

Research Article

## Complex Neuro-Fuzzy System for Function Approximation

Reza Shoorangiz<sup>1,\*</sup>, Mohammad H. Marhaban<sup>1</sup>

(1) Department of Electrical and Electronics, Faculty of Engineering, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia.



Copyright © 2014 Australian International Academic Centre, Australia

doi:10.7575/aiac.ijaep.v.1n.2p.5

### Article history:

Received 03 October 2013

Reviewed 05 October 2013

Revised 05 October 2013

Accepted 06 October 2013

Published 09 October 2013

**Abstract.** Complex fuzzy sets have been developed recently and extends truth values to unit circle in complex plane. Complex fuzzy logic then developed by employing complex fuzzy sets. In this paper, a novel adaptive complex neuro fuzzy inference system based on complex fuzzy logic is proposed for function approximation. The underlying procedure of this network and its learning rule are described. Afterwards, the performance of this system is evaluated by two functions consisting of Sine wave and Sinc function.

**Keywords:** complex fuzzy logic (CFL), hybrid learning, complex fuzzy set (CFS), complex neural networks (CNN).

### 1 Introduction

Fuzzy set theory proposed by Zadeh<sup>[1]</sup> and consequently fuzzy logic have been widely used for different purposes. Recently, complex fuzzy set<sup>[2]</sup> has been proposed as an extension to traditional fuzzy set theory. According to<sup>[2]</sup>, complex fuzzy set extends truth values to unit disk of complex plane while possible membership degree of traditional fuzzy set is limited to real numbers in the range of [0, 1]. Therefore, complex fuzzy set is defined mathematically as:

$$\{S = (x, \mu_S(x)) \mid x \in U, \mu_A(x) = r_S(x) \cdot e^{i\omega_S(x)}\} \quad (1)$$

where  $S$  is the complex fuzzy set,  $U$  is universe of discourse and  $\mu_A$  is the complex membership function to characterize complex fuzzy set  $A$ . Each complex membership function is consisted of amplitude  $r_S(x)$  bounded in [0, 1] and phase part  $\omega_S(x)$  which can range [0, 2 $\pi$ ].

Complex fuzzy set maintain a novel framework which have all advantages of type-1 fuzzy set further to the extra properties and characteristics attributed to complex-valued nature. This nature of complex membership functions allow them to obtain "wave-like" properties and therefore they can have constructive and destructive interfere with each other<sup>[3]</sup>. Nevertheless, considering all properties and interactions of complex fuzzy sets is very difficult, if not impossible, for humans and even experts. This leads to use machine learning architecture in order to elicit rules and tune complex membership functions.

Chen et al.<sup>[4]</sup> proposed adaptive neurocomplex fuzzy inferential system (ANCFIS) as the first realization of complex fuzzy logic in machine learning. ANCFIS employs signature property<sup>[5]</sup> to reduce the network size.

Moreover, complex membership functions are defined as sine to mimic the Fourier theorem<sup>[3]</sup>. Nonetheless, this system is univariate and only applied to time-series forecasting.

In this paper, we propose an adaptive complex neuro-fuzzy inferential system (ACNFIS) to deal with function approximation problems. The proposed system is based on well-known adaptive network fuzzy inference system (ANFIS)<sup>[6]</sup> with modifications to employ complex fuzzy sets. Moreover, we used the suggested approach in<sup>[5]</sup> to build complex membership function based on two separate real valued functions as amplitude and phase.

The rest of this paper is organized as follows: In section 2 a brief review of complex fuzzy sets and their operations presented. Section 3 describes the proposed ACNFIS architecture. Results are discussed in section 4. It follows by conclusion and future work.

