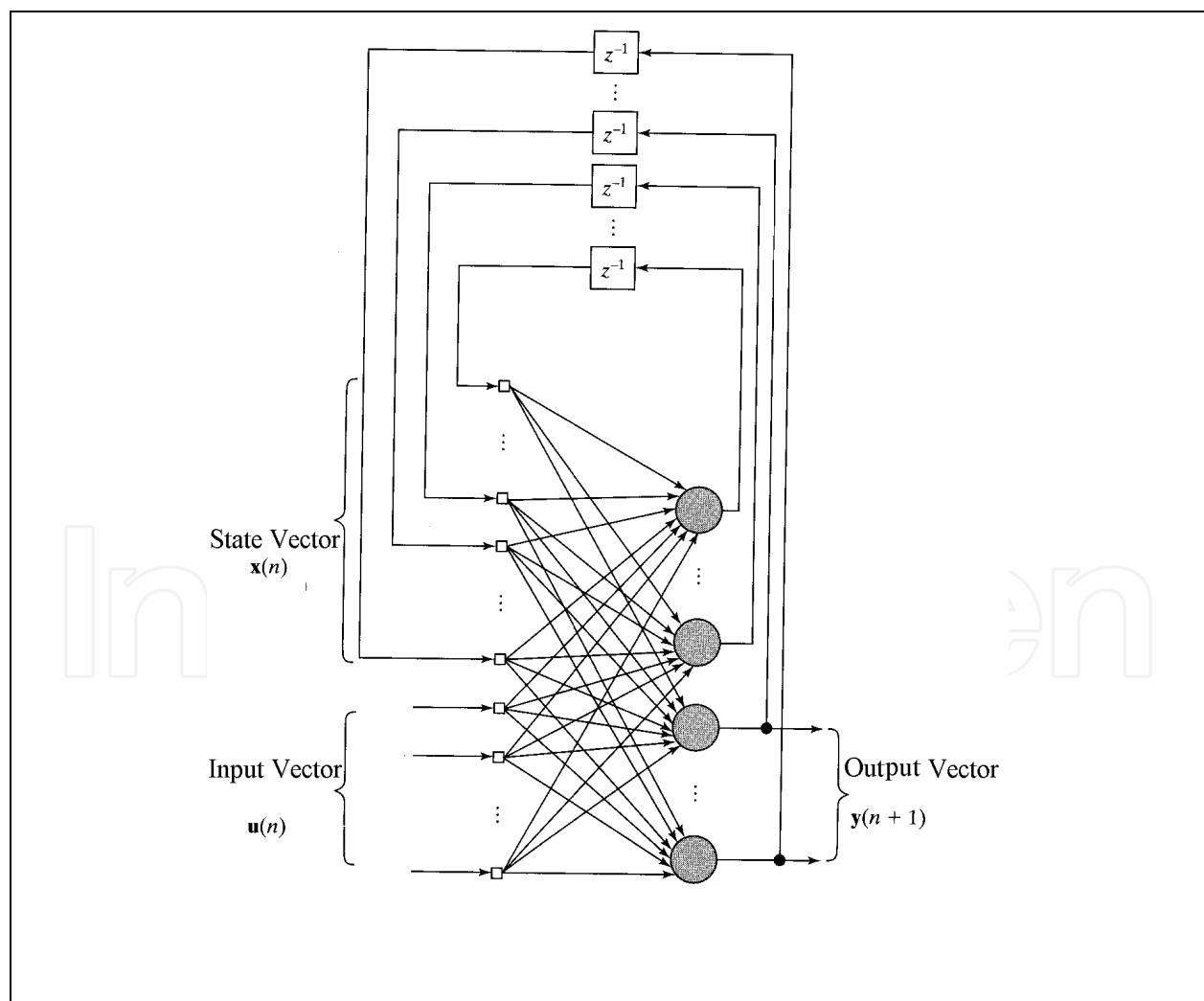


Fig. 1. Recurrent Neural Network Structure

$$\varphi^C = \varphi(\operatorname{real}(x)) + i\varphi(\operatorname{imag}(x)) \quad (4)$$

where

$$\varphi(x) = \tanh(x) = (1 - e^{-2x}) / (1 + e^{-2x}) \quad (5)$$

It should be noted that the function φ^C defined by equation (4) is scalar and is obviously different from the vector function $\underline{\varphi}^C$ defined by equation (1).

4. CTRL learning representation using a state space model

This section derives the CTRL learning algorithm in terms of a state space model presented in section 3. The process equation (1) can be written in an expanded form as

$$\begin{aligned} \underline{x}(n+1) &= [\varphi^C(\underline{w}_1^H \underline{\xi}(n)) \dots \varphi^C(\underline{w}_q^H \underline{\xi}(n))]^T = \\ &= [\varphi(\operatorname{real}(\underline{w}_1^H \underline{\xi}(n))) \dots \varphi(\operatorname{real}(\underline{w}_q^H \underline{\xi}(n)))]^T \\ &\quad + i[\varphi(\operatorname{imag}(\underline{w}_1^H \underline{\xi}(n))) \dots \varphi(\operatorname{imag}(\underline{w}_q^H \underline{\xi}(n)))]^T \end{aligned} \quad (6)$$

where it is supposed that all q neurons have the same activation function given by (4) and H is the Hermitian operator. The $(q+m)$ -by-1 vector \underline{w}_j is defined as the synaptic weight vector of neuron j in the recurrent neural network, so that

$$\underline{w}_j = \begin{pmatrix} \underline{w}_{a,j} \\ \underline{w}_{b,j} \end{pmatrix}, \quad j = 1, 2, \dots, q \quad (7)$$

where $\underline{w}_{a,j}$ e $\underline{w}_{b,j}$ are the j th columns of the transposed weight matrices \mathbf{W}_a^T e \mathbf{W}_b^T respectively. The $(q+m)$ -by-1 vector $\underline{\xi}(n)$ is defined by

$$\underline{\xi}(n) = \begin{pmatrix} \underline{x}(n) \\ \underline{u}(n) \end{pmatrix} \quad (8)$$

where $\underline{x}(n)$ is the q -by-1 state vector and $\underline{u}(n)$ is the m -by-1 input vector.

