Available online at www.sciencedirect.com

ScienceDirect

Neural Networks 20 (2007) 1061–1066

Neural Networks

www.elsevier.com/locate/neunet

An augmented CRTRL for complex-valued recurrent neural networks

Su Lee Goh[1], Danilo P. Mandic*

Imperial College London, Exhibition Road, London SW7 2AZ, United Kingdom

Abstract

Real world processes with an "intensity" and "direction" component can be made complex by convenience of representation (vector fields, radar, sonar), and their processing directly in the field of complex numbers $\mathbb{C}$ is not only natural but is also becoming commonplace in modern applications. Yet, adaptive signal processing and machine learning algorithms suitable for the processing of such signals directly in $\mathbb{C}$ are only emerging. To this cause we introduce a second order statistical learning framework for a general class of nonlinear adaptive filters with feedback realized as recurrent neural networks (RNNs). For rigour, both the so-called proper- and improper-second order statistics of complex processes is taken into account, and the proposed augmented complex real-time recurrent learning (ACRTRL) algorithm for RNNs has been shown to be suitable for processing a wide range of both benchmark and real-world complex processes.

Keywords: Complex nonlinear adaptive filters; Recurrent neural networks (RNNs); Complex-valued RTRL (CRTRL); Augmented Complex Statistics

1. Introduction

Temporal recurrent neural networks (RNNs) are a class of nonlinear adaptive filters with feedback which have found their applications when processing nonlinear and nonstationary signals, and signals coming from systems with long impulse responses. This is mainly due to their ability to represent highly nonlinear dynamic systems (Elman, 1990), the associated attractor dynamics and inherent memory within the feedback (Medsker & Jain, 2000; Principe, Euliano, & Lefebvre, 2000); this makes them suitable for nonlinear autoregressive moving average modelling (NARMA), unlike feedforward networks which by design cannot model feedback systems. Real world signals are typically nonstationary, nonlinear and nonGaussian, and temporal RNNs are perfectly suited for processing such signals; nonstationarity is accounted for by their sequential mode of operation, their feedback caters for long impulse responses, and the inherent nonlinearity helps to model nonlinear and non-Gaussian signals and the associated higher order statistics.

Due to a number of emerging applications in the complex domain $\mathbb{C}$, it is therefore natural to extend RNNs to this domain. One unique feature of signal processing in the complex domain is that properties of processes in $\mathbb{C}$ vary not only in terms of their linear/nonlinear behaviour but also in terms of their bivariate or complex-valued nature (dual univariate, split complex, fully complex) (Gautama, Mandic, & Hulle, 2003). To that end, research towards a general, 'fully complex' RNN has recently focused on the issue of analytical nonlinear activation functions (AF)s within neurons. A comprehensive account of elementary complex transcendental activation functions (ETFs) used as AFs is given in Kim and Adali (2003), whereby ETFs were employed to derive a fully complex backpropagation (CBP) algorithm. This was achieved by making use of the Cauchy–Riemann[2] conditions; this also helped to relax the requirements on the desired properties of nonlinearities within fully complex neurons.[3] A recently proposed complex-valued real-time recurrent learning (CRTRL) algorithm (Goh & Mandic, 2004) has illustrated the possibility of using this strategy for the forecasting of

* Corresponding author. Tel.: +44 (0) 2075946271; fax: +44 (0) 2075946234.
E-mail addresses: vanessa.goh@shell.com (S.L. Goh), d.mandic@imperial.ac.uk (D.P. Mandic).

[1] Tel.: +44 (0) 2075946271; fax: +44 (0) 2075946234.

[2] Cauchy–Riemann equations state that the partial derivatives of a function $f(z) = u(x, y) + \mathrm{j}v(x, y)$ along the real and imaginary axes should be equal: $f'(z) = \frac{\partial u}{\partial x} + \mathrm{j}\frac{\partial v}{\partial x} = \frac{\partial v}{\partial y} - \mathrm{j}\frac{\partial u}{\partial y}$. This way $\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y}$, $\frac{\partial v}{\partial x} = -\frac{\partial u}{\partial y}$.

[3] A fully complex activation function is analytical and bounded almost everywhere in $\mathbb{C}$.