### 2 Complex Fuzzy Logic Review

Fuzzy logic suggested an alternative way for uncertainty modeling. Fuzzy allows to use human knowledge to solve vague and imprecise informations linguistically. This gives the opportunity to use linguistic variables, if-then rules and human reasoning to model systems without precise qualitative analysis. Ramot et al.<sup>[2]</sup> proposed complex fuzzy set that extends possible values of traditional fuzzy sets from real numbers to complex plane. Complex fuzzy set  $S$  is characterized by a complex-valued membership function  $\mu_S(x)$  whose range is the unit disk in complex plane. Thus, complex membership function assign a complex number to any element  $x$  in the universe of discourse. This concept is different from fuzzy complex numbers<sup>[7-10]</sup>. Fuzzy complex number is a type-1 fuzzy set with complex-valued members. Equivalently, fuzzy complex number is a real valued function which is

\*Corresponding author: R. Shoorangiz

☎: +60 12 628 6545

✉: r.shoorangiz@gmail.com

defined on complex number set. The notion of complex fuzzy set, however, employs complex-valued membership function to map each element into  $[0, 1] \times [0, 2\pi]$ . The basic operations introduced by Ramot et al.<sup>[5]</sup> are as follows:

1. *Union*: Let  $\mu_A = r_A(x) \cdot e^{j\omega_A(x)}$  and  $\mu_B = r_B(x) \cdot e^{j\omega_B(x)}$  be complex membership functions where  $A$  and  $B$  are complex fuzzy sets defined on universe of discourse  $U$ . The complex membership function of  $A \cup B$  is

$$\mu_{A \cup B} = (r_A \oplus r_B) \cdot e^{j(\omega_A \cup \omega_B)} \quad (2)$$

where  $\oplus$  represents any s-norm function which satisfy the type-1 fuzzy union axioms. The phase part, the function can be select from other possibilities. The followings are mentioned in<sup>[5]</sup>:

$$\text{Sum} \quad \omega_{A \cup B} = \omega_A + \omega_B \quad (3)$$

$$\text{Min} \quad \omega_{A \cup B} = \min(\omega_A, \omega_B) \quad (4)$$

$$\text{Max} \quad \omega_{A \cup B} = \max(\omega_A, \omega_B) \quad (5)$$

$$\text{Winner Take All} \quad \omega_{A \cup B} = \begin{cases} \omega_A & r_A > r_B \\ \omega_B & r_B > r_A \end{cases} \quad (6)$$

2. *Intersection*: Assume  $\mu_A$  and  $\mu_B$  are two complex membership functions of two complex fuzzy sets  $A$  and  $B$ . Similar to the union, the intersection is introduced as:

$$\mu_{A \cap B} = (r_A \star r_B) \cdot e^{j(\omega_A \cap \omega_B)} \quad (7)$$

where  $\star$  can be selected as any t-norm that satisfy the type-1 fuzzy intersection axioms. The possible functions for the phase part are the same as introduced in union section. Selection of this operator is totally based on the application. In the rest of this paper, for intersection we use algebraic product as t-norm for magnitude and summation for the phase part. These selections are made in order to resemble the complex-valued production.

3. *Complement*: Suppose  $\mu_A$  is complex membership functions of complex fuzzy set  $A$ . The complement operator mentioned in<sup>[2]</sup> is defined as:

$$\begin{aligned} \mu_{\bar{A}} &= r_{\bar{A}} \cdot e^{j\omega_{\bar{A}}} \\ &= (1 - r_A) \cdot e^{j\omega_{\bar{A}}} \end{aligned} \quad (8)$$

The complement operator breaks the complex membership function into amplitude and phase parts. Traditional complement operator applies to the amplitude part, however, the complement of the phase part is based on the interpretation of membership phase. According to<sup>[2]</sup>, membership phase of  $\bar{S}$  must be the same as  $\bar{S}$  in order to satisfy the axioms. Therefore, the complement operator does not apply to the membership phase.

Complex fuzzy set is the backbone of complex fuzzy logic. The latter is a natural extension of fuzzy logic that benefits from advantages of complex fuzzy sets<sup>[5]</sup>. As a result, complex fuzzy logic can deal the problems that are difficult or impossible to be addressed with traditional fuzzy logic.

