

# Hybrid Training Algorithm for RBF Network

By

**M. Y. MASHOR**

School of Electrical and Electronic Engineering,  
University Science of Malaysia,  
Perak Branch Campus,  
31750 Tronoh, Perak,  
Malaysia.  
E-mail: yusof@eng.usm.my

## Abstract

This study presents a new hybrid algorithm for training RBF network. The algorithm consists of a proposed clustering algorithm to position the RBF centres and Givens least squares to estimate the weights. This paper begins with a discussion about the problems of clustering for positioning RBF centres. Then a clustering algorithm called moving  $k$ -means clustering algorithm was proposed to reduce the problems. The performance of the algorithm was then compared to adaptive  $k$ -means, non-adaptive  $k$ -means and fuzzy  $c$ -means clustering algorithms. Overall performance of the RBF network that used the proposed algorithm is much better than the ones that used other clustering algorithms. Simulation results also reveal that the algorithm is not sensitive to initial centres.

## 1. Introduction

The performance of radial basis function (RBF) network will be influenced by centre locations of radial basis function. In a regularisation network based on the RBF architecture that was developed by Poggio and Girosi (1990), all the training data were taken as centres. However, this may lead to network overfitting as the number of data becomes large. To overcome this problem a network with a finite number of centres was proposed by Poggio and Girosi (1990). They also showed that the updating rule for RBF centres derived from a gradient descent approach makes the centres move towards the majority of the data. This result suggests that a clustering algorithm may be used to position the centres.

The most widely used clustering algorithm to position the RBF centres is  $k$ -means clustering (Chen et al. 1992, Moody and Darken 1989, Lowe 1989). This choice was inspired by the

simplicity of the algorithm. However,  $k$ -means clustering algorithm can be sensitive to the initial centres and the search for the optimum centre locations may result in poor local minima. As the centres appear non-linearly within the network, a supervised algorithm to locate the centres has to be based on non-linear optimisation techniques. Consequently, this algorithm will also have the same problems as the  $k$ -means clustering algorithm.

Many attempts have been made to minimise these problems (Darken and Moody, 1990, 1992; Ismail and Selim, 1986; Xu et al., 1993; Kamel and Selim 1994). In this paper, an algorithm called *moving  $k$ -means clustering* is proposed as an alternative or improvement to the standard  $k$ -means clustering algorithm. The proposed algorithm is designed to give a better overall RBF network performance rather than a good clustering performance. However, there is a strong correlation between good clustering and the performance of the RBF network.

## 2. Clustering Problems

Most clustering algorithms work on the assumption that the initial centres are provided. The search for the final clusters or centres starts from these initial centres. Without a proper initialisation, such algorithms may generate a set of poor final centres and this problem can become serious if the data are clustered using an on-line clustering algorithm. In general, there are three basic problems that normally arise during clustering:

- Dead centres
- Centre redundancy
- Local minima

Dead centres are centres that have no members or associated data. Dead centres are normally located between two active centres or outside the data range. This problem may arise due to bad initial centres, possibly because the centres have been initialised too far away from the data. Therefore, it is a good idea to select the initial centres randomly from the data or to set the initial centres to some random values within the data range. However, this does not guarantee that all the centres are equally active (i.e. have the same number of members). Some centres may have too many members and be frequently updated during the clustering process whereas some other centres may have only a few members and are hardly ever updated. So a question arises, how does this unbalanced clustering affect the performance of the RBF network and how can this be overcome?

The centres in a RBF network should be selected to minimise the total distance between the data and the centres so that the centres can properly represent the data. A simple and widely used square error cost function can be employed to measure the distance, which is defined as:

$$E = \sum_{j=1}^{n_c} \sum_{i=1}^N (\|v_i - c_j\|)^2 \quad (1)$$

where  $N$ , and  $n_c$  are the number of data and the number of centres respectively;  $v_i$  is the data

sample belonging to centre  $c_j$ . Here,  $\|\bullet\|$  is taken to be an Euclidean norm although other distance measures can also be used. During the clustering process, the centres are adjusted according to a certain set of rules such that the total distance in equation (1) is minimised. However, in the process of searching for global minima the centres frequently become trapped at local minima. Poor local minima may be avoided by using algorithms such as simulated annealing, stochastic gradient descent, genetic algorithms, etc. Though there is a strong correlation between minimising the cost function and the overall performance of the RBF network, it is no guarantee that the minimum cost function solution will always give the best overall network performance (Lowe, 1989). Hence, a better clustering algorithm may consist of a constrained optimisation where the overall classification on the training data is minimised subject to the maximisation of the overall RBF network performance over both the training and testing data sets.

In order to give a good modelling performance, the RBF network should have sufficient centres to represent the identified data. However, as the number of centres increases the tendency for the centres to be located at the same position or very close to each other is also increased. There is no point in adding extra centres if the additional centres are located very close to centres that already exist. However, this is the normal phenomenon in a  $k$ -means clustering algorithm and an unconstrained steepest descent algorithm as the number of parameters or centres becomes sufficiently large (Cichocki and Unbehauen, 1993). Xu et al. (1993) introduced a method called *rival penalised competitive learning* to overcome this problem. The idea is that at each learning step, both the winning centre and its rival (the 2nd winner) are adjusted but the rival will be adjusted in the opposite direction to the winning centre.

