Malaysian Journal of Computer Science, Vol. 11 No. 1, June 1998, pp. 74-80

NONLINEAR SYSTEM IDENTIFICATION USING RBF NETWORKS WITH LINEAR INPUT CONNECTIONS

Mohd Yusoff Mashor
School of Electrical and Electronic Engineering
University Science of Malaysia
Perak Branch Campus
31750 Tronoh, Perak
Malaysia
Fax: 605-3677443
email: yusof@eng.usm.my

ABSTRACT

This paper presents a modified RBF network with additional linear input connections together with a hybrid training algorithm. The training algorithm is based on k-means clustering with square root updating method and Givens least squares algorithm with additional linear input connections features. Two real data sets have been used to demonstrate the capability of the proposed RBF network architecture and the new hybrid algorithm. The results indicated that the network models adequately represented the systems dynamic.

Keywords: ***Nonlinear system, System identification, Neural network, RBF network, Linear input connections***

1.0 INTRODUCTION

Neural networks have theoretically been proved to be capable of representing complex non-linear mappings [1], [2]. Multilayered feed forward neural networks offer an alternative to conventional modelling methods, and networks with various training laws have successfully been used to model non-linear systems [3], [4], [5], [6], [7], [8]. In general, neural network models are highly non-linear in the unknown-parameters. A drawback of this property is that the training algorithm must be based on a non-linear optimisation technique which is associated with the problems of slow parameter convergence, intensive computation and poor local minima.

Radial Basis Function (RBF) networks that overcome some of these problems have been introduced by several authors [8], [9]. This network offers a faster and efficiently-trained-network while partially avoiding the problem of local minima. Training algorithms for RBF networks normally comprise of a procedure to position the RBF centres and a linear least squares technique to estimate the weights. Various training algorithms have been proposed to train RBF networks [4], [5], [6], [7], [8].

In the present study, the convergence properties of the hybrid algorithms are further improved by proposing a RBF network with additional linear input connections. As an alternative to the hybrid algorithm [6], a new hybrid algorithm based on *k*-means clustering using square root updating method and Givens least squares algorithm (with additional linear input connections features) is proposed to train RBF networks. Givens least squares algorithm has been selected due to its superior numerical stability and accuracy.

2.0 LINEAR-INPUT-CONNECTION-RBF NETWORK

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

$$y_i(t) = w_{i0} + \sum_{j=1}^{n_h} w_{ij}\phi\left(\left\|v(t) - c_j(t)\right\|\right);\ i = 1,\ldots,m \tag{1}$$

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.

The function $\phi(\bullet)$ can be selected from a set of basis functions such as linear, cubic, thin-plate-spline, multi-quadratic, inverse multi-quadratic and Gaussian functions. In the present study, the thin-plate-spline function has been selected as the non-linear function because this function has a good modelling capability [9]. Thin-plate-spline is given by:

$$\phi(a) = a^2 \log(a) \tag{2}$$

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, in the present study a RBF network with linear input connections is proposed.

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 original RBF model as shown in Fig. 1.

The new RBF network with m outputs, n inputs, n_h hidden nodes and n_l linear input connections can be expressed as:

$$y_i(t) = w_{i0} + \sum_{j=1}^{n_l} \lambda_{ij} vl(t) + \sum_{j=1}^{n_h} w_{ij} \phi\left(\left\| v(t) - c_j(t) \right\|\right) \qquad (3)$$

where i = 1, 2, ... , m, and the λ's and vl'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 λ's appear to be linear within the network, the λ'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.

