Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2010, Article ID 124014, 16 pages
doi:10.1155/2010/124014

Research Article

Modelling of the Automatic Depth Control Electrohydraulic System Using RBF Neural Network and Genetic Algorithm

Xing Zong-yi,[1] Qin Yong,[2] Pang Xue-miao,[1] Jia Li-min,[2] and Zhang Yuan[2]

[1] *School of Mechanical Engineering, Nanjing University of Science & Technology, Jiangsu 210094, China*
[2] *State Key Laboratory of Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China*

Correspondence should be addressed to Xing Zong-yi, xingzongyi@gmail.com

Received 1 February 2010; Revised 30 June 2010; Accepted 4 November 2010

Academic Editor: Tamas Kalmar-Nagy

The automatic depth control electrohydraulic system of a certain minesweeping tank is complex nonlinear system, and it is difficult for the linear model obtained by first principle method to represent the intrinsic nonlinear characteristics of such complex system. This paper proposes an approach to construct accurate model of the electrohydraulic system with RBF neural network trained by genetic algorithm-based technique. In order to improve accuracy of the designed model, a genetic algorithm is used to optimize centers of RBF neural network. The maximum distance measure is adopted to determine widths of radial basis functions, and the least square method is utilized to calculate weights of RBF neural network; thus, computational burden of the proposed technique is relieved. The proposed technique is applied to the modelling of the electrohydraulic system, and the results clearly indicate that the obtained RBF neural network can emulate the complex dynamic characteristics of the electrohydraulic system satisfactorily. The comparison results also show that the proposed algorithm performs better than the traditional clustering-based method.

1. Introduction

The Automatic Depth Control Electrohydraulic System (ADCES) of a certain minesweeping tank (see Figure 1) is a complex nonlinear electrohydraulic system. The first step in designing a high-performance ADCES controller is to model the ADCES accurately. The traditionally and widely used approaches for modelling of such electrohydraulic system are based on the first principle method; that is, a linear model of the ADCES can be derived according to some physical laws such as the dynamic equation of valve and the force balance equation [1, 2]. However, the ADCES exhibits high nonlinear behaviors which make the linear

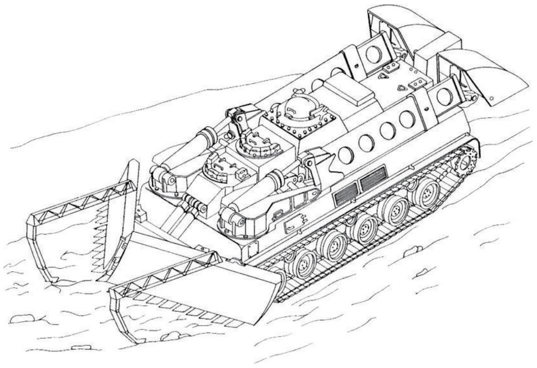


Figure 1: Graphical diagram of a certain minesweeping tank.

model obtained by the first principle method inefficient because the linear model cannot accurately describes such nonlinearities of the ADCES as flow/pressure characteristics, fluid compressibility and friction, and so forth [3]. It is highly desirable to develop a precise model of the ADCES which can be used for the following high-performance controller design.

In recent years, neural networks have been shown efficient alternatives to first principle models due to their abilities to describe highly complex and nonlinear problems in many fields of engineering [4]. Numerous applications of neural networks in electrohydraulic systems have been reported. He and Sepehri [5] demonstrated experimental modelling of dynamic behaviors of an industrial hydraulic actuator with neural network for the first time, and the results showed that neural network is capable of modelling and predicting characteristics of the highly nonlinear hydraulic actuator. Kang et al. [6] proposed a model-following adaptive algorithm using multiple neural networks to control the angle of variable displacement pump. The neural networks were used for numerical simulation and parameters adjustment. However, Most of these researches focused on the usage of multilayer perceptron (MLP) neural networks trained by back-propagation learning algorithm, which have some disadvantages such as slow training speed, local minimal convergence behavior, sensitivity to the randomly selected initial weight values, difficulty in explicit optimum network configuration. To solve these problems, Radial Basis Function (RBF) neural network can be used. Compared with MLP neural networks, RBF neural networks have only one hidden units, while MLP neural networks have one or more hidden layers; the output layer of RBF neural network is linear, while the output layer of MLP neural network is nonlinear. These characteristics make RBF neural networks having more advantages such as simple architecture and fast training speed, near-optimal solution, easy optimization of topology, and so forth. Hence, RBF neural networks have been used extensively in systems modelling [7–9].

In this paper, RBF neural network is employed to develop an accurate model for the ADCES, and an approach based on genetic algorithm is proposed to train RBF neural network. In order to improve precision of RBF neural network, a genetic algorithm is used to optimize center parameters of RBF neural network instead of traditionally used clustering-based methods. The width and the weight parameters are calculated using some fast linear techniques in order to relieve computational burden and accelerate the convergence of the proposed training technique. To our best knowledge, this is the first application of RBF neural network with genetic algorithm to model an electrohydraulic system intently and intensively.

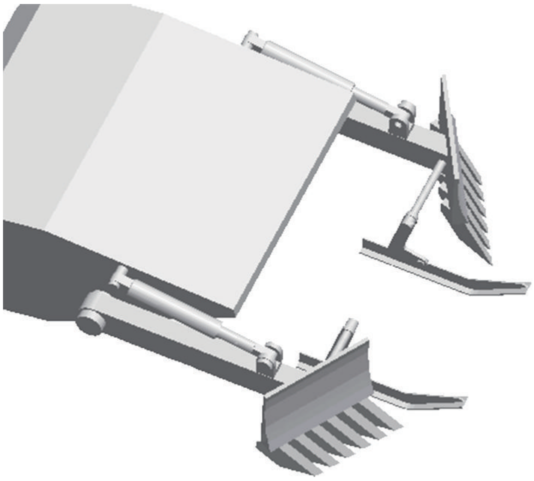


Figure 2: Schematic drawing of the ADCES.

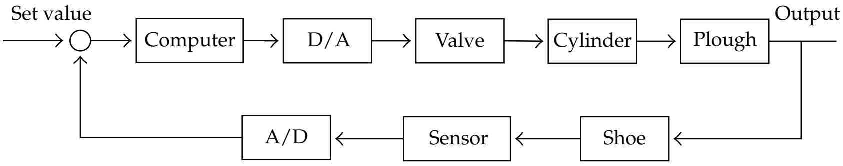


Figure 3: Block diagram of the ADCES.

The paper is organized as follows. Section 2 describes the ADCES. Section 3 devotes to explain how to construct RBF neural network using the proposed genetic algorithm-based technique. In Section 4, the proposed method is applied to modelling of the ADCES, and results of experiments and comparisons with other algorithms are also illustrated. Finally, some conclusions are presented in Section 5.

2. Automatic Depth Control Electrohydraulic System

The automatic depth control electrohydraulic system (ADCES) is composed of five main parts: proportional valves, hydraulic cylinders, copying shoes, shaft position encoders and blades, as illustrated in Figure 2. The working principle of ADCES can be briefly described as follows. During mines ploughing operation, vertical terrain changes can be detected by the copying shoes; and the angles between the plough arms and level plane can be measured by the encoders linked with copying shoes, then the ploughing depth of the plough can be calculated according to the measured angles. The automatic depth control is accomplished by oscillation movement of the hydraulic cylinders which are operated by the proportional valves according to error between the measured depth and the target value. Figure 3 shows the block diagram of the ADCES which indicates that the ADCES is a typical valve-controlled-cylinder electrohydraulic system.

In order to motivate the ADCES sufficiently and collect complete data containing all dynamic characteristics of the ADCES, it is important to select an appropriate input signal for the ADCES. In the field of linear system identification, the Pseudorandom binary signal (PRBS) that only contains two amplitude levels is widely used. However, the identifiability will be lost for the nonlinear ADCES if the PRBS is also adopted. So an input signal that contains all interesting amplitudes and frequencies and all their combinations should be employed, such as Pseudorandom MultiLevel Signals (PRMS), chirp signals, and independent sequences with a Gaussian or uniform distribution. Experience shows that the PRMS is the most suitable choice of input signal for identification of a hydraulic system [10]. So in this paper the PRMS is selected as the input signal for the ADCES.

Let $v(k)$ be a white noise. A PRMS is obtained by keeping the same amplitude value for N_s steps:

$$u(k) = v\left[\operatorname{int}\left(\frac{k-1}{N_s}\right) + 1\right] \quad k = 1, 2, \ldots, \tag{2.1}$$

where $\operatorname{int}(x)$ is the integer part of x.

A generalization of this signal is generated by introducing an additional random variable for deciding when to change the amplitude level

$$u(k) = \begin{cases} u(k) & \text{with probability } \alpha, \\ u(k-1) & \text{with probability } 1-\alpha. \end{cases} \tag{2.2}$$

In the ADCES, there are fixed single-input single-output mapping functions among the displacement of the cylinder piston, the angle measured by the encoder and the actual ploughing depth. So, without loss of generality, the control voltage of the proportional valve is adopted as the input of the ADCES, and the displacement of piston is adopted as the output of the ADCES. Although the ADCES is a high-order nonlinear system, it will not be vibrated within the normal input allowed. So the experiment to gather data is conducted without any closed loop controller. With 100 ms sampling time, 1000 data are collected, as illustrated in Figure 4(a) presents the input data, and Figure 4(b) shows the output data.

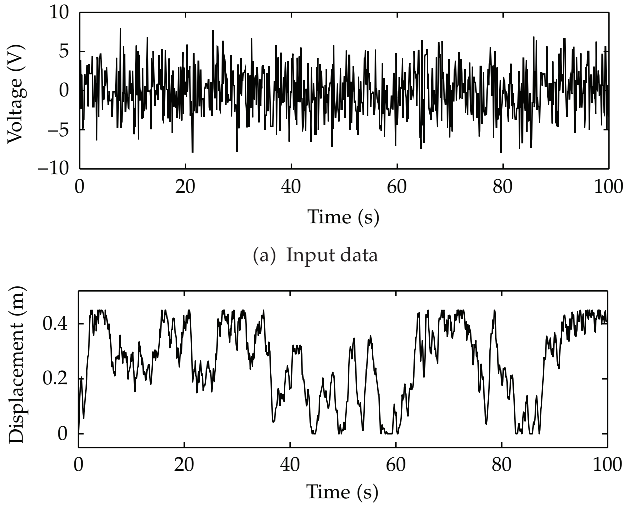
3. RBF Neural Network and the Proposed Training Technique

3.1. RBF Neural Network

The RBF neural network (RBFNN) is a three-layer feedforward neural network which consists of input layer, signal hidden layer and output layer, as depicted in Figure 5. The input layer consists of neurons which correspond to the elements of input vector. These neurons do not process the input information; they only distribute the input vector to the hidden layer. The hidden layer does all the important mathematical process. Each neuron of the hidden layer employs a radial basis function as nonlinear transfer function to operate the received input vector and emits output value to the output layer. The output layer implements a linear weighted sum of the hidden neurons and yields the output value.

A typical radial basis function that is used in this paper is the Gaussian function which assumes the form

$$\phi_m(x) = e^{-\|x-c_m\|^2/\sigma_m^2}, \tag{3.1}$$



(a) Input data

(b) Output data

Figure 4: Collected input-output data of the ADCES.

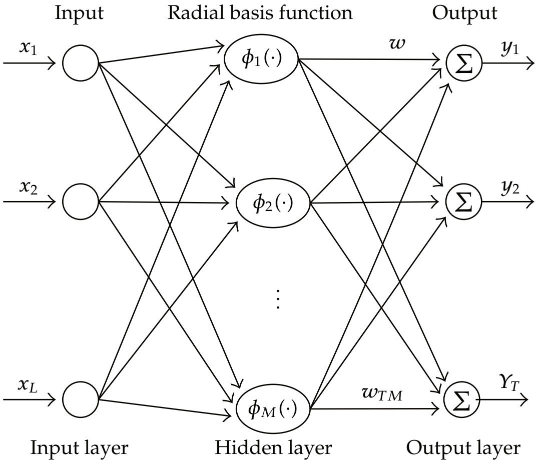


Figure 5: Radial basis function neural network.

where x is input vector, c_m is the center of RBFNN, $\|x - c_m\|$ denotes the distance between x and c_m, and σ is the width.

The output of the RBFNN has the following form:

$$y_t(x) = \sum_{m=1}^{M} w_{tm}\phi_m(x) + b_t, \tag{3.2}$$

where M is the number of independent basis functions, w_{tm} is the weight associated with the mth neuron in the hidden layer and the tth neuron in the output layer, and b_t is the bias of the tth neuron.

In general, there are three types of adjustable parameters which should be determined for the RBF neural network: basis function center, basis function width and output weight. Many methods [11–17] have been proposed for learning these parameters, which can be divided into two stages. The first stage includes the selection of appreciate centers and widths for the radial basis functions, which is a nonlinear problem. The second stage involves the adjustment of the output weights, which is a linear problem. Unsupervised learning algorithm, such as clustering-based method [14] or the orthogonal least square method [15], can be applied to the first stage, whereas linear algebra solutions, such as the least square method [11] or gradient descent algorithm [17], can be applied to the second stage.

In the process of choosing centers for the RBF neural network, the randomly selected input vectors from training set can be chosen as centers [16]. This approach may lead to acceptable results by trial-and-error if the training data are distributed in a representative manner. Nevertheless, this approach is more likely to generate large scale networks, and lead to over fitting and numerical problems. Another method is the gradient descent algorithm [17] which has applied to multiperceptron successfully; however, this method is computationally expensive and more prone to get trapped in local optima. Thirdly, the most commonly used approach employs clustering techniques, such as K-Means method [14] or the orthogonal least square method [15] to determine centers of neural network. This type of learning algorithm considers only the information of input data vectors, ignoring the output space and weights of output layer, and the performance of this approach is also sensitive to the randomly selected initial values. Therefore, the centers obtained by this clustering-based method may not be optimal with respect to the accuracy performance of the final results [18].

The training of RBF neural network can be seen as an optimization problem, where the modelling accuracy can be maximized by adjusting the parameters of neural network. Genetic algorithm (GA) is a parallel and robust optimization technique inspired by the mechanism of evolution and genetics, and it has been successfully applied to innumerable search and optimization problems [19]. Many researches have devoted to the study of training RBF neural network by GA, and the results indicate that the adoption of GA for determining the parameters of RBF neural network can avoid local minimum and improve precision performance [20–26]. However, most reported publications mentioned above estimate two types of parameters [21, 22], that is, centers and widths, or three types of parameters [23], that is, centers, widths and weights, of the RBF neural network using genetic algorithm. And these approaches suffer from problems of difficulty on structure determination and heavy computational burden due to large search space of the optimization problem [27].

Therefore, this paper proposes an approach to design RBF neural network for modelling of the ADCES based on genetic algorithm. Different from the researches mentioned above, genetic algorithm is employed only to evolve center parameters of RBF neural network, while width and weight parameters are calculated using some fast linear techniques in order to relieve computational burden and accelerate the convergence of the proposed training technique.

The width parameters of RBFNN control the domain of influence of corresponding radial basis functions. In order to obtain more accurate RBFNN, different width value is used for each radial basis function. The width of the ith center is set to the maximum Euclidean distance between ith center c_i and its candidate center c_j

$$\sigma_i = \max\left(\left\|c_i - c_j\right\|\right), \quad j = 1, 2, \ldots, M, \ j \neq i. \tag{3.3}$$

After the centers and widths have been fixed, the weights of output layer can be calculated by an algorithm suitable to solve the linear algebraic equations. In this paper, the output weights are computed by the least square algorithm.

Let

$$\Phi = \begin{bmatrix} \phi_1(x_1) & \phi_2(x_1) & \cdots & \phi_M(x_1) & 1 \\ \phi_1(x_2) & \phi_2(x_2) & \cdots & \phi_M(x_2) & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \phi_1(x_N) & \phi_2(x_N) & \cdots & \phi_M(x_N) & 1 \end{bmatrix}, \tag{3.4}$$

then the weights can be calculated using the least square algorithm

$$\Phi^{+} y = \left(\Phi^{T}\Phi\right)^{-1}\Phi^{T} y, \tag{3.5}$$

where Φ^{+} is the pseudoinverse of Φ and y is the target output data.

3.2. Design of RBF Neural Network Based on Genetic Algorithm

Genetic algorithm has been successfully employed in search and optimization problems by simulating natural evolution. Genetic algorithm has a population of individuals competing against each other in relation to a fitness function, with some individuals breeding, others dying off, and new individuals arising through crossover and mutation. In this paper, genetic algorithm is used to optimize the centers of RBF neural network. The following segments present the main areas where genetic algorithm applies to design of RBF neural network.

Genetic encoding, the choice of appropriate encoding for individuals is the first step for optimization of RBF neural network by genetic algorithm. Traditionally, encoding scheme uses binary strings. However, the bit strings of binary-coded genetic algorithm become very long and the search space blows up, while in real-coded genetic algorithm, the variables appear directly in chromosome simply, and computation burden is relieved, so real-coded scheme is adopted in this paper.

Genetic operators: There are three types of operators in genetic algorithm, that is, selection, crossover and mutation. The selection operator employs a fitness function to evaluate the individuals from the population, and selects parts of individuals for the following crossover and mutation. In this paper, the roulette wheel selection is used to choose individuals from population to operate. In order to prevent optimal chromosomes from being ignored, elitist selection is also employed, that is, the best chromosomes are always preserved in population. Crossover operator produces offspring individuals by combining genes of parent individuals. The two crossover operators used here are the simple arithmetic crossover and the whole arithmetic crossover, which are selected randomly during the process of evolution. Mutation operator is a stochastic variation of the genes of individuals. The uniform mutation and the Gaussian mutation are employed randomly during the process of evolution.

Fitness function: The fitness function is used to evaluate performance of individuals for selection. As to the modelling of the ADCES, a high precise RBF neural network is desired, so the root mean square error (RMS) which is most widely used for modelling problem is employed as the fitness function of genetic algorithm,

$$\mathrm{RMS}(y, y_t) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y(i) - y_t(i)\right)^2}, \tag{3.6}$$

where y is the measure outputs and y_t represents outputs of the neural network, and N is the number of data.

Stop Criteria

The evolution process will repeat for a fixed number of generations or being ended when the value of objective function satisfies a given accuracy performance. In the proposed approach, individuals evolve for a predefined generations, and the neural network with minimum testing error is selected for each generation. At the end of evolution, the neural network with minimum testing error will be selected as the optimal neural network.

The proposed genetic algorithm-based approach for training RBF neural network can be summarized in the following steps.

(1) Randomly choose an initial population with a number of individuals. Each individual associates the centers of an RBF neural network.

(2) Compute the widths and weights of RBF neural networks. The outputs of RBF neural networks can be obtained, and the fitness functions of initial population can also be calculated.

(3) Apply three genetic operators to the parent individuals, and the offspring individuals are generated.

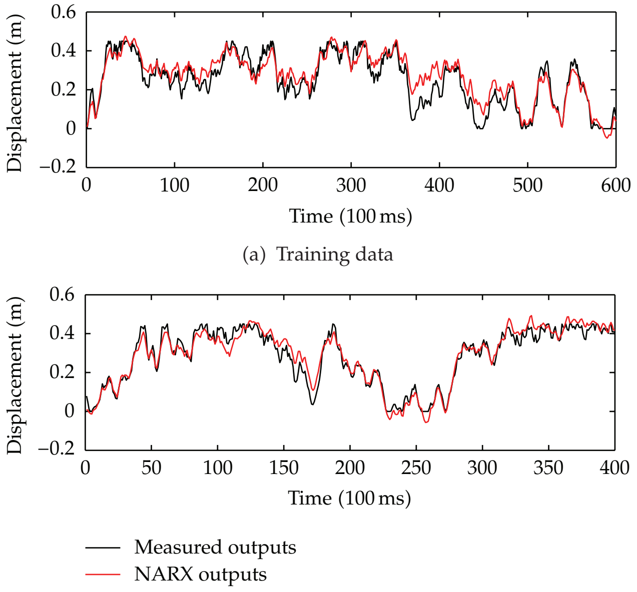
(4) Calculate the widths and weights of RBF neural network, and compute the fitness function of each offspring individual.

(5) If the number of generation is equal to the given threshold, then stop and return the best solution, otherwise go to step (3).

4. Experiments and Results

This section presents the application of the proposed genetic algorithm-based approach to evolve RBF neural network for modelling of the ADCES of a certain minesweeping plough.

To determine the nonlinearity of the ADCES, the second-order Nonlinear Auto-Regressive with eXtra inputs (NARX) is employed to model the ADCES [28]. The data are divided into two parts: the first 600 data are used to train the NARX model, while the other 400 data are employed to test the obtained NARX model. The comparison of outputs is illustrated in Figure 6, and the regression analysis of testing data is also shown in Figure 7 to assess the accuracy of the NARX model. The training RMS error and test RMS error of the NARX model are 0.0392 and 0.0427, respectively. The obtained results indicate that the ADCES is a typical nonlinear system.



(a) Training data

(b) Test data

Figure 6: Comparison of measured outputs and NARX outputs.

In order to accelerate the speed of convergence and improve the effectiveness of the proposed algorithm, the collected data are scaled between zero and one

$$x_i^{\text{scal}} = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}, \tag{4.1}$$

where x_i, $x_{\max}$, and $x_{\min}$ are the original, the maximum and the minimum values, respectively, x_i^{scal} is the value which has been preprocessed.

The lag space, that is, the number of delayed inputs and outputs, of RBF neural network is chosen as two [29], so the constructed RBF neural network is a model with four inputs and one output, thus the training set includes 598 samples and the test set includes 398 samples.

In order to compare the performance of different models of the ADCES, the Root Mean Square error (RMS) defined in Section 3.2 (see (3.6)) is applied to measure the accuracy of the obtained model.

With small value of displacement, the RMS of the model is so small that it is difficult to indicate the fitting performance between the obtained model and the ADCES distinctly. So the Variance Accounted For (VAF) is also used to assess the quality of the model by comparing target outputs and outputs of the model:

$$\text{VAF}(y, y_t) = \left[1 - \frac{\text{var}(y - y_t)}{\text{var}(y)}\right] \times 100\%, \tag{4.2}$$

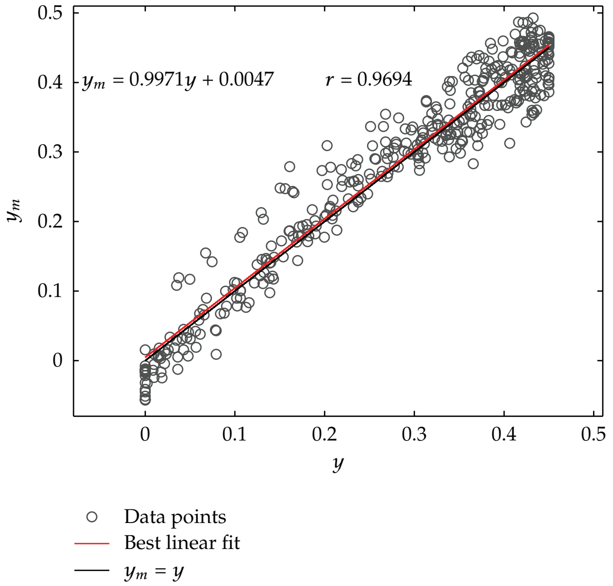


Figure 7: Regression analysis of the NARX for test data.

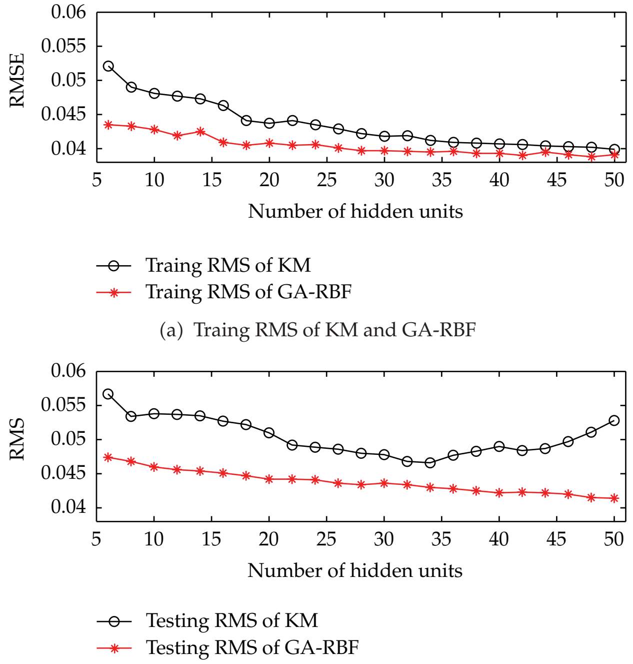
where var() is the variance operation, y is the measure outputs and y_t represents outputs of the neural network. A higher VAF means that the obtained model is more similar to the ADCES.

In order to stand out advantages of the proposed technique, the traditionally and widely used K-Means clustering algorithm [14] is also used to build RBF neural network for comparison. For the sake of simplicity, the proposed approach is abbreviated as GA-RBF, and the clustering-based approach for comparison is abbreviated as KM-RBF.

In the proposed GA-RBF algorithm, the population size is chosen as 40, and the selection rate is 0.8, the crossover rate is 0.8 and the mutation probability is 0.05, the maximum generation is 300.

The number of hidden units greatly influences performance of RBF neural network. If the number is too low, the precision of RBF neural network will be deteriorated. On the other hand, if the network employs too many hidden units, it will trend to overfit the data and increase the computational burden. In this paper, the method to determine the number of hidden units is described as follows: firstly, a number range of hidden units is determined empirically; secondly, a set of RBF neural networks are construed with different number of hidden units; then the number of hidden units of the RBF network with minimum test error is selected as optimum number.

In this paper, both the proposed GA-RBF technique and the traditionally used KM-RBF algorithm are employed to determine the number of hidden units for RBF neural network. For the problem of modelling the ADCES, the minimum number of hidden units is chosen as 6 empirically, and the maximum number of hidden units is 50. The number of hidden unit increases incrementally from 6 to 50 with an increment of 2, thus total 23 RBF neural networks is obtained. The performance of the neural networks with different initial conditions may be varied, so the training algorithms run 10 times and the average precision values of the 10 runs are used to measure the performance of the RBF neural networks.



(a) Traing RMS of KM and GA-RBF

(b) Test RMS of KM and GA-RBF

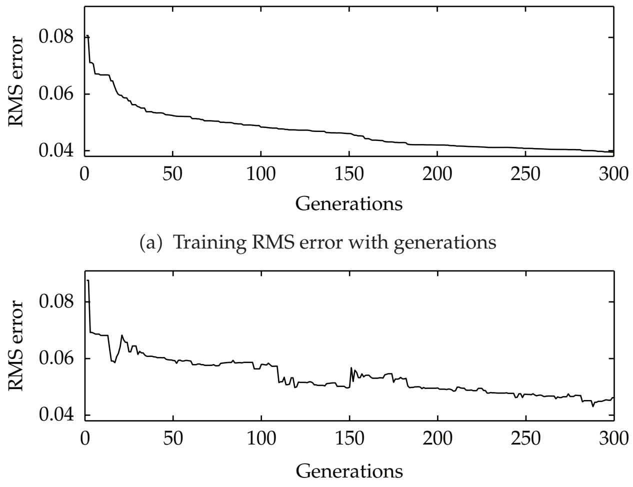
Figure 8: Determination of number of hidden units of RBF neural network.

Figure 8 shows the RMS errors for RBF neural networks with different number of hidden units trained by KM-RBF algorithm and the proposed GA-RBF algorithm. For KM-RBF algorithm, the neural network with 34 hidden units yields the minimum test error (0.0466), and overtraining will be caused for test data if the number of hidden units is more than 34. For the proposed GA-RBF algorithm, the test errors continually reduce with increased number of hidden units, however, the test errors of RBF neural networks only improve 3.72% (from 0.0430 to 0.0414) when the number of hidden units increases from 34 to 50. So taken into account of both KM-RBF algorithm and GA-RBF algorithm, the best number of hidden units of RBF neural network is chosen as 34 eventually.

In order to eliminate the influence of randomly generated initial condition, the proposed genetic algorithm-based approach runs 10 times, and the best result is select as the final RBF neural network.

Figure 9 shows the evolution of the RMS error on both training data and test data for the proposed GA-RBF algorithm with 34 hidden units. The figure illustrates that the training error of RBF neural network decreases steadily during the whole process of evolution. The RMS error of test data decreases and waves a little with increasing generation number and it trend to convergent by the end of evolution. In 288 epoch, the minimum RMS error on test data is obtained (0.0430) corresponding to the error of 0.0400 on training data.

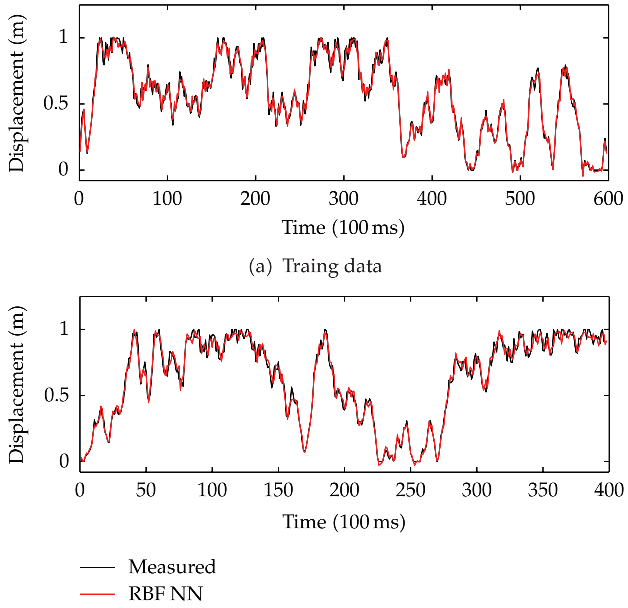
Figure 10(a) shows the outputs of the obtained RBF neural network with 34 hidden units trained by the proposed GA-RBF algorithm as compared to the target outputs for the training data, and Figure 10(b) shows the target outputs and the outputs of the obtained RBF neural network for test data.



(a) Training RMS error with generations

(b) Test RMS error with generations

Figure 9: Evolution of the RMS error on training data and test data.



(a) Traing data

(b) Testing data

Figure 10: Comparison of measured outputs and GA-RBF network outputs.

Figure 11 illustrates the regression analysis of the obtained RBF neural network for test data. It can be seen from Figures 10 and 11 that the predicted outputs of the obtained RBF neural network follow close to the target outputs for both training data and test data. The maximum errors between the target outputs and the predicted outputs of the network are 0.1149 and 0.1231 for training data and test data, respectively. The predicted outputs of RBF neural network match quiet well with the target outputs, which illustrates that the

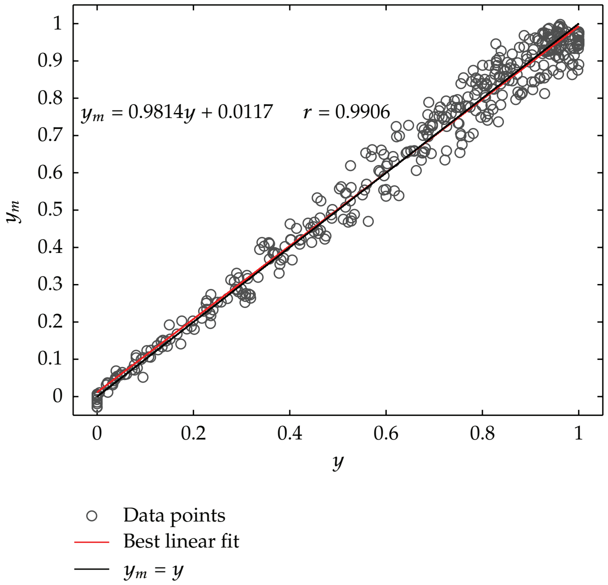
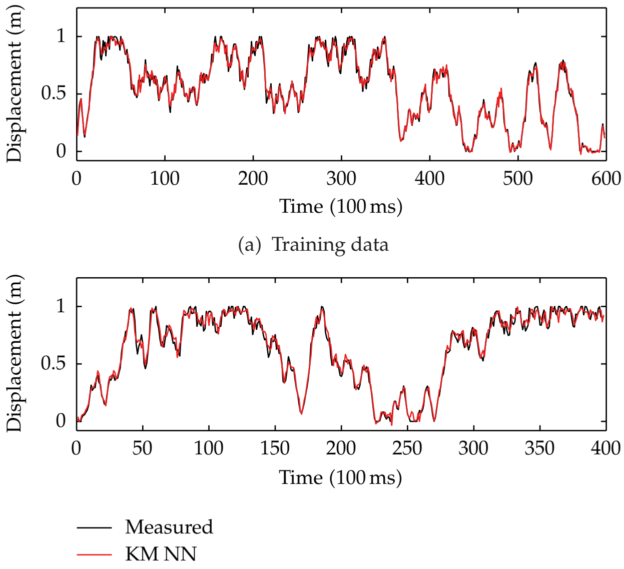


Figure 11: Regression analysis of the GA-RBF neural network for test data.



(a) Training data

(b) Test data

Figure 12: Comparison of measured outputs and KM-RBF network outputs.

dynamic characteristics of ADCES can be emulated reasonably well by the obtained RBF neural network, and that the obtained RBF neural network an be used to model the ADCES successfully.

For standing out advantages of the proposed GA-RBF technique, the widely used KM-RBF algorithm is also employed to train RBF neural network. Figure 12 shows the outputs of the obtained RBF neural network trained by the KM-RBF algorithm as compared to the target

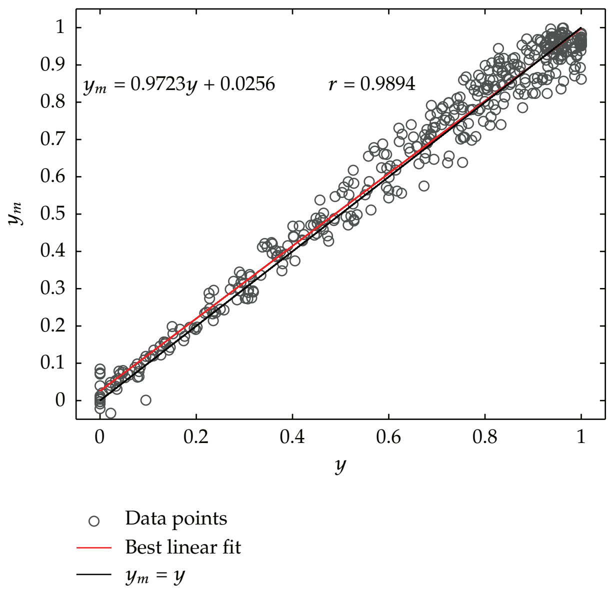


Figure 13: Regression analysis of the KM-RBF neural network for test data.

Table 1: Performance results of different modelling methods.

	Training data		Test data	
	RMS	VAF	RMS	VAF
NARX	0.0897	90.46%	0.2058	86.70%
KM-RBF	0.0431	97.88%	0.0466	97.93%
GA-RBF	0.0395	98.11%	0.0430	98.14%

outputs. Figure 13 illustrates the regression analysis of the obtained RBF neural network for test data with KM-RBF method.

The modelling results of different techniques are summarized in Table 1. It should be pointed out that the performance results of NARX model do not equal to those in Figures 6 and 7, because the data have been normalized for methods in Table 1. For modelling of the ADCES, the obtained RBF neural networks are much more accurate than the nonlinear NARX model. The test error of RBF neural network trained by the proposed GA-RBF approach is smaller than that of neural network constructed by the traditional KM-RBF method. According to the experiment results, the GA-RBF has the best performance in terms of RMS error and VAF performance, and the proposed technique is superior to the traditionally used methods.

5. Conclusions

In this paper, we present a hybrid learning algorithm to construct accurate radial basis function neural network for the ADCES of a certain minesweeping weapon which is a complex nonlinear electrohydraulic system. In order to improve accuracy of the designed neural network, a genetic algorithm is used to optimize centers of neural network. The maximum distance measure is adopted to determine widths of radial basis functions and the least square method is utilized to calculate weights of neural network, thus computational burden of the proposed technique is relieved. The simulation results and comparisons with other algorithm demonstrate effectiveness and validity of the proposed technique.

The next step of our work will be to design high-performance controller of the ADCES based on the obtained neural network [30] and to optimize the structure and parameters of neural network simultaneously based on more advanced genetic algorithm [31, 32].

Acknowledgments

The authors wish to thank the anonymous reviewers for their extensive and very helpful comments and guidance provided in making the paper more acceptable. This work was supported by Science Foundation & Purple Star Foundation and Research Funding (2010GJPY007) of NUST.

References

[1] K. Ziaei and N. Sepehri, "Modeling and identification of electrohydraulic servos," *Mechatronics*, vol. 10, no. 7, pp. 761–772, 2000.

[2] W. Kemmetmüller, S. Müller, and A. Kugi, "Mathematical modeling and nonlinear controller design for a novel electrohydraulic power-steering system," *IEEE/ASME Transactions on Mechatronics*, vol. 12, no. 1, pp. 85–97, 2007.

[3] X. Zong-yi, G. Qiang, J. Li-min, and W. Ying-ying, "Modelling and identification of electrohydraulic system and its application," in *Proceedings of the 17th World Congress of the International Federation of Automatic Control*, pp. 6446–6451, 2008.

[4] M. T. Hagan, H. B. Demuth, and M. H. Beale, *Neural Network Design*, PWS Publishing Company, Boston, Mass, USA, 2002.

[5] S. He and N. Sepehri, "Modeling and prediction of hydraulic servo actuators with neural networks," in *Proceedings of the American Control Conference (ACC '99)*, pp. 3708–3712, June 1999.

[6] Y. Kang, M.-H. Chu, Y.-L. Liu, C.-W. Chang, and S.-Y. Chien, "An adaptive control using multiple neural networks for the position control in hydraulic servo system," in *Proceedings of the 1st International Conference on Natural Computation (ICNC '05)*, pp. 296–305, August 2005.

[7] H. B. Celikoglu and H. K. Cigizoglu, "Modelling public transport trips by radial basis function neural networks," *Mathematical and Computer Modelling*, vol. 45, no. 3-4, pp. 480–489, 2007.

[8] X. Yao, Y. Wang, X. Zhang et al., "Radial basis function neural network-based QSPR for the prediction of critical temperature," *Chemometrics and Intelligent Laboratory Systems*, vol. 62, no. 2, pp. 217–225, 2002.

[9] Y. Turhan, S. Tokat, and R. Eren, "Statistical and computational intelligence tools for the analyses of warp tension in different back-rest oscillations," *Information Sciences*, vol. 177, no. 23, pp. 5237–5252, 2007.

[10] M. Jelali and A. Kroll, *Hydraulic Servo-Systems: Modelling, Identification and Control*, Springer, Berlin, Germany, 2003.

[11] M. D. Buhmann, *Radial Basis Functions: Theory and Implementations*, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, UK, 2003.

[12] R. J. Howlett and L. C. Jain, *Radial Basis Function Networks 2: New Advances in Design*, chapter 4, Physica, 2001.

[13] B. Kim and K. Park, "Modeling plasma etching process using a radial basis function network," *Microelectronic Engineering*, vol. 77, no. 2, pp. 150–157, 2005.

[14] J. Moody and C. J. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Computation*, vol. 1, no. 2, pp. 281–294, 1989.

[15] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Transactions on Neural Networks*, vol. 2, no. 2, pp. 302–309, 1991.

[16] R. Neruda and P. Kudová, "Learning methods for radial basis function networks," *Future Generation Computer Systems*, vol. 21, no. 7, pp. 1131–1142, 2005.

[17] L. Guo, DE. S. Huang, and W. Zhao, "Combining genetic optimisation with hybrid learning algorithm for radial basis function neural networks," *Electronics Letters*, vol. 39, no. 22, pp. 1600–1601, 2003.

[18] H. Du, J. Lam, and N. Zhang, "Modelling of a magneto-rheological damper by evolving radial basis function networks," *Engineering Applications of Artificial Intelligence*, vol. 19, no. 8, pp. 869–881, 2006.

[19] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.

[20] C. Harpham, C. W. Dawson, and M. R. Brown, "A review of genetic algorithms applied to training radial basis function networks," *Neural Computing and Applications*, vol. 13, no. 3, pp. 193–201, 2004.

[21] B. A. Whitehead, "Genetic evolution of radial basis function coverage using orthogonal niches," *IEEE Transactions on Neural Networks*, vol. 7, no. 6, pp. 1525–1528, 1996.

[22] E. G. M. de Lacerda and A. C. P. F. de Carvalho, "Evolutionary optimization of RBF networks," in *Proceedings of the 6th Brazilian Symposium on Neural Networks*, pp. 219–224, 2000.

[23] S. Mishra, P. K. Dash, P. K. Hota, and M. Tripathy, "Genetically optimized neuro-fuzzy IPFC for damping modal oscillations of power system," *IEEE Transactions on Power Systems*, vol. 17, no. 4, pp. 1140–1147, 2002.

[24] D. Manrique, J. Ríos, and A. Rodríguez-Patón, "Evolutionary system for automatically constructing and adapting radial basis function networks," *Neurocomputing*, vol. 69, no. 16–18, pp. 2268–2283, 2006.

[25] O. Buchtala, M. Klimek, and B. Sick, "Evolutionary optimization of radial basis function classifiers for data mining applications," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 35, no. 5, pp. 928–947, 2005.

[26] N. Qu, L. Wang, M. Zhu, Y. Dou, and Y. Ren, "Radial basis function networks combined with genetic algorithm applied to nondestructive determination of compound erythromycin ethylsuccinate powder," *Chemometrics and Intelligent Laboratory Systems*, vol. 90, no. 2, pp. 145–152, 2008.

[27] A. J. Rivera, I. Rojas, J. Ortega, and M. J. Jesus, "A new hybrid methodology for cooperative-coevolutionary optimization of radial basis function networks," *Soft Computing*, vol. 11, no. 7, pp. 655–668, 2007.

[28] L. Ljung, *System Identification: Theory for the User*, Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 1999.

[29] X. He and H. Asada, "New method for identifying orders of input-output models for nonlinear dynamic systems," in *Proceedings of the American Control Conference*, pp. 2520–2523, June 1993.

[30] Z. Yuan, S. Zhan, and X. Zongyi, "Modelling and control of the electrohydraulic systems of a certain mine sweeping plough based on neural network," *Fire Control & Command Control*, vol. 35, no. 4, pp. 141–146, 2010.

[31] G. G. Yen, "Multi-objective evolutionary algorithm for radial basis function neural network design," *Studies in Computational Intelligence*, vol. 16, pp. 221–239, 2006.

[32] T. Hatanaka, N. Kondo, and K. Uosaki, "Multi-objective structure selection for RBF networks and its application to nonlinear system identification," *Studies in Computational Intelligence*, vol. 16, pp. 491–505, 2006.