

The Proposal of Two New Recurrent Radial Basis Function Neural Networks

Niusha Shafiabady
 Department of Electrical and
 Electronics Engineering
 University of
 NottinghamMalaysia Campus

Dino Isa
 Department of Electrical and
 Electronics Engineering
 University of
 NottinghamMalaysia Campus

M.A. Nima Vakilian
 Society of Petroleum Engineers
 Texas, USA

ABSTRACT

Two types of new recurrent RBF neural networks are introduced here and are applied on four test problems that are used for identification. The proposed recurrent RBF neural networks use both the power of the recurrent neural networks together with the abilities of the RBF neural networks so it can achieve good results.

Keywords

Identification, Recurrent RBF Neural Network.

1. INTRODUCTION

Neural networks have been widely applied in many engineering fields including identification, pattern recognition, control and classification. Recent results show that neural network technique seems to be very effective to identify a broad category of complex nonlinear systems when complete model information cannot be obtained. Neural networks can be classified as feed forward and recurrent ones [6]. Feed forward networks, for example Multilayer Perceptrons (MLP), are implemented for the approximation of nonlinear functions in the right hand side of dynamic model equations. The main drawback of these neural networks is that the weights' updating does not utilize information on the local data structure and the function approximation is sensitive to the training data [7]. Since recurrent networks incorporate feedback, they have powerful representation capability and can successfully overcome disadvantages of feed forward networks [8]. RBF neural networks have been used as a powerful tool in many engineering and scientific applications as they possess the following features: 1) They are universal approximators [1]; 2) They have a simple topological structure [2]; 3) They can implement fast learning algorithms because of locally tuned neurons [3]. In this study a recurrent RBF neural network is introduced where the RBF network's ability is added to the advantages of recurrent networks.

This paper is organized as follows. Section 2 describes the proposed recurrent RBF networks and section 3 explains the training algorithms used to train them. Section 4 shows the

simulation results and finally the conclusion is given in section 5.

2. RECURRENT RBF NEURAL NETWORKS

In order to increase the abilities of the RBF neural network two types of recurrent RBF neural networks are introduced here. The structure of the recurrent RBF neural network type one called RRBF_type1 is shown in fig. 1 and the structure of the recurrent RBF neural network type two is shown in fig. 2 and is called RRBF_type2 here. The output of the recurrent RBF network is calculated in this way.

$$y = \sum_i w_i \varphi_i \quad (1)$$

The term w_i here denotes the weights of the output layer that is multiplied by the neurons' outputs named φ_i .

Considering this the term φ is totally different from the ordinary RBF network for RRBF_type1 as seen in figure 1.

$$\begin{aligned} \varphi_i &= e^{-\frac{(x_{(i)}-m_{(i)})^2}{2\delta_{(i)}^2}} + e^{-\frac{(\varphi_{(i-1)}-m_{(i)})^2}{2\delta_{(i)}^2}} \Rightarrow \\ \varphi_i &= e^{-\frac{(x_{(i)}-m_{(i)})^2}{2\delta_{(i)}^2}} + e^{-\frac{(e^{-\frac{(x_{(i-1)}-m_{(i-1)})^2}{2\delta_{(i-1)}^2}} - m_{(i)})^2}{2\delta_{(i)}^2}} \end{aligned} \quad (2)$$

Here the recurrent links convey φ_{i-1} to the neuron performing its i^{th} iteration. Term φ is calculated as mentioned below for RRBF_type2 neural network.