0893-6080/$ - see front matter
doi:10.1016/j.neunet.2007.09.015

complex-valued vector fields. The CRTRL algorithm has been derived for a general meromorphic complex activation function of a neuron and has a generic form of the real-valued RTRL (Williams & Zipser, 1989).

To design an algorithm suitable for nonlinear adaptive filtering in the complex domain $\mathbb{C}$, we need a precise mathematical model that describes the evolution of system parameters. Hence, extensions of learning algorithms from $\mathbb{R}$ to $\mathbb{C}$ are not trivial and often involve some constraints, for instance, simplified models of both complex statistics and complex nonlinearities within neurons. This might prove suboptimal for classes of signals with significant correlation between the real and imaginary parts, and we ought to seek alternative ways to include the full second order statistical information available.

To that end, we have recently introduced the so-called augmented complex statistics into the derivation of state space based algorithms (Goh & Mandic, 2007). This way, we can circumvent the usual assumptions of circularity (rotation invariant distribution), and orthogonality between the real and imaginary channel, and design algorithms suitable for general classes of complex-valued processes. Both the Augmented Complex Kalman Filter (ACKF) and Augmented Complex Extended Kalman Filter (ACEKF) algorithm were derived, whereby the corresponding variants of complex RTRL algorithm (Goh & Mandic, 2004) were used to compute the Jacobian matrix within the ACEKF.

Notice that gradient-based learning complements state space methods; however, a general framework for gradient-based learning using augmented complex statistic is still lacking. This letter therefore aims at providing a gradient-based learning strategy for the training of complex RNNs for temporal problems. This way, the properties of complex nonlinearities and so called augmented second order complex statistics, are combined in order to make full use of the available information. The proposed approach is general and rigorous; the corresponding algorithms for feedforward neural networks and linear filters can be obtained straighforwardly by respectively cancelling feedback and using liner neurons. The analysis is comprehensive and is supported by simulation examples for both the standard and augmented CRTRL, performed on benchmark complex-valued coloured and nonlinear signals, together with simulations on complex-valued real-world radar and wind measurements.

1.1. Complex-valued augmented covariance matrix

To provide brief insight into so called augmented complex statistics, observe that within the definition of complex Gaussian variable, the only nonzero second-order moment is $E[\mathbf{x}\mathbf{x}^*]$. This effectively means that both $\mathbf{x}$ and $\mathbf{x}^*$ need to be taken into account when designing learning algorithms in $\mathbb{C}$. In other words, the full information is obtained not only from individual statistics of the two constitutive variables, but also from their cross-statistics (Schreier & Scharf, 2003). In practical terms, this means that we can "augment" the input vector $\mathbf{x}$ with its conjugate $\mathbf{x}^*$ to produce the $2n \times 1$ vector $\boldsymbol{\Lambda} = [\mathbf{x}^T, \mathbf{x}^H]^T$. Such a transformation preserves linearity in $\boldsymbol{\Lambda}$, and is referred to as "widely linear" in $\mathbf{x}$, due to the dependence on both $\mathbf{x}$ and $\mathbf{x}^*$.

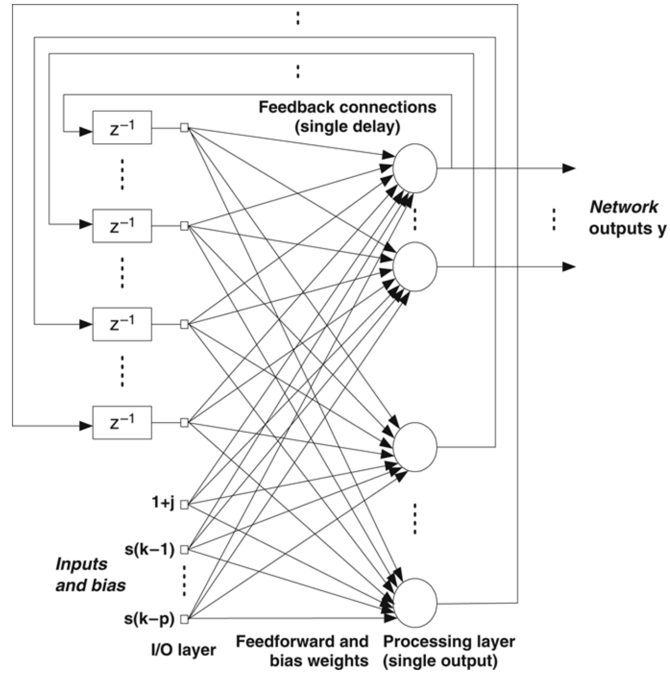

Fig. 1. A fully connected recurrent neural network (FCRNN).