### 3. RBF Network with Linear Input Connections

A RBF network with  $m$  outputs and  $n_h$  hidden nodes can be expressed as:

$$y_i(t) = w_{i0} + \sum_{j=1}^{n_h} w_{ij} \phi(\|v(t) - c_j(t)\|); \quad i = 1, \dots, m \quad (2)$$

where  $w_{ij}$ ,  $w_{i0}$  and  $c_j(t)$  are the connection weights, bias connection weights and RBF centres respectively,  $v(t)$  is the input vector to the RBF network composed of lagged input, lagged output and lagged prediction error and  $\phi(\bullet)$  is a non-linear basis function.  $\|\bullet\|$  denotes a distance measure that is normally taken to be the Euclidean norm.

Since neural networks are highly non-linear, even a linear system has to be approximated using the non-linear neural network model. However, modelling a linear system using a non-linear model can never be better than using a linear model. Considering this argument, the RBF network with additional linear input connections is used. The proposed network allows the network inputs to be connected directly to the output node via weighted connections to form a linear model in parallel with the non-linear standard RBF model as shown in Figure 1.

The new RBF network with  $m$  outputs,  $n$  inputs,  $n_h$  hidden nodes and  $n_l$  linear input

connections can be expressed as:

$$y_i(t) = w_{i0} + \sum_{j=1}^{n_l} \lambda_{ij} v_l(t) + \sum_{j=1}^{n_h} w_{ij} \phi(\|v(t) - c_j(t)\|); \quad i = 1, 2, \dots, m \quad (3)$$

where the  $\lambda$ 's and  $v$ 's are the weights and the input vector for the linear connections respectively. The input vector for the linear connections may consist of past inputs, outputs and noise lags. Since  $\lambda$ 's appear to be linear within the network, the  $\lambda$ 's can be estimated using the same algorithm as for the  $w$ 's. As the additional linear connections only introduce a linear model, no significant computational load is added to the standard RBF network training. Furthermore, the number of required linear connections are normally much smaller than the number of hidden nodes in the RBF network. In the present study, Givens least squares algorithm with additional linear input connection features is used to estimate  $w$ 's and  $\lambda$ 's. Refer to reference Chen et. al. (1992) or Mashor (1995) for implementation of Givens least squares algorithm.

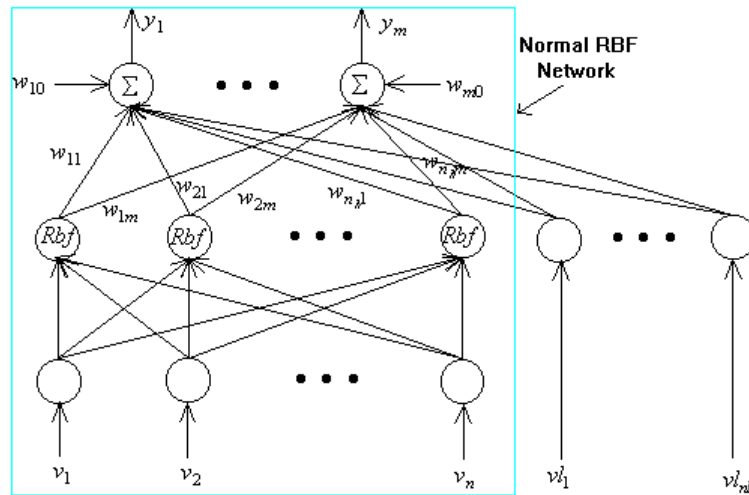


Figure 1. The RBF network with linear input connections

#### 4. The New Hybrid Algorithm

Given a set of input-output data,  $u(t)$  and  $y(t)$ , ( $t = 1, 2, \dots, N$ ), the connection weights, centres and widths may be obtained by minimising the following cost function:

$$J = \sum_{t=1}^N (y(t) - \hat{y}(t))^T (y(t) - \hat{y}(t)) \quad (4)$$

where  $\hat{y}(t)$  is the predicted output generated by using the RBF network given by equation (3). Equation (4) can be solved using a non-linear optimisation or gradient descent technique. However, estimating the weights using such algorithm will destroy the advantage of linearity in the weights. Thus, the training algorithm is normally split into two parts:

- (i) positioning the RBF centres,  $c_j(t)$  and
- (ii) estimating the weights,  $w_{ij}$ .

This approach will allow an independent algorithm to be employed for each task. The centres are normally located using an unsupervised algorithm such as  $k$ -means clustering, fuzzy clustering and Gaussian classifier whereas the weights are normally estimated using a class of linear least squares algorithm. Moody and Darken (1989) used  $k$ -means clustering method to position the RBF centres and least means squares algorithm to estimate the weights, Chen et al. (1992) used  $k$ -means clustering to position the centres and Givens least squares algorithm to estimate the weights. In the present study, a new type of clustering algorithm called *moving  $k$ -means clustering* will be introduced to position the RBF centres and Givens least squares algorithm will be used to estimate the weights.

#### 4.1 Moving $k$ -means Clustering Algorithm

In section 2, clustering problems have been discussed that are related to dead centres, centre redundancy and poor local minima. In this section, a clustering algorithm is proposed to minimise the first two problems and indirectly reduces the effect of the third problem. The algorithm is based on non-adaptive clustering technique. The algorithm is called *moving  $k$ -means clustering* because during the clustering process, the fitness of each centre is constantly checked and if the centre fails to satisfy a specified criterion the centre will be moved to the region that has the most active centre. The algorithm is designed to have the following properties:

- All the centres will have about the same fitness in term of the fitness criteria, so there is no dead centre.
- More centres will be allocated at the heavily populated data area but some of the centres will also be assigned to the rest of the data so that all data are within an acceptable distance from the centres.
- The algorithm can reduce the sensitivity to the initial centres hence the algorithm is capable of avoiding poor local minima.

