ARTICLE IN PRESS

Neurocomputing xx (xxxx) xxxx–xxxx

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Complex-valued neural network topology and learning applied for identification and control of nonlinear systems

Ieroham S. Baruch, Victor A. Quintana, Edmundo P. Reynaud*

Department of Automatic Control, CINVESTAV-IPN, D.F, Mexico

ARTICLE INFO

Keywords:
Complex-valued backpropagation
Complex-valued Levenberg-Marquard
Adaptive neural control
Recurrent complex-valued neural network
Nonlinear oscillatory plants

ABSTRACT

In this paper we present a Complex-Valued Recurrent Neural Network (CVRNN), trained with a recursive Levenberg-Marquardt (LM) learning algorithm in the complex domain, applying it to the problem of dynamic system identification, and to an adaptive neural control scheme of a nonlinear oscillatory plant. This methodology is applied to two different CVRNN topologies with different kinds of activation functions. Furthermore, we applied the CVRNN identification and control for a particular case of a nonlinear, oscillatory mechanical plant to validate the performance of the adaptive neural controller using the LM algorithm developed throughout this work, compared to a complex-valued Backpropagation learning algorithm. The obtained comparative simulation results using a flexible robot arm gives a good performance of the derived control schemes. The results show some priority of the recursive LM learning over the BP learning, and the use of constructed activation functions in the neural network topology.

1. Introduction

The fast growth of available computational resources has led to the developments of a wide number of Neural Networks (**NN**) based modeling, identification, prediction and control applications [1]. The main **NN** property, namely the ability to approximate complex nonlinear relationships without prior knowledge of the model structure makes them a very attractive alternative to the classical modeling and control techniques [2–4]. In the last decade there has been a rise in applications using Complex-Valued Recurrent **NN**s (**CVRNN**). Most of them deal with oscillatory systems which, by their physical nature, are convenient to be treated in the complex domain, such as electromagnetic waves, light waves, images processing, electric power systems, evaporator systems and mechanical systems [5,6].

Recently there has been applications of these complex-valued networks in the solution of math problems: in [7] the authors use a specific type of **RNN** to solve the Sylvester equation, a problem that commonly arises in control theory; in [8,9], different complex-valued **NN** topologies are used to solve the time-varying complex generalized inverse matrix problem, showing that the use of complex-valued **NN** yield better performance than its real-valued counterpart.

Also there is a vast use of **NN** in control applications. To name a few examples: in [10] the authors use **NN** in two adaptive systems for the control of micro-aerial vehicles with nonlinear dynamics; and in [11], another adaptive controller is proposed for controlling mobile robots in the presence of external disturbances using **NN**, yielding good results.

In [12], the authors derived a Complex-Valued Backpropagation (**CVBP**) algorithm used for pattern classification. However, the learning algorithm presented some problems because the activation functions have singularity points in their domains. Some other works [12,13], propose different activation functions that avoid singularity points. To simplify the Backpropagation (**BP**) and the Levenberg-Marquardt (**LM**), learning for the **CVRNN**, the present work uses the aid of diagrammatic rules (see [14,15]) to construct an adjoint network and propagate the complex output error through it in order to obtain the weight adjustment, with two different **CVRNN** topologies considered, each with different kinds of activation functions, to avoid singularities. The optimization based **BP** and **LM** learning techniques are used for nonlinear oscillatory plant identification and tracking error suppression by means of a direct integral term (I-term) adaptive neural control using **CVRNN**. Lastly, some comparative simulation results of **CVRNN** identification and control of the flexible-joint of a robot are given and discussed, and a validation stage is presented in order to confirm the good performance of the control scheme using the proposed learning algorithms.