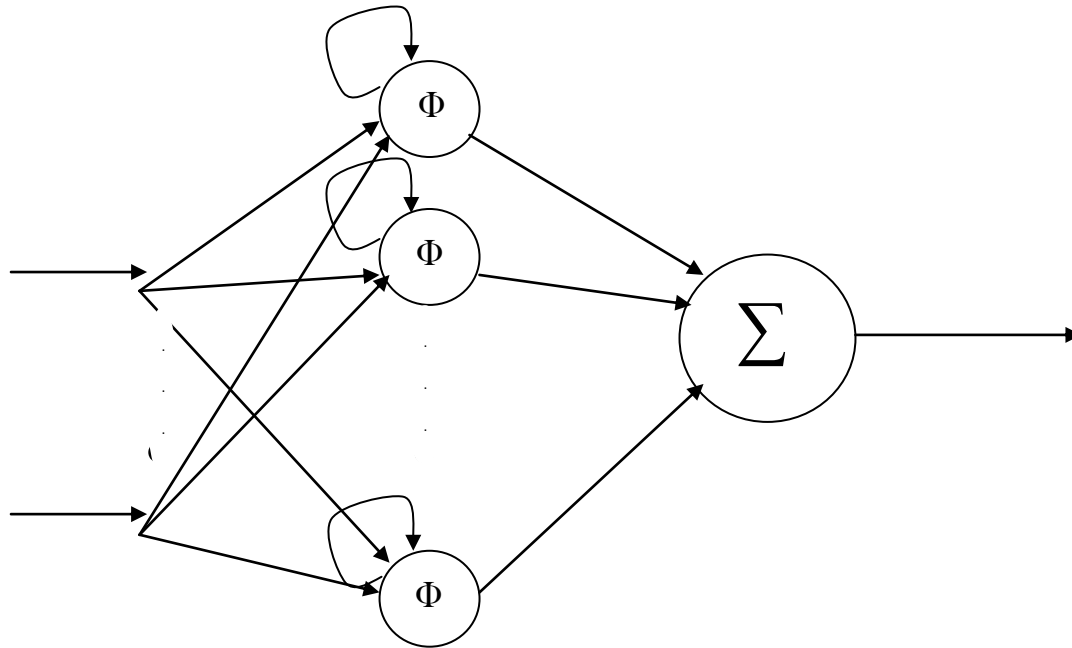


Fig. 1- The structure of RRBF_type1 neural network

$$\varphi_i = e^{-\frac{(x_{(i)}-m_{(i)})^2}{2\delta_{(i)}^2}} + e^{-\frac{(y_{(i-1)}-m_{(i)})^2}{2\delta_{(i)}^2}} \Rightarrow$$

$$\varphi_i = e^{-\frac{(x_{(i)}-m_{(i)})^2}{2\delta_{(i)}^2}} + e^{-\frac{-(\sum_j e^{-\frac{(x_{j(i-1)}-m_{j(i-1)})^2}{2\delta_{j(i-1)}^2}} w_{j(i-1)}-m_{(i)})^2}{2\delta_{(i)}^2}}$$

(3)

Here $y_{i,j}$ denotes the output of the previous step of the neural network as mentioned in Eq. (3) and $j=1, \dots, \text{Number_of_Neurons}$.

3. TRAINING THE RECURRENT RBF NEURAL NETWORKS

The recurrent RBF neural networks have three sets of parameters that are the centers and the standard deviations in the first layer and the weights in the second layer. Different methods can be used to train these parameters.

In this work back propagation has been used for training the parameters in the first layer. The training procedure for

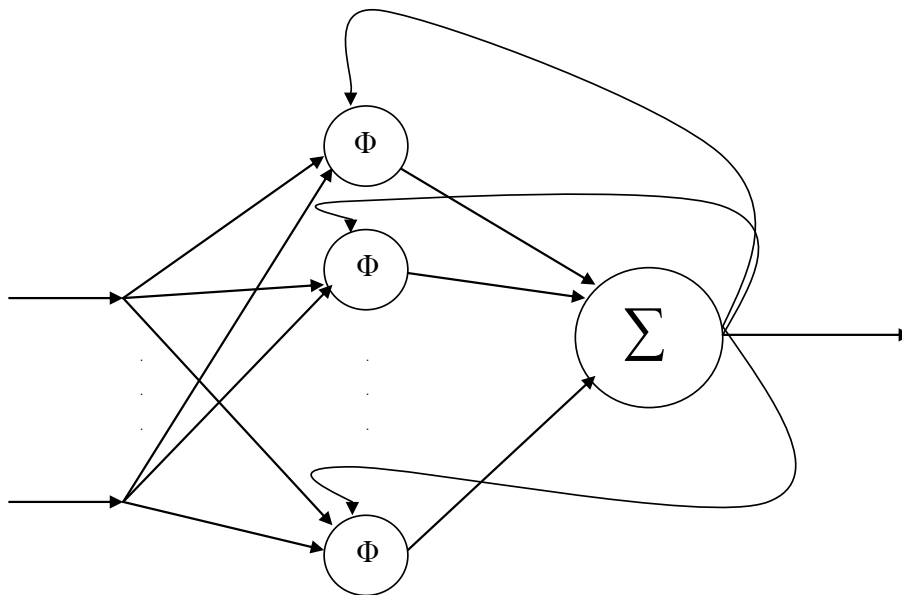


Fig. 2: The structure of RRBF_type2 neural network

standard deviations using back propagation for RRBf_type1 is mentioned in equation 4.