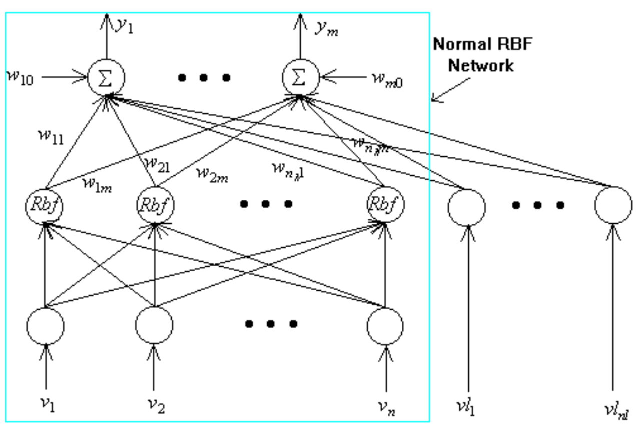

Fig. 1: The RBF network with linear input connections

3.0 MODELLING NON-LINEAR SYSTEMS

There are a number of studies that have been accomplished on modelling non-linear systems using radial basis function networks [4], [5], [6], [7], [8], [9], [10]. Modelling using RBF networks can be considered as fitting a surface in a multi-dimensional space to represent the training data set and using the surface to predict over the testing data set. Therefore, RBF networks require all the future data of the system to lie within the domain of the fitted surface to ensure a correct mapping so that good predictions can be achieved. This is normal for the non-linear modelling where the model is only valid over a certain amplitude range.

A wide class of non-linear systems can be represented by the non-linear auto-regressive moving average with exogenous input (NARMAX) model [11]. The NARMAX model can be expressed in terms of a non-linear function expansion of lagged input, output and noise terms as follows:

$$\begin{aligned} y(t) = f_s\big(& y(t-1),\cdots,y(t-n_y),u(t-1),\cdots,u(t-n_u),\\ & e(t-1),\cdots,e(t-n_e)\big) + e(t) \end{aligned} \qquad (4)$$

where

$$y(t) = \begin{bmatrix} y_1(t) \\ \vdots \\ y_m(t) \end{bmatrix};\ u(t) = \begin{bmatrix} u_1(t) \\ \vdots \\ u_r(t) \end{bmatrix} \text{ and } e(t) = \begin{bmatrix} e_1(t) \\ \vdots \\ e_m(t) \end{bmatrix}$$

are the system output, input and noise vector respectively; n_y, n_u and n_e are the maximum lags in the output, input and noise vector respectively; and m and r are the number of output and input respectively.

The non-linear function, $f_s(\bullet)$ is normally very complicated and rarely known a priori for practical systems. In the present study, $f_s(\bullet)$ will be modelled using RBF network expressed by equation (3) where $\phi(\bullet)$ is chosen to be the thin-plate-spline function. The network input vector, $v(t)$ is formed using lagged input, output and noise that are denoted as $u(t-1)\cdots u(t-n_u)$, $y(t-1)\cdots y(t-n_y)$ and $e(t-1)\cdots e(t-n_e)$ respectively in equation (4). Another method to include a noise model in a RBF network is to use only linear noise connections as in Fig. 1. This approach can reduce the complexity of the RBF network and hence accelerate the training process. However, this approach only allows linear noise model and may not be sufficient if the data is highly corrupted by non-linear noise.

Training algorithms for RBF networks normally comprise of a procedure to position the RBF centres and a linear least squares technique to estimate the weights. In the present study, a new hybrid algorithm based on k-means clustering using square root updating method and Givens least squares algorithm with additional linear input connections features is proposed to train RBF networks. Givens least squares has been selected because of the superior numerical stability and accuracy of this method [12].

4.0 PROPOSED HYBRID ALGORITHM

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

$$J = \sum_{t=1}^{N} \left(y(t) - \hat{y}(t)\right)^{\mathrm{T}} \left(y(t) - \hat{y}(t)\right) \qquad (5)$$

where N and $\hat{y}(t)$ are the number of data used for training and the predicted output generated by using the RBF network in equation (3) respectively. Equation (5) 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) allocation of the RBF centres, $c_j(t)$ and

(ii) estimation of the weights, w_{ij} .

This approach allows 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 [8] used the *k*-means clustering method to position the RBF centres and a least means squares algorithm to estimate the weights, Chen et al. [6] used a *k*-means clustering to positioned the centres and a Givens least squares algorithm to estimate the weights. In the present study, the *k*-means clustering using square root updating method is used to position the RBF centres and a Givens least squares algorithm with additional linear input connections features will be used to estimate the weights. A detailed description of the *k*-means clustering using square root updating method can be found in Darken and Moody [13].

