Study on Automatic Path Following and Collision Avoidance Algorithm for Intelligent Ships

崔, ボラ

https://doi.org/10.15017/2534443

出版情報：Kyushu University, 2019, 博士（工学）, 課程博士
バージョン：
権利関係：

Study on Automatic Path Following and Collision Avoidance Algorithm for Intelligent Ships

July 2019

Bora Choe

Table of Contents

Chapter 1. Introduction

1.1 Background of research

The concept of Artificial Intelligence (AI) was created by Alan Turing in 1950s [1], but the AI of that time did not yield research results as much as expected because of problems such as limitations in computer performance and limited application area. Hence, the study of the AI has gone through several dark ages, but with the advances in computer and the emergence of deep learning, the AI has got profitable outcomes and received attention again. Therefore, in recent years, movements have been appeared to convert existing human-operated systems into unmanned systems in many fields such as industry of vehicles and robots since the performance of AI technology has improved.

It has been found worldwide to follow the aforementioned tendency in the marine navigation area. The UK's Automated Ships Ltd and Norway's Kongsberg Maritime have signed a Memorandum of Understanding to build the world's first unmanned and fully autonomous ship for offshore operations [2]. Kongsberg Maritime has also gone into partnership with YARA to build the world's first autonomous and zero emission ship [3]. Rolls-Royce has joined forces with Google to develop intelligent awareness systems which are making vessels safer and are essential to making autonomous ships a reality [4]. According to report of Japan Ship Technology Research Association, some companies and research institutes in Japan has been cooperated each other to develop autonomous ships equipped with AI system that are capable of operating safely [5].

In particular, the International Maritime Organization (IMO) recognized that the IMO should take a proactive and leading role to encourage rapid technological development related to autonomous ships or unmanned ships. In 98th Maritime Safety Committee (MSC), the IMO introduced the concept of Maritime Autonomous Surface Ships (MASS) and started a regulatory scoping exercise (RSE) to determine whether current international regulations need to be amended for safe operation of MASS [6]. Furthermore, to facilitate the process of the regulatory scoping exercise, the degrees of autonomy for MASS are organised in 99th MSC as follows [7][8]:

- Ship with automated processes and decision support: Seafarers are on board to operate and control shipboard systems and functions. Some operations may be automated.
- Remotely controlled ship with seafarers on board: The ship is controlled and operated from another location, but seafarers are on board.
- Remotely controlled ship without seafarers on board: The ship is controlled and operated from another location. There are no seafarers on board.
- Fully autonomous ship: The operating system of the ship is able to make decisions and determine actions by itself.

Achieving the MASS can be recognized as coming up with and effective counterplan on accidents caused by human errors, which is the reason of most ocean incident, such as careless of ship operators. In general, the voyage of the ship is progressed by following a path which is considered safe while avoiding other ships or obstacles with the potential to cause a collision. Hence, it is necessary to equip with performances of automatic path following and collision avoidance in the fully autonomous ship.

In this study, the automatic path following algorithm using fuzzy inference and the automatic collision avoidance algorithm using reinforcement learning algorithm are developed in order to come up with the basic research for the fully MASS. The term MASS has recently been introduced by IMO, but research on automatic path following and collision avoidance for accomplishing autonomous ships has been in progress for decades. Path following is to follow given pre-planned route exactly by controlling a rudder. For the automatic path following, an autopilot system has been firstly developed by Sppery and Minorsky [9], [10]. The first autopilot on the ship was Proportional-Integral-Derivative (PID) type controller. Starting with the first autopilot, the PID method has been applied to the path following system by many scholars. Kallastom et al. [11] designed an autopilot on velocity scheduling and a self-tuning regulator with a Kalman filter for steady state course keeping. Jutong et al. [12] suggested a trimaran unmanned surface vehicle (TUSV) control system. The PID controller used in the system adjusted engine speed and yaw rate. The full system was tested successfully in manual operation and obtained useful data. However, the performance of the PID type autopilot is not good in various environmental conditions. The PID controller usually needs to adjust parameters corresponding to navigation condition. Additionally, the disadvantage of the PID is not suitable for nonlinear and complex systems. As for nonlinear problem, fuzzy logic is

used to enhance the PID controller due to its ability to translate the control action into rule base [13]. Nassim and Nabili [14] developed an integrated guidance and control system that enable robust tracking performance in spite of significant external disturbance using a self-tuning fuzzy controller. Cheng et al [15] suggested a fuzzy control design for waypoint tracking problem of ship autopilot. After the development of the autopilot, true automatic path following has been completed by Line-of-Sight (LOS) guidance system suggested by Fossen et al [16]. Through the LOS guidance system, the ship is able to automatically redirect according to the initial planned route near the waypoint. The system is configured so that when the ship enters a circle of acceptance of the target waypoint, the target waypoint is switched to the next waypoint. That time to change the target waypoint is important because it determines when to use rudder for modifying a heading angle of the ship. In the LOS guidance system, it additionally need to create LOS set points, which are two intersection points between a circle with certain radius around a ship and a straight track defined by two waypoints. The set points lead to extra mathematical task. As another disadvantage, the rudder is manipulated at same timing or point in order to enter a new course line because the radius of the acceptance circle is always constant. Thus, Velagic et al. [17] found out the position to execute the rudder through carrying out simulations several times. McGookin [18] attempted to adjust the size of the circle of acceptance according to the water depth near waypoints in order to find best timing for manipulating a rudder. However, their studies seem difficult to apply to all types of ships since a particular type of ship was subjected.

Automatic collision avoidance is to sense risk by the ship herself and to perform the avoidance action against objects expected to occur collisions. In order to inform the ship of the possibility of collision, collision risk assessment has been studied. The collision risk assessment can be classified into two areas: ship domain and collision risk. The ship domain is the minimum virtual safety zone around a ship. If any vessel or obstacles enters into the established safety zone, the ship should initiate the action to evade collision. The ship domain was introduced as ellipse shape by Fujii and Tanaka [19]. Since then, the appearance of ship domain has evolved into various shapes, taking into consideration various factors [20][21][22]. Goodwin [23] proposed the ship domain with three fan shaped combinations for open sea. The front right area of ship is biggest among them because it is a priority to change direction toward starboard side in the International Regulations for Preventing Collisions at Sea 1972 (COLREGs). Pietrzykowski and Uriasz [24] decided the outline of ship domain based on a survey of participants with a lot of voyage experience. In general, the collision risk has been evaluated

by fuzzy logic method. Iijima et al [25], and Kijima and Furukawa [26] used Time to Closet Point of Approach (TCPA) and Distance at Closet Point of Approach (DCPA). Kao et al. [27] defined three fuzzy variables that are ship length, speed and sea condition and the combination of the variables has total 27 fuzzy rules. The study on the collision assessment has been later utilized to determine when to start collision avoidance action. In deducing the starting point of avoidance actions, the ship domain showed good performance in situation to avoid single ship but it was difficult to apply in real coastal situation. The collision risk assessment using fuzzy theory is effective to estimate the ship of maximum collision risk in a lot of traffic [28]. After detecting a potential collision, a ship should take action to avoid a clash. Namely, the ship must move toward a new direction or path, not an existing track, while there is a risk of collision. In order to make new path, Cummings et al [29] and Blaich et al [30] used A* algorithm that is a way to find the shortest path from a given start points to an end point. In order to implement the system using A* algorithm, a rectangular grid should be created and it requires the exact information on the position and the size of obstacles, the destinations of the controlling ship and another ships. However, the related data is generally hard to know precisely. So far, the ship has been offered the optimal action to avoid the collision obtained from the outcomes of some algorithms. Since the ship cannot determine the action by herself, it is difficult to see as fully autonomous ship. If these researches on automatic path following and collision avoidance are sufficiently developed, it is expected that an intelligent ship which is fully autonomous ship proposed by IMO will be implement in the near future.

1.2 Objectives of research

In order to achieve an intelligent ship that can determine and operate action to be taken by herself, automatic path following algorithm and collision avoidance algorithm which are import functions for completing the voyage are developed and the objectives of this research are as follows.

First, automatic path following algorithm is proposed to allow a ship to follow a planned desired path accurately. The study on path following algorithm has been fulfilled by incorporating a rudder control system as an autopilot in the LOS guidance system. However, the LOS guidance algorithm has two disadvantages; namely the equal timing to use rudder and the necessity of LOS set points. To overcome the problems, the algorithm is composed with

two components using fuzzy inference that is suitable for nonlinear problem: one is a waypoint guidance system and the other is a rudder control system. In the LOS guidance algorithm, a target waypoint is changed when a ship enter a specific circular area with a same constant radius. It is expected to generate overshoots, cross track error and heading error, since the rudder is always used at same timing so as to penetrate into a new planned straight line. The waypoint guidance system in the proposed path following algorithm will be developed to be able to find an optimal timing to modify a target waypoint for reducing overshoots when the ship change her course. On the other hand, the LOS set points lead to the additional computational work. The major reason to designate the LOS set points is to reduce cross track error. If the cross track error is taken into account as input parameter in the rudder control system, it is not necessary to create the LOS set points. Therefore, the rudder control system will consider heading error, cross track error, ship's speed and yaw rate in order to gain the appropriate rudder angle which enables the ship to trace the desired route.