Although good performance of this kind of **CVRNN** topology has been achieved previously using **BP** (see [16]), the importance of this kind of neural adaptive controllers in real-time practical applications motivates us in the development of algorithms that drive the **CVRNN** behavior to the desired plant behavior in a much faster way, such that

* Corresponding author.
E-mail addresses: baruch@ctrl.cinvestav.mx (I.S. Baruch), varellano@ctrl.cinvestav.mx (V.A. Quintana), eperez@ctrl.cinvestav.mx (E.P. Reynaud).

http://dx.doi.org/10.1016/j.neucom.2016.09.109
Received 11 December 2015; Received in revised form 13 September 2016; Accepted 19 September 2016
Available online xxxx
0925-2312/

the training stage does not affect the global performance of the control system.

This work contributes a training algorithm that resembles the second order Newton optimization algorithm, with a faster behavior than a first order algorithm, but without the high computational cost or singularity problems computing the inverse Hessian matrix that it often presents. We apply this algorithm to the proposed **NN** with complex-valued synaptic weights, and then it is compared to the **BP** algorithm developed in previous works [16].

The **CVRNN** is presented as a dynamical system with a canonical Jordan form. While the use of real-valued synaptic weights yields a sum of purely exponential terms at the **CVRNN**'s output, the use of complex numbers adds oscillatory terms to the **CVRNN**'s output, expanding the kind of behaviors it can approximate. This is why a **CVRNN** with complex-valued parameters is preferred over the **CVRNN** with double number of real-valued parameters.

2. Topology, BP and LM learning of CVRNN

The general **CVRNN** topology in consideration is an extension of the Real-Valued Recurrent Neural Network topology, given in [4]. The considered **CVRNN** topology has real-valued input $U(k)$ and output $Y(k)$ signals, complex-valued internal state $X(k)$ and hidden state $Z(k)$ vectors, and complex-valued J, B, C weight matrices, defined as follows:

$$\boldsymbol{X(k+1)=JX(k)+BU(k)} \tag{1}$$

$$\boldsymbol{J=\mathrm{diag}(J_i), \quad |J_i|<1, \quad i=1,\ldots,N} \tag{2}$$

$$\boldsymbol{E(k)=Y_P(k)-Y(k)} \tag{3}$$

$$\boldsymbol{Z(k)=\Gamma[X(k)]} \tag{4}$$

$$\boldsymbol{V(k)=CZ(k)} \tag{5}$$

$$\boldsymbol{Y(k)=\Phi[V(k)]} \tag{6}$$

The vectors and matrices dimensions of the **CVRNN** topology are given by: $J\in\mathbb{C}^{n\times n}$ the feedback weight matrix, $B\in\mathbb{C}^{n\times m}$ the input weight matrix, $C\in\mathbb{C}^{L\times n}$ the output weight matrix, $X, Z\in\mathbb{C}^{n}$, $U\in\mathbb{R}^{m}$ and $Y\in\mathbb{R}^{L}$; $\Gamma[\cdot]$ is a complex-valued activation function, and $\Phi[\cdot]$ is a real-valued activation function; n, m, L are the number of internal states, inputs and outputs respectively.

The inequality in (2) is a local stability preserving condition, imposed on the diagonal blocks of the matrix J. The performance index to be minimized is given by:

$$\zeta(k)=\frac{1}{2}\sum_j [E_j(k)]^2, \quad \zeta=\frac{1}{N_e}\sum_k \zeta(k) \tag{7}$$

The instantaneous Means Squared Error (**MSE**) $\zeta(k)$ is used in real-time implementations, while the total **MSE**ζ is used for one epoch N_e for off-line implementations. Given that (7) is a real-valued function which has to be minimized in regard to complex-valued weight parameters, the optimality conditions are given in terms of the Wirtinger calculus [17].

We consider two particular **CVRNN** with different activation functions. As in the real-valued case [4], we apply complex-valued diagrammatic rules to derive an adjoint network and weight update terms for each case.