Before deriving the CTRL learning algorithm some new matrices are defined, where the indexes A and B indicate real or imaginary parts.

$$\Lambda_j^{AB}(n) = \frac{\partial \underline{x}^A(n)}{\partial \underline{w}_j^B} \begin{pmatrix} \frac{\partial x_1^A}{\partial w_{j1}^B} & \frac{\partial x_1^A}{\partial w_{j2}^B} & \dots & \frac{\partial x_1^A}{\partial w_{jq+m}^B} \\ \frac{\partial x_2^A}{\partial w_{j1}^B} & \frac{\partial x_2^A}{\partial w_{j2}^B} & \dots & \frac{\partial x_2^A}{\partial w_{jq+m}^B} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial x_q^A}{\partial w_{j1}^B} & \frac{\partial x_q^A}{\partial w_{j2}^B} & \dots & \frac{\partial x_q^A}{\partial w_{jq+m}^B} \end{pmatrix} \quad (9)$$

$$U_j A = \begin{vmatrix} 0^T \\ \xi^A T(n) \\ 0^T \end{vmatrix} \leftarrow j\text{-th row} \quad (10)$$

$$\xi^A = \begin{vmatrix} \underline{x}^A \\ \underline{u}^A \end{vmatrix} \quad (11)$$

$$\phi_R(n) = \text{diag} [\varphi'(\text{real}(\underline{w}_1^H \xi(n))) \dots \varphi'(\text{real}(\underline{w}_q^H \xi(n)))]$$

$$\phi_I(n) = \text{diag}[\varphi'(\text{imag}(\underline{w}_1^H \xi(n))) \dots \varphi'(\text{imag}(\underline{w}_q^H \xi(n)))] \quad (12)$$

Updating equations for the matrices $\Lambda_j^{A B}(n)$ is needed for the CRTRL training algorithm. There are four such matrices and they all can be obtained using their formal definitions. For instance:

$$\Lambda_j^{RR}(n) = \frac{\partial \underline{x}^R(n)}{\partial \underline{w}_j^R} \quad (13)$$

and so

$$\Lambda_j^{RR}(n) = \frac{\partial \varphi(\text{real}(W_a \underline{x}(n) + W_b \underline{u}(n)))}{\partial \underline{w}_j^R} = \frac{\partial \varphi(\underline{s}^R)}{\partial \underline{w}_j^R} = \frac{\partial \varphi(\underline{s}^R)}{\partial \underline{s}^R} \frac{\partial \underline{s}^R}{\partial \underline{w}_j^R} \quad (14)$$

However,

$$\frac{\partial \varphi(\underline{s}^R)}{\partial \underline{s}^R} = \text{diag} [\varphi'(\underline{s}_1^R(n)) \dots \varphi'(\text{real}(\underline{s}_q^R(n)))] = \phi_R(n) \quad (15)$$

and

$$\frac{\partial \varphi(\underline{s}^R)}{\partial \underline{w}_j^R} = W_a^R \Lambda_j^{RR}(n) - W_a^I \Lambda_j^{IR}(n) + U_j^R(n) \quad (16)$$

where

$$U_j^R = \begin{bmatrix} \underline{0}^T \\ \xi^{RT}(n) \end{bmatrix}, \quad \xi^{RT}(n) = [\underline{x}^{RT} \quad \underline{u}^{RT}] \quad (17)$$

So

$$\Lambda_j^{RR}(n) = \phi_R(n) [W_a^R \Lambda_j^{RR}(n) - W_a^I \Lambda_j^{IR}(n) + U_j^R(n)] \quad (18)$$

The other ones can be obtained in a similar way. The four matrices can be written in a compact form as

$$\begin{bmatrix} \Lambda_j^{RR} & \Lambda_j^{RI} \\ \Lambda_j^{IR} & \Lambda_j^{II} \end{bmatrix} (n+1) \begin{bmatrix} \phi_R & \underline{0} \\ \underline{0} & \phi_I \end{bmatrix} (n+1) \begin{bmatrix} W_a^R & W_a^I \\ W_a^I & W_a^R \end{bmatrix} \begin{bmatrix} \Lambda_j^{RR} & \Lambda_j^{RI} \\ \Lambda_j^{IR} & \Lambda_j^{II} \end{bmatrix} (n) + \begin{bmatrix} U_j^R & -U_j^I \\ U_j^I & U_j^R \end{bmatrix} \quad (19)$$

The weights updating equations are obtained by minimizing the error

$$\varepsilon(n) = \frac{1}{2} e^H(n) e(n) = \frac{1}{2} [\underline{e}_R^T(n) \underline{e}_R(n) + \underline{e}_I^T(n) \underline{e}_I(n)] \quad (20)$$

The error gradient is defined as

$$\nabla_{\underline{w}_j} \varepsilon(n) = \frac{\partial \varepsilon(n)}{\partial \underline{w}_j^R} + i \frac{\partial \varepsilon(n)}{\partial \underline{w}_j^I} \quad (21)$$

where

$$\frac{\partial \varepsilon(n)}{\partial \underline{w}_j^R} = -\Lambda_j^{RR}(n)^T C^T \underline{e}_R(n) - \Lambda_j^{IR}(n)^T C^T \underline{e}_I(n) \quad (22)$$

and

$$\frac{\partial \varepsilon(n)}{\partial \underline{w}_j^I} = -\Lambda_j^{RI}(n)^T C^T \underline{e}_R(n) - \Lambda_j^{II}(n)^T C^T \underline{e}_I(n) \quad (23)$$

The weights updating equation uses the error gradient and is written as

$$\underline{w}_j(n+1) = \underline{w}_j(n) - \eta \nabla_{\underline{w}_j} \varepsilon(n) \quad (24)$$

So the weights adjusting equations can be written as

$$\Delta \underline{w}_j(n) = \Delta \underline{w}_j^R(n) + i \Delta \underline{w}_j^I(n) = \eta \{ [\underline{e}_R^T(n) C \underline{e}_R^T(n) C] \begin{bmatrix} \Lambda_j^{RR} & \Lambda_j^{RI} \\ \Lambda_j^{IR} & \Lambda_j^{II} \end{bmatrix} (n) \} \begin{bmatrix} 1 \\ i \end{bmatrix} \quad (25)$$

The above training algorithm uses gradient estimates and convergence is known to be slow (Haykin, 2001) This motivates the use of faster training algorithms such as the one using EKF techniques which can be found in (Haykin, 2001) for real valued signals. Next section shows the application of EKF techniques in the CRTRL training.