Second, most of existing researchers have carried out simulations in the virtual situations that ships move on a geometrical track consisted of waypoints under arbitrary external disturbances such as wind, current and wave. As for actual disturbances, they have irregular characteristics and let ships swerve from a desired track. Simulations performed under the assumed virtual conditions are anticipated to have limitation to assess the effectiveness of the developed algorithm. Hence, numerical simulations are performed to verify the capability in realistic environmental situation reproduced by data obtained from official institute.

Third, automatic collision avoidance algorithm which utilizes reinforcement learning in order to make decisions on action to avoid collision is proposed. The training structure of reinforcement learning is similar to human learning method through trial and error. Thus, the reinforcement learning allows the ship to identify various patterns of surrounding environments such as the variation of encounter situations and behaviours to be taken by herself unlike previous studies. As for detecting potential collision risks, the degrees of the risks will be evaluated by using fuzzy inference because the degrees of the risks should be found in real time against the multiple ships.

Fourth, the reinforcement learning has just begun to be used in the marine filed. In existing studies [31][32], one of methods of reinforcement learning was applied problem of ship manoeuvring and results were shown. In this study, two kind of learning method such as Deep Q-network (DQN) and Deep Deterministic Policy Gradient (DDPG) are introduced and

employed. Through the comparison of the outcomes obtained by applying these two methods, it will be found which method is more suitable for the collision avoidance problem of ships.

1.3 Outline

This thesis consists of six chapters.

In Chapter 1 entitled “Introduction”, the background and purposes of this research are introduced.

In Chapter 2 entitled “Mathematical model for ship motion”, the mathematical model of manoeuvring motion and external disturbance model are described in order to carry out numerical simulations which is necessary to evaluate the performance of developed algorithm.

In Chapter 3 entitled “Development of automatic path following algorithm”, automatic path following algorithm is suggested. Two components of the algorithm developed by using fuzzy inference, the waypoint guidance system and the rudder control system, are explained through showing fuzzy rules and fuzzy membership functions. A ship equipped with the proposed algorithm is simulated in virtual situations and the results are analyzed.

In Chapter 4 entitled “Application of developed automatic path following algorithm in realistic situations”, the path following algorithm developed in Chapter 3 is additionally verified in realistic situations. In order to reproduce realistic ship voyage, an actual route planned by ship operators has been used and actual environmental data such as wind and current measured provided by official institutes are applied as wind and current vectors depending on the present ship’s location in real time.

In Chapter 5 entitled “Development of automatic collision avoidance algorithm”, automatic collision avoidance algorithm is developed. The evaluation of collision risk and the theory of reinforcement learning which are needed to be established to develop the algorithm are described. Collision avoidance problem is defined as MDP (Markov Decision Process) model in the reinforcement learning and the action to evade collision is evaluated on the basis of COLREGs. Among the deep reinforcement learning methods, DQN and DDPG are applied and simulations results using the both methods are compared and examined.

In Chapter 6 entitled “Conclusions”, the conclusions of this study are summarized and the possible future research is discussed.

Chapter 2. Mathematical Model for Ship Motion

2.1. Introduction

The suggested track keeping algorithm and collision avoidance algorithm in this study employed numerical simulations in order to verify their effectiveness. For prediction of a ship motion in the simulations, many researches dealing with mathematical model have been introduced to represent the dynamic characteristics of manoeuvring motion such as MMG model, Nomoto model, Abkowitz model, and others [33][34][35][36]. In this research, Manoeuvring Mathematical Modelling Group (MMG) model proposed by Japanese research group [37] was applied to express hydrodynamic forces acting on a ship. The MMG model consists of the individual properties of hull, propeller, rudder, and other external components. This chapter demonstrates the mathematical model of hydrodynamic forces and moment for each part of the MMG model devised by Kijima et al. [38].

2.2. Equations of manoeuvring motion

Three degrees of freedom motion such as surge, sway, and yaw is universally applicable to express ship dynamic motion in the horizontal plan. The ship motion can be described using coordinate systems as shown in Fig. 2.1. $o - x_0 y_0$ is an earth-fixed coordinate system and $G - xy$ is a body-fixed coordinate system with the center of gravity of a ship G as the origin. In the body-fixed coordinate system, the x-axis is positive in the direction of the fore perpendicular. The y-axis is positive in the direction of the starboard side and the z-axis is positive downwards. Let the forces in the x and y axes be X and Y, and the moment around vertical axis which passing through G is indicated by N. u and v denote the components of ship speed U in x and y directions respectively. α and ν mean the incident angles of current and wind. V_C and V_W are defined as the speeds of current and wind respectively. The positive direction of a heading angle ψ which is the angle between x_0 and x axes is assumed to be clockwise direction. The drift angle β is shown in Fig. 2.1 with its positive direction.

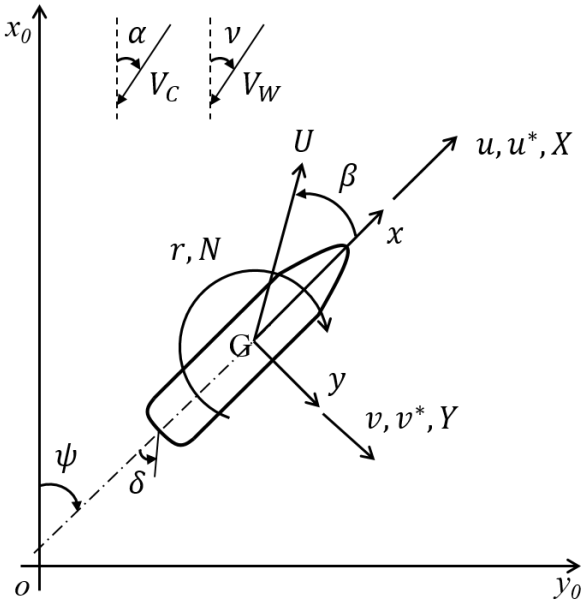


Fig. 2.1 Coordinate systems

The equations of ship motion in the earth fixed coordinate system $o - x_0 y_0$ in Fig. 2.1 are expressed as follows:

$$\begin{aligned} m\ddot{x}_0 &= X_0, \\ m\ddot{y}_0 &= Y_0, \\ I_{ZZ}\ddot{\psi} &= N. \end{aligned} \tag{2.1}$$

where,

X_0, Y_0, N : the components of external forces in x_0 and y_0 axes and moment around z-axis respectively,

m : mass of ship,

I_{zz} : moment of inertia around z-axis,

$\ddot{\psi}$: yaw acceleration,

$\ddot{x}_0, \ddot{y}_0$: components of acceleration in x_0 and y_0 axes separately.

However, hydrodynamic forces acting on the ship in the body-fixed coordinate system should be described because the application of the body-fixed coordinate system is more effective

than the earth-fixed coordinate system. There is the following relation between X, Y and X_0, Y_0 which are x_0 and y_0 components of the hydrodynamic forces in the earth fixed coordinate system.

$$\begin{aligned} X &= X_0 \cos\psi + Y_0 \sin\psi\,, \\ Y &= Y_0 \cos\psi - X_0 \sin\psi\,. \end{aligned} \tag{2.2}$$

Substituting Eq. (2.1) for Eq. (2.2), the following equations are obtained:

$$\begin{aligned} X &= m(\ddot{x}_0 \cos\psi + \ddot{y}_0 \sin\psi), \\ Y &= m(\ddot{y}_0 \cos\psi - \ddot{x}_0 \sin\psi). \end{aligned} \tag{2.3}$$

Furthermore, $\dot{x}_0, \dot{y}_0$ which are x_0 and y_0 components of the ship speed at G in the earth-fixed coordinate system is expressed by u and v which are x and y components of the speed in the body-fixed coordinate system.

$$\begin{aligned} \dot{x}_0 &= u\cos\psi\ - v\sin\psi, \\ \dot{y}_0 &= v\cos\psi + u\sin\psi\,. \end{aligned} \tag{2.4}$$

