Thesis for the Degree of Doctor of Philosophy

A Study on the Automatic Ship Control Based on Adaptive Neural Networks

Advisor
Prof. Yun-Chul Jung

February 2007

Korea Maritime University, Graduate School
Department of Ship Operation Systems Engineering

Phung-Hung Nguyen

A Study on the Automatic Ship Control Based on Adaptive Neural Networks

Advisor
Prof. Yun-Chul Jung

By
Phung-Hung Nguyen

Dissertation submitted in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

In the Department of Ship Operation Systems Engineering
Graduate School of
Korea Maritime University

February 2007

A Study on the Automatic Ship Control Based on Adaptive Neural Networks

A Dissertation By

Phung-Hung Nguyen

Approved as to style and content by

Chairman	**Dr. Gang-Gyoo Jin**
Member	**Dr. Sea-June Oh**
Member	**Dr. Ja-Yun Koo**
Member	**Dr. Yang-Bum Chae**
Member	**Dr. Yun-Chul Jung**

February 2007

Acknowledgements

Many thoughts went through my mind as I compiled the work of the past three years to write this dissertation. Most of all, thoughts about the many people who enabled me to perform this research work.

Firstly, I am extremely grateful to my advisor, Professor Yun-Chul Jung, for his outstanding guidance, support and patience throughout the course of this research. His enthusiasm, dedication and encouragement has been invaluable source of inspiration and motivation for me during the last three years. I also would like to thank his family for their help and care during my stay in Pusan.

I would like to thank the committee members, Prof. Gang-Gyoo Jin, Prof. Sea-June Oh, Prof. Yang-Bum Chae, and Prof. Ja-Yun Koo for all suggestions, evaluation steps and discussions. I also would like to express my thanks to the KMU Professors who enthusiastically taught me throughout my coursework. During writing my research papers, I received many precise reference papers from Prof. Nam-kyun Im (Mokpo Maritime University), to whom I would like to express many thanks.

I am also very grateful to Dr. Dang Van Uy, Prof. Tran Dac Suu, and Prof. Le Duc Toan from VIMARU for their encouragement during my coursework. I also received much encouragement and help from VIMARU's Dept. of International Relations, particularly Mr. Pham Xuan Duong, Mr. Le Quoc Tien, to whom I would like to express special thanks. Thanks to my teachers and colleagues in the Faculty of Navigation of VIMARU for their encouragement.

I would like to thank KMU for providing me the exemption of tuition fee for my doctoral course. Very special thanks also to the KMU's Center for International Exchange and Cooperation for their help during my time in KMU. I am very grateful to Capt. Young-Sub Chung, President of Panstar Shipping Company Ltd., and his company for the financial support during my stay in Korea. I also would like to thank Mr. G.J. Bae, General Manager of Panstar Shipping Company Ltd., for his help.

I also would like to express my gratitude to all my Lab members, Eun-Kyu Jang, Suk-Han Bae, Bu-Sang Oh, Tea-Yong Kim, Chong-Ju Chae, especially Oh-Han Kweon for their help during the last three years. I would like to thank many Korean friends, who helped and made my stay in Pusan particularly joyful. Specially thanks to Jung-Ha Shin and his wife, Gyeong-Yoon Gang for their help and care.

I also would like to thank my Vietnamese friends, especially those who are KMU students, Nguyen Tuong Long, Tran Thanh Ngon, Nguyen Duy Anh, Tran Ngoc Hoang Son, Tran Viet Hong, Vu Manh Dat, Nguyen Hoang Phuong Khanh, Tran Thi Thanh Dao, Nguyen Tien Thanh, and Ngo Thanh Hoan for their help and share during our good time in KMU.

To my parents, my sincere thanks for their love and support during all these years of my education. Their belief and encouragement made me strong enough to make my dreams become true. Thanks dad for understanding me. Thanks mom for caring of my health whenever talking to me. Thanks my younger brother and my sister-in-law, Nguyen Si Nguyen and To Ngoc Minh Phuong, for taking care of everything while I am away from home. My sincere thanks also to my mother-in-law, brother-in-law and sister-in-law for their love, support and encouragement.

Last but not the least, I would like to thank my wife, Nguyen Thi Hong Thu, with all my love. Thanks for sharing with me every joyful moments as well as difficulties and disappointments. Her endless love and support was immeasurable. Thanks to my daughter, Nguyen Hong Anh, for giving me such joyful moments and motivations to complete this thesis.

Korea Maritime University, Pusan
December 2006
Phung-Hung Nguyen

A Study on the Automatic Ship Control Based on Adaptive Neural Networks

Phung-Hung Nguyen

Department of Ship Operation Systems Engineering, Graduate School
Korea Maritime University, 2007

Abstract

Recently, dynamic models of marine ships are often required to design advanced control systems. In practice, the dynamics of marine ships are highly nonlinear and are affected by highly nonlinear, uncertain external disturbances. This results in parametric and structural uncertainties in the dynamic model, and requires the need for advanced robust control techniques. There are two fundamental control approaches to consider the uncertainty in the dynamic model: robust control and adaptive control. The robust control approach consists of designing a controller with a fixed structure that yields an acceptable performance over the full range of process variations. On the other hand, the adaptive control approach is to design a controller that can adapt itself to the process uncertainties in such a way that adequate control performance is guaranteed.

In adaptive control, one of the common assumptions is that the dynamic model is linearly parameterizable with a fixed dynamic structure. Based on this assumption, unknown or slowly varying parameters are found adaptively. However, structural uncertainty is not considered in the existing control techniques. To cope with the nonlinear and uncertain natures of the controlled ships, an adaptive neural network (NN) control technique is developed in this thesis. The developed neural network controller (NNC) is based on the adaptive neural network by adaptive interaction (ANNAI). To enhance the adaptability of the NNC, an algorithm for automatic selection of its parameters at every control cycle is introduced. The proposed ANNAI controller is then modified and applied to some ship control problems.

Firstly, an ANNAI-based heading control system for ship is proposed. The performance of the ANNAI-based heading control system in course-keeping and turning control is simulated on a mathematical ship model using computer. For comparison, a NN heading control system using conventional backpropagation (BP) training methods is also designed and simulated in similar situations. The improvements of ANNAI-based heading control system compared to the conventional BP one are discussed.

Secondly, an adaptive ANNAI-based track control system for ship is developed by upgrading the proposed ANNAI controller and combining with Line-of-Sight (LOS) guidance algorithm. The off-track distance from ship position to the intended track is included in learning process of the ANNAI controller. This modification results in an adaptive NN track control system which can adapt with the unpredictable change of external disturbances. The performance of the ANNAI-based track control system is then demonstrated by computer simulations under the influence of external disturbances.

Thirdly, another application of the ANNAI controller is presented. The ANNAI controller is modified to control ship heading and speed in low-speed maneuvering of ship. Being combined with a proposed berthing guidance algorithm, the ANNAI controller becomes an automatic berthing control system. The computer simulations using model of a container ship are carried out and shows good performance.

Lastly, a hybrid neural adaptive controller which is independent of the exact mathematical model of ship is designed for dynamic positioning (DP) control. The ANNAI controllers are used in parallel with a conventional proportional-derivative (PD) controller to adaptively compensate for the environmental effects and minimize positioning as well as tracking error. The control law is simulated on a multi-purpose supply ship. The results are found to be encouraging and show the potential advantages of the neural-control scheme.

Contents

List of Figures

List of Tables

Nomenclatures

O_i	Output of neuron *i*
I_i	Input of neuron *i*
θ_i	Threshold value of neuron *i*
w_{ij}, w_{jp}	Weight of the output and hidden neurons
$g(x)$	The activation function of a neuron
$sig(x)$	Sigmoidal activation function of a neuron
$\tan sig(x)$	Tangent sigmoidal activation function of a neuron
O_i^d	Desired output value of neuron *i*
$\gamma, \gamma_1, \gamma_2$	Learning rates
n	Number of iteration in one control cycle
p, i, j	Number of neurons in input, output and hidden layer
θ_i, θ_j	Threshold values for output and hidden layers
k	Time step indicator
ite	Iteration indicator
E_k	Cost function at time step *k*
X_k^d, X_k	Desired and actual state vector at time step *k*
α	Positive constant for automatic selection of γ and *n*
β	Positive integer for automatic selection of γ and *n*
$n_{\min}, n_{\max}$	Lower and upper bounds of *n*
$\gamma_{\min}, \gamma_{\max}$	Lower and upper bounds of γ
ψ_k^d, ψ_k	Desired and actual heading angle
δ_k^c, δ_k	Command and actual rudder angle
r_k	Rate of turn (yaw rate)
ρ_i, λ, σ	Positive penalty constants in cost functions
d	Off-track distance in track control
$\vec{V}_0, \vec{V}_c, \vec{V}_s$	Velocity of ship, speed of advance and current speed
ψ_{LOS}	Line-of-Sight heading
l_k, L_k	Ship latitude and longitude at time step *k*
R_0	Radius of circle of acceptance in LOS algorithm
R	*Reach* distance

μ_k	Normalized off-track distance
μ_k^d	Desired off-track distance
a_1, a_2	Positive constants
ζ	Course change angle
u_k^d, u_k	Desired and actual ship speed
n_k^c, n_k	Command and actual engine revolution
Φ	Drift angle
L	Ship length
B	Ship breadth
K_1, K_2, ξ	Positive constant in berthing algorithm
K_{min}, K_{max}	Lower and upper bounds of K_1
di	Off-track distance in x-axis of berthing algorithm
$J^T(\psi)$	Transformation matrix
$\eta = [x, y, \psi]^T$	Vector of ship state
$\nu = [u, v, r]^T$	Vector of linear velocities of ship
$d_H = [\Delta_x, \Delta_y, 0]^T$	Vector of position of H in vessel-fixed coordinate
z_e	Distance from ship to reference point
b	Vector of bias forces and moment of environmental disturbances
$\tau = [\tau_1, \tau_2, \tau_3]^T$	Vector of control forces and moment
τ_{PD}, τ_{NN}	Control output of PD-controller and NN controller
η_d	Vector of desired state
K_p, K_d, K_p', K_d'	PD-controller parameters
$\chi_1, \chi_2, \chi_3, \chi_4$	Positive constants
$\kappa_1, \kappa_2, \kappa_3$	Positive penalty constants in cost function
$\varepsilon = J^T(\psi)\hat{e}$	Vector of transformed error

Chapter 1 Introduction

The topic of this doctoral work is the development of adaptive neural network control system and its application to marine control problems. An adaptive neural network controller is developed and applied to course-keeping control of ship. This adaptive neural network controller is then applied to design track-keeping control system for ship. Based on the proposed neural network control scheme, an automatic berthing control system for ship is developed. A similar adaptive neural network control algorithm is applied to design a hybrid neural adaptive controller for dynamic positioning of ship. This thesis contains five main chapters which will be briefly summarized in 1.2.

1.1 Background and Motivations

1.1.1 The History of Automatic Ship Control

Generally, automatic control system development for ships is to fulfill two principal targets in maritime navigation. The first target is to ensure safe navigation and the other is to control the ship economically. Safe navigation requires that, automatic control system must be able to control the ship to avoid the risk of collision, sinking, running aground. In order to control the ship economically, the automatic control system is required to control the ship in a manner that minimizes the propulsive energy loss without degrading the safe navigation. So far, many control methods have been applied to automatic control of ship to obtain these targets.

The history of ship control started in 1908 with the invention of the gyrocompass which was the basic instrument in the first feedback control systems for heading

control or autopilots today, and it extends further with the development of local positioning systems in the 1970s. These systems and new results in feedback control resulted in new applications like dynamic positioning (DP) systems for ships and rigs. From late 1970s to date, track control system was developed to control not only ship's heading but also position with respect to a reference track. The availability of global positioning systems such as GPS and GLONASS, and successful results with controllers in ship autopilots and dynamic positioning systems resulted in a growing interest for waypoint tracking control systems [79]. More recently, studies on automatic ship maneuvering in restricted waters (such as automatic berthing systems) have been reported in literature [27], [89], and [90].

1.1.2 The Intelligent Control Systems

Generally, it is difficult to accurately represent a complex plant or process by a mathematical model or by a simple computer model. Even when the model itself is tractable, controller using a "hard" (non-soft or crisp) control algorithm might not provide satisfactory performance. Furthermore, the crisp control algorithms can not formulate the actions made by an experienced and skilled operator, who can performs high-level control of some industrial processes successfully [63].

As mentioned in [63], from the control theory point of view, model-based control can not provide satisfactory results if the process model itself is inaccurate. Even when an accurate model is known, if the parameter values are partially known, ambiguous, or vague, then approximates have to be made. In such a case, crisp control algorithms based on incomplete information usually will not give satisfactory results. To improve robustness of the control systems, classical feedback control has used methods such as: adaptive and robust control technique designed to cope with uncertainties due to large variations in parameter values, environmental conditions, and signal inputs. However, the region of operability of the control system will be restricted, although it will be considerably large in comparison with non-robust classical control systems. In complex processes in practice, the range of uncertainty may be substantially larger than can be tolerated by crisp algorithms of adaptive and robust control. In such situations, "intelligent" control techniques are useful.

Since late 1980s, research interests in automatic control have turned to developing the "intelligent control systems". Intelligent control can be classified into, but not limited to, the following areas: expert or knowledge based systems, fuzzy logic controllers and NN based controllers.

(1) Expert Systems

The first field of artificial intelligence to be commercially recognized is expert system. One of the primary objectives of expert systems is to mimic human expertise and judgment using a computer program by applying knowledge of specific areas of expertise to solve finite, well-defined problems. These computer programs contain human expertise (called *heuristic knowledge*) obtained either directly from human experts or indirectly from books, publications, codes, standards, or databases, as well as general and specialized knowledge that pertains to specific situations [42]. Expert systems have the following advantages

(a) Experts need not be present for a consultation; expert systems may be delivered to remote locations where expertise may not be otherwise available.
(b) Expert systems do not suffer from some of the shortcomings of the human beings (for example, they do not tired or careless as the work load increase) but, when properly used, continue provide dependable and consistent results.
(c) The techniques inherent in the technology of expert systems minimize the recollection of information by requesting only relevant data from the user or appropriate databases.
(d) Expert knowledge is saved and readily available because the expert system can become a repository for undocumented knowledge that might otherwise be lost (for example, through retirement).
(e) The development of expert systems forces documentation of consistent decision-making policies. The clear definition of these policies makes the overall decision-making process transparent and the implementation of policy changes instant and simultaneous at all sites.

On the other hand, expert systems have disadvantages that affect their use

(a) They usually deal with static situations.
(b) They must be kept up to date as conditions change.
(c) They often can not be used in novel or unique situations.
(d) Results are very dependent on the adequacy of the knowledge incorporated into the expert system.
(e) Perhaps most important, they do not benefit from experience except through updating the knowledge base (based on human experience).
(f) Expert systems are unable to solve problems outside their domain of expertise. In many cases they are unable to detect the limitations of their domain.

(2) Fuzzy Control Systems

Fuzzy systems are knowledge-based or rule-based systems. The heart of a fuzzy system is a knowledge base consisting of the so-called fuzzy IF-THEN rules. A fuzzy IF-THEN rule is an IF-THEN statement in which some words are characterized by continuous membership functions [46]. There are five major branches in fuzzy theory: (1) fuzzy mathematics, where classical mathematical concepts are extended by replacing classical sets with fuzzy sets; (2) fuzzy logic and artificial intelligence, where approximations to classical logic are introduced and expert systems are developed based on fuzzy information and approximate reasoning; (3) fuzzy systems, which include fuzzy control and fuzzy approaches in signal processing and communications; (4) uncertainty and information, where different kinds of uncertainties are analyzed; and (5) fuzzy decision making, which considers optimization problems with soft constraints [46]. These five branches are not independent and there are strong interconnections among them.

Practically, the most significant applications of fuzzy systems have concentrated on control problems. Fuzzy systems can be used either as open-loop controllers or closed-loop controllers. When used as an opened-loop controller, the fuzzy system usually sets up some control parameters and then the system operates according to

these control parameters. Many applications of fuzzy systems in consumer electronics belong to this category. When used as a closed-loop controller, the fuzzy system measures the outputs of the process and takes control actions on the process continuously. Applications of fuzzy systems in industrial processes belong to this category. The fundamental difference between fuzzy control and conventional control is that, conventional control starts with a mathematical model of the process and controllers are designed for the model; fuzzy control, on the other hand, starts with heuristic and human expertise (in terms of fuzzy IF-THEN rules) and controllers are designed by synthesizing these rules [46].

Many different kinds of fuzzy control systems have been introduced to control practices. The theory and typical applications of fuzzy control systems can be found in [39], [42], [46], and [84]. For marine control problems, applications of fuzzy control systems have been also investigated by many researchers. Interesting applications to surface ship control can be found in [6], [9], [22], [32], [33], [45], [66] - [68], [88], and [91] - [93].

(3) Neural Network Control Systems

In recent years, the neural network control technology has grown very rapidly. Many neural network control systems of different structures have been proposed and widely applied in a range of technical practices. NNs are very attractive in control applications because of the following properties: (1) massive parallelism; (2) inherent nonlinearity; (3) powerful learning capability; (4) capability of generalization; (5) guarantied stability for certain nonlinear control problems (see [12], [41], [63], and [75] for further details).

In addition, NNs have been proved to be universal controllers, “that is, if the system to be controlled is stabilized by a continuous controller, there exists a NN which can approximate the controller such that the controlled system by the NN is stabilized with a given bound of output error” [8]. Among neural control structures mentioned in literature and applied to practices, such as [5], [15], [17], [25], [29], [40], [52], [69], [70], [72], [73], [75], adaptive NNs control has been proposed to control

dynamical systems. The basic idea is to use NNs in connection with the adaptive control methods.

Among the above intelligent control technologies, NNs and fuzzy logic have been applied to control of dynamic systems. NNs and fuzzy logic technologies are quite different, and each has unique capabilities that are useful in information processing. Yet, they often can be used to accomplish the same results in different ways. For instant, they can speed the unraveling and specifying the mathematical relationships among the numerous variables in complex dynamic process. Both can be used to control nonlinear systems to a degree not possible with conventional linear control systems. They perform mappings with some degree of imprecision [42].

The review of literature mentioned above has shown that the application of NNs to marine control problems is very potential, and NNs are attractive in designing intelligent adaptive control systems. Therefore, in this thesis an adaptive NN control system is developed for ship control problems in direct methods and will be presented in chapter 2.

1.2 Objectives and Summaries

The goal of this research is to develop an adaptive NNC for marine vehicles. The proposed NNC is then applied to four control problems: heading control, track control, berthing control, and dynamic positioning control. The objectives of the research are summarized as follows

(a) Developing an adaptive neural network by adaptive interaction controller. The proposed ANNAI can be online-trained and its parameters can be adaptively updated;

(b) Developing an adaptive NN-based heading control system for ships using the proposed ANNAI. Investigating its performance and compare with the conventional BP based NNC;

(c) Developing an adaptive NN-based track control system for ships employing the learning ability of the ANNAI. Verifying the track control system by testing the adaptability with external effects using computer simulations;

(d) Developing an automatic berthing system applying the proposed ANNAI in controlling ship heading and speed. Adopting a berthing guidance system for ship;

(e) Proposing a DP control system of ship by combining the ANNAI with conventional proportional-derivative (PD) controller. Validating and evaluating the proposed hybrid control scheme through computer simulations.

1.3 Original Contributions and Major Achievements

The main contributions and achievements produced by this work are described as follows:

(a) We developed an adaptive NN by adaptive interaction, called ANNAI.

(b) We introduced an algorithm for automatic updating the learning rate and number of training iterations to improve the adaptability of ANNAI.

(c) We proposed an adaptive heading control system for ships with the proposed ANNAI.

(d) We designed an adaptive track control system for ships using the ANNAI controller and a modified LOS algorithm.

(e) We designed an automatic berthing control system based on the ANNAI.

(f) We proposed a berthing guidance algorithm which can guide the ship to

follow the desired berthing route.

(g) We developed a hybrid neural adaptive controller for DP control of ship. The controller can avoid the use of ship mathematical model and estimation of external disturbances.

(h) We introduced an algorithm to move the reference point in low-speed maneuvering control of ship. This algorithm can ensure that the ship can follow the intended track while ship heading is kept at the desired value.

1.4 Thesis Organization

Chapters: Chapter 2 presents the ANNAI controller which can adapt its weights at every control cycle and the algorithm for automatic updating the learning rate and number of training iterations to improve the adaptability of ANNAI; Chapter 3 introduces an application of the ANNAI to heading control of ships and compares with conventional BPNN controller; Chapter 4 presents a track control system based on the ANNAI controller; Chapter 5 discusses the application to automatic berthing control of the proposed ANNAI controller; Chapter 6 investigates a hybrid neural controller by combining the ANNAI controllers with a PD-controller for DP control of ship; and Chapter 7 summaries the advantages and limitations of the proposed NN control schemes, possible applications and the future developments of the research works.

Appendixes: This thesis uses mathematical model of ships as well as DP system for simulation studies. The mathematical model of DP ships is briefly reviewed in Appendix A. The referred mathematical model of ships and their Matlab M-files are presented in Appendix B.

Chapter 2 Adaptive Neural Network by Adaptive Interaction

2.1 Introduction

The potential of NNs for control has received much attention and rapidly grown in the 1990s, because of the ability of NNs in solving some awkward control problems where the high non-linearities of the controlled plant and unpredictable external disturbances make the plant's behaviors hard to control. In addition, the fast calculation of NNs is also suitable for real time control applications. The theory and applications of NNs in control can be found in [14], [19], [40], [75].

The application of NN control theory in the field of marine is relatively new. A study in feasibility of using NNs to control surface ships was discussed in [65]. A feedback optimal NNC for dynamic systems was proposed and applied to ship maneuvering [38]. The NNC requires off-line training phase for the synaptic weights. Later, [21] introduced a recurrent NN for ship modeling and control and compared with classical methods. To achieve an adaptive NNC for ship, Y. Zhang *et al.* used multi-layer NNC with single hidden layer and on-line training strategy of network weights as adaptive NNC for ship control including course-keeping, track-keeping and auto-berthing control [89], [90]. In their work, a BP algorithm was used for weights updating.

There have been different methods to utilize NNs as adaptive controllers and they can be categorized into indirect control (Fig. 2.1) and direct control (Fig. 2.2). In indirect control, the parameters of the plant are estimated using a NN, and the

parameters of the controller are chosen assuming that the identified parameters represent the true values of the plant parameter vector based on certainty equivalent principle. This scheme does not require any priori knowledge about the plant. Still, it requires another NN in addition to the NN for control to emulate the plant (shown in Fig. 2.1). The plant emulator needs an off-line phase of training with a sufficiently large data set for identification of the forward or inverse dynamics of the plant to be controlled [14], [40]. Direct scheme is simpler than indirect scheme. It does not require the iterative off-line training process to identify the plant parameters and provides adaptive laws for updating the NN weights.

In this thesis, a direct adaptive NNC for ship control problems is proposed. This NNC is based on the adaptation algorithm developed in [70] and the extension of NNC proposed in [23] with some modifications and improvements. The proposed NNC can be trained on-line so that, in this control scheme, off-line training phases are removed. Additionally, both the learning rate and the number of iterations for weight updating can be dynamically selected [19], [20]. With this adaptation method, the sufficient (but not excessive) training for on-line training requirement is achieved, no pre-test of the NN is required and the training time is minimized without adversely affecting the ability of the network to learn the plant's behavior. This new feature has not been found in the previous works.

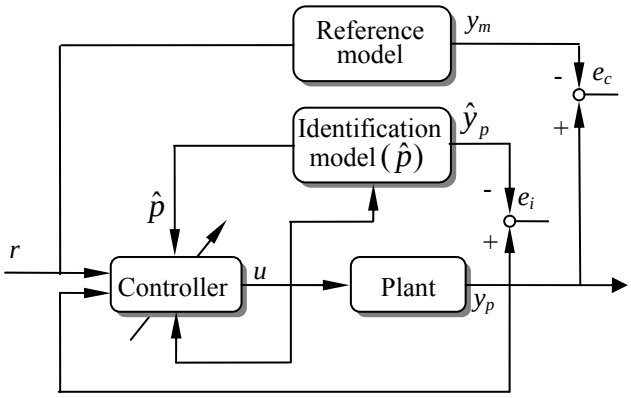


Fig. 2.1 Indirect adaptive control

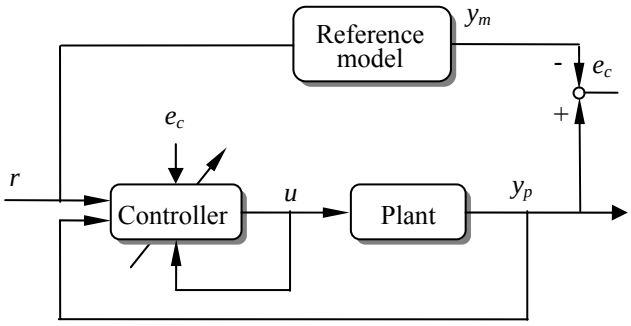


Fig. 2.2 Direct adaptive control

2.2 Adaptive Neural Network by Adaptive Interaction

This subchapter will present the summary of direct NN control scheme applications to practice and then focus on the details of the proposed ANNAI controller.

2.2.1 Direct Neural Network Control Applications

In the study of [43], comparisons are made about stability, speed of convergence, noise rejection, memory size, control effort, number of required calculations, and tracking performance for the three control algorithms. Those are neural network approach (method similar to Miller's Cerebellar Model Arithmetic Computer-CMAC) and two traditional adaptive systems methods, namely the self-tuning regulator (STR) of [36] and the Lyapunov-based model reference method by Parks (1966) (see [36] for more details). This study showed the advantages and disadvantages of the three approaches through simulation experiments and showed that, the challenge to the researchers and designers in control is to take advantage of the desirable properties of each of the classes of systems. Up to nowadays, the studies on combining the experience and dependability of classical and traditional adaptive control with the potential and promise of NN-based systems have been investigated and proposed in literature. The best characteristics of the above different classes of systems have been exploited.

Using the well-known BP algorithm, NNC can perform the direct adaptive control function. The NN weights are updated on-line at every control cycle. The NNC can learn dynamically and no trainer is necessary, so off-line training phase of the NNC can be removed. The configuration of this control scheme is described in [89]. The task of the NN is to "learn" the plant behavior from its current and previous states (through time delay operators z^{-m} and z^{-n}), and then to infer appropriate control actions in the next time step.

In [87] the authors proposed two kinds of NN-based predictor which can forecast the output of nonlinear processes over a certain horizon in the future. Based on the NN predictor, a strategy of long-range predictive control is proposed. In order to implement the on-line adaptive control, a recursive least square (RLS) type learning algorithm was proposed to speed up the learning of the feedforward NN.

In [28] a NN-based adaptive predictive control algorithm for nonlinear non-minimum phase systems was proposed. In this control scheme, the nonlinear system is separated into linear non-minimum phase part and the nonlinear part by Taylor series expansion. The resulting nonlinear part is identified by a NN and compensated in the control algorithm such that feedback linearization can be achieved.

In [50] it was shown that, NN can be used to improve upon approximate dynamic inversion for control of uncertain nonlinear systems. In one architecture, the NN adaptively cancels inversion errors through on-line learning. Such learning is accomplished by a simple weight update rule derived from Lyapunov theory, thus assuring the stability of the closed-loop system. The authors applied this in control of an agile-air missile autopilot.

More recently, to achieve NN-based control schemes with proven stability for some classes of nonlinear systems, the NN control approach have been combined with adaptive control in such way that, "the NNC exhibits a learning-while-functioning feature, instead of learning-then-control" [72]. In [72], the structure of the NN controller is derived using filtered error notations and passivity approach. A uniform ultimate boundedness of the closed-loop system is given in the sense of Lyapunov. In

[52] the author proposed a direct adaptive NN control scheme to control an under water vehicle. The NN is used to approximate the dynamics of the controlled plant, a control law is derived and the NN weights are updated based on Lyapunov method and ensure that the closed-loop error converges to zero and the boundedness of the weights can be shown. The similar methods also found in controlling a class of nonlinear systems in the face of both unknown nonlinearities and unmodeled dynamics [3], [51], and [74].

In addition, NN-based model reference adaptive control has also been discussed. [24] proposed an approach to model reference adaptive control based on NN for a class of first-order continuous-time nonlinear dynamical systems. The NN is used to compensate adaptively the nonlinearities in the plant. A stable controller-parameter adjustment mechanism, which is determined using the Lyapunov theory, is constructed using a σ-modification-type updating law. The control error converges asymptotically to a neighborhood of zero.

In [70] the authors proposed a direct adaptive NN control scheme by adaptive interaction theory. According to this study, the neurons in NN are considered subsystems which are called devices in a complex system. It is equivalent to BP algorithm but requires no feedback network to back propagate the error. The adaptive NN control of various systems using this approach was simulated in [23] to demonstrate the effectiveness of the algorithm.

2.2.2 Description of the ANNAI Controller

It is shown in [70] that, using the standard notations as follows, for $i, j \in N$

O_i the output of neuron i;
I_i the input of neuron i;
θ_i the threshold value of neuron i;
w_{ij} the weight of the connection from neuron i to neuron j;
$g(x)$ the activation function of a neuron;
O_i^d the desired output value of neuron i (for output neurons);
γ the learning rate,

the NN can be described by

$$I_i = \sum_{j \in N} w_{ij} O_j + \theta_i \ , \tag{2.1}$$

$$O_i = g(I_i) = g\left(\sum_{j \in N} w_{ij} O_j + \theta_i \right). \tag{2.2}$$

The goal is to minimize the following error

$$E = \frac{1}{2} \sum_{i \in N} e_i^2 \ , \tag{2.3}$$

where $e_i = O_i^d - O_i$ if i is output neuron.

And the adaptation algorithm for NN in [70] can be written as

$$\dot{w}_{ij} = g'(I_i) \frac{O_j}{O_i} \sum_{k \in N} w_{ki} \dot{w}_{ki} - \gamma g'(I_i) O_j e_i \ , \tag{2.4}$$

where $\dot{w}_{ij}$ is the increment of weights, $g'(I_i)$ is the derivative of $g(I_i)$ with respect to I_i.

Equation (2.4) describes Brandt-Lin algorithm for adaptation of weights in NN. Later, a NNC based on the Brandt-Lin algorithm was proposed in [23] where the simulation showed the effectiveness of the NNC, and some notes and conclusions were figured out.

(1) General Form of the On-line Trained ANNAI

The configuration of the ANNAI proposed in this thesis is shown in Fig. 2.3. Using the cost function described in [89] we have

$$E_k = \frac{1}{2}(X_k^d - X_k)^T P(X_k^d - X_k) + \frac{1}{2}u_k^T \Lambda u_k , \tag{2.5}$$

where X_k^d and X_k are desired state vector and actual state vector respectively; u_k^c is the command control vector and u_k is the actual control vector; P is a real symmetric positive semi-definite matrix reflecting the weightings of the plant variables to be controlled; Λ is a real symmetric positive definite matrix for the control vector.

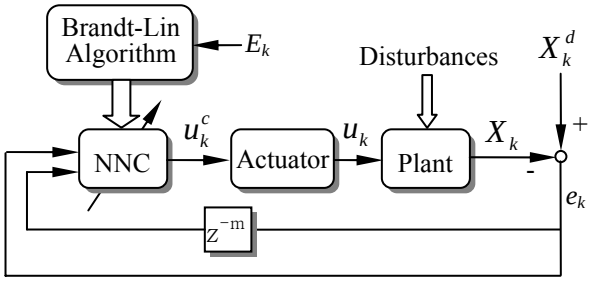


Fig. 2.3 Configuration of the ANNAI controller. The inputs of NNC consist of e_k and its delayed signals. The cost function E_k is processed by Brandt-Lin algorithm to adapt NNC weights so that E_k is minimized.

Similarly in [89], the training process of the network is carried out within each control cycle indicated by k with n being the number of the training iterations. The adaptation algorithm (2.4) is used to adjust the synaptic weights in the NN so that, cost function E_k can be minimized. The inputs to the NNC consist of error $e_k = X_k^d - X_k$ and its time delayed values. The task of the NNC is to infer appropriate control actions in the next time step after "learning" the behavior of the plant's desired and actual states through e_k. To improve adaptation speed and ability of the NNC, a method to adjust the network learning rate γ and number of iterations n automatically is proposed (Fig. 2.6).

(2) Automatic Selection of Learning Rate and Number of Training Iterations

During the training process, if the learning rate γ is too large, then the NN can fail to converge, jumping back and forth over the minimum [19]. On the other hand, if the

learning rate is too small, the adaptation may be very slow to converge. [89] and [90] applied an "intensive training" scheme to the BPNN which needs some pre-tests to achieve the sufficient (but not excessive) network training and the on-line control requirement (see Fig. 2.4). Similarly, in [57], various simulation works were carried out to verify the NNC so that the learning rate and number of training iterations n were carefully selected. Thus, adapting the learning rate can significantly speed up the convergence of the weights and remove the manual selection of this parameter.

In [20], a new strategy called "moderate training" was proposed. The number of training iterations specified for each control cycle was not fixed but dynamically selected as a function of the cost function. In the new control cycle, the previously selected weights were not discarded but used as starting values for the new updating process (see Fig. 2.5).

In this study, a new strategy for the automatic selection of both n and γ simultaneously based on [19] is proposed. Here the learning rate is increased if the cost E_k is decreasing. If the cost increases during the process, the learning rate is repeatedly reduced until the cost decreases. Simultaneously, the number of training iterations is selected such that it cooperates with the selected learning rate to achieve the sufficient (but not excessive) network training and the on-line control requirement.

The algorithm for automatic selection of n and γ can be described as

Step 1	IF $E(k+1) < E(k)$ THEN increase learning rate
	$\gamma(k+1) = (1+\alpha)\gamma(k)$, and reduce number of iteration
	$n(k+1) = n(k) - \beta$
	ELSE decrease learning rate
	$\gamma(k+1) = (1-\alpha)\gamma(k)$, and increase number of iteration
	$n(k+1) = n(k) + \beta$
Step 2	k = k + 1 and go to next control cycle

Where α is a positive constant and β is a positive integer. For "safe" learning we can select the lower and upper bounds for n and γ such that

$$n_{\min} \le n(k) \le n_{\max}, \text{ and}$$

$$\gamma_{\min} \le \gamma(k) \le \gamma_{\max}.$$

2.3 Training Method of the ANNAI Controller

Firstly in this section, comparison between the training method of the proposed ANNAI controller with those of *intensive* BP training and *moderate* BP training is reviewed.

2.3.1 *Intensive* BP Training

The flow chart of *intensive* BP training method as in [89] and [90] is shown in Fig. 2.4. Fixed values of n and γ were used. At the beginning of a control cycle indicated by k, NN weights are initialized as small random values. The outputs of neurons in hidden layer and output layer are then calculated using those weights. Next, the NN weights are updated using BP training method so that cost function E_k can be minimized. This process is iteratively repeated n times before new control cycle starts ($k = k + 1$). The NN output at iteration n is the control output at control cycle k.

2.3.2 *Moderate* BP Training

The flow chart of *moderate* BP training method as in [20] is shown in Fig. 2.5. Fixed value of γ was used. The number of training iterations n specified for each control cycle was not fixed but dynamically selected as a function of the cost function ($n_k = f(E_k)$). The iterative training was similar to that of *intensive* method except that, in the new control cycle, the previously selected weights were not discarded but used as starting values for the new updating process. The aim of this method is to terminate training at the iteration where the cost function can not be reduced any more. Hence we can reduce the training time and avoid excessive training.

2.3.3 Training Method of ANNAI Controller

The flow chart of training method of the proposed ANNAI controller is shown in Fig. 2.6. The values of n and γ are not fixed but automatically selected at every control cycle as described in 2.2.2 (page 16). At the beginning of a control cycle indicated by k, NN weights are initialized as small random values. The outputs of neurons in hidden layer and output layer are then calculated using those weights. Next, the NN weights are updated using Brandt-Lin training method so that cost function E_k can be minimized. This process is iteratively repeated n_k times before new control cycle starts ($k = k + 1$). The NN output at iteration n_k is the control output at control cycle k.

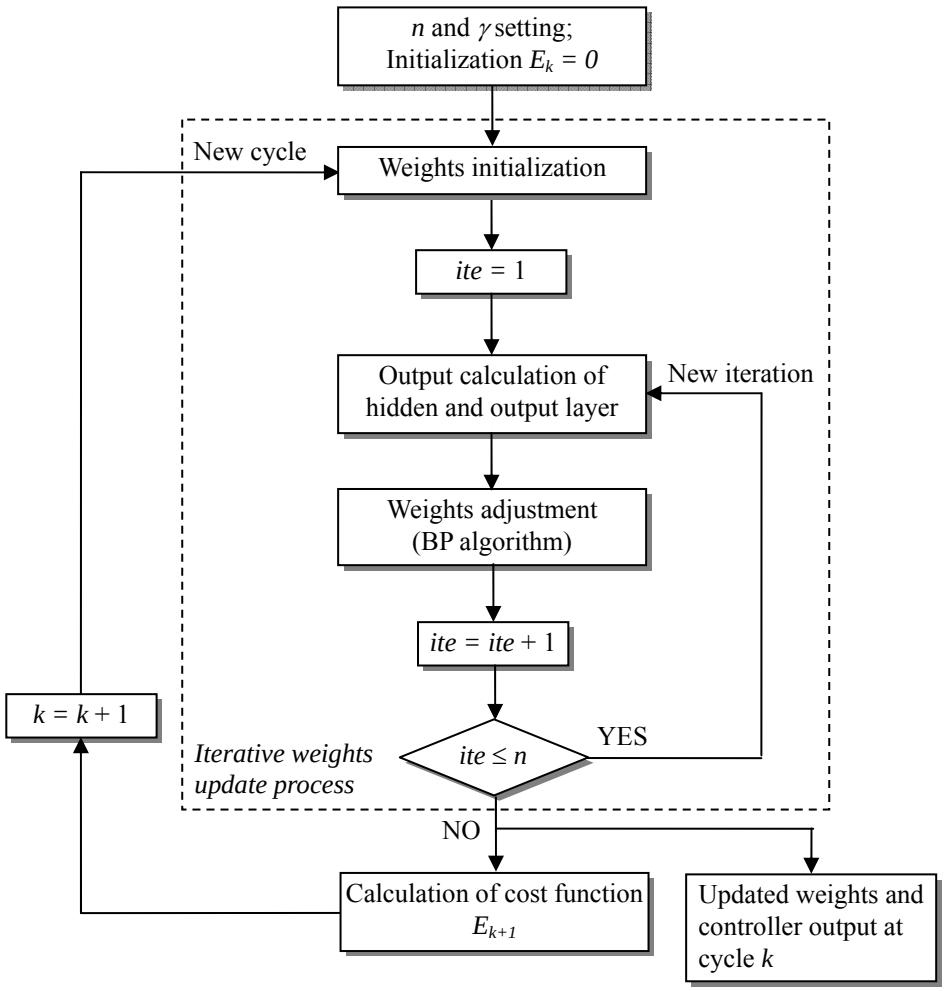


Fig. 2.4 Flow chart of "intensive" BP algorithm. n and γ is fixed.

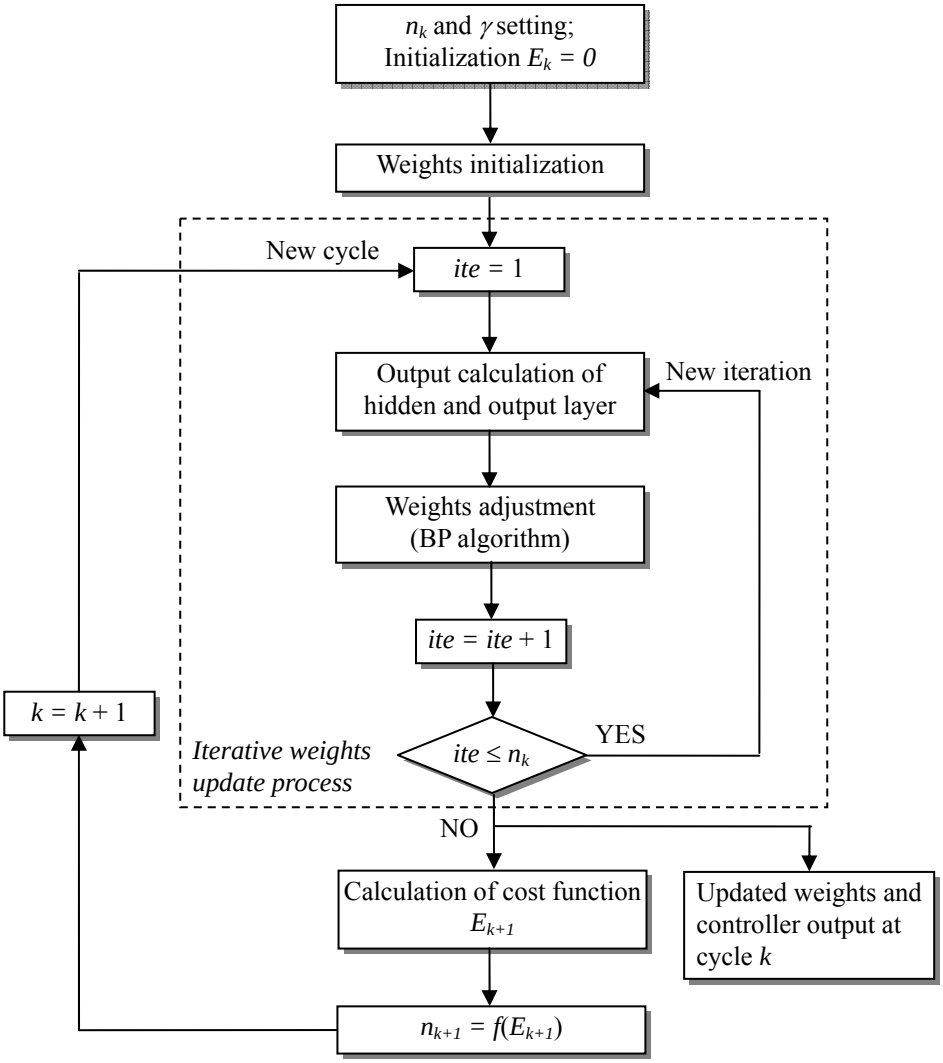


Fig. 2.5 Flow chart of "moderate" BP algorithm. *n* is adaptively selected.

The training method of ANNAI controller employs the advantages of both *intensive* and *moderate* training. Additionally, learning rate is also adaptively selected and Brandt-Lin algorithm is used for weights updating at each iteration. This approach helps to speed up training and enhance the adaptability as well as stability of the proposed ANNAI controller compared with conventional BPNN. The comparisons between ANNAI and BPNN were shown in [58], [59] and will be presented again in chapter 3.

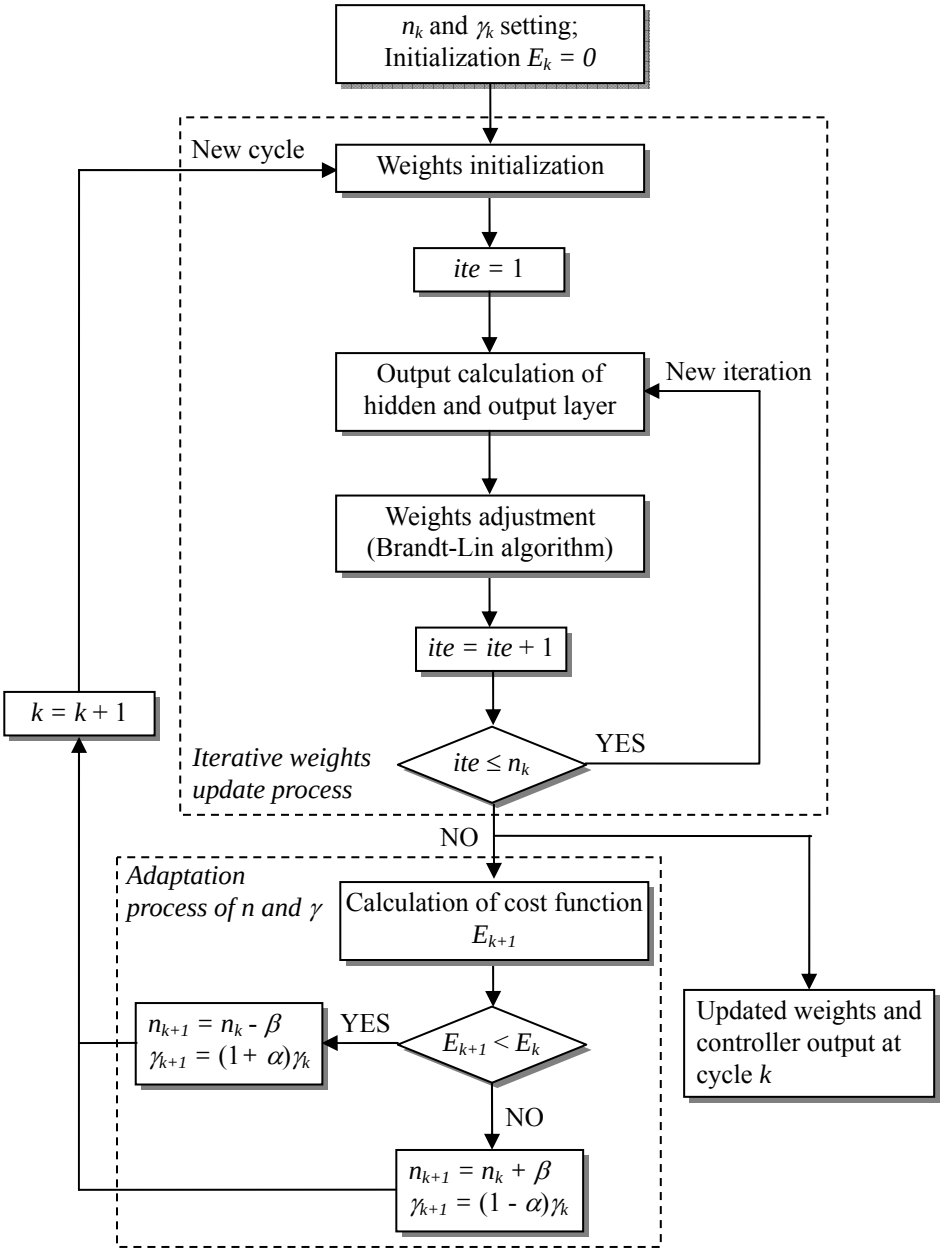


Fig. 2.6 Flow chart of the proposed ANNAI algorithm. Both n and γ is adaptively selected.

Chapter 3 ANNAI-based Heading Control System

3.1 Introduction

The course-keeping capabilities were the first applications for automatic ship control. Elmer Sperry (1860-1930) constructed the first automatic ship steering mechanism in 1911 extended from gyrocompass (see [78], [79]). Nowadays, modern autopilots can execute more complex maneuverings such as turning, docking operations and are used not only for surface ships but submarines, torpedoes, missiles as well.

In 1922, Nicolas Minorsky (1885-1970) presented a three-term control law feedback control system, which today is referred to as Proportional-Integral-Derivative (PID) control. The autopilot systems of Sperry and Minorsky were both single-input single-output (SISO) systems, and the heading of the ship was measured by a gyrocompass. The performance of the conventional PID autopilot in rough sea was analyzed by M. Blanke (1981) [79].

In the late 1970s and early 1980s, marine adaptive autopilots were rapidly developed with adaptation schemes applied to conventional PID autopilots. Other approaches like stochastic adaptive systems, self-tuning adaptive control, and model-reference adaptive control have been applied. More methods which have been recently explored include *H-infinitive* adaptive control [10], self-tuning poles assignment optimal control [13], and a good summary of autopilots development can be seen in [78], [79] and [89].

Since late 1980s, the “intelligent” control systems have been developed and

applied to marine control. Many studies in intelligent control of marine vehicles have been reported in literature. In [89], researches of these areas were listed. [92] proposed a fuzzy logic course autopilot, and an improved one was introduced in the next study [93]. [16] used genetic algorithms for ship steering control optimization. In [88], a model reference adaptive robust fuzzy autopilot was investigated. More discussions can be seen in [66] - [68]. In the research of [89] and [90], three control systems of ship (course-keeping, track-keeping, and automatic berthing) using the BP-based NNC were introduced. The authors compared their BP-based NNC with a well-tuned discrete PID controller. Compared to the PID controller, the NNC showed distinct advantages in terms of higher performance accuracy, less adjustment of rudder, and resistance to noise.

In this chapter the ANNAI controller proposed in chapter 2 is designed for heading control of ships and compared with the BP-based NNC as presented in [89] and [90]. The simulation results and discussions will be provided.

3.2 Heading Control System

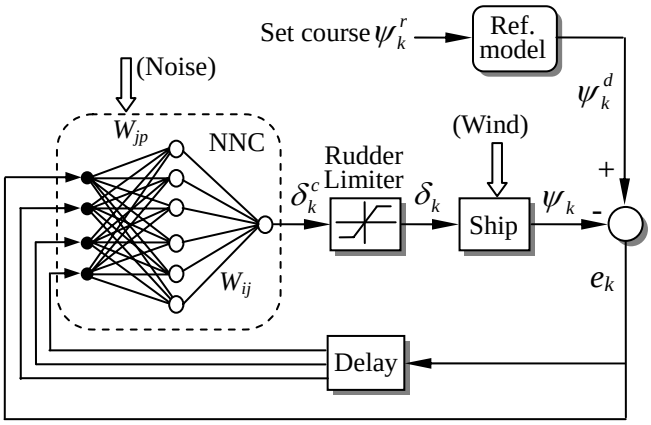


Fig. 3.1 ANNAI-based heading control system configuration.

In this subchapter, a direct adaptive ANNAI-based heading control system for course-keeping and turning control is proposed. The NNC is a multilayer feedforward

NN with one hidden layer. The configuration of the NNC is shown in Fig. 3.1 and Fig. 3.2, where w_{ij} is used to indicate the weights between output layer and hidden layer, and w_{jp} is used to indicate the weights between hidden layer and input layer. In general, the subscripts *p*, *j* and *i* indicate the number of neurons in input, output and hidden layer respectively. In this system, $u_k^c = \delta_k^c$, $u_k = \delta_k$, $X_k = \psi_k$, and $p = 4$, $j = 6$, $i = 1$ (The NNC consists of four input neurons, six hidden neurons and one output neuron). The input signals of the NNC are merely heading error and its time-delayed values. The task of the ANNAI-based heading control system is to find appropriate rudder angle to minimize the following cost function

$$E_k = \frac{1}{2}(\psi_k^d - \psi_k)^2, \tag{3.1}$$

in which, ψ_k^d and ψ_k are desired heading and actual heading respectively. Thus we have

$$\frac{\partial E_k}{\partial \psi_k} = -(\psi_k^d - \psi_k) = -e_k. \tag{3.2}$$

The output of neuron *i* in the output layer with sigmoidal activation function is

$$O_i = \delta_k^c = sig(I_i) = \frac{1}{1+\exp(-I_i)}. \tag{3.3}$$

The output of neuron *i* in the output layer with tangent sigmoidal activation function is

$$O_i = \delta_k^c = \tan sig(I_i) = \frac{2}{1+\exp(-2I_i)} - 1. \tag{3.4}$$

The output of neuron *i* in the output layer with linear activation function is

$$O_i = \delta_k^c = K \cdot I_i, \tag{3.5}$$

where, K is a constant gain and

$$I_i = \sum_j (w_{ij} O_j) + \theta_i \ . \tag{3.6}$$

Here $O_{i=1} = \delta_k^c$ is output of NNC or rudder command, I_i is the summation of the weighted inputs to the units in the output layer plus θ_i, where θ_i is the threshold value of the output layer neurons. The neurons in hidden layer have the sigmoidal activation function. The output of neurons in the hidden layer is

$$O_j = sig(I_j) = \frac{1}{1+\exp(-I_j)}, \tag{3.7}$$

where,

$$I_j = \sum_p (w_{jp} O_p) + \theta_j \ . \tag{3.8}$$

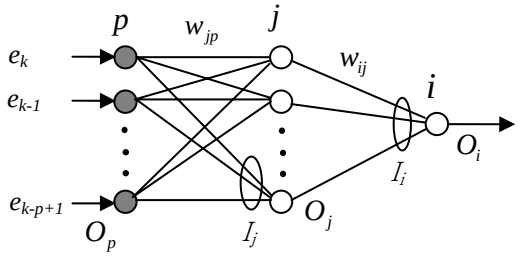


Fig. 3.2 NN configuration.

Now, applying the adaptation algorithm (2.4) for the hidden weights of the NNC, we have

$$\dot{w}_{jp} = O_p[\phi_j sig(-I_j) + \gamma \cdot 0] = O_p \phi_j sig(-I_j), \tag{3.9}$$

where,

$$\phi_j = w_{ij} \cdot \dot{w}_{ij} . \tag{3.10}$$

As stated in [23], 'the adaptation law for w_{ij} is more complicated as it is linked to the plant to be controlled'. Here O_p is the set of p inputs to the NNC consisting of current heading error e_k and its delayed signals at time steps *k-1*, *k-2*,⋯, *k-p+1* ($O_p = e_p$). Applying the adaptation algorithm (2.4) we can get the adaptation law for w_{ij} by (3.11), (3.12), or (3.13) if the activation function of the output neurons are (3.3), (3.4), or (3.5) respectively

$$\dot{w}_{ij} = \gamma[\sigma(I_i)]' sig(I_j) e_k , \tag{3.11}$$

$$\dot{w}_{ij} = \gamma[\tan sig(I_i)]' sig(I_j) e_k , \tag{3.12}$$

$$\dot{w}_{ij} = \gamma(K \cdot I_i)' sig(I_j) e_k . \tag{3.13}$$

According to [23], instead of (3.11), (3.12), and (3.13) the update law for w_{ij} can be approximated by

$$\dot{w}_{ij} = \gamma \cdot sig(I_j) \cdot e_k = \gamma \cdot O_j \cdot e_k . \tag{3.14}$$

Based on the work in [89], we can modify the cost function in (3.1) in the form of

$$E_k = \frac{1}{2}\left[\rho(\psi_k^d - \psi_k)^2 + \lambda\delta_k^2 + \sigma r_k^2\right], \tag{3.15}$$

in which r_k is the yaw rate at time step k, σ is a constant. Using the chain rule we can modify (3.2) as the following

$$\frac{\partial E_k}{\partial \psi_k} = \frac{\partial E_k}{\partial \psi_k} - \frac{\partial E_k}{\partial \delta_k} + \frac{\partial E_k}{\partial r_k}\frac{\partial r_k}{\partial \delta_k} . \tag{3.16}$$

Also similarly in [89], replacing $\partial r_k / \partial \delta_k$ by $sign(\partial r_k / \partial \delta_k) = -1$ yields

$$\frac{\partial E_k}{\partial \psi_k} = \frac{\partial E_k}{\partial \psi_k} - \frac{\partial E_k}{\partial \delta_k} - \frac{\partial E_k}{\partial r_k} = -(\rho e_k + \lambda \delta_k + \sigma r_k) . \tag{3.17}$$

Now (3.2) is replaced by (3.17), and (3.14) can be rewritten as

$$\dot{w}_{ij} = \gamma \cdot sig(I_j) \cdot (\rho e_k + \lambda \delta_k + \sigma r_k) = \gamma \cdot O_j \cdot (\rho e_k + \lambda \delta_k + \sigma r_k) . \tag{3.18}$$

To summary, the ANNAI-based heading control system has the adaptation law for the hidden layer weights and output layer weights as described in equations (3.9) and (3.18) respectively.

3.3 Simulation Results

In this subchapter, computer simulations for course-keeping and track-keeping control performance of the proposed NNC are undertaken. In these simulations, the effects of random measurement noise and wind disturbances are considered to test the reliability and the robustness of the NNC.

To compare with the proposed ANNAI-based heading control system, simulations of BPNN-based heading control system of previous studies are also shown with the same number of training iterations and the effects of measurement noise and wind disturbances. Additionally, the algorithm for automatic adapting NN parameters is applied to the ANNAI-based heading control system.

The NNC is designed under the assumption that an accurate measurement of the ship's state (heading, position, yaw rate) is available on board. With the availability of the general and additional navigational aids such as gyrocompass or satellite compass, rate gyro, and GPS/DGPS receiver, accurate measurement of the ship's state is possible. In this study, a mathematical ship model is used to provide the ship's state and to verify the performance of the controllers. The ship model used in this study is a realistic model of a *Mariner Class Vessel*. The planar motion mechanism tests and full-scale

steering and maneuvering predictions for this *Mariner Class Vessel* were performed by the hydro-aerodynamics laboratory in Lyngby, Denmark (see Appendix B.1 for more details). To be able to do turning control and cope with large steps of set courses, a reference model that reflects the dynamics of the vessel is used to produce a feasible desired course [79]. The simulations are carried out using the MATLAB© 7.0.

In the previous study we showed that the proposed ANNAI-based heading control system needs much less iteration for training than BPNN-based autopilot does [57]. This significantly reduces calculation time of the NNC, which is important in digital controller design. Many simulation works have been carried out to verify the ANNAI-based heading control system to select the proper n and γ to achieve the best performance. Also in the previous study, we selected the initial weights with opposite signs in the hidden neurons as suggested in [23], and activation function of the output neuron was sigmoid and linear gain. But in the following simulation we select the initial weights as rather small random values and good adaptation does occur.

Firstly in this subchapter, an ANNAI-based heading control system is simulated in the case when the activation function of the output neuron is tangent sigmoid with fixed values of n and γ. And next, the adaptation strategy of n and γ is used in the proposed heading control system to show its effectiveness and improvement. In order to test the robust of the ANNAI-based heading control system, wind disturbance and measurement noise are used. The effect wind disturbance against the body of the ship is based on the work of Isherwood (1972) introduced in [79], with wind speed changes randomly every 5 s and assumes values between 15 and 25 knots, relative wind direction varies randomly between [-60^o, +60^o] every 30 s. A random uniformly distributed signal on [-0.1^o, +0.1^o] is used as the sensor noise in the heading sensor.

The constraints in the actuator are $\delta \in$[-35^o, 35^o] and $\dot{\delta} \in$[-2.5^o/s, 2.5^o/s]. Firstly, the desired course against 0^o is 20^o from 0s to 300s, then -20^o from 300s to 600s, and finally 20^o from 600s to 900s. Next, the desired course is 30^o from 0s to 300s, then -30^o from 300s to 600s, and finally 30^o from 600s to 900s. These rather large steps in course changing are for testing turning control performance. In all simulation works, the initial speed is 15 knots (or 7.7175 m/s). The parameters for the automatic selection algorithm

of n and γ are selected as follows

$$[\alpha, \beta, n_{\max}, n_{\min}, \gamma_{\max}, \gamma_{\min}] = [0.105, 1, 150, 30, 2.5, 0.5].$$

The BPNN-based heading control system based on [89] and [90] is used for comparison. A set of performance indices is also defined to provide a numerical comparison

$$E_{\psi} = \sum_{k} (\psi_k^d - \psi_k)^2 , \tag{2.24}$$

$$E_{\delta} = \sum_{k} (\delta_k - \delta_{k-1})^2 , \tag{2.25}$$

where, E_{ψ} is the squared amplitude of the heading error, E_{δ} is the squared variation in rudder adjustment.

3.3.1 Fixed Values of n and γ

(1) Course change from -20° to +20°

In Fig. 3.3 and Fig. 3.4, the learning rate and number of training iterations are fixed (n = 50, γ = 1 for ANNAI and γ = 0.25 for BPNN). The ANNAI and BPNN based heading control systems have shown good performance with and without noise and disturbances. These simulations show the feasibility and effectiveness of the proposed ANNAI-based heading control system. However, as shown in [57], if n or/and γ is increased, the large overshoot in heading and oscillations in rudder will occur due to exceed of training. Thus, pre-tests are necessary here.

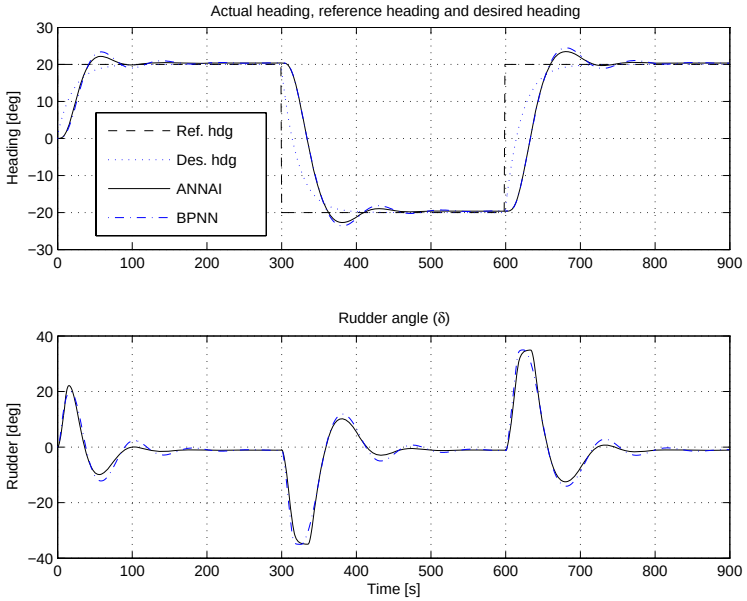


Fig. 3.3 Simulations of ANNAI and BPNN based heading control system without wind and noise, course change from -20^o to +20^o. ANNAI : n = 50, γ = 1, ρ = 1, λ = σ = 0.2. BPNN : n =50, γ = 0.25, ρ = 1.5, λ = σ = 0.1

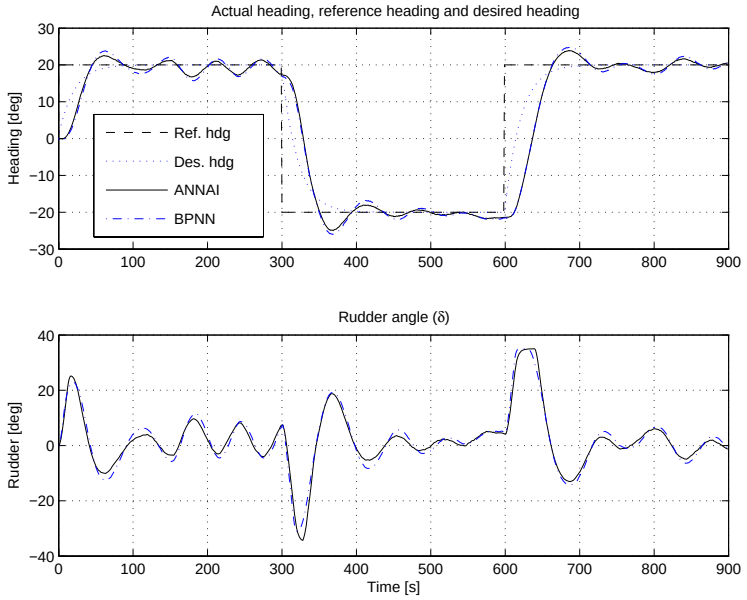


Fig. 3.4 Simulations of ANNAI and BPNN based heading control system with wind and noise, course change from -20^o to +20^o. ANNAI : n = 50, γ = 1, ρ = 1, λ = σ = 0.2. BPNN : n =50, γ = 0.25, ρ = 1.5, λ = σ = 0.1

(2) Course change from -30^o to +30^o

Simulations in Fig. 3.5 and Fig. 3.6 showed good performance of both heading control systems in case the course change is from -30^o to +30^o, with and without noise and disturbances. From Fig. 3.3 – 3.6, better course-keeping, smaller overshoot and less rudder efforts of ANNAI-based heading control system in comparison with BPNN-based heading control system are observed.

In Table 3.1, the numerical comparisons of the two autopilots in Fig. 3.3 – 3.6 are shown. These numerical results show that, E_ψ of ANNAI-based heading control system is smaller than that of BPNN-based heading control system with almost same E_δ.

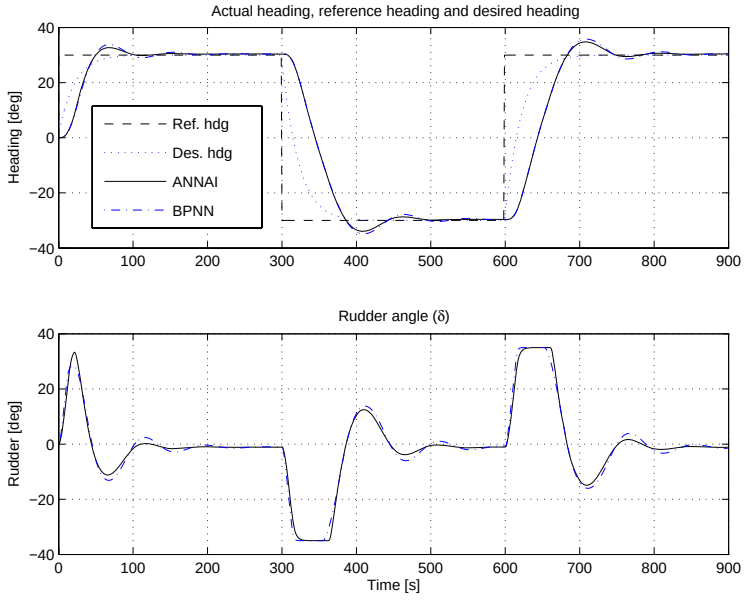


Fig. 3.5 Simulations of ANNAI and BPNN based heading control system without wind and noise, course change from -30^o to +30^o. ANNAI : n = 50, γ = 1, ρ = 1, λ = σ = 0.2. BPNN : n =50, γ = 0.25, ρ = 1.5, λ = σ = 0.1

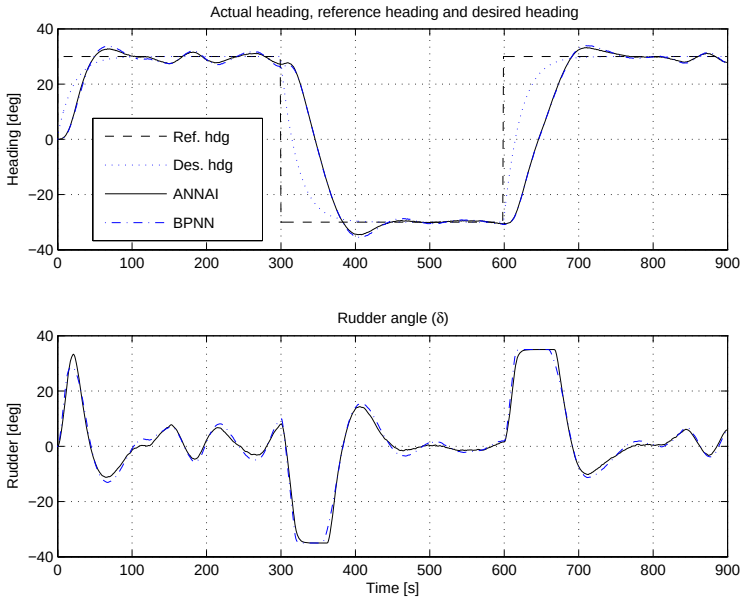


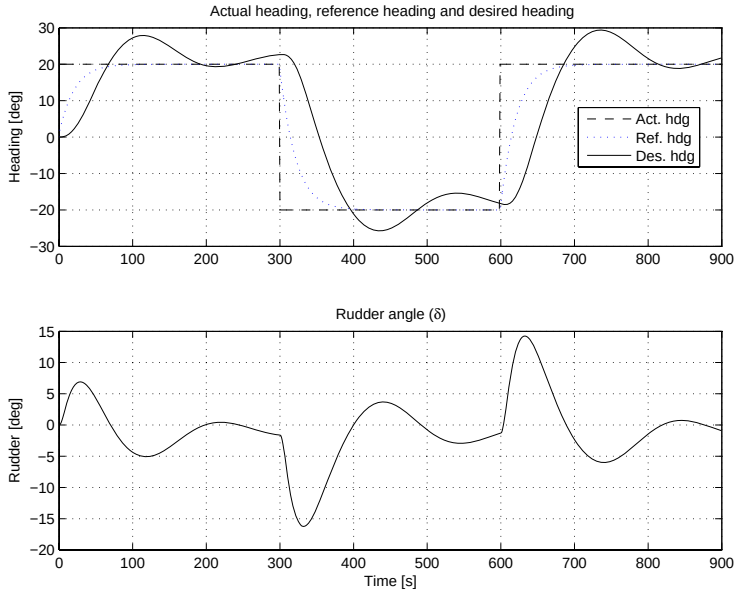
Fig. 3.6 Simulations of ANNAI and BPNN based heading control system with wind and noise, course change from -30^o to +30^o. ANNAI : n = 50, γ = 1, ρ = 1, λ = σ = 0.2. BPNN : n =50, γ = 0.25, ρ = 1.5, λ = σ = 0.1

Table 3.1 Comparison performance indices

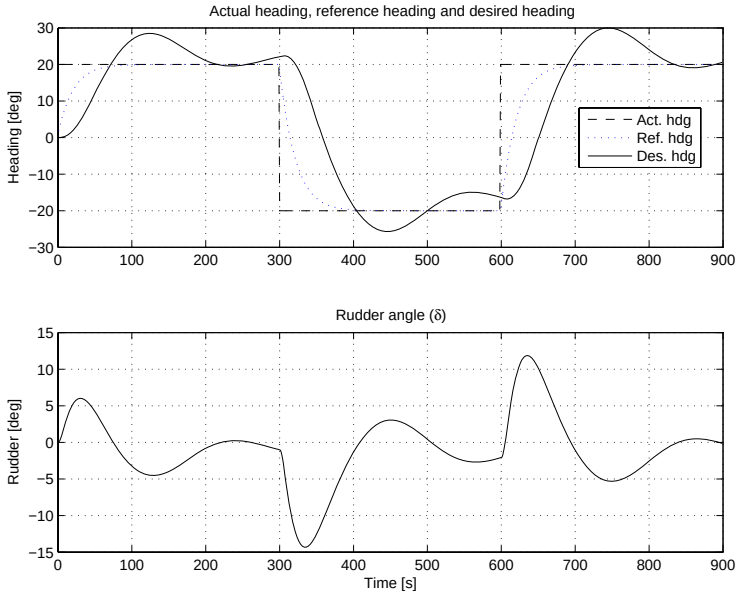
	Fig. 3.3		Fig. 3.4		Fig. 3.5		Fig. 3.6	
	ANNAI	BPNN	ANNAI	BPNN	ANNAI	BPNN	ANNAI	BPNN
E_ψ	18704	19822	19726	20822	73040	74429	79910	81214
E_δ	346	351	459	465	404	402	451	453

(3) ANNAI-based heading control system with improper initial parameters

In Fig. 3.7 the simulations have been carried out with the improper value of learning rate γ = 0.1 (Fig. 3.7a) and improper number of training iterations n = 5 (Fig. 3.7b) for the ANNAI-based heading control system. The adaptation is poor even no wind and noise applied and course change is from -20^o to +20^o. Actually, many pre-tests have been done to select proper value of learning rate and number of training iterations in order to achieve the good performance described in Figs. 3.3 - 3.6.



(a) Simulation with n = 50, γ = 0.1, ρ = 1, λ = σ = 0.2



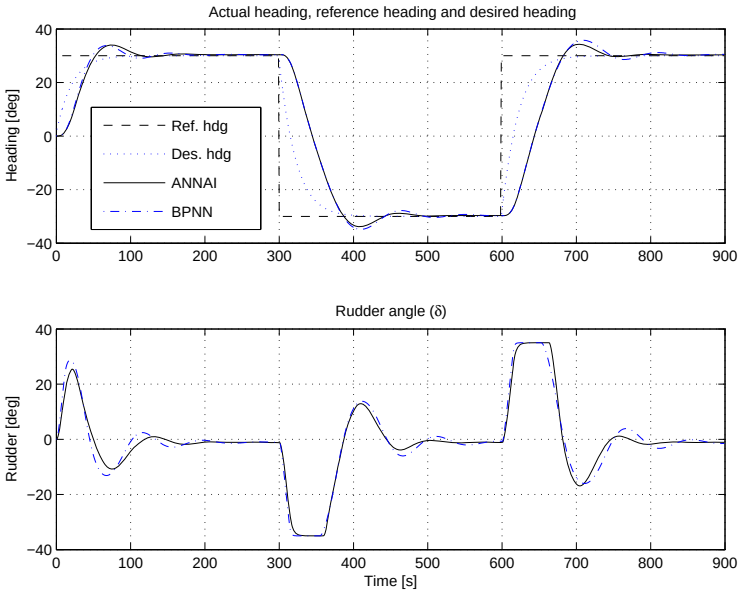
(b) Simulation with n = 5, γ = 1, ρ = 1, λ = σ = 0.2

Fig. 3.7 Simulations of ANNAI-based heading control system with improper values of learning rate (a); number of training iterations (b)

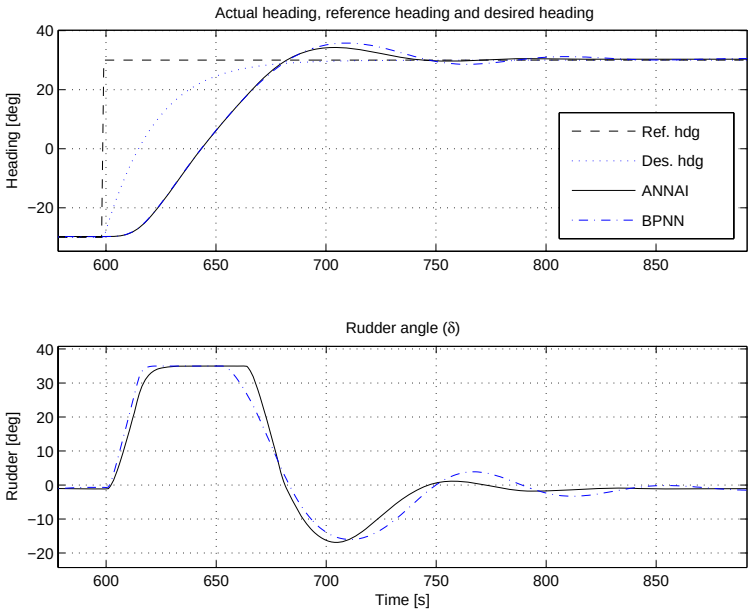
3.3.2 With adaptation of *n* and γ

To improve the proposed ANNAI-based heading control system performance and remove the time-consumed manual selections of n and γ, an automatic adaptation algorithm for these parameters is proposed. Computer simulations are shown in Fig. 3.8 and Fig. 3.9. In these simulations, no pre-tests are necessary and we try to use improper initial values of n and γ (n = 5, γ = 0.01) but they do not degrade the adaptation and performance of the NNC. Because both n and γ are dynamically updated at every control cycle. The small average values of n are also observed. The poor performance shown in Fig. 3.7 has been overcome.

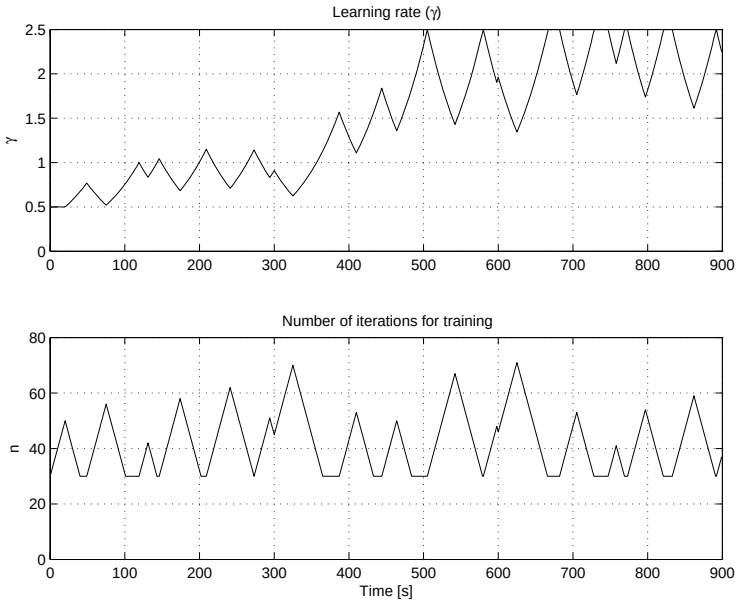
Fig. 3.8 is the simulation result of ANNAI and BPNN based heading control system in case of no noise and wind applied and course change from -30^o to +30^o. In Fig. 3.9, the effects of measurement noise and wind disturbances are included. These simulations show good adaptation ability of the heading control system when coping with large change of reference course and the robustness are maintained through time. We do not need to adjust the NNC learning rate and number of training iterations manually as they can be automatically selected.



(a) Heading and rudder angle

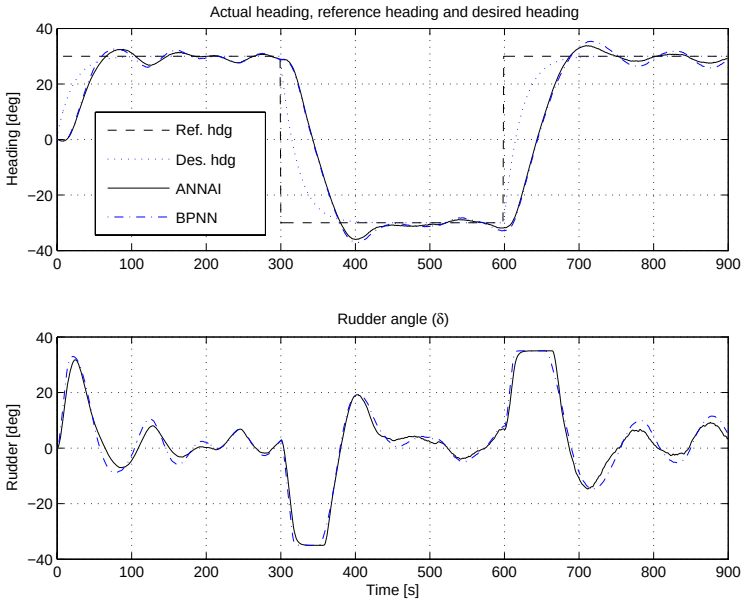


(b) Zoom in of course change performance

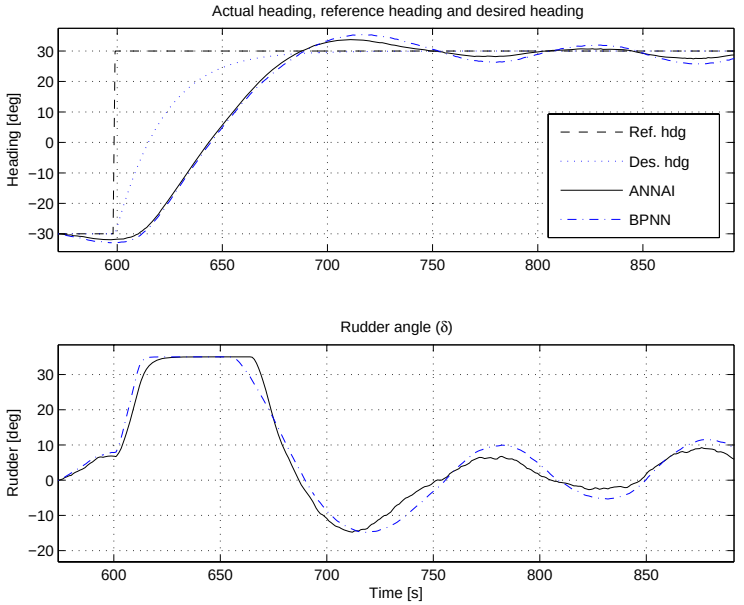


(c) Learning rate and number of training iterations

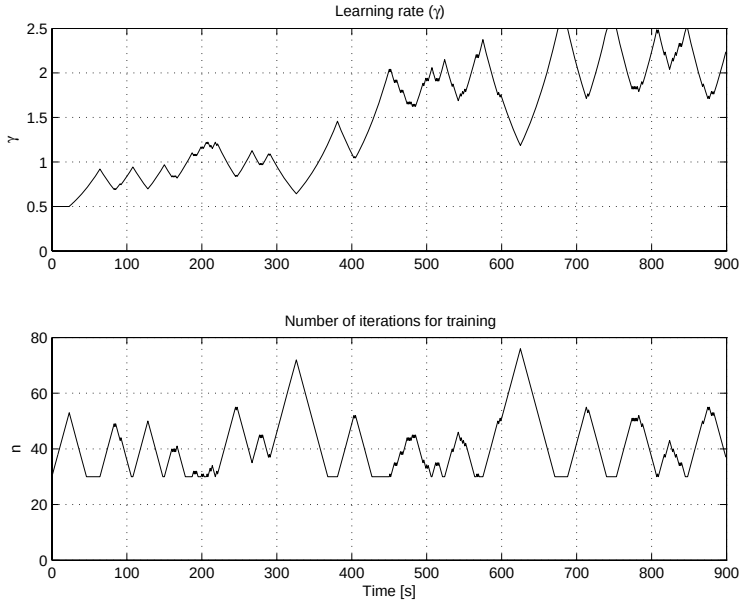
Fig. 3.8 Simulations of ANNAI and BPNN based heading control system with initial n = 5, initial γ = 0.01; ρ = 1, λ = σ = 0.2, no wind and noise, course change from -30^o to +30^o



(a) Heading and rudder angle



(b) Zoom in of course change performance



(c) Learning rate and number of training iterations

Fig. 3.9 Simulations of ANNAI and BPNN based heading control system with initial n = 5, initial γ = 0.01; ρ = 1, λ = σ = 0.2, with wind and noise, course change from -30^o to +30^o

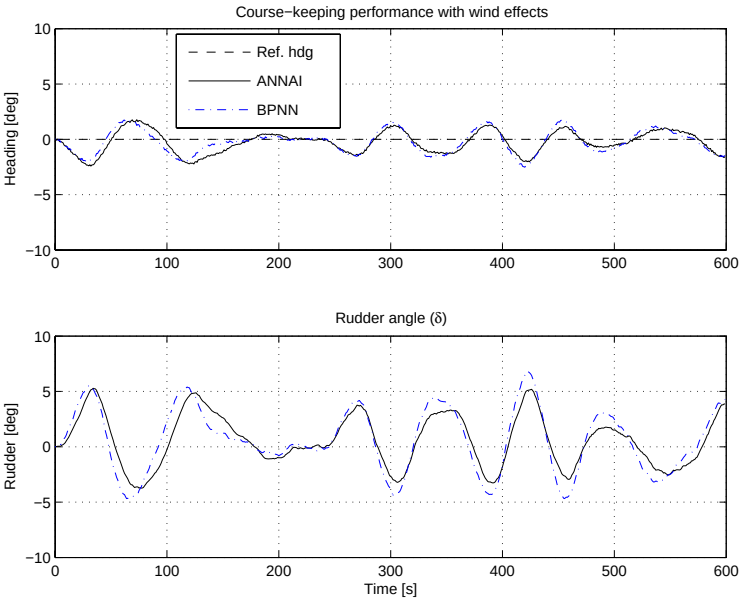


Fig. 3.10 Course-keeping performance of ANNAI and BPNN based heading control systems

In Fig. 3.10, we test the course-keeping performance of both heading control systems under the effects of strong wind. The wind force increases to the extent from 20 to 30 knots. The set course is 0^o and simulation time is 600 s. The ANNAI-based heading control system uses less rudder movement than the BPNN-based heading control system does, but shows a better course-keeping performance. This comparison can be confirmed by numerical results of $E_\psi = 628$, $E_\delta = 20$ for ANNAI-based heading control system; $E_\psi = 660$, $E_\delta = 33$ for BPNN-based heading control system.

Fig. 3.11 shows the training process of both ANNAI and BPNN within one control cycle. The squared error of the NNC output is reduced iteration by iteration. This example is taken from control cycle indicated by k = 30 s in the simulation of Fig. 3.8. In one control cycle, the squared error at the first iteration where training process starts is the maximum error. Within one control cycle, NNCs require a number of iterations to reduce this squared error to the minimum value. Fig. 3.12 and Fig. 3.13 show the adaptation process of connection weight in output layer and hidden layer, respectively, for both ANNAI and BPNN controllers. During control process, the weights of ANNAI have a faster convergence and smoother transient phase than those of BPNN. In Fig. 3.14 we can see a better minimization process for the cost function of ANNAI controller in comparison with that of BPNN controller.

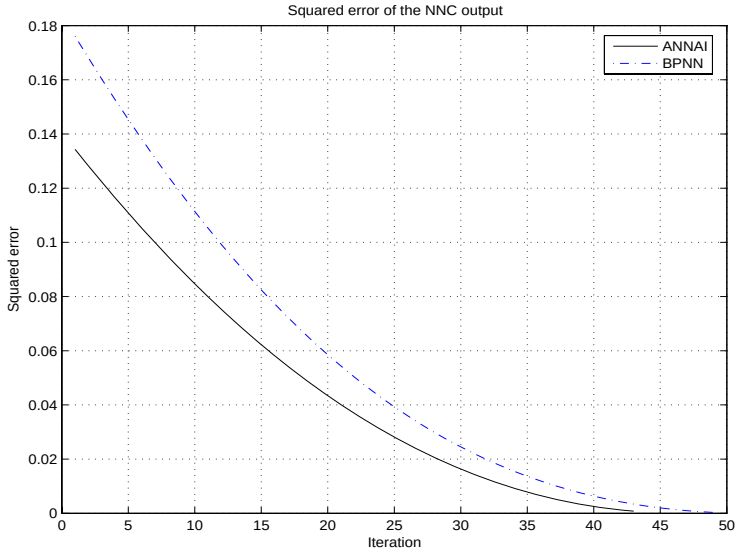


Fig. 3.11 Training process of ANNAI and BPNN within one control cycle at k = 30s

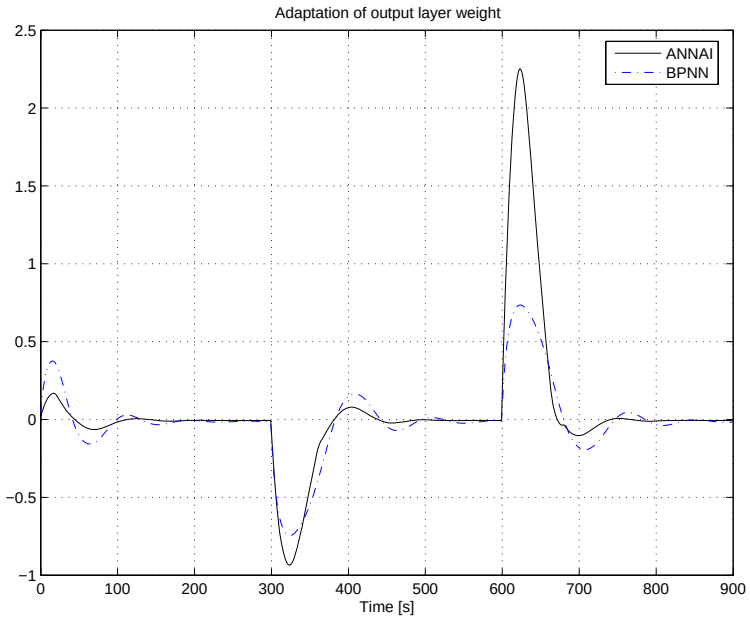


Fig. 3.12 Adaptation process of output layer weight of ANNAI and BPNN

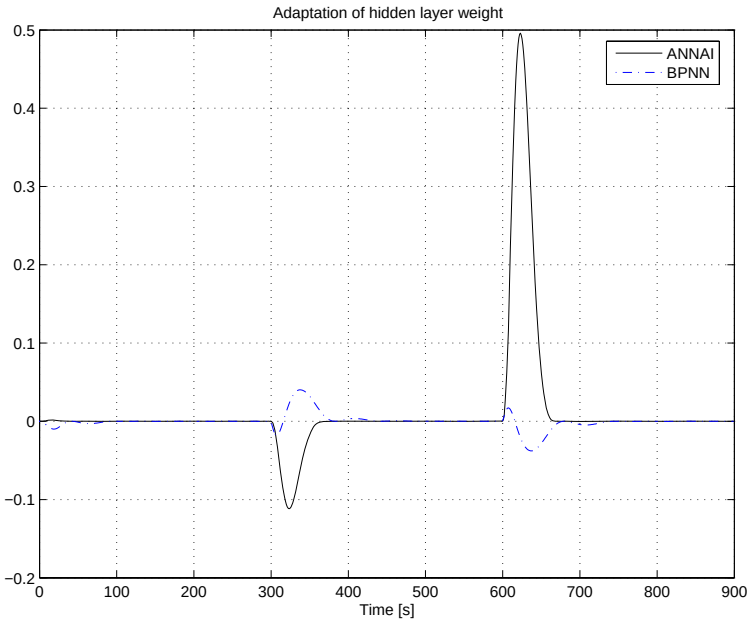


Fig. 3.13 Adaptation process of hidden layer weight of ANNAI and BPNN

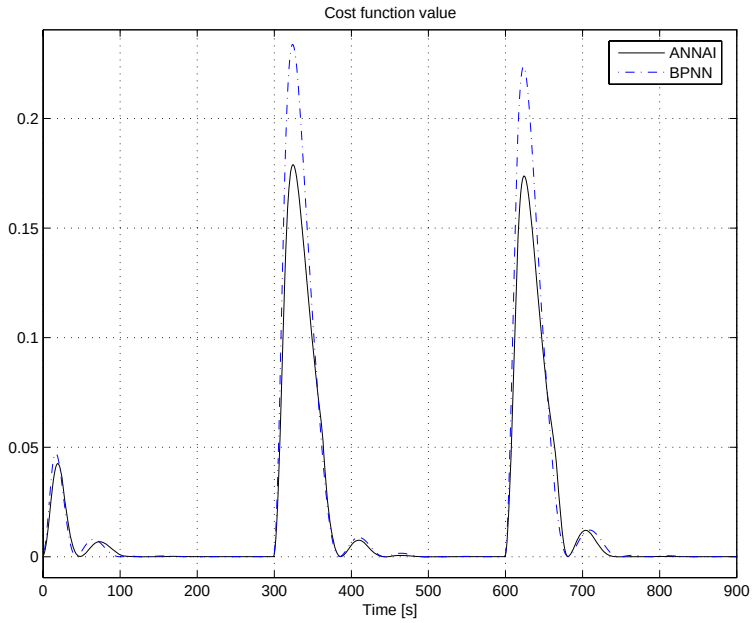


Fig. 3.14 Cost function value of ANNAI and BPNN

3.4 Conclusion

This chapter presented an application of NN control to automatic course-keeping and turning control of ships. The new approach of adaptive NN controller developed in chapter 2 was applied to heading control system for ships. Various computer simulations were undertaken to validate the proposed ANNAI-based heading control system. The obtained results lead to the following conclusions which are the advantages of the proposed NNC

(a) It can work well with good performance when coping with non-linear and time-varying characteristics of the ship.

(b) It is not necessary to use the ship model parameters in designing the controller, the error in ship model can be avoided.

(c) Its parameters can be dynamically updated to ensure the robustness

through time and speed up adaptation process while maintaining sufficient training.

(d) The on-line training ability can help to cope with new situations, including different ships or environmental conditions.

(e) The proposed NNC is also stable as all its parameters are updated at every control cycle.

(f) It is not very sensitive with measurement noise of input signals.

(g) The automatic selection algorithm for learning rate and number of iterations worked well to maintain the stability of the control system.

The NNC can adapt directly without approximating the ship dynamics by a NN. This not only eliminates the error in approximation, but also significantly reduces the complexity of design. Furthermore, the proposed ANNAI can adapt faster than BPNN and its configuration is simpler. With the proposed algorithm for automatic adaptation of learning rate and number of training iteration, the adaptation of NNC can be improved and manual time-consuming selection of the NNC parameters is removed.

The proposed NNC can be applied to other types of ship and more complicated control problems because of its adaptation ability. To improve the performance, it might be used in combination with other techniques and theory such as fuzzy control. These will be further research topics of the authors.

Chapter 4 ANNAI-based Track Control System

4.1 Introduction

Since late 1970s, track control system has been studied. Advanced control theories have been applied to seek optimal rudder control to track a predetermined path. The transformation of the way-points to a feasible path or trajectory is in general a nonlinear optimization problem [79]. The controller can be design using linear or nonlinear theory. Extensions to nonlinear trajectory tracking and maneuvering control are currently new fields of research. More recent studies can be seen in [1], [11], [16], [37], [44], [47] - [49], [76], and [77].

Recently, intelligent controllers design for track control of ship has been paid attention by many researchers, such as in [9], [89], [90], [91] – [94]. To apply neural networks in track control system, most authors used “off-line learning”. The training methods mostly base on knowledge of the way conventional controllers operate on a given ship. On the other hand, [89] and [90] proposed a neural network controller, which used an online learning method for track control problem. In this single-input multi-output (SIMO) control law, rudder is used to minimize both tracking error and heading error.

In our previous studies [58] and [59], the ANNAI controller (chapter 2) whose parameters can be automatically adapted was proposed for course-keeping and track-keeping control. The ANNAI controller combined with Line-of-Sight (LOS) guidance algorithm (see [79] for more details about LOS algorithm) showed good performance without the influence of sea current acting on ship. But in practice, external disturbances, especially sea currents make the ship deviate from the intended track. To

improve the performance of track control system, the lateral off-track distance from the intended track will be considered.

In this chapter, the improvement of the ANNAI-based track control system is conducted by considering the lateral off-track distance from the intended track in the learning process of the ANNAI controller. A new solution is proposed for track control by combining the modified LOS guidance algorithm with such an improved ANNAI controller. Thus both the advantage of LOS guidance algorithm and the adaptability of the ANNAI controller are exploited. Using this solution the control problem becomes SIMO control similar in [89] and [90], in which the rudder is used to minimize both the lateral off-track distance and the heading error.

For simulating the track control performance, a mathematical ship model is used. MATLAB modules built for guidance system and displaying the movement of ship on Mercator projection chart are proposed. For visually displaying, M-Maps toolbox for MATLAB is applied.

4.2 Track Control System

The main results of this research were already published in [60], where the ANNAI-based track control system proposed in [58] was reviewed and the improved ANNAI-based track control system was introduced. This subchapter presents new results of the ANNAI-based track control system.

The NNC is a multilayer feedforward neural network with one hidden layer. The configuration of the NNC is shown in Fig. 4.1 where w_{ij} is used to indicate the weights between output layer and hidden layer, w_{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 NNC are merely heading error and its time-delayed values. ψ_k^d and ψ_k are desired heading and actual heading respectively. $O_{i=1} = \delta_k^c$ is output of NNC or rudder command, I_i is the summation of the weighted inputs to the units in the output layer plus θ_i, where θ_i is the

threshold value of the output layer neurons. The intended track consists of waypoints and the straight lines connecting the waypoints from departure to arrival. LOS guidance was presented in [79] and applied to surface ships in previous works such as [58], [91] and a modified version in [6]. This chapter also proposes another modified version of the LOS algorithm.

The configuration of the ANNAI-based track control system is shown in Fig. 4.1. The LOS guidance system produces guidance heading signals for ship to follow and make the ship position converge to the predefined track. Off-track distance is added to learning process of the NNC to improve the adaptability of the ANNAI-based track control system with the effects of external disturbances.

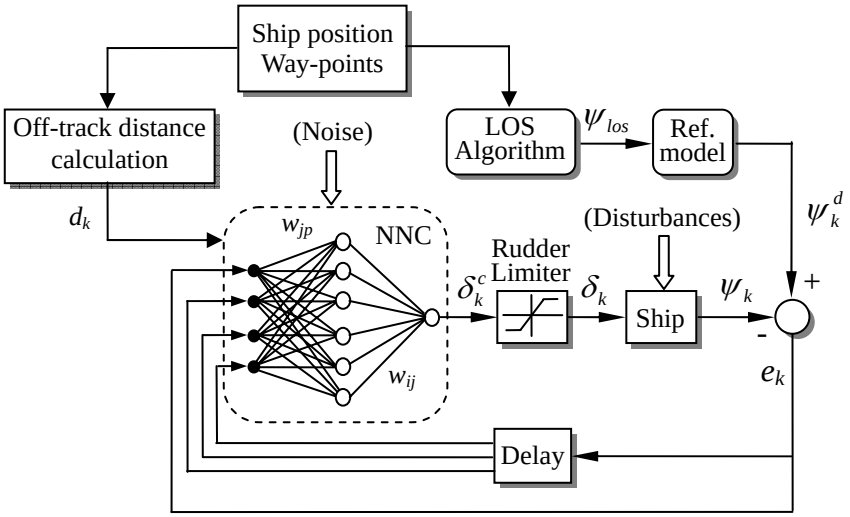


Fig. 4.1 ANNAI-based track control system using ANNAI controller and modified LOS guidance algorithm; off-track distance is considered

(1) Effect of Disturbances on Ship's Track

In navigational practice, precise track control performance of a ship is always affected by external disturbances such as sea current and wind. If those effects are considerable then the accurate track control of ship is deteriorated, especially in the case of navigation in restricted waters. In such situations the conventional LOS guidance algorithm can not help to perform good track control ability [92].

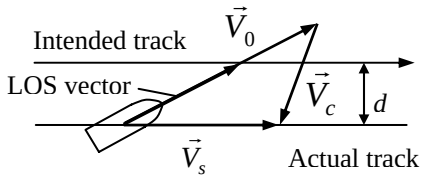


Fig. 4.2 Track control using LOS guidance under influence of sea current

Fig. 4.2 shows LOS guidance algorithm performance for track control under the influence of sea current. If the following equality holds

$$\vec{V}_s = \vec{V}_0 + \vec{V}_c , \tag{4.1}$$

where, $\vec{V}_s$, $\vec{V}_0$, and $\vec{V}_c$ are ship's actual speed, speed of advance and current speed, respectively, and $\vec{V}_0$ has same direction with LOS vector, then actual track of the ship is clear off the intended track a certain distance *d*.

(2) ANNAI-based Track Control System with Off-track Distance Included

In [89] a SIMO adaptive NNC for track control using BP training method was introduced. The rudder is used to minimize both the lateral off-track distance and the heading error which is tangent with desired path.

Later, [32] presented an adaptive fuzzy autopilot for track control. In this study, the lateral off-track distance was used as an input to "*Adjustable scaling factor mechanism*" which provided the fuzzy autopilot the adaptability with external disturbances.

In [91] an intelligent track control system for ship with specialized learning using neurofuzzy as SIMO controller was introduced.

To improve the tracking control ability, in this research the off-track distance (see Fig. 4.2) is considered to design a SIMO track control system based on the ANNAI

controller developed in chapter 2. The output of the ANNAI controller (rudder command) is calculated to minimize both the lateral off-track distance caused by external disturbances and the error between actual ship heading and LOS guidance signal.

The LOS guidance signal (ψ_{LOS}) is calculated by LOSMERCATOR module (see 4.3.1) which is described in Fig. 4.3. Let A(l_{k-1}, L_{k-1}), B(l_k, L_k) be previous and current waypoint positions in latitude and longitude respectively. N is intersection point of LOS vector and AB. The middle latitude (see [86]) as follows in (4.2)

$$l_m = \frac{1}{2}(l_{k-1} + l_k)\,. \tag{4.2}$$

In Fig. 4.3, S is ship position (l_{ship}, L_{ship}) and

$$\mathrm{SN} = nL\,, \tag{4.3}$$

where n is number of ship length L.

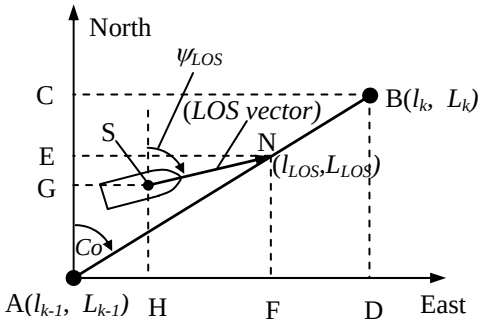


Fig. 4.3 Calculation of LOS guidance signal

$$\mathrm{AC} = \mathrm{BD} = \Delta l = l_k - l_{k-1}\,, \tag{4.4}$$

$$\mathrm{AD} = \mathrm{BC} = \Delta L = (L_k - L_{k-1})\cos l_m\,, \tag{4.5}$$

$$\mathrm{AG} = l_{ship} - l_{k-1}\,, \tag{4.6}$$

$$\text{AH} = (L_{ship} - L_{k-1})\cos l_m . \tag{4.7}$$

Using MECATOR or MIDLAT module introduced later in 4.3.1, we can determine the course *Co* from A to B and distance AB. Similar in [79], NF and NE can now be determined. Position of N in latitude and longitude can be calculated as

$$l_{LOS} = l_{k-1} + NF , \tag{4.8}$$

$$L_{LOS} = L_{k-1} + AF / \cos l_m . \tag{4.9}$$

From S(l_{ship}, L_{ship}), N(l_{LOS}, L_{LOS}), and using MECATOR or MIDLAT module, we can calculate ψ_{LOS}.

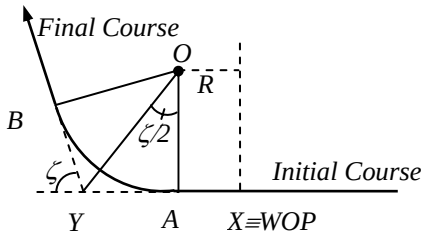


Fig. 4.4 Wheel-Over-Point and Reach while changing course

In the LOS algorithm, *wheel over point* (WOP) is the point at which rudder angler must be applied to achieve the required course alteration. To determine WOP the study in [79] introduced fixed value of the radius of circle of acceptance R_0 in the LOS algorithm. In [92], R_0 was calculated using table look up method where the value of R_0 at each course changing situation obtained from ship model testing. Alternatively, in this study R_0 is determined as the distance from waypoint Y to A (starting point of turning circle) plus *Reach* (*R*) as expressed in (4.10) and shown in Fig. 4.4 (see [86]). *R* is the advanced distance from WOP to A and depends on the condition of ship and the rudder angle. Value of R_0 is determined as follow

$$R_0 = XY = YA + R = 5\zeta + R . \tag{4.10}$$

where $YA \cong 5\zeta$ depends on the course change angle ζ. Equation (4.10) is included in simulation program for automatic calculation of R_0. For further details of determination of R_0, see [86].

We have discussed the modified LOS algorithm. Now let's see how the off-track distance is employed. In this research, the cost function is modified from that of [58]. Using the method introduced in [93], we can determine the off-track distance *d* and its sign. We also define normalized off-track distance as $\mu = d/L$, where *L* is the length of ship. The cost function in [58] is modified as follow

$$E_k = \frac{1}{2}\left[\rho_1(\psi_k^d - \psi_k)^2 + \rho_2(\mu_k^d - \mu_k)^2 + \lambda\delta_k^2 + \sigma r_k^2\right]. \tag{4.11}$$

Similar in [58], it is possible to write

$$\frac{\partial E_k}{\partial \psi_k} = \frac{\partial E_k}{\partial \psi_k} - \frac{\partial E_k}{\partial \delta_k} + 2\frac{\partial E_k}{\partial r_k}\frac{\partial r_k}{\partial \delta_k} - \frac{\partial E_k}{\partial \mu_k}\frac{\partial \mu_k}{\partial \delta_k}. \tag{4.12}$$

From (4.12), by replacing $\partial r_k / \partial \delta_k$ and $\partial \mu_k / \partial \delta_k$ with $sign(\partial r_k / \partial \delta_k) = -1$ and $sign(\partial \mu_k / \partial \delta_k) = -1$ respectively, we can obtain

$$\frac{\partial E_k}{\partial \psi_k} = -\rho_1(\psi_k^d - \psi_k) - \rho_2(\mu_k^d - \mu_k)sign(\frac{\partial \mu_k}{\partial \delta_k}) - \lambda\delta_k + 2\sigma r_k sign(\frac{\partial r_k}{\partial \delta_k}), \tag{4.13}$$

$$\frac{\partial E_k}{\partial \psi_k} = -[\rho_1(\psi_k^d - \psi_k) - \rho_2(\mu_k^d - \mu_k) + \lambda\delta_k + 2\sigma r_k]. \tag{4.14}$$

Note that the desired value $\mu_k^d = 0$, hence (3.18) can be modified as

$$\dot{w}_{ij} = \gamma \cdot sig(I_j) \cdot (\rho_1 e_k + \rho_2 \mu_k + \lambda\delta_k + \sigma r_k) = \gamma \cdot O_j \cdot (\rho_1 e_k + \rho_2 \mu_k + \lambda\delta_k + \sigma r_k). \tag{4.15}$$

Depending on the value μ_k, we select ρ_2 as follows

$$\rho_2 = a_1 + a_2 \cdot |\mu_k|, \tag{4.16}$$

where a_1 and a_2 are positive constants. Equation (4.16) can improve the adaptability of the autopilot since the ANNAI is more sensitive to the off-track distance.

The adaptation law for hidden layer weights similar to (3.9) is as follows

$$\dot{w}_{jp} = O_p[\phi_j sig(-I_j) + \gamma \cdot 0] = O_p \phi_j sig(-I_j), \tag{4.17}$$

where

$$\phi_j = w_{ij} \cdot \dot{w}_{ij}, \tag{4.18}$$

$$sig(I_j) = \frac{1}{1 + \exp(-I_j)} \tag{4.19}$$

To summarize, the ANNAI controller has the adaptation law for the hidden layer weights and output layer weights as described in (4.17) and (4.15) respectively. Based on the above ANNAI controller and modified LOS algorithm, we propose a track control system as shown in Fig. 4.1.

4.3 Simulation Results

4.3.1 Modules for Guidance Using MATLAB

In this study the modules written in MATLAB are introduced for guidance and control using Mercator chart. They can be used to calculate and display ship's movement on the navigational equipment monitors. The followings are brief descriptions of the module programs:

LOSMERCATOR calculates LOS guidance signal using Mercator formula [86].

MERCATOR calculates distance and course between two points on Mercator chart using Mercator formula.

MIDLAT calculates distance and course between two points on Mercator chart using middle-latitude formula [86].

NEXTPOS calculates next position from one position if distance and course between the two positions are known. This calculation is based on middle latitude method.

OFFTRACKDIST calculates off-track distance from ship position to desired track in way-point navigation.

SHIPICON returns vector of points for drawing ship icon on Mercator chart for moving animation of ship motion. It can use real dimensions of ship.

WOP returns distance from current way-point to the respective Wheel-Over-Point at which ship turns to new course.

4.3.2 M-Maps Toolbox for MATLAB

M-Maps Toolbox, which is a set of mapping tools written in MATLAB and is available in [64], is used for visual simulation of ship's movement. Fig. 4.5 shows an example of simulation of a ship departing from Pusan bay using M-Maps Toolbox and module programs in subchapter 4.3.1.

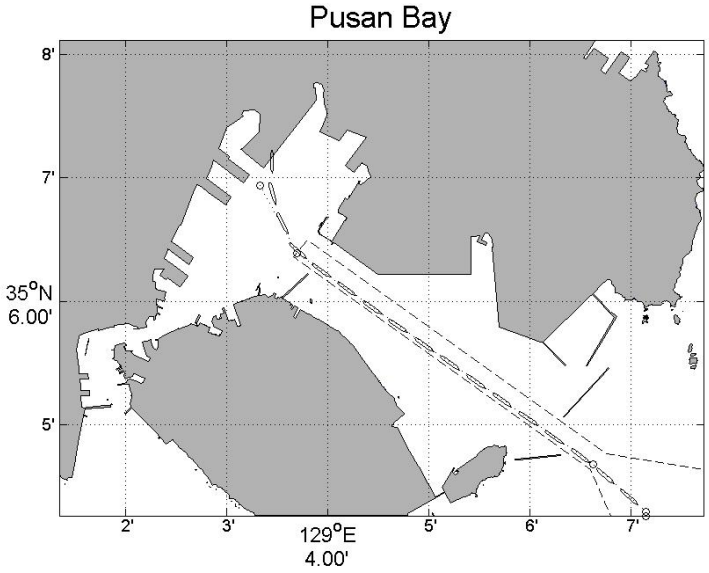


Fig. 4.5 Simulation of a ship departing from Pusan bay

4.3.3 Ship Model

In this research the mathematical ship model is used for simulating and testing the performance of the controller. The simulations are carried out for a Mariner Class Vessel, the nonlinear model of which can be found in GNC Toolbox for MATLAB [78] - [80]. The planar motion mechanism tests and full-scale steering and maneuvering predictions for this Mariner Class Vessel were performed by the hydro-aerodynamics laboratory in Lyngby, Denmark (see Appendix B.1 for more details).

4.3.4 External Disturbances and Noise

Among many possible external disturbances acting on a ship only two will be applied here. They are sea current and wind, which mainly influence the track control performance.

To simulate sea current, the two-dimensional current model described in [79] is used here. Sea current true direction of 220^{o} and velocity varying from 0.5 to 1 m/s is used.

The effect of wind disturbance against the body of ship is based on the work of Isherwood (1972) introduced in [79] with wind speed changes randomly from 10 knots to 20 knots every 5 s, wind direction changes randomly from 0^o to 90^o every 30 s.

A random uniformly distributed signal on [-0.1^o, $+0.1^o$] is used as the sensor noise in the heading sensor.

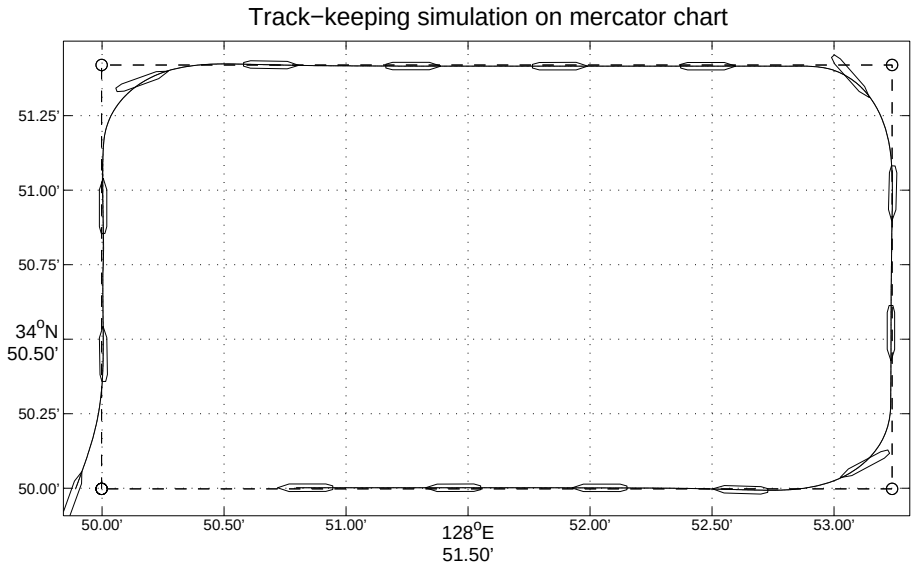
4.3.5 Simulation Results

In this subchapter, simulation results of the ANNAI-based track control system are presented without and with the influence of measurement noise and external disturbances. We select the ANNAI controller with p=4, i=1, and j=6. The input neurons have linear activation functions, the hidden neurons have sigmoidal activation functions, and the output neuron has tangent sigmoidal activation function. The following parameters are used

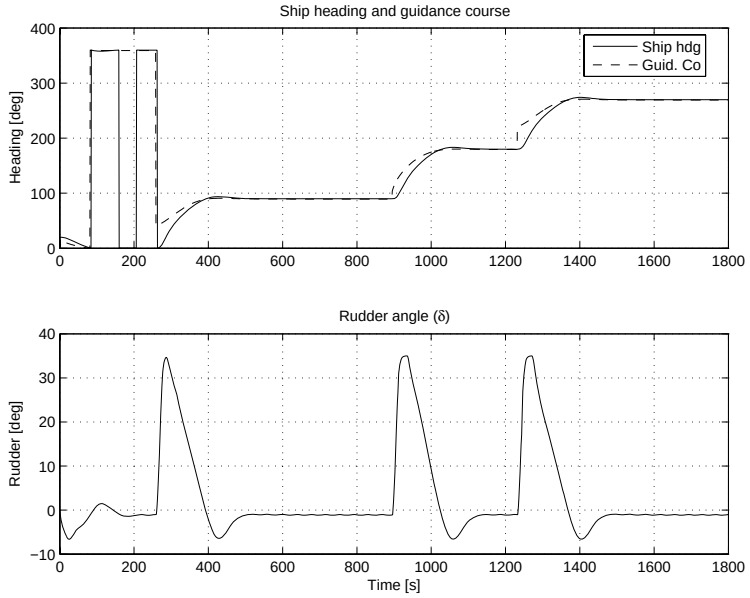
$$[\rho_1, \lambda, \sigma, a_1, a_2] = [1.0, 0.2, 0.2, 0.4, 0.6]. \quad (4.20)$$

The intended track consists of four waypoints (34.8333^oN, 128.8333^oE), (34.857^oN, 128.8333^oE), (34.857^oN, 128.8873^oE), (34.8333^oN, 128.8873^oE). The initial ship position and heading is (34.8333^oN, 128.83153^oE) and 020^o. For each simulation, the ship position on the track is plotted every 120 s.

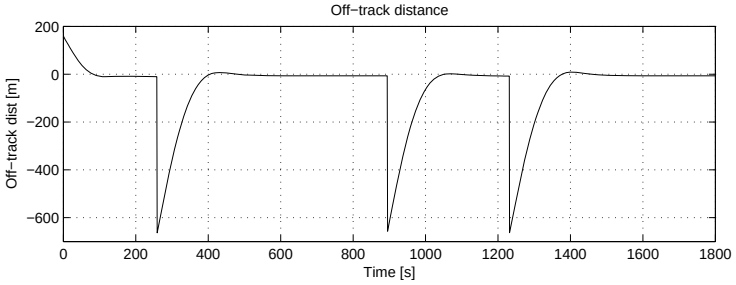
In Fig. 4.6, track control simulation of the ANNAI-based track control system without the influence of disturbances is shown. The intended track is dashed line and the actual track is solid line. Without external disturbances the ship is nicely kept along the intended track (Fig. 4.6a).



(a) Plot of the ship track without disturbances



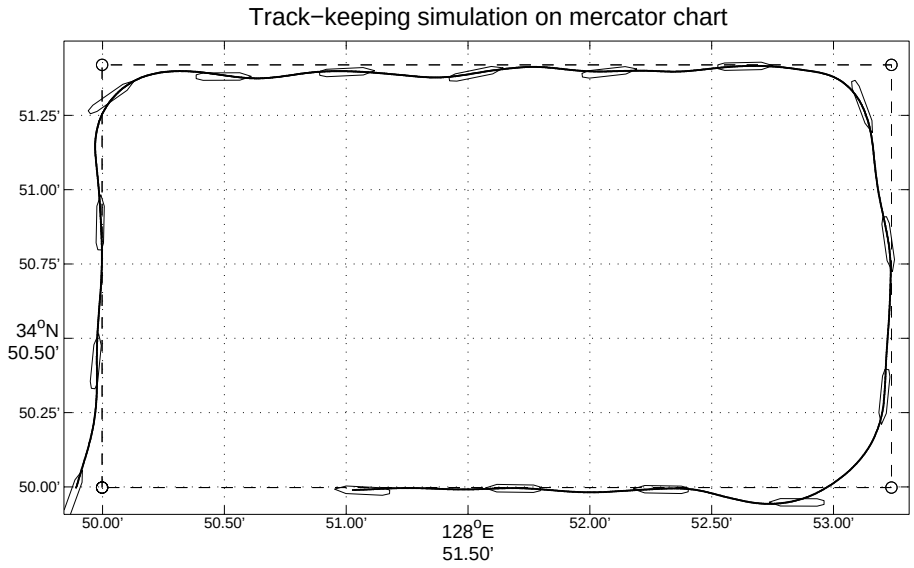
(b) Ship heading and rudder angle.



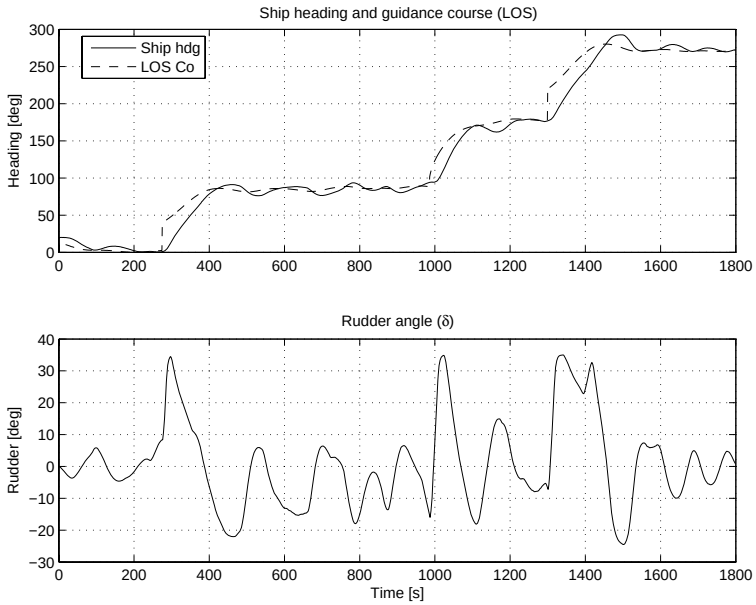
(c) Off-track distance

Fig. 4.6 Track control performance of the ANNAI-based track control system without the influence of disturbances

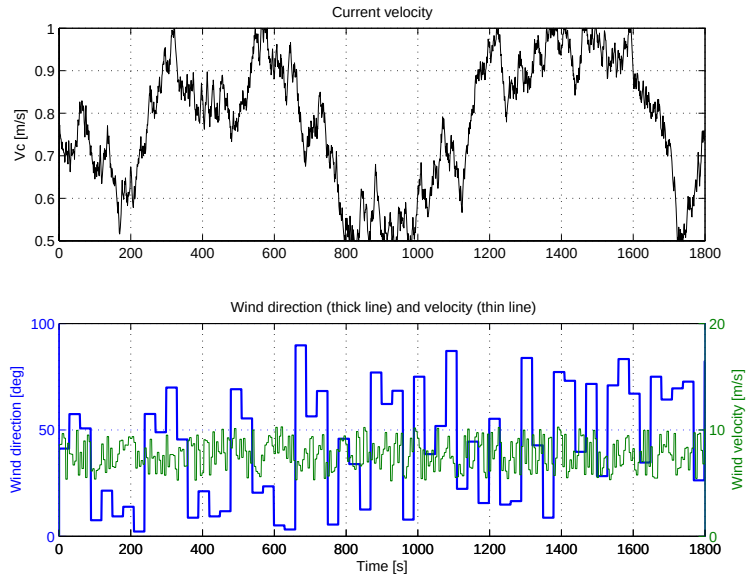
In Fig. 4.7, the simulation of track control performance under the influence of external disturbances is shown. Although affected by the external disturbances, the ship can converge fast to new course and keep close to the intended track (see Fig. 4.7a, d). It is observed in Fig 4.7b that the rudder responded actively against the effects of the external disturbances. This simulation illustrates the positive effect of off-track distance on learning process of the ANNAI controller to cope with external disturbances. More simulation works and comparisons between conventional LOS-based and the proposed ANNAI-based track control system can be found in [60].



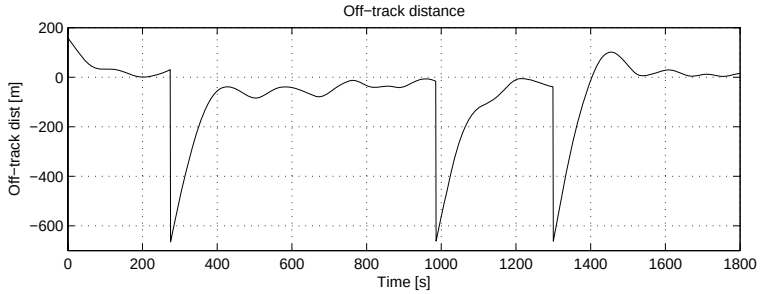
(a) Plot of the ship track with disturbances



(b) Ship heading and rudder angle



(c) Current and wind



(d) Off-track distance

Fig. 4.7 Track control performance of the ANNAI-based track control system with the influence of disturbances

4.4 Conclusion

In this study, an adaptive ANNAI-based track control system for ships is developed. LOS guidance algorithm is applied to calculate guidance course for the track control system. The proposed method has shown good performance under the influence of external disturbances.

Principally, the differences between the proposed adaptive track control scheme and the ones proposed in previous studies are

(a) The method to calculate guidance signal and radius of circle of acceptance R_0 in the LOS algorithm.

(b) The use of normalized off-track distance in learning process of the ANNAI.

(c) The combination of the improved ANNAI controller with modified LOS algorithm.

For simulation purpose, module programs written in MATLAB are introduced for guidance and control using Mercator chart. They can be used to calculate and display

ship's movement on the navigational equipment monitors such as ECDIS (Electronic Chart Display and Information System). M-Maps toolbox for MATLAB is applied for visually displaying ship's movement.

Our purpose is to improve the ANNAI-based track control system employing the learning ability of the ANNAI controller. The new ANNAI-based track control system must have adaptability with external effects. This purpose has been achieved.

Chapter 5 ANNAI-based Berthing Control System

5.1 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 [55]. Therefore, almost recent researches in automatic berthing control tried to employ “intelligent control” that can in some extents mimics human operators. These control techniques include knowledge-based control systems, expert systems, fuzzy logic controllers and neural network-based controllers. NNs have proved to be an effective and attractive option in developing automatic ship berthing controllers.

In [33] a berthing control system using fuzzy neural network was presented. In [89], a multi-variable controller for automatic ship berthing using multi-layer feedforward NN was introduced. This controller used BP 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, [54] 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 the authors presented a motion identification method using NN and its application to automatic ship berthing [55]. In their study, motion identification was used to estimate the effect of environmental disturbances. Off-line training scheme using BP method was also applied with teaching data consisting of 6 sets of automatic berthing simulation. One of the most recent research was presented in [27] for automatic berthing of Kaisho Maru

with PID controller and *Reference Point* guidance.

Recently, in [58] – [60], we have proposed direct adaptive NNC for course-keeping and track-keeping control of ship based on the adaptation algorithm developed by [70] and the extension of the NNC proposed in [23] (see chapter 2).

Employing the advantages of the NNC developed in our previous studies, in this chapter 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 BP network is not required in this kind of NNC so the configuration is simplified and the speed of training is considerably improved [58].

Firstly in this chapter, the ANNAI (developed in chapter 2) 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 shown.

5.2 Berthing Control System

In this chapter an automatic berthing control system using ANNAI controllers and a berthing guidance algorithm is 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 chapter 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. 5.1.

The controller consists of ANNAI1 and ANNAI2 which control rudder and engine respectively. These are ANNAI which are similar to the multi-layer feedforward NN

with one hidden layer developed in [58]. 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

$$e1_k = \psi_k^d - \psi_k, \tag{5.1}$$

$$e2_k = u_k^d - u_k. \tag{5.2}$$

These error signals and their time delays are inputs of ANNAI1 and ANNAI2 (Fig. 5.1). The output of ANNAI1 is command rudder angle (δ_k^c) whereas that of ANNAI2 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.

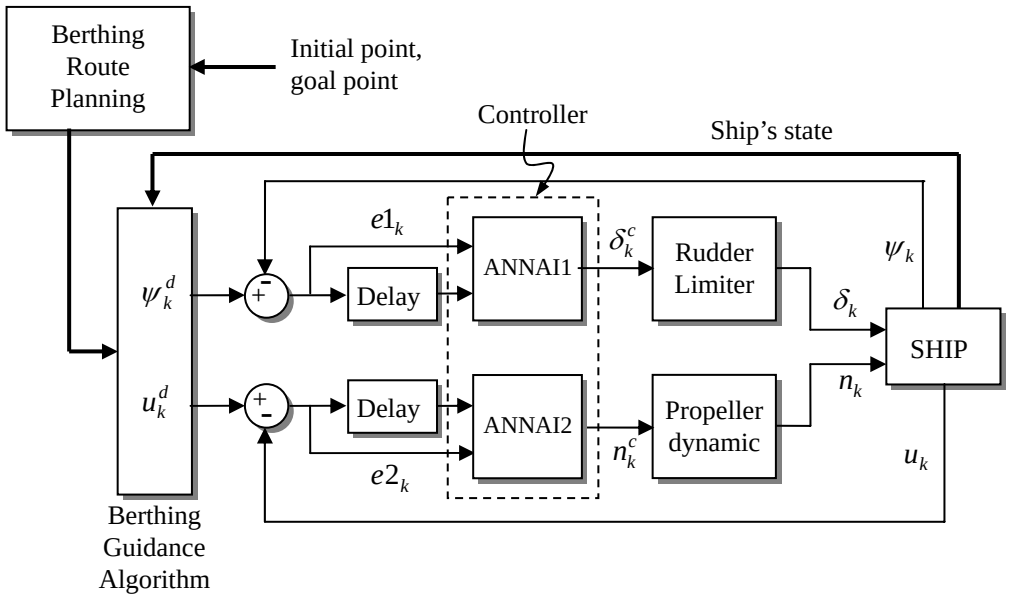


Fig. 5.1 Configuration of automatic berthing control system

5.2.1 Control of Ship Heading

The configuration of the ANNAI1 is shown in Fig. 5.2 where $w1_{ij}$ 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 ANNAI1 are merely heading error and its time-delayed values.

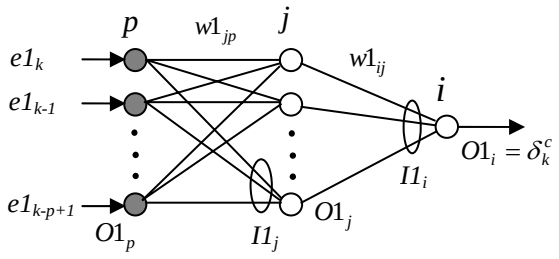


Fig. 5.2 ANNAI1 configuration

Similar to chapter 4, we also define normalized off-track distance as $\mu = di/L$, where L is the length of ship, di will be defined later in (5.21). The cost function used for the ANNAI1 is similar to that of (4.11), and is rewritten as follows

$$E1_k = \frac{1}{2}[\rho_{11}(\psi_k^d - \psi_k)^2 + \rho_{12}(\mu_k^d - \mu_k)^2 + \lambda_1\delta_k^2 + \sigma_1 r_k^2], \tag{5.3}$$

where ρ_{11}, ρ_{12}, λ_1, and σ_1 are positive penalty constants; r_k is yaw rate; and μ_k^d is desired value of μ_k. Note that, we want the ship position to be as close to intended route as possible, therefore $\mu_k^d = 0$ is selected.

Similar to the ANNAI proposed in [60], the adaptation laws for the hidden layer weights and output layer weights of the ANNAI1 are as follows respectively

$$\dot{w}1_{jp} = O1_p[\phi 1_j\, sig(-I1_j) + \gamma_1 \cdot 0] = O1_p\, \phi 1_j\, sig(-I1_j), \tag{5.4}$$

$$\dot{w}1_{ij} = \gamma_1 \cdot O1_j \cdot (\rho_{11} e1_k + \rho_{12}\mu_k + \lambda_1\delta_k + \sigma_1 r_k), \tag{5.5}$$

where

$O1_p$ is the set of p inputs to the ANNAI1 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,

$$O1_j = sig(I1_j) = \frac{1}{1+\exp(-I1_j)}, \tag{5.6}$$

$I1_j$ 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,

$$I1_j = \sum_p (w_{jp} O1_p) + \theta 1_j , \tag{5.7}$$

γ_1 is the learning rate, and

$$\phi 1_j = w1_{ij} \cdot \dot{w}1_{ij} . \tag{5.8}$$

Using the adaptation laws (5.4) and (5.5), ANNAI1 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 5.2.3.

5.2.2 Control of Ship Speed

The configuration of the ANNAI2 is similar to that of ANNAI1 and shown in Fig. 5.3, but $e1_k$ and its time delayed signals are replaced by $e2_k$ and its delayed signals at time steps *k-1*, *k-2*,⋯, *k-p+1*. The task of ANNAI2 is to infer proper engine revolution command such that the following cost function is minimized

$$E2_k = \frac{1}{2}[\rho_2 (u_k^d - u_k)^2 + \lambda_2 n_k^2 + \sigma_2 (u_k - u_{k-1})^2] . \tag{5.9}$$

The adaptation law for hidden layer weights of ANNAI2 is in the form of equation (5.4)

$$\dot{w}2_{jp} = O2_p[\phi 2_j \cdot sig(-I2_j) + \gamma_2 \cdot 0) = O2_p \cdot \phi 2_j \cdot sig(-I2_j)\,. \qquad (5.10)$$

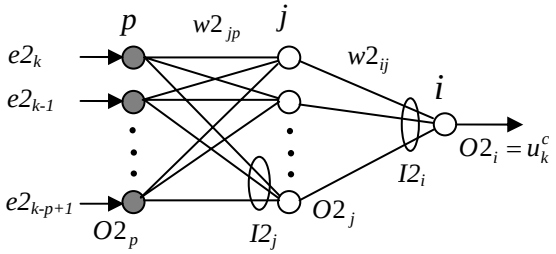


Fig. 5.3 ANNAI2 configuration

In ANNAI2 the output neuron is tangent sigmoidal activation function type, where output signal is

$$O2_i = u_k^c = \tan sig(I2_i) = \frac{2}{1+\exp(-2 \cdot I2_i)} - 1\,. \qquad (5.11)$$

Based on [58], the adaptation law for the output layer weight can be written as

$$\dot{w}2_{ij} = -\gamma_2[\tan sig(I2_i)]' sig(I2_j)\frac{\partial E2_k}{\partial u_k}\,. \qquad (5.12)$$

Taking derivative of $\tan sig(I2_i)$, (5.12) can be expressed as

$$\dot{w}2_{ij} = -\gamma_2 \cdot O2_j \frac{\exp(-2 \cdot I2_i)}{[1+\exp(-2 \cdot I2_i)]^2} \cdot \frac{\partial E2_k}{\partial u_k}\,. \qquad (5.13)$$

Using the chain rule we can write

$$\frac{\partial E2_k}{\partial u_k} = \frac{\partial E2_k}{\partial u_k} - \frac{\partial E2_k}{\partial n_k} + \frac{\partial E2_k}{\partial \dot{u}_k}\frac{\partial \dot{u}_k}{\partial n_k}\,, \qquad (5.14)$$

in which,

$$\dot{u}_k = u_k - u_{k-1}. \tag{5.15}$$

Note that $\dot{u}_k$ increases or decreases following the increase or decrease of the engine revolution n_k. So $\partial \dot{u}_k / \partial n_k$ in the equation (5.14) can be replaced with $sign(\partial \dot{u}_k / \partial n_k) = 1$ to yield

$$\frac{\partial E2_k}{\partial u_k} = \frac{\partial E2_k}{\partial u_k} - \frac{\partial E2_k}{\partial n_k} + \frac{\partial E2_k}{\partial \dot{u}_k} = -(\rho_2 e2_k + \lambda_2 n_k - \sigma_2 \dot{u}_k). \tag{5.16}$$

Replacing (5.16) into (5.13) yields

$$\dot{w}2_{ij} = \gamma_2 \cdot O2_j \frac{\exp(-2 \cdot I2_i)}{[1+\exp(-2 \cdot I2_i)]^2} \cdot (\rho_2 e2_k + \lambda_2 n_k - \sigma_2 \dot{u}_k). \tag{5.17}$$

To summarize, the adaptation law for the hidden layer weights and output layer weights of ANNAI2 are described in equations (5.10) and (5.17) respectively. Using these adaptation laws, ANNAI2 can make the ship speed track the desired value u_k^d generated by the berthing guidance algorithm.

5.2.3 Berthing Guidance Algorithm

In this research, the automatic berthing control system is designed 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 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 (Φ) while ship moves along the route (Fig. 5.4).

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.

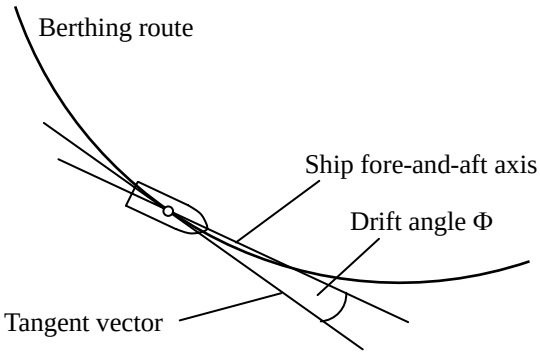


Fig. 5.4 Concept of the drift angle

(1) Calculation of ψ_k^d

If the ship is on the desired berthing route (M in Fig. 5.5), the desired heading ψ_k^d is the direction from M to N, where y_{Nk} is determined by a step of $K \cdot L$ forward from current y_k

$$y_{Nk} = y_k + K \cdot L \,. \tag{5.18}$$

Equation (5.18) is based on the method in [89]. Here, K is a constant and L is the ship length. This ψ_k^d ensures that ship moves with a certain drift angle Φ. However, the radius of the planned berthing route is not equal at every point on the route, hence Φ 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. 5.5. In this case, the new desired heading Φ is determined as the direction from F or F' to N1, with y_{N1k} is determined by a step of $K_1 \cdot L$ forward from current y_k $(0 < K_1 < K)$

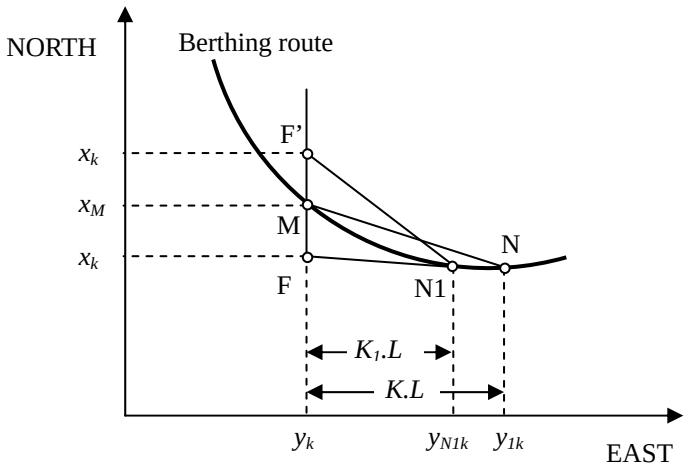


Fig. 5.5 Determination of desired heading

$$y_{N1k} = y_k + K_1 \cdot L\,, \tag{5.19}$$

where K_1 can be obtained from

$$K_1 = (K_{\max} - K_{\min}) \cdot \exp(-\xi \frac{di}{L}) + K_{\min}\,, \tag{5.20}$$

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

$$di = x_k - x_M\,. \tag{5.21}$$

From (5.20) 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 emphasized in the early stages of the berthing process. Only when the ship approaches the berth the velocity values do become more important [89]. Similar to this work, the desired speed u_k^d can be determined as

$$\text{IF} \quad D/L > K_2 \quad \text{THEN} \quad u_k^d = u_k \tag{5.22}$$

$$\text{IF} \quad D/L \le K_2 \quad \text{THEN} \quad u_k^d = \frac{D}{K_2 L} u_k \tag{5.23}$$

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.

5.3 Simulation Results

5.3.1 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. The parameters for NNC1 and NNC2 are selected as follows

$$[\rho_{11}, \rho_{12}, \lambda_1, \sigma_1, N_1, \gamma_1] = [1.5,\ 1.75,\ 0.045,\ 0.2,\ 50,\ 1.5], \tag{5.24}$$

$$[\rho_2, \lambda_2, \sigma_2, N_2, \gamma_2] = [1.5,\ 0.15,\ 0.2,\ 50,\ 2], \tag{5.25}$$

$$[K_{\max}, K_{\min}, K_2, \xi] = [0.3,\ 0.1,\ 0.4,\ 2]. \tag{5.26}$$

Here, N_1, N_2 are the initial number of iterations in one control cycle of NNC1, NNC2; γ_1, γ_2 are the initial learning rates of NNC1, NNC2. During control process, N_1, N_2, γ_1, γ_2 are automatically updated (see chapter 2 and [58]). The initial and final ship position and heading is (34.833^{o}N, 128.83^{o}E) and 150^o; (34.828^{o}N, 128.84^{o}E) and 90^o, respectively. The berth is assumed to be the South border of the chart with latitude of 34.82795^{o}N.

In this research, 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 [80] with length L = 175 m and breadth B = 25.4 m (see more in Appendix B.2).

The effect of wind disturbance against the body of the ship is based on the work of Isherwood (1972) introduced in [79]. A random uniformly distributed signal on [-0.1^o, +0.1^o] 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.

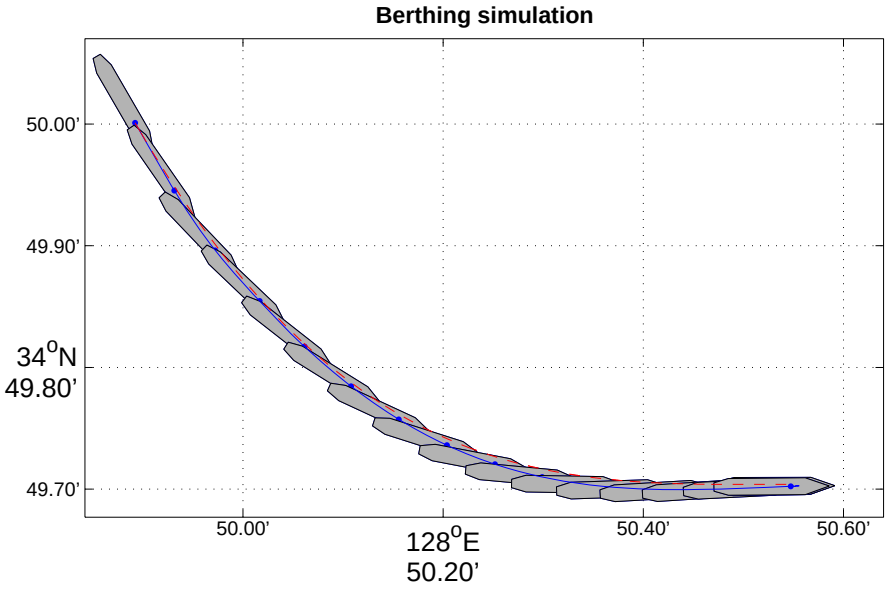
For visual simulation of ship's movement, M-Maps Toolbox, which is a set of mapping tools written in MATLAB and available in [64], is used. The modules written in MATLAB introduced in [60] and chapter 4 for guidance and control using Mercator chart are applied here.

5.3.2 Simulation Results and Discussion

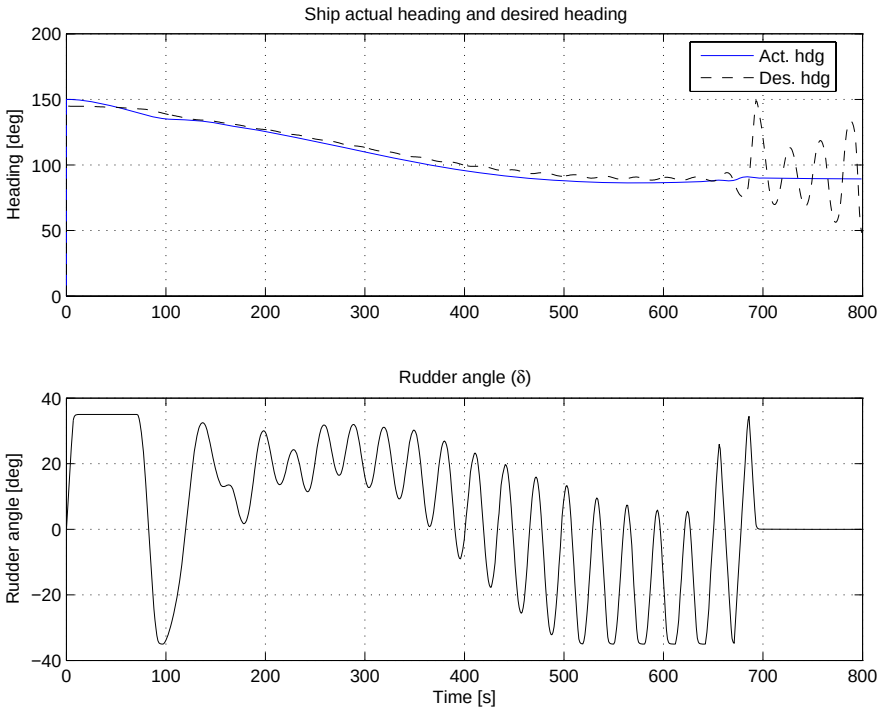
In this subchapter, the berthing control simulations with and without the effect of wind disturbance and measurement noise are presented.

(1) Without wind and measurement noise

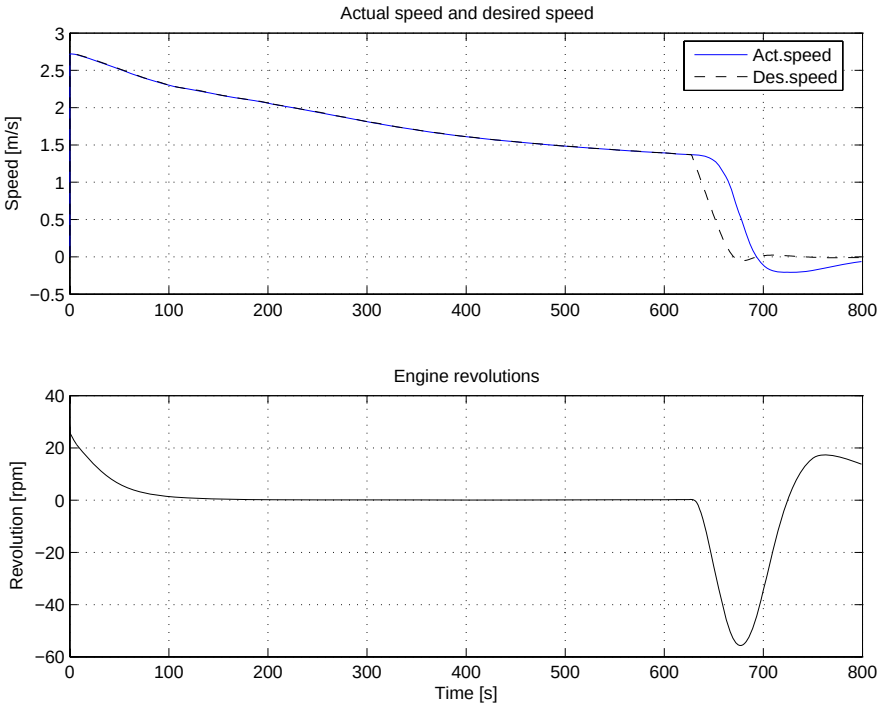
Fig. 5.6 shows the simulation results in case no wind and no measurement noise applied. The ship position on the berthing trajectory is plotted every 45 seconds. It is shown that the tracking target has been satisfactorily achieved. The maximum di is about 12 m, less than half of ship breadth B.



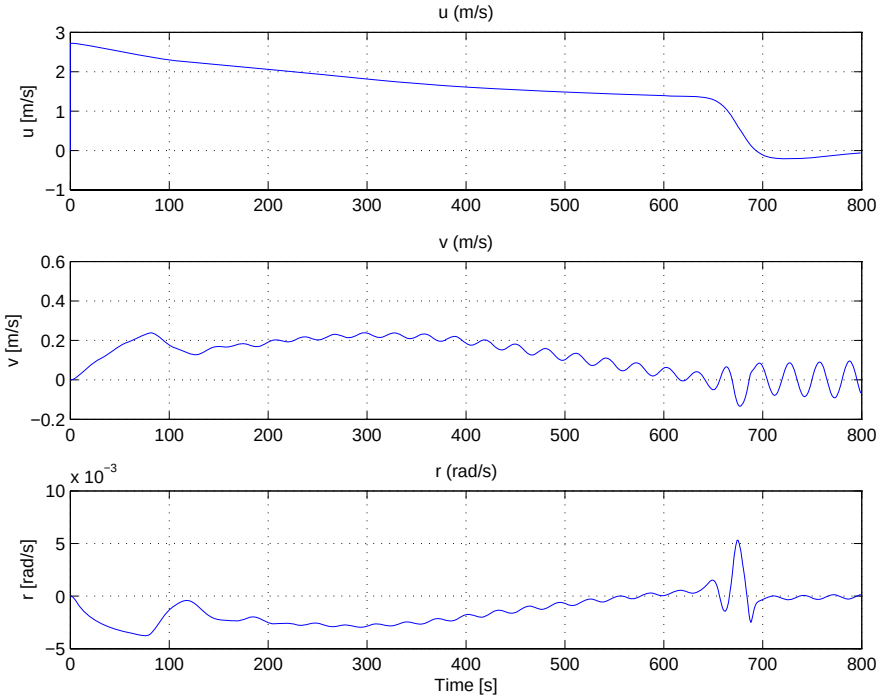
(a) Plot of ship positions: intended berthing route (dashed), actual track (solid)



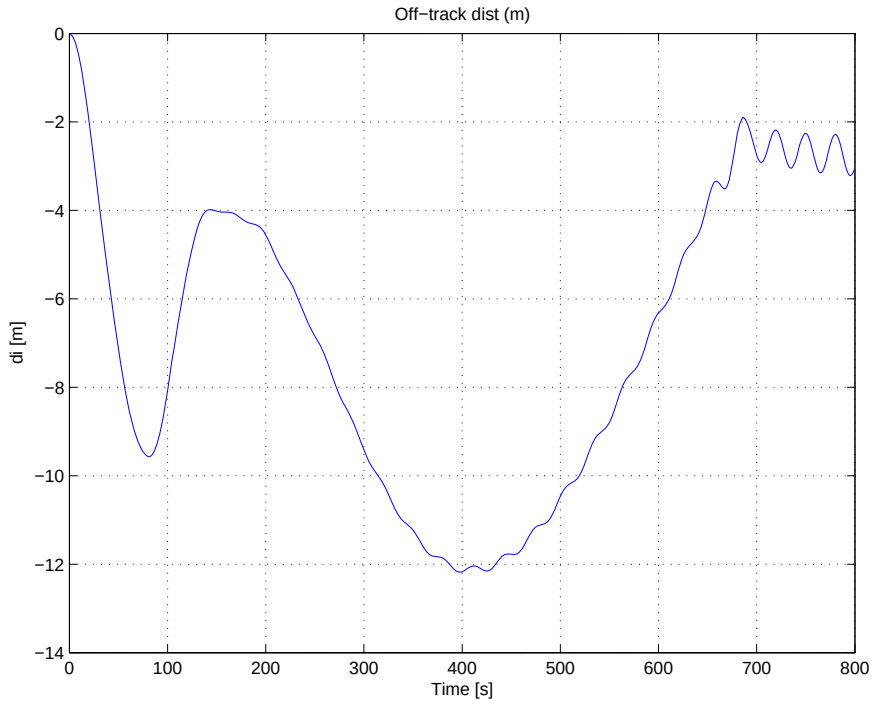
(b) Actual heading, desired heading and rudder angle



(c) Actual speed, desired speed and engine revolution



(d) Surge, sway velocity and yaw rate

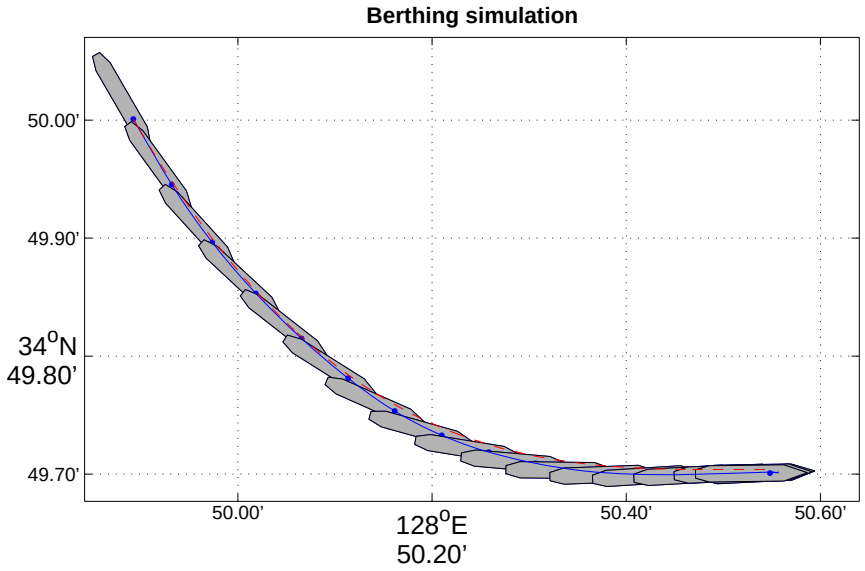


(e) Off-track distance

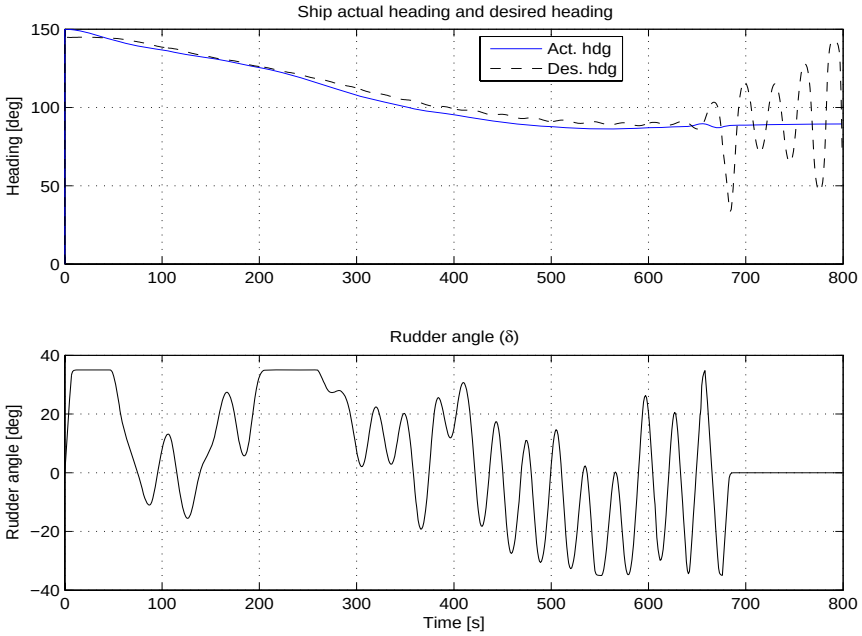
Fig. 5.6 Automatic berthing control without wind and noise

(2) With Wind and Measurement Noise

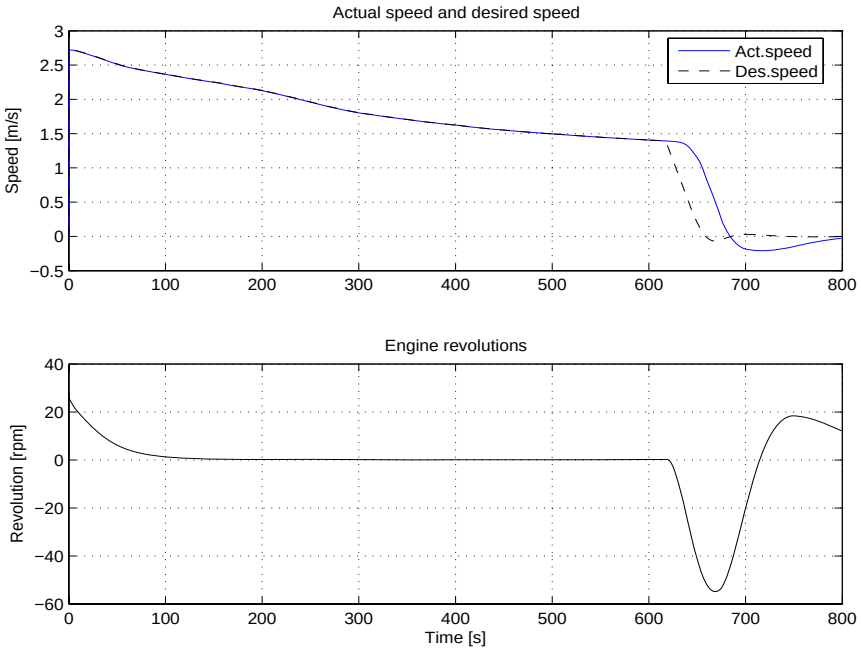
The berthing simulations are undertaken under random offshore and onshore wind and measurement noise. The wind speed changes randomly every 5 seconds and assumes values between 10 knots and 20 knots. Firstly, to represent offshore wind disturbance, the wind direction varies between 90^o and 270^o every 30 s. Secondly, to represent onshore wind disturbance, the wind direction varies from 270^o via 360^0 to 90^o every 30 s. 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 NNC is maintained. The maximum value of di is about 12 m. Only the simulation of onshore wind condition is shown (Fig. 5.7).



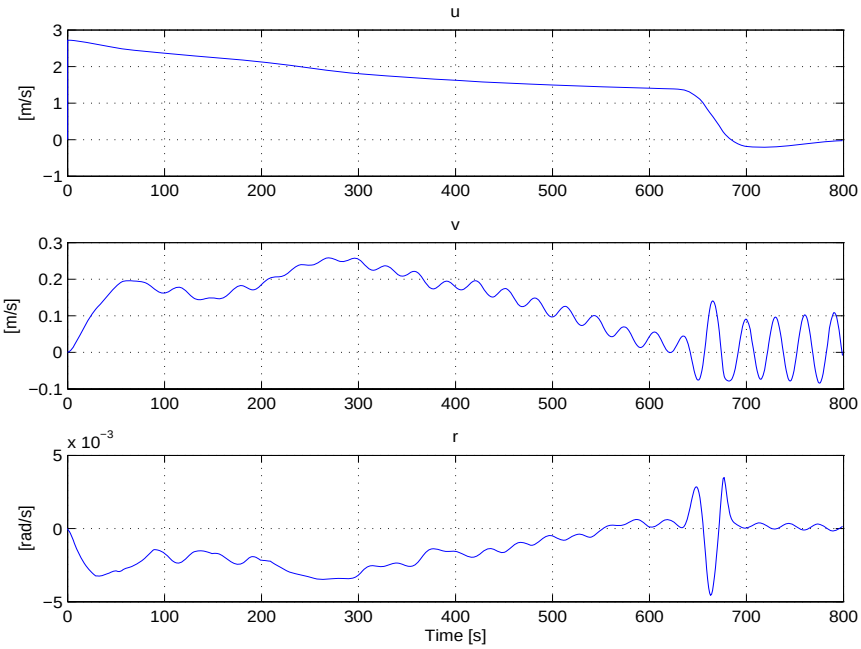
(a) Plot of ship positions: intended berthing route (dashed), actual track (solid)



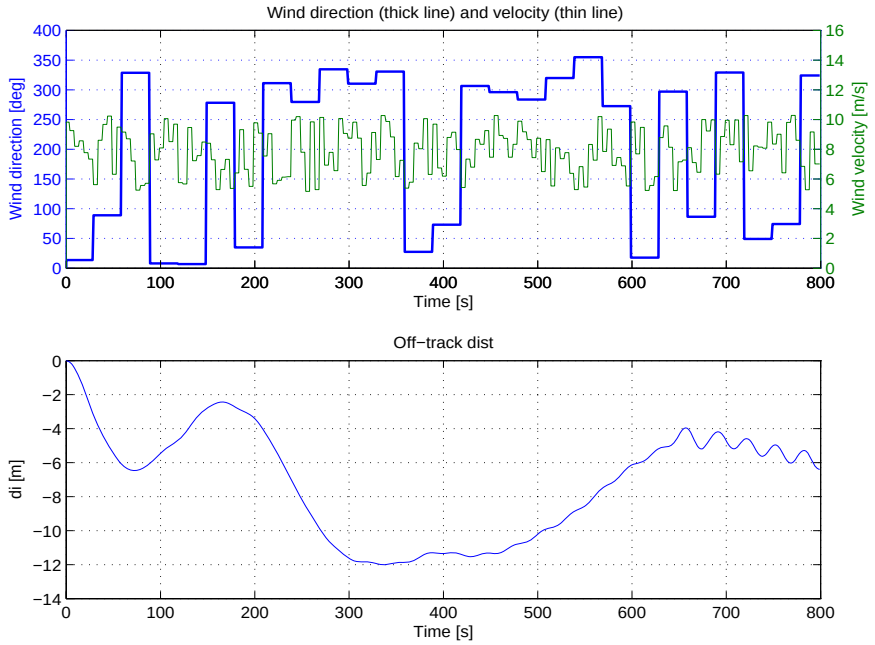
(b) Actual heading, desired heading and rudder angle



(c) Actual speed, desired speed and engine revolution

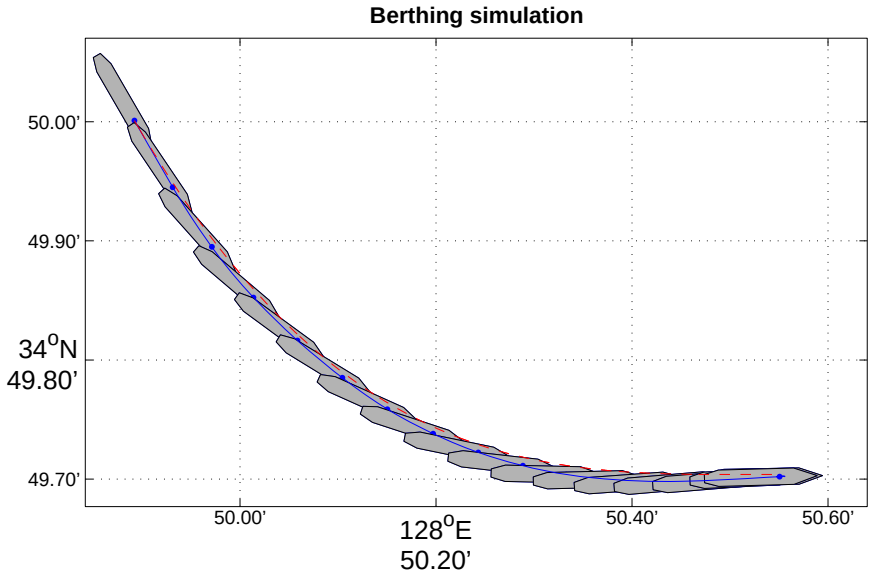


(d) Surge, sway velocity and yaw rate

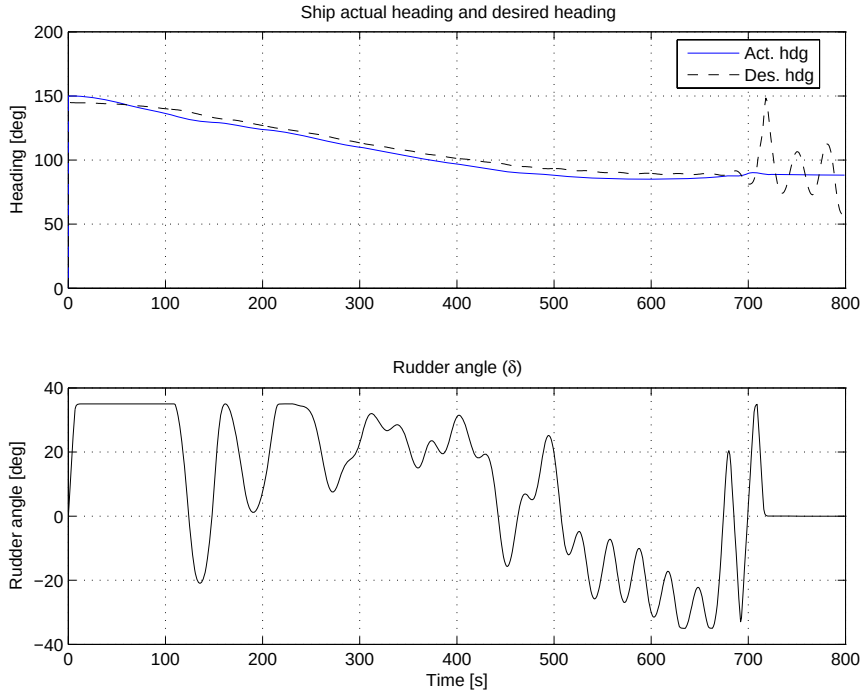


(e) Wind direction, velocity and off-track distance

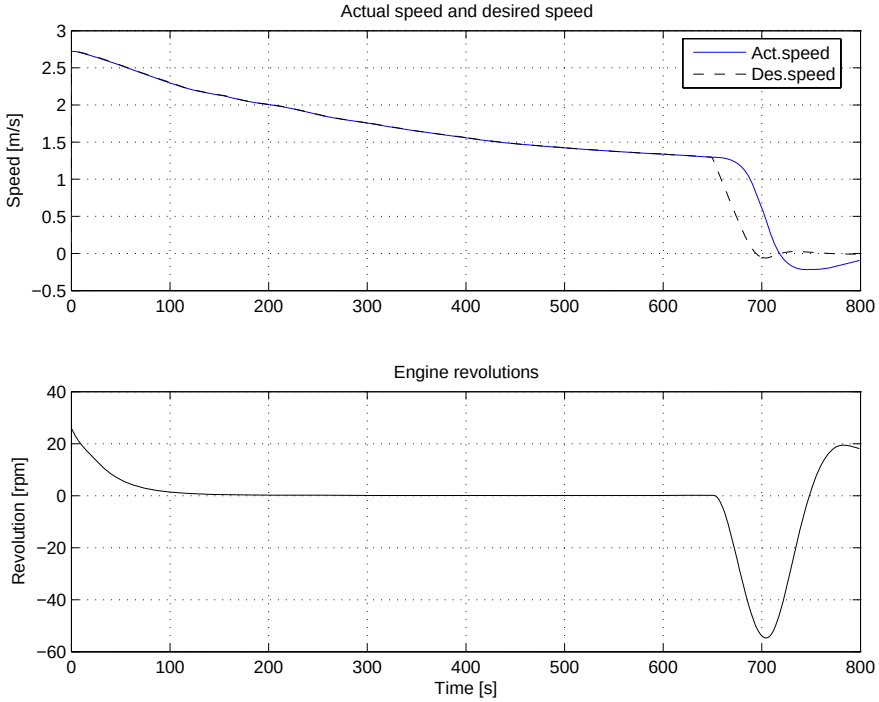
Fig. 5.7 Automatic berthing with onshore wind and noise, wind speed changes randomly from 10 knots to 20 knots



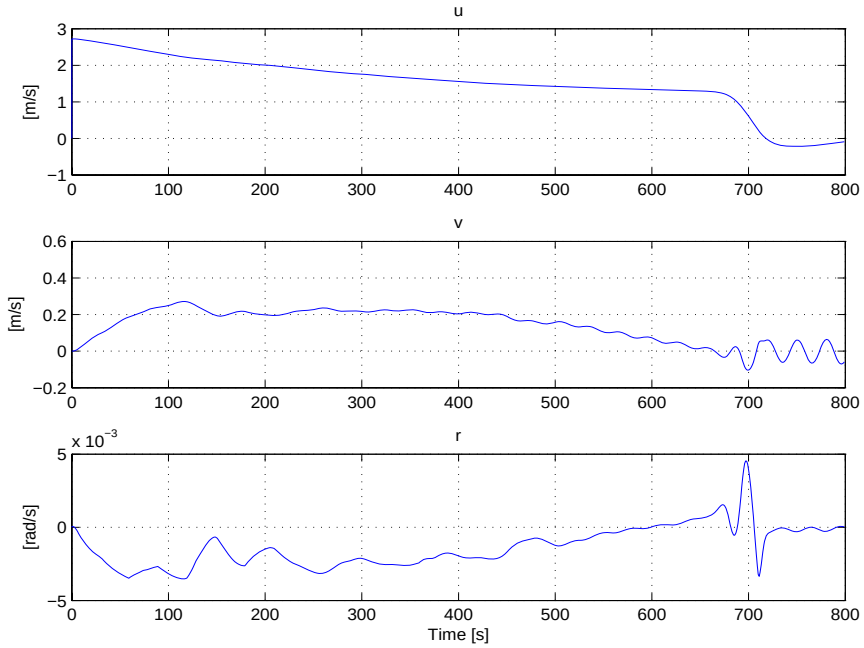
(a) Plot of ship positions: intended berthing route (dashed), actual track (solid)



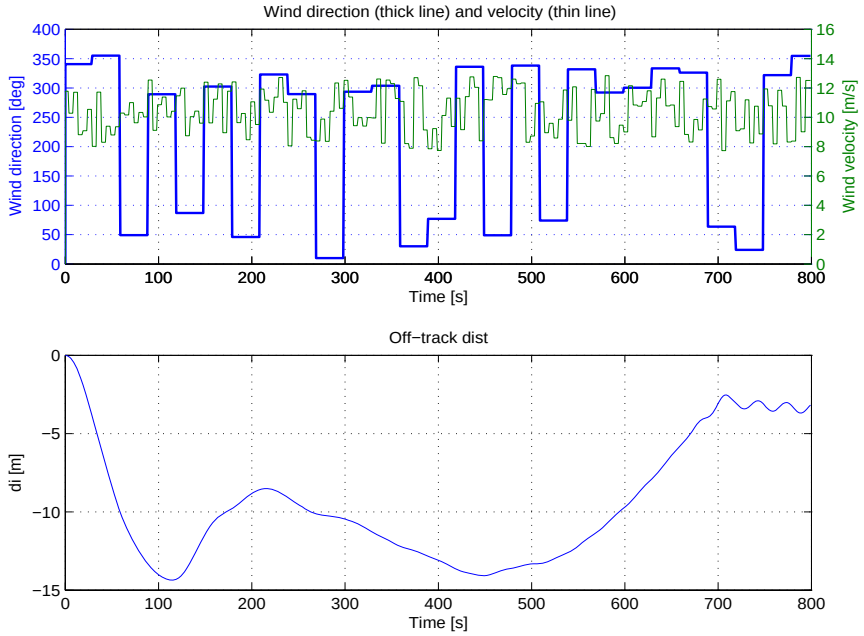
(b) Actual heading, desired heading and rudder angle



(c) Actual speed, desired speed and engine revolution

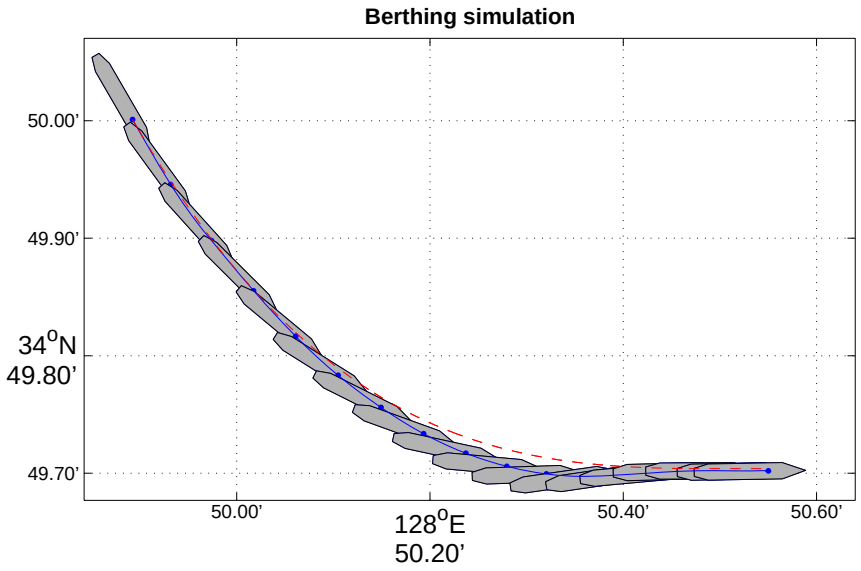


(d) Surge, sway velocity and yaw rate

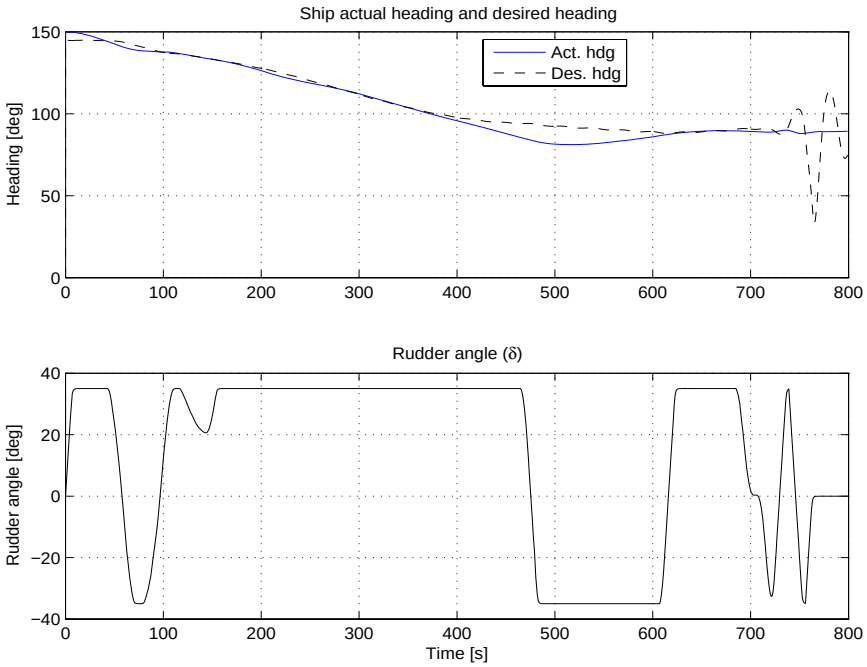


(e) Wind direction, velocity and off-track distance

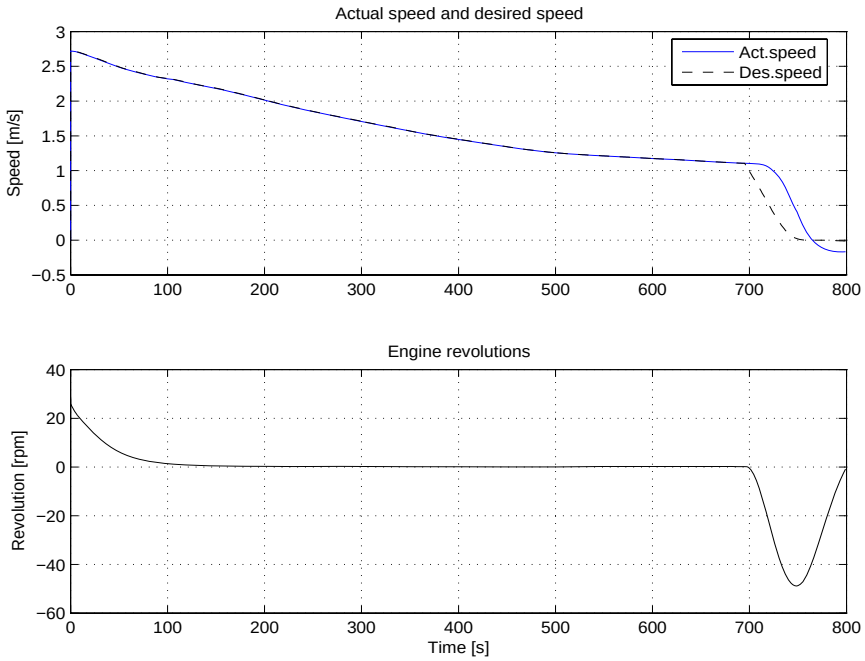
Fig. 5.8 Automatic berthing with onshore wind and noise, wind speed changes randomly from 15 knots to 25 knots



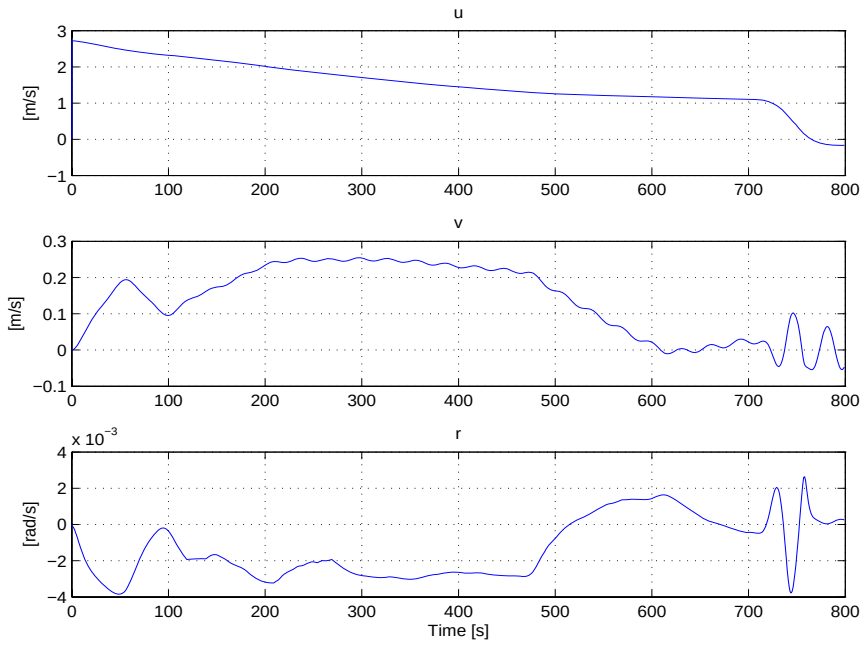
(a) Plot of ship positions: intended berthing route (dashed), actual track (solid)



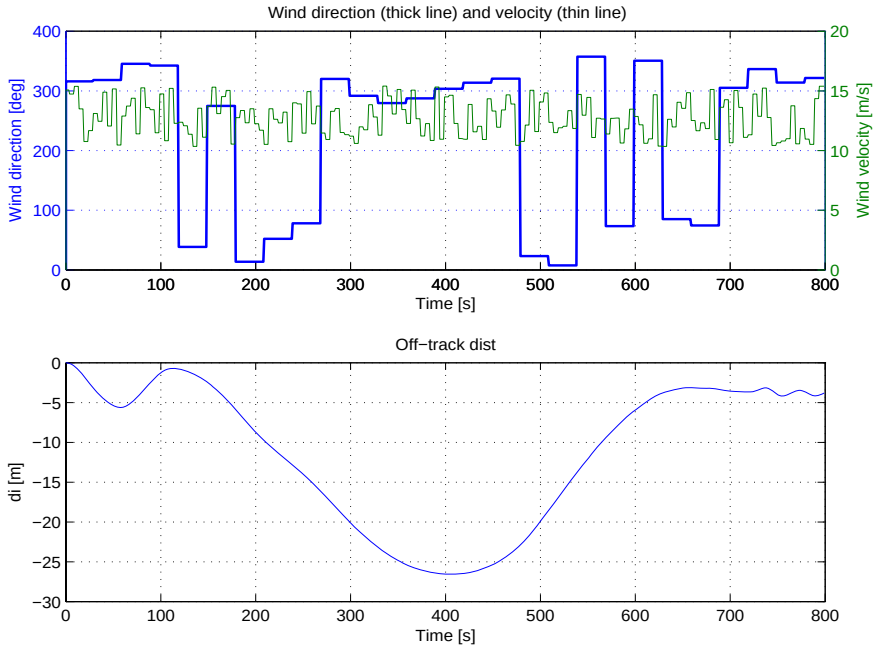
(b) Actual heading, desired heading and rudder angle



(c) Actual speed, desired speed and engine revolution



(d) Surge, sway velocity and yaw rate



(e) Wind direction, velocity and off-track distance

Fig. 5.9 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. 5.8); and then from 20 knots to 30 knots (Fig. 5.9). In Fig. 5.8e, ship is off the desired route with maximum value of about 14.5 m (approximately a half of 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. 5.9 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.

5.4 Conclusion

In this chapter, 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

(a) The NNC can be trained online without the necessity of any teaching data and offline training phase.

(b) The NNC can make both ship's heading and speed track the desired values.

(c) The proposed berthing guidance algorithm works effectively in berthing process.

(d) The unknown and non-linearity ship can be controlled satisfactorily; no priori knowledge of ship is required.

(e) The NNC is not so sensitive to measurement noise of input signals.

(f) The control system is robust under the light effect of wind disturbance.

(g) 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 ship to evaluate the safe distance from final goal point to the berth, which is required for safe berthing operation. These works will be considered in the future.

Chapter 6 ANNAI-based Dynamic Positioning System

6.1 Introduction

Dynamic positioning (DP) systems for ships are commonly the systems that have station-keeping and low-speed maneuvering functions [79]. The ship is controlled in three degree of freedom (surge, sway, and yaw) by means of the ship propulsion system, which includes main propellers aft of the ship, tunnel thrusters and azimuth thruster mounted under the hull. Rudders are not used during station-keeping since the rudder forces are quite small at low speed [83]. Since the 1960s, DP systems have been developed using conventional PID controllers in cascade with low pass and/or notch filters, model-based control utilizing stochastic optimal control theory and Kalman filtering techniques. Conventional DP systems are designed by linearizing the kinematic equations of motions about a set of predefined constant yaw angles such that linear optimal control theory and gain-scheduling techniques can be applied. These control methods and their later extensions as well as modifications proposed by numerous authors for DP systems are briefly mentioned in [79].

More lately DP systems utilizing modified Linear Quadratic Gaussian (LQG) feedback controller and a model reference feedforward controller [83]; nonlinear output feedback [4]; passive nonlinear observer based control [2] and [81]; nonlinear control based on robust observer [52]; nonlinear passive weather optimal positioning control (WOPC) system [82] have been developed. The main trend of these studies is to remove the assumptions of linearization of the kinematics by using nonlinear observer and feedback control theory. These DP systems are designed requiring mathematical models of the controlled ships, based on which nonlinear observer can be constructed.

The main motivation of this research is to remove the necessity of mathematical ship model by using an “intelligent” control technique. In the control systems where the controlled plants are highly nonlinear and external disturbances are highly nonlinear uncertainties, intelligent control techniques are useful. Especially, the NN control, one of the intelligent control techniques, has grown very rapidly in recent years. Many NN control systems of different structures have been proposed and widely applied in a range of technical practices. NNs are very attractive in control applications (see [12], [41], [63], and [75] for more details).

However, very few papers on the application of intelligent control, particular NN control, to DP systems have been found, such as [45], [85]. In this chapter, a hybrid neural adaptive control scheme which can perform station-keeping and low-speed maneuvering of ships is developed. The aim is to take advantage of the learning ability of NNs, and to derive a NN-based control algorithm which is independent of the exact mathematical model of the ship. Furthermore, it is not necessary to estimate the bias term representing slowly-varying external environmental forces and moments. A conventional PD-controller for nonlinear DP model as suggested in [79] is modified and combined with the ANNAI controller introduced in chapter 2 and [58], [59]. In the proposed hybrid neural adaptive control scheme, PD-controller provides an approximate control, and ANNAI controllers with on-line training ability are introduced to improve the DP system performance. For low-speed maneuvering function, we propose an algorithm to guide the ship along the desired track.

At first, the DP system configuration is described. And then, to validate the proposed DP system, computer simulations of station-keeping and low-speed maneuvering performance of a multi-purpose supply ship are presented under the influence of measurement noise and external disturbances.

6.2 Dynamic Positioning System

This section presents a new DP system based on the ANNAI controller. The DP

system has two functions, position-keeping and low-speed maneuvering.

Fig. 6.1 shows the configuration of the proposed DP system, where hybrid neural adaptive controller consists of a PD-controller (which has gains K_p' and K_d') and three adaptive neural networks ANNAI$_1$, ANNAI$_2$, ANNAI$_3$. In Fig. 6.1, each element of vector $J^T(\psi)\hat{e}$ is input to one ANNAI controller. Elements of the vector $O_n = [O_{n1}, O_{n2}, O_{n3}]^T$ are outputs of the ANNAI controllers. *Thruster allocation* block is used to calculate the contribution of each actuator of the ship propulsion system. More detail of the DP system model is presented in Appendix A.

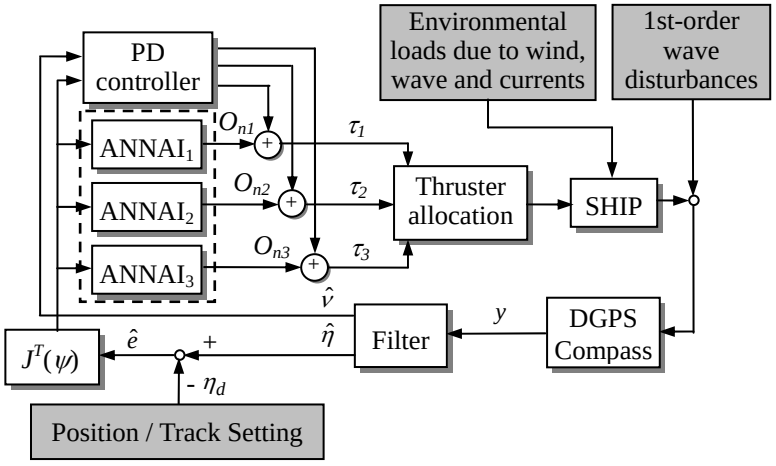


Fig. 6.1 Configuration of the proposed ANNAI-based DP system. Conventional PD controller is combined with the ANNAI controllers

6.2.1 Station-keeping Control

(1) Hybrid Neural Adaptive Controller

It is shown in [79] that the DP system was designed based on the nonlinear DP model

$$\dot{\eta} = J(\eta)\nu\,, \tag{6.1}$$

$$M\dot{\nu} + D\nu = \tau + J^T(\eta)b\,, \tag{6.2}$$

$$y = \eta + \eta_w\,, \tag{6.3}$$

where η is the vector of earth-fixed position and heading, ν is the vector of vessel-fixed linear velocity, $J(\eta)$ is the transformation matrix between the earth-fixed coordinate and the vessel-fixed coordinate (for surface vessel $J(\eta) = J(\psi)$ and $J^{-1}(\psi) = J^T(\psi)$, τ is the control vector of forces and moment, b is a vector of bias forces and moment representing slowly-varying environmental disturbances, η_w is a zero means bounded disturbance vector, y is the measurement vector, M is the inertia matrix including hydrodynamic added inertia, and D is the damping matrix. The more detail of DP model is presented in Appendix A.

Instead of using integral action to compensate for b, in [79] (page 430) a PD-controller

$$\tau = -J^T(\psi)K_p e - K_d\nu - J^T(\psi)b\,, \tag{6.4}$$

$$e = \eta - \eta_d \tag{6.5}$$

was used under the assumption that b is known (perfect compensation) and η_d is the desired states, $\dot{\eta}_d = 0$. However, it is impossible to measure b, so in that study, a state observer which can generate estimates of η, ν, and b and at the same time provide wave filtering was needed. Hence, a nonlinear mathematical model of the ship was used. The controller using the estimates states $\hat{\eta}$, $\hat{\nu}$, and $\hat{b}$ in [79] is

$$\tau = -J^T(\psi)K_p\hat{e} - K_d\hat{\nu} - J^T(\psi)\hat{b}\,, \tag{6.6}$$

$$\hat{e} = \hat{\eta} - \eta_d\,. \tag{6.7}$$

In this research we avoid using any mathematical model of ship in designing the controller. Additionally, it is not necessary to estimate the bias term b. In order to do that, we propose a hybrid neural adaptive control scheme as described in Fig. 6.1. Equation (6.6) suggests that we can use a conventional PD-controller with parameters K'_p and K'_d approximately selected, that is

$$\tau_{PD} = -J^T(\psi)K'_p\hat{e} - K'_d\hat{v}. \tag{6.8}$$

The control input τ_{PD} by (6.8) can provide an approximate control of ship. However, the controller must be able to compensate optimally the effects of b as well as minimize $\hat{e}$. So a proper compensation value must be added to τ_{PD} in every control cycle. We know that the ship dynamics and external disturbances are highly nonlinear, hence neural network controller is a good choice in this situation.

The ANNAI controller introduced in chapter 2 is stable and can be adaptively trained in fast time [58], [59]. Further more, we can use this ANNAI controller in direct control method. Hence, it is suitable to use the ANNAI controller in parallel with the PD-controller. For these reasons we apply the ANNAI controller in combination with the PD-controller in (6.8). The control output of the proposed controller has the form

$$\tau = \tau_{PD} + \tau_{NN} = -J^T(\psi)K'_p\hat{e} - K'_d\hat{v} - O_n, \tag{6.9}$$

where $\tau_{NN} = -O_n = -F[J^T(\psi)\hat{e}]$, $O_n \in \Re^3$ is outputs of three ANNAI controllers, $\hat{e}$ is determined by (6.7). For the ANNAI controllers of this application, $X_k = \eta$, $X_k^d = \eta_d$, and $u_k^c = \tau_{NN}$. The controller described by (6.9) is independent of an explicit mathematical model of ship and requires no estimation of b. It is expected that the model error and estimation error can be removed. This ability is an advantage in comparison with prior works in terms of coping with nonlinearities and uncertainties in ship dynamics as well as external disturbances.

(2) Adaptation Law of ANNAI Controllers

Now we discuss how the ANNAI controllers are designed and can derive the control vector τ_{NN} in (6.9). For the proposed DP system, the estimates states $\hat{\eta}$ and $\hat{\nu}$ are obtained with proper wave filters. The selected ANNAI controllers are multi-layer feedforward neural networks with one hidden layer. The vector of transformed error $\varepsilon = J^T(\psi)\hat{e}$ is the input vector of ANNAI controllers and must be converged to zero. The cost functions for these controllers have the following form

$$E_i = \frac{1}{2}[\rho_i \varepsilon_i^2 + \lambda_i O_{ni}^2 + \kappa_i \dot{\hat{v}}_i^2], \tag{6.10}$$

where ρ_i, λ_i, κ_i (i = 1, …,3) are positive constants.

The adaptation law for hidden layer of the ANNAI controller as in [58] can be written as

$$\dot{w}_i^{hid} = \varepsilon_i \phi_i sig(-I_i^{hid}), \tag{6.11}$$

where $sig(\cdot)$ is a sigmoidal activation function and

$$\phi_i = w_i^{out} \cdot \dot{w}_i^{out}, \tag{6.12}$$

$$I_i^{hid} = \sum w_i^{hid} \varepsilon_i + \theta_i^{hid}, \tag{6.13}$$

where θ_i^{hid} is the threshold values of the hidden layers. For the output layer, the adaptation law has the following form

$$\dot{w}_i^{out} = \gamma_i sig(I_i^{hid})(\rho_i \varepsilon_i + \lambda_i O_{ni} + \kappa_i \dot{\hat{v}}_i). \tag{6.14}$$

Here, γ_i are learning rates of the ANNAI controllers. In this study, output neurons have tangent sigmoidal activation function, such that

$$O_{ni} = \tan sig(I_i^{out}), \tag{6.15}$$

$$I_i^{out} = \sum w_i^{out} sig(I_i^{hid}) + \theta_i^{out}, \tag{6.16}$$

where θ_i^{out} is the threshold values of the output layers.

To summarize, the ANNAI controllers can minimize the cost function (6.10) using adaptation laws (6.11) and (6.14). Once the outputs O_n of ANNAI controllers are determined, the control input of DP system is determined by (6.9). Using the control scheme described in (6.9), the DP system can compensate for unknown bias term representing slowly-varying environmental disturbances, and minimize positioning error. Further details of the ANNAI adaptation laws were shown in chapter 2 and can be found in [58] and [59].

6.2.2 Low-speed Maneuvering Control

This subsection presents the low-speed maneuvering control function of the DP system. To maneuver the ship the *reference point* method is used. At every control cycle, the ship is stabilized on a moving reference point R(x_d, y_d) (Fig. 6.2) at a desired heading ψ_d. In this case the desired states vector is $\eta_d = [x_d, y_d, \psi_d]^T$. Suppose that we want to make a certain point H(x_H, y_H) of the ship (as shown in Fig. 6.2) follow the desired track (be stabilized at R). If $\eta_H = [x_H, y_H, \psi]^T$ is the ship states at H, the error vector is now expressed as

$$e = \eta_H - \eta_d. \tag{6.17}$$

In Fig. 6.2, position of H in the vessel-fixed reference coordinate is determined by Δx and Δy. The position of H in the earth-fixed reference coordinate can be easily

obtained as

$$\eta_H = \hat{\eta} + J(\psi)\begin{bmatrix}\Delta x\\ \Delta y\\ 0\end{bmatrix} = \hat{\eta} + J(\psi)d_H\,, \tag{6.18}$$

where $d_H = [\Delta_x, \Delta_y, 0]^T$. From (6.8), (6.17), and (6.18) the hybrid neural adaptive control scheme to stabilize H at the reference point R is proposed as

$$\tau = \tau_{PD} + \tau_{NN} = -J^T(\psi)K_p'\hat{e} - K_d'\hat{\nu} - O_n\,, \tag{6.19}$$

$$O_n = F[J^T(\psi)\hat{e}]\,, \tag{6.20}$$

$$\hat{e} = \hat{\eta} + J(\psi)d_H - \eta_d\,. \tag{6.21}$$

Here, the adaptation laws of the ANNAI controllers are similarly determined as in the previous subsection. Using the control scheme expressed in (6.19), (6.20), and (6.21), the DP system can compensate for unknown bias term representing slowly-varying environmental disturbances, and minimize tracking error.

In order to make the ship follow the desired track, we propose an algorithm to move the reference point R along the desired track. Let z_e be the distance HR, we can have

$$z_e = \sqrt{(x_H - x_d)^2 + (y_H - y_d)^2}\,. \tag{6.22}$$

Based on the work in [35], the speed u of R is chosen as

$$u(t, z_e, \Delta\psi) = u^*(1 - \chi_1 e^{-\chi_2(t-t_0)})e^{-\chi_3 z_e}e^{-\chi_4|\Delta\psi|}\,, \tag{6.23}$$

where additional term $e^{-\chi_4|\Delta\psi|}$ is added to the original algorithm, and $u^* \neq 0$, $\chi_i > 0$, $i = 1, \ldots 4$, $\chi_1 < 1$, and $\Delta\psi = \psi - \psi_d$.

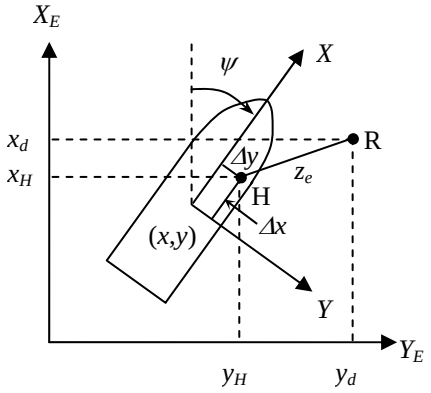


Fig. 6.2 General framework of low-speed maneuvering. In the vessel-fixed reference coordinate, H is determined by Δx and Δy. In the earth-fixed reference coordinate, H is determined by x_H and y_H, R is determined by x_d and y_d.

The choice of $u(t,z_e)$ in (6.23) has the following desired features: when the tracking error z_e and/or heading error $\Delta\psi$ are large, the reference point R will wait for the ship position and her heading to reach to the set point η_d; when z_e and $\Delta\psi$ are small, the reference point R will move along the desired track at the speed closed to u^* and the ship follows it within the specified look ahead distance while maintaining the desired heading [35].

6.3 Simulation Results

In order to validate the proposed DP control system, we carry out computer simulations using the nonlinear model of an off-shore supply ship *Northern Clipper* which was presented in [81]. The length of *Northern Clipper* is L = 72.6 m and the mass is $m = 4.591 \cdot 10^6$ kg (see more in Appendix B.3). The coordinate system is located in the center of gravity. The bias time constants are chosen as $T = diag\{1000,1000,1000\}$.

The wave model parameters are also chosen as in [81] with ζ_i = 0.1 and ω_{oi} = 0.8976 rad/s corresponding to a wave period of 7.0 s in surge, sway and yaw.

The ANNAI controllers are feedforward neural networks with four input neurons, six hidden neurons and one output neuron. The input vector of each neural network consists of ε_i and their three delayed signals. Number of training iterations in one control cycle of each neural network is fixed at 50. The other parameters are

$$[\rho_1, \rho_2, \rho_3] = [0.125, 0.175, 0.25], \tag{6.24}$$

$$[\lambda_1, \lambda_2, \lambda_3] = [1, 0.025, 0.2], \tag{6.25}$$

$$[\kappa_1, \kappa_2, \kappa_3] = [1.5, 0.02, 0.2], \tag{6.26}$$

$$[\gamma_1, \gamma_2, \gamma_3] = [0.3, 0.5, 0.5]. \tag{6.27}$$

The gains of PD-controller are chosen as:

$$K'_p = diag\{50e3, 50e3, 200e3\}, \tag{6.28}$$

$$K'_d = diag\{10e3, 10e3, 40e3\}. \tag{6.29}$$

6.3.1 Station-keeping

In Fig. 6.3 the simulation results of four situations are shown. Firstly, we apply no control action and the ship moves away from original position (0, 0) due to external disturbances. Secondly, we apply PD-controller and maximum distance from ship to the original position is about 3 m. Thirdly, we use ANNAI controllers. Although ship does not move away, her position is not stationed around the original position. Finally, we apply the proposed hybrid adaptive neural controller. The ship is stationed around the original position with maximum distance in *x* is about 0.5 m, and in *y* is about 1 m. Simulation time is 600 s.

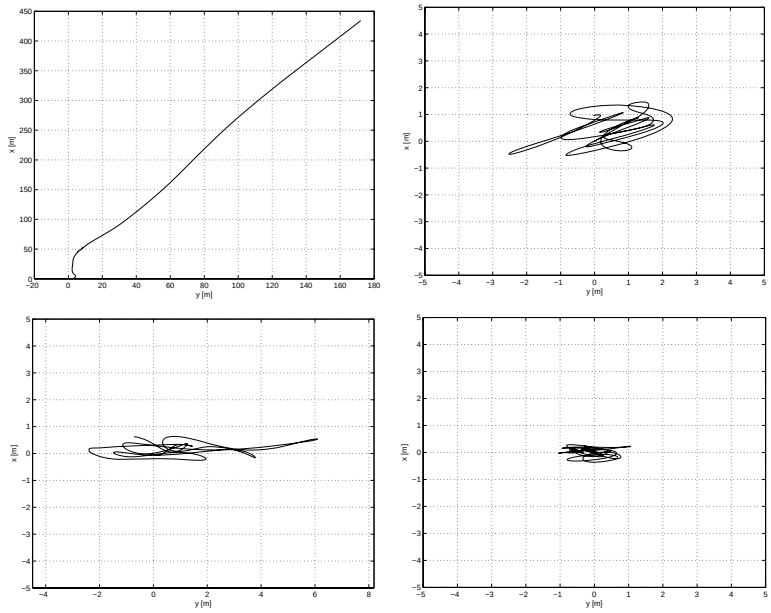
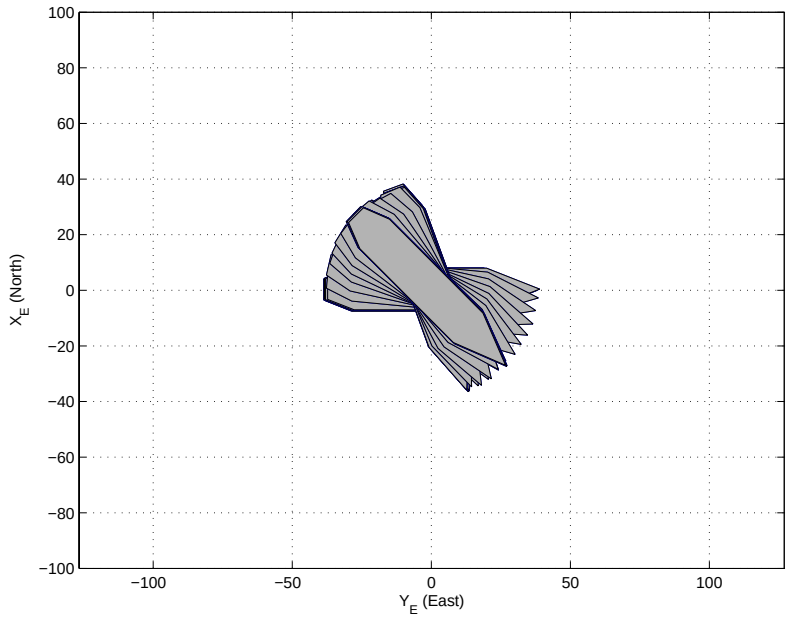
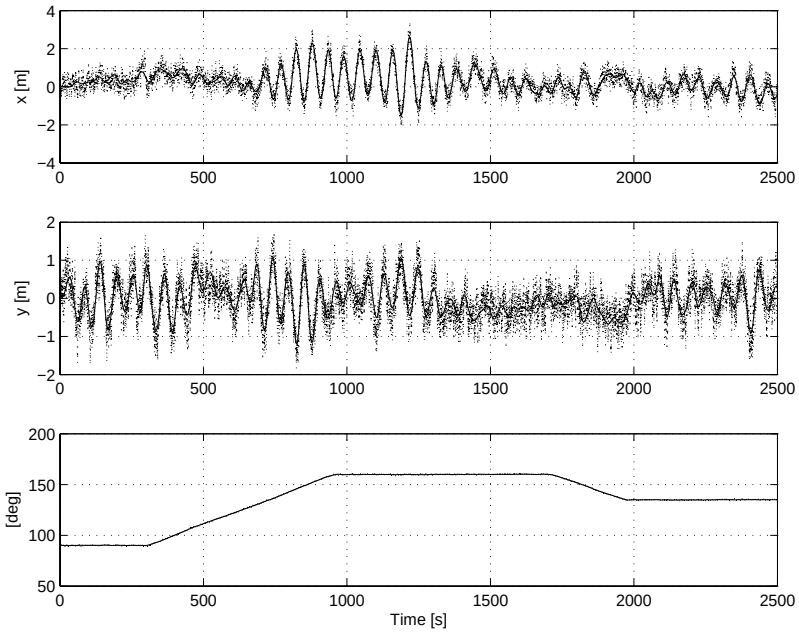


Fig. 6.3 Plot of ship position. Without controller (upper-left); with PD-controller (upper-right); with ANNAI controllers (lower-left); with hybrid adaptive neural controller (lower-right)

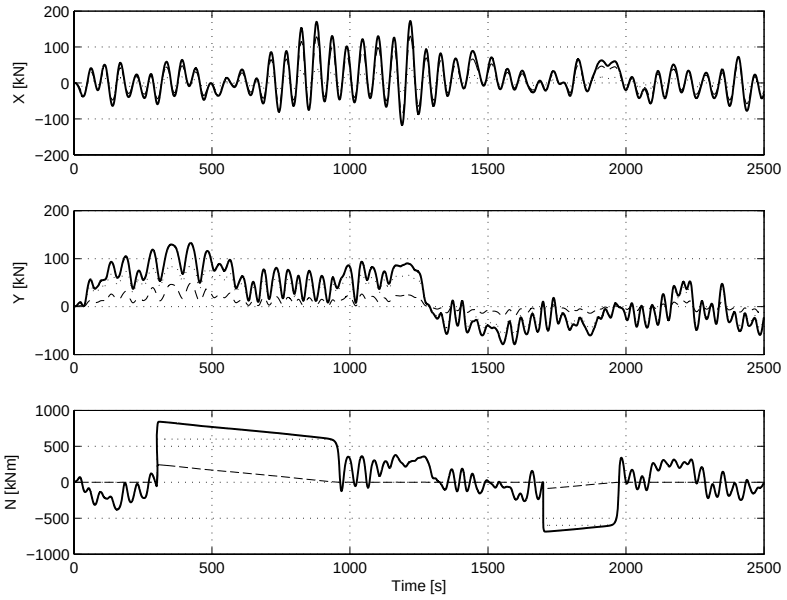
In next simulation (Fig. 6.4), the center of gravity is stationed at the point (0, 0). Initial ship heading is 90^{o}, after 300 s the heading is changed to 160^{o}, and after 1700 s it is changed to 135^{o}. The simulation result has shown the ability of the DP system in station-keeping. The ship is stably kept at desired position and direction is correctly changed under the effect of external disturbances represented by bias term. As shown in Fig. 6.4b, the maximum error in x is about 2.5 m, and in y about 1.2 m.



(a) Plot of ship position in *xy* coordinates in station-keeping simulation.



(b) Measured (dotted), filtered (solid) position in x, y, and heading.



(c) Control output of controllers. ANNAI controllers (dotted), PD-controller (dashed), and hybrid neural adaptive controller (solid).

Fig. 6.4 Station-keeping simulation results

6.3.2 Low-speed Maneuvering

In these simulations we select a desired track connecting four *marked points* (0, 0), (100,-100), (200, 0), (100, 100), (0, 0). In all simulations the ship positions are plotted every 60 seconds. Off-track distance, filtered position in *x*, *y*, and ship heading, control forces and moment are shown. In low-speed maneuvering following marked points, we use distance from R to current marked point Z_{ep} as in [35] to modify (6.23) as follow

$$u(Z_{ep}, z_e, \Delta\psi) = u^*(1 - \chi_1 e^{-\chi_2 Z_{ep}})e^{-\chi_3 z_e} e^{-\chi_4 |\Delta\psi|} \tag{6.30}$$

where

$$[\chi_1, \chi_2, \chi_3, \chi_4] = [0.95, 0.2, 0.2, 15]. \tag{6.31}$$

Equation (6.30) can reduce ship speed exponentially while approaching the marked point to prevent position overshoot. By selecting values of Δx, Δy to

determine position of H, we can make a specific point of ship follow the desired track. The following three cases are simulated:

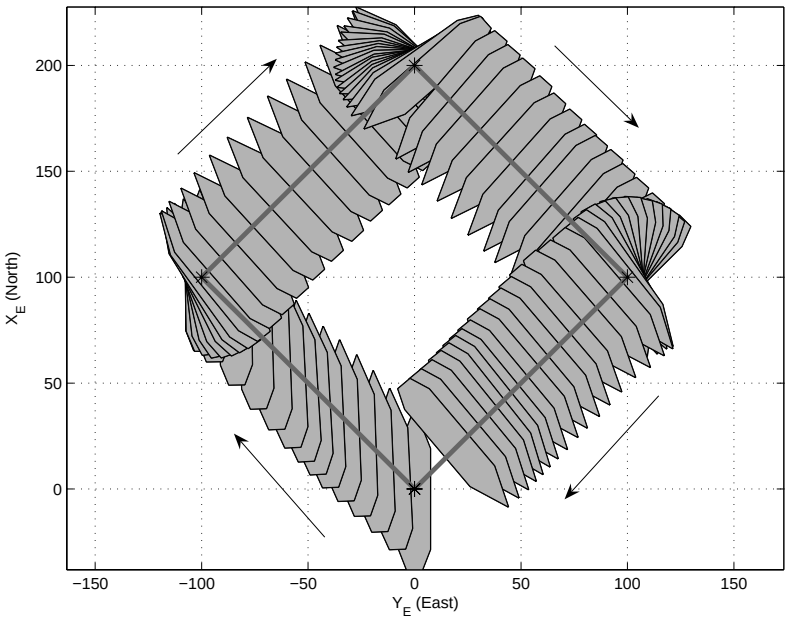
Case 1: The point H is located at the center of gravity of the ship and follows the desired track while ship heading on each segment is set to 0^o, 315^o, 225^o, 135^o. In this case, position of H in vessel-fixed reference coordinate is chosen as: $\Delta x = 0$, $\Delta y = 0$. The initial position and heading of ship is (0, 0) and 0^o. The simulation result is shown in Fig. 6.5.

In this simulation, the center of gravity of the ship moves along the desired track with small off-track distance and heading is kept at desired value. At each marked point, ship heading changes to new desired value before the ship continues to move along new segment. This action is resulted in by the effect of new item $e^{-\chi_4|\Delta\psi|}$ in (6.23).

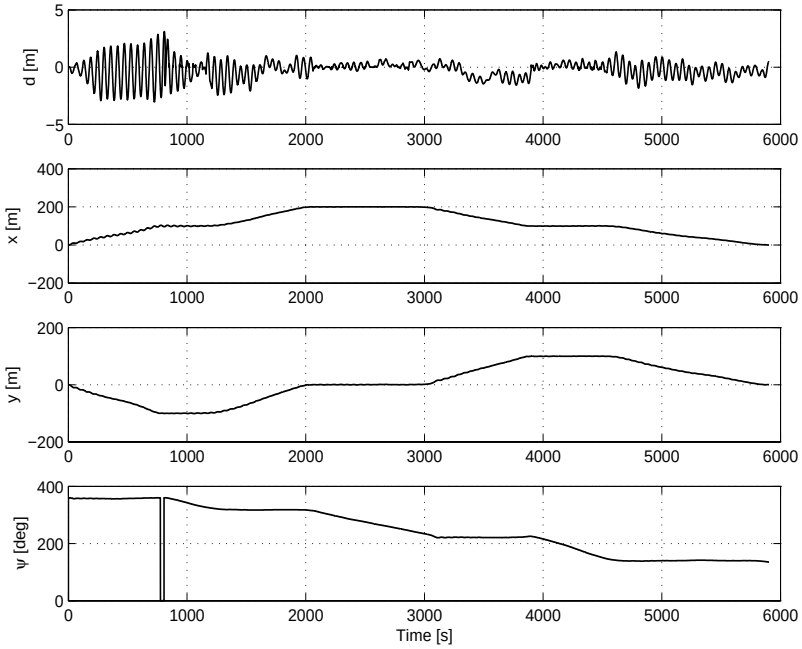
Case 2: The point H is located on the bow of the ship and follows the desired track while ship heading on each segment is set to 315^o, 45^o, 135^o, 225^o. In this case, position of H in vessel-fixed reference coordinate is chosen as: $\Delta x = L/2$, $\Delta y = 0$. The initial position and heading of ship is (-L/2, 0) and 0^o. The plot of the ship positions is shown in Fig. 6.6. In low-speed maneuvering, the effects of disturbances on ship are considerable. However, in this simulation the bow of the ship can follow the desired track while ship heading on each segment is maintained at the desired value.

Case 3: Similar to the case 2 but the ship heading on each segment is set to 0^o, 90^o, 180^o, 270^o. This case is more difficult than case 2 because ship moves in a direction different from her heading. The plot of the ship positions is shown in Fig. 6.7.

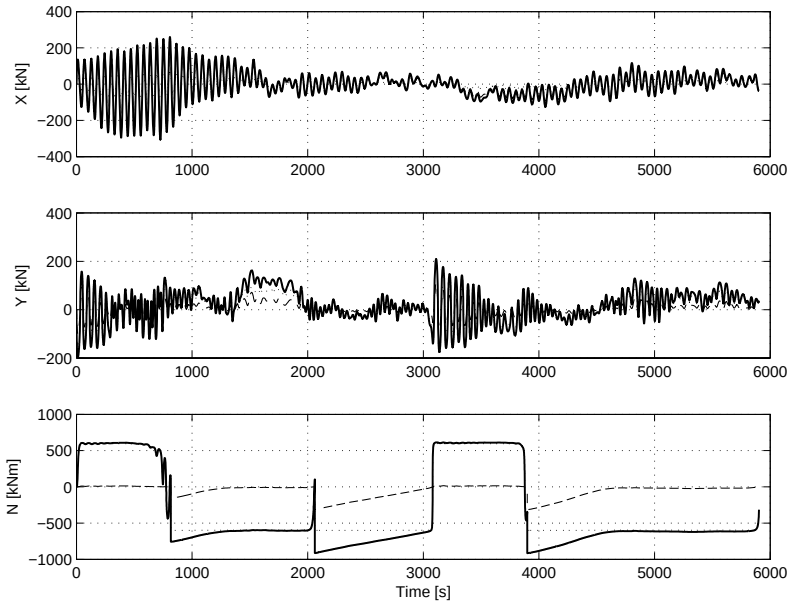
Similar to the simulation of case 2, ship's bow in the case 3 can follow the desired track, and the desired heading is still maintained. In both cases 2 and 3, at each marked point the ship's bow is stationed at the point so that ship heading changes to new value before moving along new segment. Clearly, this action is also resulted in by the effect of new item $e^{-\chi_4|\Delta\psi|}$ in (6.23).



(a) xy plot of ship position in low-speed maneuvering simulation.

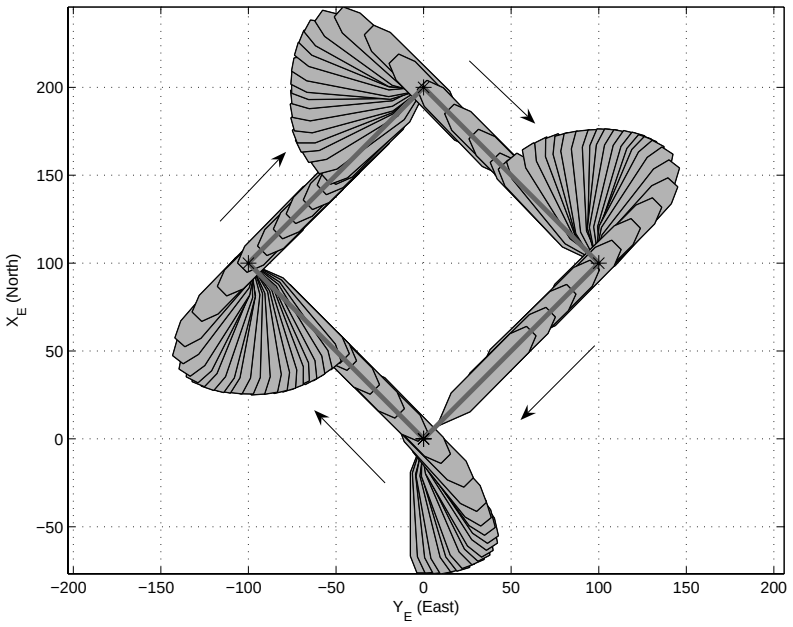


(b) Off-track distance, position in x, y, and ship heading.

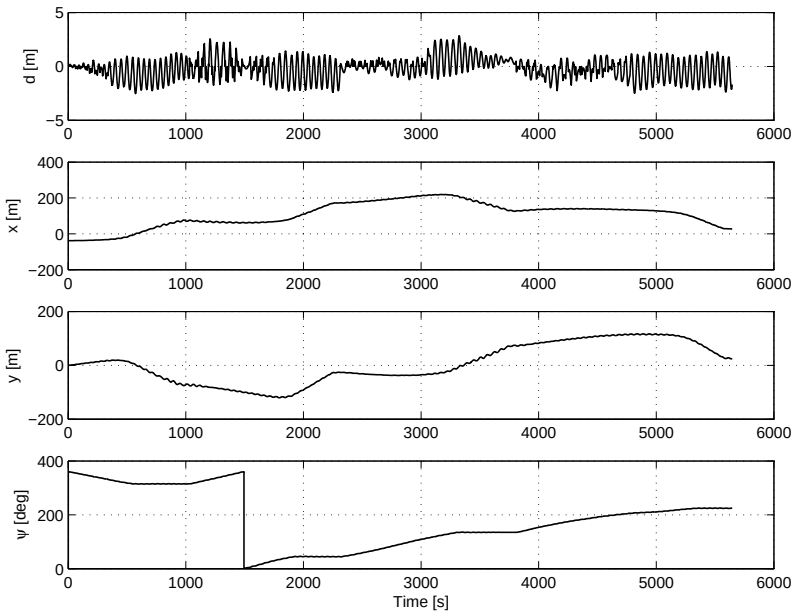


(c) Control output of controllers. ANNAI controllers (dotted), PD-controller (dashed), and hybrid neural adaptive controller (solid).

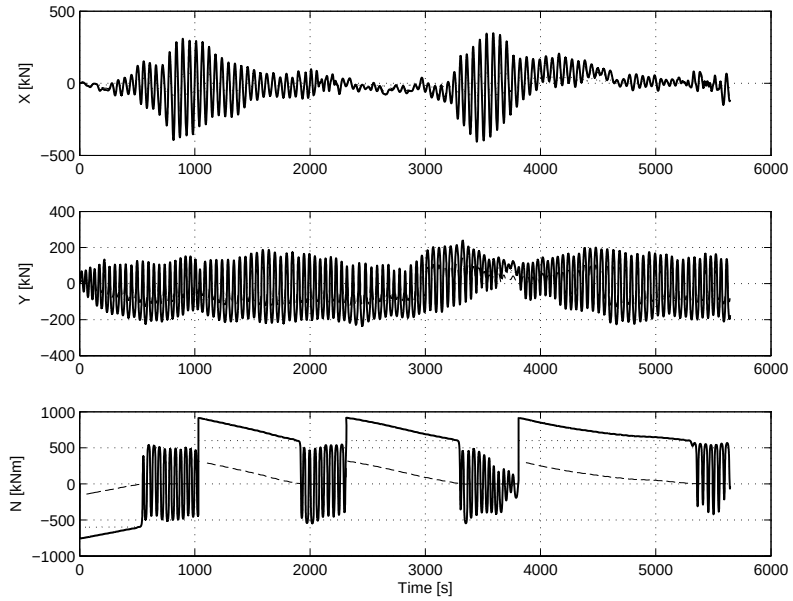
Fig. 6.5 Low-speed maneuvering simulation result of case 1. The desired track connecting four marked points is gray line



(a) xy plot of ship position in low-speed maneuvering simulation.

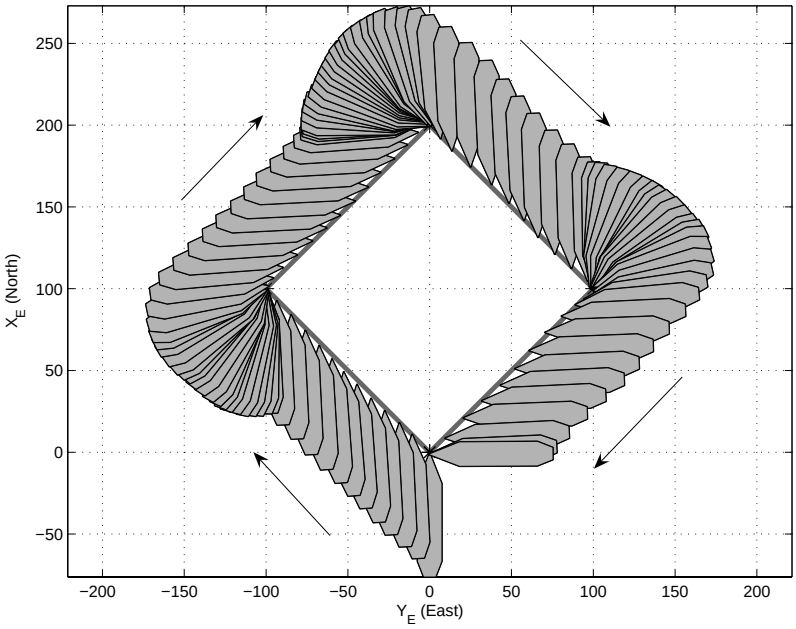


(b) Off-track distance, position in x, y, and ship heading.

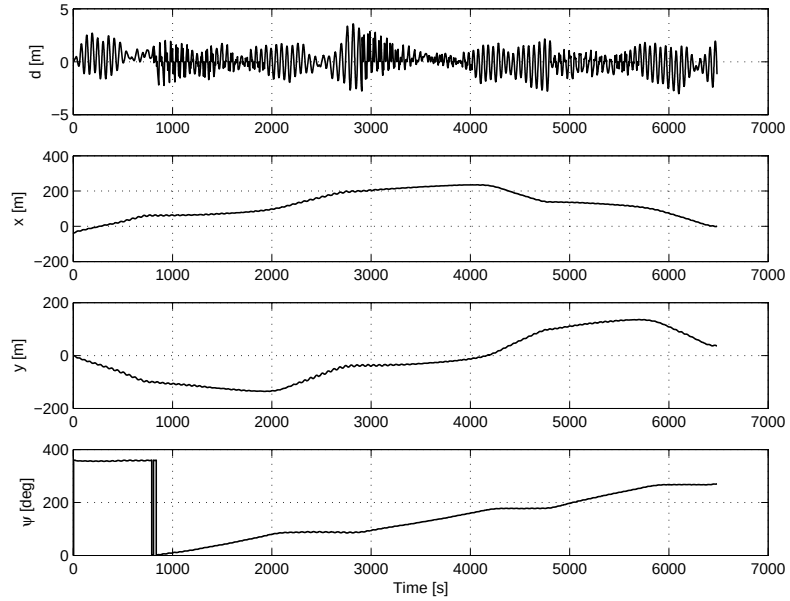


(c) Control output of controllers. ANNAI controllers (dotted), PD-controller (dashed), and hybrid neural adaptive controller (solid).

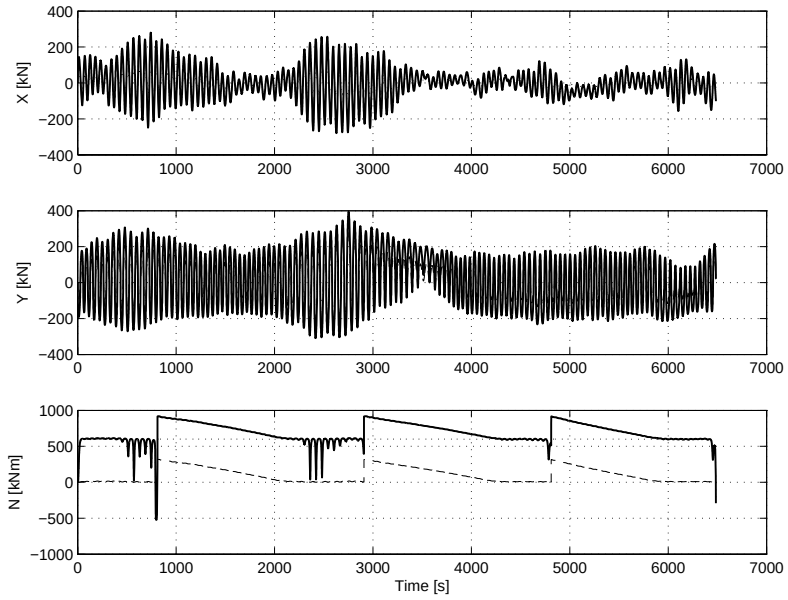
Fig. 6.6 Low-speed maneuvering simulation result of case 2. The desired track connecting four marked points is gray line



(a) xy plot of ship position in low-speed maneuvering simulation.



(b) Off-track distance, position in x, y, and ship heading.



(c) Control output of controllers. ANNAI controllers (dotted), PD-controller (dashed), and hybrid neural adaptive controller (solid).

Fig. 6.7 Low-speed maneuvering simulation result of case 3. The desired track connecting four marked points is gray line

6.4 Conclusion

Practically, it is still difficult to obtain an exact ship model to improve the control performance. To cope with this challenge, this chapter has presented a new hybrid neural adaptive DP system, which is independent of the exact mathematical model of ship, using ANNAI controllers and a conventional PD-controller. The ANNAI controller has been introduced to adaptively compensate for unknown bias term representing slowly-varying environmental disturbances and minimize positioning and tracking error. Computer simulations have been carried out to prove the feasibility of the proposed controller and test its performance. The external disturbance effects have been added to ensure that the proposed control scheme remains stable during operation.

It has been shown in Fig. 6.3 that the performance of the hybrid adaptive neural controller in stabilizing ship at desired position has been improved in comparison with

PD-controller and NNC. The proposed DP system has shown the ability to control dynamically positioned ship in position-keeping as well as various maneuvering situations under the effects of external disturbances. The independence of a mathematical ship model proved that, we can apply the DP system to other ship to do similar tasks without redesign the controller. Additionally, the proposed DP system does not require the estimation of the bias term. So the model error and estimation error can be removed. Furthermore, a method of moving the reference point has been modified and applied to low-speed maneuvering. This method has shown the effectiveness of stabilizing ship at reference point as well as maintaining the desired heading. This ability is useful for specialized tracking functions for supply ships, cable and pipe laying ships.

In practice, if it is possible to design the PD-controller more carefully and the ANNAI controllers' parameters are optimally selected, for example if we apply to a real ship with some prior knowledge of the ship, the control performance can be improved more. Future work will consider the actuator allocation and saturation, as well as the extreme environmental situations. The role of the NNC within the hybrid control scheme will be more closely analyzed, especially the issue of stability will be addressed more comprehensively.

Chapter 7 Conclusions and Recommendations

In this thesis, the overall goal was to develop methodologies which can improve and enhance performance and reliability of marine vehicles utilizing recent developments in the field of adaptive neural network control techniques. A new adaptive neural network controller was developed and applied to heading control of ships. This adaptive neural network controller was then improved and extended to design a track control system, which can adapt with the external disturbances acting on ship. Based on the proposed neural network control scheme, an automatic berthing control system for ship was developed. A similar adaptive neural network control algorithm was applied to design a hybrid neural adaptive controller for dynamic positioning of ship. In this chapter, conclusions are summarized for each topic in chapters 2 – 6. Finally, some recommendations for future research are presented.

7.1 Conclusions

The conclusions for each topic of this research can be summarized as follows.

7.1.1 ANNAI Controller

(1) A new approach of neural network training was further developed in order to apply to automatic ship control. The ANNAI controller can be online-trained.

(2) To enhance the adaptability of the neural network controller, an algorithm for automatic selection of learning rate and number of training iterations was proposed. This algorithm also helped to speed up training speed and maintain

the stability of the control system. It also helped to avoid manual time-consuming selection of neural network parameters by *trial and error* method.

7.1.2 Heading Control System

(1) The proposed ANNAI controller was applied to heading control of ships. In digital control system, the time needed for calculation of control output is very important. The proposed neural network takes less time for calculation of control output in comparison with conventional backpropagation neural network.

(2) The proposed neural network heading control system can cope with new situations, including different ships or environmental conditions. This ability is resulted in by the online training scheme applied to the neural network.

(3) Ship model error and approximation error can be removed. Because the neural network controller can adapt directly without approximating the ship dynamics, and no ship mathematical model was required in designing the controller.

7.1.3 Track Control System

(1) In practice, external disturbances especially sea current, make the ship deviate from the intended track. To enhance the adaptability of the track control system, off-track distance from ship to the intended track was included in the learning process of the neural network controller developed in chapter 2. Employing the learning ability of the neural network, the track control system can adapt with the changes of external disturbances as well as ship dynamics.

(2) Modification of the well-known LOS guidance algorithm was proposed. An alternative method to calculate guidance signal and radius of circle of acceptance R_0 was proposed. This calculation considered the dependence of R_0 on ship turning characteristic and the difference between the present course

and new course.

(3) For visual simulation purpose, module programs written in MATLAB are introduced for guidance and control using Mercator chart. They can be used to calculate and display ship's movement on the navigational equipment monitors such as ECDIS.

7.1.4 Berthing Control System

(1) An automatic berthing control system was developed. The proposed neural network controller was modified to be suitable to control ship heading and speed in low-speed maneuvering in harbor. The off-track distance from ship to intended berthing route was included in ship heading control algorithm. This modification can make the ship tend to get close to the intended track.

(2) A useful berthing guidance algorithm is proposed by modifying the prior works. This algorithm can calculate desired heading and speed for the controllers.

(3) The computer simulations demonstrated that, 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.

7.1.5 Dynamic Positioning System

(1) Practically, it is still difficult to obtain an exact ship model to improve the control performance. To cope with this challenge, a new hybrid neural adaptive DP system was proposed. This DP system is independent of the exact mathematical model of ship, using the proposed neural network controllers in parallel with a conventional PD-controller.

(2) No estimation of external disturbances was required, because the neural network controllers can compensate for the external disturbances by learning

ship dynamics.

(3) The DP system can minimize positioning error and tracking error. For low-speed maneuvering, an algorithm to guide the ship along the intended track and maintain ship heading was proposed.

7.2 Recommendations for Future Research

The results of the present work indicate the following potential topics for future research.

(1) The adaptive neural network controller developed in this study can be applied to other marine control problems such as rudder roll control, floating structures control, and control of under water vehicles.

(2) Extensive study on the selection of the best structure of neural network for each control problems is recommended.

(3) The parameters of the neural network controller in cost function require proper selection to obtain optimal performance. Investigation of possible application of genetic algorithms to find optimal values is recommended for further study.

(4) The improvement in hybrid control algorithm suggests that, we can employ the advantages of conventional control methods and neural network control, as well as other intelligent control. Future work will consider the combination of these control methods in designing marine control systems.

(5) The role of the neural network controller within the hybrid control scheme will be more closely analyzed, especially the issue of stability will be addressed more comprehensively in further study.

References

[1] A. Wahl and E.D. Gilles (1998). Track-keeping on waterways using model predictive control. Control Applications in Maritime Systems, A proceedings volume from the IFAC Conference, Fukuoka, Japan, 27-30 Oct/1998, pp. 149-154 (Edited by K. Kijima and T.I. Fossen).

[2] A. Loris, T.I. Fossen, and E. Panteley (2000). A separation principle for dynamic positioning of ships: theoretical and experimental results. IEEE Transactions on Control Systems Technology, Vol. 8, No. 2, pp. 332-343.

[3] A.J. Calise, N. Hovakimyan, and M. Idan (2001). Adaptive output feedback control of nonlinear systems using neural networks. Elsevier 2001.

[4] A.J. Sørensen, S.I. Sagatun, and T.I. Fossen (1996). Design of a dynamic positioning system using model-based control. Control Eng. Practice, Vol. 4, No. 3, pp. 359-368.

[5] B. Bavarian (1988). Introduction to neural networks for intelligent control. 1988 IEEE Transactions on Neural Networks.

[6] B.K. Lee (2005). An optimal route decision and LOS guidance system for automatic navigation of ships. PhD. thesis, Department of Control & Instrumentation Engineering, Graduate school, Korea Maritime University.

[7] C.Y. Tzeng, K.F. Lin (1998). Adaptive ship steering autopilot design with saturating and slew rate limiting actuator. Control Applications in Maritime Systems, A proceedings volume from the IFAC Conference, Fukuoka, Japan, 27-30 Oct/1998, pp. 5-13 (Edited by K. Kijima and T.I. Fossen).

[8] C.K. Chak, G. Feng, J. Ma, and M. Palaniswami (1997). Universal neural network controllers. 1997 IEEE Transactions on Neural Networks.

[9] C.V. Nguyen, L. Morawski, and A. Rak (2006). Track keeping autopilot with fuzzy logic controller. Proc. of the 6th Asian Conf. on Marine Simulator and Simulation Research, Hai Phong, Vietnam, pp. 137-144.

[10] D.S. Desanj, D.C. Donha, M.R. Katebi, and M.J. Grimble (1997). H-inf adaptive controllers for autopilot applications. Eleventh Ship Control Systems Symposium Vol. 1 (Edited by P.A. Wilson), pp. 265-279.

[11] D. Bertin (1998). Track-keeping controller for a precision manoeuvring autopilot.

Control Applications in Maritime Systems, A proceedings volume from the IFAC Conference, Fukuoka, Japan, 27-30 Oct/1998, pp. 143-148 (Edited by K. Kijima and T.I. Fossen).

[12] D.M. Skapura (1996). Building neural networks. Addison-Wesley Publishing Company. ISBN 0-201-53921-7.

[13] D.H. Nguyen (2000). Self-tuning pole assignment and optimal control systems for ships. PhD. thesis, Tokyo University of Mercantile Marine, Tokyo, Japan.

[14] D.T. Pham and X. Liu (1995). Neural networks for identification, prediction and control. Springer-Verlag London Ltd. ISBN 3-540-19959-4.

[15] E. Tulunay (1991). Introduction to neural networks and their applications to process control. Neural Networks: Advances and Applications, pp. 241-273 (E. Gelenbe edited). Elsevier Science Publishers B.V. (North-Holland).

[16] E.W. McGookin, D.J.M. Smith, Y. Li, and T.I. Fossen (1999). Ship steering control system optimization using genetic algorithms. Elsevier, Control Engineering Practice 8, pp. 429-443.

[17] F.J. Lin and R.J. Wai (1999). Intelligent control for ultrasonic motor drive. Intelligent Adaptive Control: Industrial Applications, pp. 277-309 (L.C. Jain, C.W. de Silva edited). CAC Press LLC. ISBN 0-8493-9805-3.

[18] F. Lin, R.D. Brandt, and G. Saikalis (2000). Seft-tuning of PID controllers by adaptive interaction. Proceedings of the American Control Conference, Chicago, Illinois, June 2000.

[19] F.L. Lewis, S. Jagannathan, and A. Yesildirek (1999). Neural Network Control of Robot Manipulators and Nonlinear Systems. Taylor & Francis. ISBN 0-7484-0596-8.

[20] G.E. Hearn, Y. Zhang, and P. Sen (1997). Alternative designs of neural network based autopilots: a comparative study. Manoeuvring and Control of Marine Craft, A proceedings volume from the IFAC Conference, Brijuni, Croatia, 10-12 Sep/1997, pp. 83-88 (Edited by Z. Vukic and G.N. Roberts).

[21] G. Hardier (1997). Recurrent neural networks for ship modeling and control. Eleventh Ship Control Systems Symposium Vol. 1 (Edited by P.A. Wilson), pp. 39-61.

[22] G.N. Roberts, R. Sutton, A. Zirilli, and A. Tiano (2003). Intelligent ship autopilots-a historical perspective. Elsevier, Mechatronics 13, pp. 1091-1103.

[23] G. Saikalis and F. Lin (2001). A neural network controller by adaptive interaction. Proceeding of the American Control Conference, Arlington, pp. 1247-1252.

[24] H.D. Patino and D. Liu (2000). Neural network-based model reference adaptive control system. IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 30 No.1.

[25] H. Wang, G.P. Liu, C.J. Harris, and M. Brown (1995). Advanced adaptive control. Elsevier Science Ltd. ISBN 0-08-042020-6.

[26] H.N. Teodorescu and C. Bonciu (1999). Feature space neural filters and controllers. Intelligent Adaptive Control: Industrial Applications, pp. 105-147 (L.C. Jain, C.W. de Silva edited). CAC Press LLC. ISBN 0-8493-9805-3.

[27] H. Tamaru, H. Hagiwara, H. Yoshida, T. Tasaki, and H. Miyabe (2005). Development of automatic berthing system for Kaisho Maru and its performance evaluation. The Journal of Japan Institute of Navigation, Vol. 113, pp. 157-164.

[28] J. Jiya, C. Shao, and T.Y. Chai (1999). Neural network-based adaptive predictive control algorithm of nonlinear non-minimum phase systems. Proceedings of the American Control Conference, pp. 1082-1085. San Diego, California, June 1999.

[29] J.C. Principe, N.R. Euliano, and W.C. Lefebvre (2000). Neural and adaptive systems: fundamentals through simulations. John Willey & Sons, Inc. ISBN 0-471-35167-9.

[30] J.K. Wu (1994). Neural networks and simulation methods. Marcel Dekker, Inc., ISBN 0-8247-9181-9.

[31] J.S. Kim (2002). A study on the sensorless speed control of induction motor by neural network. PhD. thesis, Graduate School of Korea Maritime University.

[32] J. Velagic, Z. Vukiz, and E. Omerdic (2003). Adaptive Fuzzy Ship Autopilot for Track-keeping. Control Engineering Practice 11, pp. 433-443.

[33] J.Y. Koo (1995). On the ship's berthing control using fuzzy neural network. PhD. thesis, Graduate School of Korea Maritime University.

[34] J.W. Hines (1997). Matlab supplement to fuzzy and neural approaches in engineering. John Willey & Sons, Inc., ISBN 0-471-19247-3.

[35] K.D. Do, Z.P. Jiang, and J. Pan (2004). Robust adaptive path following of underactuated ships. Automatica 40, pp. 929-944.

[36] K.J. Astrom and B. Wittenmark (1995). Adaptive control. Addison-Wesley Publishing Company, Inc. ISBN 0-201-55866-1.

[37] K.E. Husa and T.I. Fossen (1997). Backstepping designs for nonlinear way-point tracking of ships. Manoeuvring and Control of Marine Craft, A proceedings volume from the IFAC Conference, Brijuni, Croatia, 10-12 Sep/1997, pp. 111-116 (Edited by Z. Vukic and G.N. Roberts).

[38] K. Djouani and Y. Haman (1996). Feedback optimal neural network controller for dynamic systems-A ship maneuvering example. Elsevier, Mathematics and Computaters in Simulation 41, pp. 117-127.

[39] K.M. Passino (2005). Biomimicry for optimization, control, and automation. Springer, ISBN 1-85233-804-0.

[40] K.S. Narendra (1990). Adaptive control using neural networks. Neural Networks for Control (W.T. Miller, R.S. Sutton and P.J. Werbos edited). The MIT Press. ISBN 0-262-13261-3.

[41] K.S. Narendra and K. Parthasarathy (1990). Identification and control of dynamical systems using neural networks. IEEE Transactions on Neural Networks, Vol. 1, No. 1, pp. 4-27.

[42] L.H. Tsoukalas and R.E. Uhrig (1997). Fuzzy and neural approaches in engineering. John Willey & Sons, Inc., ISBN 0-471-16003-2.

[43] L.G. Kraft, III, and D.P. Campagna (1990). A summary comparison of CMAC neural network and traditional adaptive control systems. Neural Networks for Control (W. Thomas Miller, Richard S. Sutton and Paul J. Werbos edited). The MIT Press. ISBN 0-262-13261-3.

[44] L. Morawski and J. Pomirski (1998). Ship track-keeping: experiments with physical tanker model. Elsevier, Control Engineering Practice 6, pp. 763-769.

[45] L. Morawski, C.V. Nguyen, and A. Rak (2006). The fuzzy control of vessel's motion in manoeuvring situations. Proc. of the 6th Asian Conf. on Marine Simulator and Simulation Research, Hai Phong, Vietnam, pp. 130-136.

[46] L.X. Wang (1997). A course in fuzzy systems and control. Prentice-Hall International, Inc., ISBN 0-13-593005-7.

[47] M. Breivik and T.I. Fossen (2004). Path following of straight lines and circles for marine surface vessels. Proceedings of the 6th IFAC CAMS, Ancona, Italy 2004.

[48] M. Breivik (2003). Nonlinear maneuvering control of underactuated ships. MSc thesis. Dept. of Eng. Cybernetics, Nowegian University of Science and Technology.

[49] M. Sandler, A. Wahl, R. Zimmermann, M. Faul, U. Kabatek, and E.D. Gilles (1996). Autonomous guidance of ships on waterway. Elsevier, Robotic and Autonomous Systems 18, pp. 327-335.

[50] M.B. McFarland and A.J. Calise (1997a). Multilayer neural networks and adaptive nonlinear control of agile anti-air missiles. Guidance, Navigation and Control Conference, American Institute of Aeronautics and Astronotics, August, 1997. Paper No. 97-3540.

[51] M.B. McFarland and A.J. Calise (1997b). Robust adaptive control of uncertain nonlinear systems using neural networks. American Control Conference, Albuquerque, New Mexico, June 1997.

[52] M.H. Kim (2000). Nonlinear control and robust observer design for marine vehicles. PhD. thesis. Virginia Polytechnic Institute and State University.

[53] M. Nogaard, O. Ravn, N.K. Poulsen, and L.K. Hansen (2003). Neural networks for modeling and control of dynamic systems. Springer, ISBN 1-85233-227-1.

[54] N.K. Im and K. Hasegawa (2001). A study on automatic ship berthing using parallel neural controller. J. Kansai Soc. N.A., Japan, No. 236, pp. 65-70.

[55] N.K. Im and K. Hasegawa (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.

[56] O. Constantin (2003). Adaptive neural predictive techniques for nonlinear control. Studies in Informatics and Control, Vol. 12, No.4, December 2003, pp. 285-291.

[57] P.H. Nguyen (2005). The theory and applications in automatic ship's control of neural networks. Research Report, Dept. of Ship Operation Systems Eng., Korea Maritime University.

[58] P.H. Nguyen and Y.C. Jung (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, 2005 (ISSN-1598-5725).

[59] P.H. Nguyen and Y.C. Jung (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, 2006 (ISSN-1598-5725).

[60] P.H. Nguyen and Y.C. Jung (2006b). Improved adaptive neural network autopilot

for track-keeping control of ships: Design and Simulation. International Journal of Navigation and Port Research (KINPR), Vol.30, No.4 pp. 259~265, 2006 (ISSN-1598-5725).

[61] P.H. Nguyen and Y.C. Jung (2006c). A hybrid neural adaptive controller for dynamic positioning of ship. Submitted to IEEE Transactions on Control Systems Technology.

[62] P.H. Nguyen and Y.C. Jung (2006d). A Study on Automatic Berthing Control of Ship Using Adaptive Neural Network Controller. Proceedings of Korean Institute of Navigation and Port Research Conference, Seoul, Korea (2006/Jun), pp. 67-74.

[63] Q.M.J. Wu, K. Stanley, and C.W. de Silva (1999). Neural control systems and applications. Intelligent Adaptive Control: Industrial Applications, pp. 63-103 (L.C. Jain, C.W. de Silva edited). CAC Press LLC. ISBN 0-8493-9805-3.

[64] R. Powlowicz's M_Map Toolbox for MATLAB [Online]. Available: http://www2.ocgy.ubc.ca/~rich/map.html, accessed 2005/Dec.

[65] R. Burns and R. Richter (1996). A neural network approach to the control of surface ships. Elsevier, Control Engineering Practice, Vol. 4, No. 3, pp. 411-416.

[66] R. Sutton and G.N. Roberts (1997). Approaches to fuzzy autopilot design optimization. Manoeuvring and Control of Marine Craft, A proceedings volume from the IFAC Conference, Brijuni, Croatia, 10-12 Sep/1997, pp. 77-82 (Edited by Z. Vukic and G.N. Roberts).

[67] R. Richter, R.S. Burns, M.N. Polkinghorne, and P. Nurse (1997). A predictive ship control using a fuzzy-neural autopilot. Eleventh Ship Control Systems Symposium Vol. 1 (Edited by P.A. Wilson), pp. 161-172.

[68] R. Sutton (1997). Optimization of fuzzy autopilots. Eleventh Ship Control Systems Symposium Vol. 1 (Edited by P.A. Wilson), pp. 63-76.

[69] R.P. Lippmann (1987). An introduction to computing with neural nets. 1987 IEEE Transactions on Neural Networks.

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

[71] R.R. Bitmead, M. Gevers, and V. Wertz (1990). Adaptive optimal control: The Thinking Man’s GPC. Prentice Hall. ISBN 0-13-012377-2.

[72] S. Jaganathan and F.L. Lewis (1999). Discrete-time neural network control of nonlinear systems. Intelligent Adaptive Control: Industrial Applications, pp. 149-

176 (L.C. Jain, C.W. de Silva edited). CAC Press LLC. ISBN 0-8493-9805-3

[73] S. Shekhar, M.B. Amin, and P. Khandelwal (1992). Neural networks: advances and applications, Vol. 2, pp. 13-38 (E. Gelenbe edited). Elsevier Science Publishers B.V. (North-Holland).

[74] S.S. Ge and C. Wang (2002). Direct adaptive neural network control of a class of nonlinear systems. IEEE Transactions on Neural Networks, Vol. 13, No. 1.

[75] S. Haykin (1999). Neural networks: a comprehensive foundation. Prentice-Hall, Inc. ISBN 0-13-273350-1.

[76] S.P. Berge, K. Ohtsu, and T.I. Fossen (1998). Nonlinear control of ships minimizing the position tracking errors. Control Applications in Maritime Systems, A proceedings volume from the IFAC Conference, Fukuoka, Japan, 27-30 Oct/1998, pp. 129-134 (Edited by K. Kijima and T.I. Fossen).

[77] T.I. Fossen, M. Breivik, and R. Skjetne (2003). Line-of-sight path following of underactuated marine craft. Proceedings of the 6th IFAC MCMC, Girona, Spain 2003, pp. 244-249.

[78] T.I. Fossen (1994). Guidance and control of ocean vehicles. John Wiley & and Sons Ltd. ISBN 0-471-94113-1.

[79] T.I. Fossen (2002). Marine control systems: guidance, navigation and control of ships, rigs and underwater vehicles. Marine Cybernetics, Trondheim, Norway. ISBN 82-92356-00-2.

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

[81] T.I. Fossen and J.P. Strand (1999). Passive nonlinear observer design for ships using Lyapunov methods: Full-scale experiments with a supply vessel. Automatica 35, pp. 3-16.

[82] T.I. Fossen and J.P. Strand (2001). Nonlinear passive weather optimal positioning control (WOPC) system for ships and rigs: Experimental results. Automatica 37, pp. 701-715.

[83] T.I. Fossen and Å. Grøvlen (1998). Nonlinear output feedback control of dynamically positioned ships using vectorial observer backsteping. IEEE Transactions on Control Systems Technology, Vol. 6, No. 1, pp. 121-128.

[84] T.J. Ross (1995). Fuzzy logic with engineering applications. McGraw-Hill, Inc., ISBN 0-07-053917-0.

[85] T.H. Lee, Y. Cao, and Y.M. Lin (2001). Application of an on-line training predictor/controller to dynamic positioning of floating structures. Tamkang Journal of Science and Engineering, Vol.4, No.3, pp. 141-154.

[86] Y.J. Yoon and S.H. Jeon (2005). Terrestrial Navigation (in Korean). Korea Maritime University.

[87] Y. Tan and R. de Keyser (1994). Neural network-based adaptive predictive control. Advances in Model-Based Predictive Control, pp. 358-369. (Edited by D. Clarke). Oxford University Press. ISBN 0198562926.

[88] Y. Yang, C. Zhou, and J. Ren (2003). Model reference adaptive robust fuzzy control for ship steering autopilot with uncertain nonlinear systems. Elsevier, Applied Soft Computing 3, pp. 305-316.

[89] Y. Zhang, G.E. Hearn, and P. Sen (1997a). Neural network approaches to a class of ship control problems (Part I: Theoretical design). Eleventh Ship Control Systems Symposium Vol. 1 (Edited by P.A. Wilson), pp. 115-133.

[90] Y. Zhang, G.E. Hearn, and P. Sen (1997b). Neural network approaches to a class of ship control problems (Part II: Simulation studies). Eleventh Ship Control Systems Symposium Vol. 1 (Edited by P.A. Wilson), pp. 135-150.

[91] Y. Zhuo and G.E. Hearn (2004). Specialized learning for ship intelligent track-keeping using neurofuzzy. Proceedings of IFAC-CAMS, Ancona, Italy, pp. 291-296.

[92] Z. Vukic, E. Omerdic, and L. Kuljaca (1997). Fuzzy autopilot for ships experiencing shallow water effect in manoeuvring. Manoeuvring and Control of Marine Craft, A proceedings volume from the IFAC Conference, Brijuni, Croatia, 10-12 Sep/1997, pp. 99-104 (Edited by Z. Vukic and G.N. Roberts).

[93] Z. Vukic, E. Omerdic, and L. Kuljaca (1998). Improved fuzzy autopilot for track-keeping. Control Applications in Maritime Systems, A proceedings volume from the IFAC Conference, Fukuoka, Japan, 27-30 Oct/1998, pp. 123-128 (Edited by K. Kijima and T.I. Fossen).

[94] Z. Vukic and B. Borovic (2000). Guidance and control systems for marine vehicles. The Ocean Engineering Handbook, pp. 1.14-1.33. CLC Press LLC. ISBN: 0849385989.

Appendix A

Mathematical Model of Dynamic Positioning Ships

This Appendix presents a brief mathematical model for dynamic positioning of ships based on [79].

A.1 Equations of Motion

The earth-fixed position (*x*, *y*) and heading ψ of the vessel relative to an earth-fixed coordinate $X_E Y_E Z_E$ are expressed in vector form by $\eta = [x, y, \psi]^T$, and the vessel-fixed linear velocity vector is expressed by $\nu = [u, v, r]^T$. These three modes are referred to as the *surge*, *sway* and *yaw* modes of a ship. The origin of the vessel-fixed coordinate *XYZ* is located at the vessel center line in a distance x_G from the center of gravity. The low frequency motion of DP ships in surge, sway, and yaw can be described as follow

$$M\dot{\nu} + D\nu = \tau + J^T(\eta)b\,, \tag{A.1}$$

$$\dot{\eta} = J(\eta)\nu\,. \tag{A.2}$$

Here, $\tau = [\tau_1, \tau_2, \tau_3]^T$ is a control vector of forces and moment provided by the propulsion system. $M \in \Re^{3x3}$ is the inertia matrix including hydrodynamic added inertia, and $D \in \Re^{3x3}$ is the damping matrix.

$$M = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & mx_G - Y_{\dot{r}} \\ 0 & mx_G - N_{\dot{v}} & I_z - N_{\dot{r}} \end{bmatrix}, \tag{A.3}$$

$$D = \begin{bmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & mu_0 - Y_r \\ 0 & -N_v & mx_G u_0 - N_r \end{bmatrix}, \tag{A.4}$$

where m is the mass, I_z is the moment of the ship about the vessel-fixed z-axis, $X_{\dot{u}}, Y_{\dot{v}}, Y_{\dot{r}}, N_{\dot{v}}, N_{\dot{r}}$ are added inertia, X_u, Y_v, N_v, N_r are linear damping forces and moment, and u_0 is the nominal velocity of the ship.

Unmodeled external forces and moment due to wind, currents and waves are lumped together into an earth-fixed constant (or slowly-varying) bias term $b \in \Re^3$, $J(\eta)$ is the transformation matrix between the earth-fixed coordinate and the vessel-fixed coordinate. The transformation matrix has the following form

$$J(\eta) = J(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{A.5}$$

where $J(\psi)$ is nonsingular for all ψ and $J^{-1}(\psi) = J^T(\psi)$.

A.2 Bias Modeling

A common model for the bias forces in surge, sway and yaw moment for marine vehicle control application is

$$\dot{b} = -T^{-1}b + \Psi n, \tag{A.6}$$

where $b \in \Re^3$ is a vector of bias forces and moment, n is a vector of zero-mean Gaussian white noise, T is a diagonal matrix of positive bias time constants and $\Psi \in \Re^{3x3}$ is a diagonal matrix scaling the amplitude of n. This model can be used to describe slowly-varying environmental forces and moments due to 2nd order wave loads, ocean currents, wind and unmodeled dynamics.

A.3 Wave Force Modeling

Wave forces can be divided into 1st-order wave disturbances and 2nd-order wave drift forces. For the practical application to control system design, the 1st-order wave disturbances can be described by three harmonic oscillators with some damping. Linear 2nd order wave forces are generally expressed as

$$\dot{\xi} = A\xi + Ew, \tag{A.7}$$

$$\eta_w = C\xi, \tag{A.8}$$

where $\eta_w = [x_w, y_w, \psi_w]^T$, $\xi \in \Re^6$, and $w \in \Re^3$ is a zero means bounded disturbance vector and

$$A = \begin{bmatrix} 0 & I \\ \Omega_{21} & \Omega_{22} \end{bmatrix}, \; E = \begin{bmatrix} 0 \\ \Sigma_2 \end{bmatrix}, \; C = \begin{bmatrix} 0 & I \end{bmatrix}, \tag{A.9}$$

where

$$\Omega_{21} = -diag\{\omega_{01}^2, \omega_{02}^2, \omega_{03}^2\},$$

$$\Omega_{22} = -diag\{2\zeta_1\omega_{01}, 2\zeta_2\omega_{02}, 2\zeta_3\omega_{03}\},$$

$$\Sigma_2 = diag\{\sigma_1, \sigma_2, \sigma_3\}.$$

Here ω_{oi}, ζ_i, and σ_i (i = 1, …, 3) are wave frequency, relative damping ratio and parameters related to wave intensity, respectively.

A.4 Measurement Systems

For conventional ships, positions and yaw angles are usually measured by global positioning system (GPS) or hydroacoustic positioning reference (HPR) systems, and

gyro compasses. However, for ship positioning systems the *differential* GPS is usually applied to reduce positioning errors. The measurement can be written as

$$y = \eta + \eta_w + v\,, \tag{A.10}$$

where $v \in \Re^3$ is the zero mean Gaussian white measurement noise. It is assumed that the total position of the ship can be obtained by superposition of the position and direction of the ship and the wave displacements.

Appendix B

Parameters used in the Simulations

B.1 Mariner Class Vessel

Both planar motion mechanism tests and full-scale steering and maneuvering predictions for this *Mariner Class Vessel* were performed by the hydro-aerodynamics laboratory in Lyngby, Denmark. The main data and dimensions of the *Mariner Class Vessel* are shown in [78]

Table B.1 Main dimensions of Mariner Class Vessel

Length overall (LOA)	171.80	m
Length between perpendiculars (LPP)	160.93	m
Maximum beam (B)	23.17	m
Design draft (T)	8.23	m
Design displacement (∇)	18541	m3
Design speed (u_0)	15	knots

Matlab M-File for Nonlinear Model of Mariner Class Vessel

```
function [xdot,U] = mariner(x,ui,U0)
% [xdot,U] = mariner(x,ui) returns the speed U in m/s (optionally) and the
% time derivative of the state vector: x = [ u v r x y psi delta n ]'  for
% the Mariner class vessel L = 160.93 m, where
%
% u     = perturbed surge velocity about Uo (m/s)
% v     = perturbed sway velocity about zero (m/s)
% r     = perturbed yaw velocity about zero (rad/s)
% x     = position in x-direction (m)
% y     = position in y-direction (m)
% psi   = perturbed yaw angle about zero (rad)
```

```
% delta = actual rudder angle (rad)
%
% The inputs are :
%
% ui    = commanded rudder angle (rad)
% U0    = nominal speed (optionally). Default value is U0 = 7.7175 m/s = 15
% knots.
%
% Reference: M.S. Chislett and J. Stroem-Tejsen (1965). Planar Motion
%            Mechanism Tests and Full-Scale Steering and Maneuvering
%            Predictions for a Mariner Class Vessel,Technical Report Hy-5,
%            Hydro- and Aerodynamics Laboratory, Lyngby, Denmark.
%
% Author:    Trygve Lauvdal
% Date:      12th May 1994
% Revisions: 19th July 2001 (Thor I. Fossen): added input/ouput U0 and U,
%                           changed order of x-vector
%            20th July 2001 (Thor I. Fossen): replaced inertia matrix with
%                           correct values
%            11th July 2003 (Thor I. Fossen): max rudder is changed from 30
%                           deg to 40 deg to satisfy IMO regulations for 35 deg
%                           rudder execute

% Check of input and state dimensions
if (length(x)  ~= 7),error('x-vector must have dimension 7 !'); end
if (length(ui) ~= 1),error('ui must be a scalar input!'); end
if nargin==2, U0 = 7.7175; end

% Normalization variables
L = 160.93;
U = sqrt((U0 + x(1))^2 + x(2)^2);

% Non-dimensional states and inputs
delta_c = -ui;   % delta_c = -ui such that positive delta_c -> positive r

u     = x(1)/U;
v     = x(2)/U;
r     = x(3)*L/U;
psi   = x(6);
delta = x(7);

% Parameters, hydrodynamic derivatives and main dimensions
delta_max  = 35;           % max rudder angle      (deg)
Ddelta_max = 2.5;          % max rudder derivative (deg/s)

m  = 798e-5;
Iz = 39.2e-5;
xG = -0.023;
```

```
Xudot =  -42e-5;   Yvdot =  -748e-5;   Nvdot = 4.646e-5;
Xu    = -184e-5;   Yrdot =-9.354e-5;   Nrdot = -43.8e-5;
Xuu   = -110e-5;   Yv    = -1160e-5;   Nv    =  -264e-5;
Xuuu  = -215e-5;   Yr    =  -499e-5;   Nr    =  -166e-5;
Xvv   = -899e-5;   Yvvv  = -8078e-5;   Nvvv  =  1636e-5;
Xrr   =   18e-5;   Yvvr  = 15356e-5;   Nvvr  = -5483e-5;
Xdd   =  -95e-5;   Yvu   = -1160e-5;   Nvu   =  -264e-5;
Xudd  = -190e-5;   Yru   =  -499e-5;   Nru   =  -166e-5;
Xrv   =  798e-5;   Yd    =   278e-5;   Nd    =  -139e-5;
Xvd   =   93e-5;   Yddd  =   -90e-5;   Nddd  =    45e-5;
Xuvd  =   93e-5;   Yud   =   556e-5;   Nud   =  -278e-5;
                   Yuud  =   278e-5;   Nuud  =  -139e-5;
                   Yvdd  =    -4e-5;   Nvdd  =    13e-5;
                   Yvvd  =  1190e-5;   Nvvd  =  -489e-5;
                   Y0    =    -4e-5;   N0    =     3e-5;
                   Y0u   =    -8e-5;   N0u   =     6e-5;
                   Y0uu  =    -4e-5;   N0uu  =     3e-5;

% Masses and moments of inertia
m11 = m-Xudot;
m22 = m-Yvdot;
m23 = m*xG-Yrdot;
m32 = m*xG-Nvdot;
m33 = Iz-Nrdot;

% Rudder saturation and dynamics
if abs(delta_c) >= delta_max*pi/180,
   delta_c = sign(delta_c)*delta_max*pi/180;
end
delta_dot = delta_c - delta;
if abs(delta_dot) >= Ddelta_max*pi/180,
   delta_dot = sign(delta_dot)*Ddelta_max*pi/180;
end

% Forces and moments
X = Xu*u + Xuu*u^2 + Xuuu*u^3 + Xvv*v^2 + Xrr*r^2 + Xrv*r*v + Xdd*delta^2 +...
    Xudd*u*delta^2 + Xvd*v*delta + Xuvd*u*v*delta;

Y = Yv*v + Yr*r + Yvvv*v^3 + Yvvr*v^2*r + Yvu*v*u + Yru*r*u + Yd*delta + ...
    Yddd*delta^3 + Yud*u*delta + Yuud*u^2*delta + Yvdd*v*delta^2 + ...
    Yvvd*v^2*delta + (Y0 + Y0u*u + Y0uu*u^2);

N = Nv*v + Nr*r + Nvvv*v^3 + Nvvr*v^2*r + Nvu*v*u + Nru*r*u + Nd*delta + ...
    Nddd*delta^3 + Nud*u*delta + Nuud*u^2*delta + Nvdd*v*delta^2 + ...
    Nvvd*v^2*delta + (N0 + N0u*u + N0uu*u^2);

% Dimensional state derivative
detM22 = m22*m33-m23*m32;
```

```
xdot = [            X*(U^2/L)/m11
        -(-m33*Y+m23*N)*(U^2/L)/detM22
         (-m32*Y+m22*N)*(U^2/L^2)/detM22
           (cos(psi)*(U0/U+u)-sin(psi)*v)*U
           (sin(psi)*(U0/U+u)+cos(psi)*v)*U
                    r*(U/L)
                    delta_dot                  ];
```

B.2 Container Ship

A mathematical model for a single-screw height-speed container ship in surge, sway, roll, and yaw is shown in [78]. The main data of the ship model is presented below.

Table B.2 Main dimensions of Container Ship

Length (L)	175.00	m
Breadth (B)	25.40	m
Draft fore (d_F)	8.00	m
aft (d_A)	9.00	m
mean (d)	8.50	m
Displacement volume	21,222	m^3
Height from keel to transverse metacenter (KM)	10.39	m
Height from keel to center of buoyancy (KB)	4.6154	m
Block coefficient (C_B)	0.559	
Rudder area (A_R)	33.0376	m^2
Aspect ratio (Λ)	1.8219	
Propeller diameter (D)	6.533	m

Matlab M-File for Nonlinear Model of Container Ship

```
function [xdot,U] = container(x,ui)
% [xdot,U] = container(x,ui) returns the speed U in m/s (optionally) and the
% time derivative of the state vector: x = [ u v r x y psi p phi delta n ]'
% for a container ship L = 175 m, where
```

```
%
% u     = surge velocity          (m/s)
% v     = sway velocity           (m/s)
% r     = yaw velocity            (rad/s)
% x     = position in x-direction (m)
% y     = position in y-direction (m)
% psi   = yaw angle               (rad)
% p     = roll velocity           (rad/s)
% phi   = roll angle              (rad)
% delta = actual rudder angle     (rad)
% n     = actual shaft velocity   (rpm)
%
% The input vector is :
%
% ui      = [ delta_c n_c ]'  where
%
% delta_c = commanded rudder angle   (rad)
% n_c     = commanded shaft velocity (rpm)
%
% Reference:  Son og Nomoto (1982). On the Coupled Motion of Steering and
%             Rolling of a High Speed Container Ship, Naval Architect of
%             Ocean Engineering, 20: 73-83. From J.S.N.A. , Japan, Vol. 150,
%             1981.
%
% Author:    Trygve Lauvdal
% Date:      12th May 1994
% Revisions: 18th July 2001 (Thor I. Fossen): added output U, changed order
%                    of x-vector
%            20th July 2001 (Thor I. Fossen): changed my = 0.000238 to
%                    my = 0.007049

% Check of input and state dimensions

if (length(x) ~= 10),error('x-vector must have dimension 10 !');end
if (length(ui) ~= 2),error('u-vector must have dimension  2 !');end

% Normalization variables
L = 175;                      % length of ship (m)
U = sqrt(x(1)^2 + x(2)^2);    % service speed (m/s)

% Check service speed
if U <= 0,error('The ship must have speed greater than zero');end
if x(10) <= 0,error('The propeller rpm must be greater than zero');end

delta_max  = 35;              % max rudder angle (deg)
Ddelta_max = 2.5;             % max rudder rate (deg/s)
n_max      = 160;             % max shaft velocity (rpm)

% Non-dimensional states and inputs
```

```
delta_c = ui(1);
n_c     = ui(2)/60*L/U;

u     = x(1)/U;   v   = x(2)/U;
p     = x(7)*L/U; r   = x(3)*L/U;
phi   = x(8);     psi = x(6);
delta = x(9);     n   = x(10)/60*L/U;

% Parameters, hydrodynamic derivatives and main dimensions
m  = 0.00792;    mx     = 0.000238;   my = 0.007049;
Ix = 0.0000176;  alphay = 0.05;       lx = 0.0313;
ly = 0.0313;     Ix     = 0.0000176;  Iz = 0.000456;
Jx = 0.0000034;  Jz     = 0.000419;   xG = 0;

B     = 25.40;   dF = 8.00;    g     = 9.81;
dA    = 9.00;    d  = 8.50;    nabla = 21222;
KM    = 10.39;   KB = 4.6154;  AR    = 33.0376;
Delta = 1.8219;  D  = 6.533;   GM    = 0.3/L;
rho   = 1025;    t  = 0.175;   T     = 0.0005;

W     = rho*g*nabla/(rho*L^2*U^2/2);

Xuu      = -0.0004226;  Xvr    = -0.00311;    Xrr      = 0.00020;
Xphiphi  = -0.00020;    Xvv    = -0.00386;

Kv       =  0.0003026;  Kr     = -0.000063;   Kp       = -0.0000075;
Kphi     = -0.000021;   Kvvv   =  0.002843;   Krrr     = -0.0000462;
Kvvr     = -0.000588;   Kvrr   =  0.0010565;  Kvvphi   = -0.0012012;
Kvphiphi = -0.0000793;  Krrphi = -0.000243;   Krphiphi =  0.00003569;

Yv       = -0.0116;     Yr     =  0.00242;    Yp       =  0;
Yphi     = -0.000063;   Yvvv   = -0.109;      Yrrr     =  0.00177;
Yvvr     =  0.0214;     Yvrr   = -0.0405;     Yvvphi   =  0.04605;
Yvphiphi =  0.00304;    Yrrphi =  0.009325;   Yrphiphi = -0.001368;

Nv       = -0.0038545;  Nr     = -0.00222;    Np       =  0.000213;
Nphi     = -0.0001424;  Nvvv   =  0.001492;   Nrrr     = -0.00229;
Nvvr     = -0.0424;     Nvrr   =  0.00156;    Nvvphi   = -0.019058;
Nvphiphi = -0.0053766;  Nrrphi = -0.0038592;  Nrphiphi =  0.0024195;

kk     =  0.631;  epsilon =  0.921;  xR    = -0.5;
wp     =  0.184;  tau     =  1.09;   xp    = -0.526;
cpv    =  0.0;    cpr     =  0.0;    ga    =  0.088;
cRr    = -0.156;  cRrrr   = -0.275;  cRrrv =  1.96;
cRX    =  0.71;   aH      =  0.237;  zR    =  0.033;
xH     = -0.48;

% Masses and moments of inertia
m11 = (m+mx);
```

```
m22 = (m+my);
m32 = -my*ly;
m42 = my*alphay;
m33 = (Ix+Jx);
m44 = (Iz+Jz);

% Rudder saturation and dynamics
if abs(delta_c) >= delta_max*pi/180,
   delta_c = sign(delta_c)*delta_max*pi/180;
end

delta_dot = delta_c - delta;
if abs(delta_dot) >= Ddelta_max*pi/180,
   delta_dot = sign(delta_dot)*Ddelta_max*pi/180;
end

% Shaft velocity saturation and dynamics
n_c = n_c*U/L;
n   = n*U/L;
if abs(n_c) >= n_max/60,
   n_c = sign(n_c)*n_max/60;
end

if n > 0.3,Tm=5.65/n;else,Tm=18.83;end
n_dot = 1/Tm*(n_c-n)*60;

% Calculation of state derivatives
  vR     = ga*v + cRr*r + cRrrr*r^3 + cRrrv*r^2*v;
  uP     = cos(v)*((1 - wp) + tau*((v + xp*r)^2 + cpv*v + cpr*r));
   J     = uP*U/(n*D);
  KT     = 0.527 - 0.455*J;
  uR     = uP*epsilon*sqrt(1 + 8*kk*KT/(pi*J^2));
  alphaR = delta + atan(vR/uR);
  FN     = - ((6.13*Delta)/(Delta + 2.25))*(AR/L^2)*(uR^2 +
vR^2)*sin(alphaR);
  T      = 2*rho*D^4/(U^2*L^2*rho)*KT*n*abs(n);

% Forces and moments
 X    = Xuu*u^2 + (1-t)*T + Xvr*v*r + Xvv*v^2 + Xrr*r^2 + Xphiphi*phi^2 + ...
        cRX*FN*sin(delta) + (m + my)*v*r;

 Y    = Yv*v + Yr*r + Yp*p + Yphi*phi + Yvvv*v^3 + Yrrr*r^3 + Yvvr*v^2*r + ...
        Yvrr*v*r^2 + Yvvphi*v^2*phi + Yvphiphi*v*phi^2 + Yrrphi*r^2*phi + ...
        Yrphiphi*r*phi^2 + (1 + aH)*FN*cos(delta) - (m + mx)*u*r;

 K    = Kv*v + Kr*r + Kp*p + Kphi*phi + Kvvv*v^3 + Krrr*r^3 + Kvvr*v^2*r + ...
        Kvrr*v*r^2 + Kvvphi*v^2*phi + Kvphiphi*v*phi^2 + Krrphi*r^2*phi + ...
        Krphiphi*r*phi^2 - (1 + aH)*zR*FN*cos(delta) + mx*lx*u*r - W*GM*phi;
```

```
 N     = Nv*v + Nr*r + Np*p + Nphi*phi + Nvvv*v^3 + Nrrr*r^3 + Nvvr*v^2*r + ...
         Nvrr*v*r^2 + Nvvphi*v^2*phi + Nvphiphi*v*phi^2 + Nrrphi*r^2*phi + ...
         Nrphiphi*r*phi^2 + (xR + aH*xH)*FN*cos(delta);

% Dimensional state derivatives  xdot = [ u v r x y psi p phi delta n ]'
detM = m22*m33*m44-m32^2*m44-m42^2*m33;

xdot =[                          X*(U^2/L)/m11
          -((-m33*m44*Y+m32*m44*K+m42*m33*N)/detM)*(U^2/L)
           ((-m42*m33*Y+m32*m42*K+N*m22*m33-N*m32^2)/detM)*(U^2/L^2)
                    (cos(psi)*u-sin(psi)*cos(phi)*v)*U
                    (sin(psi)*u+cos(psi)*cos(phi)*v)*U
                                cos(phi)*r*(U/L)
           ((-m32*m44*Y+K*m22*m44-K*m42^2+m32*m42*N)/detM)*(U^2/L^2)
                                  p*(U/L)
                                delta_dot
                                  n_dot                  ];
```

B.3 Multi-purpose Offshore Supply Ship

For the computer simulations, the nonlinear model of an off-shore supply ship *Northern Clipper* which was presented in [81] is used. The length of *Northern Clipper* is L = 72.6 m and the mass is $m = 4.591 \cdot 10^6$ kg. The coordinate system is located in the center of gravity. The values for the inertia matrix and damping matrix are

$$M = \begin{bmatrix} 5.3122e6 & 0 & 0 \\ 0 & 8.2831e6 & 0 \\ 0 & 0 & 3.7454e9 \end{bmatrix}, \tag{B.1}$$

$$D = \begin{bmatrix} 5.0242e4 & 0 & 0 \\ 0 & 2.7229e5 & -4.3933e6 \\ 0 & -4.3933e6 & 4.1894e8 \end{bmatrix}. \tag{B.2}$$

The values for the bias time constants are chosen as

$$T = \begin{bmatrix} 1000 & 0 & 0 \\ 0 & 1000 & 0 \\ 0 & 0 & 1000 \end{bmatrix}. \tag{B.3}$$

The wave model parameters are also chosen as in [81] with ζ_i = 0.1 and ω_{oi} = 0.8976 rad/s corresponding to a wave period of 7.0 s in surge, sway and yaw.

Matlab M-File for Nonlinear Model of Multi-purpose Offshore Supply Ship

```
function xdot = nclipper(x,b,tau)
% Ship model for DP control simulation (supply ship Northern Clipper)
% x   = [ x y psi u v r]'
% tau = [tau1 tau2 tau3]'  % control vector of forces and moment
% b   = [b1 b2 b3]'        % bias term vector
%
% Reference:  T.I. Fossen (1994), "Guidance and Control of Ocean Vehicles",
%             John Wiley & Sons.
%
% Author:     Phung-Hung Nguyen
% Date:       12th Jul 2006

L    = 76.2;               % length of Northern Clipper (m)
mass = 4.591e6;            % mass of Northern Clipper   (kg)

% inertia matrix
M = [5.3122e6   0           0
     0          8.2831e6    0
     0          0           3.7454e9];

% damping matrix
D = [5.0242e4   0           0
     0          2.7229e5    -4.3933e6
     0          -4.3933e6   4.1894e8];

% Check of input and state dimensions
if (length(x)  ~= 6),error('x-vector must have dimension 6 !');end
if (length(tau) ~= 3),error('u-vector must have dimension 3 !');end

J = [cos(x(3)) -sin(x(3)) 0
     sin(x(3))  cos(x(3)) 0
     0          0         1];

nu      = [x(4) x(5) x(6)]';
eta_dot = J*nu;
nu_dot  = -inv(M)*D*nu + inv(M)*tau + inv(M)*inv(J)*b;

xdot    = [eta_dot' nu_dot']';
```