$$\delta_{(i+1)} = \delta_{(i)} - \eta e \times \left[\frac{4\delta_{(i)}(x_{(i)} - m_{(i)})^2}{4\delta_{(i)}^4} \times \frac{e^{-\frac{(x_{(i)} - m_{(i)})^2}{2\delta_{(i)}^2}}}{2\delta_{(i)}^2} \times w_{(i)} + \frac{4\delta_{(i)}(e^{-\frac{(x_{(i-1)} - m_{(i-1)})^2}{2\delta_{(i-1)}^2}} - m_{(i)})^2}{4\delta_{(i)}^4} \times \frac{e^{-\frac{(x_{(i-1)} - m_{(i-1)})^2}{2\delta_{(i-1)}^2}}}{2\delta_{(i-1)}^2} \times w_{(i)} + \frac{w_{(i)} \times e^{-\frac{(x_{(i-1)} - m_{(i-1)})^2}{2\delta_{(i-1)}^2}}}{2\delta_{(i-1)}^2} \times \frac{4\delta_{(i-1)}(x_{(i-1)} - m_{(i-1)})^2}{4\delta_{(i-1)}^4} \times \frac{e^{-\frac{(x_{(i-1)} - m_{(i-1)})^2}{2\delta_{(i-1)}^2}}}{2\delta_{(i-1)}^2} - m_{(i)})^2 \times e^{-\frac{(x_{(i-1)} - m_{(i-1)})^2}{2\delta_{(i-1)}^2}} \right] \quad (4)$$

The centers are also trained by back propagation using the following equation in RRBf_type1.

$$m_{(i+1)} = m_{(i)} - \eta e \left[w_{(i)} \times e^{-\frac{(x_{(i)} - m_{(i)})^2}{2\delta_{(i)}^2}} \times \frac{2(x_{(i)} - m_{(i)})}{2\delta_{(i)}^2} + w_{(i)} \times \frac{2(e^{-\frac{(x_{(i-1)} - m_{(i-1)})^2}{2\delta_{(i-1)}^2}} - m_{(i)})}{2\delta_{(i)}^2} \times \frac{e^{-\frac{(x_{(i-1)} - m_{(i-1)})^2}{2\delta_{(i-1)}^2}}}{2\delta_{(i-1)}^2} + w_{(i)} \times e^{-\frac{(x_{(i-1)} - m_{(i-1)})^2}{2\delta_{(i-1)}^2}} \times \frac{e^{-\frac{(x_{(i-1)} - m_{(i-1)})^2}{2\delta_{(i-1)}^2}}}{2\delta_{(i-1)}^2} - m_{(i)})}{2\delta_{(i-1)}^2} \times \frac{e^{-\frac{(x_{(i-1)} - m_{(i-1)})^2}{2\delta_{(i-1)}^2}}}{2\delta_{(i-1)}^2} \right] \quad (5)$$

$$e^{-\frac{(x_{(i-1)} - m_{(i-1)})^2}{2\delta_{(i-1)}^2}} \times \frac{2(x_{(i-1)} - m_{(i-1)})}{2\delta_{(i-1)}^2} \quad (5)$$

The training procedure for standard deviations using back propagation for RRBf_type2 is also given below.

$$\delta_{(i+1)} = \delta_{(i)} - \eta e \times \left[\frac{4\delta_{(i)}(x_{(i)} - m_{(i)})^2}{4\delta_{(i)}^4} \times \frac{e^{-\frac{(x_{(i)} - m_{(i)})^2}{2\delta_{(i)}^2}}}{2\delta_{(i)}^2} \times w_{(i)} + \frac{4\delta_{(i)}(y_{(i-1)} - m_{(i)})^2}{4\delta_{(i)}^4} \times \frac{e^{-\frac{(x_{(i)} - m_{(i)})^2}{2\delta_{(i)}^2}}}{2\delta_{(i)}^2} \times w_{(i)} + w_{(i)} \times \frac{-2(y_{(i-1)} - m_{(i)})}{2\delta_{(i)}^2} \times e^{-\frac{(y_{(i-1)} - m_{(i)})^2}{2\delta_{(i)}^2}} \times \frac{\sum_j e^{-\frac{(x_{j(i-1)} - m_{j(i-1)})^2}{2\delta_{j(i-1)}^2}} (x_{j(i-1)} - m_{j(i-1)})^2 \delta_{(i-1)} w_{j(i-1)}}{2\delta_{j(i-1)}^4} \right] \quad (6)$$