Our aim is therefore to provide a rigorous derivation of gradient-based learning of complex valued RNNs, which, for generality, is based on the augmented complex statistics, making this framework suitable for the processing of general complex valued signals. Consequently, for so-called improper[4] random variables, it is the augmented $(2n \times 2n)$ complex 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 can be evaluated as Neeser and Massey (1992), Schreier and Scharf (2003)

$$\mathbf{P}_{\mathbf{x}^a\mathbf{x}^a} = E\begin{bmatrix}\mathbf{x}\\ \mathbf{x}^*\end{bmatrix}\begin{bmatrix}\mathbf{x}^T & \mathbf{x}^H\end{bmatrix} = \begin{bmatrix}\mathbf{P}_{\mathbf{xx}} & \mathbf{P}^{\xi}_{\mathbf{xx}}\\ \mathbf{P}^{\xi *}_{\mathbf{xx}} & \mathbf{P}^{*}_{\mathbf{xx}}\end{bmatrix}. \tag{1}$$

2. The augmented complex-valued RTRL (ACRTRL) algorithm

Fig. 1 shows an FCRNN, which consists of N neurons with p external inputs. The network has two distinct layers consisting of the external input-feedback layer and a layer of processing elements. Let $y_l(k)$ denote the complex-valued output of each neuron, $l = 1, \ldots, N$ at time index k, and $\mathbf{s}(k)$ the $(1 \times p)$ external complex-valued input vector. The

[4] A complex RV $\mathbf{x}$ is called *proper* if its pseudo-covariance $\mathbf{P}^{\xi}_{\mathbf{xx}}$ vanishes (Neeser & Massey, 1992; Schreier & Scharf, 2003). 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.

overall input to the network $\mathbf{I}(k)$ represents the concatenation of vectors $\mathbf{y}(k)$, $\mathbf{s}(k)$ and the bias input $(1 + \mathrm{j})$, and is given by

$$\begin{aligned}\mathbf{I}(k) &= [s(k-1), \ldots, s(k-p), 1+\mathrm{j},\\ &\quad y_1(k-1), \ldots, y_N(k-1)]^T\\ &= I_n^r(k) + \mathrm{j} I_n^i(k), \quad n = 1, \ldots, p+N+1. \end{aligned} \tag{2}$$

In general, a complex-valued weight matrix of the network is denoted by $\mathbf{W}$, where for the lth neuron, its weights form a $(p + F + 1) \times 1$ dimensional weight vector $\mathbf{w}_l = \left[w_{l,1}, \ldots, w_{l,p+F+1}\right]^T$ where F is the number of feedback connections. Since in our case, the feedback connections represent the delayed output signals of the FCRNN, from Fig. 1, we have $F = N$.

The output of each neuron can be expressed as

$$y_l(k) = \Phi(\mathrm{net}_l(k)), \quad l = 1, \ldots, N \tag{3}$$

where

$$\mathrm{net}_l(k) = \sum_{n=1}^{p+N+1} w_{l,n}(k) I_n(k) \tag{4}$$

is the net input to lth node at time index k. The symbol Φ is a complex nonlinear activation function. The weight update is given as

$$\begin{aligned}\Delta w_{s,t}(k) &= \eta \sum_{l=1}^{N} e_l(k) \left(\pi_{s,t}^l\right)^*(k)\\ &\quad 1 \le l,\ s \le N,\ 1 \le t \le p+N+1\end{aligned} \tag{5}$$

where the update for the sensitivities $(\pi_{s,t}^l)^*(k)$ is given by[5] Goh and Mandic (2004)

$$\left(\pi_{s,t}^l\right)^*(k) = \{\Phi'(k)\}^* \left[\mathbf{w}_l^H(k)\boldsymbol{\pi}^*(k-1) + \delta_{sl} I_t^*(k)\right] \tag{6}$$

where

$$\delta_{sl} = \begin{cases} 1, & l = s\\ 0, & l \neq s \end{cases} \tag{7}$$

is the Kronecker delta.