### 3 Architecture of ACNFIS

In this section, we introduce the ACNFIS and its underlying procedure. ACNFIS is a multilayer feed-

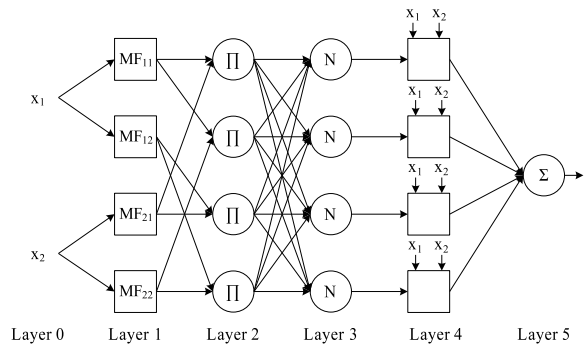


Fig. 1. ACNFIS architecture

forward complex-valued neural network whose structure is given in Fig. 1. The architecture of ACNFIS is based on well known ANFIS<sup>[6]</sup> in which the node functions are modified to employ complex fuzzy logic. Each layer and its node function is as follows.

*Layer 0:* This layer is pass the input vector to the next layer without any modification.

*Layer 1:* This layer transforms the input value into complex membership grade for each input. Thus, fuzzification of input data is utilized at this stage. The node function of  $j$ th node can be written in terms of amplitude ( $A$ ) and phase ( $P$ ) as:

$$O_j^1 = \mu(x) = A(x) \cdot e^{iP(x)} = A(x) \angle P(x) \quad (9)$$

We selected two separate real valued functions as  $A(x)$  and  $B(x)$  to represent amplitude and phase parts. This type of selection is made based on Liouville's theorem<sup>[11]</sup>. According to this theorem, fully complex-valued function cannot be both analytic and bounded unless it is constant. However, the amplitude of complex membership function must be bounded in unit interval  $[0, 1]$ . Selecting two separate real valued function gives the opportunity to build bounded complex membership function. Nevertheless, other approaches can bound the complex-valued grade of membership such as Elliot function that has been utilized in<sup>[4]</sup>. Subsequently, in selection of function for phase part, there would be no restriction. Indeed, a function that spread on whole range of  $[0, 2\pi]$  is more effective. To this end, we selected two Gaussian functions to build the complex-valued membership functions as follows:

$$\mu(x) = \exp\left(-\left(\frac{x - c_{Aj}}{a_{Aj}}\right)^2\right) \angle \left(2\pi \exp\left(-\left(\frac{x - c_{Pj}}{a_{Pj}}\right)^2\right)\right) \quad (10)$$

where subscripts  $A$  and  $P$  indicate amplitude and phase parameters. Fig. 2 is visual representation of a sample complex membership function of this family. Each complex membership function has four nonlinear (antecedent) parameters to be identified and fine tuned. Therefore, the nodes at this layer are adaptive and the antecedent parameters will be updated iteratively based on mean squared error and gradient vector.

*Layer 2:* Nodes of this layer are static nodes. Each node represent a specific rule and its output is the firing strength of the rule. So this layer may be interpreted as rule base of complex fuzzy logic. Normally, we consider all the possible rules at this stage. Also complex-valued multiplication is selected as node functions since it satisfies both amplitude and phase t-norm axioms (see

