Identification of dynamical systems using recurrent complex-valued neural networks

Victor M. Arellano-Quintana and Ieroham S. Baruch

Abstract—The existing in the literature backpropagation algorithms for training Complex-Valued Neural Networks (CVNN) are very complicated, especially where we deal with a Recurrent CVNN (RCVNN). For that reason the paper proposed to use diagrammatic rules so to construct an adjoined RCVNN and propagate the complex output error through it so to perform the weight adjustment. Two different RCVNN topologies with different activation functions avoiding singularity are considered and their CV backpropagation learning algorithms are derived using the proposed learning methodology. Finally, some comparative simulation results of RCVNN identification of flexible-joint robot are given and discussed, and then a validation stage is presented in order to confirm the good quality of the proposed learning methodology.

Keywords—Diagrammatic rules, learning algorithms, recurrent complex-valued neural networks, systems identification

I. INTRODUCTION

Until now there are few applications using Recurrent Complex-Valued Neural Networks (RCVNN). Most of them deal with oscillatory systems which by their physical nature it is convenient to be treated in the complex domain, such as electromagnetic waves, light waves, images processing, electric power systems etc. (see [1], [2], [3]). In [2] the authors apply a special type of a RCVNN for modeling of power transformer, obtaining good results. The drawback here is that nothing is mentioned about the presence of singularity points due to the activation function.

In the field of mechanical systems identification the RCVNN have had a minor presence. In spite of that, some papers like [3], [4] proposed to use RCVNN for mechanical plants identification and control, obtaining good results. In [3] the authors applied a CVNN for an industrial evaporator system identification using an evolutionary algorithm to design the network. They use also radial basis functions NN avoiding the gradient terms computation in the learning algorithm. Other papers like [4], [5] used CVNN for this kind of systems, obtaining satisfactory results.

In [6], a type of RCVNN is used for modeling of Lorenz system. In [7], Leung and Haykin derived a Complex Value Backpropagation (CVBP) algorithm used for pattern classification. However, this learning algorithm presented some problems because of the activation function singularity. For that reason some authors (see [8]-[11]) proposed different activation functions that avoid activation function singularity.

So, to simplify the backpropagation learning of the RCVNN, the present paper proposed to use diagrammatic rules so to construct an adjoined RCVNN and propagate the complex output error through it so to obtain the weight adjustment. Two different RCVNN topologies with different activation functions avoiding singularity are considered and their CV backpropagation learning algorithms are derived using the diagrammatic rules and the constructed by them adjoined RCVNN. Finally, some comparative simulation results of RCVNN identification of flexible-joint robot are given and discussed, and then a validation stage is presented in order to confirm the good quality of the proposed learning methodology.

II. TOPOLOGY AND BACKPROPAGATION LEARNING OF REAL-VALUED RECURRENT NEURAL NETWORK

In Fig.1 it is shown the block-diagram of a Real Value Recurrent Neural Network (RVRNN) with Jordan Canonical topology, [12]. The canonical RNN topology possesses the controllability, observability, reachability and identifiability-properties, and minimum number of parameters, subject to learning. The RVRNN model is described by the following equations:

\[ X(k+1) = JX(k) + BU(k) \]  
\[ J = \text{block-diag}(J_i); |J_i| < 1, i = 1, \ldots, N \]  
\[ E(k) = Y_p(k) - Y(k) \]  
\[ Z(k) = \Gamma[X(k)] \]  
\[ Y(k) = \Phi[CZ(k)] \]

Where: \( X(\cdot) \in \mathbb{R}^N \) - State vector; \( U(\cdot) \in \mathbb{R}^M \) - Input vector; \( Y(\cdot) \in \mathbb{R}^L \) - Output vector; \( Z(\cdot) \in \mathbb{R}^L \) - Output vector of the hidden layer; \( \Gamma(\cdot) \text{, } \Phi(\cdot) \) - Vector valued activation functions with compatible dimensions; \( J \) - Weight state diagonal matrix with elements \( J_i \). The inequality in (1) is a stability preserving condition, imposed on the weights \( J_i \); Band C are weight input and output matrices with compatible dimensions. The RVRNN...
model is completely parallel, capable to issue weights $J, B, C$, and states $X$.