2.1. Derivation of the augmented complex-valued RTRL (ACRTRL) algorithm

Based on the analysis of augmented complex statistics given in Section 2, we shall derive the corresponding CRTRL algorithm using the augmented complex-valued input and augmented weight matrix. From (5), the overall 'augmented' input to the network $\mathbf{I}^a(k)$ becomes

$$\begin{aligned}\mathbf{I}^a(k) &= [\mathbf{I}(k), \mathbf{I}^*(k)]^T = (I_n^a)^r(k) + j(I_n^a)^i(k),\\ &\quad n = 1, \ldots, 2(p+N+1).\end{aligned} \tag{8}$$

[5] Recall that for the gradient calculation, we need to calculate the conjugate of the sensitivities $\{(\pi_{s,t}^l)^*(k)\}$. For a straightforward explanation, please look at the complex LMS (Widrow, McCool, & Ball, 1975).

A complex-valued augmented weight matrix of the network is denoted by $\mathbf{W}^a$, where for the lth neuron, its weights form a $2(p + N + 1) \times 1$ dimensional weight vector $\mathbf{w}_l^a = [w_{l,1}, \ldots, w_{l,p+N+1}, w_{l,1}^*, \ldots, w_{l,p+N+1}^*]^T$. The output of each neuron can be expressed as

$$y_l(k) = \Phi(\mathrm{net}_l^a(k)), \quad l = 1, \ldots, N \tag{9}$$

where

$$\mathrm{net}_l^a(k) = \sum_{n=1}^{2(p+N+1)} w_{l,n}^a(k) I_n^a(k) \tag{10}$$

is the net input to lth node at time index k.

For simplicity we state that

$$\begin{aligned}y_l(k) &= \Phi^r(\mathrm{net}_l^a(k)) + \mathrm{j}\Phi^i(\mathrm{net}_l^a(k)) = u_l(k) + \mathrm{j}v_l(k)\\ \mathrm{net}_l^a(k) &= \sigma_l^a(k) + \mathrm{j}\tau_l^a(k).\end{aligned} \tag{11}$$

For real-time applications, the cost function of the recurrent network is given by Widrow et al. (1975)

$$\begin{aligned}E(k) &= \frac{1}{2}\sum_{l=1}^{N} |e_l(k)|^2 = \frac{1}{2}\sum_{l=1}^{N} e_l(k) e_l^*(k)\\ &= \frac{1}{2}\sum_{l=1}^{N} \left[(e_l^r)^2 + (e_l^i)^2\right].\end{aligned} \tag{12}$$

Notice that $E(k)$ is a real-valued function and we are required to derive the gradient $E(k)$ with respect to both the real and imaginary part of the augmented complex-valued weights, that is

$$\begin{aligned}\nabla_{w_{s,t}^a} E(k) &= \frac{\partial E(k)}{\partial (w_{s,t}^a)^r} + \mathrm{j}\frac{\partial E(k)}{\partial (w_{s,t}^a)^i},\\ &\quad 1 \le l,\ s \le N,\ 1 \le t \le 2(p+N+1).\end{aligned} \tag{13}$$

The ACRTRL algorithm minimizes cost function $E(k)$ by recursively altering such weight coefficients based on gradient descent, which yields

$$\begin{aligned}w_{s,t}^a(k+1) &= w_{s,t}^a(k) + \Delta w_{s,t}^a(k)\\ &= w_{s,t}^a(k) - \eta \nabla_{w_{s,t}^a} E(k)_{|w_{s,t}^a = w_{s,t}^a(k)}\end{aligned} \tag{14}$$

where η is the learning rate, a small positive constant.

