AN AUGMENTED EXTENDED KALMAN FILTER ALGORITHM FOR COMPLEX-VALUED RECURRENT NEURAL NETWORKS

Su Lee Goh and Danilo P. Mandic

Imperial College London
Exhibition Road, London, SW7 2BT, UK
E-mail: {su.goh,d.mandic}@imperial.ac.uk

ABSTRACT

An augmented complex-valued Extended Kalman Filter (ACEKF) algorithm for the class of nonlinear adaptive filters realised as fully connected recurrent neural networks (FCRNNs) is introduced. The algorithm is derived based on the recent developments in augmented complex statistics, and the Jacobian matrix within the ACEKF algorithm is computed using a general fully complex real time recurrent learning (CRTRL) algorithm. This makes ACEKF suitable for processing general complex-valued nonlinear and nonstationary signals and bivariate signals with strong component correlations. Simulations on benchmark and real-world complex-valued signals support the approach.

1. INTRODUCTION

Recent progress in biomedicine, wireless and mobile communications, seismics, sonar and radar has brought to light new problems, where signals are often complex-valued or have a higher-dimensional compact representation [1]. To process such signals, research has largely been directed towards extending the results from real-valued adaptive filters to the complex domain, $\mathbb{C}$. More recently, complex-valued learning algorithms have also been introduced for training of neural networks (NNs) [2][3].

Fully connected recurrent neural networks (FCRNNs) have been employed as nonlinear adaptive filters[1] [1] where for real-time applications in the complex domain, a recently proposed fully complex-valued real-time recurrent learning (CRTRL)[2] algorithm [3] has been applied successfully to forecasting of the complex-valued wind field. These initial results were promising, but it was also realised that gradient based learning may experience problems for signals with rich dynamics and non-Gaussian distributions (e.g intermittent).

A possible solution to this problem may be based on Kalman filters [4], which have been shown to exhibit superior performance in several practical applications, including state estimation for road navigation [5], parameter estimation for time series modeling, and neural network training [6]. Variants of the extended Kalman filter (EKF) algorithm have been used to train temporal NNs [7], with promising results.

Recently, EKF training of NNs has been extended to the complex domain [8]. Notice that, to design an algorithm suitable for the complex domain, we need a precise mathematical model that describes the evolution of system parameters. Hence, extensions of learning algorithms to the complex domain are not trivial and often involve some constraints, for instance, a simplified model of both complex statistics and complex nonlinearities within neurons. This might be sub-optimal for classes of signals with significant correlation between the real and imaginary parts, which affects both the choice of the complex activation function and complex statistics.

To this cause, we first provide mathematical foundations for complex-valued second order statistics and highlight the need to consider the so-called augmented statistics in the derivation of learning algorithms. Next, both the augmented complex-valued Kalman filter (ACKF) and the the augmented extended complex-valued Kalman filter (ACEKF) algorithm are derived, whereby the recently developed CRTRL algorithm is used to compute the Jacobian matrix within the ACEKF. This is supported by simulation examples on a benchmark complex-valued nonlinear signal, together with simulations on complex-valued real-world wind and radar measurements.

2.COMPLEX-VALUED SECOND ORDER STATISTICS

The four covariance matrices of two zero-mean complex-valued random vectors (RVs), $\mathbf{x} = \mathbf{x}^r + j\mathbf{x}^i$ and $\mathbf{y} = \mathbf{y}^r + j\mathbf{y}^i$, are given by [9]

$$\begin{aligned} \mathbf{P}_{\mathbf{x}^r\mathbf{y}^r} &= cov[\mathbf{x}^r, \mathbf{y}^r] \quad \mathbf{P}_{\mathbf{x}^r\mathbf{y}^i} = cov[\mathbf{x}^r, \mathbf{y}^i] \\ \mathbf{P}_{\mathbf{x}^i\mathbf{y}^r} &= cov[\mathbf{x}^i, \mathbf{y}^r] \quad \mathbf{P}_{\mathbf{x}^i\mathbf{y}^i} = cov[\mathbf{x}^r, \mathbf{y}^i] \end{aligned} \tag{1}$$

where $j = \sqrt{-1}$, $(\cdot)^T$ denotes the vector transpose operator, and superscripts $(\cdot)^r$ and $(\cdot)^i$ denote respectively the real and imaginary part of a complex number or complex vector.