Differentiating Eq. (2.4) by time, x_0 and y_0 components of the acceleration of G is given by:

$$\begin{aligned} \ddot{x}_0 &= \dot{u}\cos\psi - u\dot{\psi}\sin\psi - \dot{v}\sin\psi + v\dot{\psi}\cos\psi, \\ \ddot{y}_0 &= \dot{v}\cos\psi - v\dot{\psi}\sin\psi + \dot{u}\sin\psi + u\dot{\psi}\cos\psi\,. \end{aligned} \tag{2.5}$$

Substituting Eq. (2.5) for Eq. (2.3), the following equations of motion are derived as follows:

$$\begin{aligned} m(\dot{u} - v\dot{\psi}) &= X, \\ m(\dot{v} + u\dot{\psi}) &= Y, \\ I_{ZZ}\ddot{\psi} &= N. \end{aligned} \tag{2.6}$$

Since the ship moves in viscous fluid, the influence of added mass and added moment of inertia should be considered as shown in Eq. (2.7).

$$\begin{aligned}(m+m_x)\dot{u}-(m+m_y)vr&=X,\\(m+m_y)\dot{v}+(m+m_x)ur&=Y,\\(I_{ZZ}+i_{ZZ})\dot{r}&=N.\end{aligned}\tag{2.7}$$

where,

m, m_x, m_y : the mass and *x*- and *y*- axis components of added mass of ship,

I_{zz}, i_{zz} : the moment and added moment of inertia of a ship,

u , v : *x*- and *y*- axis speed components ,

r : yaw rate.

u and v can be expressed using ship speed U and drift angle β as the following equation:

$$\begin{aligned}u&=U\cos\beta\,,\\v&=-U\sin\beta\,.\end{aligned}\tag{2.8}$$

Therefore, Eq. (2.7) is rewritten as:

$$\begin{aligned}(m+m_x)(\dot{U}\cos\beta-U\dot{\beta}\sin\beta)+(m+m_y)Ur\sin\beta&=X,\\-(m+m_y)(\dot{U}\sin\beta+U\dot{\beta}\cos\beta)+(m+m_x)Ur\cos\beta&=Y,\\(I_{ZZ}+i_{ZZ})\dot{r}&=N.\end{aligned}\tag{2.9}$$

The non-dimensional equations of motion is derived by non-dimensionalizing Eq. (2.9) as follows.

$$\begin{aligned}(m'+m'_x)\left(\frac{L}{U}\right)\left(\frac{\dot{U}}{U}\cos\beta-\dot{\beta}\sin\beta\right)+(m'+m'_y)r'\sin\beta&=X',\\-(m'+m'_y)\left(\frac{L}{U}\right)\left(\frac{\dot{U}}{U}\sin\beta+\dot{\beta}\cos\beta\right)+(m'+m'_x)r'\cos\beta&=Y',\\(I'_{ZZ}+i'_{ZZ})\left(\frac{L}{U}\right)^2\left(\frac{\dot{U}}{L}r'+\frac{U}{L}\dot{r'}\right)&=N'.\end{aligned}\tag{2.10}$$

Here, superscript " ′ " signifies non-dimensional parameter and non-dimensionalization of the equations was proceeded by Eq. (2.11).

$$
\begin{aligned}
m', m'_x, m'_y &= m, m_x, m_y / \frac{1}{2}\rho L^2 d, \\
I'_{zz}, i'_{zz} &= I_{zz}, i_{zz} / \frac{1}{2}\rho L^4 d, \\
X', Y' &= X, Y / \frac{1}{2}\rho L d U^2, \\
N' &= N / \frac{1}{2}\rho L^2 d U^2 , \\
r' &= rL/U.
\end{aligned}
\tag{2.11}
$$

where,

ρ	:	density of fluid,
L	:	ship length,
d	:	draft.

2.3. Mathematical model of external forces

The right-hand side terms of Eq. (2.10) with respect to the non-dimensional external forces X' , Y' and yaw moment N' can be expressed as Eq. (2.12) using MMG mathematical model.

$$
\begin{aligned}
X' &= X'_H + X'_P + X'_R + X'_W, \\
Y' &= Y'_H + \quad\; Y'_R + Y'_W, \\
N' &= N'_H + \quad\; N'_R + N'_W.
\end{aligned}
\tag{2.12}
$$

Here, subscript "*H*", "*P*", "*R*" and "*W*" intend hull, propeller, rudder, and wind. The lateral force and moment by a propeller were omitted because they have a very small effect. The hydrodynamic forces due to external disturbance such as current and wind will be described in the next section in detail.

2.3.1 Hydrodynamic forces and moment acting on a hull

The non-dimensional hydrodynamic forces X'_H , Y'_H and moment N'_H acting on a hull in Eq. (2.12) are calculated using mathematical model proposed by Kijima et al. [38]. $X'_{\beta r}$ presents the variation of longitudinal force due to drift angle and yaw rate, and X'_{uu} indicates ship's resistance in forward straight motion.

$$
\begin{aligned}
X'_H &= X'_{\beta r} r' \sin\beta + X'_{uu} \cos^2\beta\,, \\
Y'_H &= Y'_{\beta}\beta + Y'_{r} r' + Y'_{\beta\beta}\beta|\beta| + Y'_{rr} r'|r'| + (Y'_{\beta\beta r}\beta + Y'_{\beta rr} r')\beta r', \\
N'_H &= N'_{\beta}\beta + N'_{r} r' + N'_{\beta\beta}\beta|\beta| + N'_{rr} r'|r'| + \left(N'_{\beta\beta r}\beta + N'_{\beta rr} r'\right)\beta r'.
\end{aligned}
\tag{2.13}
$$

2.3.2 Longitudinal force produced by a propeller

Since the lateral component of a propeller force and moment generated by a propeller are negligible small comparing with its longitudinal component, they are ignored in general. Hence, only the longitudinal force X_P generated by a propeller are taken into account in the MMG model. The non-dimensional force X'_P is determined by the following equations:

$$
\begin{aligned}
X'_P &= (1 - t_P) n^2 D^4{}_P K_T(J_P) / \frac{1}{2} L d U^2, \\
K_T(J_P) &= C_1 + C_2 J_P + C_3 J_P^2, \\
J_P &= U\cos\beta\,(1 - w_P)/nD_P.
\end{aligned}
\tag{2.14}
$$

Where, n means the number of propeller revolution and D_P is the diameter of a propeller. t_P represents thrust deduction rate. Thrust coefficient $K_T(J_P)$ can be calculated with the function of advance coefficient J_P. C_1, C_2, and C_3 are constants and w_P is effective wake fraction.

2.3.3 Hydrodynamic forces and moment produced by a rudder

When a rudder exists in wake, additional lateral force is induced and rudder resistance is decreased due to interaction between a rudder and a hull. Therefore, the non-dimensional hydrodynamic forces X'_R, Y'_R, and moment N'_R generated by a rudder are represented by considering the influence of hydrodynamic interaction as Eq. (2.15).

$$
\begin{aligned}
X'_R &= -(1 - t_R) F'_N \sin\delta\,, \\
Y'_R &= -(1 + a_H) F'_N \cos\delta\,, \\
N'_R &= -(x'_R + a_H x'_H) F'_N \cos\delta\,.
\end{aligned}
\tag{2.15}
$$

Where, δ is rudder angle and F'_N is non-dimensional rudder normal force. t_R and a_H indicate interaction coefficient. x'_R and x'_H define the position of a rudder and the acting point of additional lateral force induced by steering.

F_N' can be expressed as Eq. (2.16) and Eq. (2.17) describes C_N, $U_R'^2$, α_R separately.

$$F_N' = \left(\frac{A_R}{Ld}\right) C_N U_R'^2 \sin\alpha_R \tag{2.16}$$

$$\begin{aligned}
C_N &= 6.13K_R/(K_R + 2.25) \\
U_R'^2 &= v(1-w_R)^2\{1 + C \cdot g(s)\} \\
g(s) &= \eta K\{2-(2-K)s\}s/(1-s)^2 \\
\eta &= D_P/h_R \\
K &= 0.6(1-w_P)/(1-w_R) \\
s &= 1.0 - (1-w_P)U\cos\beta\,/nP \\
w_P &= w_{P0} \cdot \exp(-4.0\beta'^2_P) \\
\beta'_P &= \beta - x'_P \cdot r' \\
w_R &= w_{R0} \cdot w_{P0}/w_{P0} \\
w_{P0} &= 0.5C_B - 0.05 \\
\alpha_R &= \delta - \gamma \cdot \beta'_R \\
\beta'_R &= \beta - 2x'_R \cdot r' \\
x'_P &\cong -0.5 \\
x'_R &\cong -0.5
\end{aligned} \tag{2.17}$$

where,

C_N : gradient coefficient of rudder normal force,

U_R : effective inflow velocity at rudder location,

A_R : rudder area,

α_R : effective inflow angle at rudder location,

C : coefficient describing a difference between starboard and port steering,

C_B : block coefficient

K_R : aspect ratio of a rudder,

h_R : height of a rudder,

P : propeller pitch,

γ : flow straightening factor,

$1 - w_R$: effective wake fraction at rudder location,

$1 - w_{R0}$: effective wake fraction at rudder location in straight forward running,

s : slip ratio.