The moving  $k$ -means clustering algorithm will be described next. Consider a problem that has  $N$  data that have to be clustered into  $n_c$  centres. Let  $v_i$  be the  $i$ -th data and  $c_j$  be the  $j$ -th centre where  $i = 1, 2, \dots, N$  and  $j = 1, 2, \dots, n_c$ . Initially, centres  $c_j$  are initialised to some values and each data is assigned to the nearest centre and the position of the centre  $c_j$  is calculated according to:

$$c_j = \frac{1}{n_j} \sum_{i \in c_j} v_i \quad (5)$$

After all data are assigned to the nearest centres, the fitness of the centres is verified by using a distance function. The distance function is based on the total Euclidean distance between the centre and all the data that are assigned to the centre, defined as

$$f(c_j) = \sum_{i \in c_j} (\|v_i - c_j\|)^2; \quad j = 1, 2, \dots, n_c; \quad i = 1, 2, \dots, N \quad (6)$$

In general, the smaller the value of  $f(c_j)$  the less suitable is the centre  $c_j$  and  $f(c_j) = 0$  suggests that the centre has no members (i.e. no data has been assigned to  $c_j$ ) or the centre is placed outside the data range.

The moving  $k$ -means clustering algorithm can be implemented as:

- (1) Initialise the centres and  $\alpha_0$ , and set  $\alpha_a = \alpha_b = \alpha_0$ .
- (2) Assign all data to the nearest centre and calculate the centre positions using equation (5)
- (3) Check the fitness of each centre using equation (6).
- (4) Find  $c_s$  and  $c_l$ , the centre that has the smallest and the largest value of  $f(\cdot)$ .
- (5) **If**  $f(c_s) < \alpha_a f(c_l)$ ,
  - (5.1) Assign the members of  $c_l$  to  $c_s$  if  $v_i < c_l$ , where  $i \in c_l$ , and leave the rest of the members to  $c_l$ .
  - (5.2) Recalculate the positions of  $c_s$  and  $c_l$  according to:

$$\left. \begin{aligned} c_s &= \frac{1}{n_s} \sum_{i \in c_s} v_i \\ c_l &= \frac{1}{n_l} \sum_{i \in c_l} v_i \end{aligned} \right\} \quad (7)$$

Note that  $c_s$  will give up its members before step (5.1) and,  $n_s$  and  $n_l$  in equation (7) are the number of the new members of  $c_s$  and  $c_l$  respectively, after the reassigning process in step (5.1).

- (6) Update  $\alpha_a$  according to  $\alpha_a = \alpha_a - \alpha_a/n_c$  and **repeat** step (4) and (5) **until**  $f(c_s) \geq \alpha_a f(c_l)$
- (7) Reassign all data to the nearest centre and recalculate the centre positions using equation (5).
- (8) Update  $\alpha_a$  and  $\alpha_b$  according to  $\alpha_a = \alpha_0$  and  $\alpha_b = \alpha_b - \alpha_b/n_c$  respectively, and **repeat** step (3) to (7) **until**  $f(c_s) \geq \alpha_b f(c_l)$ .

where  $\alpha_0$  is a small constant value,  $0 < \alpha_0 < \frac{1}{3}$ . The computational time will increase as the values of  $\alpha_0$  get larger. Hence  $\alpha_0$  should be selected to compromise between good performance and computational load. The centres for the algorithm can be initialised to any values but a slightly better result can be achieved if the centres are initialised within the input and output data range. If the centres are badly initialised then  $\alpha_0$  should be selected a little bit bigger (typically  $> 0.2$ ).

Moving  $k$ -means clustering algorithm is specially designed for RBF network and may not give a good clustering performance in solving other problems such as pattern classification. The idea of clustering in RBF networks is to locate the centres in such a way that all the data are within an acceptable distance from the centres. In a normal clustering problem the centres have to be located where the data are concentrated and a few data may be situated far away from the centres. Furthermore, in the RBF network clustering problem, data with different patterns may be assigned to the same centre if those data are closely located.

## 4.2 Givens Least Squares Algorithm

After the RBF centres and the non-linear functions have been selected, the weights of the RBF network can be estimated using a least squares type algorithm. In the present study, exponential weighted least squares was employed based on the Givens transformation. The estimation problem using weighted least squares can be described as follows:

Define a vector  $\mathbf{z}(t)$  at time  $t$  as:

$$\mathbf{z}(t) = [z_1(t), \dots, z_{n_h}(t)] \quad (8)$$

where  $\mathbf{z}(t)$  and  $n_h$  are the output of the hidden nodes and the number of hidden nodes to the RBF network respectively. If linear input connections are used, equation (8) should be modified to include linear terms as follows:

$$\mathbf{z}(t) = [z_1(t) \quad \dots \quad z_{n_h}(t) \quad zl_1(t) \quad \dots \quad zl_{n_l}(t)] \quad (9)$$

where  $zl$ 's are the outputs of linear input connection nodes in Figure 1. Any vector or matrix size  $n_h$  should be increased to  $n_h + n_l$  in order to accommodate the new structure of the network. A bias term can also be included in the RBF network in the same way as the linear input connections.