[1] Nonlinear autoregressive (NAR) processes can be modelled using feedforward networks, whereas nonlinear autogressive moving average (NARMA) processes can be represented using RNNs.

[2] The CRTRL algorithm uses a fully complex activation function (AF) that is analytic and bounded almost everywhere in $\mathbb{C}$. On the other hand, in a split-complex AF, the real and imaginary components of the input signal x are separated and fed through the real-valued activation function $f_R(x) = f_I(x)$, $x \in \mathbb{R}$. A split-complex activation function is therefore given as $f(x) = f_R(Re(x)) + jf_I(Im(x))$, for example $f(x) = \frac{1}{1+e^{-\beta(Re(x))}} + j\frac{1}{1+e^{-\beta(Im(x))}}$.

These four real-valued matrices are equivalent to the following two complex-valued matrices

$$\mathbf{P}_{\mathbf{xy}} = E\left[\mathbf{x}\mathbf{y}^H\right], \quad \mathbf{P}^{\xi}_{\mathbf{xy}} = E\left[\mathbf{x}\mathbf{y}^T\right] \tag{2}$$

where $(\cdot)^H$ the Hermitian transpose. We can solve (2) for $\mathbf{P}_{\mathbf{x}^r\mathbf{y}^r}, \mathbf{P}_{\mathbf{x}^i\mathbf{y}^i}, \mathbf{P}_{\mathbf{x}^i\mathbf{y}^r}$ and $\mathbf{P}_{\mathbf{x}^r\mathbf{y}^i}$ to obtain

$$\begin{aligned} \mathbf{P}_{\mathbf{x}^r\mathbf{y}^r} &= \frac{1}{2}\mathbb{R}(\mathbf{P}_{\mathbf{xy}} + \mathbf{P}^{\xi}_{\mathbf{xy}}), \quad \mathbf{P}_{\mathbf{x}^i\mathbf{y}^i} = \frac{1}{2}\mathbb{R}(\mathbf{P}_{\mathbf{xy}} - \mathbf{P}^{\xi}_{\mathbf{xy}}) \\ \mathbf{P}_{\mathbf{x}^i\mathbf{y}^r} &= \frac{1}{2}\mathbb{I}(\mathbf{P}_{\mathbf{xy}} + \mathbf{P}^{\xi}_{\mathbf{xy}}), \quad \mathbf{P}_{\mathbf{x}^r\mathbf{y}^i} = \frac{1}{2}\mathbb{I}(-\mathbf{P}_{\mathbf{xy}} + \mathbf{P}^{\xi}_{\mathbf{xy}}) \end{aligned} \tag{3}$$

where symbols $\mathbb{R}$ and $\mathbb{I}$ denote respectively the real and imaginary part of a complex quantity. This shows that the four real-valued covariance matrices (3) are in a one-to-one relationship with the two complex-valued covariance matrices in (2). In the literature, nearly always only $\mathbf{P}_{\mathbf{xy}}$ is considered, and is referred to as the *covariance matrix*, whereas $\mathbf{P}^{\xi}_{\mathbf{xy}}$ [3] is the so-called *pseudo-covariance* matrix.

