International Journal of Neural Systems, Vol. 19, No. 4 (2009) 253–267

World Scientific
www.worldscientific.com

A FULLY COMPLEX-VALUED RADIAL BASIS FUNCTION NETWORK AND ITS LEARNING ALGORITHM

R. SAVITHA*, S. SURESH† and N. SUNDARARAJAN*
*School of Electrical and Electronics Engineering
Nanyang Technological University, Singapore
†CIST, Korea University, Seoul, Republic of South Korea
Sundaram.Suresh@hotmail.com

In this paper, a fully complex-valued radial basis function (FC-RBF) network with a fully complex-valued activation function has been proposed, and its complex-valued gradient descent learning algorithm has been developed. The fully complex activation function, *sech*(.) of the proposed network, satisfies all the properties needed for a complex-valued activation function and has Gaussian-like characteristics. It maps $C^n \rightarrow C$, unlike the existing activation functions of complex-valued RBF network that maps $C^n \rightarrow R$. Since the performance of the complex-RBF network depends on the number of neurons and initialization of network parameters, we propose a K-means clustering based neuron selection and center initialization scheme. First, we present a study on convergence using complex XOR problem. Next, we present a synthetic function approximation problem and the two-spiral classification problem. Finally, we present the results for two practical applications, viz., a non-minimum phase equalization and an adaptive beam-forming problem. The performance of the network was compared with other well-known complex-valued RBF networks available in literature, viz., split-complex CRBF, CMRAN and the CELM. The results indicate that the proposed fully complex-valued network has better convergence, approximation and classification ability.

Keywords: Radial basis function; complex-valued activation functions; k-means clustering; phase equalization; adaptive beam-forming.

1. Introduction

With the evolution of adaptive array signal processing, communication engineering and medical imaging involving complex-valued signals, adaptive algorithms to restore and retrieve signals, and to reconstruct images are in demand. The adaptive nature of neural networks are being increasingly exploited for such applications. This calls for efficient complex-valued neural network algorithms, and hence, research on developing complex-valued neural networks is on the rise, and a brief review of the same is given below. A complex-valued Hopfield Network (CHN) is proposed and is used in proving how synchronization can be achieved in a high-dimensional chaotic neural network in Ref. 1. Complex-valued Independent Component Analysis is discussed in Ref. 2. The activation dynamics of a complex-valued neural network is studied in Ref. 5. Research interests in Complex-valued neural networks also includes proving the uniqueness theorem for the 3-layered complex-valued MLP neural networks with non-zero threshold parameters[3] and evaluation of the ability of real and complex-valued MLP's in performing the classification tasks of quaternionic random processes.[4] Research focus is also on developing efficient complex-valued logic gates and memory. Complex-valued Associative memory (CAM), which is complex-valued Hopfield Associative Memory (HAM), was first introduced in Ref. 6. The capacity of the CAM was improved using the pseudo-relaxation algorithm proposed in Ref. 7. A novel logic gate is proposed in Ref. 8. This logic gate is capable of learning multiple functions at frequencies different from each other, and analyzing the

frequency-domain multiplexing ability in the learning based on complex-valued Hebbian rule. Associative memory based on quaternionic Hopfield network are investigated in Ref. 9. The networks used in Ref. 9 are composed of quaternionic neurons and the input, output, threshold and connection weights are all represented in quaternions.

Developing efficient neural network algorithms for function approximation, classification and array signal processing requires a fully complex-valued networks, capable of manipulating on complex-valued inputs with complex-valued weights and thresholds. Conventionally, split complex-valued networks, which manipulated on two real-valued signals i.e., either real-imaginary components or magnitude-phase components of the complex-valued signal, were employed.[10] For example, in Ref. 11, splitting neurons, working on the real and imaginary part of the complex signal are used in realizing the nonlinear functions involved in the processing of the complex signals for Independent Component Analysis (ICA). Identification of split-complex and fully complex nature of signals is the most important aspect of complex nonlinear modeling, and this is addressed in Ref. 12. It is shown here, that the algorithms derived are capable of tracking changes in the modality of a few complex vector fields. The split-complex-valued networks, however, suffer from inability to approximate phase accurately.[13] The need for accurate phase approximations require fully complex-valued neural networks with a fully complex-valued activation function. But, selection of an appropriate complex-valued activation function is difficult, as an analytic and bounded function in the complex domain C is a constant function according to Liouville's theorem.[14] Besides, the performance of a complex-valued neural network depends largely on the activation function used, the number of neurons in the hidden layer, learning rate parameter and the region of weight initializations.[15] Several activation functions have been proposed in the literature for complex-valued Multi-Layer Perceptron (CMLP).[14–17]

On the other hand, although several real-valued RBF learning algorithms[18–24] have been proposed for different applications, only a few of these real-valued neural networks are extended to the complex-valued domain.[25–30] However, the architecture of these complex-valued RBF networks remain split complex, their activation functions response remains real-valued, and little work has been done in identifying a fully complex-valued activation function for a RBF network. The basic architecture and learning algorithms of complex-valued radial basis function networks were first introduced in Ref. 25 and they were applied to communication channel equalization in Refs. 26 and 27. These networks used the Gaussian RBF with Euclidean norm $\|(x-c)\|$ to process complex-valued signals. However, in these networks, the input is not efficiently transmitted to the output, as the activation functions used in these networks maps $C^n \rightarrow R$. Complex Minimal Resource Allocation Network (CMRAN) algorithm, a sequential learning algorithm, which is an extension of the MRAN algorithm, is developed in Ref. 28. CMRAN has the ability to add and prune the number of hidden neurons to ensure a parsimonious network structure. Another complex-valued RBF algorithm, using a sequential learning scheme, referred to as the Complex-valued Growing and Pruning (CGAP) algorithm, has been proposed in Ref. 29. The learning algorithm in this network links the significance of hidden neurons to the learning accuracy, thereby, a growing and pruning strategy for an RBF neural network with complex inputs is derived. When there is no growing or pruning, a complex, extended Kalman filter is used to adjust the RBF network parameters. Thus the number of centers of the RBF network is decided by this growing and pruning strategy. Using the recently developed real-valued Extreme Learning Machines, a fully complex Extreme Learning Machine (CELM) was proposed by Ming-bin *et al.*[30] All the parameters of CELM are analytically determined instead of being tuned and also, CELM has faster learning speed compared to other complex-valued learning algorithms. However, in all these approaches, in spite of the centers and weights being complex-valued, the activation function maps $C^n \rightarrow R$. This results in a real-valued gradient at the hidden nodes during learning, which does not reflect a fully complex-valued gradient. The lack of a fully complex-valued gradient, during learning, affects the phase approximation ability of the network. Hence the above mentioned complex-valued RBF networks approximate phase less accurately. This calls for a fully complex- valued activation function for the complex-valued RBF network, so that a fully complex-valued gradient is used during training.

In this paper, we propose a fully complex-valued Gaussian like activation function which has a well defined complex-valued gradient.[31] The activation function is *sech*(.) with a complex scaling factor. The *sech*(.) function is analytic and bounded almost everywhere in the complex plane. The network which uses *sech*(.) activation function is referred to, here, as the "Fully Complex-valued Radial Basis Function" (FC-RBF) network. The well defined complex-valued gradient descent learning algorithm for FC-RBF is presented, and the performance of the network is studied, in comparison with other well-known complex-valued RBF networks. The complex XOR problem is used to study the convergence characteristics, the synthetic complex-valued function approximation problem to study the approximating ability and the two-spiral classification problem to highlight the highly non-linear classification ability. The advantage of the proposed algorithm is also evaluated in two real-world applications, viz., the non-minimum phase equalization problem and the adaptive beam-forming problem. The choice of number of hidden neurons and their centers are initially done through a heuristic procedure similar to Ref. 32. A number of clustering algorithms are available in literature, to select the network architecture and to initialize the parameters of the network. For example, in Ref. 33, a statistical multivariate analysis based on Mahalanobis distance is employed to perform data clustering and parameter reduction. Later, K-means clustering algorithm (KMC)[34–36] was used in the selection of the number of hidden neurons and the initialization of their complex-valued centers. The results are compared with the existing well-known complex-valued RBF algorithms in literature.

The paper is organized as follows: In Sec. 2, the architecture of the FC-RBF network is presented, along with the characteristics of the proposed activation function. The complex-valued gradient descent learning algorithm for the parameter updates of the FC-RBF network is presented in Sec. 3. The KMC algorithm, used in selection of network size and initializations of the network parameters, is also briefly discussed. In Sec. 4, the performance of the proposed network is evaluated on the complex XOR problem, a synthetic complex-valued function approximation problem and the two-spiral problem. The performance was also evaluated on two real-world problems, viz., a non-minimum phase equalization problem and an adaptive beam-forming problem. The performance of the proposed network is compared against complex-valued networks available in literature. Section 5 presents the conclusions from this study.