Define a matrix  $\mathbf{Z}(t)$  at time  $t$  as:

$$\mathbf{Z}(t) = \begin{bmatrix} \mathbf{z}(1) \\ \mathbf{z}(2) \\ \vdots \\ \mathbf{z}(t) \end{bmatrix}; \quad (10)$$

and an output vector,  $\mathbf{y}(t)$  given by:

$$\mathbf{y}(t) = [y(1), \dots, y(t)]^T, \quad (11)$$

then the normal equation can be written as:

$$\mathbf{y}(t) = \mathbf{Z}(t)\Theta(t) \quad (12)$$

where  $\Theta(t)$  is a coefficient vector given by:

$$\Theta(t) = [w_1(t), \dots, w_{n_h + n_l}(t)]^T \quad (13)$$

The weighted least squares algorithm estimates  $\Theta(t)$  by minimising the sum of weighted squared errors, defined as:

$$e(t)_{wLS} = \sum_{i=1}^t \beta(t-1) [\mathbf{y}(i) - \mathbf{Z}(i-1)\Theta(t)]^2 \quad (14)$$

where  $\beta$ ,  $0 < \beta < 1$ , is an exponential forgetting factor. The solution for the equation (12) is given by,

$$\Theta(t) = [\mathbf{Z}^T(t)\mathbf{Q}(t)\mathbf{Z}(t)]^{-1}\mathbf{Z}^T(t)\mathbf{Q}(t)\mathbf{y}(t) \quad (15)$$

where  $\mathbf{Q}(t)$  is  $n_t \times n_t$  diagonal matrix defined recursively by:

$$\mathbf{Q}(t) = [\beta(t)\mathbf{Q}(t-1) \quad \mathbf{1}], \quad \mathbf{Q}(1) = \mathbf{1}; \quad (16)$$

and  $\beta(t)$  and  $n_t$  are the forgetting factor and the number of training data at time  $t$  respectively.

Many solutions have been suggested to solve the weighted least squares problem (15) such as recursive modified Gram Schemit, fast recursive least squares, fast Kalman algorithm and Givens least squares. In the present study, Givens least squares without square roots was used. The application of the Givens least squares algorithm to adaptive filtering and estimation have stimulated much interest due to superior numerical stability and accuracy (Ling, 1991).

## 5. Application Examples

The RBF network trained using the proposed hybrid algorithm based on moving  $k$ -means clustering and Givens least squares algorithm, derived in section 4 was used to model two systems. In these examples thin-plate-spline was selected as the non-linear function of RBF network and the RBF centres were initialised to the first few samples of the input-output data. During the calculation of the mean squared error (MSE), the noise model was excluded from the model since the noise model will normally cause the MSE to become unstable in the early stage of training. All data samples were used to calculate the MSE in all examples unless stated otherwise.

### Example 1

System S1 is a simulated system defined by the following difference equation:

$$S1:- \quad y(t) = 0.3y(t-1) + 0.6y(t-2) + u^2(t-1) + 0.3u^2(t-1) - 0.4u(t-1) + e(t)$$

where  $e(t)$  was a Gaussian white noise sequence with zero mean and variance 0.05 and the input,  $u(t)$  was a uniformly random sequence (-1,+1). System S1 was used to generate 1000 pairs of data input and output. The first 600 data were used to train the network and the remaining 400 data were used to test the fitted model.

The RBF centres were initialised to the first few samples of input and output data and the network was trained based on the following configuration:

$$v(t) = [u(t-1) \ y(t-1) \ y(t-2)]$$

$$\rho = 1000.0, \beta_0 = 0.99, \beta(0) = 0.95, n_h = 25, \alpha_0 = 0.1$$

Note: Refer to Chen et. al. (1992) or Mashor (1995) for the definition of these parameters.

One step ahead prediction (OSA) and model predicted output (MPO) of the fitted network model over both the training and testing data sets are shown in Figures 2 and 3 respectively. These plots show that the model predicts very well over both training and testing data sets. The correlation tests in Figure 4 are very good, all the correlation tests are within the 95% confidence limits. The evolution of the MSE obtained from the fitted model is shown in Figure 5. During the learning process, the MSE of the RBF network model was reduced from an initial value of 5.76dB to a noise floor of -21.10dB. The good prediction, MSE and correlation tests suggest that the model is unbiased and adequate to represent the identified system.