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 were employed based on the Givens transformation. The estimation problem using weighted least squares can be described as follows:

Define a vector $z(t)$ at time t as:

$$\mathbf{z}(t)=[z_1(t),\ldots,z_{n_h}(t)] \tag{6}$$

where $\mathbf{z}(t)$ and n_h are the output of the hidden nodes vector and the number of hidden nodes of the RBF network respectively. If linear input connections are used, equation (6) should be modified to include linear terms as follows:

$$\mathbf{z}(t)=\left[z_1(t)\ \cdots\ z_{n_h}(t)\ \ zl_1(t)\ \cdots\ zl_{n_l}(t)\right] \tag{7}$$

where zl's are the output of the linear input connection nodes. Any vector or matrix of 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 Z(t) at time t as:

$$\mathbf{Z}(t)=\begin{bmatrix}\mathbf{z}(1)\\ \mathbf{z}(2)\\ \vdots\\ \mathbf{z}(t)\end{bmatrix} \tag{8}$$

and an output vector, y(t) given by:

$$\mathbf{y}(t)=[\,y(1),\ldots,y(t)\,]^{\mathrm{T}} \tag{9}$$

then the normal equation can be written as:

$$\mathbf{y}(t)=\ \mathbf{Z}(t)\Theta(t) \tag{10}$$

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

$$\Theta(t)=\left[w_1(t),\ldots,w_{n_h+n_l}(t)\right]^{\mathrm{T}} \tag{11}$$

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

$$e(t)_{WLS}=\sum_{i=1}^{t}\beta(t-1)[\mathbf{y}(i)-\mathbf{Z}(i-1)\Theta(t)]^2 \tag{12}$$

where β, $0 < \beta < 1$, is an exponential forgetting factor. The solution for the equation (10) is given by

$$\Theta(t)=[\mathbf{Z}^T(t)\mathbf{Q}(t)\mathbf{Z}(t)]^{-1}\mathbf{Z}^T(t)\mathbf{Q}(t)\mathbf{y}(t) \tag{13}$$

where Q(t) is $n_t \times n_t$ diagonal matrix defined recursively by

$$\mathbf{Q}(t)=[\beta(t)\mathbf{Q}(t-1)\ \ 1\ ],\qquad \mathbf{Q}(1)=1; \tag{14}$$

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 in equation (13) such as recursive modified Gram Schmidt, 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 [12].

Givens least squares without square roots is summarised below. Introduce a $n_t \times n_k$ diagonal matrix D(t) as:

$$\mathbf{D}(t)=diag\left[d_1(t)\ \cdots\ d_{(n_k-1)}(t)\ \ \sigma_\varepsilon^2(t)\right];\ n_k=n_h+n_l+1 \tag{15}$$

where n_k , d's and $\sigma_\varepsilon(t)$ are the number of the estimated RBF weights, the least squares estimation errors and the standard deviation at time t respectively. Define a $n_k \times n_k$ dimensional upper triangular matrix R(t) as:

$$\mathbf{R}(t)=\begin{bmatrix}1 & r_{12}(t) & r_{13}(t) & \cdots & \cdots & r_{1n_k}(t)\\ 0 & 1 & r_{23}(t) & \ddots & \ddots & r_{2n_k}(t)\\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots\\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots\\ \vdots & \ddots & \ddots & \ddots & \ddots & r_{(n_k-1)n_k}(t)\\ 0 & \cdots & \cdots & 0 & 0 & 1\end{bmatrix}$$

... (16)

where the r's are the least squares estimated coefficients after estimating $x_k^{(i)}(t)$ from $x_k^{(i-1)}(t)$.

The Givens Least Squares algorithm solves for $\Theta(t)$ in equation (13) by performing a Givens transformation:

$$\begin{bmatrix} \mathbf{D}^{1/2}(t-1) & [\mathbf{R}(t-1)] \\ (\delta^{(0)})^{1/2} & [x_1^{(0)}(t) \ldots x_{n_k}^{(0)}(t)] \end{bmatrix} \Rightarrow \begin{bmatrix} \mathbf{D}^{1/2}(t) & [\mathbf{R}(t)] \\ 0 & [0 \ldots 0] \end{bmatrix} \quad (17)$$

where

$$\left[x_1^{(0)}(t) \quad \cdots \quad x_{n_k}^{(0)}(t)\right] = \left[\, z_1(t) \quad \cdots \quad z_{n_k-1}(t) \quad y(t) \,\right] \quad (18)$$

and $\delta^{(0)}$ is called a maximum likelihood factor [12], initialised to $1/\beta(t)$, where $\beta(t)$ is updated using equation (24).