2. A Fully Complex-Valued RBF Network (FC-RBF)

The complex-valued RBF networks available in literature, viz., CRBF, CMRAN, CGAP etc., lack a fully complex-valued activation function. Instead, the Gaussian activation function (Eq. (1))

$$f(\underline{Z}) = \exp^{-((\underline{Z}-\underline{C})^H(\underline{Z}-\underline{C})/\sigma^2)} \tag{1}$$

used in these networks, maps $C^n \rightarrow R$ and hence, fail to approximate phase accurately.[29] So, a fully complex-valued RBF network with complex-valued gradients and fully complex-valued activation function becomes essential for more accurate phase approximations. One such network and activation function is proposed in this section.

2.1. *Network architecture*

The architecture of FC-RBF network is as shown in Fig. 1. The network consists of an input, a hidden and a output layer. Let $\mathbf{X}$ be the m-dimensional complex-valued input to the network, i.e., $\mathbf{X} = [x_1 x_2 \cdots x_m]^T$ and target $\mathbf{T}^*$ be the n-dimensional complex-valued output vector, i.e., $\mathbf{T}^* = [t_1 t_2 \cdots t_n]^T$, where the superscript T denotes the transpose of the vectors. The basic functional

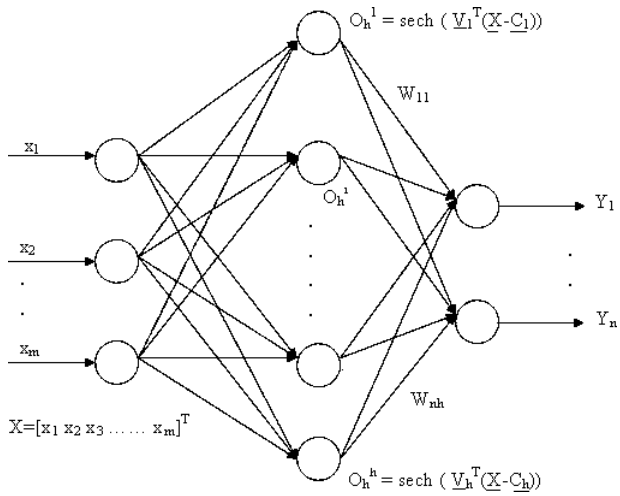

Fig. 1. The fully complex-valued RBF network.

unit of complex-valued hidden neuron is $'sech(\cdot)'$ function. The activation function used in hidden neurons are described in Eq. (2).

$$O_h^i = sech(\underline{V}_i^T(\underline{X} - \underline{C}_i)), \quad i = 1, 2, \ldots, h. \quad (2)$$

where $\underline{V}_i$, the complex-valued scaling factor and $\underline{C}_i$, the center of the ith neuron, are vectors of dimension m. h is the number of hidden neurons. The scaling factor V_i plays a role similar to the deviation σ in the real-valued Gaussian function. The output of the network Y_i is given by Eq. (3).

$$Y_i = \sum_{j=1}^{h} W_{ij} O_h^j, \quad i = 1, 2, \ldots, n. \quad (3)$$

where the W_{ij} are the complex-valued output weights. The number of neurons in the hidden layer and their initial centers affect the performance of the FC-RBF network. The procedure followed in choosing these parameters have been briefly discussed in Sec. 3.1.

2.2. *The activation function*

The network uses $sech(.)$ as a basis for developing fully complex-valued activation function, and the functional unit of the hidden neuron using $sech(.)$ is the form shown in Eq. (2). The Gaussian like response of the *sech* function can be seen from Fig. 2.

The proposed activation function $sech(.)$ satisfies the desired properties for a complex-valued activation function stated in Ref. 14, namely,

> *In a bounded domain of complex plane C, a fully complex nonlinear activation function $f(z)$ needs to be analytic and bounded almost everywhere.*

$sech(.)$ has periodic isolated singularities at $(1/2 + n)\pi \ i$ where $n \epsilon \aleph$. Therefore, it is analytic and bounded a.e. in a bounded domain of a complex plane. The proof for the approximation capability of $sech(.)$ activation, with isolated singularities, is straightforward from Ref. 13. i.e., the isolated singularity of $sech(.)$ can provide uniformly converging approximation in the deleted annulus of the singularity in the region of convergence defined by the Laurent series. Figures. 2 and 3 gives the magnitude and phase response plot of this function over the complex plane. The Gaussian-like characteristic of the function is understandable from the magnitude response plot.

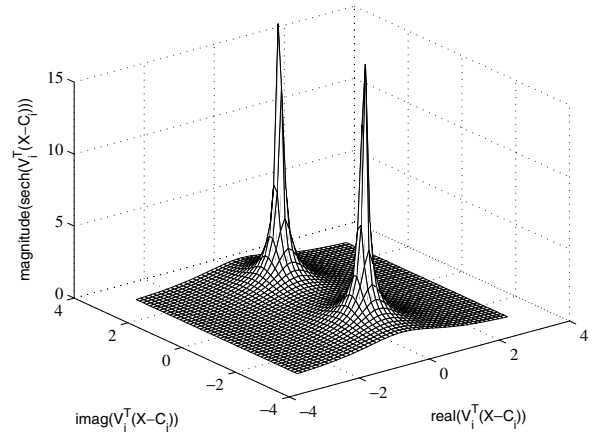

Fig. 2. Magnitude plot of $sech(.)$.

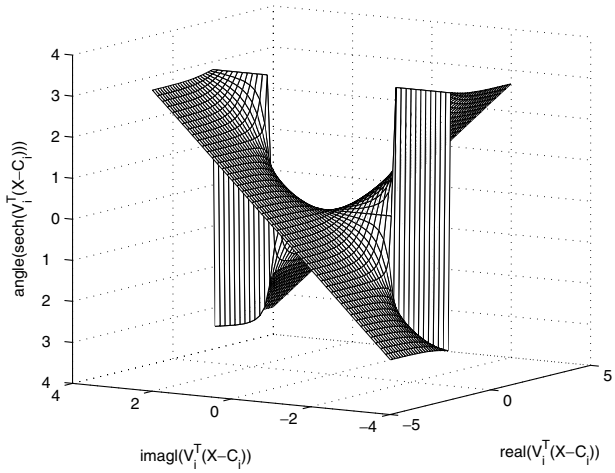

Fig. 3. Phase plot of $sech(.)$.

3. Fully Complex-Valued Gradient Descent Learning Algorithm for FC-RBF

The complex-valued $sech(.)$ proposed in Sec. 2 satisfies the desirable properties of a complex-valued activation functions presented in Ref. 14. It is analytic and bounded almost everywhere. Therefore, Cauchy-Riemann equations can be used to simplify the fully complex-valued RBF algorithm. i.e.,

$$f'(z) = f_x = -if_y \quad (4)$$

The sum-squared error at the output layer can be written as

$$E = \frac{1}{2}\sum_k (\|e_k\|^2); \quad \text{where } e_k = T_k^* - Y_k, \quad (5)$$

If h is the number of neurons in the hidden layer, and $O_h^j = u_j + jv_j$ is the response of the jth

hidden node,

$$Y_n = \sum_{j=1}^{h}(W_{nj}O_h^j) = \sum_{j=1}^{h}(W_{nj}^R + jW_{nj}^I)(u_j + jv_j)$$
$$= \sum_{j=1}^{h}(W_{nj}^R u_j - W_{nj}^I v_j) + j(W_{nj}^R v_j + W_{nj}^I u_j) \quad (6)$$

Superscripts R and I indicate the real and imaginary components, respectively.