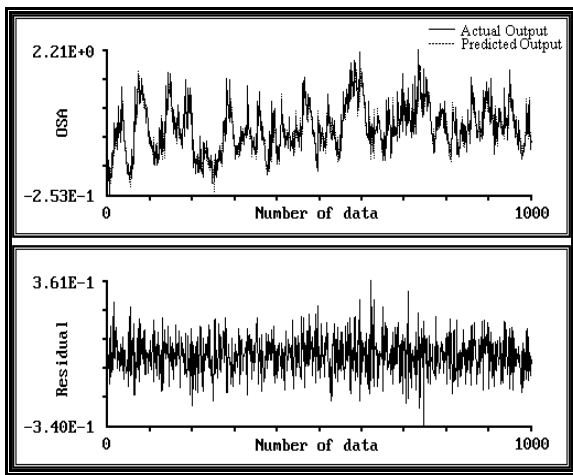


Figure 2. OSA superimposed on actual output

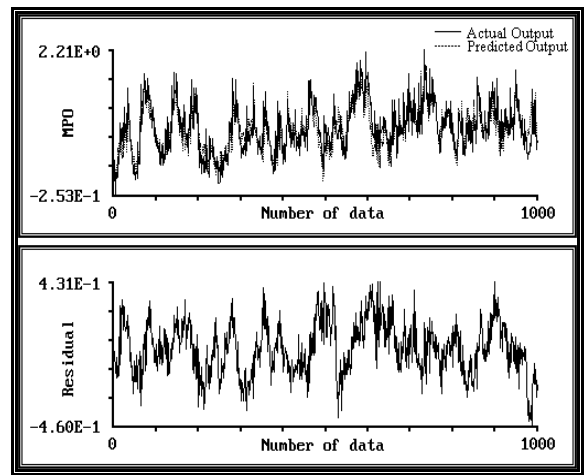


Figure 3. MPO superimposed on actual output

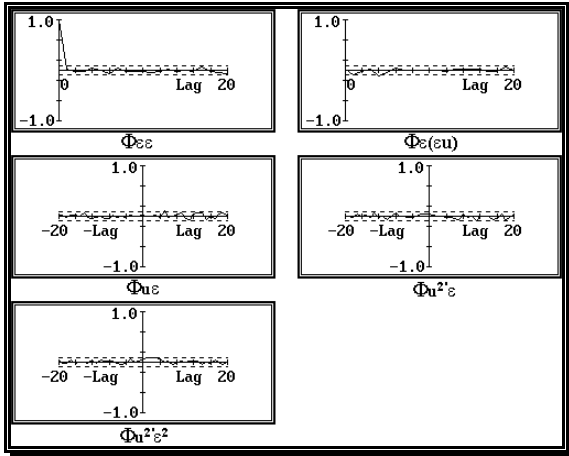


Figure 4. Correlation tests

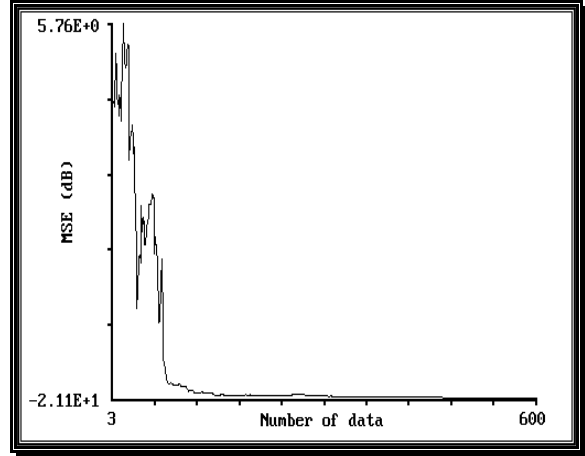


Figure 5. MSE

*Example 2*

A data set of 1000 input-output samples were taken from system S2 which is a tension leg platform. The first 600 data were used to train the network and the rest were used to test the fitted network model. The RBF centres were initialised to the first few input-output data samples and the network was trained using the following structure:

OSA and MPO generated by the fitted model are shown in Figures 6 and 7 respectively. The plots show that the network model predicts reasonably over both the training and testing data sets. All the correlation tests, shown in Figure 8, are inside the 95% confidence limits except for  $\Phi_{u^2\varepsilon^2}(\tau)$  which is marginally outside the confidence limits at lag 7. The evolution of the MSE plot is shown in Figure 9. During the learning process, the MSE was reduced from 17.54dB initially to a final value of -2.98dB. Since the model predicts reasonably and has good correlation tests, the model can be considered as an adequate representation of the identified system.

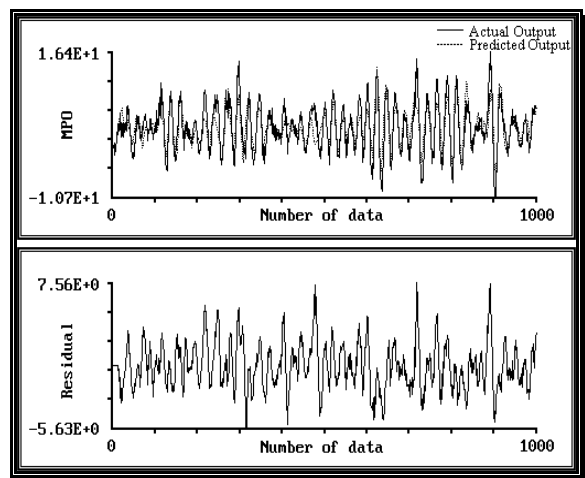
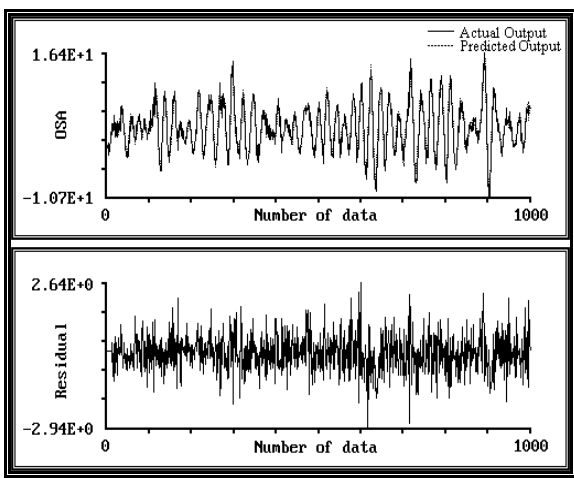