t_R is resistance increase ratio due to the rudder-on-hull interaction and written as follows [39]:

$$1 - t_R = 0.28C_B + 0.55 \tag{2.18}$$

2.4. Mathematical models for environmental disturbances

During the voyage, the motion of a ship is influenced by the existence of environmental disturbances such as wind and current. Therefore, the influence of wind and current as environmental disturbances were considered for realistic ship motion. The notations related to the forces and moment due to the disturbances are appeared in Fig. 2.1

2.4.1 Wind effects

Forces and moment caused by wind are expressed using relative wind speed U_W^* and wind direction v^* as below:

$$\begin{aligned} X_W &= \frac{1}{2}\rho_A A_T {U_W^*}^2 C_X(v^*), \\ Y_W &= \frac{1}{2}\rho_A A_L {U_W^*}^2 C_Y(v^*), \\ N_W &= \frac{1}{2}\rho_A A_L L_{OA} {U_W^*}^2 C_N(v^*). \end{aligned} \tag{2.19}$$

U_W^* and v^* are relative wind velocity and angles respectively. They are expressed as follows:

$$\begin{aligned} {U_W^*}^2 &= \sqrt{{u_W^*}^2 + {v_W^*}^2}, \\ u_W^* &= u + U_W \cos(\psi - v), \\ v_W^* &= v - U_W \cos(\psi - v), \\ v^* &= \tan^{-1}\left(\frac{u_W^*}{v_W^*}\right). \end{aligned} \tag{2.20}$$

Eq. (2.19) is non-dimensionalized as shown in Eq. (2.21),

$$
\begin{aligned}
X'_W &= X_W / \frac{1}{2}\rho L d U^2 &&= \frac{\rho_a}{\rho} \cdot \frac{A_T}{Ld} \frac{{U_W^*}^2}{U^2} \cdot C_X(v^*), \\
Y'_W &= Y_W / \frac{1}{2}\rho L d U^2 &&= -\frac{\rho_a}{\rho} \cdot \frac{A_L}{Ld} \frac{{U_W^*}^2}{U^2} \cdot C_Y(v^*), \\
N'_W &= N_W / \frac{1}{2}\rho L^2 d U^2 &&= -\frac{\rho_a}{\rho} \cdot \frac{L_{OA} A_L}{L^2 d} \frac{{U_W^*}^2}{U^2} \cdot C_N(v^*).
\end{aligned}
\tag{2.21}
$$

Where, ρ_a means the density of the air. A_T and A_L represent transverse and lateral projected areas respectively. L_{OA} is the length over all of the ship. C_X, C_Y and C_N are coefficients of wind forces and moment, and they are estimated by prediction formulae presented by Fujiwara [40]. The formulae were derived based on a wide range of experimental data including LNG carrier, PCC and so on. Coefficients used in the estimation formulae for calculating wind effects can be defined as Eq. (2.22).

$$
\begin{aligned}
C_X &= X_0 + X_1 \cos v^* + X_3 \cos 3v^* + X_5 \cos 5v^*, \\
C_Y &= Y_1 \sin v^* + Y_3 \sin 3v^* + X_5 \sin 5v^*, \\
C_N &= N_1 \sin v^* + N_2 \sin 2v^* + N_3 \sin 3v^*.
\end{aligned}
\tag{2.22}
$$

Where, coefficients associated with Eq. (2.22) are represented as follows.

$$
\left.
\begin{aligned}
X_0 &= x_{00} + x_{01}\frac{BH_{BR}}{A_T} + x_{02}\frac{C}{H_C} + x_{03}\frac{A_{OD}}{{L_{OA}}^2} \\
X_1 &= x_{10} + x_{11}\frac{A_L}{L_{OA}B} + x_{12}\frac{LH_L}{A_L} + x_{13}\frac{L_{OA}H_{BR}}{A_L} + x_{14}\frac{A_{OD}}{A_L} \\
&\quad + x_{11}\frac{A_T}{L_{OA}B} + x_{16}\left(\frac{A_T}{{L_{OA}}^2}\right)^{-1} + x_{17}\left(\frac{H_C}{L_{OA}}\right)^{-1} \\
X_3 &= x_{30} + x_{31}\left(\frac{L_{OA}H_{BR}}{A_L}\right)^{-1} + x_{32}\frac{A_L}{A_T} + x_{33}\frac{L_{OA}H_C}{A_L} + x_{34}\frac{A_{OD}}{A_L} \\
&\quad + x_{35}\frac{A_{OD}}{{L_{OA}}^2} + x_{36}\frac{C}{H_C} + x_{37}\frac{C_{BR}}{L_{OA}} \\
X_5 &= x_{50} + x_{51}\left(\frac{A_{OD}}{A_L}\right)^{-1} + x_{52}\frac{C_{BR}}{L_{OA}} + x_{53}\frac{A_L}{L_{OA}B}
\end{aligned}
\right\}
\tag{2.23}
$$

$$
\left.
\begin{aligned}
Y_1 &= y_{10} + y_{11}\frac{C_{BR}}{L_{OA}} + y_{12}\frac{C}{L_{OA}} + y_{13}\left(\frac{A_{OD}}{A_L}\right)^{-1} + y_{14}\frac{C}{H_C} \\
&\quad + y_{15}\left(\frac{BH_{BR}}{A_T}\right)^{-1} \\
Y_3 &= y_{30} + x_{31}\frac{A_L}{L_{OA}B} + y_{32}\frac{L_{OA}H_L}{A_L} + y_{33}\frac{C_{BR}}{L_{OA}} + y_{34}\left(\frac{H_{BR}}{B}\right)^{-1} \\
&\quad + y_{35}\frac{A_{OD}}{A_L} + y_{36}\left(\frac{BH_{BR}}{A_T}\right)^{-1} \\
Y_5 &= y_{50} + y_{51}\frac{A_L}{L_{OA}B} + y_{52}\left(\frac{H_{BR}}{L_{OA}}\right)^{-1} + y_{53}\frac{C_{BR}}{L_{OA}} + y_{54}\left(\frac{H_{BR}}{B^2}\right)^{-1} \\
&\quad + y_{55}\frac{C}{L_{OA}} + y_{36}\frac{L_{OA}H_C}{A_L}
\end{aligned}
\right\} \qquad (2.24)
$$

$$
\left.
\begin{aligned}
N_1 &= n_{10} + n_{11}\frac{C}{L_{OA}} + n_{12}\frac{L_{OA}H_C}{A_L} + n_{13}\left(\frac{A_L}{A_T}\right)^{-1} + n_{14}\frac{C}{H_C} \\
&\quad + n_{16}\frac{A_T}{{L_{OA}}^2} + n_{17}\left(\frac{A_T}{B^2}\right)^{-1} + n_{15}\frac{C_{BR}}{L_{OA}} \\
N_2 &= n_{20} + n_{21}\frac{C_{BR}}{L_{OA}} + n_{22}\frac{C}{L_{OA}} + n_{23}\left(\frac{A_{OD}}{A_L}\right)^{-1} + n_{24}\frac{A_T}{B^2} \\
&\quad + n_{25}\left(\frac{H_{BR}}{L_{OA}}\right)^{-1} + n_{26}\left(\frac{BH_{BR}}{A_T}\right)^{-1} + n_{27}\frac{A_L}{L_{OA}B} \\
&\quad + n_{28}\frac{A_L}{{L_{OA}}^2} \\
N_3 &= n_{30} + n_{31}\frac{C_{BR}}{L_{OA}} + n_{32}\left(\frac{BH_{BR}}{A_T}\right)^{-1} + n_{33}\frac{A_L}{A_T}
\end{aligned}
\right\} \qquad (2.25)
$$

where,

L_{OA} : length over all (m)

B : breadth (m)

A_T : transverse projected area (m^2)

A_L : lateral projected area (m^2)

A_{ss} : lateral projected area of superstructure (m^2)

A_{OD} : total area of A_{ss} and lateral projected area of LNG tanks and containers etc. on the deck (m^2)

C : distance from midship section to the center of A_L (m)

C_{BR} : distance from midship section to the center of the A_{ss} (m)

H_{BR} : height to the top of superstructure (bridge) (m)

H_C : height to the center of lateral projected area (m)

The parameters for calculation of wind force are express in Fig. 2.2. Fig. 2.2 indicates the frontal shape of the hull when viewed from the bow direction and the shape of the lateral projection seen from the starboard direction, respectively. The values of coefficients in Eqs. (2.23), (2.24), and (2.25) are shown in Table 2.1.