The centers for RRBf_type2 are trained in the following way.

$$m_{(i+1)} = m_{(i)} - \eta e \left[w_{(i)} \times e^{-\frac{(x_{(i)} - m_{(i)})^2}{2\delta_{(i)}^2}} \times \frac{2(x_{(i)} - m_{(i)})}{2\delta_{(i)}^2} + w_{(i)} \times \frac{2(y_{(i-1)} - m_{(i)})}{2\delta_{(i)}^2} \times \frac{e^{-\frac{(y_{(i-1)} - m_{(i)})^2}{2\delta_{(i)}^2}}}{2\delta_{(i)}^2} + w_{(i)} \times e^{-\frac{(y_{(i-1)} - m_{(i)})^2}{2\delta_{(i)}^2}} \times \frac{(y_{(i-1)} - m_{(i)})}{\delta_{(i)}^2} \times \frac{\sum_j e^{-\frac{(x_{j(i-1)} - m_{j(i-1)})^2}{2\delta_{j(i-1)}^2}} \times w_{j(i-1)} \times (x_{j(i-1)} - m_{j(i-1)})}{\delta_{j(i-1)}^2} \right] \quad (7)$$

LS (Least Square) is used here to calculate the weights in the second layer. After matrix Φ is calculated in the first layer of the network, the weight vector of the second layer is calculated as expressed here:

$$\underline{W} = \begin{bmatrix} w1 \\ \vdots \\ w11 \end{bmatrix} = \text{inv}(\varphi^T \text{inv}(\varphi)) \varphi^T \times d \quad (8)$$

The shortcoming of this method is matrix φ 's dimensions that is related to the number of our input data and if it is large then computing the inverse of this matrix is really difficult or impossible therefore using RLS is proposed that is calculated as follows.

$$PO = \alpha I$$

$$\begin{aligned} k(n) &= P(n-1)\varphi(n)[1 + (\varphi(n))^T P(n-1)\varphi(n)]^{-1} \\ w(n) &= w(n-1) + k(n)(y(n) - (\varphi(n))^T w(n-1)) \\ P(n) &= (I - k(n)(\varphi(n))^T)P(n-1) \end{aligned} \quad (9)$$

Here α is a constant that is usually a big number. φ is the Gaussian function calculated in the first layer and y is the desired output of our function. After calculating P and k the weights will be updated. It is prior to LS as the weights are calculated one by one here and there is no need for an inverted matrix.

4. THE SIMULATION RESULTS

The recurrent RBF neural networks have been used for identification of the three test functions mentioned below and a set of data that is related to a CD player. Eleven neurons have been used found by 'try and error'. The test functions are given in equation 10.

$$f_1 = \frac{\sin x}{x} \times \frac{\sin y}{y} \quad (10)$$

$$\begin{aligned} f_2 &= y(k+1) = 0.3y(k) + 0.6y(k-1) + g(u(k)) \\ g(u) &= 0.6\sin(\pi u) + 0.3\sin(3\pi u) + 0.1\sin(5\pi u) \end{aligned} \quad (11)$$

$$f_3 = 1 + x^{0.5} + y^{-1} + z^{-1.5} \quad (12)$$

The CD player data has been downloaded from the site <http://homes.esat.kuleuven.be/~smc/daisy/daisydata.html>.

The simulations are done in Matlab using 100 epochs and the results achieved from RRBf_type1 are given in table 1.

Table 1- The simulation results from RRBf_type1

	Test function and the second layer's training method	Test RMSE
1	f_1 with RLS	0.0023
2	f_2 with RLS	0.0236
3	f_3 with RLS	0.0032

4	CD player with RLS	0.0156
5	f_1 with LS	0.0035
6	f_2 with LS	0.0354
7	f_3 with LS	0.0019
8	CD player with LS	0.0168

The first function has 121 train and 71 test data whereas the second function has 101 train and 101 test data and the third function has 216 train and 125 test data. For the CD player, about 70 percent of the data has been used for training the network and the rest has been used for the test stage.