The output weight update rule requires the computation of the gradient $\frac{\partial E}{\partial W_{nj}}$. Since the cost function is the real-valued function of a complex-valued variable, the gradient of the error function with respect to the real and imaginary components of the W_{nj} is

$$\frac{\partial E}{\partial W_{nj}} = \nabla W_{nj}E = \frac{\partial E}{\partial W_{nj}^R} + j\frac{\partial E}{\partial W_{nj}^I} \quad (7)$$

Using the chain rule

$$\frac{\partial E}{\partial W_{nj}^R} = \frac{\partial E}{\partial Y_n^R}\frac{\partial Y_n^R}{\partial W_{nj}^R} + \frac{\partial E}{\partial Y_n^I}\frac{\partial Y_n^I}{\partial W_{nj}^R} \quad (8)$$

$$\frac{\partial E}{\partial W_{nj}^I} = \frac{\partial E}{\partial Y_n^R}\frac{\partial Y_n^R}{\partial W_{nj}^I} + \frac{\partial E}{\partial Y_n^I}\frac{\partial Y_n^I}{\partial W_{nj}^I} \quad (9)$$

Defining $\delta_n^R \equiv -\partial E/\partial Y_n^R$ and $\delta_n^I \equiv -\partial E/\partial Y_n^I$ and using the following partial derivatives obtainable from Eq. (6),

$$\frac{\partial Y_n^R}{\partial W_{nj}^R} = u_j; \quad \frac{\partial Y_n^I}{\partial W_{nj}^R} = -v_j;$$
$$\frac{\partial Y_n^R}{\partial W_{nj}^I} = v_j; \quad \frac{\partial Y_n^I}{\partial W_{nj}^I} = u_j \quad (10)$$

$$\frac{\partial E}{\partial W_{nj}^R} = -\delta_n^R u_j - \delta_n^I v_j \quad (11)$$

$$\frac{\partial E}{\partial W_{nj}^I} = -\delta_n^R(-v_j) - \delta_n^I u_j \quad (12)$$

Combining (11) and (12) the gradient of the error can be written as

$$\nabla W_{nj}E = (-\delta_n^R u_j - \delta_n^I v_j) + j(\delta_n^R v_j - \delta_n^I u_j)$$
$$= -(\delta_n^R + j\delta_n^I)\overline{O_h}^j \quad (13)$$

$$\triangle W_{nj} = \alpha\overline{O_h}^j\delta_n \quad (14)$$

where α can be a real, imaginary or complex learning rate and $\overline{O_h}^j$ denotes the complex-conjugate of O_h^j.

Similarly, letting $x_j + jy_j = \sum_{i=1}^{m}\underline{V}_i^T(\underline{X} - \underline{C}_i)$, the update for the weights V_i requires the gradient of the real cost function with respect to the real and imaginary components of $\underline{V}_i$.[a]

$$\frac{\partial E}{\partial V_i} = \nabla V_i E = \frac{\partial E}{\partial V_i^R} + j\frac{\partial E}{\partial V_i^I} \quad (15)$$

$$\frac{\partial E}{\partial V_i^R} = \frac{\partial E}{\partial Y_n^R}\left[\frac{\partial Y_n^R}{\partial u_j}\frac{\partial u_j}{\partial x_i}\frac{\partial x_i}{\partial V_i^R} + \frac{\partial Y_n^R}{\partial u_j}\frac{\partial u_j}{\partial y_i}\frac{\partial y_i}{\partial V_i^R} + \frac{\partial Y_n^R}{\partial v_j}\frac{\partial v_j}{\partial x_i}\frac{\partial x_i}{\partial V_i^R} + \frac{\partial Y_n^R}{\partial v_j}\frac{\partial v_j}{\partial y_i}\frac{\partial y_i}{\partial V_i^R}\right]$$
$$+ \frac{\partial E}{\partial Y_n^I}\left[\frac{\partial Y_n^I}{\partial u_j}\frac{\partial u_j}{\partial x_i}\frac{\partial x_i}{\partial V_i^R} + \frac{\partial Y_n^I}{\partial u_j}\frac{\partial u_j}{\partial y_i}\frac{\partial y_i}{\partial V_i^R} + \frac{\partial Y_n^I}{\partial v_j}\frac{\partial v_j}{\partial x_i}\frac{\partial x_i}{\partial V_i^R} + \frac{\partial Y_n^I}{\partial v_j}\frac{\partial v_j}{\partial y_i}\frac{\partial y_i}{\partial V_i^R}\right] \quad (16)$$

$$\frac{\partial E}{\partial V_i^I} = \frac{\partial E}{\partial Y_n^R}\left[\frac{\partial Y_n^R}{\partial u_j}\frac{\partial u_j}{\partial x_i}\frac{\partial x_i}{\partial V_i^I} + \frac{\partial Y_n^R}{\partial u_j}\frac{\partial u_j}{\partial y_i}\frac{\partial y_i}{\partial V_i^R} + \frac{\partial Y_n^R}{\partial v_j}\frac{\partial v_j}{\partial x_i}\frac{\partial x_i}{\partial V_i^I} + \frac{\partial Y_n^R}{\partial v_j}\frac{\partial v_j}{\partial y_i}\frac{\partial y_i}{\partial V_i^I}\right]$$
$$+ \frac{\partial E}{\partial Y_n^I}\left[\frac{\partial Y_n^I}{\partial u_j}\frac{\partial u_j}{\partial x_i}\frac{\partial x_i}{\partial V_i^I} + \frac{\partial Y_n^I}{\partial u_j}\frac{\partial u_j}{\partial y_i}\frac{\partial y_i}{\partial V_i^I} + \frac{\partial Y_n^I}{\partial v_j}\frac{\partial v_j}{\partial x_i}\frac{\partial x_i}{\partial V_i^I} + \frac{\partial Y_n^I}{\partial v_j}\frac{\partial v_j}{\partial y_i}\frac{\partial y_i}{\partial V_i^I}\right] \quad (17)$$

Identifying the partial derivatives of Eqs. (16) and (17) from Eqs. (4), (5) and (10) leads to

$$\frac{\partial E}{\partial V_i} = \delta_n\overline{W}_{nj}\overline{\phi}'(V_i^T(X - C_i))\overline{(X - C_i)} \quad (18)$$

and hence

$$\triangle V_i = \beta\delta_n\overline{W}_{nj}\overline{\phi}'(V_i^T(X - C_i))\overline{(X - C_i)} \quad (19)$$

where β is the learning rate parameter for the weight parameter V_i.

Similar derivation for the update of centers of the hidden neurons gives

$$\frac{\partial E}{\partial C_i} = \nabla C_i E = \frac{\partial E}{\partial C_i^R} + j\frac{\partial E}{\partial C_i^I} \quad (20)$$

$$\frac{\partial E}{\partial C_i^R} = \frac{\partial E}{\partial Y_n^R}\left[\frac{\partial Y_n^R}{\partial u_j}\frac{\partial u_j^R}{\partial x_i}\frac{\partial x_i}{\partial C_i^R} + \frac{\partial Y_n^R}{\partial u_j}\frac{\partial u_j}{\partial y_i}\frac{\partial y_i}{\partial C_i^R} + \frac{\partial Y_n^R}{\partial v_j}\frac{\partial v_j}{\partial x_i}\frac{\partial x_i}{\partial C_i^R} + \frac{\partial Y_n^R}{\partial v_j}\frac{\partial v_j}{\partial y_i}\frac{\partial y_i}{\partial C_i^R}\right]$$

[a]Henceforth, $\underline{V}_i$, $\underline{C}_i$ and $\underline{X}$ have been replaced with V_i, C_i and X respectively, for convenience.

$$+\frac{\partial E}{\partial Y_n^I}\left[\frac{\partial Y_n^I}{\partial u_j}\frac{\partial u_j}{\partial x_i}\frac{\partial x_i}{\partial C_i^R}+\frac{\partial Y_n^I}{\partial u_j}\frac{\partial u_j}{\partial y_i}\frac{\partial y_i}{\partial C_i^R}+\frac{\partial Y_n^I}{\partial v_j}\frac{\partial v_j}{\partial x_i}\frac{\partial x_i}{\partial C_i^R}+\frac{\partial Y_n^I}{\partial v_j}\frac{\partial v_j}{\partial y_i}\frac{\partial y_i}{\partial C_i^R}\right] \quad (21)$$

$$\frac{\partial E}{\partial C_{ij}^I}=\frac{\partial E}{\partial Y_n^R}\left[\frac{\partial Y_n^R}{\partial u_j}\frac{\partial u_j}{\partial x_i}\frac{\partial x_i}{\partial C_i^I}+\frac{\partial Y_n^R}{\partial u_j}\frac{\partial u_j}{\partial y_i}\frac{\partial y_i}{\partial C_i^I}+\frac{\partial Y_n^R}{\partial v_j}\frac{\partial v_j}{\partial x_i}\frac{\partial x_i}{\partial C_i^I}+\frac{\partial Y_n^R}{\partial v_j}\frac{\partial v_j}{\partial y_i}\frac{\partial y_i}{\partial C_i^I}\right]$$
$$+\frac{\partial E}{\partial Y_n^I}\left[\frac{\partial Y_n^I}{\partial u_j}\frac{\partial u_j}{\partial x_i}\frac{\partial x_i}{\partial C_i^I}+\frac{\partial Y_n^I}{\partial u_j}\frac{\partial u_j}{\partial y_i}\frac{\partial y_i}{\partial C_i^I}+\frac{\partial Y_n^I}{\partial v_j}\frac{\partial v_j}{\partial x_i}\frac{\partial x_i}{\partial C_i^I}+\frac{\partial Y_n^I}{\partial v_j}\frac{\partial v_j}{\partial y_i}\frac{\partial y_i}{\partial C_i^I}\right] \quad (22)$$