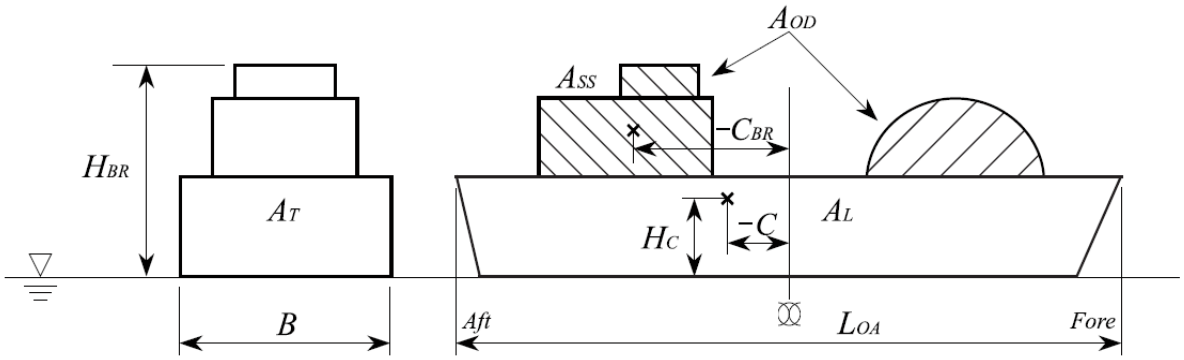


Fig. 2.2 Definitions of superstructures of a ship

Table 2.1 Manoeuvring coefficients for whole ship model [40]

$m=$	0	1	2	3	4	5	6	7	8
x_{0m}	-0.330	0.293	0.0193	0.682					
x_{1m}	-1.353	1.700	2.87	-0.463	-0.570	-6.640	-0.0123	0.0202	
x_{3m}	0.830	-0.413	-0.0827	-0.563	0.804	-5.67	0.0401	-0.132	
x_{5m}	0.0372	-0.0075	-0.103	0.0921					
y_{1m}	0.684	0.717	-3.22	0.0281	0.0661	0.298			
y_{3m}	-0.400	0.282	0.307	0.0519	0.0526	-0.0814	0.0582		
y_{5m}	0.122	-0.166	-0.0054	-0.0481	-0.0136	0.0864	-0.0297		
n_{5m}	0.299	1.71	0.183	-1.09	-0.0442	-0.289	4.24	-0.0646	0.0306
n_{2m}	0.117	0.123	-0.323	0.0041	-0.166	-0.0109	0.174	0.214	-1.06
n_{3m}	0.0230	0.0385	-0.0339	0.0023					

2.4.2 Current effects

In case that a ship travels under current in numerical simulations, hydrodynamic forces acting on the ship in flow should be contained the influence of current because the inflow velocities on the hull is altered due to current. To calculate the forces including the current effects, relative speed is applied and expressed in Eq. (2.26). Where, $u^*$and v^* are donated as the x- and y-components of relative speed U^* as shown in Fig. (2.1).

$$\begin{aligned} u^* &= u + V_c \cos(\psi - \alpha)\,, \\ v^* &= v - V_c \sin(\psi - \alpha)\,. \end{aligned} \tag{2.26}$$

Therefore, Eq. (2.9) is replaced as:

$$\begin{aligned} (m + m_x)\dot{u} - \big(m + m_y\big)vr - \big(m_x - m_y\big)V_C r \sin(\psi - \alpha) &= X, \\ \big(m + m_y\big)\dot{v} + (m + m_x)vr - (m_y - m_x)V_C r \cos(\psi - \alpha) &= Y, \\ (I_{ZZ} + i_{ZZ})\dot{r} &= N. \end{aligned} \tag{2.27}$$

The non-dimensionalized equations of Eq. (2.27) are demonstrated as Eq. (2.28).

$$\begin{aligned} (m' + m'_x)\left(\frac{L}{U}\right)\left(\frac{\dot{U}}{U}\cos\beta - \dot{\beta}\sin\beta\right) + (m' + m'_y)r'\sin\beta & \\ -(m'_x - m'_y)\left(\frac{V_C}{U}\right)r'\sin(\psi - \alpha) &= X', \\ -\big(m' + m'_y\big)\left(\frac{L}{U}\right)\left(\frac{\dot{U}}{U}\sin\beta + \dot{\beta}\cos\beta\right) + (m' + m'_x)r'\cos\beta & \\ -(m'_y - m'_x)\left(\frac{V_C}{U}\right)r'\cos(\psi - \alpha) &= Y', \\ (I'_{ZZ} + i'_{ZZ})\left(\frac{L}{U}\right)^2\left(\frac{\dot{U}}{L}r' + \frac{U}{L}\dot{r'}\right) &= N'. \end{aligned} \tag{2.28}$$

2.5. Subjective ship

A Tanker named KVLCC2 was adopted as a subject ship for numerical simulations to verify the effectiveness of the systems suggested in this thesis. The numerical simulations carried out in two situations under virtual and realistic environments. A 1:128 scale model ship was applied in virtual situation and an actual-sized ship was used in realistic situation, respectively. The principal particulars of the model ship and the real ship are shown in Table 2.2.

Table 2.2 Principal particulars of the subjective ship (KVLCC2)

	Real ship		Model ship	
Ship type	Tanker		Tanker	
Length between perpendicular	320.00	[m]	2.5000	[m]
Breadth	58.000	[m]	0.4531	[m]
Depth	30.000	[m]	0.2344	[m]
Draft	20.800	[m]	0.1625	[m]
Block coefficient	0.8098		0.8098	
Diameter of propeller	9.8600	[m]	0.0770	[m]
Rudder area	273.30	[m^2]	0.0167	[m^2]

The subject ship was influenced by environmental disturbances such as wind and current in the numerical simulations in order to depict ship motion during a voyage. For computation of wind effects, the dimensions of superstructures of the subject ship were assumed based on the conventional design of takers. Table 2.3 signifies the used superstructures dimensions of the subject ship.

Table 2.3 Superstructures dimensions of the subject ship (KVLCC2)

	Real ship		Model ship	
A_T	1234.064	[m^2]	0.0753	[m^2]
A_L	2526.587	[m^2]	0.1542	[m^2]
A_{OD}	698.0166	[m^2]	0.0426	[m^2]
C	6.0285	[m]	0.0471	[m]
C_{BR}	-105.3735	[m]	-0.8232	[m]
H_{BR}	33.2308	[m]	0.2596	[m]
H_C	3.8892	[m]	0.0304	[m]

Through the captive model test of KVLCC2, the required hydrodynamic derivatives for the simulations were obtained and the values are presented in Table 2.4.

Table 2.4 Hydrodynamic derivative values of the subject ship (KVLCC2)

Item	Value	Item	Value
X'_{uu}	-0.0250000	$X'_{\beta r}$	0.0897830
Y'_{β}	0.2387000	N'_{β}	0.1377000
Y'_{r}	0.0892543	N'_{r}	-0.0478000
$Y'_{\beta\beta}$	0.5928000	$N'_{\beta\beta}$	-0.0533000
Y'_{rr}	0.0661000	N'_{rr}	0.0214000
$Y'_{\beta\beta r}$	0.2844000	$N'_{\beta\beta r}$	-0.0228000
$Y'_{\beta rr}$	0.5367000	$N'_{\beta rr}$	-0.2660000

2.6. Conclusions

In this chapter, the mathematical model of ship motion was introduced. The main conclusions of this chapter are drawn as follows:

- Two kinds of coordinate systems are used to demonstrate the ship motion. The relationship between the earth-fixed coordinate system and the body-fixed coordinate system was introduced from the equations of manoeuvring motion.

- KVLCC2 was adopted as a subject ship and mathematical model based on the MMG model was selected in order to predict ship dynamic motion in numerical simulations. The forces and moment acting on a hull, the forces generated by a propeller and the forces and moments due to a rudder as parts of the MMG model were described respectively.

- Since the effects of wind and current were applied to numerical simulations, the relevant mathematical models were explained.

For verification of automatic track keeping algorithm presented in Chapter 3, the MMG model including wind effects and current effects was applied. On the other hand, simulations of automatic collision avoidance based on the developed algorithm in Chapter 5 were carried out without disturbances to focus on inspecting only the effectiveness of the algorithm.