After i-th steps, Givens transformation transfers

$$\begin{bmatrix} 0 & \cdots & 0 & d_i^{1/2}(t-1) & d_i^{1/2}(t-1) r_{ii+1}(t-1) & \cdots & d_i^{1/2}(t-1) r_{in_k}(t-1) \\ 0 & \cdots & 0 & (\delta^{(i-1)})^{1/2} x_i^{(i-1)}(t) & (\delta^{(i-1)})^{1/2} x_{i+1}^{(i-1)}(t) & \cdots & (\delta^{(i-1)})^{1/2} x_{n_k}^{(i-1)}(t) \end{bmatrix}$$

... (19)

into

$$\begin{bmatrix} 0 & \cdots & 0 & d_i^{1/2}(t) & d_i^{1/2}(t) r_{ii+1}(t) & \cdots & d_i^{1/2}(t) r_{in_k}(t) \\ 0 & \cdots & 0 & 0 & (\delta^{(i)})^{1/2} x_{i+1}^{(i)}(t) & \cdots & (\delta^{(i)})^{1/2} x_{n_k}^{(i)}(t) \end{bmatrix}$$

... (20)

The explicit computation of Givens transformation can be expressed as follows:

$$\left.\begin{aligned} d_i(t) &= d_i(t-1) + \delta^{(i-1)} (x_i^{(i-1)}(t))^2 \\ c &= \frac{d_i(t-1)}{d_i(t)} \\ b &= \delta^{(i-1)} \frac{x_i^{(i-1)}(t)}{d_i(t)} \\ \delta^{(i)} &= c\delta^{(i-1)} \end{aligned}\right\} \quad (21)$$

and

$$\left.\begin{aligned} x_k^{(i)}(t) &= x_k^{(i-1)}(t) - x_k^{(i-1)}(t) r_{ik}(t-1) \\ r_{ik}(t) &= c r_{ik}(t-1) + b x_k^{(i-1)}(t) \end{aligned}\right\} \; k = i+1, \ldots, n_k \quad (22)$$

The algorithm is initialised by setting

$$\begin{aligned} \sigma_\varepsilon(0) &= 0 \\ \mathbf{R}(0) &= \mathbf{I} \\ \mathbf{D}(0) &= \frac{1}{\rho} \end{aligned} \quad (23)$$

where $\sigma_\varepsilon(0)$ and ρ are the initial standard deviation of the estimated parameters and a large positive scalar respectively; and I is an $n_k \times n_k$ identity matrix. The forgetting factor $\beta(t)$ is normally computed according to:

$$\beta(t) = \beta_0 \beta(t-1) + 1 - \beta_0 \quad (24)$$

where β_0 and $\beta(0)$ are typically chosen to be less but close to one and β_0 is normally larger than $\beta(0)$.

After Givens transformation is completed, the estimated parameter vector, $\Theta(t)$ can be calculated by back substitution.

$$w_{(n_k-1)}(t) = r_{(n_k-1)n_k}(t) \quad (25)$$

and

$$w_i(t) = r_{in_k}(t) - \sum_{j=i+1}^{(n_k-1)} r_{ij}(t) w_j(t); \quad i = (n_k-2), (n_k-3), \ldots, 1 \quad (26)$$

The RBF network using the hybrid algorithm based on k-means clustering using square root updating method and Givens Least Squares with additional linear input connections feature can be used to model non-linear systems. Initially, the RBF centres are positioned using the k-means clustering using square root updating method and then the Givens least squares algorithm is used to estimate the weights of the RBF network, w's.

5.0 MODEL VALIDATION

There are several ways of testing a model such as one step ahead predictions (OSA), model predicted output (MPO), mean squared error (MSE), correlation tests, and chi-square tests. In the present study, the first four tests were used to justify the performance of the fitted models. OSA is a common measure of predictive accuracy of a model that has been considered by many researchers. OSA can be expressed as:

$$\begin{aligned} \hat{y}(t) = f_s\big(& u(t-1), \cdots, u(t-n_u) \; y(t-1), \cdots, y(t-n_y) \\ & \hat{\varepsilon}(t-1, \hat{\theta}), \cdots, \hat{\varepsilon}(t-n_\varepsilon, \hat{\theta})\big) \end{aligned} \quad (27)$$

and the residual or prediction error is defined as

$$\hat{\varepsilon}(t, \hat{\theta}) = y(t) - \hat{y}(t) \quad (28)$$

where $f_s(\bullet)$ is a non-linear function, in this case the RBF network.