The related figures are shown in fig. 3~10.

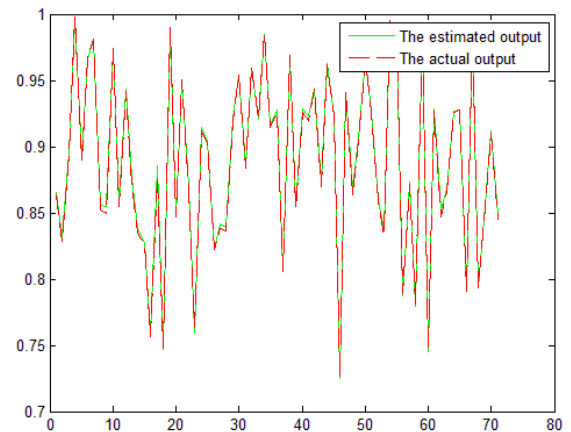


Fig. 3: The test result of F1 with RLS

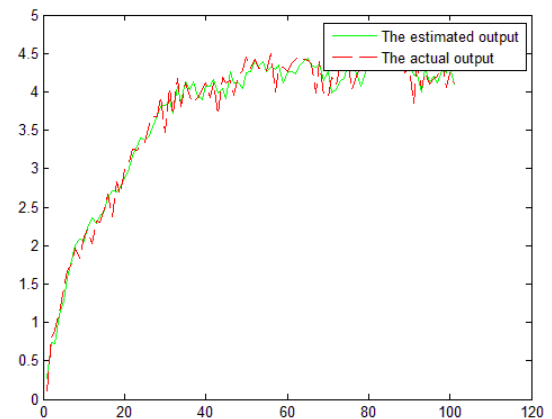


Fig. 4: The test result of F2 with RLS

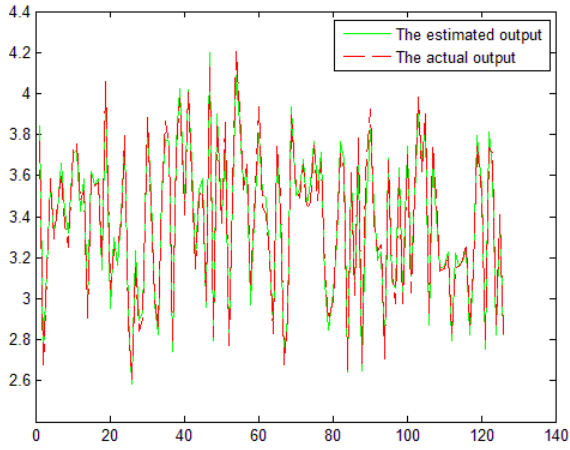


Fig. 5: The test result of F3 with RLS

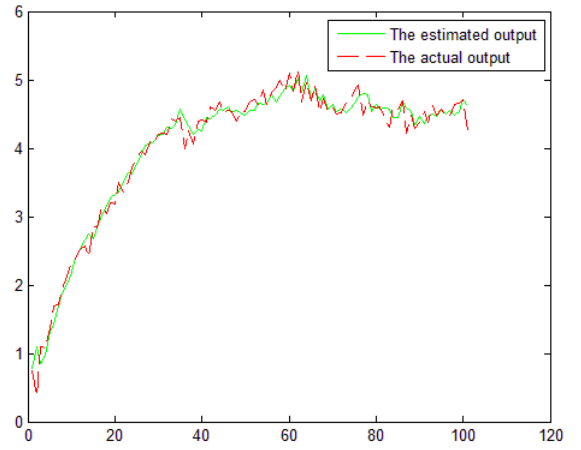


Fig. 8: The test result of F2 with LS

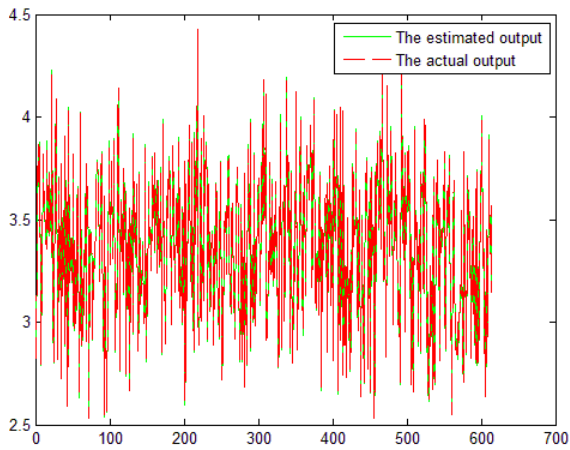


Fig. 6: The test result of CD player with RLS

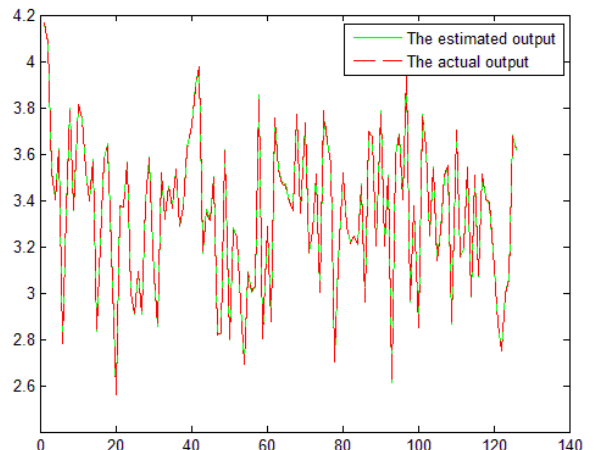


Fig. 9: The test result of F3 with LS

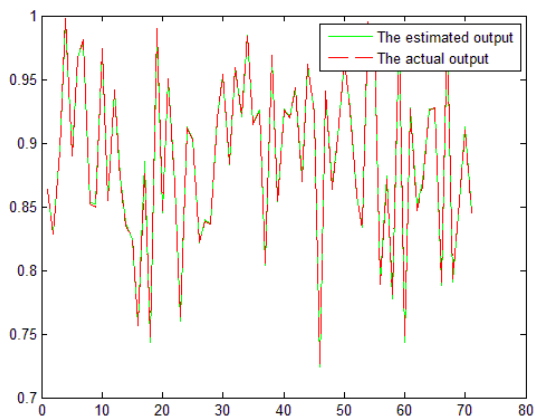