Chapter 3. Development of Automatic Path Following Algorithm

3.1 Introduction

Research on autonomous navigation has received considerable attention because it has potential to prevent marine accidents due to human error which is caused when ships are manually operated. Before ships depart a port, ship operators always make a route plan with waypoints positions in consideration of weather condition data, geometric factors at sea and so on. Ships should follow a desired track comprised of several waypoints set in the route plan in order to ensure safe navigation. Automatic path following which is one of the functions of autonomous navigation enables ships to trace the planned path accurately. The path following problem is basically how to make ships follow a set of given waypoints by controlling a rudder [41][42][43]. To solve the problem, a guidance system which leads a ship to prescribed waypoints is required as well as a control system that steers a ship according to orders from the guidance system. Some of guidance laws commonly used in marine community have been directly influenced by outcomes in missile community, such as the Line-of-Sight (LOS) guidance, the Pure Pursuit (PP) guidance, or the Constant Bearing (CB) guidance [44][45].

Among these guidance laws, a most widely used method is the LOS guidance algorithm including a waypoint switching system [16][46][47]. The LOS guidance algorithm creates arbitrary points called as LOS set points to be approached by a ship, which are two intersection points of a circle with certain radius around the ship and a straight line which connects two waypoints. In addition, a circle with the radius of one ship's length is established surrounding a waypoint, and the target waypoint toward where the ship goes is changed when the ship enters into the circle area. The size of the circle determines timing when a rudder is manipulated to change a maintained heading angle to enter new desired path. If a ship always uses a rudder at the same timing applying constant enclosed radius like in the LOS guidance algorithm, it is most likely to cause overshoots such as cross track error and heading error. Therefore, studies to derive optimal timing to use a rudder have been continued as adjusting the radius of the circle depending on the property of a ship, water depth, and so on so as to decrease the errors [17][18][48].

Rudder control system has progressed in various directions starting with the materialization of an autopilot for automatic steering [9][10][49]. The first generation autopilots implemented on ships were produced based on Proportional-Integral-Derivative (PID) controllers, PID-type autopilots are still very popular and represent a majority of autopilots in use. The PID controllers are effective for linear problem, but not suitable for nonlinear and complex system [50]. Motion of a ship is nonlinear problem and the problem becomes more complex when the ship is affected by external disturbances such as wind, current, and wave. Ship operators can perform a task of path following successfully under the external disturbances using their knowledge and experiences. Fuzzy logic applying the action of human experts has been known to be effective for the nonlinear and complex system. Hence, it used to enhance the rudder control system by combining the existing control theory [51][52][53][54].

In this chapter, path following algorithm consists of two components with fuzzy inference system based on fuzzy logic: one is a waypoint guidance system and the other is a rudder control system is presented. The waypoint guidance system is motivated by the LOS guidance algorithm and the system also uses a circle centered on a waypoint. However, the specific circular area is formed for each waypoint, and the radius of the circle is decided by the fuzzy inference system using course change angles. In other words, different optimal timing or a point to use a rudder is derived respectively according to course change angles in the waypoint guidance system. The timing is regarded as the setting of a point which minimizes overshoots when a ship enters a new desired straight line. In case of the rudder control system, three elements of cross track error, heading error, and yaw rate are considered. As taking the cross track error into account, it is not necessary to arrange a circle around a ship and it is able to omit troublesome computation for the setting of the LOS set points additionally.

However, the path following algorithm does not consider ship speed that is one of important factor which affects ship motion. Directly after using a rudder near a waypoint, the yaw rate of a ship at high speed increases faster than that at low speed. Therefore, rudder control system described above was ameliorated in consideration of the relationship between the current ship speed and the yaw rate. The former algorithm is named “the basic path following algorithm” and the speed consideration algorithm is called “the improved path following algorithm”. In order to prove the effectiveness of the proposed algorithm, numerical

simulations will be carried out in virtual environment having arbitrary external disturbances that are wind and current.

3.2 Fuzzy inference

In order to develop automatic path following algorithm, Fuzzy inference that is similar to human decision making process is utilized. Fuzzy inference is the process of mapping the output from given input with fuzzy rules using fuzzy logic that introduced by Zadeh [55]. Fuzzy logic can quantify the ambiguity such as linguistic variables of human. Fuzzy inference has two types, mamdani-type [56] and sugeno-type [57], that vary somewhat in the way to determine outputs. The path following algorithm of this study adopts mamdani-type fuzzy inference which is the most commonly utilized in fuzzy control filed. The process of mamdani-fuzzy inference involes all the pieces that are described in membership functions, logical operations and If-Then rules. The mamdani-type fuzzy inferece is performed in four steps:

- Fuzzifer of input variables
- Fuzzy rules definition
- Fuzzy inference
- Defuzzifier

The basic structure of the fuzzy inference is presented in a diagram shown in Fig. 3.1:

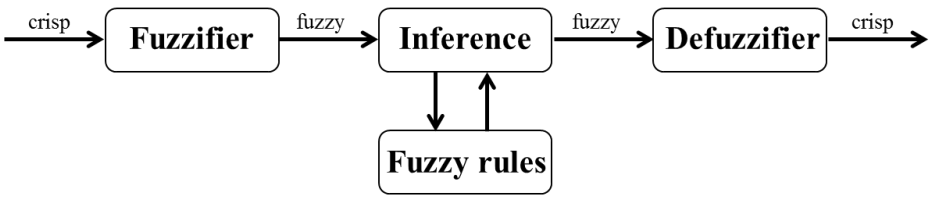


Fig. 3.1 Diagram of mamdani-type fuzzy inferece

As the first step, fuzzifier converts a crisp input to a linguistic variable using membership functions with the value between 0 and 1. Namely, fuzzification is to take inputs and determine the degree to let them belong to the each of appropriate fuzzy set via membership functions. Fuzzy rules are defined by expert's knowledge and indicate relationship between antecedent and consequent parts employing IF-THEN form. Using the fuzzy rules and inputs, fuzzy output is inferred by implication method and aggregation. Each linguistic rule can be computed through the implication. The input for the implication process is a single number given by the antecedent part, and the output is a fuzzy set. Aggregation is the process by which the fuzzy sets that represent the outputs of each rule are combined into a single fuzzy set. The input of the aggregation process is an output returned by the implication process for each rule. Lastly, the input for the defuzzification process is a fuzzy set and the output is a crisp value. In this research, all systems employing the fuzzy inference are applied as the implication with minimum method, the aggregation with maximum method and the defuzzification with centroid method.

3.3 Basic path following algorithm

3.3.1 Overview

A ship is required to travel on a pre-planned route consisting of several straight lines specified by two waypoints as shown in Fig. 3.2. In the figure, "from-waypoint" is the last passed waypoint and "to-waypoint" defines the target waypoint where the ship is approaching. Finally, "next-waypoint" indicates the waypoint following the "to-waypoint" [58]. Through path following algorithm proposed in this research, the ship goes to her destination without swerving from the given track. The algorithm is designed with waypoints guidance system and rudder control system. Fig. 3.3 displays the algorithm in diagram form. First of all, waypoints position data are fed to a track generator in order to configure a desired track. Using calculated course change angles θ, the waypoint switching system determines automatically a target waypoint. Parameters for the rudder control system are computed by contrasting the target waypoint positon with the current ship's position. Suitable rudder angle δ is derived from a heading error ψ_{e}, a non-dimensional cross track error $d'_{\mathrm{e}}(= d_{\mathrm{e}}/L)$ and a non-dimensional yaw rate $r'(= rL/U)$.