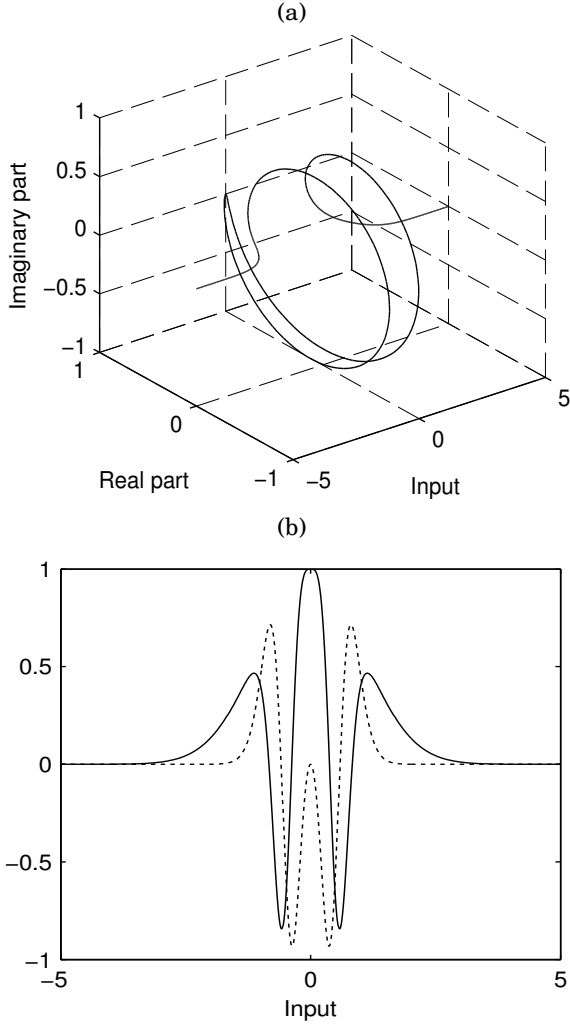


Fig. 2. A sample complex membership function with parameter set  $[c_A, a_A, c_P, a_P] = [0, 1, 0, 0.5]$ ; (a) 3D visualisation and (b) 2D visualisation: real (solid) and imaginary (dashed) parts

section 2). The output for this layer is:

$$O_j^2 = w_j = \prod_{k=1}^n O_k^1 = \prod_{k=1}^n A_k^1 \cdot e^{i \sum_{k=1}^n P_k^1} \quad (11)$$

where  $j$  represent the index of the rule,  $k$  is the index of antecedents and  $n$  is total number of inputs for the  $j_{th}$  rule.

**Layer 3 :** Normalized rule firing strengths are calculated at this layer. Output of each node is evaluated using:

$$O_j^3 = \bar{w}_j = \frac{w_j}{\sum_{j=1}^n |w_j|} \quad (12)$$

We used summation of weight amplitudes in order to normalize the weights. This leads to have a rotation invariant operator. In addition, denominator might be zero if we use summation of complex weights and will cause singularity at this stage which makes the system unreliable.

**Layer 4:** Takagi and Sugeno's inferential system<sup>[12]</sup> is implemented at this layer. Each node corresponds one consequent rule of Takagi and Sugeno's inferential system and therefore has linear parameters to find. The

node function that we used is:

$$O_j^4 = |\bar{w}_j| (\cos(\angle \bar{w}_j) f_j + \sin(\angle \bar{w}_j) g_j) \quad (13)$$

$$f_j = \sum_k^{N_0} (p_{j,k} x_k) + r_j \quad (14)$$

$$g_j = \sum_k^{N_0} (q_{j,k} x_k) + s_j \quad (15)$$

where  $|\bar{w}_j|$  and  $\angle \bar{w}_j$  are amplitude and phase of normalized weights and  $\{\bar{p}_j, \bar{q}_j, r_j, s_j\}$  is consequent parameter set.

**Layer 5:** This is output layer which calculates the output value of the system by calculating summation of all its input values.

$$O_1^5 = \sum_j O_j^4 \quad (16)$$

Aforementioned system is an adaptive network which implements complex fuzzy logic. This system, as it mentioned, has two set of parameters: premise (nonlinear) and consequent (linear) parameters. To find consequent set of parameters, we employed famous least square estimation (LSE)<sup>[6]</sup>. Let vector  $\vec{X}$  be the set of consequent parameters. Based on the operations of layers four and five, we can write the output as a linear combination of consequent parameters ( $\vec{X}$ ). Thus, having 2 inputs and  $P$  training data pairs leads us to (17) where  $A$  is the known coefficient matrix (18) and  $B$  is the desired output vector (19).

$$AX = B \quad (17)$$

$$A = \begin{bmatrix} \alpha_{11}x_{11} & \alpha_{11}x_{12} & \alpha_{11} & \cdots & \beta_{21}x_{11} & \beta_{21}x_{12} & \beta_{21} \\ \alpha_{12}x_{11} & \alpha_{12}x_{12} & \alpha_{12} & \cdots & \beta_{22}x_{11} & \beta_{22}x_{12} & \beta_{22} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \alpha_{1P}x_{11} & \alpha_{1P}x_{12} & \alpha_{1P} & \cdots & \beta_{2P}x_{11} & \beta_{2P}x_{12} & \beta_{2P} \end{bmatrix} \quad (18)$$