Calculating the gradient of the cost function with respect to the real part of the complex weight gives

$$\begin{aligned}\frac{\partial E(k)}{\partial (w_{s,t}^a)^r} &= \frac{\partial E}{\partial u_l}\left(\frac{\partial u_l}{\partial (w_{s,t}^a)^r}\right) + \frac{\partial E}{\partial v_l}\left(\frac{\partial v_l}{\partial (w_{s,t}^a)^r}\right)\\ &= \frac{\partial E}{\partial u_l}\left(\Lambda_{s,t}^{l,(rr)}(k)\right) + \frac{\partial E}{\partial v_l}\left(\Lambda_{s,t}^{l,(ir)}(k)\right)\\ &\quad 1 \le l,\ s \le N\ 1 \le t \le 2(p+N+1).\end{aligned} \tag{15}$$

Similarly, the partial derivative of the cost function with respect to the imaginary part of the complex weight yields

$$\begin{aligned}\frac{\partial E(k)}{\partial (w_{s,t}^a)^i} &= \frac{\partial E}{\partial u_l}\left(\frac{\partial u_l}{\partial (w_{s,t}^a)^i}\right) + \frac{\partial E}{\partial v_l}\left(\frac{\partial v_l}{\partial (w_{s,t}^a)^i}\right)\\ &= \frac{\partial E}{\partial u_l}\left(\Lambda_{s,t}^{l,(ri)}(k)\right) + \frac{\partial E}{\partial v_l}\left(\Lambda_{s,t}^{l,(ii)}(k)\right)\\ &\quad 1 \le l,\ s \le N\ 1 \le t \le 2(p+N+1).\end{aligned} \tag{16}$$

Table 1
Statistical properties of the input data sets

Datasets	$AR(4)$ (20)	Nonlinear (21)	Radar (low)	Radar (high)	Wind
Cumulative samples	3000	3000	3000	3000	3000
Minimum value	−0.0615 − j0.0564	0.0203 + j0.1828	0.0129 − j0.0069	0.0142 + j0.0023	0
Maximum value	7.4477 − j7.1359	79.5360 + j33.7422	3.3640 + j0.7781	−2.3618 + j2.1945	11.5835 − j6.6877
Mean value	0.1459 + j0.1660	2.9655 + j0.3822	−0.0035 − j0.0019	−0.0030 − j0.0057	−1.4798 − j1.8394
Standard deviation	3.6685	9.3578	1.1314	1.2429	4.3301

The partial derivatives $\frac{\partial y_l(k)}{\partial (w^a_{s,t})^r(k)} = \Lambda^{l,(rr)}_{s,t}(k) + \mathrm{j}\Lambda^{l,(ir)}_{s,t}(k)$ and $\frac{\partial y_l(k)}{\partial (w^a_{s,t})^i(k)} = \Lambda^{l,(ri)}_{s,t}(k) + \mathrm{j}\Lambda^{l,(ii)}_{s,t}(k)$ are measures of sensitivity of the output at time k to a small variation in the value of the augmented weight $w^a_{s,t}(k)$.

To compute these sensitivities, we start with differentiating (11) which yields

$$\frac{\partial \sigma^a_l(k)}{\partial (w^a_{s,t})^r(k)} = \left[\sum_{q=1}^{N}\left(\frac{\partial u_q(k-1)}{\partial (w^a_{s,t})^r(k)}(w^a_{l,p+1+q})^r(k) - \frac{\partial v_q(k-1)}{\partial (w^a_{s,t})^r(k)}(w^a_{l,p+1+q})^i(k)\right)\right] + \delta_{sl}(I^a_n)^r(k)$$

$$\frac{\partial \tau^a_l(k)}{\partial (w^a_{s,t})^r(k)} = \left[\sum_{q=1}^{N}\left(\frac{\partial v_q(k-1)}{\partial (w^a_{s,t})^r(k)}(w^a_{l,p+1+q})^r(k) + \frac{\partial u_q(k-1)}{\partial (w^a_{s,t})^r(k)}(w^a_{l,p+1+q})^i(k)\right)\right] + \delta_{sl}(I^a_n)^i(k)$$

$$\frac{\partial \sigma^a_l(k)}{\partial (w^a_{s,t})^i(k)} = \left[\sum_{q=1}^{N}\left(\frac{\partial u_q(k-1)}{\partial (w^a_{s,t})^i(k)}(w^a_{l,p+1+q})^r(k) - \frac{\partial v_q(k-1)}{\partial (w^a_{s,t})^i(k)}(w^a_{l,p+1+q})^i(k)\right)\right] - \delta_{sl}(I^a_n)^i(k)$$

$$\frac{\partial \tau^a_l(k)}{\partial (w^a_{s,t})^i(k)} = \left[\sum_{q=1}^{N}\left(\frac{\partial v_q(k-1)}{\partial (w^a_{s,t})^i(k)}(w^a_{l,p+1+q})^r(k) + \frac{\partial u_q(k-1)}{\partial (w^a_{s,t})^i(k)}(w^a_{l,p+1+q})^i(k)\right)\right] + \delta_{sl}(I^a_n)^r(k).$$