5. EKF-CRTRL learning

This section derives the EKF-CRTRL learning algorithm. For that, the supervised training of the fully recurrent neural network in figure 1 can be viewed as an optimal filtering problem, the solution of which, recursively utilizes information contained in the trained data in a manner going back to the first iteration of the learning process. This is the essence of Kalman filtering (Kalman, 1960). The state-space equations for the network may be modeled as

$$\begin{aligned} \underline{w}_j(n+1) &= \underline{w}_j(n) + \underline{\omega}_j(n) \quad j=1, \dots, q \\ \underline{x}(n) &= \underline{\varphi}^C (W_a \underline{x}(n-1) + W_b \underline{u}(n-1)) + \underline{v}(n) = \begin{bmatrix} \varphi^C(\underline{w}_1^H \underline{\xi}(n-1)) \\ \vdots \\ \varphi^C(\underline{w}_j^H \underline{\xi}(n-1)) \\ \vdots \\ \varphi^C(\underline{w}_q^H \underline{\xi}(n-1)) \end{bmatrix} + \underline{v}(n) \end{aligned} \quad (26)$$

where $\underline{\omega}_j(n)$ is the process noise vector, $\underline{v}(n)$ is the measurement noise vector, both considered to be white and zero mean having diagonal covariance matrices Q and R respectively and now the weight vectors \underline{w}_j ($j=1, q$) play the role of state. It is also supposed that all q neurons have the same activation function given by (4). It is important to stress that when applying the extended Kalman filter to a fully recurrent neural network one can see two different contexts where the term state is used (Haykin, 1999). First, in the evolution of the system through adaptive filtering which appears in the changes to the recurrent network's weights by the training process. That is taken care by the vectors \underline{w}_j ($j=1, q$). Second, in the operation of the recurrent network itself that can be observed by the recurrent nodes activities. That is taken care by the vector $\underline{x}(n)$. In order to pave the way for the application of Kalman filtering to the state-space model given by equations (6), it is necessary to linearize the second equation in (6) and rewrite it in the form

$$\begin{aligned} \underline{x}(n) &= [\Lambda_1(n-1) \dots \Lambda_j(n-1) \dots \Lambda_q(n-1)] \begin{bmatrix} \underline{w}_1 \\ \vdots \\ \underline{w}_j \\ \vdots \\ \underline{w}_q \end{bmatrix} \\ &= \sum_{j=1}^q \Lambda_j(n-1) \underline{w}_j + \underline{v}(n) \end{aligned} \quad (27)$$

The synaptic weights were divided in q groups for the application of the decoupled extended Kalman filter (DEKF) (Haykin, 1999). The framework is now set for the application of the Kalman filtering algorithm (Haykin, 1999) which is summarized in Table 1. Equations in Table 1 are extensions from real to complex values. The expression involving Λ_j can be evaluated through the definition

$$\Lambda_j(n) = \frac{\partial \underline{x}(n)}{\partial \underline{w}_j^R} - i \frac{\partial \underline{x}(n)}{\partial \underline{w}_j^I} \quad (28)$$

The training procedure was improved in the EKF training by the use of heuristic fine-tuning techniques for the Kalman filtering. The tuning incorporated in the filter algorithm in Table

1 is based on the following. It is known that initial values of both the observation and the process noise covariance matrices affect the filter transient duration. These covariances not only account for actual noises and disturbances in the physical system, but also are a means of declaring how suitably the assumed model represents the real world system (Maybeck, 1979). Increasing process noise covariance would indicate either stronger noises driving the dynamics or increased uncertainty in the adequacy of the model itself to depict the true dynamics accurately. In a similar way, increased observation noise would indicate the measurements are subjected to a stronger corruptive noise, and so should be weighted less by the filter. That analysis indicate that a large degree of uncertainty is expected in the initial phase of the Kalman filter trained neural network so that it seems reasonable to have large initial covariances for the process and observation noises. Therefore the authors suggest a heuristic mechanism, to be included in the extended Kalman filter training for the recurrent neural network, that keeps those covariances large in the beginning of the training and then decreases during filter operation. In order to achieve that behavior, a diagonal matrix is added both to the process and to the observation noise covariance matrices individually. Each diagonal matrix is composed by an identity matrix times a complex valued parameter which decreases at each step exponentially. The initial value of this parameter is set by means of simulation trials. Simulation results indicated the success of such heuristic method.

<p>Initialization:</p> <ol style="list-style-type: none"> 1. Set the synaptic weights of the recurrent network to small values selected from a complex uniform distribution. 2. Set $K_j(0) = (\delta_R + i \delta_I) I$ where δ_R e δ_I are small positive constants. 3. Set $R(0) = (\gamma_R + i \gamma_I) I$ where γ_R and γ_I are large positive constants, typically $10^2 - 10^3$. <p>Heuristic Filter Tuning:</p> <ol style="list-style-type: none"> 1. Set $R(n) = R(n) + \alpha I$, where α decreases exponentially in time. 2. Set $Q(n) = Q(n) + \beta I$, where β decreases exponentially in time.
<p>Compute for $n = 1, 2, \dots$</p> $\Gamma(n) = [\sum_{j=1}^q \Lambda_j(n) K_j(n-1) \Lambda_j^H(n) + R(n)]^{-1}$ $G_j(n) = K_j(n) \Lambda_j^H(n) \Gamma(n)$ $\underline{\alpha}(n) = \underline{d}(n) - C \sum_{j=1}^q \Lambda_j(n-1) \underline{w}_j$ $\underline{w}_j(n+1) = \underline{w}_j(n) + G_j(n) \underline{\alpha}(n)$ $K_j(n+1) = K_j(n) - G_j(n) \Lambda_j(n) K_j(n) + Q_j(n)$ $\Lambda_j(n) = \Lambda_j^{RR}(n) + \Lambda_j^{II}(n) + i(\Lambda_j^{IR}(n) - \Lambda_j^{RI}(n))$ <p>$\underline{d}(n)$ is the desired output at instant n</p>