Fig. 7: The test result of F1 with LS

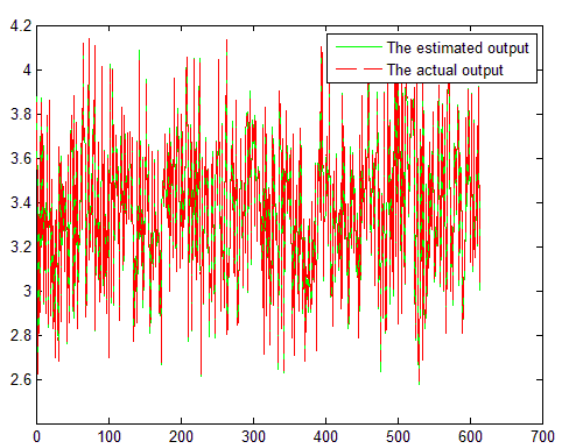


Fig. 10: The test result of CD player with LS

The simulation results achieved from RRBf_type2 are given in table 2. The same number of test and train data has been used for this network too.

Table 2- The simulation results from RRBf_type2

	Test function and the second layer's training method	Test RMSE
1	f_1 with RLS	0.0021
2	f_2 with RLS	0.0309
3	f_3 with RLS	0.0021
4	CD player with RLS	0.0082
5	f_1 with LS	0.0027
6	f_2 with LS	0.0235
7	f_3 with LS	0.0031
8	CD player with LS	0.0090

The related figures are shown in fig. 11~18.