Identifying the partial derivatives of Eqs. (21) and (22) from Eqs. (4), (5) and (10), it can be derived that

$$\triangle C_i=-\eta\delta_n\overline{W}_{nj}\overline{\phi}'(V_i^T(X-C_i))\overline{V}_i \quad (23)$$

where η is the learning rate parameter for the neuron centers. Thus, the complex-valued gradient update rule for the three parameters, (W, C and V) of the FC-RBF network discussed in Sec. 2 is given by

$$\triangle W_{nj}=\alpha\overline{O_h}^j\delta_n \quad (24)$$

$$\triangle V_i=\beta\delta_n\overline{W}_{nj}\overline{\phi}'(V_i^T(X-C_i))\overline{(X-C_i)} \quad (25)$$

and

$$\triangle C_i=-\eta\delta_n\overline{W}_{nj}\overline{\phi}'(V_i^T(X-C_i))\overline{V}_i \quad (26)$$

where α, β and η are the learning rate parameters, which can be real, imaginary or complex. These parameters affect the convergence of the FC-RBF network. With higher learning rate parameters, the network fails to converge. As will be shown in Sec. 4, the size of the network and the choice of initial parameters affect the performance of the network. In this study, initially, the number of hidden neurons were arbitrarily chosen and the initial centers were chosen using a procedure similar to Ref. 32. The search for the size of the network and the initial centers for best performance, using this method, is an extensive, laborious procedure. To improve the performance with lesser search time and effort, the K-Means Clustering (KMC) algorithm, which is widely used in selection of centers for RBF networks,[35] is used.

3.1. *Network initialization: The K-Means Clustering algorithm*

The KMC algorithm that is used in the study has been summarized below:

Step 1: Begin with N neurons.
Step 2: Choose N random samples from the training set and assign them as the neuron centers.
Step 3: For each complex-valued input of the training set, find the Euclidean distance from each of the randomly chosen centers

$$d_{ij}=\sqrt{(\underline{X}-\underline{C}_j)\overline{(\underline{X}-\underline{C}_j)}} \quad (27)$$

where X is the input and C_j is the center of the jth hidden neuron.
Step 4: The training samples are clustered depending on the minimum d_{ij} from the centers.
Step 5: Calculate the center of each hidden neuron as the average of the samples belonging to that particular cluster. The width of each cluster is also calculated.
Step 6: After clustering, if the distance between two neuron centers is lesser than the width of either of the cluster, then merge them into single neuron.
Step 7: Repeat steps 3 to 6 until there is no change in cluster center.

4. Performance Evaluation of the FC-RBF Algorithm

The performance of the proposed FC-RBF with the well defined complex-valued gradient descent learning algorithm is studied in this section, using a number of problems. The convergence study of the network was first studied using the complex XOR problem. This is followed by a study on the function approximation ability of the algorithm, using a synthetic example. Then, the two spiral problem, which is a highly non-linear classification problem, is used to evaluate the performance of the network, in classification problems. Finally, the algorithm was applied in two real-world applications, and their advantages are highlighted. In all these problems, the results of the FC-RBF network are compared against CRBF with real activation function,[27] CMRAN[28] with Gaussian hidden neurons and CELM.[30]

In the following study, the mean square error for magnitude and phase are as defined in Eq. (28).[15]

$$J_{Me} = \frac{1}{N_t}\sum_{j=1}^{N_t}((e_i \cdot \overline{e}_i))$$

and

$$\Phi_e = \frac{1}{N_t} \mid [\arg(z) - \arg(\hat{z})] \mid \qquad (28)$$

where e_i is the error of the ith output neuron and N_t is the number of samples.

4.1. *Convergence study: The complex XOR problem*

The complex-valued XOR (CXOR) is a commonly used problem in literature to study the convergence of the network.[37] In complex domain, the XOR problem is defined as:

- The real part of the output is unity if the first input is equal to the second input, else it is zero.
- The imaginary part of the output is unity if the second input is equal to unity, else it is zero.

The centers of the FC-RBF algorithm were initialized from the set $[\pm 1, \pm 1]$, as the target of the CXOR problem is one of 0, 0+i, 1, 1+i. 3000 training and 1000 testing samples were randomly generated, and the number of epochs required for the network to converge were observed in 50 successive, independent training trials. The best learning rate and initializations were also observed. Here, convergence is defined as meeting the preset 1×10^{-3} mean square error (J_{Me}) level within 5,000 epochs. The best performances of the FC-RBF network and the complex-valued RBF network (with Gaussian activation function) are tabulated in Table 1.

Table 1. Learning statistics for CXOR problem.

Network	NH	Best η	Convergence		
			Maximum	Minimum	Average
CRBF	8	0.001	5000	1660	4254.11
FC-RBF	8	0.001	624	24	113.66
FC-RBF with KMC	5*	0.001	685	47	193.55

*Approximate average number of neurons for 50 training trials.

CMRAN, being a sequential learning algorithm, was excluded from the convergence study of CXOR problem. CELM, on the other hand, is an analytical learning algorithm and does not tune the parameters of the network. Hence, it was also excluded from the convergence study. It can be observed from the table that with proper choice of the initial parameters,the FC-RBF network required fewer number of epochs, than the CRBF network, to achieve convergence. With the use of KMC for initialization, the network size was reduced considerably, although at the cost of few more epochs, the least being 3 hidden neurons. On an average, approximately 5 hidden neurons were selected through KMC, over 50 independent training trials.

4.2. *Function approximation: A synthetic example*

The synthetic complex-valued function approximation problem I defined in Ref. 15 as

$$Z = Z_1^2 + Z_2^2 \qquad (29)$$

where $Z_1, Z_2 = [\pm 2.5, \pm 2.5]$ is considered to study the approximation ability of the network. 3000 and 1000 randomly chosen samples are used as the training and testing set, respectively. The training sample set was trained over 5000 epochs. Extensive search for the optimal number of neurons required (h), learning rates (α, β, η) to obtain a optimal performance was done. Initially, the selection of number of hidden neurons h was done according to Ref. 32 and the initialization of centers and weights were chosen through an extensive search on the basis of random selection. Best values of learning rate parameters α, β and η were chosen by varying them between 10^{-6} and 10^{-1}, in orders of 10. It was observed that the best performance for each initialization was obtained at the values of $\alpha = 10^{-3}$, $\beta = 10^{-3}$ and $\eta = 10^{-2}$. From Table 2, it can be observed that the network is sensitive to initializations of centers and weights. Also, the search for the choice of hidden neurons and the best initialization, with this method, was a lengthy procedure. So, KMC algorithm, explained in Sec. 3.1, was used to identify the number of hidden neurons required and the best initialization of the weights and centers. Beginning with 25 neurons, the KMC algorithm downsized the clusters to match the requirement based on the inputs.

Table 2. Effect of Initialization on the FC-RBF net work in function Approximation Problem.

h	Initialization			Training		Testing	
	V_r	C_r	W_r	J_{Me}	ϕ_e	J_{Me}	ϕ_e
15	0.01	0.01	0.001	0.004	0.067	0.007	0.2098
15	0.1	0.1	0.001	0.002	0.045	0.004	0.1353
15	1	1	0.001	0.002	0.003	0.004	0.0163
15	2	2	0.001	0.0019	0.0059	0.003	0.006
20	0.01	0.01	0.001	0.012	0.162	0.021	0.4857
20	0.1	0.1	0.001	0.004	0.035	0.004	0.0353
20	1	1	0.001	0.001	0.007	0.001	0.0070
20	2	2	0.001	0.00185	0.0066	0.002	0.008

Performances of the proposed network were compared against the networks available in literature viz., CRBF (with real-valued activation function), CMRAN and CELM and the performance results are presented in Table 3. It was observed that even with fewer neurons, the proposed network approximated the function with lesser magnitude error (J_{Me}) and phase error (ϕ_e). The approximation performance was improved at least by 10%. The table also gives the performance of the network initialized with the KMC. With fewer hidden neurons, chosen using the KMC algorithm, there was no substantial compromise in the performance.

4.3. *Classification: The two-spiral problem*

The two-spiral problem is a famous benchmark problem, because it is an extremely difficult nonlinear classification problem available in literature.[38] Though several variations of the two-spiral problem have been briefly discussed,[39] the original two-spiral classification problem has been considered in this study.