Applying the diagrammatic rules, [13], to the RVRNN topology, given on Fig.1 we could obtain the adjoined RVRNN model, given on Fig.2. The adjoined RVRNN is used for the backward pass of the backpropagation algorithm to pass the output error through it so to train RVRNN weights. The performance index to be minimized is given by:

$$
\zeta(k) = \frac{1}{2} \sum_j [E_j(k)]^2, \quad j \in C, \quad \zeta = \frac{1}{N_e} \sum_j \zeta(k) \quad (5)
$$

Where the instantaneous Means Squared Error (MSE) $\zeta(k)$ is minimized in real-time applications and the total MSE $\zeta$ is minimized for one epoch $N_e$ in off-line applications. The general RVRNN real-time backpropagation learning algorithm with momentum term is given by the following vector-matricial equation:

$$
W(k+1) = W(k) + \eta \Delta W(k) + \alpha \Delta W(k-1)
$$

$$
|W_{ij}| < W_0
$$

Where: $W(\cdot)$ is a general weight matrix (in fact $J, B, C$); $\Delta W(\cdot)$ is the modification of $\Delta W(\cdot)$; $\eta$ is a diagonal constant matrix of learning; $\alpha$ is a diagonal momentum term matrix; $W_o$ is a restricted region for the weight $W_{ij}$. Using the specified in Fig.2 errors and the obtained in the forward pass (1)-(3) intermediate vectors, we could obtain the following weight update algorithm for the matrices $J, B, C$.

For the output layer:

$$
\Delta C(k) = E_1(k)Z^T(k) \quad (7)
$$

$$
E_1(k) = \Phi'[Y(k)]E(k) \quad (8)
$$

$$
E(k) = Y_p(k) - Y(k) \quad (9)
$$

For the hidden layer:

$$
\Delta J(k) = E_3(k)X^T(k) \quad (10)
$$

$$
E_3(k) = \Gamma'[Z(k)]E_2(k) \quad (11)
$$

$$
E_2(k) = C^T(k)E_1(k) \quad (12)
$$

$$
\Delta \nu(k) = E_3(k) \otimes X(k) \quad (13)
$$

$$
\Delta B(k) = E_3(k)U^T(k) \quad (14)
$$

Where: $\Delta C, \Delta J, \Delta B$ are weight corrections of $C, J, B$; the output error $E$ is given by (9) and passed through the adjoined RVRNN so to obtain the intermediate error vectors $E_1, E_2, E_3$; $\Phi', \Gamma'$ are derivatives of the respective activation functions expressed with respect to its correspondent outputs. The equation (13) gives the learning solution for the diagonal matrix $J$ as a dot product of two vectors. Equations (7)-(14) represented the complete backpropagation learning algorithm for RVRNN.

This solution could be applied for any NN topology and does not require computations of partial derivatives. Based on the diagrammatic rules we could derive a similar learning solution for the complex value case.

$$
U(k) \quad X(k) \quad Z(k) \quad V(k) \quad Y(k)
$$

$$
\begin{align*}
E(k) & \quad E_1(k) \quad E_2(k) \quad E_3(k) \\
\Phi(k) & \quad C^T \quad \Gamma(k) \quad X(k) \quad U(k) \\
\end{align*}
$$

Fig. 1 Block-diagram of the RVRNN topology

Fig. 2 Block-diagram of the adjoined RVRNN topology, obtained from the RVRNN topology using the diagrammatic rules

III. TOPOLOGY AND BACKPROPAGATION LEARNING OF RECURRENT COMPLEX-VALUED NEURAL NETWORK

The general Complex Valued Neural Network topology below consideration has complex valued input, output, and state vectors, and complex $A, B, C$ weight matrices. In this part we consider two particular Recurrent CVNNs (RCVNNs) with different activation functions. In the same manner as in the real value case we could apply complex valued diagrammatic rules so to derive an adjoined RCVNN for each CVNN case.