$$B = \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_P \end{bmatrix} \quad (19)$$

where  $T_p$  is the target value for the  $p_{th}$  training data pair and  $\alpha_{ip}$  and  $\beta_{ip}$  are representing  $|\bar{w}_{ip}| \cos(\angle \bar{w}_{ip})$  and  $|\bar{w}_{ip}| \sin(\angle \bar{w}_{ip})$  for  $p_{th}$  training data, respectively. Thus, we utilized pseudo-inverse least square estimation to find the best set of consequent parameters. The solution is given as:

$$X = (A^T A)^{-1} A^T B \quad (20)$$

It has to be noted that during the forward pass, we identify consequent parameter set while premise parameters are assumed as constants. When the consequent set of parameters are identified, we use Levenberg–Marquardt algorithm<sup>[13]</sup> to update premise parameters. The error vector is defined:

$$\mathbf{e} = [ e_1, e_2, \cdots, e_P ] \\ = [ O_1 - T_1, O_2 - T_2, \cdots, O_P - T_P ] \quad (21)$$

where  $O_i$  and  $T_i$  are ACNFIS and desired output for  $i_{th}$  data pair, respectively. Accordingly, the squared error can be written as dot product of error vector:

$$\mathbf{E} = \frac{1}{2} \langle \mathbf{e}, \mathbf{e} \rangle = \frac{1}{2} \mathbf{e}^T \mathbf{e} \quad (22)$$

For the sake of simplicity, we added  $\frac{1}{2}$  in (22). According to LM back-propagation, update parameters are based on the following.

$$\mathbf{x}(k+1) = \mathbf{x}(k) + \Delta \mathbf{x} \quad (23)$$

where  $\mathbf{x}$  is antecedent parameter vector and  $\Delta \mathbf{x}$  is the update vector which is calculated based on the Jacobian and Hessian matrices of error w.r.t. antecedent parameters. So the update matrix becomes

$$\Delta \mathbf{x} = -[\mathbf{J}^T \mathbf{J} + \mu \mathbf{I}]^{-1} \mathbf{J}^T \mathbf{e} \quad (24)$$

where  $J$  is the Jacobian matrix of error w.r.t. antecedent parameters and  $\mu$  is LM coefficient. The Jacobian matrix has the form

$$\mathbf{J} = \frac{\partial \mathbf{e}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial e_1}{\partial x_1} & \frac{\partial e_1}{\partial x_2} & \dots & \frac{\partial e_1}{\partial x_n} \\ \frac{\partial e_2}{\partial x_1} & \frac{\partial e_2}{\partial x_2} & \dots & \frac{\partial e_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_p}{\partial x_1} & \frac{\partial e_p}{\partial x_2} & \dots & \frac{\partial e_p}{\partial x_n} \end{bmatrix} \quad (25)$$

Value of  $\mu$  changes adaptively based on the difference in two consequent errors. If the error decreased,  $\mu$  is divided by factor  $\beta$ . However, if the error increased,  $\mu$  is multiplied by  $\beta$  where  $\beta$  is a constant greater than one. In our design, we chose  $\beta = 10$ . at each iteration. During training phase, the system uses mentioned hybrid learning algorithm iteratively to optimize all the antecedent and consequent parameters.

### 4 Results and Discussion

We evaluate performance of the system using approximation of Sine and Sinc functions. We chose input domain of  $[0, 2\pi]$  for Sine function. Then, we extracted 41 data pairs uniformly which consisted of 21 training data and 20 test data. ACNFIS with two rules has been trained using training data and mean squared error reached  $1.4 \times 10^{-7}$ . Afterwards, we tested the trained system using testing data and it gives mean squared error of  $1.3 \times 10^{-7}$ . Fig. 3 shows the output and error of ACNFIS for Sine testing data. Complex membership function parameters are given in Table 1.