Table 1. Recurrent Neural Network Training Via Decoupled Extended Kalman Filter DEKF Algorithm Complex (Decoupled Extended Kalman Filter)

The authors applied the EKF-CRTRL training in channel equalization problems for mobile and cell communications scenarios and representative papers are (Coelho & Biondi, 2006), (Coelho, 2002) and, (Coelho & Biondi, 2006).

6. Results and conclusion

Numerical results were obtained for the EKF-CRTRL neural network derived in the previous section using the adaptive complex channel equalization application. The objective of such an equalizer is to reconstruct the transmitted sequence using the noisy measurements of the output of the channel (Proakis, 1989).

A WSS-US (Wide Sense Stationary-Uncorrelated Scattering) channel model was used which is suitable for modeling mobile channels (Hoeher, 1992). It was assumed a 3-ray multipath intensity profile with variances (0.5, 0.3, 0.2). The scattering function of the simulated channel is typically that depicted in figure 2. This function assumes that the Doppler spectrum has the shape of the Jakes spectrum (Jakes, 1969). The input sequence was considered complex, QPSK whose real and imaginary parts assumed the values +1 and -1. The SNR was 40 dB and the EKF-CRTRL equalizer had 15 input neurons and 1 processing neuron. It was used a Doppler frequency of zero. The inputs comprised the current and previous 14 channel noisy measurements. Figure 3 shows the square error in the output vs. number of iterations.

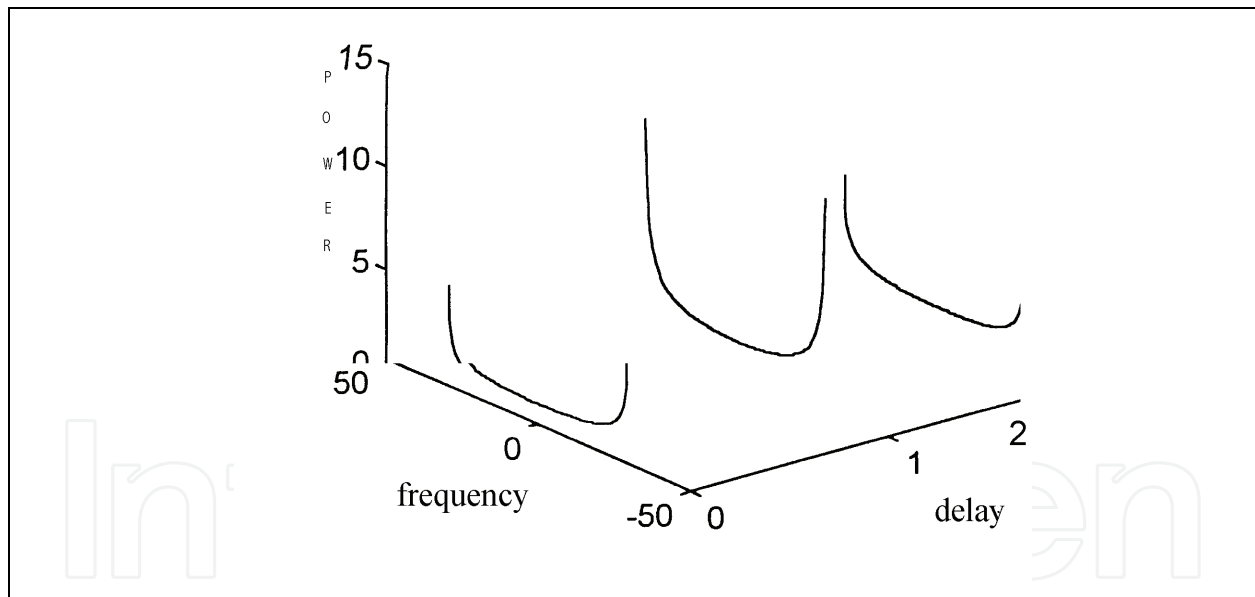


Fig. 2. Scattering Function of the Simulated Mobile Channel

Figure 3 shows a situation where the mobile receiving the signal is static, e.g. Doppler frequency zero Hz. Figure 4 shows a scenario where the mobile is moving slowly, e.g. Doppler frequency 10 Hz. To assess the benefits in the EKF-CRTRL training algorithm one can compare the square error in its output with that in the output of the CRTRL algorithm that uses gradient techniques as described in section 3. Figure 5 shows the square error in the output of the CRTRL equalizer for a 0 Hz Doppler frequency. It can be noted that convergence is slower than the EKF-CRTRL algorithm and that was obtained consistently with all simulations performed.