The performance index to be minimized is given by:

$$
\zeta(k) = \frac{1}{2} \sum_j [E_j(k)][E_j^*(k)], \quad j \in C, \quad \zeta = \frac{1}{N_e} \sum_j \zeta(k) \quad (15)
$$

This function $\zeta(k)$ is a mapping of the form $f: \mathbb{C} \rightarrow \mathbb{R}$, so it is not analytic in the sense that it does not have derivative and also it does not satisfy the Cauchy-Riemann equations. This complicates the use of the gradient descendent algorithm, because we have to use the so-called Wirtinger’s calculus. Using diagrammatic rules we avoid this complicated problem.

A. RCVNN Topology with First Type Activation Function

The first activation function that we consider is given by (16). This activation function has singularities in some parts of the complex domain and because of that, this function is evaluated in a region of the complex plane avoiding singularity points, [14]. The activation function is defined as:

$$
f(z) = \tanh z, \quad z \in \mathbb{C} \setminus \left[ -\frac{\pi}{2} - \epsilon; -\frac{\pi}{2} + \epsilon \right] \cup \left[ -\epsilon; \epsilon \right] \quad (16)
$$

The NN using this activation function is described in the given in Fig.3 block diagram. This type of representation allows us to apply diagrammatic rules so to derive the adjoined RCVNN and use it for NN learning. In the present
paper we propose a new approach of the CVBP for the given in Fig. 3 RCVNN extending the real-time real value NN algorithm of learning, [12], to the complex value case. The topology of this RCVNN is given on Fig.3.

The mathematical description of that topology is the same as (1)-(4) for the real valued case but the variables are complex. The vectors and matrices of the RCVNN topology are given as follows:

\[ A \in \mathbb{C}^{n \times n}; \text{ Feedback Matrix;} \]
\[ B \in \mathbb{C}^{n \times n}; \text{ Input matrix;} \]
\[ C \in \mathbb{C}^{n \times n}; \text{ Output matrix;} \]
\[ X(k) \in \mathbb{C}^{n \times n}; \text{ State vector;} \]
\[ U(k) \in \mathbb{C}^{n \times n}; \text{ Network input;} \]
\[ Y(k) \in \mathbb{C}^{n \times n}; \text{ Network output;} \]
\[ G[\cdot], S[\cdot]: \text{ Complex-valued vector-tanh - activation functions, given by (16)}; \]
\[ m: \text{ Number of inputs;} \]
\[ n: \text{ Number of neurons in the hidden layer;} \]
\[ p: \text{ Number of neurons in the output layer.} \]

The complex state feedback matrix \( A \) is defined as diagonal with the same weight restriction as the given in (1) for \( J \).

As we could see the used \( \text{tanh} \) - activation function has singularity in some parts of the complex domain. To overcome that, the activation function is evaluated in regions outside these points, as it could be seen in (16).

Applying the complex valued diagrammatic rules we could obtain the adjoined RCVNN, given on Fig.4.

\[
\Delta C(k) = E_c(k)Z^*(k) \tag{17}
\]
\[
E_c(k) = S'[Y[V^*(k)][k]]E(k) \tag{18}
\]

For the hidden layer:

\[
\Delta A(k) = E_3(k)X^*(k) \tag{20}
\]
\[
E_3(k) = G'[Z[X^*(k)](k)]E_2(k) \tag{21}
\]
\[
E_2(k) = C'(k)E_1(k) \tag{22}
\]
\[
\Delta vA(k) = E_3(k) \otimes X^*(k) \tag{23}
\]
\[
\Delta B(k) = E_3(k)U^*(k) \tag{24}
\]

Where:

\[ G[\cdot], S[\cdot]: \text{ Derivatives of the activation functions G, S;} \]
\[ a^*: \text{ Transpose and conjugate of the complex number a;} \]
\[ T(k) : \text{ Desired output vector.} \]

As it could be seen, the application of the diagrammatic rules and the adjoined RCVNN topology simplified the learning with respect to the classical gradient descent learning in complex domain, [7].