By using the Cauchy–Riemann equations $\Lambda^{l,(rr)}_{s,t}(k) = \Lambda^{l,(ii)}_{s,t}(k)$ and $\Lambda^{l,(ir)}_{s,t}(k) = -\Lambda^{l,(ri)}_{s,t}(k)$, a more compact representation of gradient $\nabla_{w^a_{s,t}}E(k)$ is given by

$$\nabla_{w^a_{s,t}}E(k) = \sum_{l=1}^{N} e_l(k)\left(\Lambda^l_{s,t}\right)^*(k). \tag{17}$$

Extending the approach from Goh and Mandic (2004) by taking into account the augmented complex statistics, the update for the $N \times 2(p+N+1)$ matrix of sensitivities $(\Lambda^l_{s,t})^*(k)$ can be derived as

$$\left(\Lambda^l_{s,t}\right)^*(k) = \{\Phi'(\mathrm{net}^a_l(k))\}^* \times \left[(\mathbf{w}^a_l(k))^H \boldsymbol{\Lambda}^*(k-1) + \delta_{sl}(I^a_t(k))^*\right]. \tag{18}$$

Finally, the update of the augmented weights within the RNN can be expressed as

$$\mathbf{w}^a_l(k) = \mathbf{w}^a_l(k-1) + \eta\sum_{l=1}^{N} e_l(k)\left(\Lambda^l_{s,t}\right)^*(k) \tag{19}$$

which completes the derivation of the augmented real time recurrent learning (ACRTRL) algorithm.

3. Simulations

For the experiments, the nonlinearities within neurons were 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 (Fig. 1) consisted of $N = 3$ neurons with the tap input length of $p = 5$. Two sets of simulations were conducted, based on complex-valued nonlinear one step ahead prediction. In the first set of experiments, comprehensive statistical tests comparing ACRTRL and CRTRL were performed on benchmark complex coloured and nonlinear signals (learning curves produced by averaging of 100 iterations of independent trials). In the second set of simulations, single-trial experiments were performed on real-world complex-valued wind[6] and radar[7] data.

The benchmark linear input (coloured noise) used in simulations was an autoregressive $AR(4)$ complex process, given by

$$r(k) = 1.79r(k-1) - 1.85r(k-2) + 1.27r(k-3) - 0.41r(k-4) + n(k) \tag{20}$$

whose coefficients were chosen so that the process was stable. The driving input to (20) was complex white Gaussian noise (CWGN) denoted by $n(k)$, with zero mean and unit variance.

The nonlinear input considered was generated using a benchmark feedback process given by Mandic and Chambers (2001)

$$z(k) = \frac{z^2(k-1)(z(k-1)+2.5)}{1+z(k-1)+z^2(k-2)} + n(k-1). \tag{21}$$

Table 1 shows the statistical properties of the datasets considered. The measurement used to assess the performance

[6] The wind profile data used in simulations are publicly available from "http://mesonet.agron.iastate.edu/". The complex representation of the wind vector $\mathbf{v}(t)$ is based on the joint modelling of the wind speed $v(t)$ and direction $\theta(t)$, given by $\mathbf{v}(t) = v(t)e^{\mathrm{j}\theta(t)}$ or alternatively by using its projection onto the North (N) – East (E) coordinate system, given by $\mathbf{v}(t) = v_E(t) + \mathrm{j}v_N(t)$.

[7] Radar (high) is referred to as "high sea state data" and radar (low) is referred to as "low sea state data". Publicly available from "http://soma.ece.mcmaster.ca/ipix/".

