Prediction of Ship Roll Motion based on Optimized Chaotic Diagonal Recurrent Neural Networks

Li Zhanying, Xing Jun, Li Bo and Wang Jue

School of Information Science and Engineering
Dalian PolytechnicUniversity
Da Lian 116034, China
l_zy1979@126.com

Abstract

Chaotic diagonal recurrent neural network is optimized and proposed to predict ship rolling motion. Aiming at the traditional arithmetic deductions of each weight value mediate do not give the specific sample time, it makes some lacks in this kind of deduction, the paper tries to optimize sampling time, carried out the derivation of the weight training and a convergence theorem of each weight learn algorithm based on Lyapunov function is given and proofed. Simulation results demonstrate the use of optimal sampling times increases the accuracy of all of the weight and algorithm convergence, to advance predicted precision so that forecast time of ten seconds efficiently. obviously better than using feed-forward BP neural network to predict.

Keywords: *chaotic diagonal recurrent neural network, ship rolling motion, momentum gradient learning algorithm, feed-forward BP neural network*

1. Introduction

A ship has six degrees of free movements in waves: rolling, pitching, yawing, swaying, surging, and heaving. Among them, rolling has the biggest influence on ship's movements, if the rolling angle under the function of wind, wave and flows can be predicted beforehand, which will be very important. The extreme short-term of prediction technique for ship rolling has much mean to improve the system quality of ship motion control, adaption of arm equipment system, dynamic location and special work on sea. However it is very difficult to forecast and analyze the time series due to the serious effect of non-linear factor to ship motion, such as wind and wave. Now we have made some achievements in this field. There are some methods used for the research. The way of time series (such as AR [1], ARMA[2]etc.) takes the time phase of ship rolling motion as smoothly random process, and ignore the effect of non-linear factor to ship motion. The chaotic identity has been confirmed to exist in ship motion in some [3-5], so that it cannot be satisfied to the requirement of prediction with traditional way of linear analysis. The neural network that regarded as one model of data drive can predict the system with highly complicated, non-linear and unstable. It is also great advantage for prediction of time series. But existed methods are mainly based on feed-forward network to progress prediction [6], it is one of static network and ignore the time relationship of time series. The each point in time series has correlation with time, so it cannot obtain the accurate results by means of feed-forward network. The recurrent network is one of feedback network, it has ability to mapping dynamic character through storage internal status. It makes the system to be able to adapt time variable character by responding the dynamic character of system more directly and vividly, therefore make the model of time series more intact and comprehensive. So the recursive way is one of substitution for feed-forward network in terms of time series prediction. Among kinds of recurrent neural

ISSN: 1975-0080 IJMUE

network, the diagonal recurrent is simple one. It is put forward by Chao-chee Ku and Kwang Y Lee [7] and be paid attention by more scholars with its simple structure and easy to realize. Then it is put into use with improvement [8-13], as in [13] mentioned the chaotic diagonal recurrent neural network, it is combined with neural network and chaotic character to comply with non-linear and chaotic character during the vessel navigation. But poor convergence does exist in mathematical deduction due to not specify the mediate weight values for sample time k. This article is based on the original mathematics to deduce kinds of weight values learning arithmetic of chaotic diagonal recurrent neural network and give corrected method for calculation weight values. The simulating results show that the systemic convergence had been improved, meanwhile it can both increase the precision of prediction and extend time of prediction effectively.

2. Chaotic Diagonal Recursive Neural Network

On the basis of dynamic recurrent neural network mentioned in [13], one kind of chaotic neural network which can be learned and approached to dynamic non-linear has been proposed to fit non-linear mapping's function. As Figure 1 is shown.

The hidden layer neuron is one kind of dynamic neuron with time postponed feedback. These dynamic neurons make the whole network to be possessed dynamic character. The hidden layer includes two layers, the first layer consists of two groups of neurons: one is that receive forward input and feedback input (F neuron), another is that receive feedback input only (B neuron); the output of second layer neuron is real hidden one that we need, called H neuron. The feedback input received by F neuron is the previous output of H neuron, however, the feedback input received by B neuron is the previous output of B neuron. Each of F neuron is connected with all input neurons, and every B neuron connects with itself only. The output of each F neuron, together with one B neuron acts as input of H neuron.

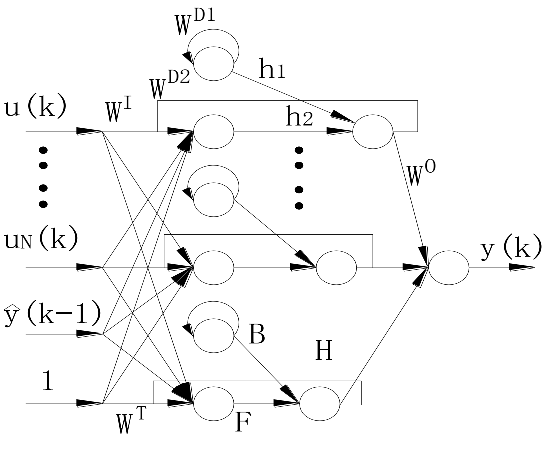


Figure 1. Chaotic Diagonal Recurrent Neural Networks Structure

For each discrete time k, $u_i(k)$ is the ith input, $y(k)$ is the output of the network.

$X=[u_1(k),u_2(k),\cdots\hat{y}(k-1)]$ is the input vector of the network, w_{ij}^I represents the input weights from the ith input neural neurons to the jth hidden F neural neurons, w_j^T is bias value of the jth F neurons, h_j^1 and h_j^2 represents the output of the jth B and F neural neurons. $w_j^{D_1}$ and $w_j^{D_2}$,represents the weights from the jth B and F neural neurons. h_j represents the output of the jth H neural neurons. w_j^o represents the weights from the jth H neural neurons and output neural neurons.

Then the relationship between input and output across the network can be described by:

$$h_i^1(k) = w_i^{D_1} f(h_i^1(k\text{-}1)) \tag{1}$$

$$\phi(x) = x(M-x)/M \tag{2}$$

$$h_i^2(k) = \sum_{j=1}^{m+1} w_{ij}^I x_j(k) + w_i^T + w_i^{D_2} h_i\ (k-1)) \tag{3}$$

$$h_i\ (k) = \rho(h_i^1(k) + h_i^2(k)) = \rho(s_i(k)) \tag{4}$$

$$\hat{y}(k) = \sum_{i=1}^{N} w_i^O\ h_i\ (k) \tag{5}$$

$$\rho(x) = \frac{1-e^x}{1+e^x} \tag{6}$$

Where $\phi(x)$ is chaotic neuron mapping function, and it takes Logistic map. The active function $\rho(x)$ is the sigmoid function. Since B neuron activation function is expressed by the Logistic map in the equation (2), which can generate chaos, so this neural network is called chaotic diagonal recurrent neural network.

3. Optimization of Dynamic Back-propagation Algorithm for Chaotic Diagonal Recurrent Neural networks

Momentum gradient algorithm, also known as Dyna-mic Back-propagation(DBP) algorithm, α is momentum factor, which is introduced based on momentum gradient algorithm and used the results of the previous amendment to influence this amendment. Usually,the value of α is 0 to 1($0 < a < 1$). η is learn rate.

Let $y(k)$ and $\hat{y}(k)$ be the desired and actual responses output of this model, then an error function can be defined as:

$$E(k) = \frac{1}{2}e(k)^2 = \frac{1}{2}[y(k) - \hat{y}(k)]^2 \tag{7}$$

Where $e(k) = y(k) - \hat{y}(k)$ is error between the plant and the network responses.

The gradient of error in equation (7),with respect to an arbitrary weight vector W ,is represented by:

$$\frac{\partial E}{\partial W} = -e(k)\frac{\partial y(k)}{\partial W} = -e(k)\frac{\partial O(k)}{\partial W} \tag{8}$$

The traditional arithmetic deductions of each weight value mediate do not give the specific sample time k, it makes some lacks in this kind of deduction. This article re-deduce the weight value based on the sample time k, and take the weight value between input neuron and hidden neuron as one training example.

3.1. $w_{ij}^I(k)$ Adjustment Algorithm for Input Neural Neurons and Hidden Neural Neurons

At sampling time k, $w_{ij}^I(k)$ is input weight of the ith input neural neurons and the jth hidden F neural neurons.

$$w_{ij}^I(k+1) = w_{ij}^I(k) + \eta e(k)\frac{\partial \hat{y}(k)}{\partial w_{ij}^I(k)} + \alpha(\Delta w_{ij}^I(k) - \Delta w_{ij}^I(k-1)) \tag{9}$$

$$\frac{\partial \hat{y}(k)}{\partial w_{ij}^I} = w_i^o(k)\frac{\partial h_i(k-1)}{\partial w_{ij}^I} = w_i^o(k) g_{ij}(k-1) \tag{10}$$

$$g_{ij}(k)=\frac{\partial h_i(k)}{\partial w_{ij}^I}=\frac{\partial h_i(k)}{\partial s_i(k)}\cdot\frac{\partial s_i(k)}{\partial w_{ij}^I}=\rho^{'}(s_i(k))(x_j(k)+w_i^{D_2}(k)g_{ij}(k-1)) \tag{11}$$

Where： $i=1,2\cdots h, j=1,2\cdots m+1, g_{ij}(0)=0$

If adding sampling time k, then equation(10) is incorrect. So equation (11) is new described by:

$$g_{ij}(k)=\frac{\partial h_i(k)}{\partial w_{ij}^I(k)}=\frac{\partial h_i(k)}{\partial s_i(k)}\cdot\frac{\partial s_i(k)}{\partial w_{ij}^I(k)}=\rho^{'}(s_i(k))(x_j(k)+w_i^{D_2}(k)g_{ij}(k-1))$$

So the above equation (10) is be changed:

$$\frac{\partial \hat{y}(k)}{\partial w_{ij}^I(k)}=w_j^o(k)\frac{\partial h_j(k-1)}{\partial w_{ij}^I(k)}\neq w_j^o(k)g_{ij}(k-1) \tag{12}$$

Where

$$g_{ij}(k-1)=\frac{\partial h_i(k-1)}{\partial w_{ij}^I(k-1)} \tag{13}$$

Adding sampling time k, adjustment algorithm of optimized value of $w_i^{D_1}(k)$ refer to (14-16).

$$\frac{\partial \hat{y}(k)}{\partial w_i^{D_1}(k)}=w_i^o(k)\frac{\partial h_i(k)}{\partial w_i^{D_1}(k)}=w_i^o(k)y_i(k) \tag{14}$$

Where

$$y_i(k)=\frac{\partial h_i(k)}{\partial w_i^{D_1}(k)}=\frac{\partial h_i(k)}{\partial s_i(k)}\cdot\frac{\partial s_i(k)}{\partial w_i^{D_1}(k)}=\rho^{'}(s_i(k))(c_i(k)+w_i^{D_2}(k)y_i(k-1)) \tag{15}$$

$$c_i(k)=\frac{\partial h_i^1(k)}{\partial w_i^{D_1}(k)}=\phi(h_i^1(k-1))+w_i^{D_1}(k)c_i(k-1)(M-2h_i^1(k-1))/M \tag{16}$$

Where $i=1,2\cdots h, c_i(0)=0$

Adding sampling time k, adjustment algorithm of optimized value of $w_i^{D_2}(k)$ refer to (17-18).

$$\frac{\partial \hat{y}(k)}{\partial w_i^{D_2}(k)}=w_i^o(k)\frac{\partial h_i(k)}{\partial w_i^{D_2}(k)}=w_i^o(k)p_i(k) \tag{17}$$

$$p_i(k)=\frac{\partial h_i(k)}{\partial w_i^{D_2}(k)}=\frac{\partial h_i(k)}{\partial s_i(k)}\cdot\frac{\partial s_i(k)}{\partial w_i^{D_2}(k)}=\rho^{'}(s_i(k))(h_i^2(k-1)+w_i^{D_2}(k)p_i(k-1)) \tag{18}$$

Where $i=1,2\cdots h, p_i(0)=0$

Same as adjustment algorithm of optimized value of $w_i^T(k)$ refer to (19-20).

$$\frac{\partial \hat{y}(k)}{\partial w_i^T(k)}=w_i^o(k)\frac{\partial h_i(k)}{\partial w_i^T(k)}=w_i^o(k)r_i(k) \tag{19}$$

$$r_i(k)=\frac{\partial h_i(k)}{\partial w_i^T(k)}=\frac{\partial h_i(k)}{\partial s_i(k)}\cdot\frac{\partial s_i(k)}{\partial w_i^T(k)}=\rho^{'}(s_i(k))(1+w_i^{D_2}(k)r_i(k-1)) \tag{20}$$

Where $i=1,2\cdots h, r_i(0)=0$

Adjustment algorithm of optimized value of $w_i^o(k)$ refer to (21).

$$\frac{\partial \hat{y}(k)}{\partial w_i^O(k)}=h_i(k) \tag{21}$$

3.2. Convergence of Algorithm

Dynamic back-propagation (DBP) algorithm is most widely used in training of diagonal recurrent neural network, it is the improvement of BP algorithm, which considering to the back feed unit. Network parameters are usually adjusted by gradient descent.

The update parameters using dynamic back-propagation(DBP) algorithm calls for a proper choice of the learning rate η .For a small value of η the convergence is guaranteed but the speed is very slow; on the other hand if η is too big, the algorithm becomes unstable. This section develops a guideline in selecting the learning rate properly, which leads to adaptive learning rate.

The valuable learning rate adjustment range is given in [7] and [14], and using Lyapunov stability theorem to analysis the impact on the learning rate η and DBP algorithm convergence.

A discrete-type Lyapunov function can be given by (7).Thus, the change of the Lyapunov function due to the training process is obtained by:

$$\Delta \mathrm{E}(k)=\mathrm{E}(k+1)-\mathrm{E}(k)=\frac{1}{2}\left[e^2(k+1)-e^2(k)\right] \tag{22}$$

The universal learning rate adjustment algorithm based on Lyapunov function of all weights of have the following convergence theorem.

Theorem 1：Let η is learning rate,

$$0<\eta_{\mathrm{m}}<\frac{2}{g_{L\max}^2} \tag{23}$$

Where $g_{L\max}:=\max_k \| g\ (k)\|$

$$g\ (k)=\frac{\partial O(k)}{\partial W\ (k)} \tag{24}$$

$W\,(k)$ is a weight vector composed of all the weight values in CDRNN ,and $\|\cdot\|$ is the usual Euclidean norm in $|\,\mathrm{R}^n$.

Proof:

Let W^I represents the input weight matrix , W^{T} represent bias matrix ,W^O represent the output weight matrix. W^{D_1} and W^{D_2} is back-feed weight matrix. Assume there are p inputs in the input layer, the numbers of F neurons and B neurons are q in the hidden layer ,one neuron in the output layer.

Putting all weights into one vector as

$$W\ =\left[[W^I]^T[W^T]^T[W^{D_1}]^T[W^{D_2}]^T[W^O]^T\right]^T \tag{25}$$

In which

$$W^I = \left[[W_1^I]^T[W_2^I]^T[W_3^I]^T \cdots [W_p^I]^T\right]^T \tag{26}$$

$$W^T = [W_1^T]^T \tag{27}$$

$$W^{D_1} = \left[[W_1^{D_1}]^T[W_2^{D_1}]^T \cdots [W_q^{D_1}]^T\right]^T \tag{28}$$

$$W^{D_2} = \left[[W_1^{D_2}]^T[W_2^{D_2}]^T \cdots [W_q^{D_2}]^T\right]^T \tag{29}$$

$$W_I^O = [W_1^O]^T \tag{30}$$

Let

$$\eta = \begin{bmatrix} \eta^I & & & & \\ & \eta^T & & & \\ & & \eta^{D_1} & & \\ & & & \eta^{D_2} & \\ & & & & \eta^O \end{bmatrix} \tag{31}$$

Where η^I , η^T , η^{D_1} , η^{D_2} , η^O represent the learning rate matrix corresponding to W^I , W^T , W^{D_1} , W^{D_2} ,and W^O , thus

$$\begin{aligned}\Delta W &= -\eta \frac{\partial E(k)}{\partial W(k)} = -\eta \frac{\partial e(k)}{\partial W(k)} e(k) \\ &= -e(k) \cdot \begin{bmatrix} \eta^I & & & & \\ & \eta^T & & & \\ & & \eta^{D_1} & & \\ & & & \eta^{D_2} & \\ & & & & \eta^O \end{bmatrix} \cdot \left[[\frac{\partial e(k)}{\partial W^I(k)}]^T [\frac{\partial e(k)}{\partial W^T(k)}]^T [\frac{\partial e(k)}{\partial W^{D_1}(k)}]^T \cdot [\frac{\partial e(k)}{\partial W^{D_2}(k)}]^T [\frac{\partial e(k)}{\partial W^O(k)}]^T\right]\end{aligned} \tag{32}$$

$$\Delta e(k) = [\frac{\partial e(k)}{\partial W(k)}]^T \Delta W = -e(k) \cdot [\eta_1 \left\| \frac{\partial e(k)}{\partial W^I(k)} \right\|^2 + \eta_2 \left\| \frac{\partial e(k)}{\partial W^T(k)} \right\|^2 + \eta_3 \left\| \frac{\partial e(k)}{\partial W^{D_1}(k)} \right\|^2 + \eta_4 \left\| \frac{\partial e(k)}{\partial W^{D_2}(k)} \right\|^2 + \eta_5 \left\| \frac{\partial e(k)}{\partial W^O(k)} \right\|^2] \tag{33}$$

$$\Delta E(k) = \Delta e_m(k)\left[e_m(k) + \frac{1}{2}\Delta e_m(k)\right] = \left[\frac{\partial e_m(k)}{\partial W\ (k)}\right]^T \eta\ e_m(k) \frac{\partial O(k)}{\partial W\ (k)} \cdot \left[e_m(k) + \frac{1}{2}\left[\frac{\partial e_m(k)}{\partial W\ (k)}\right]^T \eta\ e_m(k) \frac{\partial O(k)}{\partial W\ (k)}\right] \tag{34}$$

$$\frac{\partial e_m(k)}{\partial W\ (k)} = -\frac{\partial O(k)}{\partial W\ (k)} \tag{35}$$

$$\Delta E(k) = -\eta\ e_m^2(k) \cdot \left\| \frac{\partial O(k)}{\partial W\ (k)} \right\|^2 + \frac{1}{2}\eta^2 e_m^2(k) \cdot \left\| \frac{\partial O(k)}{\partial W\ (k)} \right\| = -\lambda\ e_m^2(k)^4 \tag{36}$$

So

$$\lambda = \frac{1}{2}\|g\ (k)\|^2 \eta\ (2 - \eta\ \|g\ (k)\|^2) \tag{37}$$

Where

$$\eta\ \|g\ (k)\|^2 = [\eta_1 \left\| \frac{\partial e(k)}{\partial W^I(k)} \right\|^2 + \eta_2 \left\| \frac{\partial e(k)}{\partial W^T(k)} \right\|^2 + \eta_3 \left\| \frac{\partial e(k)}{\partial W^{D_1}(k)} \right\|^2 + \eta_4 \left\| \frac{\partial e(k)}{\partial W^{D_2}(k)} \right\|^2 + \eta_5 \left\| \frac{\partial e(k)}{\partial W^O(k)} \right\|^2] \tag{38}$$

$\eta(i = 1, 2, ...5)$ is a positive constant,

$$\|g\ (k)\|^2 = \left\| \frac{\partial e(k)}{\partial W^I(k)} \right\|^2 + \left\| \frac{\partial e(k)}{\partial W^T(k)} \right\|^2 + \left\| \frac{\partial e(k)}{\partial W^{D_1}(k)} \right\|^2 + \left\| \frac{\partial e(k)}{\partial W^{D_2}(k)} \right\|^2 + \left\| \frac{\partial e(k)}{\partial W^O(k)} \right\|^2 \tag{39}$$

According to the Lyapunov stability theory, if convergence can be guaranteed, then $\Delta V(k) < 0$, thus $\lambda > 0$ that is $0 < \eta\ \|g\ (k)\|^2 < 2$,Let $g_{L\max} = \max_k \|g\ (k)\|$,So $0 < \eta\ g_{L\max}{}^2 < 2$.

Let $\eta_{\mathrm{m}} = \max_{i=1}^{5}\{\eta_i\}$,Then, the convergence of the algorithm is guaranteed if η_{m} is chosen

as $0 < \eta_m < \frac{2}{{g_{L\max}}^2}$.

4. Simulation Experiment

Below is one typical example of time series prediction of ship pitching motion, use traditional chaotic diagonal recursive neural network and optimized one separately to carry on the prediction of simulation. The data used for this prediction are from experimental data of one certain vessel navigating on the beam sea condition. There are 1000 data, sample step is 0.5s, 800 of them are used for learning and others are used for predicting the effection of test model and arithmetic.

Basis Procedures of Prediction:

1) Taking 90 deg of head sea as example, rebuilding the historical data of time series derived from the observed ship roll motion. It makes up of learning sample and lecturer value.
2) Recomposing theoretical calculating minimum embedding dimensions m and best delay time τ based on G-P phase space, and rebuild it.
3) Specify the structure of chaotic diagonal recursive neural network. Obtaining m through G-P arithmetic and select the number of nodes from input layer of neural network. The character τ decides the delay of input data among the nodes and construct predicted model of chaotic diagonal recursive neural network.
4) Network learning and training. Selecting part of training data from original data to input and carry on the network training until the training complies with requirement. Record network parameters at the moment, if cannot meet with training target, then go back to 3).
5) Model prediction. Select tested sample to input and get the first predicted value. Add the actual value of first point to the original input unit to process the second tested point prediction. On the analogy of this, generating the predict results step y step.

Use G-P arithmetic to calculate values of m and τ . This article takes $m = 6$ and $\tau = 2$, network's structures are given below: two input neurons, present measured rolling time series value and output rolling predicted value from network previous time; ten FB neurons, ten B neurons, ten H neurons and one output neuron are taken as network's rolling forecasting values.

5. Results and Analysis

The real line is actual output from system and blue dotted line is predicted output from every model. Analysis predictions of traditional chaotic diagonal recurrent neural network (CDRNN) are shown in Figure 2 and 3. The optimized chaotic diagonal recurrent neural network (OCDRNN) is shown in Figure 4 and 5. In order to compare with results of feed-forward neural network model, this article also provides one way used BP network to predict (as shown in Figure 6 and 7). To make the predicted results of each model more distinct, 35 time samples are taken out and the predicted results can be reflected on Figure 8. The error curve predicted among methods above is shown in Figure 9. The black line is predicted error from optimized chaotic diagonal recurrent neural network, blue line is predicted error from traditional chaotic diagonal recurrent neural network and green line is predicted error from BP neural network.

We can make a conclusion from Figure 9 that the effect used feed-back neural network to predict ship rolling is obviously better than using feed-forward neural network to predict. Moreover the systemic convergence has been improved and predicted accuracy has been increased using optimized chaotic diagonal recurrent neural network to forecast. The effective prediction time gets to ten seconds around.

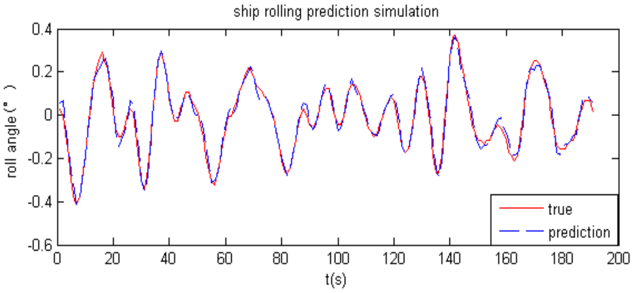


Figure 2. Traditional Chaotic Diagonal Recurrent Neural Network Prediction

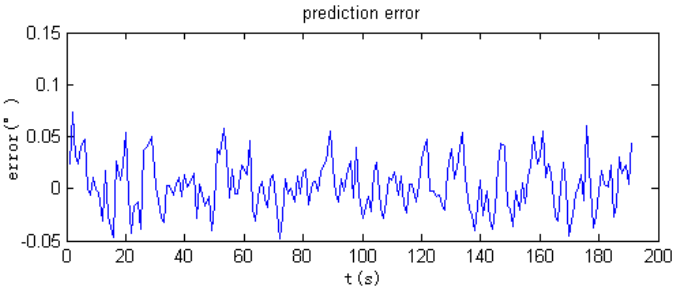


Figure 3. Traditional Chaotic Diagonal Recurrent Neural Network Prediction Error

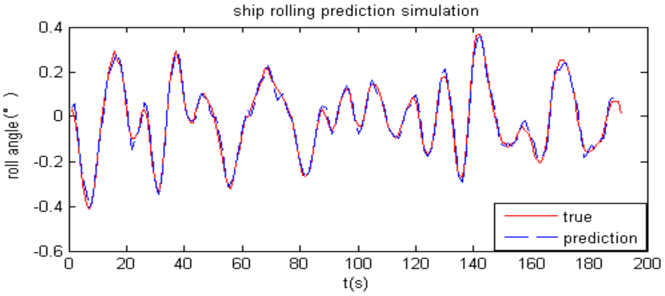


Figure 4. Optimized Chaotic Diagonal Recurrent Neural Network Prediction

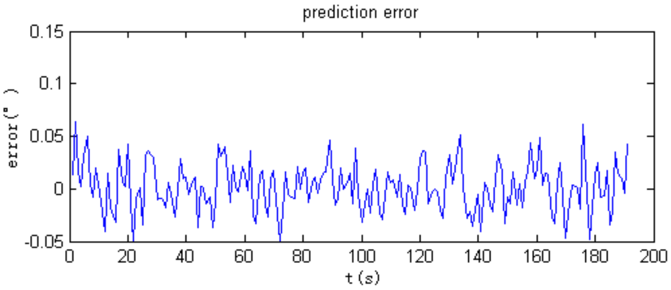


Figure 5. Optimized Chaotic Diagonal Recurrent Neural Network Prediction Error

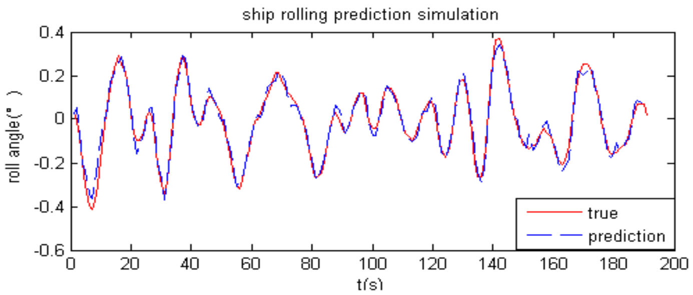


Figure 6. BP Neural Network Prediction

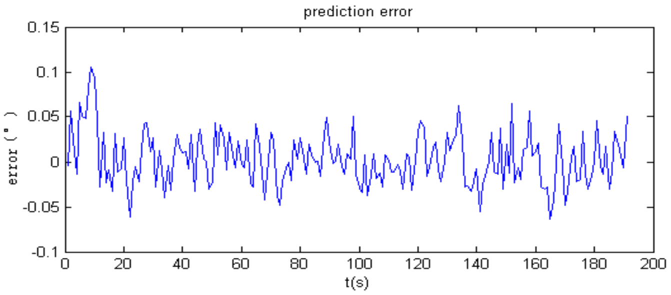


Figure 7. BP Neural Network Prediction Error

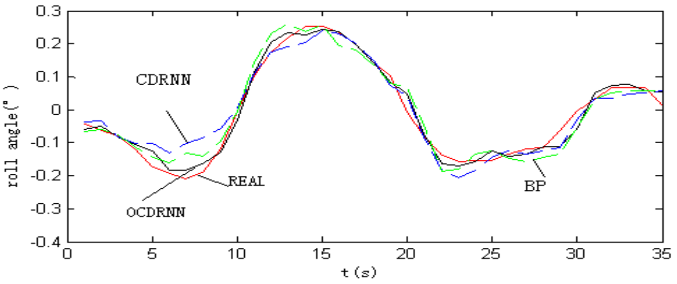


Figure 8. Ship Rolling Prediction

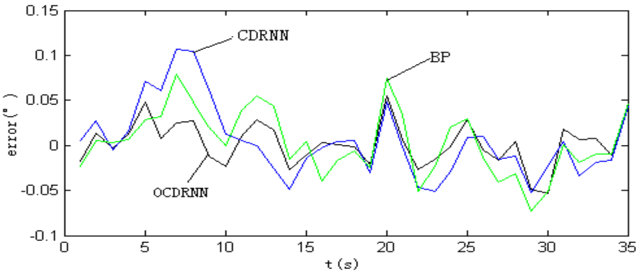


Figure 9. Prediction Error Curve

So as to analyze the results of these three models quantitatively, this article also gives effect's comparisons of them from MSE and MAE separately.

Where:

$$MSE = \sqrt{\frac{\sum_{i=1}^{T}(y_{id} - y_i)^2}{T}} \tag{40}$$

$$MAE = \frac{1}{T}\sum_{i=1}^{T}|(y_{id} - y_i)| \tag{41}$$

The comparison from MSE and MAE is shown in Table 1.

Table 1 shows that the chaotic diagonal recurrent network was better prediction effection than BP feed forward network, and the prediction time is extended. The optimized network is better prediction accuracy than the traditional network, less time used, and predicted the time spent far less than the time spent of the BP network.

Table 1. Comparison of MSE and MAE

Model	Error and Time		
	MSE	MAE	t^a
Traditional Chaotic Diagonal Recurrent Neural Network	0.0292	0.0108	10.4574s
Optimized Chaotic Diagonal Recurrent Neural Network	0.0218	0.0020	8.2810s
BP Neural Network	0.0422	0.0042	134.4187s

[a.]t means the elapsed time from network start to training to predict end in the same net environment.

This article brings up one new method to predict time series for ship rolling motion. Make a combination with chaotic theory and diagonal recurrent neural network, using phase space restructured technique to set up historical data of ship rolling time series and optimize traditional chaotic diagonal recurrent neural network. The simulating results have been confirmed that each of weight value's precision is increased by optimized sample time k. The systemic convergence has been improved through adding sample time k and revise feed-back input including previous time in network. Utilize dynamic information of system adequately to advance predicted precision so that forecast the rolling time of ten seconds around efficiently. Provide one new way for ship prediction. The means of article is also available to ship motion in other degrees of freedom.

References

[1] J. Ma, T. Li and G. Li, "Comparison of Representative Method for Time Series Prediction", Proceedings of the 2006 IEEE International Conference on Mechatronics and Automation, Luoyang, China, **(2006)** June, pp. 2448-2453.

[2] I. R. Yumori, "Real Time Prediction of Ship Response to Ocean Waves using Time Series Analysis", OCEANS.81, **(1981)** September, pp. 1082-1089.

[3] A H Nayfeh and A A. Khdeir, "Nonlinear rolling of ships in regular beam seas", International Shipbuilding Progress, vol. 33, no. 379, **(1986)**, pp. 40-49.

[4] O. Ruquan and J. Zhu, "Nonlinear Oscillations and Chaos of Ship Rolling Motion", Shipbuilding of China, **(1999)**, pp. 21-28.

[5] F. Cai, A. Shi and L. Wan, "Chaotic Dynamics Analysis of Ship Swaying Motion", Journal of Dalian Naval Academy, vol. 29, no. 5, **(2006),** pp. 36–39.

[6] J. Hou, D. Fang, F. Cai, "Extrem e short term prediction of ship swaying motions based on com bination of chaos and neural network", Ship Science and Technology, vo1. 30, no. 1, **(2008)**, 2, pp. 67-70.

[7] C.-C. Ku and K. Lee, "Diagonal Recurrent Neural Networks for Dynamic Systems Control", IEEE Transactions on Neural Networks, vol. 6, no. 1, **(1993)** January, pp. 144-156.

[8] G. Barbounis Thanasis, B. Theocharis John, C. Alexiadis Minas and S. Dokopoulos Petros, "Long-term wind speed and power forecasting using local recurrent neural network models", IEEE Trans. Energy Conversion, vol. 21, no. 1, **(2006),** pp. 273-284.

[9] A. Kazemy, S. A. Hosseini and M. Farrokhi, "Second Order Diagonal Recurrent Neural Network, pp. 251-256.

[10] K. Wang and G. Li, "Time Series Prediction of Ship Roll Using Diagonal Recurrent Neural Network",Journal of Harbin Engineering University, vol.1 8, no. 1, **(1997)** 2, pp. 39-45.

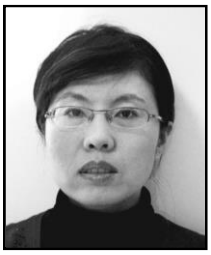
[11] X. Zhang, X. Peng and X. Zhao, "Diagonal Recurrent Neural Network Algorithm for Extreme Short Prediction of Ship Motion", Journal of System Simulation, China, **(2002)** 5, pp. 641-642.

[12] W. Yi and G. Wei, "Chaotic Neural Network Approach to Chaotic Time Series Prediction", Journal of Electronic Measurement and Instrument, vol. 21, no. 6, **(2007)**. pp. 5-9.

[13] L. Zheng, "Study and Application on Chaotic Neural Network and Fuzzy Chaotic Neural Network", Harbin Engineering Universty, **(2002)**, pp. 61-64.

[14] X. Liang, "Comments On "Diagonal Recurrent Neural Networks For Dynamic Systems Control"—Reproof of Theorems 2 and 4, IEEE Transactions on Neural Networks, vol. 8, no. 3, **(1997)**, pp. 811-812.

Authors



Li Zhanying, she was born in Jilin, China, in 1979. She took her Ph.D. degree in Pattern Recognition and Intelligent Systems at the Harbin Engineering University, in 2012. She is also Associate Professor at the School of Information Science and Engineering, Dalian Polytechnic University. Dr. Li's areas of interests are intelligent control, neural networks and time series prediction.

Xing Jun, he was born in 1972. He is a Ph.D. and Associate Professor at the School of Information Science and Engineering, Dalian Polytechnic University since 2003. His main research interests include information system, the internet of things and intelligent algorithm.

Li Bo, he was born in 1986. He took his Ph.D. degree in signal and information processing at the Harbin Engineering University in 2013. He is a lecturer at the college of Information Science and Engineering, Dalian Polytechnic University. His main research interest is intelligent information processing.

Wang Jue, she was born in 1986. She took her PhD degree in signal and information processing at the Harbin Engineering University in 2013. She is currently a lecturer in the college of Information Science and Engineering from Dalian Polytechnic University. Her main research interest is intelligent information processing.