Another test that often gives a better measurement of the fitted model predictive capability is the model predicted output. Generally model predicted output can be expressed as:

$$\begin{aligned} \hat{y}_d(t) = f_s\big(& u(t-1), \cdots, u(t-n_u), \hat{y}_d(t-1), \cdots, \hat{y}_d(t-n_y), \\ & 0, \cdots, 0\big) \end{aligned} \quad (29)$$

and the deterministic error or deterministic residual is

$$\hat{\varepsilon}_d(t) = y(t) - \hat{y}_d(t) \quad (30)$$

MSE is an iterative method of model validation where the model is tested by calculating the mean squared errors after each training step. MSE test will indicate how fast a prediction error or residual converges with the number of training data. The MSE at the t-th training step, is given by

$$E_{MSE}\left(t,\Theta(t)\right)=\frac{1}{n_d}\sum_{i=1}^{n_d}\left(y(i)-\hat{y}\left(i,\Theta(t)\right)\right)^2 \qquad (31)$$

where $E_{MSE}\left(t,\Theta(t)\right)$ and $\hat{y}\left(i,\Theta(t)\right)$ are the MSE and OSA for a given set of estimated parameters $\Theta(t)$ after t training steps respectively, and n_d is the number of data that were used to calculate the MSE.

An alternative method of model validation is to use correlation tests to determine if there is any predictive information in the residual after model fitting [11]. The residual will be unpredictable from all linear and non-linear combinations of past inputs and outputs if the following hold [11]:

$$\left.\begin{array}{ll}
\Phi_{\varepsilon\varepsilon}(\tau)=\mathrm{E}\left[\varepsilon(t-\tau)\varepsilon(t)\right]=\delta(t) & \text{for all } \tau\\
\Phi_{u\varepsilon}(\tau)=\mathrm{E}\left[u(t-\tau)\varepsilon(t)\right]=0 & \text{for all } \tau\\
\Phi_{\varepsilon(\varepsilon u)}(\tau)=\mathrm{E}\left[\varepsilon(t)\left(\varepsilon(t-\tau)u(t-1-\tau)\right)\right]=0 & \text{for } \tau\geq 0\\
\Phi_{u^{2'}\varepsilon}(\tau)=\mathrm{E}\left[\left(u^2(t-\tau)-\bar{u}^2(t)\right)\varepsilon(t)\right]=0 & \text{for all } \tau\\
\Phi_{u^{2'}\varepsilon^2}(\tau)=\mathrm{E}\left[\left(u^2(t-\tau)-\bar{u}^2(t)\right)\varepsilon^2(t)\right]=0 & \text{for all } \tau
\end{array}\right\} \qquad (32)$$

where $\bar{u}^2(t)$ and $\mathrm{E}[\bullet]$ are the mean value of $u^2(t)$ and the expectation respectively. In practice, if the correlation tests lie within the 95% confidence limits, $\pm 1.96/\sqrt{N}$, then the model is regarded as adequate, where N is the number of data used to train the network.

6.0 APPLICATION EXAMPLES

The proposed RBF network trained using the hybrid algorithm was used to model two systems. In all the examples, the centres were initialised to the first few samples of the input-output data. During the calculation of the 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. In all examples, all the 1000 data were used to calculate the MSE and the designing parameters were taken as $\rho=1000.0$, $\beta_0=0.99$, and $\beta(0)=0.95$.