2.1. Augmented Covariance Matrix

It is often assumed that the theory of complex-valued random vectors (RVs) is no different than that of the real RVs, as long as the definition of the covariance matrix of an RV $\mathbf{x}$ is changed from $E[\mathbf{x}\mathbf{x}^T]$ in the real case to $E[\mathbf{x}\mathbf{x}^H]$, using the Hermitian transpose $(\cdot)^H$ [10]. This assumption, however, is not justified since the covariance matrix $E[\mathbf{x}\mathbf{x}^H]$ will **not** completely describe the second-order statistical behavior of $\mathbf{x}$. For complex-valued Gaussian random variables, we need to consider both the variable $\mathbf{x}$ and its complex conjugate $\mathbf{x}^*$ in order to make use of for all the available statistical information. This additional information is therefore contained in the cross-moments [9]. We therefore set out to derive a complex-valued Kalman filter which takes into account the augmented complex statistics. Thus, we consider complex RV $\mathbf{x}$ and produce a $2n \times 1$ vector $\mathbf{x}^a = [\mathbf{x}, \mathbf{x}^*]$. It is the augmented $2n \times 2n$ covariance matrix $\mathbf{P}_{\mathbf{x}^a\mathbf{x}^a} = E[\mathbf{x}^a(\mathbf{x}^a)^T]$ (rather than just the $n \times n$ matrix $\mathbf{P}_{\mathbf{xx}} = E[\mathbf{x}\mathbf{x}^H]$) that contains the complete second-order statistical information. Such an augmented covariance matrix[4] is given by [10]

$$\mathbf{P}_{\mathbf{x}^a\mathbf{x}^a} = E\begin{bmatrix} \mathbf{x} \\ \mathbf{x}^* \end{bmatrix}\left[\mathbf{x}^T\mathbf{x}^H\right] = \begin{bmatrix} \mathbf{P}_{\mathbf{xx}} & \mathbf{P}^{\xi}_{\mathbf{xx}} \\ \mathbf{P}^{\xi*}_{\mathbf{xx}} & \mathbf{P}^{*}_{\mathbf{xx}} \end{bmatrix} \tag{4}$$

3. THE AUGMENTED COMPLEX-VALUED KALMAN FILTER (ACKF) ALGORITHM

Based on the analysis of augmented complex statistics, we first derive a Kalman filter learning algorithm for complex-valued inputs using the augmented states and augmented covariance matrix. A general state-space model is given as [6]

$$\mathbf{x}_{k+1} = \mathbf{F}_{k+1}\mathbf{x}_k + \boldsymbol{\omega}_k, \quad \mathbf{y}_k = \mathbf{H}_k\mathbf{x}_k + \boldsymbol{\nu}_k \tag{5}$$

where $\boldsymbol{\omega}_k$ and $\mathbf{v}_k$ are independent, zero-mean complex-valued Gaussian processes of covariance matrices $\mathbf{Q}_k$ and $\mathbf{R}_k$ respectively, and $\mathbf{F}$ and $\mathbf{H}$ are the transition and measurement matrix. From (5), the augmented state-space model is obtained as

$$\mathbf{x}^a_{k+1} = \mathbf{F}^a_{k+1}\mathbf{x}^a_k + \boldsymbol{\omega}^a_k, \quad \mathbf{y}^a_k = \mathbf{H}^a_k\mathbf{x}^a_k + \boldsymbol{\nu}^a_k \tag{6}$$

where $\mathbf{x}^a_k = [\mathbf{x}_k, \mathbf{x}^*_k]$, $\mathbf{y}^a_k = [\mathbf{y}_k, \mathbf{y}^*_k]$, $\mathbf{F}^a_k = [\mathbf{F}_k, \mathbf{F}^*_k]$, $\mathbf{H}^a_k = [\mathbf{H}_k, \mathbf{H}^*_k]$, $\boldsymbol{\omega}^a_k = [\boldsymbol{\omega}_k, \boldsymbol{\omega}^*_k]$ and $\boldsymbol{\nu}^a_k = [\boldsymbol{\nu}_k, \boldsymbol{\nu}^*_k]$. The augmented covariance matrices of the zero-mean complex-valued Gaussian noise processes are denoted respectively by $\mathbf{Q}^a_k$ and $\mathbf{R}^a_k$. In the initialization of the algorithm ($k = 0$), we set

$$\hat{\mathbf{x}}^a_0 = E\left[\mathbf{x}^a_0\right], \quad \mathbf{P}_0 = E\left[\left(\mathbf{x}^a_0 - E\left[\mathbf{x}^a_0\right]\right)\left(\mathbf{x}^a_0 - E\left[\mathbf{x}^a_0\right]\right)^T\right] \tag{7}$$

The computation steps for $k = 1, \ldots, L$ are given below using similar notation as in [6]:-