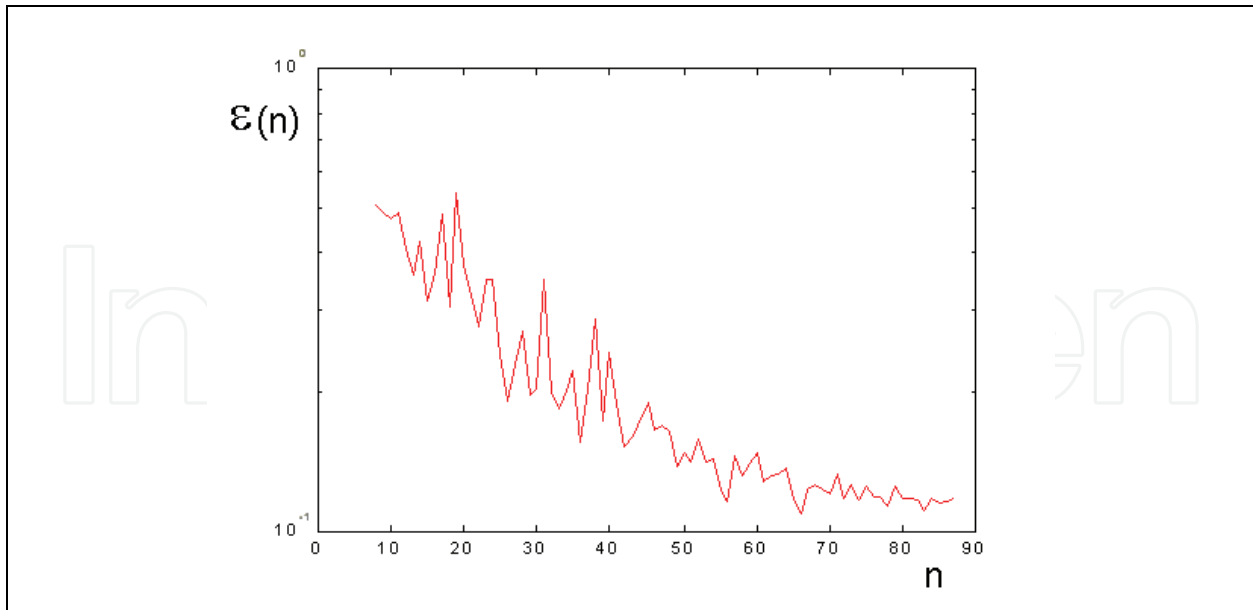


Fig. 3. Square error in the output of the EKF-CRTRL Equalizer ($m=15, q=1$, SNR=40 dB, 6 symbol delay and Doppler Frequency Zero Hz) vs. number of iterations

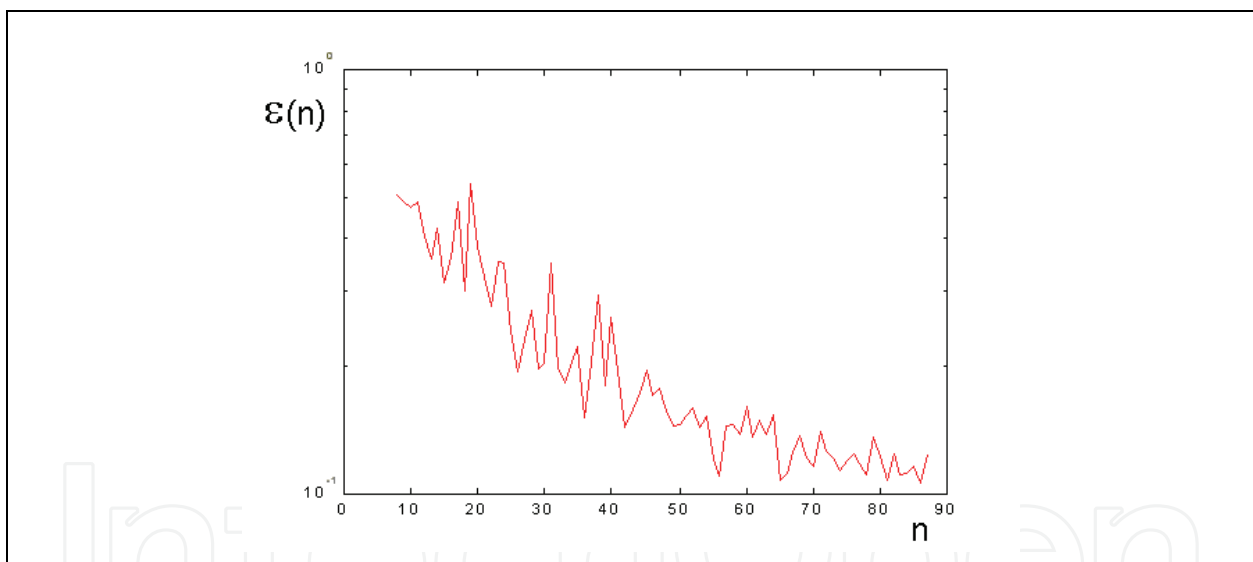


Fig. 4. Square error in the output of the EKF-CRTRL Equalizer ($m=15, q=1$, SNR=40 dB, 6 symbol delay and Doppler Frequency 10 Hz) vs. number of iterations

The results achieved with the EKF-CRTRL equalizer were superior to those of (Kechriotis et al, 1994). Their derivation of CRTRL uses gradient techniques for training the recurrent neural network as the revisited CRTRL algorithm described in section 3 of this chapter. Faster training techniques are useful particularly in mobile channel applications where the number of training symbols should be small, typically about 30 or 40. The EKF-CRTRL training algorithm led to a faster training for fully recurrent neural networks. The EKF-CRTRL would be useful in all real time engineering applications where fast convergence is needed.