Figure 6. OSA superimposed on actual output

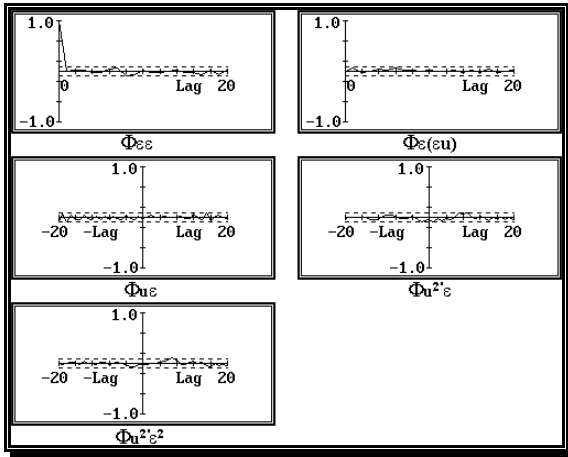


Figure 8. Correlation tests

Figure 7. MPO superimposed on actual output

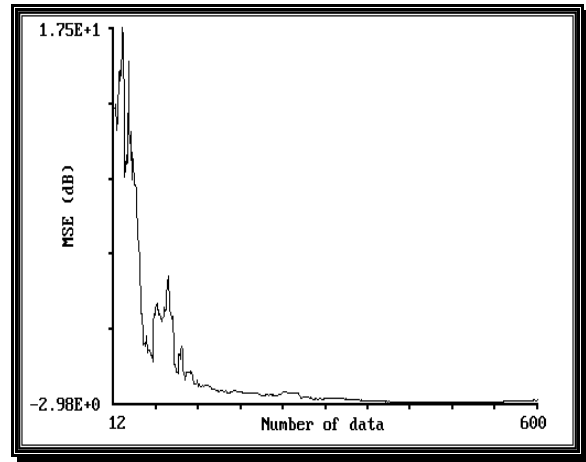


Figure 9. MSE

## 6. Performance Comparison

In this section the RBF network sensitivity to initial centres was compared between four clustering methods namely adaptive  $k$ -means clustering, non-adaptive  $k$ -means clustering, fuzzy  $c$ -means clustering and the proposed moving  $k$ -means clustering. The adaptive  $k$ -means clustering, non-adaptive clustering and fuzzy  $c$ -means clustering algorithms are based on the algorithm by MacQueen (1967), Lloyd (1957) and Bezdek (1981) respectively. The non-linear function was selected to be the thin-plate-spline function and all the network models have the same structure except that the RBF centres were clustered differently. The two examples that were used in the previous section were used again for this comparison. Two centre initialisation methods were used in this comparison:

- IC1:- The centres were initialised to the first few samples of input and output data.
- IC2:- All the centres were initialised to a value which is slightly outside the input and output data range. The centres are initialised to 3.0 and 18.0 for example S1 and S2 respectively.

Notice that, IC1 represents good centre initialisation whereas IC2 represents bad centre initialisation.

In this comparison, the additional linear connections are excluded because these connections may compensate the deficiency of the clustering algorithms. The networks were structured using the following specifications:

*Example (1)*

$$\begin{aligned} v(t) &= [u(t-1) \ y(t-1) \ y(t-2)] \\ \rho &= 1000.0, \beta_0 = 0.99, \beta(0) = 0.95 \\ \alpha_0 &= 0.1 \text{ for IC1 and } \alpha_0 = 0.2 \text{ for IC2} \end{aligned}$$

*Example (2)*

$$\begin{aligned} v(t) &= [u(t-1) \ u(t-3) \ u(t-4) \ u(t-6) \ u(t-7) \ u(t-8) \ u(t-11) \ y(t-1) \ y(t-3) \ y(t-4)] \\ \rho &= 1000.0, \beta_0 = 0.99, \beta(0) = 0.95 \\ \alpha_0 &= 0.1 \text{ for IC1 and } \alpha_0 = 0.2 \text{ for IC2} \end{aligned}$$

In general, the centres of the RBF network should be selected in such a way that the identified data can be properly represented. In other words, the total distance between the centres and the training data should be minimised. Therefore, mean squared distance (MSD) defined as in equation (17) below can be used to measure how good the centres that are produced by the clustering algorithms to represent the data. MSD for the resulting centres of a clustering algorithm is defined as:

$$MSD = \frac{1}{N} \sum_{t=1}^N M_{jt} \left( \|v(t) - c_j\| \right)^2; \quad j = 1, 2, \dots, n_h \quad (17)$$

where the  $n_h$ ,  $N$ ,  $v(t)$  and  $c_j$  are the number of centres, number of training data, input vector and centres respectively.  $M_{jt}$  is a membership function which means that the data  $v(t)$  belongs

to centre  $c_j$  and the data are assigned to the nearest centres. Both the training and testing data sets were used to calculate MSD. For each data set two sets of MSD were calculated, one set for IC1 initialisation and another set for IC2 initialisation. Mean squared error (MSE) will be used to test the suitability of the centres produced by the clustering algorithms to be used by RBF network. Two sets of MSE were generated for each data set, one set for IC1 initialisation and another set for IC2 initialisation.

The MSD plots in Figures (10), (11), (14) and (15) show that the proposed moving  $k$ -means clustering algorithm is significantly better than the three standard algorithms. The improvement is very large in the case of bad initialisation (see Figures (11) and (15)). The plots indicate that fuzzy  $c$ -means clustering and adaptive  $k$ -means clustering are very sensitive to initial centres. Additional centres just fail to improve the performance of these algorithms. On the other hand, the proposed clustering algorithm produced good result which was similar to the one initialised using IC1. Therefore, the results from the two examples suggest that the proposed moving  $k$ -means clustering is not sensitive to initial centres and always produce better results than the three standard clustering algorithms.

The suitability of the centres produced by the clustering algorithms were tested using MSE and plotted in Figures (12) and (13) for system S1 and Figures (16) and (17) for system S2. As mention earlier there is a strong correlation between clustering performance and the overall performance of RBF network. Thus, the MSE plots in Figures (12), (13), (16) and (17) show that the proposed moving  $k$ -means clustering algorithm gives the best performance. The performance is tremendously improved in the case of bad initialisation since moving  $k$ -means clustering algorithm is not sensitive to initial centres (see Figures (13) and (17)).