State estimate propagation:

$$\hat{\mathbf{x}}^{a-}_k = \mathbf{F}^a_{k,k-1}\hat{\mathbf{x}}^{a-}_{k-1} \tag{8}$$

Error covariance propagation:

$$\mathbf{P}^-_k = \mathbf{F}^a_{k,k-1}\mathbf{P}_{k-1}(\mathbf{F}^a_{k,k-1})^H + \mathbf{Q}^a_{k-1} \tag{9}$$

Kalman gain matrix:

$$\mathbf{G}_k = \mathbf{P}^-_k(\mathbf{H}^a_k)^H\left[\mathbf{H}^a_k\mathbf{P}_k(\mathbf{H}^a_k)^H + \mathbf{R}^a_k\right]^{-1} \tag{10}$$

State estimate update:

$$\hat{\mathbf{x}}^a_k = \hat{\mathbf{x}}^{a-}_k + \mathbf{G}_k\left(\mathbf{y}^a_k - \mathbf{H}^a_k\hat{\mathbf{x}}^{a-}_k\right) \tag{11}$$

Error covariance update:

$$\mathbf{P}_k = \left(\mathbf{I} - \mathbf{G}_k\mathbf{H}^a_k\right)\mathbf{P}^-_k \tag{12}$$

This completes the description of the augmented complex-valued Kalman filter.

4. FCRNN TRAINED WITH AUGMENTED EXTENDED COMPLEX-VALUED KALMAN FILTER

4.1. The FCRNN Architecture

Figure 1 shows an FCRNN, which consist of N neurons with p external inputs and N feedback connections. Let $y_{l,k}$ denote the complex-valued output of a neuron, $l = 1, \ldots, N$ at time index k and $\mathbf{s}_k$ the $(1 \times p)$ external complex-valued input vector. The overall input to the network $\mathbf{u}_k$ then represents a concatenation of vectors $\mathbf{y}_k$, $\mathbf{s}_k$ and the bias input $(1 + j)$, and is given by

$$\begin{aligned} \mathbf{u}_k &= \left[s_{k-1}, \ldots, s_{k-p}, 1 + j, y_{1,k-1}, \ldots, y_{N,k-1}\right]^T \\ u_{n,k} &\in \mathbf{u}_k = u^r_{n,k} + ju^i_{n,k}, \quad n = 1, \ldots, p + N + 1 \end{aligned} \tag{13}$$

For the lth neuron, its weights form a $(p + N + 1) \times 1$ dimensional weight vector $\mathbf{w}^T_l = [w_{l,1}, \ldots, w_{l,p+N+1}]$, $l = 1, \ldots, N$, which are encompassed in the complex-valued weight

[3] A complex RV $\mathbf{x}$ is called *proper* if its pseudo-covariance $\mathbf{P}^{\xi}_{\mathbf{xx}}$ vanishes [9, 10]. For convenience, in many applications, complex-valued random vectors (RVs) are treated as proper. However, $\mathbf{P}^{\xi}_{\mathbf{xx}}$ may not be necessarily zero, in this case is called *improper* complex-valued RVs.

[4] The covariance matrix $\mathbf{P}_{\mathbf{x}^a\mathbf{x}^a}$ is invertible and thus positive definite. Besides $\mathbf{P}_{\mathbf{xx}}$ being positive semi-definite (PSD) and $\mathbf{P}^{\xi*}_{\mathbf{xx}}$ being symmetric, the Schur complement $\mathbf{P}^*_{\mathbf{xx}} - \mathbf{P}^{\xi*}_{\mathbf{xx}}\mathbf{P}^{-1}_{\mathbf{xx}}\mathbf{P}^{\xi}_{\mathbf{xx}}$ must be PSD to ensure that $\mathbf{P}_{\mathbf{x}^a\mathbf{x}^a}$ is PSD and thus a valid covariance matrix for $\mathbf{x}^a$ [9].