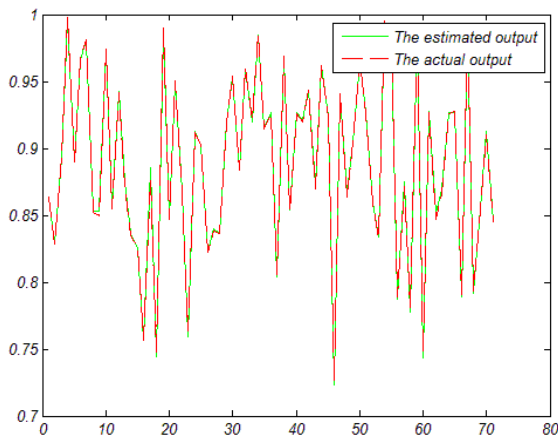


Fig. 11: The test result of F1 with RLS

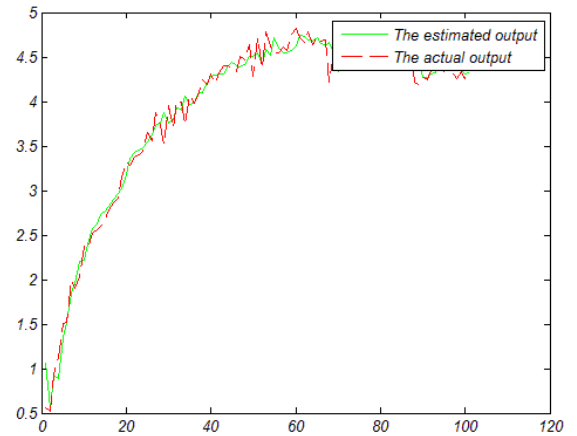


Fig. 12: The test result of F2 with RLS

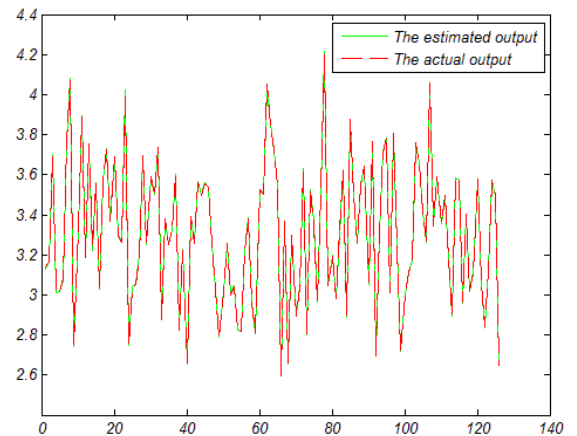


Fig. 13: The test result of F3 with RLS

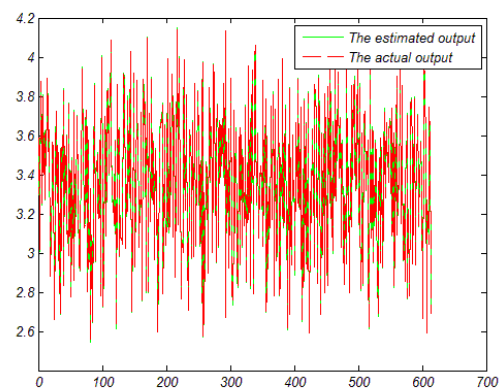


Fig. 14: The test result of CD player with RLS

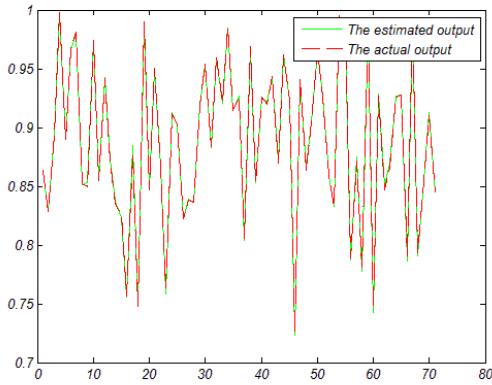


Fig. 15: The test result of F1 with LS

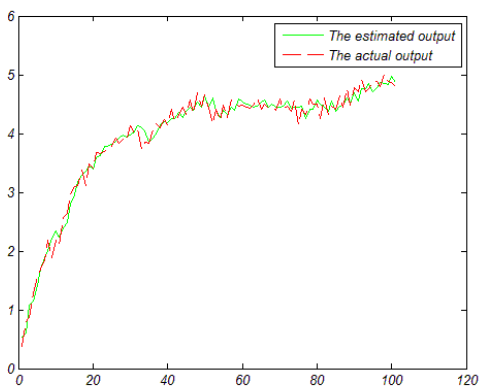


Fig. 16: The test result of F2 with LS

The results show that the proposed recurrent RBF neural networks, RRBf_type1 and RRBf_type2, have been able to achieve better results compared with the previous works [4, 5, 9].