Example 1

The first data set, S1 was taken from a heat exchanger system and consists of 1000 samples. A detailed description of the process can be found in Billings and Fadhil [14]. The first 500 data were used to identify the system and the remaining 500 data were used to test the fitted RBF network model. The network has been trained using the following specifications:

$v(t)=\left[u(t-1)\;\; u(t-2)\;\; y(t-1)\;\; y(t-4)\right]$ with a bias input

$vl(t)=\left[u(t-1)\;\; u(t-2)\;\; y(t-1)\;\; y(t-4)\;\; e(t-3)\;\; e(t-4)\;\; e(t-5)\right];\quad n_h=20$

Notice that $vl(t)$ denotes the linear input connections vector and the e's represent the linear noise terms. OSA and MPO generated by the network model over both the training and testing data sets are shown in Fig. 2 and 3 respectively. The plots show that the model predicts very well over both the training and testing data sets. Correlation tests shown in Fig. 4 are quite reasonable where $\Phi_{u^{2'}\varepsilon^2}(\tau)$ and $\Phi_{u^{2'}\varepsilon}(\tau)$ plots are marginally outside the 95% confidence limits. The evolution of MSE obtained from the network model is shown in Fig. 5. As the model predicts very well and has reasonable correlation tests, the model was considered to be sufficient to represent the identified system.

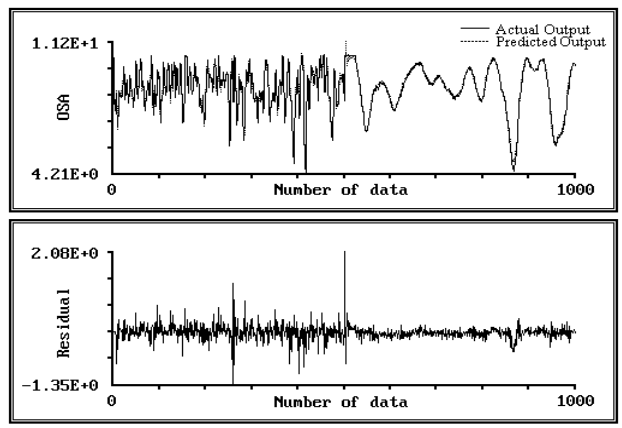

Fig. 2: OSA superimposed on actual output for example 1

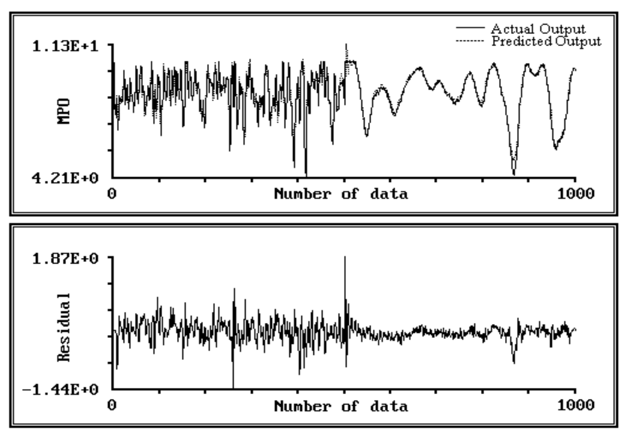

Fig. 3: MPO superimposed on actual output for example 1

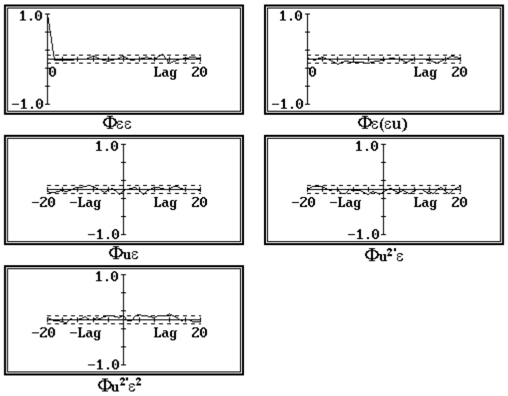

Fig. 4: Correlation tests for example 1

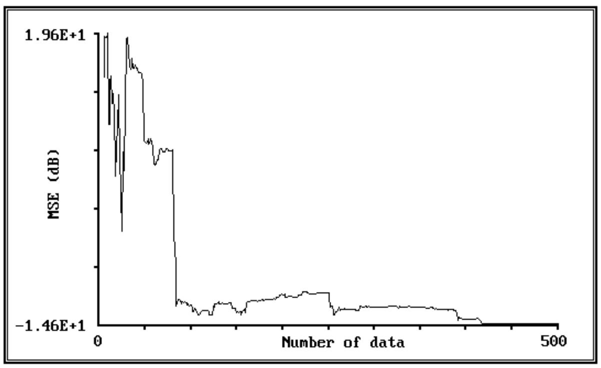

Fig. 5: MSE for example 1

Example 2

S2 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 network was trained using the following structure