The original two-spiral data can be generated using a C-routine, available in the CMU repository. The number of data points and the range of values generated are determined by the parameters, density and radius respectively. Here, 194 training points are generated with the default parameters of density 1 and radius 6.5. The 194 data points are equally shared between the two spirals (97 data points each). Each spiral has three turns of 32 points per turn, and one end point. Both the spirals have the same origin and orientation, but are displaced by 180 degrees to each other. The first spiral is generated by the set of equations in Eq. (30)

$$\begin{aligned} \phi &= \frac{i}{16} \cdot \pi \\ r &= \frac{d.(104 - i)}{104]} \\ x &= r\ \cos\phi; \quad y = r\ \sin\phi \\ z &= x + iy \end{aligned} \tag{30}$$

The second spiral can be obtained by plotting $(-x, -y)$ and hence $z1 = -x - iy$. The problem at hand is to classify the points into their respective spirals, which makes up a highly non-linear two class classification problem of complex-valued data, with one input neuron and one output neuron. Rigorous study was done to identify the number of neurons required to achieve 100% classification accuracy. The training and testing statistics of the problem is summarized in Table 4. Figure 4 gives the spiral classification result of the networks. It is evident from the table that the FC-RBF performed the highly non-linear classification with fewer neurons compared to the CMRAN, CRBF and CELM networks. The training time of CELM is the least of all networks, as the parameters are determined analytically. However, it can also be observed from

Table 3. Performance comparison for function approximation problem.

Network	η	h	Training		Testing	
			J_{Me}	ϕ_e	J_{Me}	ϕ_e
CRBF	0.001	20	0.592	0.791	0.623	0.823
CMRAN	0.01	27	0.0594	0.2423	0.0614	0.3287
CELM	—	20	0.6886	0.609	0.704	0.631
FC-RBF	0.001	15	0.0019	0.0059	0.003	0.006
FC-RBF with KMC	0.001	11	0.00142	0.00971	0.00137	0.003388

Table 4. Performance comparison for two-spiral classification problem.

Network	h	Epochs	Training time	Classification accuracy (%)
CRBF	100	7226	10865.21	100
CMRAN	130	—	12.766	100
CELM	120	—	0.3281	100
FC-RBF	80	7829	3183.6	100
FC-RBF with KMC	70	5126	2102.89	100

the table that it required more network resources, compared to the proposed FC-RBF network. The KMC, required fewer neurons and hence, network resources, and lesser time, to achieve 100% efficiency of classification.

4.4. *Real-world problems*

4.4.1. *Non-minimum phase equalization problem*

In this section, a well-known complex nonminimum-phase channel model introduced by Cha and Kassam[27] is used to evaluate the performance of the FC-RBF equalizer, and to compare it's performance against the CRBF, CMRAN and CELM equalizers. The equalization model is of order three, with nonlinear distortion for 4-QAM signaling. The channel output z_n (which is also the input to the equalizer) is given by

$$z_n = o_n + 0.1o_n^2 + 0.05o_n^3 + v_n, v_n \aleph(0, 0.01)$$

$$o_n = w_1 s_n + w_2 s_{n-1} + w_3 s_{n-2}$$

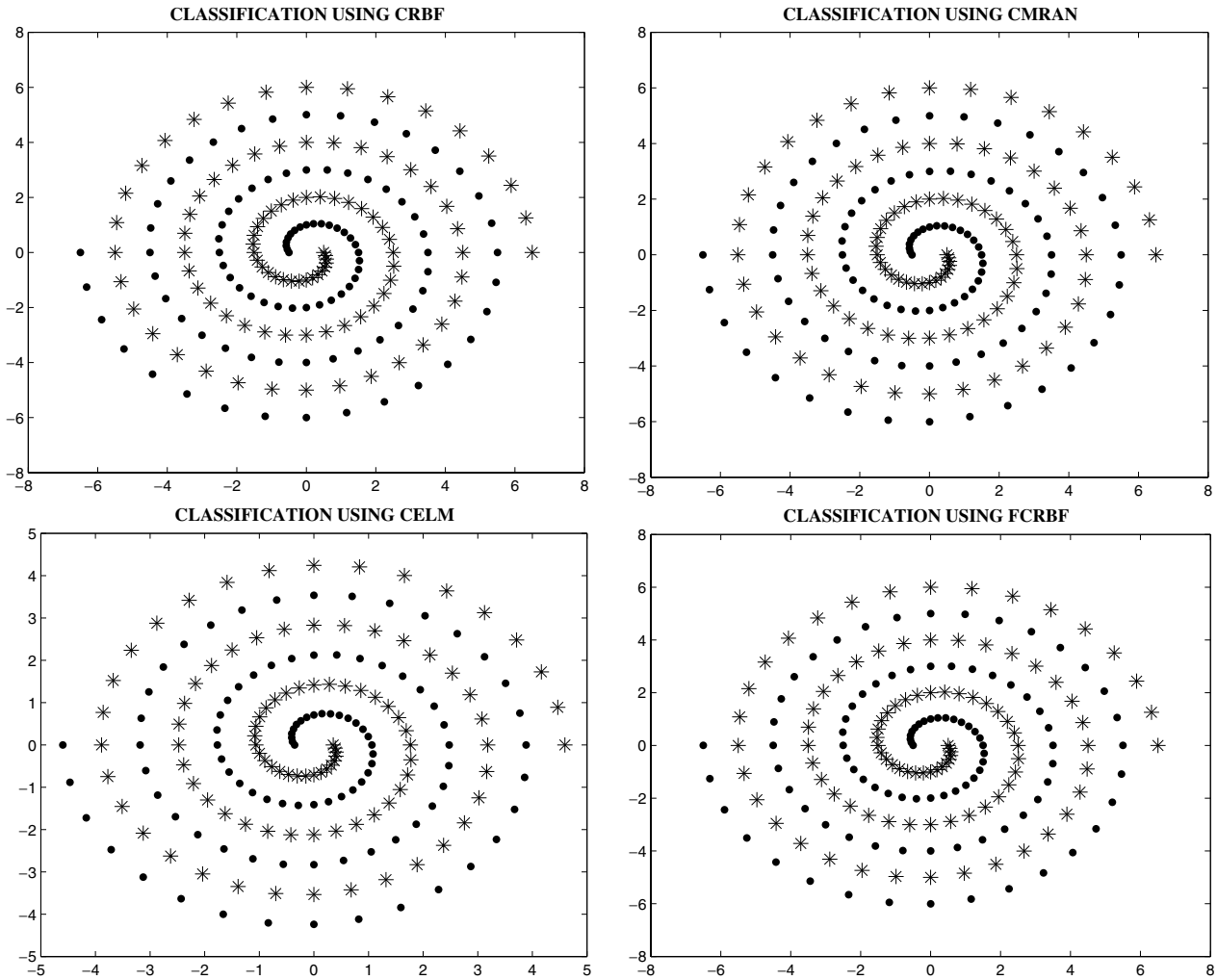

Fig. 4. Two-spiral classification results.

where $w_i = 0.34 - 0.27i$, $w_2 = 0.87 + 0.43i$, $w_3 = 0.34 - 0.21i$, $\aleph$ (0, 0.01) means the white Gaussian noise (of the non-minimum phase channel) with mean 0 and variance 0.01. The equalizer input dimension is chosen as 3. As usually done in equalization problems, a decision delay τ is introduced in the equalizer so that at time n, the equalizer estimates the input symbol $s_{n-\tau}$ rather than s_n and we set $\tau = 1$. 4-QAM symbol sequence s_n is passed through the channel and the real and imaginary parts of the symbol are valued from the set ± 0.7. 5000 randomly generated samples at $20dB$ SNR was used to train the network over 1000 epochs. Studies were conducted to identify the optimum size of network (h), learning rates (η) and the initial weight/center radius using the heuristic procedure similar to Ref. 32. Then the KMC was used to fix the network structure and the initial centers. It was observed that the reduced network performed symbol classification equally well. Table 5 gives the performance comparison of the proposed fully complex-valued RBF network against the other existing complex-valued RBF networks.