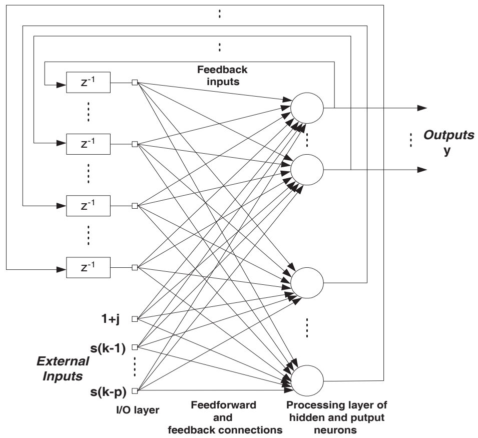


Fig. 1. A fully connected recurrent neural network (FCRNN) for prediction.

matrix of the network $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_N]$. The output of every neuron can be expressed as $y_{l,k} = \Phi(net_{l,k})$ where Φ is a complex nonlinear activation function of a neuron.
To establish a mathematical framework for the Kalman filter training of FCRNNs, the dynamical behaviour of the FCRNN can be described by the state-space model given as

$$\mathbf{w}^a_{k+1} = \mathbf{w}^a_k + \boldsymbol{\omega}^a_k, \quad \mathbf{y}^a_k = \mathbf{h}(\mathbf{w}^a_k, \mathbf{u}_k) + \boldsymbol{\nu}^a_k \tag{14}$$

where $\mathbf{h}$ is a nonlinear operator associated with observations, $\mathbf{w}^a_k$ is the augmented weight vector and $\mathbf{y}^a_k$ is the augmented output of the network. The basic idea of training such architecture is to linearise the state space model in equation (14) at each time instant. Once a linear model is obtained, the ACKF equations are applied. In practice, the process noise covariance $\mathbf{Q}^a_k$ and measurement noise covariance $\mathbf{R}^a_k$ matrices might be time-varying, but here, we assume they are constant. The Jacobian $\mathbf{H}^a_k$ is defined as a set of partial derivatives of the outputs with respect to the weights, $\mathbf{w}^a_k$, and needs to be calculated for every time instant.

4.2. Derivation of the Augmented Complex-valued Extended Kalman Filter (ACEKF) Algorithm

The derivation of the complex-valued Jacobian required within the matrices is nontrivial and the linearization can produce highly unstable filter performance if the discrete time intervals are not sufficiently small. Therefore, careful parameter selection is required when using the ACEKF algorithm. The derivation of the ACEKF is an extension from the ACKF given in the previous section. For consistency, the ACEKF is summarised using the notation from [6] and is given by the following recursion

$$\mathbf{G}_k = \mathbf{P}^-_k(\mathbf{H}^a_k)^H\left[\mathbf{H}^a_k\mathbf{P}^-_k(\mathbf{H}^a_k)^H + \mathbf{R}^a_k\right]^{-1} \tag{15}$$

$$\hat{\mathbf{w}}^a_k = \hat{\mathbf{w}}^{a-}_k + \mathbf{G}_k\left[\mathbf{y}^a_k - \mathbf{h}(\hat{\mathbf{w}}^{a-}_k, \mathbf{u}_k)\right] \tag{16}$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{G}_k\mathbf{H}^a_k)\,\mathbf{P}^-_k + \mathbf{Q}^a_k \tag{17}$$

The initialization of the ACEKF is given as

$$\hat{\mathbf{w}}^a_0 = E\left[\mathbf{w}_0\right], \, \mathbf{P}_0 = E\left[\left(\mathbf{w}^a_0 - E\left[\mathbf{w}^a_0\right]\right)\left(\mathbf{w}^a_0 - E\left[\mathbf{w}^a_0\right]\right)^T\right] \tag{18}$$

For generality, the augmented Jacobian matrix $\mathbf{H}^a_k$ is computed via the augmented CRTRL algorithm[5] for recurrent networks [3] which uses fully complex nonlinear filters within neurons. The vector $\hat{\mathbf{w}}^a_k$ represents the estimate of the augmented state of the system update at step k. The Kalman gain matrix $\mathbf{G}_k$ is a function of the approximate error covariance matrix $\mathbf{P}_k$, the Jacobian matrix $\mathbf{H}^a_k$ and a global scaling matrix $\mathbf{H}^a_k\mathbf{P}^-_k(\mathbf{H}^a_k)^H + \mathbf{R}^a_k$.