2.1. Topology, BP and LM learning of CVRNN with first type activation function

The first type activation function, which is used for $\Gamma[\cdot]$, $\Phi[\cdot]$ is defined as follows:

$$f(z)=\tanh(z),\ z\in\mathbb{C}\setminus\left\{zz=0\pm\frac{2p-1}{2}\pi i,\quad \forall p\in\mathbb{N}\right\} \tag{8}$$

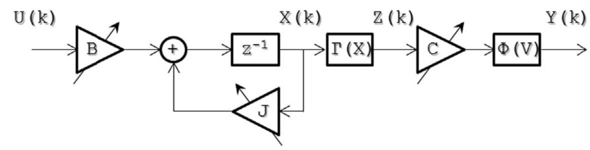

Fig. 1. Topology of the first type CVRNN.

This activation function has singularities at periodic points of the complex plane; we avoid them by capping the image of the function to a threshold, in a neighborhood around these points, of radius $\varepsilon>0$. The topology of **CVRNN** using this activation function is given on Fig. 1.

The application of the diagrammatic rules given in [15] consists in reversing the signal flow of every branch, interchanging inputs with outputs, delay operators with forward operators, sum points with junctions, vectors with their transpose, and activation functions with their derivatives. For the **CVRNN** topology, given on Fig. 1, the adjoint **CVRNN** topology is given by Fig. 2, where the gradient terms can be easily derived.

The general complex-valued **BP** (**CVBP**) learning algorithm (see [16]) with a momentum term is given by the following update equation:

$$W(k+1)=W(k)+\eta\Delta W(k)+\alpha\Delta W(k-1)|W_j|<W_0 \tag{9}$$

Where: W is a general weight matrix (J, B, C); ΔW is the weight update term of W given by the gradient term of the output with respect to each one of the weight vectors; $\eta>0$ is a diagonal constant learning rate matrix, $\alpha>0$ is a diagonal constant momentum term matrix and W_0 is a restricted region for each weight W_j.

The complex-valued Levenberg-Marquardt (**CVLM**) algorithm for any weight vector W is described by the following update equation:

$$W(k+1)=W(k)+P(k)DY[W(k)]E(k)|W_j|<W_0 \tag{10}$$

The gradient terms for the complex-valued network with the first type activation function are described by the following equations:

$$D_1(k)=\Phi'[Y(k)] \tag{11}$$

$$D_2(k)=\Gamma'[Z(k)]C^{*}D_1(k) \tag{12}$$

$$DY[C(k)]:=\frac{\partial Y(k)}{\partial C(k)}=D_1(k)Z^{*}(k) \tag{13}$$

$$DY[J(k)]:=\frac{\partial Y(k)}{\partial J(k)}=D_2(k)X^{*}(k) \tag{14}$$

$$DY[B(k)]:=\frac{\partial Y(k)}{\partial B(k)}=D_2(k)U^{*}(k) \tag{15}$$

Where the (*) superscript denotes a complex conjugate and transposed vector. For the derivation of the **CVBP** learning algorithm we use $D(k)=E(k)$ and the weight update learning rule (9). For the derivation of the **CVLM** learning algorithm we use $D(k)=I$, where I is an identity matrix input for the adjoint topology, and learning rule (10).

The matrix P of the **CVLM** learning algorithm is an approximation to the inverse Hessian matrix used in the second order Newton optimization algorithm, and it is computed recursively using the following Riccati difference equation [18]:

$$P(k)=\alpha^{-1}[P(k-1)-P(k-1)\ldots\ldots\Omega_{W(k)}S_{W(k)}^{-1}\Omega_{W(k)}^{*}P(k-1)] \tag{16}$$

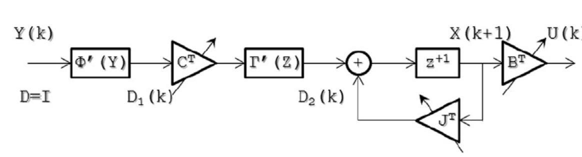

Fig. 2. Adjoint topology of the first type CVRNN.