The superiority in the generalization performance of the FC-RBF network can be observed from the Table 5. It can be observed from the table that the testing errors using the FC-RBF network are almost half of those in the CELM and CRBF networks. The error in phase has been reduced considerably. Also, with proper selection of parameters in the FC-RBF network through KMC, the phase error is reduced further, with fewer neurons. The classification performance can be observed from the plot of the Symbol Error Rate (SER) curve shown in Fig. 5. It can be seen that the FC-RBF has improved the symbol classification performance by a margin of at least 0.4 dB SER at a noise level of 16 dB SNR. Initialization of the FC-RBF network with KMC improved the SER performance of the network further, by a margin of another 0.2 dB SER at a noise level of 16 dB SNR. Besides these, from the comparison of symbol classification performance, shown in Fig. 6, it can be observed that the network classified the symbols in their quadrants much better than the other networks.

Table 5. Performance comparison for non-minimum phase equalization problem.

Network	h	Training		Testing	
		J_{Me}	ϕ_e	J_{Me}	ϕ_e
CRBF	15	0.5630	0.6142	0.5972	0.6956
CMRAN	40	0.4153	0.305	1.3114	2.0053
CELM	15	0.5720	0.5958	0.5772	0.6128
FC-RBF	15	0.3700	0.5440	0.4142	0.2564
FC-RBF with KMC	14	0.3399	0.1079	0.3476	0.1092

4.4.2. *Adaptive beam-forming problem*

Beam-forming is an array signal processing problem of forming a beam pattern of an array of sensors. In doing so, beams are directed to the desired direction (beam-pointing) and the nulls are directed to interference direction (null-steering).[40] The beam-forming antenna array consists of M single-transmit antenna of the same carrier frequency and N uniformly spaced elements at the receiver. In this paper, a Uniform Linear Array (ULA) of sensors, similar to the one shown in Fig. 7, is considered.

The spacing between elements, d, equals half the wavelength of the received signal. If θ is the angle of incidence that an incoming signal makes with the receiver array broadside, it can be derived from the basic trigonometric identities and the geometry of the sensor array that the signal received at the nth element of the receiver antenna array is given by Eq. (31),

$$\bar{x}_n = e^{\frac{j2\pi n d \sin\theta}{\lambda}} \tag{31}$$

Including a noise of η_n, signals induced at the N receiver array elements due to the m sources, impinging on each array element will be,

$$X = [\bar{x}_0 + \eta_0 \bar{x}_1 + \eta_1 \bar{x}_2 + \eta_2 \cdots \bar{x}_N + \eta_N]^T \tag{32}$$

This is the input to the beam-former. In Eq. (32), the superscript T represents the transpose of the vector.

Let w_n is the weight of the nth sensor, W be the array of weights such that, $W = [w_1 w_2 \cdots w_n]^{T'}$ and G be the gain of the beam-former array for the signal from source m.

$$G = \sum_{n=1}^{N} x_n w_n = X^T W \tag{33}$$

w_n can now be computed by setting $G = 0$ for null-steering and $G = 1$ for beam pointing. The optimum weight vector W would, now, be:

$$W = X^{-1} G \tag{34}$$

This analytical method of computing the optimum weight is known as the matrix method.

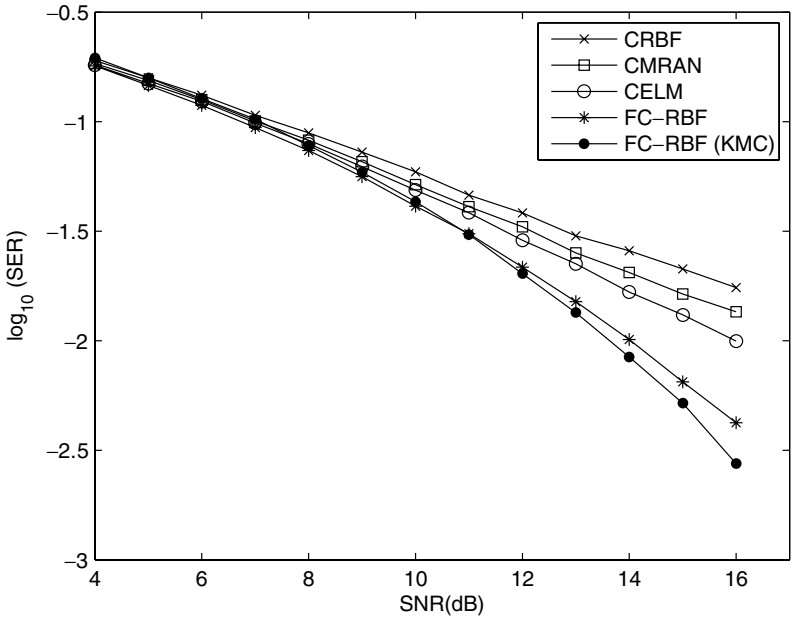

Fig. 5. Error probability curve.

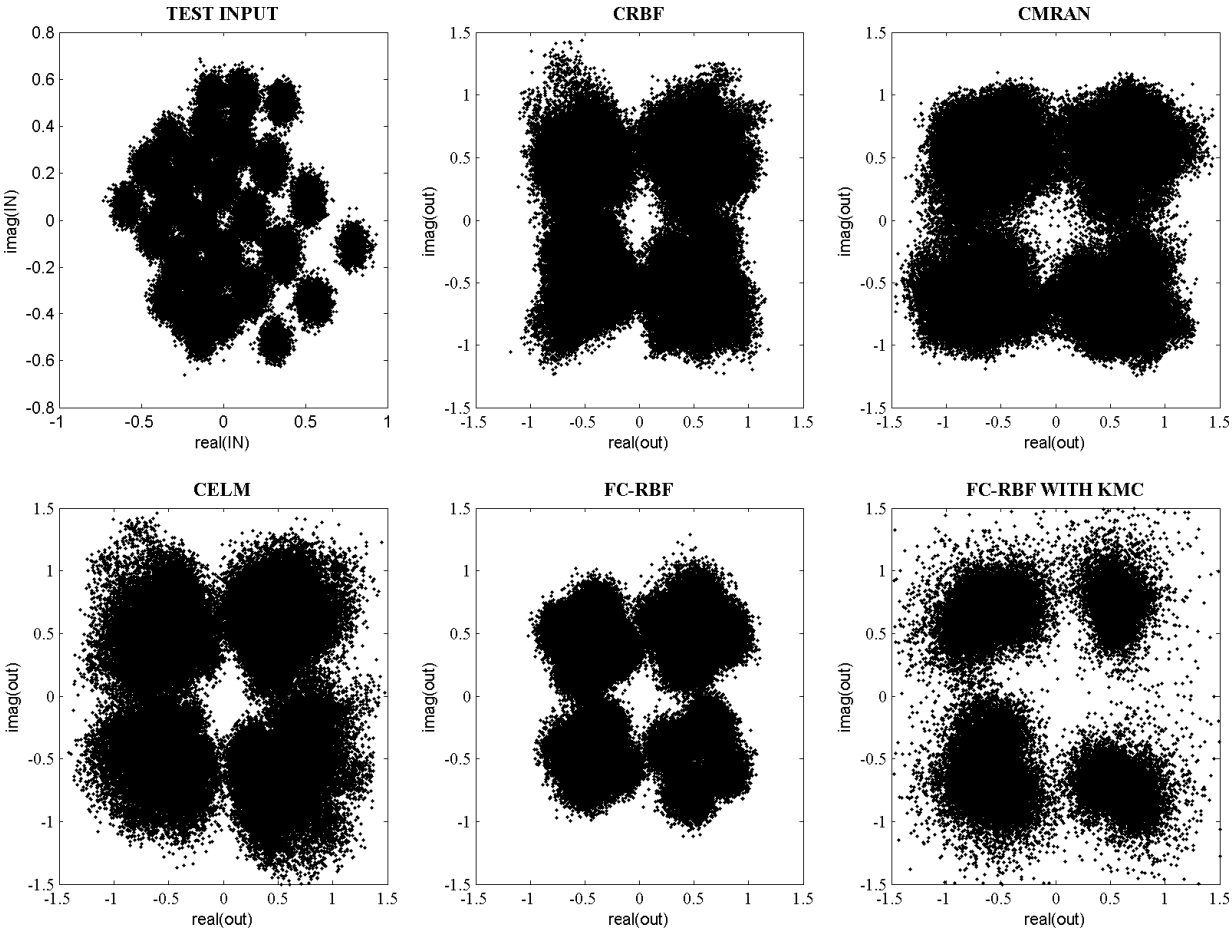

Fig. 6. Eye diagram of symbol classification performance of different equalizers.

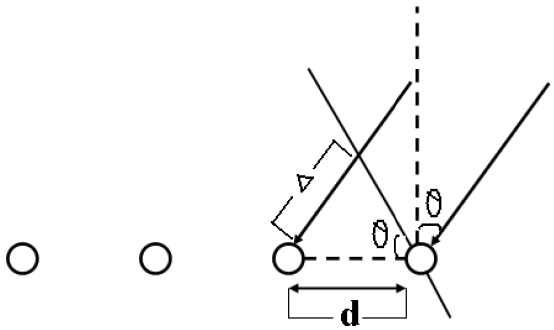

Fig. 7. The adaptive beam-forming sensor array.