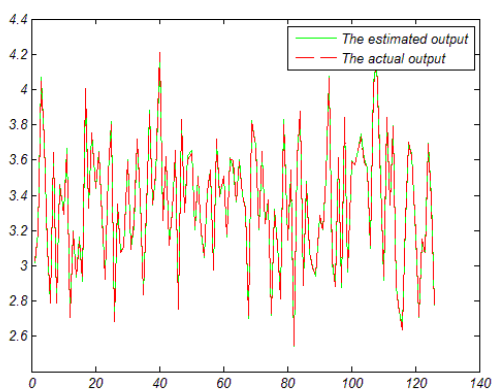


Fig. 17: The test result of F3 with LS

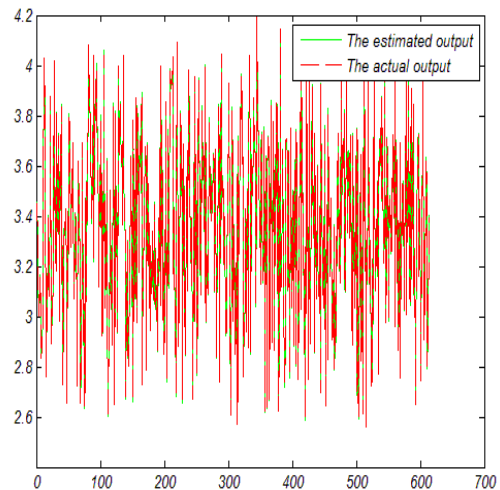


Fig. 18: The test result of CD player with LS

5. CONCLUSION

The proposed recurrent RBF neural networks, RRBf_type1 and RRBf_type2, have been tested on four identification problems and the results show that the proposed neural networks are more powerful in identification than the normal RBF network. The recurrent RBF neural network uses both the advantages of the RBF neural network and the recurrent networks together to improve the efficiency. These kinds of neural network can be used for other applications in science and engineering and they are expected to have a better performance in compare with non-recurrent neural networks and the reason for this lies on the dynamic peculiarity of the proposed network. Being able to apply the data related to the past steps other than the current steps give better generalization ability to the proposed neural networks. In cases that accuracy and generalization are considered to be very important in real world applications, the proposed recurrent RBF neural networks can be used as a good alternative for the other commonly used types of neural networks.

6. REFERENCES

- [1] J. Park and J. Wsandberg, "Universal approximation using radial basis functions network," *Neural Comput.*, vol. 3, pp. 246–257, 1991.
- [2] S. Lee and R. M. Kil, "A Gaussian potential function network with hierarchically self-organizing learning," *Neural Netw.*, vol. 4, pp. 207–224, 1991.
- [3] J. Moody and C. J. Darken, "Fast learning in network of locally-tuned processing units," *Neural Computat.*, vol. 1, pp. 281–294, 1989.
- [4] N. Shafiabady, M. Teshnehlab and M. Aliyari, "A Comparison of PSO and Backpropagation Combined with LS and RLS in Identification Using Fuzzy Neural Networks", ICIT 2006.
- [5] N. Shafiabady, M. Teshnehlab, M. Aliyari, "Training Matrix Parameters by Particle Swarm Optimization Using a Fuzzy Neural Network for Identification", *icias2007*.

- [6] S.Haykin, “*Neural Networks- A Comprehensive Foundation*“, Macmillan College Publ. Co., New York, 1994.
- [7] A. G. Parlos, S. K. Menon and A. F. Atiya, “An Algorithm Approach to Adaptive State Filtering Using Recurrent Neural Network”, *IEEE Transactions on Neural Networks*, Vol.12, No.6, 1411-1432, 2001.
- [8] E.B. Kosmatopoulos, M.M. Polycarpou, M.A. Christodoulou and P.A. Ioannou, “High-Order Neural Network Structures for Identification of Dynamical Systems”, *IEEE Transactions on Neural Networks*, Vol.6, No.2, 422-431, 1995.
- [9] S.R. Jang, "ANFIS: Adaptive-Network-Based Fuzzy Inference System", *IEEE Trans. on Syst., Man and Cybern.*, Vol. 23, pp. 665-685, 1993 .