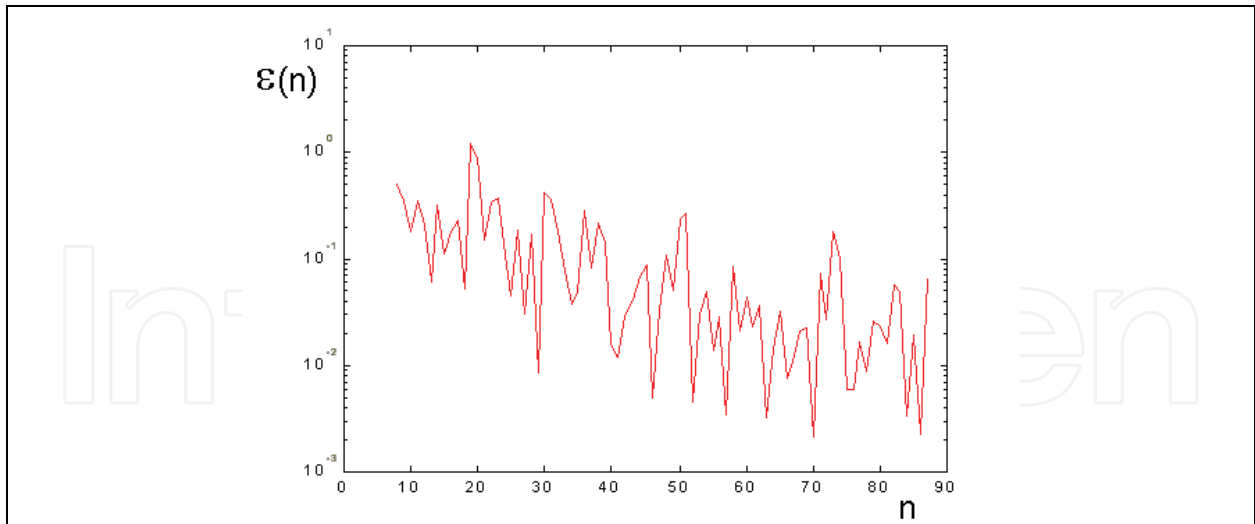


Fig. 5. Square error in the output of the CRTRL Equalizer($m=12,q=1$, SNR=40 dB, 6 symbol delay and Doppler Frequency Zero Hz) vs. number of iterations

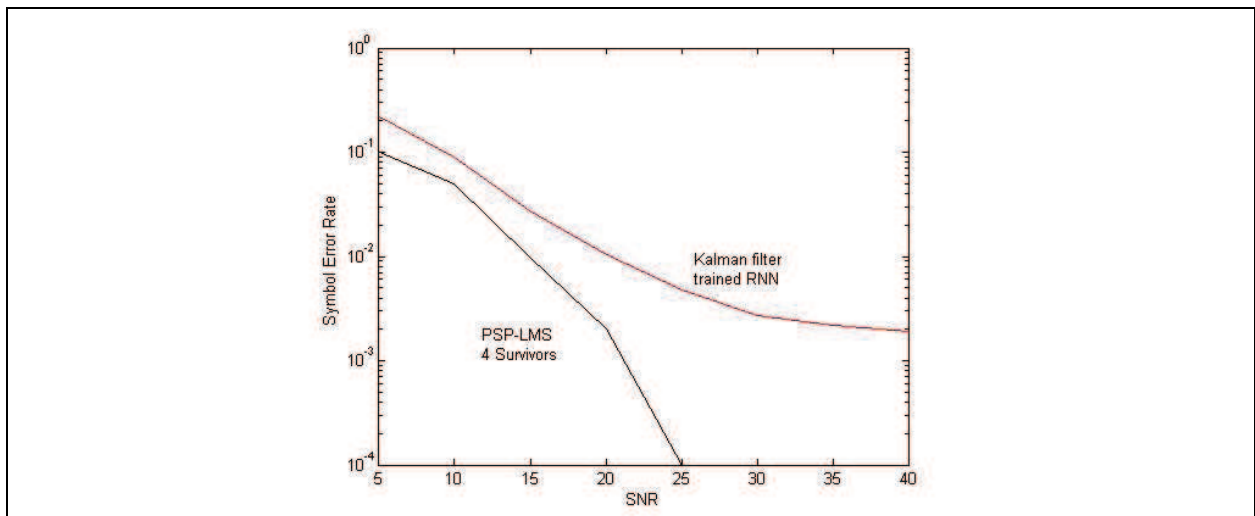


Fig. 6. Performance of the EKF-CRTRL equalizer and the PSP-LMS equalizer for $f_D=0$ Hz

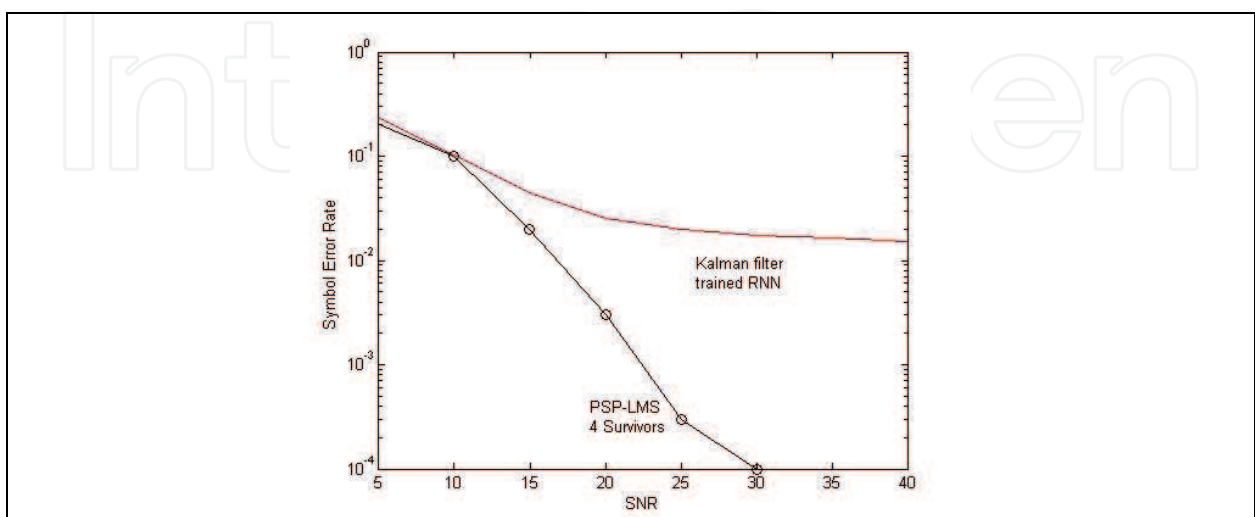


Fig. 7. Performance of the EKF-CRTRL equalizer and the PSP-LMS equalizer for $f_D=10$ Hz