This becomes, computationally expensive,[41] especially, when dealing with large arrays, or when the Angle of Arrival is not known exactly. Therefore neural networks, being more adaptive,[42] are employed to estimate the weights. The objective is to train the neural network, such that the weights W of Eq. (34) are estimated. Here, Y is the signal transmitted, and X is the signal received at the sensor array, given by Eq. (32).

$$Y = W^H X \tag{35}$$

In this paper, a Uniform Linear Array (ULA) of 5 sensors is considered. The array is trained to look at desired signals coming from $-30°$ and $+30°$, and to suppress interferences from $-15°$, $0°$ and $+15°$ directions.[40] The input to the network is given by Eq. (32), with an additive Gaussian noise of 50 dB SNR. Training data set consists of 250 randomly chosen samples (50 for each signal/interference angles). Figure 8 gives the beam-pattern before

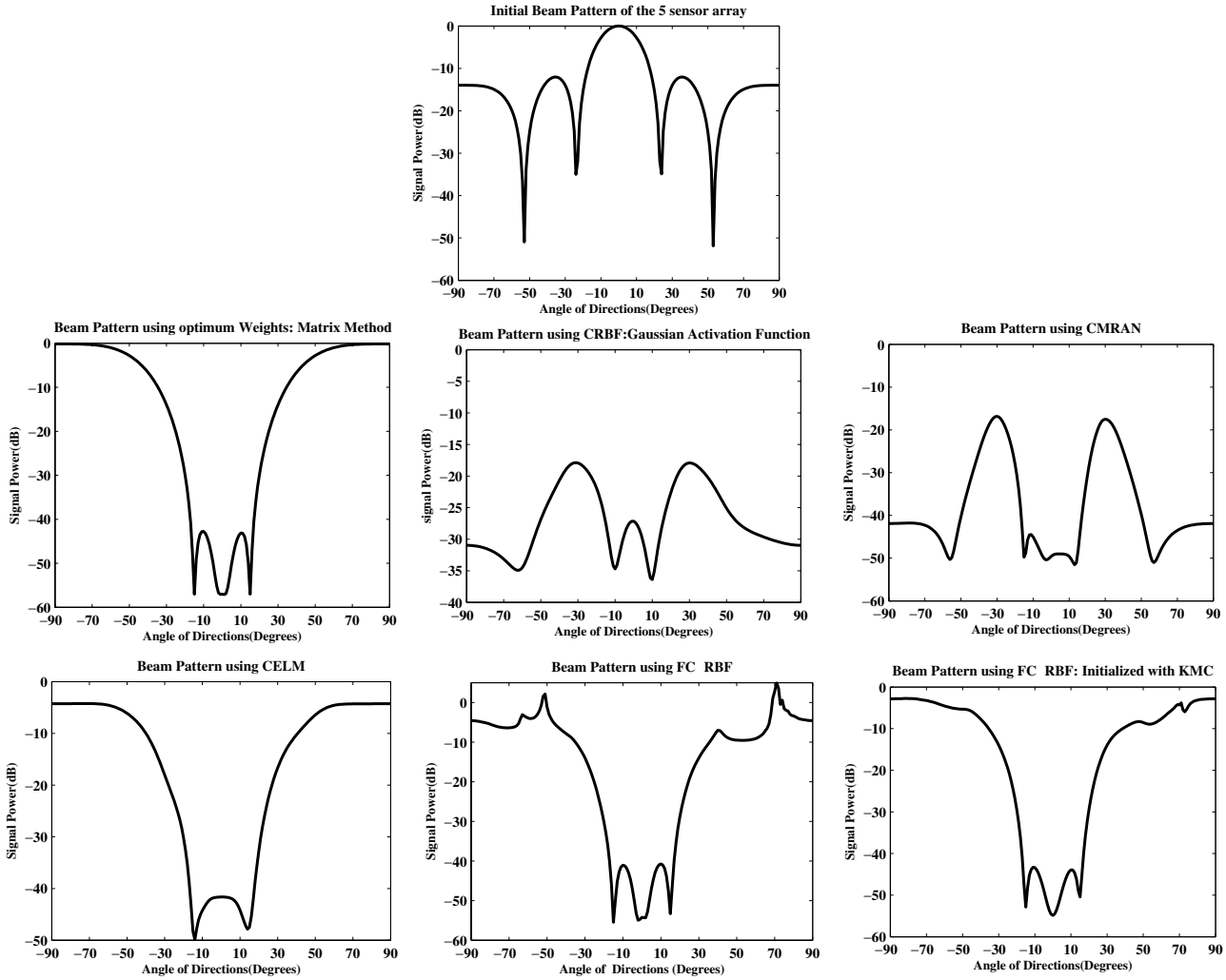

Fig. 8. Beam pattern of 5 sensor array.

Table 6. Performance comparison of different complex-valued RBF beamformers.

DOA*	Gain(dB)					
	CRBF	CMRAN	CELM	FC-RBF	FC-RBF with KMC	MM**
Beam-1:−30°	−17.94	−16.84	−18.05	−16.99	−13.98	−13.98
Null-1:−15°	−27.53	−49.77	−48.28	−58.45	−52.86	−57.016
Null-2:0°	−27	−49.6	−41.64	−57.23	−54.86	−57
Null-3:15°	−28.33	−48.18	−47.5	−56.32	−50.45	−57.016
Beam-2:30°	−17.92	−17.52	−16.68	−17.00	−13.96	−13.98

Note: *Direction of arrival; **Matrix method.

beamforming/null-steering, beam-pattern using the matrix method beam-former (optimum method) and the beam patterns after beam-forming using different complex-valued RBF neural network beamformers-viz., CRBF, CMRAN, CELM and FC-RBF. While a 5-5-1(5-input neurons, 5-hidden neurons and 1 output neuron) networks were used for CRBF, CELM and FC-RBF networks, the CMRAN chose 32 neurons to perform beam-forming, by growing and pruning. It can be observed that the FC-RBF beamformer performed better isolation of nulls, with gains at nulls closer to the optimum values.

Table 6 gives the amplification gains at beam-pointings and attenuation gains at null-steering for the different methods considered. The table clearly shows that the FC-RBF network performs better, in comparison to the other complex-valued RBF networks. The selection of centers using KMC for the FC-RBF network further improved the beam pointing and the gains at the desired signal angles are quite similar to the optimum gains.

From this performance study, it can be inferred that the FC-RBF network does better mapping of signals and isolation of interferences or nulls, better than other complex-valued RBF networks, available in literature. Besides, it was also observed during the study that the convergence speed of the FC-RBF network, during training, is faster than the CRBF network. Based on all these studies, the main results are summarized in the next section.

4.5. *Summary of the performance study results*

- From the convergence study on the CXOR problem, it was observed that the proposed network required fewer epochs to converge to an error of 10^{-3}.
- Based on the training/testing errors of function approximation problem, it can be seen that the proposed network improves the performance of function approximation with at least, 10% reduction in magnitude and phase errors, compared to other existing complex-valued RBF algorithms.
- The classification performance of the algorithm was observed via the two spiral classification problem, where the proposed network required atleast 20 fewer neurons to achieve a classification accuracy of 100%.
- Study on the non-minimum phase equalization problem showed that the symbol classification performance of the proposed FC-RBF network is much better than the other networks, by a margin of at least 0.4 dB SER at a noise level of 16 dB SNR. The KMC improved the performance, further, by a margin of another 0.2 dB SER at a noise level of 16dB SNR, with fewer neurons.
- Comparing the performances of adaptive beamforming with the optimum Matrix method, it was observed that the beam pattern generated with the FC-RBF beam-former performs closer to the beam-pattern of the matrix method, than the other available complex-valued RBF networks.
- Initializations with the K-means clustering algorithm showed that the network performance remain largely unaffected, even with a fewer neurons in the hidden layer.

5. Conclusion

In this paper, a fully complex radial basis function (FC-RBF) network with a fully complex-valued activation function has been proposed. Also, its complex-valued gradient descent learning algorithm has been

developed. The fully complex activation function of the proposed network, satisfies all the properties needed for a complex-valued activation function and has Gaussian-like characteristics. It maps $C^n \rightarrow C$, unlike the existing activation functions of CRBF network that maps $C^n \rightarrow R$. Initially, the network architecture and initialization were chosen using a heuristic procedure. As the performance of the network depended on the initialization heavily, the K-means clustering algorithm for complex-valued signals was proposed, and the architecture chosen using this algorithm. The performance of the network was compared with other well-known complex-valued RBF networks available in literature, viz., split-complex CRBF, CMRAN and the CELM on a few benchmark and real-world problems. The study results showed the advantages of the proposed network, as shown in Sec. 4.5.