**B. RCVNN Topology with Second Type Activation Function**

The second type activation function, [9], does not have singularity points. It is given by the next equation:

\[
f(z) = \tanh \text{Re}(z) + i \tanh \text{Im}(z) \tag{25}
\]

Unlike the RVNN, the topology of the CVNN will be defined by the second type activation function. In this case, the RCVNN is given by the following equations:

\[
X(k + 1) = AX(k) + BU(k) \tag{26}
\]
\[
Z(k) = G[X \Re (k)] + iG[X \Im (k)] \tag{27}
\]
\[
V(k) = Z \Re G \Re + iZ \Im G \Im \tag{28}
\]
\[
Y(k) = S[V \Re (k)] + iS[V \Im (k)] \tag{29}
\]

The topology of the RCVNN in this case is given on Fig.5.

\[
E(k) = T(k) - Y(k) \tag{19}
\]

From Fig.5 and equation (25) it is clear that this topology separates the real and imaginary parts of the CV activation functions in order to simplify the CV learning algorithm.

The vectors and matrices of the RCVNN topology are defined as follows:

\[ G[\cdot], S[\cdot]: \text{ Complex valued vector-tanh-activation functions given by (25).} \]
A \in \mathbb{C}^{n \times n}$: Feedback Matrix;
$B \in \mathbb{C}^{n \times m}$: Input matrix;
$C \in \mathbb{C}^{m \times n}$: Output matrix.

$X(k) \in \mathbb{C}^{n \times 1}$: State vector;
$U(k) \in \mathbb{C}^{n \times 1}$: Network input;
$Y(k) \in \mathbb{C}^{n \times 1}$: Network output;
$m$: Number of inputs;
$n$: Number of neurons in the hidden layer;
$p$: Number of neurons in the output layer.

The complex state feedback matrix $A$ is defined as diagonal with the same restriction as (1) for $J$.

Applying the complex valued diagrammatic rules we could obtain the adjoined RCVNN, given on Fig.6.

Fig. 6 Adjoined topology of the second type RCVNN

Now, the backpropagation learning rule (6) could be defined in a complex domain with the same significance of the participating variables. Using the adjoined RCVNN topology we could derive the following weight update algorithm:

For the output layer:

$$\Delta C(k) = E_1(k)Z^*(k)$$

(30)

$$E_1(k) = E_{Re}S'[Y[V_{Re}(k)]] + iE_{Im}S'[Y[V_{Im}(k)]]$$

(31)

$$E(k) = T(k) - Y(k)$$

(32)

For the hidden layer:

$$\Delta A(k) = E_3(k)X^*(k) - 1$$

(33)

$$E_3(k) = E_{Re}G[Z[X_{Re}(k)]] + iE_{Im}G[Z[X_{Im}(k)]]$$

(34)

$$E_2(k) = C^*(k)E_1(k)$$

(35)

$$\Delta v A(k) = E_3(k) \otimes X^*(k)$$

(36)

$$\Delta B(k) = E_3(k)U^*(k)$$

(37)

Where:

$G[\cdot], S[\cdot]$: The derivatives of the activation functions $G, S$;
$a^*$: Transpose and conjugate of the complex number $a$;
$T(k)$: Desired output vector;
$E_{Re}$: Real part of $E(k)$;
$E_{Im}$: Imaginary part of $E(k)$.

As it could be seen, the application of the complex diagrammatic rules in the design of the adjoined RCVNN topology simplified the learning with respect to the classical gradient descent learning in complex domain.

IV. Complex-Valued Neural Solution of Nonlinear Identification Problem

This part of the paper illustrates the application of the RCVNN for nonlinear oscillatory plant identification. The nonlinear oscillatory plant model under investigation is a flexible – joint robot arm. The model, the output and the input of the plant are given in continuous time. In order to use a recurrent neural network for its identification, the output/input signals of the plant are discretized with sampling period $T_0$.