Table 2
Comparison of prediction gains R_p for the various classes of signals

R_p (dB)	CRTRL	ACRTRL
$AR(4)$ (20)	3.22	4.10
Nonlinear (21)	4.52	5.81
Radar (low)	11.40	13.57
Radar (high)	4.56	5.41
Wind	12.78	13.82

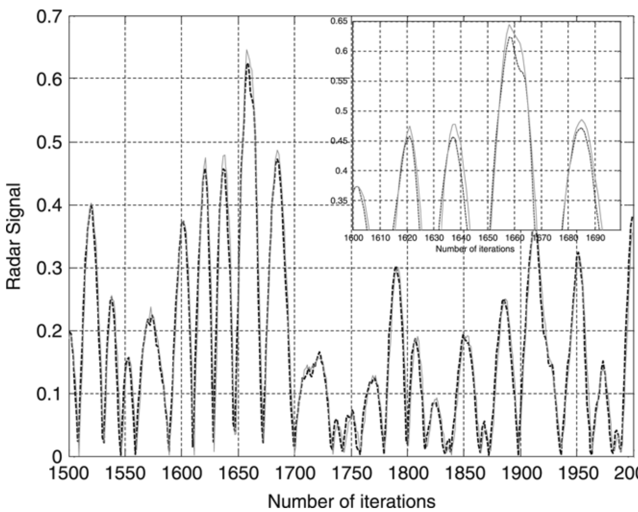

Fig. 2. Prediction performance for radar signal using the proposed ACRTRL. Solid curve: actual signal. Dashed curve: predicted signal.

was the prediction gain $R_p(k) \triangleq 10\log_{10}\left(\frac{\sigma_x^2}{\hat{\sigma}_e^2}\right)$ [dB] where σ_x^2 denotes the variance of the input signal $x(k)$, whereas $\hat{\sigma}_e^2$ denotes the estimated variance of the forward prediction error $e(k)$.

Table 2 shows a comparison of the prediction gains R_p (dB) between the proposed ACRTRL and standard CRTRL (without the augmented states) for the classes of signals considered. In all the cases, there was a significant improvement in the prediction gain when the ACTRL approach was employed over the performance of the standard CRTRL algorithm.

To further illustrate the advantage of using the ACRTRL over CRTRL, we compared the performances of FCRNNs trained with these algorithms in experiments on complex-valued single trial real world radar and wind data. Figs. 2 and 4 show subsegments of the predictions generated by the ACRTRL for radar (low) and wind signals, which illustrate the desired property of the proposed approach to adapt to the large changes in the dynamics of the processed signal. Compared with the performance of standard CRTRL in Figs. 3 and 5, the proposed ACRTRL algorithm was more stable, and has exhibited higher accuracy and more consistent performance.

4. Conclusions

The augmented complex-valued real-time recurrent learning (ACRTRL) algorithm has been introduced for nonlinear adaptive filtering in the complex domain. The ACRTRL has been derived using some recent advances in so-called augmented complex statistics, whereby a complete second-order

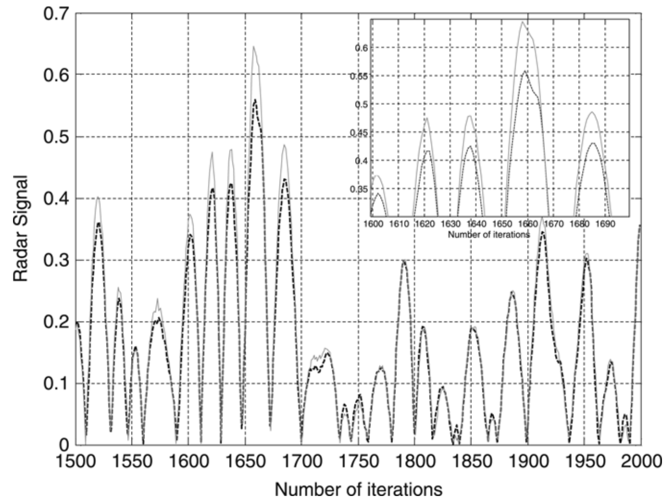

Fig. 3. Prediction performance for radar signal using the standard CRTRL. Solid curve: actual signal. Dashed curve: predicted signal.

