한국항해항만학회 제30권 제1호 춘계학술대회논문집 pp. 67~74, 2006. 6

A Study on Automatic Berthing Control of Ship Using Adaptive Neural Network Controller

Phung-Hung NGUYEN and Yun-Chul JUNG***

* *Graduate school, Korea Maritime University, Pusan 606-791, Korea*

** *Professor, Div. of Navigation Systems Engineering, Korea Maritime University, Pusan 606-791, Korea*

Abstract : *In this paper, an adaptive neural network controller and its application to automatic berthing control of ship is presented. The neural network controller is trained online using adaptive interaction technique without any teaching data and off-line training phase. Firstly, the neural networks used to control rudder and propeller during automatic berthing process are presented. Finally, computer simulations of automatic ship berthing are carried out to verify the proposed controller with and without the influence of wind disturbance and measurement noise.*

Key words : *Adaptive neural networks, Adaptive interaction, Autopilot, Ship control, Automatic berthing.*

1. Introduction

Since last two decades, studies on automatic ship berthing have been carried out by many researchers. This topic of study is one of the difficult problems in ship control fields (Im and Hasegawa, 2002). Neural networks (hereinafter called NN) have proved to be an effective and attractive option in developing automatic ship berthing controllers.

The potential of NN for control has received much attention and rapidly grown in the 1990s, because of the ability of NN in solving some awkward control problems where the high non-linearity of the controlled plant and unpredictable external disturbances make the plant's behaviors hard to control. In addition, the fast calculation in NN is also suitable for real time control applications.

In Zhang *et al.*(1997), a multi-variable controller for automatic ship berthing using multi-layer feedforward NN was introduced. This controller used backpropagation training method to adapt the NN weights with an on-line training scheme. The effectiveness and robustness of the NNC were shown by computer simulations in ideal environmental condition and under the influence of noise and wind.

Later, Im and Hasegawa(2001) introduced a parallel NNC for automatic ship berthing which has separated hidden layers that output the engine and rudder respectively, and the improvements were shown through various computer simulations. Then they presented a motion identification method using NN and its application to automatic ship berthing (Im and Hasegawa, 2002). In their study, motion identification was used to estimate the effect of environmental disturbances. Off-line training scheme using backpropagation method was also applied with teaching data consisting of 6 sets of automatic berthing simulation.

Recently, in Nguyen and Jung(2005, 2006a,b), we have proposed direct adaptive NNCs for course-keeping and track-keeping control of ship based on the adaptation algorithm developed by Brandt and Lin(1999) and the extension of the NNC proposed in Saikalis and Lin(2001).

Employing the advantages of the NNC developed in our previous studies, in this paper an adaptive NNC and its application to automatic berthing control of ship is presented. The proposed NNC can be trained online using adaptive interaction technique without any teaching data and off-line training phase. The backpropagation network is not required in this kind of NNC so the configuration is simplified and the speed of training is considerably improved (Nguyen and Jung, 2005).

Firstly in this paper, the adaptive NN by adaptive interaction (hereinafter called ANNAI) used to control rudder and propeller during automatic berthing process are presented. Then a berthing guidance algorithm is proposed. To test the proposed controller, computer simulations of automatic ship berthing are carried out with and without the influence of wind and measurement noise. Finally, the discussion and conclusion are presented.

* hungdktb@hotmail.com, 051)410-4286

** ycjung@hhu.ac.kr, 051)410-4286

2. Automatic Berthing Control System

In this chapter an automatic berthing control system using ANNAI controllers and a berthing guidance algorithm will be presented. Our goal is to maneuver the ship automatically to a desired point near planned berth and stop the ship there with almost zero final speed and desired heading. We only focus on designing and validating the NNC, so within the limited extent of this paper the use of side thrusters or tugs is not considered. Therefore the control problem is to control of an underactuated ship where rudder and propeller are used to control the ship in 3 DOF. The configuration of proposed automatic berthing control system is shown in Fig. 1.

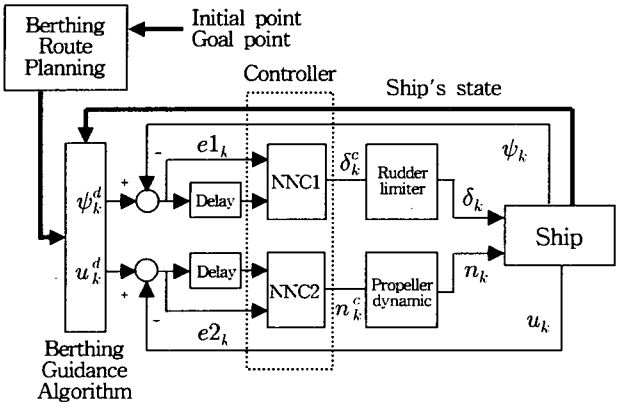

Fig. 1 Configuration of automatic berthing control system

The controller consists of NNC1 and NNC2 which control rudder and engine respectively. These are ANNAI which are similar to the multi-layer feedforward NN with one hidden layer developed in Nguyen and Jung (2005). Ship actual heading and speed at time step k are ψ_k and u_k; corresponding desired heading and speed are ψ_k^d and u_k^d. The heading error and speed error are respectively defined as

$$e1_k = \psi_k^d - \psi_k \tag{1}$$

$$e2_k = u_k^d - u_k \tag{2}$$

These error signals and their time delays are inputs of NNC1 and NNC2 (Fig. 1). The output of NNC1 is command rudder angle (δ_k^c) whereas that of NNC2 is command engine revolution (n_k^c) at time step k. The actual rudder angle and engine revolution acted on ship are δ_k and n_k respectively.

2.1 Control of Ship's Heading

The configuration of the NNC1 is shown in Fig. 2 where $\omega 1_{ij}$ is used to indicate the weights between output layer and hidden layer, and $w1_{jp}$ is used to indicate the weights between hidden layer and input layer. The subscripts p, i and j indicate the number of neurons in input, output and hidden layer respectively. The input signals of the NNC1 are merely heading error and its time-delayed values.

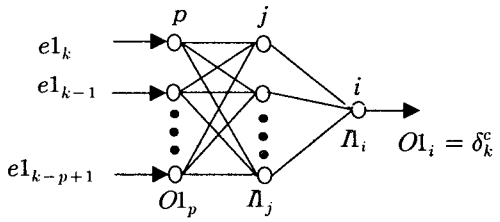

Fig. 2 NNC1 configuration

The cost function used for the NNC1 is similar to that of Zhang *et al.*(1997) and Nguyen and Jung (2005), and is written in (3)

$$E1_k = \frac{1}{2}[\rho_1(\psi_k^d - \psi_k)^2 + \lambda_1\delta_k^2 + \sigma_1 r_k^2] \tag{3}$$

where ρ_1, λ_1 and σ_1 are positive penalty constants; r_k is yaw rate.

Similar to the ANNAI proposed in Nguyen and Jung (2005), the adaptation laws for the hidden layer weights and output layer weights of the NNC1 are as follows respectively

$$\dot{w}1_{jp} = O1_p[\phi 1_j \sigma(-\Pi_j) + \gamma_1.0] = O1_p \phi 1_j \sigma(-\Pi_j) \tag{4}$$

$$\begin{aligned}\dot{w}1_{ij} &= \gamma_1.\sigma(\Pi_j).(\rho_1 e1_k + \lambda_1\delta_k + \sigma_1 r_k) \\ &= \gamma_1.O1_j.(\rho_1 e1_k + \lambda_1\delta_k + \sigma_1 r_k)\end{aligned} \tag{5}$$

where,

$O1_p$: is the set of p inputs to the NNC1 consisting of current heading error $e1_k$ and its delayed signals at time steps *k-1, k-2,···, k-p+1,*

$O1_j$: is the output of neurons in the hidden layer,

$$O1_j = \sigma(\Pi_j) = \frac{1}{1 + exp(-\Pi_j)}) \tag{6}$$

Π_j : is the summation of the weighted inputs to the units in the hidden layer plus threshold value $\theta 1_j$ of the hidden layer neurons,

$$\Pi_j = \sum_p (w_{jp} O1_p) + \theta 1_j \tag{7}$$

γ_1 : is the learning rate, and

$$\phi 1_j = w1_{ij}.\dot{w}1_{ij} \tag{8}$$

Using the adaptation laws (4) and (5) NNC1 can make the ship heading ψ_k track the desired value ψ_k^d generated by the berthing guidance algorithm which will be discussed later in subchapter 2.3.

2.2 Control of Ship's Speed

The configuration of the NNC2 is similar to that of NNC1 and shown in Fig. 3, but $e1_k$ and its time delayed signals are replaced by $e1_k$ and its delayed signals at time steps *k-1, k-2,···, k-p+1*. The task of NNC2 is to infer proper engine revolution command to minimize the following cost function

$$E2_k = \frac{1}{2}\left[\rho_2 (u_k^d - u_k)^2 + \lambda_2 u_k^2 + \sigma_2 (u_k - u_{k-1})^2\right] \quad (9)$$

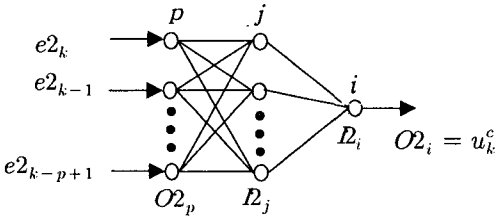

Fig. 3 NNC2 configuration

The adaptation law for hidden layer weights of NNC2 is in the form of equation (4)

$$\dot{w}2_{jp} = O2_p\left[\phi 2_j \sigma(-I2_j) + \gamma_2.0\right] = O2_p \phi 2_j \sigma(-I2_j) \quad (10)$$

In NNC2 the output neuron is tangent sigmoidal activation function type, where output signal is

$$O2_i = u_k^c = tansig(I2_i) = \frac{2}{1 + exp(-2.I2_i)} - 1 \quad (11)$$

Based on Nguyen and Jung (2005), the adaptation law for the output layer weight can be written as

$$\dot{w}2_{ij} = -\gamma_2 [\tan sig(I2_i)]' \sigma(I2_j) \frac{\partial E2_k}{\partial u_k} \quad (12)$$

Taking derivative of $tansir(I2_i)$, equation (12) can be expressed as

$$\dot{w}2_{ij} = -\gamma_2 O2_j \frac{\exp(-2.I2_i)}{(1 + \exp(-2.I2_i))^2} \cdot \frac{\partial E2_k}{\partial u_k} \quad (13)$$

Using the chain rule one can write

$$\frac{\partial E2_k}{\partial u_k} = \frac{\partial E2_k}{\partial u_k} - \frac{\partial E2_k}{\partial u_k} + \frac{\partial E2_k}{\partial \dot{u}_k} \frac{\partial \dot{u}_k}{\partial n_k} \quad (14)$$

in which,

$$\dot{u}_k = u_k - u_{k-1} \quad (15)$$

Note that $\dot{u}_k$ increases or decreases following the increase or decrease of engine revolution n_k. So $\partial \dot{u}_k / \partial n_k$ in the equation (14) can be replaced with $sign(\partial \dot{u}_k / \partial n_k) = 1$ to yield

$$\begin{aligned} \frac{\partial E2_k}{\partial u_k} &= \frac{\partial E2_k}{\partial u_k} - \frac{\partial E2_k}{\partial u_k} + \frac{\partial E2_k}{\partial \dot{u}_k} \\ &= -(\rho_2 e2_k + \lambda_2 n_k - \sigma_2 \dot{u}_k) \end{aligned} \quad (16)$$

Replacing (16) into (13) yields

$$\begin{aligned} w2_{ij} &= \gamma_2 . O2_j \frac{exp(-2.I2_i)}{(1 + exp(-2.I2_i))^2} \\ &\cdot (\rho_2 e2_k + \lambda_2 n_k - \sigma_2 \dot{u}_k) \end{aligned} \quad (17)$$

To summarize, the adaptation law for the hidden layer weights and output layer weights of NNC2 are described in equations (10) and (17) respectively. Using the adaptation laws (10) and (17) NNC1 can make the ship speed u_k track the desired value u_k^d generated by the berthing guidance algorithm.

2.3 Berthing Guidance Algorithm

In this paper, the automatic berthing control system to be designed is to use rudder and propeller to control the state of an unknown and non-linear ship. A predefined berthing route is a curve automatically generated using spline function for the given position and heading of ship at initial and goal points. Practically, to be able to track such a curved route ship's heading and tangent vector of the curved route at ship's position should make a proper drift angle (β) while ship moves along the route (Fig. 4).

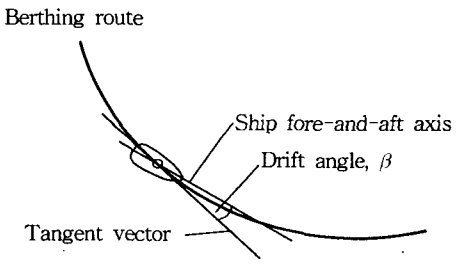

Fig. 4 Drift angle

The berthing guidance algorithm proposed here calculates ψ_k^d to ensure that ship can track the route and stop at goal point with desired heading.

1) Calculation of ψ_k^d

If the ship is on the desired berthing route (M in Fig.

5), the desired heading ψ_k^d is the direction from M to N, where y_{Nk} is determined by a step of $K.L$ forward from current y_k.

$$y_{Nk} = y_k + K.L \tag{18}$$

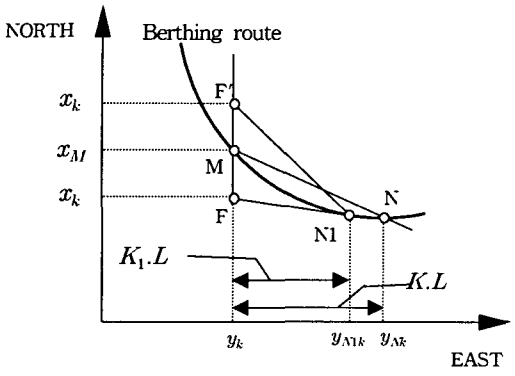

Fig. 5 Determination of desired heading

Equation (18) is based on the method in Zhang *et al.*(1997). Here, K is a constant and L is the ship length. This ψ_k^d ensures that ship moves with a certain drift angle β. However, the radius of the planned berthing route is not equal at every point on the route and β should be properly varied.

Now consider the situation where the ship is not on the desired route but at the point F or F' in Fig. 5. In this case, the new desired heading ψ_k^d is determined as the direction from F or F' to N1, with y_{N1k} is determined by a step of $K_1.L$ forward from current y_k $(0 < K_1 < K)$.

$$y_{N1k} = y_k + K_1.L \tag{19}$$

where K_1 can be obtained from

$$K_1 = (K_{\max} - K_{\min}).exp(-\xi\frac{di}{L}) + K_{\min} \tag{20}$$

in which, $K_{\max}$, $K_{\min}$ are maxima and minima of K; ξ is a positive constant, and di is length of FM or F'M, off-track distance on x axis.

$$di = x_k - x_M \tag{21}$$

From equation (20) we can see that K_1 varies from $K_{\max}$ to $K_{\min}$ according to di: K_1 becomes $K_{\max}$ when di equals 0; K_1 approaches $K_{\min}$ while di increases. Using this method to calculate ψ_k^d, ship can move back the desired route whenever it deviates.

2) Calculation of u_k^d

In practice, the heading of the ship is emphasised in the early stages of the berthing process. Only when the ship approaches the berth the velocity values do become more important (Zhang *et al*, 1997). Similar to this work, the desired speed u_k^d can be determined as

$$\text{IF } D/L > K_2, \text{ THEN } u_k^d = u_k \tag{22}$$

$$\text{IF } D/L \leq K_2, \text{ THEN } u_k^d = \frac{D}{K_2 L}u_k \tag{23}$$

where, D is the distance between the current ship position (y_k, x_k) and the goal point, L is the length of the ship, and K_2 is a constant given by the designer according to the stopping characteristics of the ship.

3. Simulation

3.1 Simulation Setup

The NNC1 and NNC2 are multilayer feedforward NN with one hidden layer. Each NNC consists of four input neurons, six hidden neurons and one output neuron. The input neurons have linear activation functions, the hidden neurons have sigmoidal activation function, and the output neurons have tangent sigmoidal activation function. In this study we apply fixed values of learning rate and number of training iteration in one control cycle. The parameters for NNC1 and NNC2 are selected as follows

$$[\rho_1, \lambda_1, \sigma_1, N_1, \gamma_1] = [1.5,\ 0.045,\ 0.2,\ 50,\ 1.5], \tag{24}$$

$$[\rho_2, \lambda_2, \sigma_2, N_2, \gamma_2] = [1.5,\ 0.0015,\ 0.2,\ 50,\ 2], \tag{25}$$

$$[K_{\max}, K_{\min}, K_2, \xi] = [0.3,\ 0.1,\ 0.4,\ 2]. \tag{26}$$

Here, N_1, N_2 are the number of iterations in one control cycle for NNC1, NNC2 (see Nguyen and Jung, 2005). The initial and final ship position and heading are (34.833^{0}N, 128.83^{0}E) and 150^0; (34.828^{0}N, 128.84^{0}E) and 90^0, respectively. The berth is assumed to be the South border of the chart with latitude of 34.82795^{0}N.

In this paper, the mathematical ship model is used for simulation and testing the performance of the controllers. The ship model used in this study is a nonlinear model of a container ship taken from GNC Toolbox of Fossen (2005) with length L = 175m and breadth B = 25.4m.

The effect of wind disturbance against the body of the ship is based on the work of Isherwood(1972) introduced in Fossen(2002). A random signal with a uniform distribution on [-0.1, +0.1] degree is used as the sensor noise in the heading sensor. The random noise in ship position is set with ratio of 0.1, and 0.01 in speed and yaw

rate measurement.

For visual simulation of ship's movement, M-Maps Toolbox, which is a set of mapping tools written in MATLAB and is available in Rich(2005), is used. The modules written in MATLAB introduced in Nguyen and Jung(2005b) for guidance and control using Mercator chart are applied here.

3.1 Simulation Results and Discussion

In this subchapter, the berthing control simulations with and without the effect of wind disturbance and measurement noise are presented. For each simulation only the plotting of ship position is shown here. Other recorded data is attached in Appendix.

1) Without wind and measurement noise

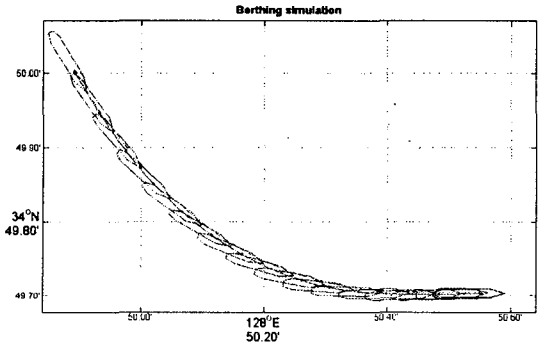

Fig. 5 Automatic berthing control without wind and noise

Fig. 5 shows the simulation results in case no wind and no measurement noise applied. The ship position on the berthing trajectory is plotted every 45 seconds. It is shown that the tracking target has been satisfactorily achieved. The maximum $d\dot{i}$ is about half of ship breadth B (Fig. 9).

1) With wind and measurement noise

The berthing simulations are undertaken under random offshore and onshore wind and measurement noise. The wind speed changes randomly every 5 seconds and assumes values between 10 knots and 20 knots. Firstly, to represent offshore wind disturbance, the wind direction varies between 90^0 and 270^0 every 30 seconds. Secondly, to represent onshore wind disturbance, the wind direction varies from 270^0 via 360^0 to 90^0 every 30 seconds. The simulations show that the offshore and onshore wind effect on the lateral speed of the ship and final x ordinate, but the robustness of the NNCs is maintained. The maximum value of $d\dot{i}$ is about 14 m. Only the simulation of onshore wind condition is shown (Fig. 6).

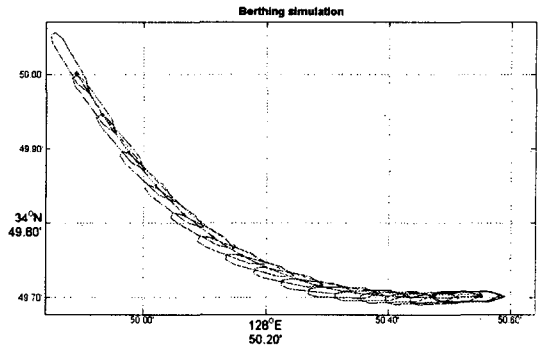

Fig. 6 Automatic berthing with onshore wind and noise, wind speed changes randomly from 10 knots to 20 knots.

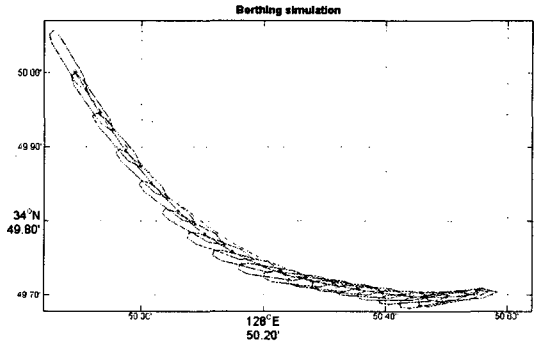

Fig. 7 Automatic berthing with onshore wind and noise, wind speed changes randomly from 15 knots to 25 knots.

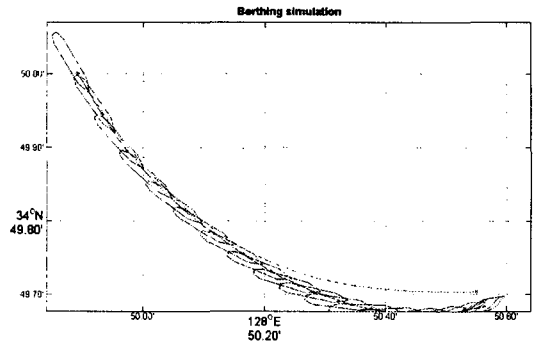

Fig. 8 Automatic berthing with onshore wind and noise, wind speed changes randomly from 20 knots to 30 knots.

In the next simulations, more difficult situations are selected. Only onshore wind simulations are shown. The wind speed changes randomly every 5 seconds and assumes values between 15 knots and 25 knots (Fig. 7); and then from 20 knots to 30 knots (Fig. 8). In Fig. 7, ship

is off the desired route with maximum di is about 24 m (approximately one ship breadth B). At final stage, ship is pushed toward the berth. The tracking target has also been reasonably maintained. However, care should be taken when selecting the final goal point.

In Fig. 8 where wind speed increases, the safe berthing is not maintained. This is dangerous situation for berthing without the use of side thrusters and tugs.

In all above simulations the fluctuation in the rudder movement was observed, and consequently lateral speed and yaw rate also fluctuated. This fluctuation may be reduced by carefully selecting the penalty constancies in the cost function.

4. Conclusion

In this paper, an automatic berthing control system for ship is developed. The ANNAI controller is applied to control the ship's rudder and engine revolution in order to automatically control the ship berthing. A useful berthing guidance algorithm is proposed. This algorithm can calculate desired heading and speed for the controllers. The obtained simulation results lead to the following conclusions:

- The NNC can be trained online without the necessity of any teaching data and offline training phase.
- The NNC can make both ship's heading and speed track the desired values.
- The proposed berthing guidance algorithm works effectively in berthing process.
- The unknown and non-linearity ship can be controlled satisfactorily, no prior knowledge of ship is required.
- The NNC is not so sensitive to measurement noise of input signals.
- The control system is robust under the light effect of wind disturbance.
- When the wind disturbance is considerable, the use of side thrusters and/or tugs is required.

However, more simulations should be undertaken in various external environmental conditions and for other types of ships to verify the proposed automatic berthing control system and to test its stableness and robustness. These works will be considered in the future.

References

[1] Brandt, R.D. and Lin, F. (1999), "Adaptive interaction and its application to neural networks", Elsevier, Information Science 121 (pp. 201-215).

[2] Fossen, T. I. (2002), "Marine Control Systems: Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles", Marine Cybernetics, Trondheim, Norway. ISBN 82-92356-00-2.

[3] Fossen, T.I. (2005), GNC Toolbox for MATLAB (http://www.cesos.ntnu.no/mss/MarineGNC/index.htm, accessed 2005/Dec).

[4] Im, Namkyun and Hasegawa, Kazuhiko (2001), "A Study on Automatic Ship Berthing using Parallel Neural Controller", J.Kansai Soc. N.A., Japan, No. 236, pp. 65~70.

[5] Im, Namkyun and Hasegawa, Kazuhiko (2002), "Motion Identification using Neural Networks and Its Application to Automatic Ship Berthing under Wind", Journal of Ship & Ocean Technology, Vol. 6, No. 1, pp. 16~26.

[6] Nguyen, P. H. and Jung, Y. C. (2005), "An Adaptive Autopilot for Course-keeping and Track-keeping Control of Ships using Adaptive Neural Network (Part I: Theoretical Study)", International Journal of Navigation and Port Research (KINPR), Vol.29, No.9 pp. 771~776, ISSN-1598-5725.

[7] Nguyen, P. H. and Jung, Y. C. (2006a), "An Adaptive Autopilot for Course-keeping and Track-keeping Control of Ships using Adaptive Neural Network (Part II: Simulation Study)", International Journal of Navigation and Port Research (KINPR), Vol.30, No.2 pp. 119~124, ISSN-1598-5725.

[8] Nguyen, P. H. and Jung, Y. C. (2006b), "Improved Adaptive Neural Network Autopilot for Track-keeping Control of Ships: Design and Simulation", Proceedings of the 6th Asian Conference on Marine Simulator and Simulation Research, Haiphong, Vietnam, pp. 115~121.

[9] Rich Powlowicz's M_Map Toolbox for MATLAB (http://www2.ocgy.ubc.ca/~rich/map.html, accessed 2005/Dec).

[10] Saikalis, G. and Lin, F. (2001), "A Neural Network Controller by Adaptive Interaction", Proceeding of the American Control Conference, Arlington, pp. 1247-1252.

[11] Zhang, Y., Hearn, G. E. & Sen, P. (1997), "Neural network approaches to a class of ship control problems (Part I & II)", Eleventh Ship Control Systems Symposium Vol. 1 (Edited by P. A. Wilson).

Appendixes

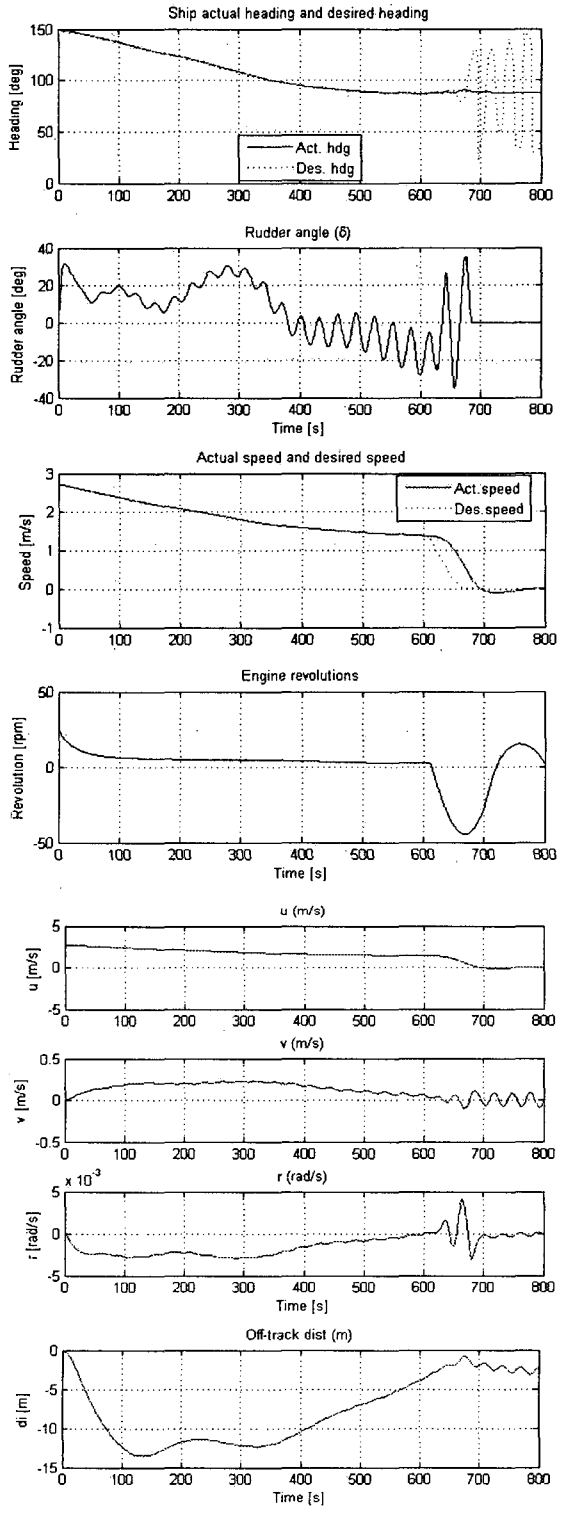

Fig. 9 Recorded data of the simulation without wind and noise in Fig. 5

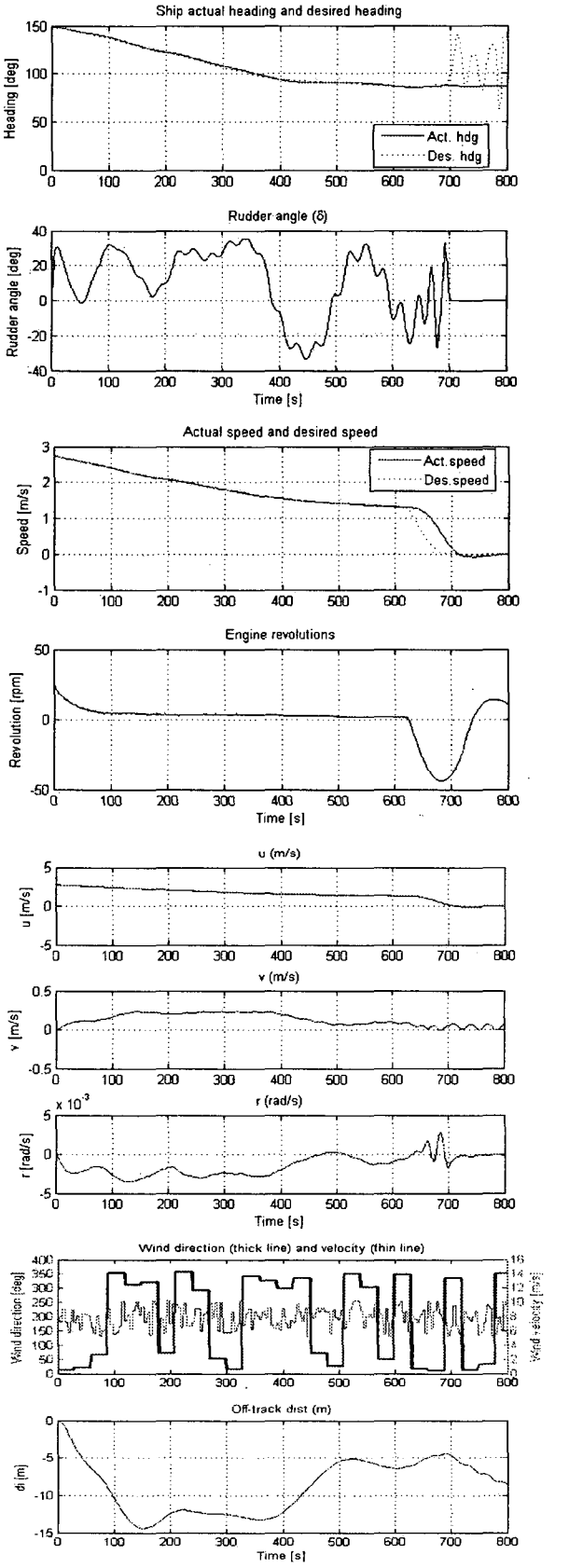

Fig. 10 Recorded data of the simulation with onshore wind and noise wind speed from 10 knots to 20 knots in Fig. 6 .

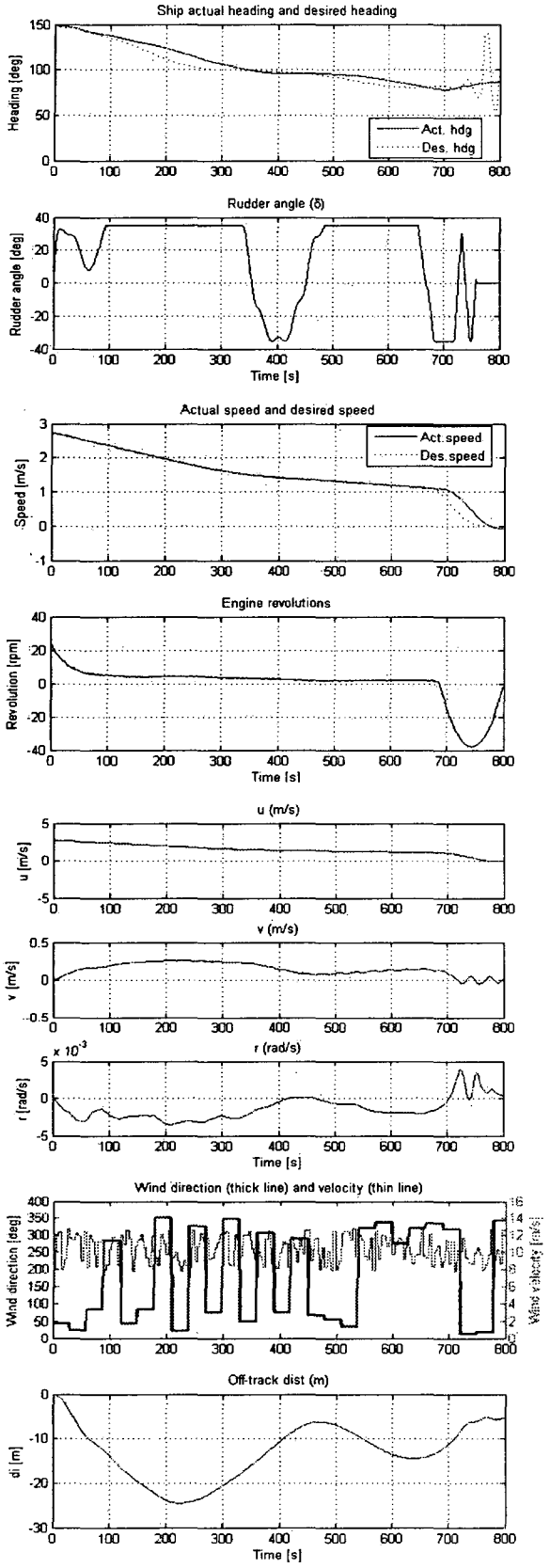

Fig. 11 Recorded data of the simulation with onshore wind and noise, wind speed from 15 knots to 25 knots in Fig. 7 .

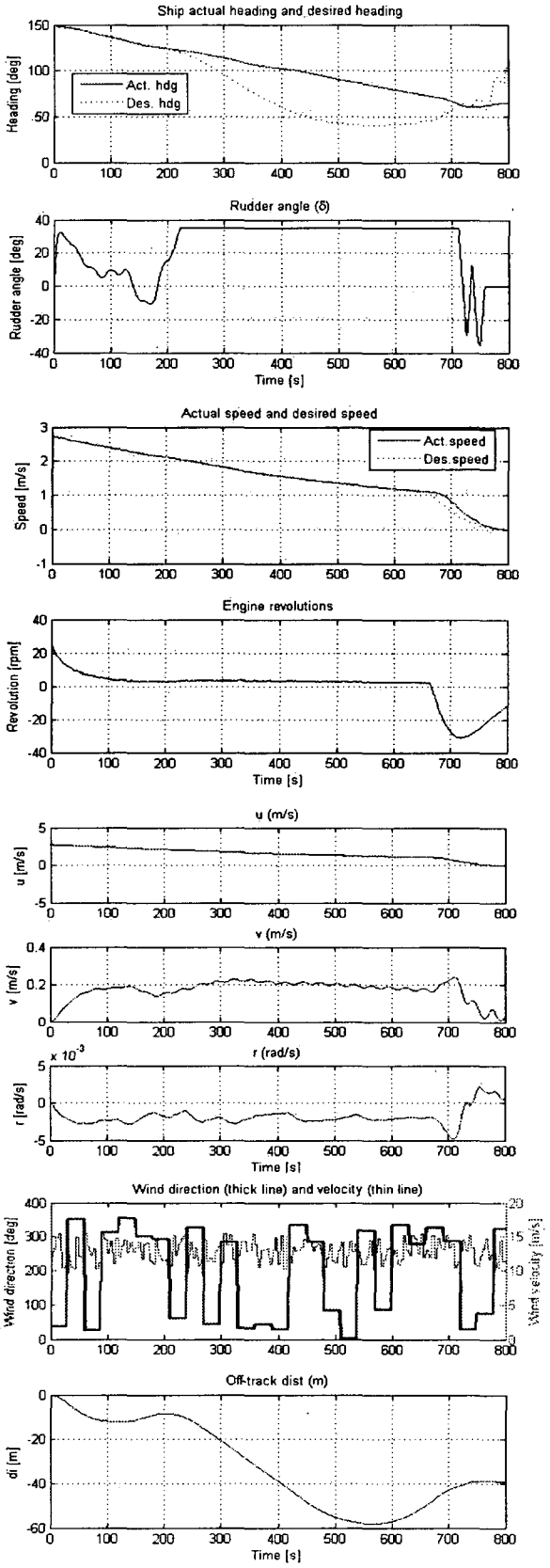

Fig. 12 Recorded data of the simulation with onshore wind and noise wind speed from 20 knots to 30 knots in Fig. 8 .