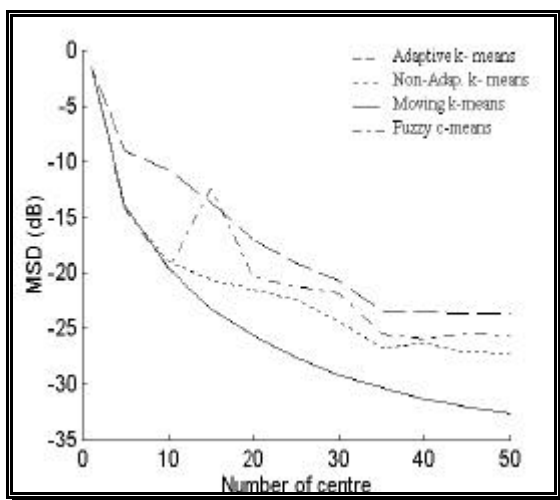


Figure 10. MSD for S1 using IC1

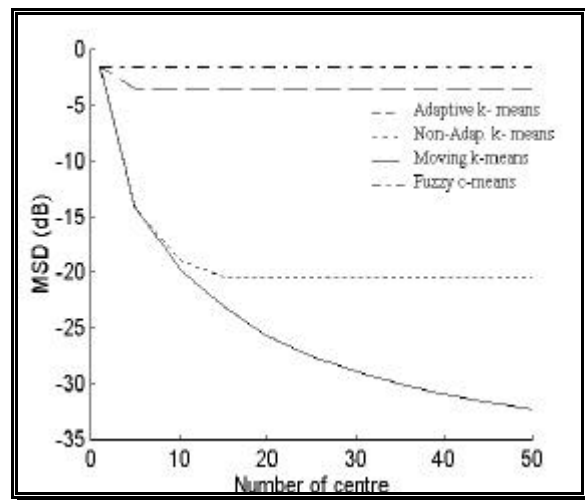


Figure 11. MSD for S1 using IC2

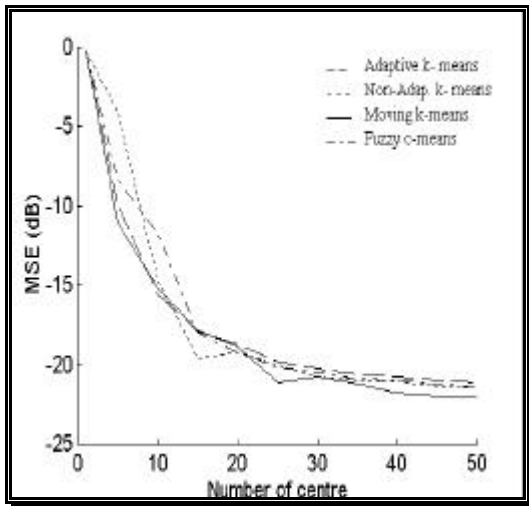


Figure12. MSE for S1 using IC1

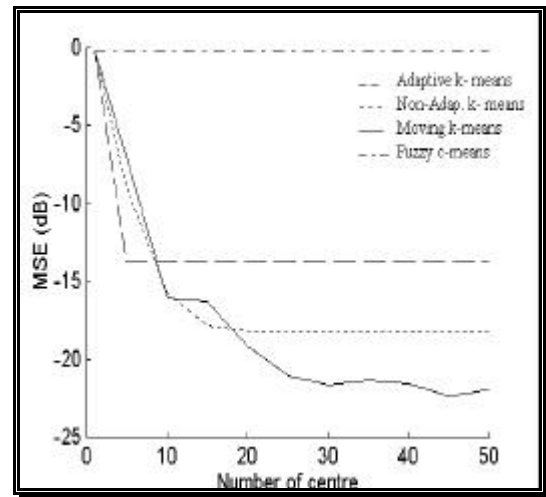


Figure 13. MSE for S1 using IC2

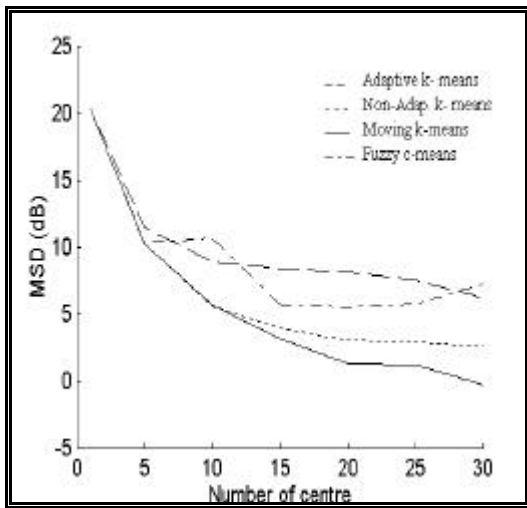


Figure14. MSD for S2 using IC1

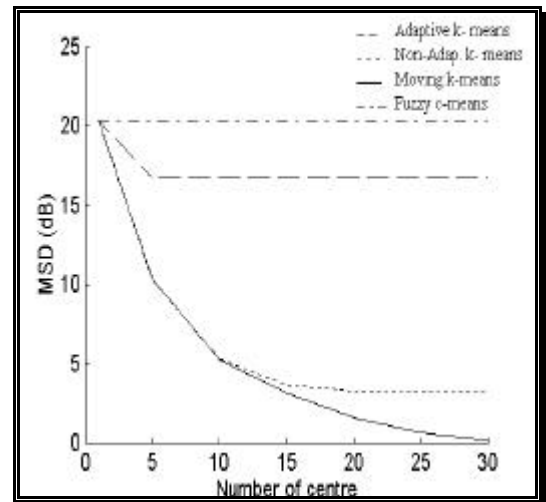


Figure 15. MSD for S2 using IC2

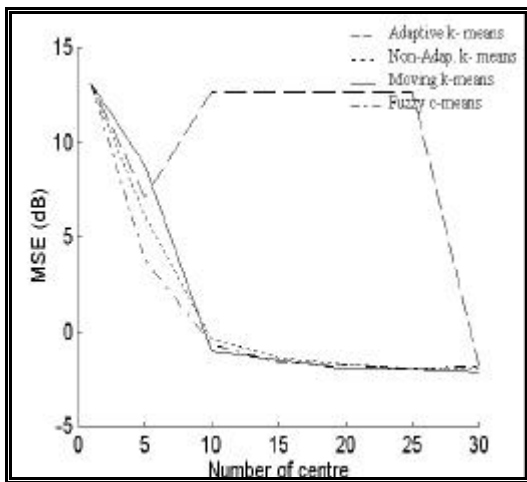


Figure16. MSE for S2 using IC1

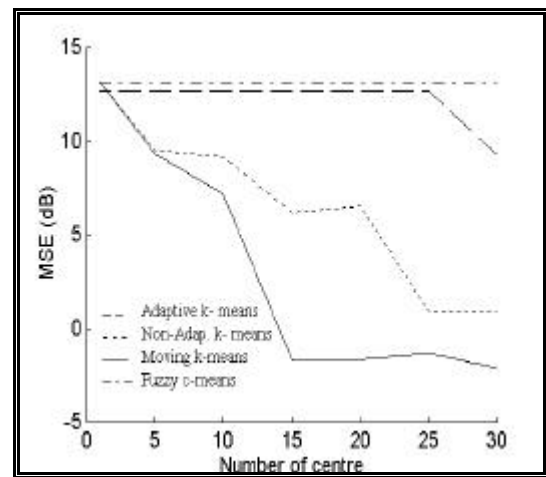


Figure 17. MSE for S2 using IC2



## 7. Conclusion

A new hybrid algorithm based on moving  $k$ -means clustering and Givens least squares has been introduced to train RBF networks. Two examples were used to test the efficiency of the hybrid algorithm. In these examples the fitted RBF network models yield good predictions and correlation tests. Therefore, the proposed algorithm is considered as adequate to train RBF network. The advantages of the proposed moving  $k$ -means clustering over adaptive  $k$ -means clustering, non-adaptive  $k$ -means clustering and fuzzy  $c$ -means clustering were demonstrated using MSD and MSE. It is perceptible from the MSD and MSE plots that moving  $k$ -means clustering has significantly improved the overall performance of the RBF network. The results also reveal that the RBF networks that used moving  $k$ -means clustering are not sensitive to initial centres and always give good performance.

## REFERENCES

- [1] Bezdek, J.C., 1981, *Pattern recognition with fuzzy objective function algorithms*, Plenum, New York.
- [2] Chen, S., Billings, S.A. and Grant, P.M., 1992, "Recursive hybrid algorithm for non-linear system identification using radial basis function networks", *Int. J. of Control*, 5, 1051-1070.
- [3] Cichocki, A., and Unbehauen, R., 1993, *Neural networks for optimisation and signal processing*, Wiley, Chichester.