However, the proposed equalizer is outperformed by the class of equalizers known in the literature as (PSP (Per Surviving Processing) equalizers which are a great deal more computational complex than the proposed equalizer. Figure 6 and 7 show comparisons involving the recurrent neural network equalizer regarding symbol error rate performances where one can see the superiority of the PSP-LMS equalizer. Details of such class of equalizers can be found in (Galdino & Pinto, 1998) and are not included here because is beyond the scope of the present chapter. In order to assess the EKF-CRTRL equalizer performance compared with traditional equalizers figure 8 shows symbol error rates for the equalizer presented in this chapter and the traditional Decision feedback equalizer (Proakis,

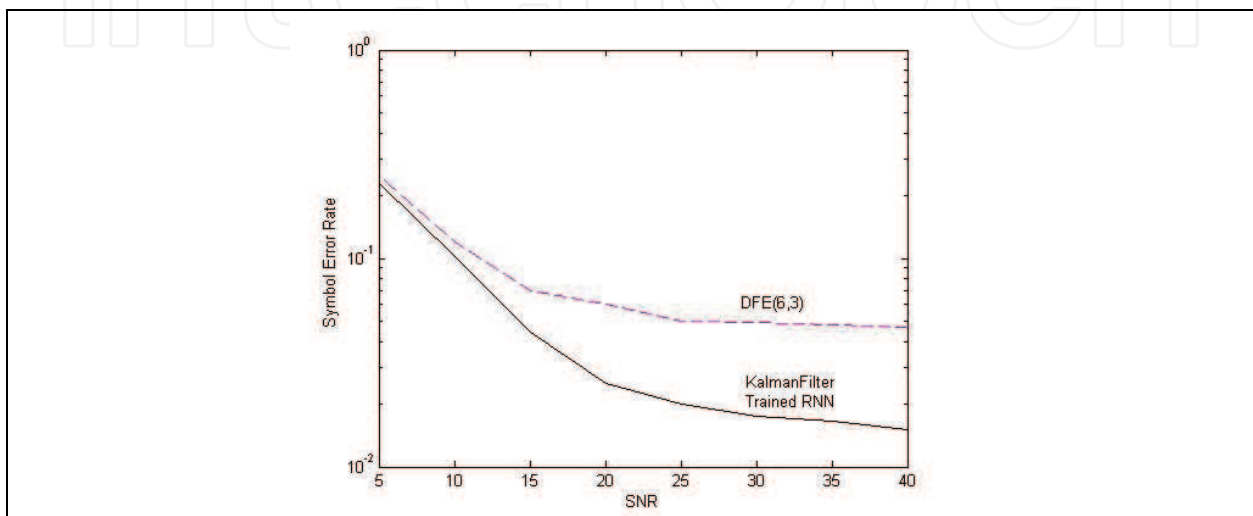


Fig. 8. Symbol Error Rate (SER) \times SNR for $f_D = 10$

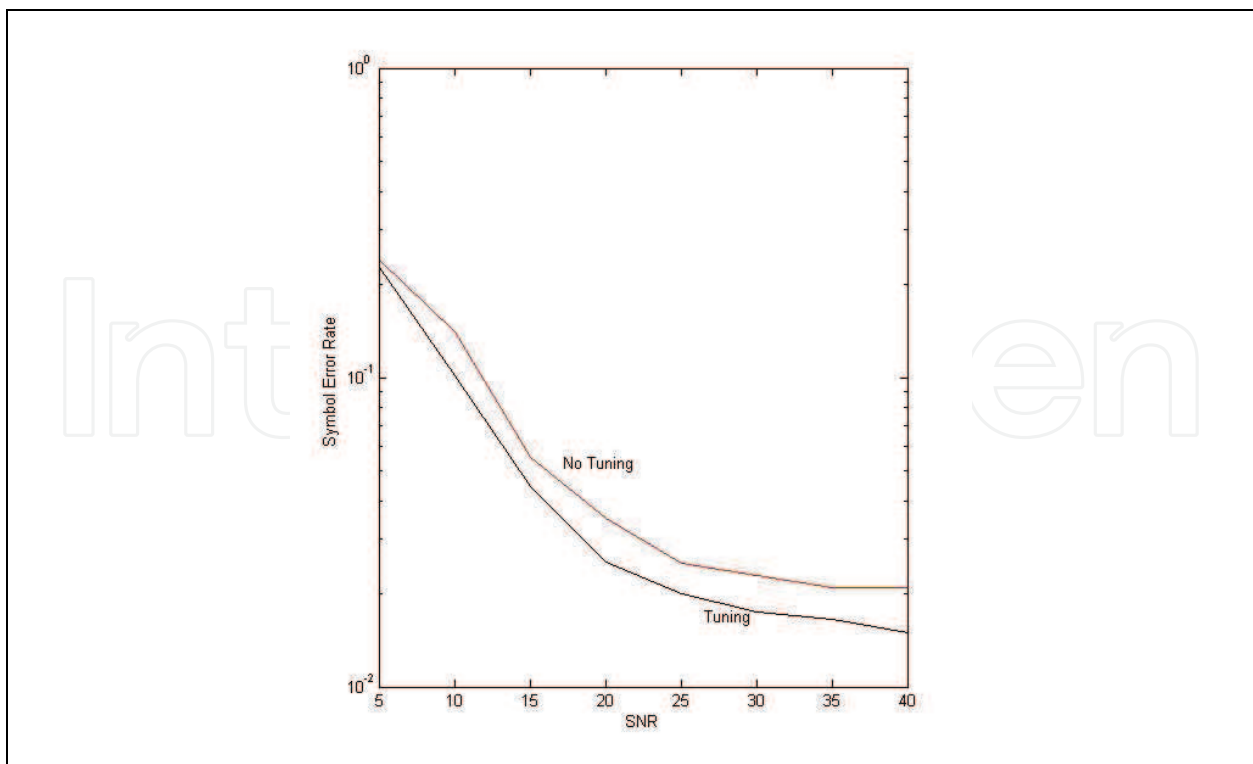


Fig. 9. Performance of the Kalman filter trained equalizer with Tuning for $f_D = 0$

1989). One can see the superiority of the EKF-CRTRL equalizer in the figure. Such results suggest for future work to include in the recurrent network a mechanism to enhance temporal tracking for fast time varying scenarios in high mobility speeds, typically above 50 km/h, e.g. Doppler frequencies above 40 Hz. The authors are currently working on that. Finally, figure 9 show the efficiency of the heuristic tuning mechanism proposed in this chapter in connection with the EKF-CRTRL equalizer. One can see the superior results in terms of error rate for equalizers with such tuning algorithm.