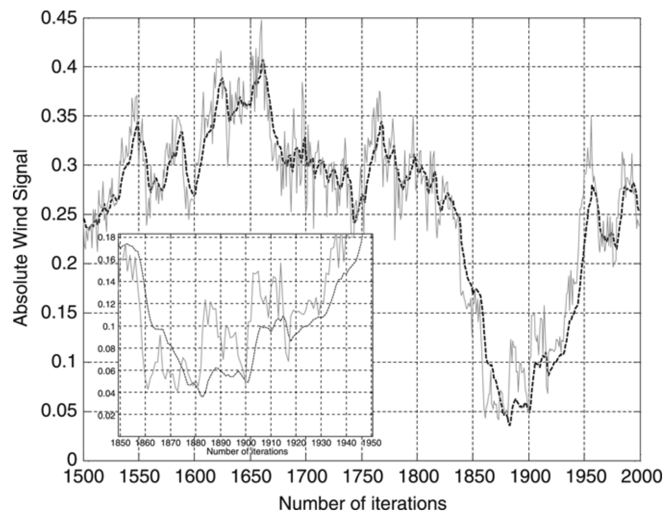

Fig. 4. Prediction performance for wind signal using the proposed ACRTRL. Solid curve: actual signal. Dashed curve: predicted signal.

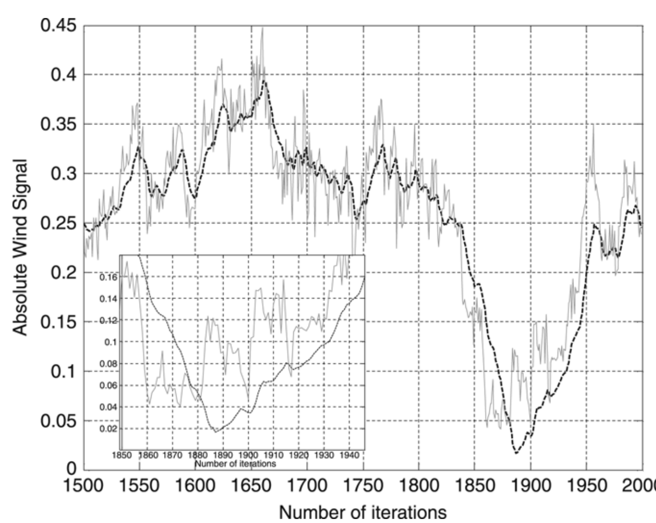

Fig. 5. Prediction performance for wind signal using the standard CRTRL. Solid curve: actual signal. Dashed curve: predicted signal.

information from complex-valued random processes is taken into account. The performance of the ACRTRL has been evaluated on benchmark complex-valued nonlinear input signals, and also on real-life complex-valued wind signals. Simulation results have justified the potential of ACRTRL in nonlinear complex-valued neural adaptive filtering applications.

References

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*, 179–211.

Gautama, T., Mandic, D. P., & Hulle, M. M. V. (2003). A non-parametric test for detecting the complex-valued nature of time series. In: *Knowledge-based intelligent information and engineering systems: 7th international conference* (pp. 1364–1371) *vol. 2773*.

Goh, S. L., & Mandic, D. P. (2004). A complex-valued RTRL algorithm for recurrent neural networks. *Neural Computation*, *16*(12), 2699–2713.

Goh, S. L., & Mandic, D. P. (2007). An augmented extended Kalman filter algorithm for complex-valued recurrent neural networks. *Neural Computation*, *19*, 1039–1055.

Kim, T., & Adali, T. (2003). Approximation by fully complex multilayer perceptrons. *Neural Computation*, *15*(7), 1641–1666.

Mandic, D. P., & Chambers, J. A. (2001). *Recurrent neural networks for prediction: Learning algorithms, architectures and stability*. John Wiley & Sons.

Medsker, L. R., & Jain, L. (2000). *The CRC Press International Series on Computational Intelligence*, *Recurrent neural networks: Design and applications*.

Neeser, F., & Massey, J. (1992). Proper complex random processes with applications to information theory. *IEEE Transactions on Information Theory*, *39*(4), 1293–1302.

Principe, J. C., Euliano, N. R., & Lefebvre, W. C. (2000). *Neural and adaptive systems: Fundamentals through simulations*. Wiley.

Schreier, P. J., & Scharf, L. L. (2003). Second-order analysis of improper complex random vectors and processes. *IEEE Transactions on Signal Processing*, *51*(3), 714–725.

Widrow, B., McCool, J., & Ball, M. (1975). The complex LMS algorithm. *Proceedings of the IEEE*, *63*, 712–720.

Williams, R. J., & Zipser, D. A. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, *1*(2), 270–280.