Table 1. Complex membership functions parameters for Sine approximation

	$a_A$	$c_A$	$a_P$	$c_P$
$MF_1$	1.55	10.02	12.41	-0.37
$MF_2$	2.53	20.04	4.00	-0.88

Same procedure has been done for Sinc function. We extracted 81 data pairs from input domain  $[-2\pi, 2\pi]$  uniformly. We used 41 data pairs as training data where after 200 epochs, mean square error reached  $2.76 \times 10^{-4}$  and therefore training process stopped. Then, we tested the system using other 40 data pairs which gives mean squared error  $2.52 \times 10^{-4}$ . Fig. 4 depicts ACNFIS output and its error for Sinc function. Table 2 shows the values of complex membership functions parameters. The performance of ACNFIS is compared with ANFIS and feedforward neural network in Table 3. ACNFIS shows better

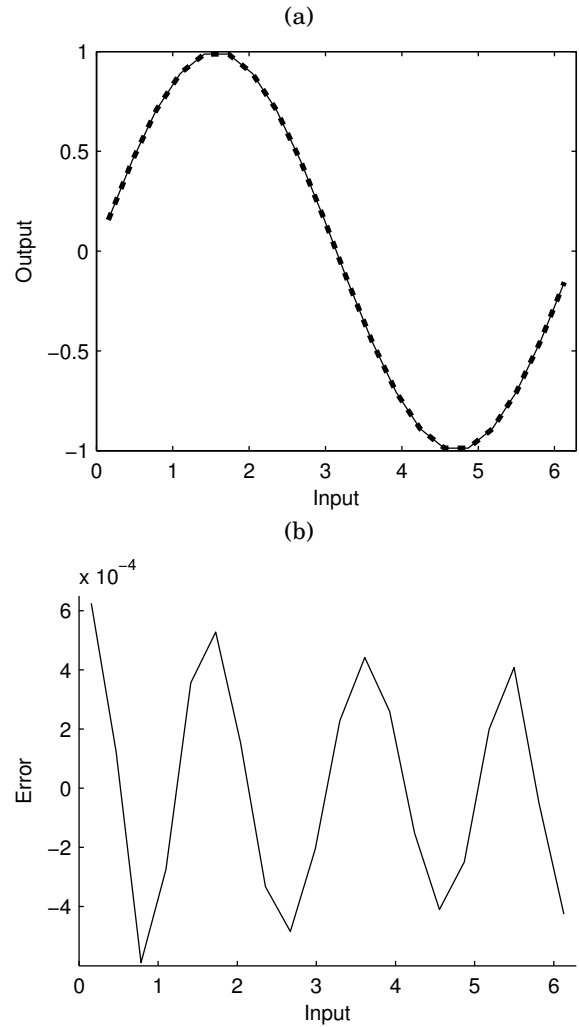


Fig. 3. Approximation of Sine function; (a) test data (solid) versus ACNFIS output (dashed) and (b) testing error

performance compared to two other approaches while it has fewer number of nonlinear and linear parameters as well.

Table 2. Complex membership functions parameters for Sinc wave approximation

	$a_A$	$c_A$	$a_P$	$c_P$
$MF_1$	3.74	-1.98	1.44	-2.49
$MF_2$	3.78	0.29	1.44	2.49

Table 3. Performance comparison for Sinc function approximation

	Nonlinear/Linear parameters	$MSE_{trn}$	$MSE_{chk}$
ACNFIS	8/8	2.76e-4	2.52e-4
ANFIS	10/10	3.3e-4	3.4e-4
Feedforward NN	9/9	-	5.32e-4

### 5 Conclusion

Adaptive complex neuro fuzzy inference system (ACNFIS) has been proposed in this paper. This system is capable of learning various functions based on input-output data and approximate them accurately. Underlying procedure of ACNFIS network shows that its network is equivalent to well-known ANFIS system; however, all the node functions are modified to utilize complex fuzzy logic. Moreover, closed form learning rule for ACNFIS is derived to optimize the parameters of ACNFIS in forward and backward passes. The performance of this system evaluated by Sine and Sinc functions which shows