A. Description of the Nonlinear Plant Model

The identified system is an idealized nonlinear model of a flexible – joint robot arm, illustrated by Fig. 7. The flexibility of the robot joint is caused by a harmonic drive, which is a type of robot gear mechanism with high torque transmission, low backlash and compact size.

Fig. 7 Idealized model representing robot joint flexibility

The robot joint model consists of an actuator connected to a load through a torsional spring representing the joint flexibility. We take the motor torque as an input $u$. The equations of motion of the flexible - joint robot are given as follows.

\begin{equation}
J_1 \ddot{\theta}_1 + B_1 \dot{\theta}_1 + Mg \sin \theta_1 + k(\theta_1 - \theta_m) = 0
\end{equation}

\begin{equation}
J_m \ddot{\theta}_m + B_m \dot{\theta}_m - k(\theta_1 - \theta_m) = u
\end{equation}

(378)

Where: $J_1, J_m$ are load and motor inertias; $B_1$ and $B_m$ are load and motor damping constants; $u$ is the input torque applied to the motor shaft; $M$ and $L$ are the mass of the link and the length between the shaft and the center of mass of the link; $k$ represents the torsional stiffness constant of the harmonic drive gear.

As we can see, the plant is an oscillatory system, described by two second order differential equations, representing a system with two degrees of freedom but only one input which makes the system sub-actuated.

B. Plant Identification

The plant identification using the given up two RCVNN models is illustrated by Fig.8. Here the desired complex target vector is the output of plant. The identification objective is that the complex weight parameters of the RCVNN are adjusted in such manner that the RCVNN output follows the plant output with minimum MSE.
The RCVNN used has three neurons in the hidden layer, one input and one output neuron. The RCVNN dimensions are as follows: \( m = 1; \ p = 1; \ n = 3 \) and a sampling time \( T_0 = 0.01 \).

Fig. 8 Block diagram of RCVNN plant identification

C. Simulation Results

This section described the simulation results obtained using both type RCVNNs for nonlinear oscillatory plant identification. The simulation is done in two stages: stage of system identification and stage of NN generalization. During the first stage the RCVNN weights are learned until convergence and after that they are frozen and the input signal is changed so to validate (generalized) the result of identification. As a measure of comparison we use the final MSE\( \% \) of identification and generalization. The input plant signals used for system identification \( u(t) \) and generalization \( u_g(t) \) are given by:

\[
\begin{align*}
    u(t) &= \sin \left( \frac{t}{10} \right) + 0.5 \sin \left( \frac{2t}{50} \right) \\
    u_g(t) &= 0.5 \sin \left( \frac{t}{10} \right) + 0.8 \sin \left( \frac{t}{30} \right)
\end{align*}
\]

Simulation results obtained with the first type activation function RCVNN-1 topology. The graphical results of plant identification for the first case are given on Fig. 9 where the output of the plant is compared with the output of the RCVNN-1. The final MSE of RCVNN-1 convergence during system identification is given on Table 1 for 1000, 2000, and 3000 learning iterations. The results show a constant MSE decreasing exhibiting a good BP NN convergence.

The graphical results of RCVNN-1 generalization are given on Fig. 10, where the NN weights are frozen and input signal is changed. The output of the plant is compared with the output of the RCVNN-1 exhibiting a good generalization. The final results of RCVNN-1 generalization are given on Table 1 for 3000 steps. The validation results show a constant MSE decreasing exhibiting a good RCVNN-1 generalization.

Simulation results obtained with the second type activation function RCVNN-2 topology. The graphical results of plant identification for the first case are given on Fig. 11 where the output of the plant is compared with the output of the RCVNN-2.

The final MSEs of RCVNN-2 convergence during system identification is given on Table 1 for 1000, 2000, and 3000 learning iterations. The results show a constant MSE decreasing exhibiting a good BP NN convergence.