$$v(t) = [u(t-1)\ \ u(t-3)\ \ u(t-4)\ \ u(t-6)\ \ u(t-7)$$
$$u(t-8)\ \ u(t-11)\ \ y(t-1)\ \ y(t-3)\ \ y(t-4)]$$
$$vl(t) = [y(t-1)\ y(t-2)\ e(t-1)\ e(t-2)\ e(t-3)\ e(t-5)]$$

with bias input; $n_h = 40$

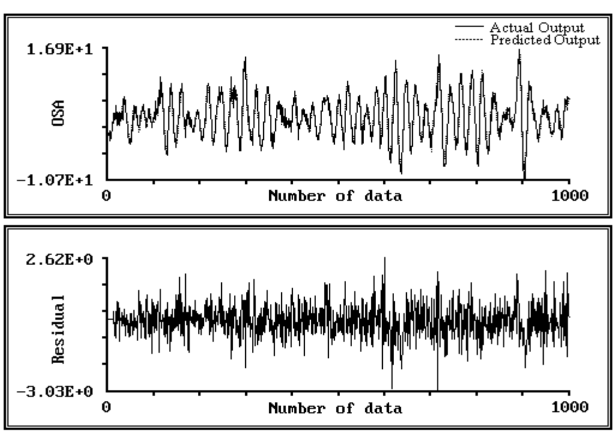

Fig. 6: OSA superimposed on actual output for example 2

OSA and MPO generated by the fitted model are shown in Fig. 6 and 7 respectively. The plots show that the model predicts very well over both the training and testing data sets. All the correlation tests, shown in Fig. 8, are well 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 Fig. 9. Since the model predicts very well and has good correlation tests, the model can be considered as an adequate representation of the identified system.