- [4] Darken, C., and Moody, J., 1990, "Fast adaptive  $k$ -means clustering: Some empirical results", *Int. Joint Conf. on Neural Networks*, 2, 233-238.
- [5] Darken, C., and Moody, J., 1992, "Towards fast stochastic gradient search", In: *Advance in neural information processing systems 4*, Moody, J.E., Hanson, S. J., and Lippmann, R.P. (eds.), Morgan Kaufmann, San Mateo.
- [6] Ismail, M.A., and Selim, S.Z., 1986, "Fuzzy  $c$ -means: Optimality of solutions and effective termination of the algorithm", *Pattern Recognition*, 19, 481-485.
- [7] Kamel, M.S., and Selim, S.Z., 1994, "New algorithms for solving the fuzzy clustering problem", *Pattern Recognition*, 27 (3), 421-428.
- [8] Ling, F., 1991, "Givens rotation based on least squares lattice and related algorithms", *IEEE Trans. on Signal Processing*, 39, 1541-1551.
- [9] Lowe, D., 1989, "Adaptive radial basis function non-linearities and the problem of generalisation", *IEE Conf. on Artificial Intelligence and Neural Networks*.

- [10] Lloyd, S.P., 1957, "Least squares quantization in PCM", Bell Laboratories Internal Technical Report, *IEEE Trans. on Information Theory*.
  - [11] Macqueen, J., 1967, "Some methods for classification and analysis of multi-variate observations". In: *Proc. of the Fifth Berkeley Symp. on Math., Statistics and Probability*, LeCam, L.M., and Neyman, J., (eds.), Berkeley: U. California Press, 281.
  - [12] Mashor, M.Y., *System identification using radial basis function network*, PhD thesis, University of Sheffield, United Kingdom, 1995.
  - [13] Moody, J., and Darken, C.J., 1989, "Fast learning in neural networks of locally- tuned processing units", *Neural Computation*, **1**, 281-294.
  - [14] Poggio, T., and Girosi, F., 1990, "Network for approximation and learning", *Proc. of IEEE*, 78 (9), 1481-1497.
  - [15] Xu, L., Krzyzak, A., and Oja, E., 1993, "Rival penalised competitive learning for clustering analysis, RBF net and curve detection", *IEEE trans. on Neural Networks*, 4 (4).
-