Acknowledgment

The authors would like to thank the reviewer for his insightful comments and suggestions.

References

1. V. S. Chakravarthy, N. Gupte, S. Yogesh and A. Salhotra, Chaotic synchronization using a network of neural oscillators, *International Journal of Neural Systems* **18**(2) (2008) 157–164.
2. S. Fiori, Learning by criterion optimization on a unitary unimodular matrix group, *International Journal of Neural Systems* **18**(2) (2008) 87–103.
3. T. Nitta, The uniqueness theorem for complex-valued neural networks with threshold parameters and the redundancy of the parameters, *International Journal of Neural Systems* **18**(2) (2008) 123–134.
4. S. Buchholz and N. L. Bihan, Polarized signal classification by complex and quaternionic multilayer perceptron, *International Journal of Neural Systems* **18**(2) (2008) 75–85.
5. V. S. H. Rao and G. R. Murthy, Global dynamics of a class of complex valued neural networks, *International Journal of Neural Systems* **18**(2) (2008) 165–171.
6. A. J. Noest, Phasor neural networks, *Neural Information Processing Systems*, in D. Z. Anderson (ed.), American Institute of Physics, New York, 1998 584–591.
7. M. Kobayashi, Pseudo-relaxation learning algorithm for complex-valued associative memory, *International Journal of Neural Systems* **18**(2) (2008) 147–156.
8. S. Kawata and A. Hirose, Frequency-multiplexing ability of complex-valued Hebbian learning in logic gates, *International Journal of Neural Systems* **18**(2) (2008) 173–184.
9. T. Isokawa, H. Nishimura, N. Kamiura and N. Matsui, Associative memory in quaternionic Hopfield neural network, *International Journal of Neural Systems* **18**(2) (2008) 135–145.
10. H. Leung and S. Haykin, Complex backpropagation algorithm, *IEEE Trans. on Signal Processing* **39**(9) (1991) 2101–2104.
11. D. Vigliano, M. Scarpiniti, R. Parisi and A. Uncini, Flexible nonlinear blind signal separation in the complex domain, *International Journal of Neural Systems* **18**(2) (2008) 105–122.
12. D. P. Mandic, P. Vayanos, M. Chen and S. L. Goh, Online detection of the modality of complex-valued real world signals, *International Journal of Neural Systems* **18**(2) (2008) 67–74.
13. T. Kim and T. Adali, Approximation by fully complex multilayer perceptrons, *Neural Computation* **15**(7) (2003) 1641–1666.
14. T. Kim and T. Adali, Fully-complex multilayer perceptron network for nonlinear signal processing *Journal of VLSI, Signal Processing*, **32**(1–2) (2002) 29–43.
15. R. Savitha, S. Suresh, N. Sundararajan and P. Saratchandran, Complex-valued function approximation using an improved BP learning algorithm for feedforward networks, *International Joint Conference on Neural Networks 2008 (World Congress on Computational Intelligence)* (Hong Kong) (2008) 2251–2258.
16. G. M. Georgios and C. Koutsougeras, Complex domain backpropagation, *IEEE Trans. on Circuit and Systems II* **39**(5) (1992) 330–333.
17. C. You, and D. Hong, Nonlinear blind equalization schemes using complex valued multilayer feedforward neural networks, *IEEE Trans. on Neural Networks* **9**(6) (1998) 1442–1455.
18. A. Karim and H. Adeli, Radial basis function neural network for work zone capacity and queue estimation, *Journal of Transportation Engineering* **129**(5) (2003) 494–503.
19. H. Adeli and A. Karim, Fuzzy-wavelet RBFNN model for freeway incident detection, *Journal of Transportation Engineering* **126**(6) (2000) 464–471.
20. S. Ghosh-Dastidar, H. Adeli and N. Dadmehr, Principal component analysis enhanced cosine radial basis function network for robust epilepsy and seizure detection, *IEEE Trans. on Biomedical Engineering* **55**(2) (2008) 512–518.
21. A. Karim and H. Adeli, Comparison of the fuzzy-wavelet RBFNN freeway incident detection model with the california algorithm, *Journal of Transportation Engineering* **128**(1) (2002) 21–30.
22. T. D. Jogensen, B. P. Haynes and C. C. F. Norlund pruning artificial neural networks using neural

complexity measures, *International Journal of Neural Systems* **18**(5) (2008) 389–403.
23. R. V. Mayorga and J. Carrera, A radial basis function network approach for the computational of inverse continuous time variant functions, *International Journal of Neural Systems* **17**(3) (2007) 149–160.
24. W. Pedrycz, R. Rai and J. Zurada, Experience-consistent modeling for radial basis function neural networks, *International Journal of Neural Systems* **18**(4) (2008) 279–292.
25. S. Chen, S. McLaughlin and N. Mulgrew, Complex-valued radial basis function network, Part I: Network architecture and learning algorithms, *EURASIP Signal Processing Journal* **35**(1) (1994) 19–31.
26. S. Chen, S. McLaughlin and B. Mulgrew, Complex-valued radial basis function network, Part II: application to digital communications channel equalization, *EURASIP Signal Processing Journal* **36**(2) (1994) 175–188.
27. I. Cha and S. A. Kassam, Channel equalization using adaptive complex radial basis function networks, *IEEE Journal on Selected Areas in Communications* **13**(1) (1995) 122–131.
28. J. P. Deng, N. Sundararajan and P. Saratchandran, Communication channel equalization using complex-valued minimal radial basis function neural networks, *IEEE Trans. on Neural Networks* **13**(3) (2002) 687–696.
29. M.-B. Li, G.-B. Huang, P. Saratchandran and N. Sundararajan, Complex-valued growing and pruning RBF neural networks for communication channel equalization, *IEE Proceedings. Vision, Image and Signal Processing* **153**(4) (2006) 411–418.
30. M.-B. Li, G.-B. Huang, P. Saratchandran and N. Sundararajan, Fully complex extreme learning machines, *Neurocomputing* **68** (2005) 306–314.
31. R. Savitha, S. Suresh and N. Sundararajan, Complex-valued function approximation using a fully complex-valued RBF (FC-RBF) learning algorithm, *Presented in Proc. of International Joint Conference on Neural Networks 2009 (IJCNN 2009)*, Atlanta (Georgia, USA, 2009).
32. S. Suresh, S. N. Omkar, V. Mani and T. N. Guru Prakash, Lift coefficient prediction at high angle of attack using recurrent neural network, *Aerospace Science and Technology* **7**(8) (2003) 595–602.
33. S. Ghosh-Dastidar and H. Adeli, Wavelet-clustering-neural network model for freeway incident detection, *Computer-Aided Civil and Infrastructure Engineering* **18**(5) (2008) 325–338.
34. L. Xu and B. YingYang Machine, Clustering and number of clusters, *Pattern Recognition Letters* **18** (1997) 1167–1178.
35. R. Rollet, G. B. Benie, W. Li and S. Wang J.-M. Boucher, Image classification algorithm based on the RBF neural network and K-means, *International Journal of Remote Sensing* **19**(15) (1998) 3003–3009.
36. H. Liu, X. Wang and W. Qiang, A fast method for implicit surface reconstruction based on radial basis functions network from 3D scattered points, *International Journal of Neural Systems* **17**(6) (2007) 459–465.
37. T. Nitta, An extension of the back-propagation algorithm to complex numbers, *Neural Networks* **10**(8) (1997) 1391–1415.
38. M. L. Minsky and S. Papert, *Perceptrons: An Introduction to Computational Geometry (Expanded Edition)* (Cambridge, MA, USA: The MIT Press, 1988).
39. S. K. Chalup and L. Wiklendt, Variations of the two-spiral task, *Connection Science* **19**(2) (2007) 183–199.
40. A. B. Suksmono and A. Hirose, Intelligent beamforming by using a complex-valued neural network, *Journal of Intelligent and Fuzzy Systems* **15**(3–4) (2004) 139147.
41. R. A. Monzingo and T. W. Miller, *Introduction to Adaptive Arrays* (SciTech Publishing, 2003).
42. K.-L. Dua, A. K. Y. Laib, K. K. M. Chengb and M. N. S. Swamya, Neural methods for antenna array signal processing: A review, *Signal Processing* **82**(4) (2002) 547–561.