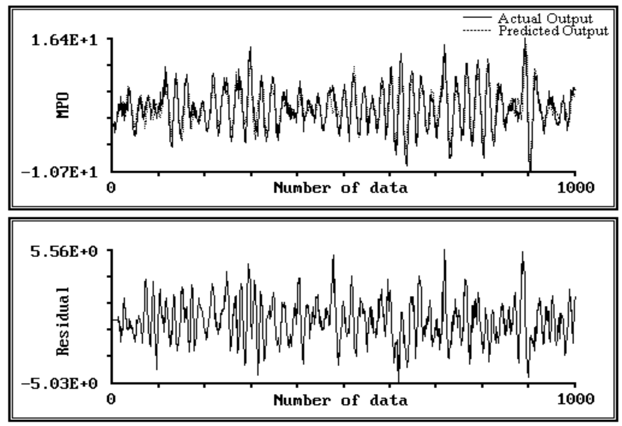

Fig. 7: MPO superimposed on actual output for example 2

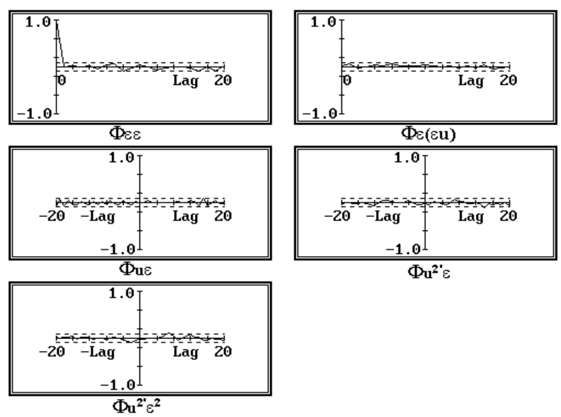

Fig. 8: Correlation tests for example 2

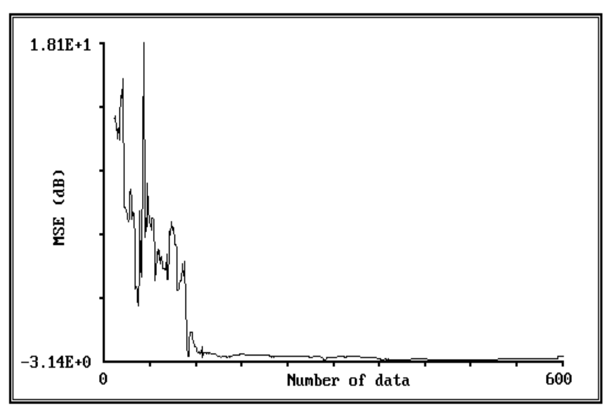

Fig. 9: MSE for example 2

7.0 CONCLUSIONS

A RBF network with additional linear input connections and a hybrid training algorithm based on square roots *k*-means clustering and Givens least squares (with additional linear input connections feature) has been introduced. Two practical data sets were used to test the efficiency of the modified RBF networks and the hybrid training algorithm. The examples showed that the fitted RBF network models yield good predictions and correlation tests. Hence, the proposed modified RBF networks together with the hybrid training algorithm is considered as an adequate representation of the identified system.

REFERENCES

[1] G. Cybenko, "Approximations by superposition of a sigmoidal function", *Mathematics of Control, Signal and Systems*, Vol. 2, 1989, pp. 303-314.

[2] K. Hornik, "Approximation capabilities of multilayer feed forward networks", *Neural Networks*, Vol. 4, 1991, pp. 251-257.

[3] S. A. Billings, H. B. Jamaluddin and S. Chen, "Properties of neural networks with applications to modelling non-linear dynamical systems", *Int. J. of Control*, Vol. 55, 1992, pp. 193-224.

[4] S. Chen, S. A. Billings, C. F. N. Cowan and P. M. Grant, "Non-linear systems identification using radial basis functions", *Int. J. Systems Science*, Vol. 21, 1990, pp. 2513-2539.

[5] S. Chen, C. F. N. Cowan, and P. M., Grant, "Orthogonal least squares learning algorithm for radial basis function networks", *IEEE Trans. on Neural Networks*, Vol. 2, 1991, pp. 302-309.

[6] S. Chen, S. A. Billings and P. M. Grant, "Recursive hybrid algorithm for non-linear system identification using radial basis function networks", *Int. J. of Control*, Vol. 55, 1992, pp. 1051-1070.

[7] M. Y. Mashor, *System identification using radial basis function network*, PhD thesis, University of Sheffield, United Kingdom, 1995.

[8] J. Moody, and C. J. Darken, "Fast learning in neural networks of locally-tuned processing units", *Neural Computation*, Vol. 1, 1989, pp. 281-294.

[9] M. J. D. Powell, "Radial basis function approximations to polynomials", *Proc. 12th Biennial Numerical Analysis Conf.*, Dundee, 1987, pp. 223-241.

[10] S. Chen, S. A. Billings, C. F. N. Cowan, and P. M. Grant, "Practical identification of NARMAX models using radial basis functions", *Int. J. Control*, Vol. 52, 1990, pp. 1327-1350.

[11] S. A. Billings, and W. S. F. Voon, "Structure detection and model validity tests in the identification of non-linear systems", *Proc. IEE, Part D*, Vol. 127, 1986, pp. 272-285.

[12] F. Ling, "Givens rotation based on least squares lattice and related algorithms", *IEEE Trans. on Signal Processing*, Vol. 39, 1991, pp. 1541-1551.

[13] C. Darken, and J. Moody, "Fast adaptive *k*-means clustering: Some empirical results", *Int. Joint Conf. on Neural Networks*, Vol. 2, 1990, pp. 233-238.

[14] S. A. Billings, and M. B. Fadhil, "The practical identification of system with non-linearities", *Proc. 7th IFAC Symp. on Identification and System Parameter Estimation*, York, U.K., 1985, pp. 155-160.

BIOGRAPHY

Mohd Yusoff Mashor obtained his B.Eng.(Hons.) in Control and Computer Eng. from Polytechnic of Central London in 1990, M.Sc. in Control and Information Technology from Sheffield University in 1991 and PhD from Sheffield University in 1995. Currently, he is a lecturer at Department of Electrical and Electronic Eng., University Science Malaysia. His research areas include Neural Network (theory and application), System Identification, Adaptive and Predictive Control. He is also a senior member of Instrumentation & Control Society of Malaysia (ICSM).