5. SIMULATIONS

For the experiments, the nonlinearity at the neuron was chosen to be the complex $tanh$ function $\Phi(x) = \frac{e^{\beta x} - e^{-\beta x}}{e^{\beta x} + e^{-\beta x}}$ where $x \in \mathbb{C}$. The value of the slope of $\Phi(x)$ was $\beta = 1$. The architecture of the FCRNN (Figure 1) consisted of $N = 3$ neurons with the tap input length of $p = 5$. To support the analysis, we tested the ACEKF on complex nonlinear signals. To further illustrate the approach and verify the advantage of using the ACEKF over the CRTRL, single-trial experiments were performed on real-world complex-valued wind[6] and radar[7] data. The complex benchmark nonlinear input signal was [1]

$$z(k) = \frac{z^2(k-1)(z(k-1)+2.5)}{1 + z(k-1) + z^2(k-2)} + n(k-1) \tag{19}$$

with complex white Gaussian noise (CWGN) $n(k) \sim \mathcal{N}(0,1)$ as the driving input. The input signals used here are *improper* complex-valued in order to show the advantage of the proposed ACEKF over the standard algorithms. The measurement used to assess the performance was the prediction gain $R_p(k) \triangleq 10log_{10}\left(\frac{\sigma^2_x}{\hat{\sigma}^2_e}\right)[dB]$ where σ^2_x denotes the variance of the input signal $x(k)$, whereas $\hat{\sigma}^2_e$ denotes the estimated variance of the forward prediction error $e(k)$. The signal-to-noise (SNR) ratio for all the simulations is 3[dB] . Table 1 shows a comparison of the prediction gains R_p [dB] between the ACEKF, standard complex-valued (CEKF) (without considering the augmented states) and CRTRL for various classes of signals. In all the cases, there was a significant improvement in the prediction gain when ACEKF method was employed over the performance of the CRTRL algorithm. The proposed ACEKF exhibited a more stable and better performance than the standard CEKF for a range of complex signals used.

Figure 2 shows a subsegment of the predictions generated by both the ACEKF and CRTRL for complex-valued nonlinear (19) signal. The performance of ACEKF is clearly much better than that of CRTRL for both cases. To verify the advantage

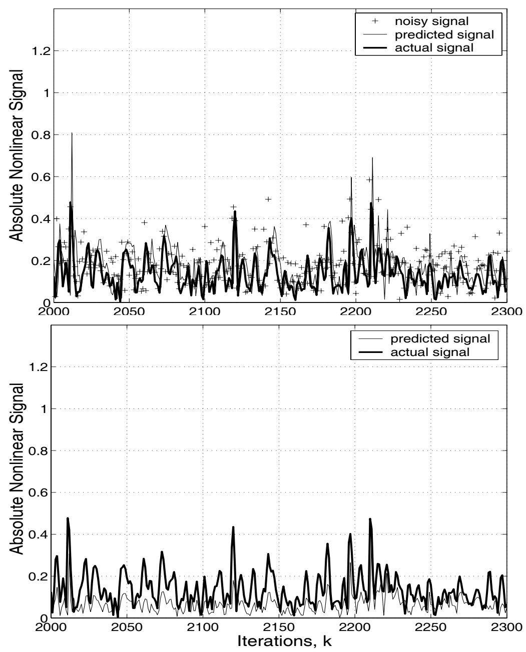
[5] Matrix $\mathbf{H}^a_k$ denotes the matrix of partial derivatives of the network's augmented output $\mathbf{y}^a_k$ with respect to weight parameters.

[6] Publicly available from *"http://mesonet.agron.iastate.edu/"*. The wind vector can be expressed in the complex domain $\mathbb{C}$ as $v(t)e^{j\theta(t)} = v_E(t) + jv_N(t)$.

[7] Publicly available from *"http://soma.ece.mcmaster.ca/ipix/"*.