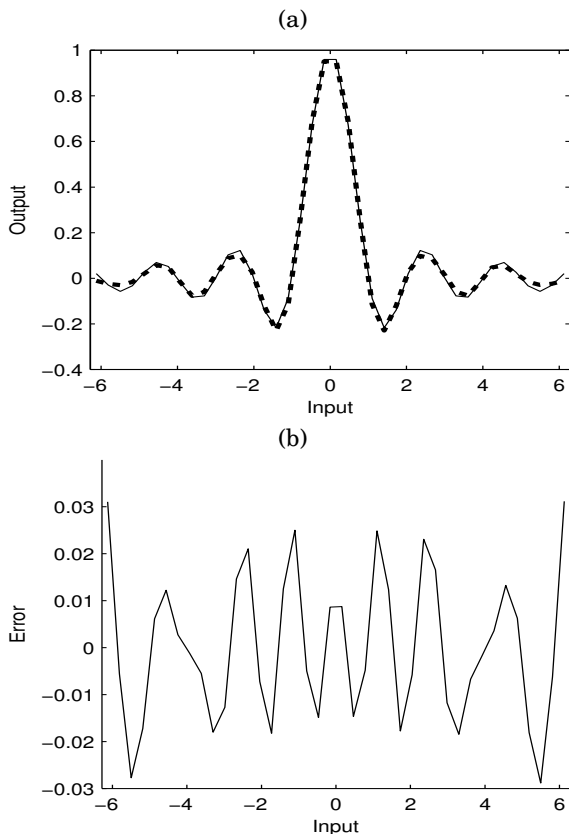


Fig. 4. Approximation of Sinc function; (a) test data (solid) versus ACNFIS output (dashed) and (b) testing error capability of proposed system for approximating nonlinear functions.

The future works can be concentrated on interpretation of complex fuzzy sets and complex valued grade of memberships. Furthermore, various structures have been developed for traditional fuzzy sets which can be modified to utilize complex fuzzy logic.

## Acknowledgement

The authors would like to thank the reviewers for their helpful comments.

## REFERENCES

- [1] L. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.
- [2] D. Ramot, R. Milo, M. Friedman, and A. Kandel, "Complex fuzzy sets," *Fuzzy Systems, IEEE Transactions on*, vol. 10, no. 2, pp. 171–186, apr 2002.
- [3] J. Man, Z. Chen, and S. Dick, "Towards inductive learning of complex fuzzy inference systems," in *Fuzzy Information Processing Society, 2007. NAFIPS '07. Annual Meeting of the North American*, june 2007, pp. 415–420.
- [4] Z. Chen, S. Aghakhani, J. Man, and S. Dick, "Ancfis: A neurofuzzy architecture employing complex fuzzy sets," *Fuzzy Systems, IEEE Transactions on*, vol. 19, no. 2, pp. 305–322, april 2011.
- [5] D. Ramot, M. Friedman, G. Langholz, and A. Kandel, "Complex fuzzy logic," *Fuzzy Systems, IEEE Transactions on*, vol. 11, no. 4, pp. 450–461, aug. 2003.
- [6] J.-S. Jang, "Anfis: adaptive-network-based fuzzy inference system," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 23, no. 3, pp. 665–685, may/jun 1993.
- [7] J. Buckley, "Fuzzy complex numbers," *Fuzzy Sets and Systems*, vol. 33, no. 3, pp. 333–345, 1989.
- [8] K. Yoon, "A probabilistic approach to rank complex fuzzy numbers," *Fuzzy Sets and Systems*, vol. 80, no. 2, pp. 167–176, 1996.
- [9] J. Buckley and Y. Qu, "Fuzzy complex analysis i: Differentiation," *Fuzzy Sets and Systems*, vol. 41, no. 3, pp. 269–284, 1991.
- [10] J. J. Buckley, "Fuzzy complex analysis ii: Integration," *Fuzzy Sets and Systems*, vol. 49, no. 2, pp. 171–179, 1992.
- [11] T. Apostol, *Mathematical analysis*, ser. World student series. Addison-Wesley Pub. Co., 1974.
- [12] T. Takagi and M. Sugeno, "Fuzzy Identification of Systems and Its Applications to Modeling and Control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 15, no. 1, pp. 116–132, Feb. 1985.
- [13] M. Hagan and M. Menhaj, "Training feedforward networks with the marquardt algorithm," *Neural Networks, IEEE Transactions on*, vol. 5, no. 6, pp. 989–993, nov 1994.