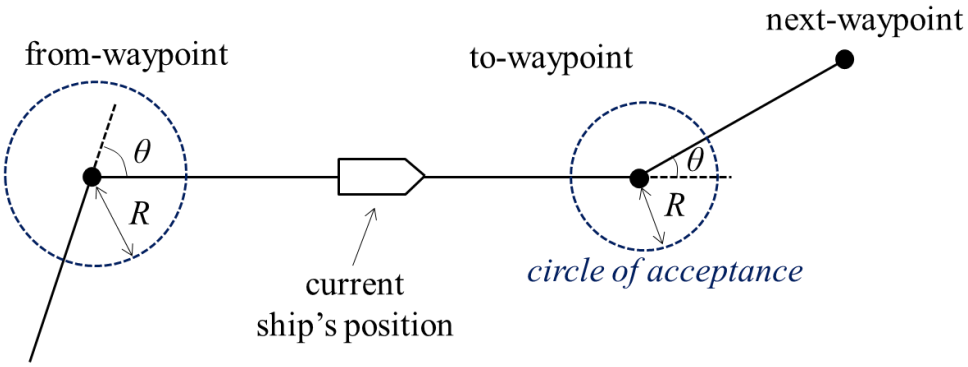


Fig. 3.2 Principle of waypoints guidance system

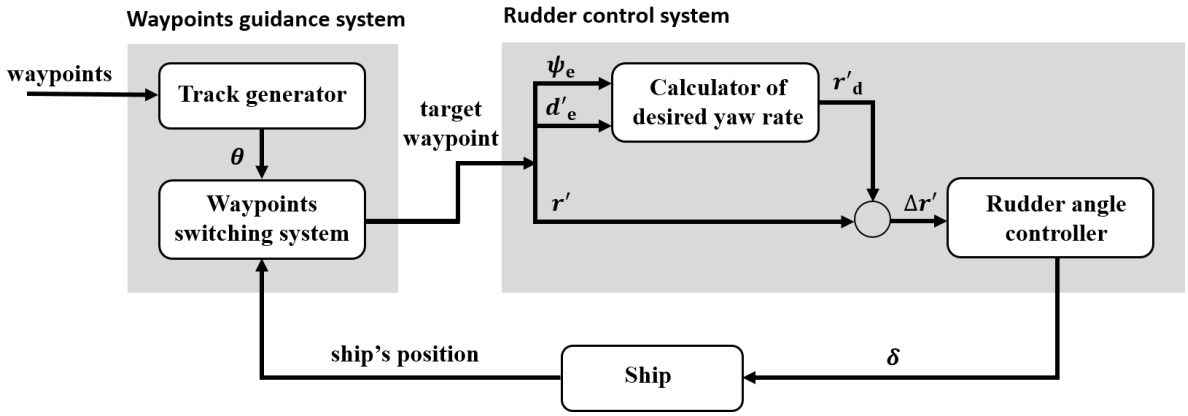


Fig. 3.3 Block diagram of path following algorithm

3.3.2 Waypoints guidance system

Waypoints guidance system is comprised of a track generator which makes a desired tack and waypoints switching system that guides a ship from one waypoint to the next one. The ship advances by repeating forward straight motion and turning motion along the desired route at sea. When a ship modifies her direction using a rudder in the proximity of a waypoint, there is a possibility that large overshoots occur such as heading error ψ_{e} and cross track error d_{e}. The waypoints switching system helps to find out optimal timings to manipulate a rudder to decrease the overshoots. The waypoint switching system was developed based on inspiration

from the LOS guidance algorithm. Once a ship comes within the circle of acceptance area, "next-waypoint" is automatically changed to "to-waypoint" that the ship should head toward as shown in Fig. 3.2. The LOS guidance algorithm needs a certain circle with constant radius around a waypoint. It is able to regard the length of the radius as an important factor to determine timing or a position at where a rudder is used. Namely, a rudder is executed at the same stage in the LOS guidance algorithm. However, the proposed waypoints guidance system deduces the suitable radius of a circle according to course change angles θ so as that a ship veers precisely towards new straight line without overshoots. In order to acquire the suitable non-dimensional radius value R' in the waypoints switching system which the fuzzy inference system was applied to, the following process was implemented as shown in Fig. 3.3.

Subsequent to inputting the locations of waypoints determined by ship operators in advance into a track generator, the course change angles of each waypoint are released as outputs. The obtained angles are entered into the waypoints switching system and each angle is utilized as an element to ordain an optimal timing to use a rudder. The course change angle θ is considered as the antecedent part and the non-dimensional radius of the circle of acceptance area R' is the consequent part of the membership functions of fuzzy inference respectively. Fuzzy rules and membership functions for the waypoints switching system are indicated in Table 3.1 and Fig. 3.4.

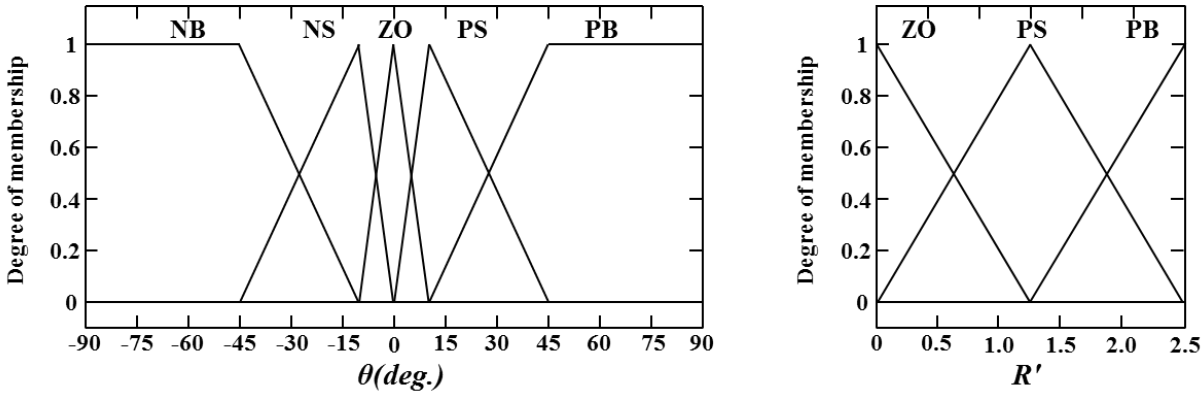
In Table 3.1, linguistic labels are defined as; NB: Negative Big, NS: Negative Small, ZO: Near Zero, PS: Positive Small, and PB: Positive Big. θ is subdivided into five classes that are NB, NS, ZO, PS and PB. The non-dimensional radius R' is classified into three classes such as ZO, PS and PB. In a typical ocean voyage, it is not common to set a course change angle more than 45˚ except for the case that a ship should change her direction greatly due to geographical conditions of ports. Hence, θ is established less than 45˚ in general situation. The minimum requirement for θ is set as 10˚ and maximum absolute value is limited not to exceed 90˚, namely θ is assigned within a range from -90˚ to 90˚. When a path which is specified by "to-waypoint" and "next-waypoint" locates on the clockwise direction of a previous track which consists of "from-waypoint" and "to-waypoint", the sign of θ is defined as positive.

The turning ability of ships is taken into consideration at settling the R' value because ships should trace a circular path in order to modify their courses. In accordance with the criteria for initial turning ability legislated by the IMO for ship's manoeuvrability [59], ships should

not have travelled more than 2.5 ship lengths by the time heading angle has changed by 10˚ from the original heading with the application of 10˚ rudder angle. The distance is able to be deemed as the maximum distance which is neccesary to execute turning motion since a ship can not turn immediately after rudder execution. Therefore, 2.5 was utilized as the value of R' to set up the consequent part of fuzzy inference, and R' has always positive sign because it presents the magnitude of radius. Consequently, a ship can closely follow preplanned track since the value of R'decreases as the value of θ decreases. In other words, the large θ, the slower steering is performed for turning. It is also expected that less overshoots will occur because the rudder angle is adjusted immediately after effective rudder force is generated by the rudder control system described in the next section.

Table 3.1 Rules for the waypoint swithcing system of fuzzy inference

	θ	
NB:PB	ZO	NS:PS
PB	ZO	PS



(a) antecedent part (b) consequent part

Fig. 3.4 Membership functions for the waypoints switching system of fuzzy inference

3.3.3 Rudder control system

The path following problem is decomposed into two subtasks. The first one is to follow a desired heading and the second is to bring a ship onto a desired path and keep track. The two assignments can be resolved by means of the minimization of both heading error ψ_{e} and cross track error d_{e} through rudder control system. The heading error ψ_{e} is a difference between the desired heading angle ψ_{d} and the current heading angle ψ. The cross track error d_{e} is defined as the distance from the ship's present position to the closest point on a straight line drawn by connecting two waypoints which represents the leg of current route as shown in Fig. 3.5. (x_t, y_t) means the current position of a ship and an arrow which goes out from the point indicates the heading direction of own ship. (x_1, y_1) and (x_2, y_2) show the positions of "from-waypoint" and "to-waypoint" respectively. The red line denotes the cross track error d_{e} and is calculated as Eq. (3.1).

$$d_{\mathrm{e}} = \sqrt{(x_2 - x_t)^2 + (y_2 - y_t)^2} \cdot \sin\psi_{\mathrm{to}} \tag{3.1}$$

Where, ψ_{to} is a bearing angle formed by difference between ψ_{leg} and ψ_{d}. ψ_{leg} represents a bearing angle of a leg composed with (x_1, y_1) and (x_2, y_2). They are described by Eq. (3.2).

$$\begin{cases} \psi_{\mathrm{leg}} = \tan^{-1}\dfrac{x_2 - x_1}{y_2 - y_1}, \\ \psi_{\mathrm{d}} = \tan^{-1}\dfrac{x_2 - x_t}{y_2 - y_t}, \\ \psi_{\mathrm{to}} = \psi_{\mathrm{leg}} - \psi_{\mathrm{d}}. \end{cases} \tag{3.2}$$

To decrease the errors ψ_{e} and d_{e}, the rudder control system exploited not only the both errors but also the yaw rate of a ship as input data. However, consideration of the three parameters in the antecedent part puts some strain on the fuzzy control system and is likely to cause a delay in response. Hence, the rudder control system was divided into two phases. The first phase is calculator of desired yaw rate and the second phase is rudder angle controller as shown in Fig. 3.1. The desired yaw rate r'_{d} is yielded from ψ_{e} and $d'_{\mathrm{e}}(= d_{\mathrm{e}}/L)$ at the calculator of desired yaw rate.