7. References

- Grewal, M., S., & Andrews, A., P. (2001). *Kalman Filtering Theory and Practice Using MATLAB (Second ed.)*, John Wiley & Sons, Inc., ISBN 0-471-39254-5, New York, NY, USA
- Sorenson, H. W. (1970). Least-Squares estimation: from Gauss to Kalman. *IEEE Spectrum*, (July 2001), pages 63-68, ISSN 0018-9235
- Haykin, S., Ed. (2001). *Kalman Filtering and Neural Networks*, John Wiley & Sons, Inc., ISBN 0-471-36998-5, New York, NY, USA
- Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Transaction of the ASME – Journal of Basic Engineering*, Vol . 82 (Series D), pages 35-45
- Grewal, M., S., & Andrews, A., P. (2001). *Kalman Filtering Theory and Practice Using MATLAB (2nd Edition)*, John Wiley & Sons, Inc., ISBN 0-471-39254-5, New York, NY, USA
- Sorenson, H. W. (1970). Least-Squares estimation: from Gauss to Kalman. *IEEE Spectrum*, (July 2001), pages 63-68, ISSN 0018-9235
- Haykin, S., Ed. (2001). *Kalman Filtering and Neural Networks*, John Wiley & Sons, Inc., ISBN 0-471-36998-5, New York, NY, USA
- Bar-Shalom, Y., & Li, X.-R. (1993). *Estimation and Tracking: Principles, Techniques, and Software*, Artech House, Inc.
- Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation (2nd Edition)*, Prentice Hall, ISBN-13: 978-0132733502, USA
- Kechriotis, G., and Manolakos, E. S.(1994). Training Fully Recurrent Neural Networks with Complex Weights, *IEEE Trans. Circuits Syst. II*, (March 1994), pages 235-238
- Rao, D. K., Swamy, M. N. S. and Plotkin, E. I. (2000). Complex EKF Neural Network for Adaptive Equalization. *ISCAS 2000, Proceedings of the International Symposium on Circuits and Systems*, pp. II-349- II-352, Geneva May 28-31, 2000, Switzerland
- Williams, R. J. and Zipper. D. (1989). A Learning Algorithm for Continually Running Fully Recurrent Neural Networks, *Neural Computation*, Vol.1, 1989, pages 270-280
- Maybeck, P. S.(1979). *Stochastic Models, Estimation, and Control*, Vol. 1, Academic Press, ISBN-0-12-480701-1, NY, USA
- Coelho, P.H.G. & Biondi Neto L. (2006). Complex Kalman Filter Trained Recurrent Neural Network Based Equalizer for Mobile Channels, *Proceedings of the 2006 International Joint Conference on Neural Networks*, pp. 2349-2353, Sheraton Vancouver Wall Centre Hotel, July 16-21, 2006, Vancouver, BC, Canada
- Coelho, P. H. G. and Biondi Neto L., (2005). Further Results on The EKF-CRTRL Equalizer for Fast Fading and Frequency Selective Channels, *Proceedings of the 2005 International Joint Conference on Neural Networks*, pp. 2367-2371, July 31-August 4 2005, Montreal, Quebec, Canada

- Coelho, P.H.G. (2002). Adaptive Channel Equalization Using EKF-CRTL Neural Networks, *Proceedings of the 2002 International Joint Conference on Neural Networks*, pp. 1195-1199, , May 12-17, 2002, Honolulu, Hawaii, USA
- Coelho, P. H. G. & Biondi Neto L. (2006). Complex Kalman Filter Trained Recurrent Neural Network Based Equalizer for Mobile Channels, *Proceedings of the 2006 International Joint Conference on Neural Networks*, pp. 2349-2353, Sheraton Vancouver Wall Centre Hotel, July 16-21, 2006, Vancouver, BC, Canada
- Proakis, J. G. (1989). *Digital Communications (2nd Edition)*, McGraw-Hill, ISBN 0-07-100269-3, USA
- Hoehner, P. (1992). Statistical Discrete- Time Model for the WSSUS Multipath Channel, *IEEE Transactions on Vehicular Technology*, Nov. 1992, Vol 41. Number 4, pp. 461-468
- Jakes, W. C. Jr. (1969). *Microwave Mobile Communications*, Wiley, USA
- Kechriotis, G., Zervas, E. and Manolakos, E. S. (1994). Using Recurrent Neural Networks for Adaptive Communication Channel Equalization, *IEEE Transactions on Neural Networks*, March 1994, Vol. 5, Number 2, pages 267-278
- Galdino J. F. and Pinto E. L., (1998). A New MLSE-PSP Scheme Over Fast Frequency-Selective Fading Channels, *International Symposium on Information Theory and Its Applications*, Mexico City, Mexico, 14-16 October 1998.

IntechOpen



Kalman Filter

Edited by Vedran Kordic

ISBN 978-953-307-094-0

Hard cover, 390 pages

Publisher InTech

Published online 01, May, 2010

Published in print edition May, 2010

The Kalman filter has been successfully employed in diverse areas of study over the last 50 years and the chapters in this book review its recent applications. The editors hope the selected works will be useful to readers, contributing to future developments and improvements of this filtering technique. The aim of this book is to provide an overview of recent developments in Kalman filter theory and their applications in engineering and science. The book is divided into 20 chapters corresponding to recent advances in the field.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Coelho Pedro H G and Biondi Neto Luiz (2010). Complex Extended Kalman Filters for Training Recurrent Neural Network Channel Equalizers, Kalman Filter, Vedran Kordic (Ed.), ISBN: 978-953-307-094-0, InTech, Available from: <http://www.intechopen.com/books/kalman-filter/complex-extended-kalman-filters-for-training-recurrent-neural-network-channel-equalizers>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen