Modelling of Electrohydraulic System using RBF Neural Networks and Genetic Algorithm

Abstract—**this paper presents an approach to model the nonlinear dynamic behaviors of the Automatic Depth Control Electrohydraulic System (ADCES) of a certain mine-sweeping weapon using Radial Basis Function (RBF) neural networks. In order to obtain accurate RBF neural networks efficiently, a hybrid learning algorithm is proposed to train the neural networks, in which centers of neural networks are optimized by genetic algorithm, and widths and centers of neural networks are calculated by linear algebra methods. The proposed algorithm is applied to the modelling of the ADCES, and the results clearly indicate that the obtained RBF neural network can emulate the complex dynamic characteristics of the ADCES satisfactorily. The comparison results also show that the proposed algorithm performs better than the traditional clustering-based method.**

Keywords-electrohydraulic system; neural network; genetic algorithm; modelling

I. INTRODUCTION

The Automatic Depth Control Electrohydraulic System (ADCES) of a certain mine-sweeping weapon is a complex nonlinear electrohydraulic servo system. The first step in designing a high-performance ADCES controller is to model the ADCES accurately. The traditional and widely used approach for the modelling of such electrohydraulic system is based on the first principle methods, i.e. 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 significant nonlinear behaviors which make the linear model obtained by the first principle methods inefficient because the linear model can’t accurately describes such nonlinearities of the ADCES as the flow/pressure characteristics, fluid compressibility and friction, etc. It is highly desirable to develop a precise model of the ADCES which can be used for the following high-performance controller design.

Neural networks have been employed in recent years as an alternative to the first principle models due to their ability to describe highly complex and nonlinear problems in many fields of engineering. Numerous applications of neural networks in electrohydraulic systems have been reported [3, 4]. However, all these papers mentioned above focus on the usage of the multi-layer perceptron neural networks which have some disadvantages such as slow learning speed, local minimal convergence behavior and sensitivity to the randomly selected initial weight values. To solve these problems, Radial Basis Function (RBF) neural networks can be used, which own the merits of simple architecture, small training times and global minimum. A few researches have paid attention to the application of RBF Neural Networks (RBFNN) in electrohydraulic system [5].

In this paper, the RBF neural networks based on hybrid learning algorithm are employed to develop an accurate model for the ADCES of a certain mine-sweeping weapon. In order to improve the accuracy performance of the RBFNN, a genetic algorithm is used to optimize the center parameters of RBFNN in stead of traditionally used clustering-based methods. The width and the weight parameters are calculated using some fast linear techniques, i.e., the maximum distance measure and the least square algorithm, in order to relieve computational burden and accelerate the convergence of the proposed hybrid learning algorithm. To our best knowledge, this is the first application of RBFNN to model an electrohydraulic system intently and intensively with genetic algorithm.

II. THE AUTOMATIC DEPTH CONTROL ELECTROHYDRAULIC SYSTEM

The Automatic Depth Control Electrohydraulic System (ADCES) of a certain type of mine-sweeping weapon is composed of five parts: a proportional valve, a hydraulic cylinder piston, a copying shoe, a shaft position encoder and a plough, as illustrated in Fig.1. In the process of operation of the mine-sweeping weapon, the shape variation of ground surface is detected by the copying shoe, and the encoder linked with the copying shoe measures the angle between the plough arm and level plane, thus the actual embedded depth of the plough can be calculated. The automatic depth control is accomplished by reciprocating movement of the hydraulic cylinder, which is operated by the proportional valve according to error between the actual embedded depth and the target value. In the ADCES, there are fixed single-input single-output mapping functions among the displacement of the piston, the angle measured by the encoder and the actual embedded 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.

In order to motivate the ADCES sufficiently and collect complete data containing all the 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 Pseudo- Random 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 Pseudo-Random Multi-Level 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 [6]. So in this paper the PRMS is selected as the input signal for the ADCES.

III. Methologies: RBF neural network and the proposed learning algorithm

A. *RBFNN and its training algorithm*

The radial basis function neural network is a three-layer feedforward neural network which consists of input layer, signal hidden layer and output layer, as depicted in Fig.2. The input layer consists of neurons which corresponding to the elements of input vector. These neurons does not process the input information, they only distribute the input vector to the hidden layer. The hidden layer does all the important process. Each neuron of the hidden layer employs a radial basis function as nonlinear transfer function to operate the received input vector and emits the 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

$$\phi_m(x) = e^{-\frac{\|x-c_m\|^2}{\sigma_m^2}}$$

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

The output of the RBFNN has the following form

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

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, b_t is the bias of the tth neuron.

In general, three types of adjustable parameters which should be determined for the RBFNN: basis function center c_m, basis function width σ and output weight w_{tm} . Several algorithms available in the literature have been proposed for training 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, for example clustering-based method, can be applied to the first stage, whereas linear algebra solutions, for example least square method, can be applied to the second stage.

The training of the RBFNN can be seen as an optimization problem, where the modelling accuracy can be maximized by adjusting the parameters of the RBFNN. 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. Many researches have devoted to the study of training RBFNN by GA, and the results indicate that the adoption of GA for determining the parameters of RBFNN can avoid local minimum and improve performance [7-10].

In this paper, a hybrid learning algorithm named GA-RBF is proposed to train the RBF neural network, in which the centers are optimized by genetic algorithm, while the widths and weights are calculated using traditional matrix operation described as follows.

The widths of RBFNN control the domain of influence of the 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 [11] between ith center c_i and its candidate center c_j

$$\sigma_i = \max(\|c_i - c_j\|), \quad j = 1,2,\cdots,M \quad j \neq i .$$

After the centers and widths have been fixed, the weights of the 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

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

then the weights can be calculated using the least square algorithm [11],

$$w = \Phi^+ y = (\Phi^T \Phi)^{-1} \Phi^T y ,$$

where Φ+ is the pseudo-inverse of Φ, and y is the target output data.

B. *The proposed learning algorithm*

Genetic algorithm has been successfully employed in search and optimization problems by simulating natural evolution. The GA 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, the GA is used to optimize the centers of RBF neural networks. The following segments present the main areas where the GA applies to RBF neural networks.

Genetic encoding of the GA-RBF algorithm: The choice of the appropriate encoding for the individuals is the first step for the optimization of RBF neural network by the GA. Traditionally, encoding scheme uses binary strings. However, the bit strings of binary-coded genetic algorithm becomes 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 of the GA-RBF algorithm: There are three operators in the GA, i.e., selection, crossover and mutation. The selection operator employs a fitness function to evaluation the individuals from the population, assigning the fitness for each individual according a predefined criterion. In this paper, the *roulette wheel* selection method is used to select individuals to operate. In order to prevent optimal chromosomes from being ignored, elitist selection are also used, i.e., 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. Mutation operator is a stochastic variation of the genes of individuals. The uniform mutation and the Gaussian mutation are employed randomly in the proposed GA-RBF algorithm.

Objective function of the GA-RBF algorithm: The Root Mean Square Error (RMSE) which is most widely used for modelling problem is employed as the objective function of the GA-RBF algorithm.

Stop criteria of the GA-RBF algorithm: The evolution process will repeat for a fixed number of generations or being ended when the objective function satisfies a given accuracy performance. In the proposed approach, the 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 GA-RBF algorithm used to evolve the RBF neural network can be summarized in the following steps.

1) Randomly choose an initial population with a fixed number of individuals. Each individual associates the centers of an RBF neural network.
2) Compute the widths and weights of RBFNN. The outputs of RBFNN 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 RBFNN, and compute the fitness function of each offspring individual.
5) If the number of generation is equal to the given threshold, then stop, otherwise go to step 3.

IV. Experiments and results

This section presents the application of the proposed GA-RBF algorithm to evolve the radial basis function neural network for modelling of the Automatic Depth Control Electrohydraulic System (ADCES) of a certain type of weapon.

In the ADECS, the input signal is the control voltage of servo valve in the range of [-8 8] volt, and the output signal is the displacement of the piston in the range of [0 0.45] meter. Although the ADECS 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 100ms sampling time, 10000 data are collected, as illustrated in Fig.3: (a) presents the input data, and (b) shows the output data. The first 600 data are used to train the model, while the other 400 data are employed to validate the obtained model.

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

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

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 pre-processed.

In order to weigh the performance of different models of the ADCES, the Root Mean Square Error (*RMSE*) is applied to measure the precision of the obtained model

$$RMS(y, y_m) = \sqrt{\frac{1}{N}\sum\nolimits_{i=1}^{N}(y(i) - y_m(i))^2},$$

where y is the target value of displacement, y_m is the output of the obtained model, N is the number of data.

The number of hidden units greatly influences the performance of an RBF neural network. If the number is too low, the precision of the network will be deteriorated. On the other hand, if the network employs too many hidden units, it will trend to overfit the data and increases 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 testing error is selected as optimum number.

In order to stand out the advantages of the proposed GA-RBF algorithm, the conventional K-Means (KM-RBF) training algorithm is also used for comparison.

In the 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 KM-RBF algorithm and GA-RBF algorithm are both employed to determine the number of hidden units. Empirically, the minimum number of hidden units is 6, 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 algorithm runs 10 times and the average precision values of the 10 runs are used to measure the performance of the RBF neural networks.

Fig. 4 shows the results obtained for the RBF neural networks with different number of hidden units for both KM-RBF algorithm and GA-RBF algorithm. The training errors of neural networks are illustrated in Fig.5 (a), and the testing errors of neural networks are showed in Fig.5 (b). Obviously, for KM-RBF algorithm, the neural network with 34 hidden units yields the minimum amount of testing error (0.0466), and an over-training was caused for the testing data when the number of hidden units more than 34. It is also seen that, for GA-RBF algorithm, the testing errors continue reduce with increased number of hidden units, however, the testing error performance 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 KM-RBF algorithm and GA-RBF algorithm, the best number of hidden units of the RBF neural networks is chosen as 34 eventually.

Fig.5 shows the evolution of the RMSE on both training data and testing data. In 288 generation, the minimum RMSE on testing data is obtained (0.466) corresponding to the RMSE of 0.0413 on training data.

Fig.6 (a) shows the outputs of the obtained RBF neural network with 34 hidden nodes by the proposed GA_RBF algorithm as compared to the target outputs for the training data, and Fig.6 (b) shows the target outputs and the outputs of the obtained RBF neural network for the testing data. It can be seen that the predicted outputs of GA-RBFNN follow reasonably close to the target outputs for both training data and testing data.

V. CONCLUSIONS

In this paper, we present a hybrid learning algorithm, named GA-RBF, to construct accurate radial basis function neural network for the ADCES of a certain mine-sweeping weapon. The simulation results and comparisons with other algorithm demonstrate its effectiveness and validity.

The next step of our work will be the design of high performance controller of the ADCES based on the obtained neural network.

REFERENCES

[1] K. Ziaei, N. Sepehri. Modeling and identification of electrohydraulic servos [J]. Mechatronics. 2000, 10, 761-772.

[2] W. Kemmetmuller, S. Muller, A. Kugi. Mathmatical modeling and nonlinear controller design for a novel electrohydraulic power-steering system [J]. IEEE Trans. on Mechatronics. 2007, 12(1): 85-97.

[3] S. He, N. Sepehri. Modeling and prediction of hydraulic servo actuators with neural networks [A]. Proceeding of American Control Conference, 1999, San Diego, 3708-3712.

[4] Y. Kang, M. H.Chu. An adaptive control using multiple neural networks for the position control in hydraulic servo system. Lecture Note on Computer Science. 2005, 3661: 296-305.

[5] Dang Kai-fang, Zhao Hong, Lin Ting-qi. Supervised control using fuzzy radial basis function networks for an electrohydraulic servo system [J]. Journal of System Simulation. 2004, 16(3), 593-596.

[6] M Jelali, A Kroll. Hydraulic servo-systems: modeling, identification and control. [M]. Springer, London, 2003.

[7] C. Harpham, C. W. Dawson, M. R. Brown. A review of genetic algorithms applied to training radial basis function networks [J]. Neural Computing and Applications. 2004, 13(3): 193-201.

[8] B. A. Whitehead. Genetic evolution of radial basis function coverage using orthogonal niches [J]. Neural Networks.1996, 7(6):869-880.

[9] E. G. M. de Lacerda, A. C. P. F. de Carvalho. Evolutionary optimization of RBF networks [A]. Proceedings of Sixth Brazilian Symposium on Neural Networks[C]. Los Alamitos, 2000: 219-224.

[10] S. Mishra, P. K. Dash, P. K. Hota, M. Tripthy. Genetically Optimized Neuro-Fuzzy IPFC for Damping Modal Oscillations of Power System [J].IEEE Trans. on Power Systems, 2002, 17(4): 1140-1147.

[11] Zhong Lu, Rao Wen-bi, Zou Cheng-ming. Artificial neural networks and its application [M]. Beijing. Science Press.2007.

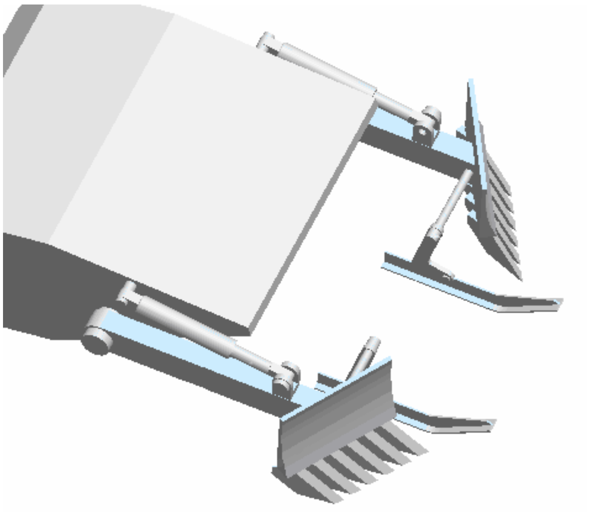


Fig.1. The automatic depth electrohydraulic control system

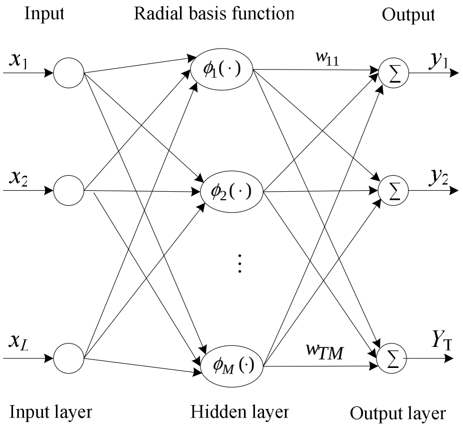
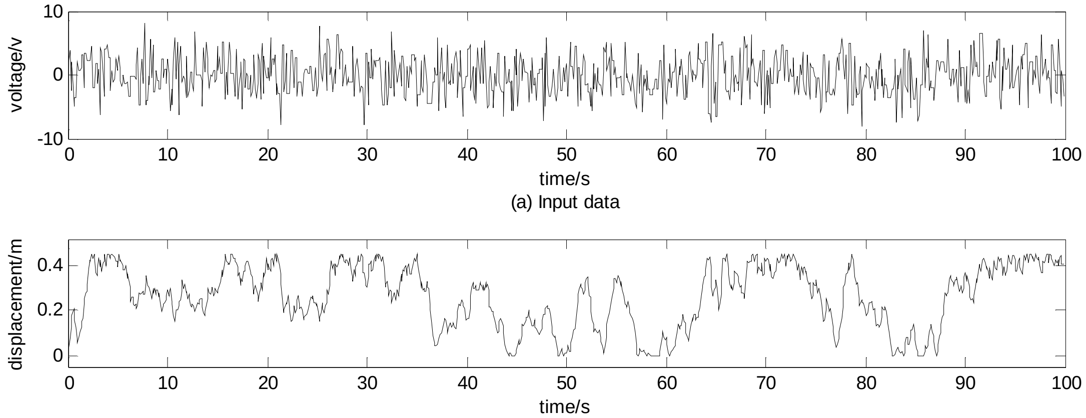


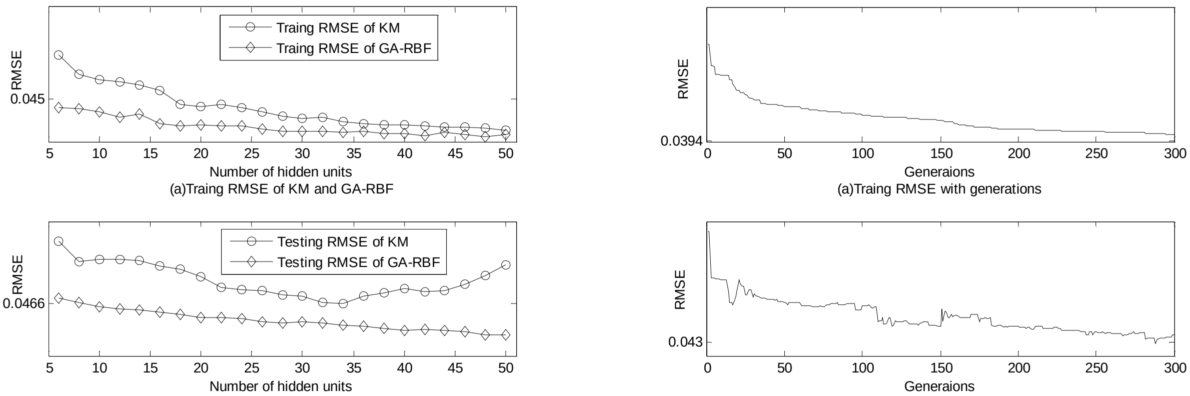
Fig.2. Radial basis function neural network



(a) Input data

(b) Output data

Fig.3. Input-output data of the ADECS



(a)Traing RMSE of KM and GA-RBF

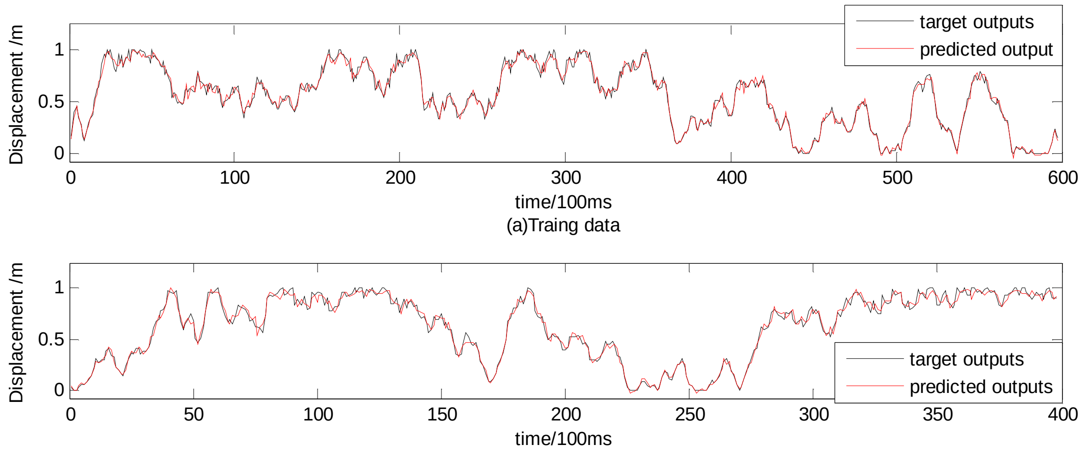
(b)Testing RMSE of KM and GA-RBF

(a)Traing RMSE with generations

(a)Testing RMSE with generations

Fig.4. Determination number of hidden units

Fig.5. RMSE with different generations



(a)Traing data

(a)Testing data

Fig.6. Comparison of target outputs and predicted outputs of RBFNN