The graphical results of RCVNN-2 generalization are given on Fig. 12, where the NN weights are frozen and input signal is changed. The output of the plant is compared with the output of the RCVNN-2 exhibiting a good generalization. The final results of RVCNN-2 generalization are given on Table 1 for 3000 steps. The validation results show a constant MSE decreasing exhibiting a good RCVNN-2 generalization.

Comparative final MSE simulation results of nonlinear oscillatory plant identification and generalization using RCVNN-1 and RCVNN-2. The final MSE values obtained during identification and generalization experiments with both RCVNN-1,2 (see Table 1) show that the RCVNN-2 outperformed the RCVNN-1 because the RCVNN-1 activation func-
tion possess singular points that affects the BP learning and
the RCVNN-2 activation function does not possess such sin-
gular points so its learning is perfect. In generalization stage
the result are opposite but still good so the identification re-
sults are dominating.

Finally we could make the conclusion that it is better to use
RCVNN-2 which does not have singular points then to use
RCVNN-1 which have singular points and try to overcome
them.

V. CONCLUSIONS

In the present paper we use the diagrammatic rules, [13], for
the complex case and applied them to obtain the backpropa-
gation training algorithm. We studied two different RCVNNs
topologies with two different activation functions. Using the
diagrammatic rules we obtain the RCVNNs adjoined and
derived the backpropagation algorithm for both cases in a
diagrammatic rules we obtain the RCVNNs adjoined and
nonlinear oscillatory plant confirm the quality of the BP lear-
ing algorithm. The comparison of both RCVNN-1, and
RCVNN-2 topologies for both activation functions give some
priority to the second case in the sense that it is better to use
activation functions without singular points then to use activa-
tion functions with singular points and try to avoid them.

Table I. Final MSE of learning and generalization

<table>
<thead>
<tr>
<th>Number of steps</th>
<th>First type of activation function</th>
<th>Second type of activation function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>0.0264</td>
<td>0.0045</td>
</tr>
<tr>
<td>2000</td>
<td>0.0178</td>
<td>0.0029</td>
</tr>
<tr>
<td>3000</td>
<td>0.0097</td>
<td>0.0028</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Number of steps</th>
<th>First type of activation function</th>
<th>Second type of activation function</th>
</tr>
</thead>
<tbody>
<tr>
<td>3000</td>
<td>0.0014</td>
<td>0.0025</td>
</tr>
</tbody>
</table>

REFERENCES

Zimmermann , and A. Knoll, "Complex Valued Open
Recurrent Neural Network for Power Transformer
Modeling ," International Journal of Applied
Mathematics and Informatics, vol. 6, no. 1, pp. 41-48,
2012.
[3] L. Ferariu, "Nonlinear System Identification Based on
Evolutionary Dynamic Neural Network," in Proc. of
European Control Conference, Cambridge, UK, paper
Network for Group-Movement Control of Mobile
Robots," in Proc. SICE Annual Conference 2012, Japan,
2012, pp. 1806 - 1809.
Neural Networks with Feedback Loops," in Proc. IEEE
International Conference on Neural Networks, vol. 1, San
Francisco, CA, 1993, pp. 156-161.
[6] H. G. H.G. Zimmermann, A. Minin, and V. Kusheebaeva,
"Historical Consistent Complex Valued Recurrent Neural
185-192.
gation Algorithm," IEEE Transactions on Signal
[8] T. Nitta, Complex-Valued Neural Networks: Utilizing
High-Dimensional Parameters. IGI Global, 2009.
the complex backpropagation algorithm," in Proc. of
IEEE International Conference on Neural Networks, vol.
[10] N. Miklos, and B. Salik, "Neural Networks with Complex
Activations and Connection Weights," Complex Systems,
Backpropagation," IEEE Transactions on Circuits and
[12] I. S. Baruch, and C. R. Mariaca-Gaspar, "A Levenberg-
Marquardt Learning Applied for Recurrent Neural
Identification and Control of a Wastewater Treatment
Bioprocess," International Journal of Intelligent Systems,
deriving and relating temporal neural networks
Valued Recurrent Neural Network: From Architecture to
Training," Journal of Signal and Information Processing,