Table 1. Comparison of prediction gains R_p for the various classes of signals

Signal	Nonlinear	Wind	Radar
R_p [dB] (ACEKF)	5.12	11.17	10.58
R_p [dB] (standard CEKF)	4.50	10.05	9.91
R_p [dB] (CRTRL)	3.15	7.52	7.22



(b) CRTRL Algorithm

Fig. 2. Comparison of one-step ahead prediction performance between the ACEKF and CRTRL for nonlinear signal (19)

of using the ACEKF over CRTRL, we compared the performances of FCRNNs trained with these algorithms in experiments on real world wind data. Figure 3 shows the prediction performance of the ACEKF applied to the complex-valued real-world (velocity and angle components) wind signal. The ACEKF algorithm was more stable and has exhibited better and more consistent performance than the CRTRL algorithm.

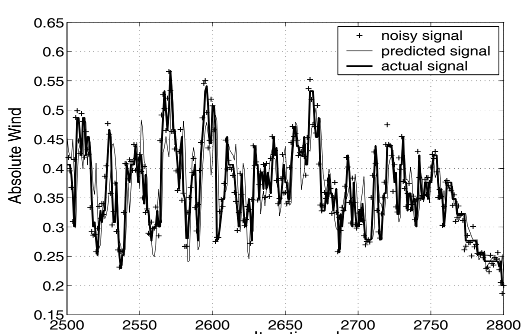


Fig. 3. Prediction performance for complex wind signal using ACEKF

6. CONCLUSIONS AND DISCUSSIONS

The augmented complex-valued extended Kalman filter (ACEKF) has been introduced for nonlinear adaptive filtering using FCRNNs in the complex domain. The ACEKF has been derived using the augmented complex-valued statistics, whereby, a complete second order statistics for complex-valued quantities is taken into account. The computational cost of the ACEKF is relatively more expensive than that of the CRTRL. In the future, a decoupled complex-valued EKF could be implemented to relax the computation complexity. The performance of the ACEKF has been evaluated on benchmark complex-valued nonlinear input signals, and also on real-life complex-valued wind and radar signals. Simulation results have justified the potential of ACEKF in nonlinear complex-valued neural adaptive filtering applications.

7. REFERENCES

[1] D. P. Mandic and J. A. Chambers, *Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability*, John Wiley & Sons, 2001.

[2] T. Kim and T. Adali, "Approximation by fully complex MLP using elementary transcendental activation functions," *In Proceedings of XI IEEE Workshop on Neural Networks for Signal Processing*, pp. 203–212, 2001.

[3] S. L. Goh and D. P. Mandic, "A complex-valued RTRL algorithm for recurrent neural networks," *Neural Computation*, vol. 16, no. 12, pp. 2699–2713, 2004.

[4] G. V. Puskorius and L. A. Feldkamp, "Neurocontrol of nonlinear dynamical systems with Kalman filter trained recurrent networks," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 279–297, 1994.

[5] D. Obradovic, H. Lenz, and M. Schupfner, "Sensor fusion in Siemens car navigation system," *In Proceeding of the IEEE Signal Processing Society Workshop on Machine Learning for Signal Processing*, pp. 655–664, 2004.

[6] S. Haykin, *Kalman Filtering and Neural Networks*, John Wiley & Sons, 2001.

[7] D. Mandic, J. Baltersee, and J. Chambers, "Non-linear adaptive prediction of speech with a pipelined recurrent neural network and advanced learning algorithms," *A. Prochazka, J. Uhlir, P. W. Rayner, and N. G. Kingsbury, Eds.,* Signal Analysis and Prediction*, Birkhauser, Boston*, 1998.

[8] R. C. Huang and M. S. Chen, "Adaptive equalization using complex-valued multilayered neural network based on the extended kalman filter," *Proceedings of Signal Processing, International Conference on WCCC-ICSP*, vol. 1, pp. 519–524, 2000.

[9] F.D. Neeser and J.L. Massey, "Proper complex random processes with applications to information theory," *IEEE Transactions on Information Theory*, vol. 39, no. 4, pp. 1293–1302, 1992.

[10] P. J. Schreier and L. L. Scharf, "Second-order analysis of improper complex random vectors and processes," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 51, no. 3, pp. 714–725, 2003.