Table 3.2 and Fig. 3.7 show rules and membership functions of the first phase. In case that a target waypoint is changed by the waypoint switching system while a ship travels on the existing track correctly, the value of ψ_e becomes similar to the value of θ. Thus, the antecedent part ψ_e was settled by reflecting the value of θ of Fig. 3.4. ψ_e and d'_e are subdivided into five classes: NB, NS, ZO, PS and PB. The clock-wise direction of ψ_e means as positive sign. If a ship locates at the right-hand side of the desired path, d'_e is defined as positive. As a consequent part, r'_d required for the second phase is in the range between -0.3 and 0.3. The membership function values for r'_d brought from the results of spiral test at rudder angle of 10˚ regarded as practically used rudder angle [38]. Through the rudder angle controller, a proper rudder angle δ is reasoned according to $\Delta r'(= r'_d - r')$.

Table 3.3 and Fig. 3.6 present the rules and the membership functions of the rudder control system. $\Delta r'$ is subdivided into three classes with the following linguistic labels; N: Negative, ZO, and P: Positive. The membership function of δ is set at 10˚ as a normal rudder angle and the maximum rudder angle will not exceed 35˚ because the purpose of this research is to realize a realistic ship operation using an angle of less than 10˚.

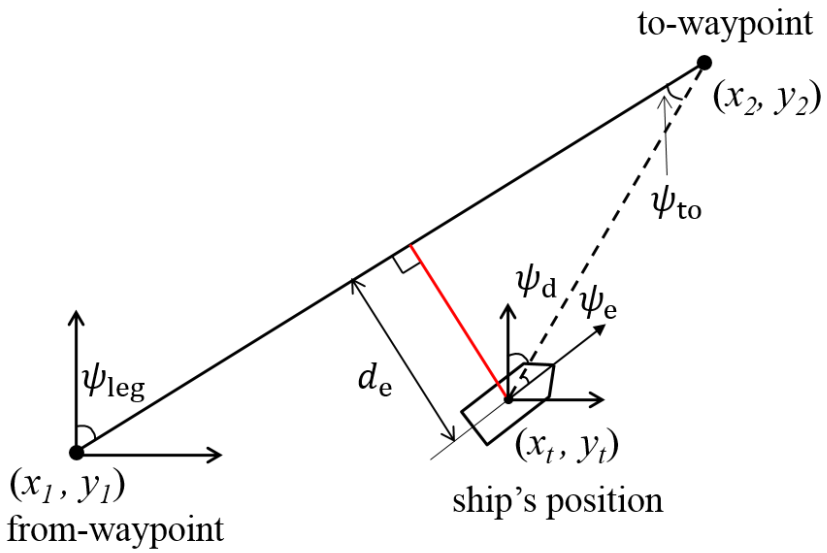


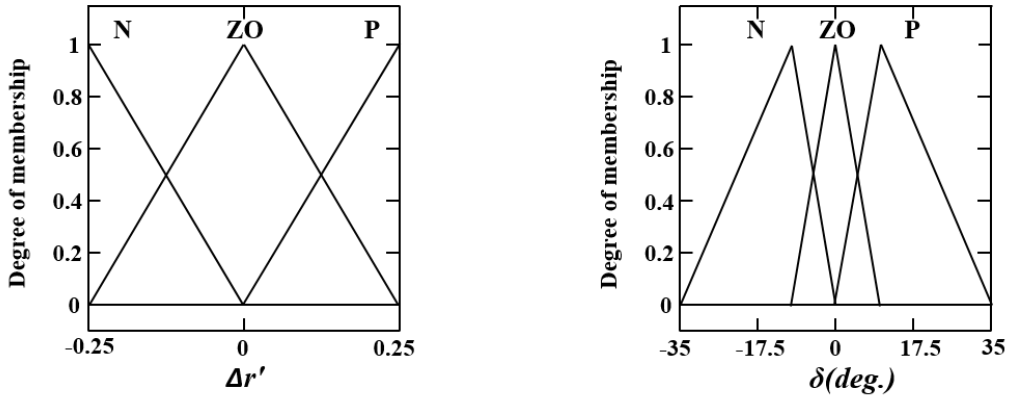
Fig. 3.5 Parameters used in the rudder control system

Table 3.2 Rules of fuzzy inference for the rudder angle controller

		ψ_e				
		NB	NS	ZO	PS	PB
d'_e	NB	PB	PB	PM	PS	NS
	NS	PB	PS	ZO	NS	NM
	ZO	PB	PS	ZO	NS	NB
	PS	PM	PS	ZO	NM	NB
	PB	PS	NS	NM	NB	NB

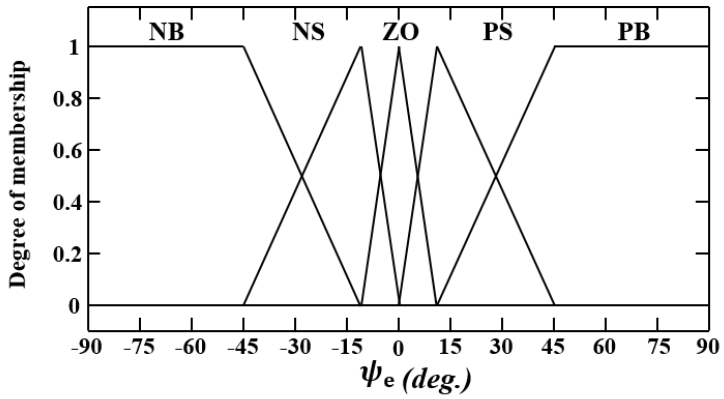
Table 3.3 Rules of fuzzy inference for the waypoint swithcing system

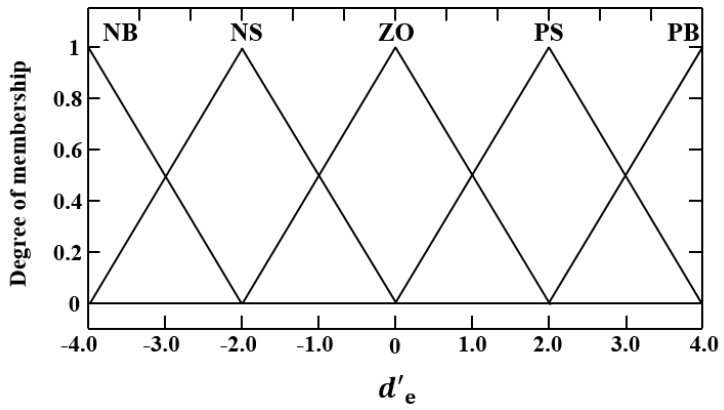
$\Delta r'$		
N	ZO	P
P	ZO	P



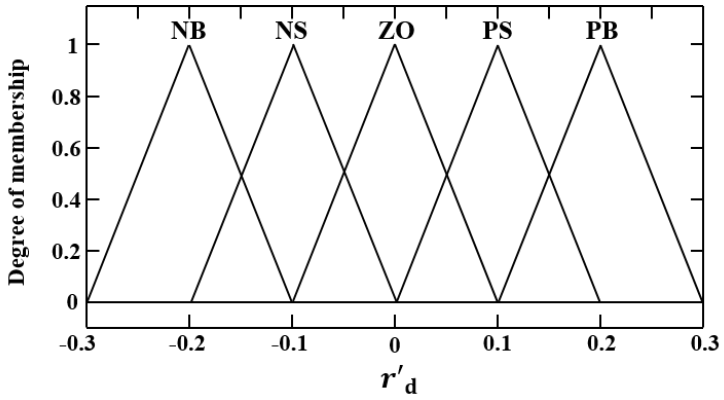
(a) antecedent part (b) consequent part

Fig. 3.6 Membership functions of fuzzy inference for the rudder angle controller





(a) antecedent parts



(b) consequent part

Fig. 3.7 Membership functions of fuzzy inference for the calculator of desired yaw rate

3.4 Numerical simulations in virtual situation

3.4.1 Simulation conditions

A 1:128 scale model of the KVLCC2 was adopted as a subject ship for numerical simulations to verify the effectiveness of the aforementioned path following algorithm. The principle particulars of the ship are shown in Table 2.2 in Chapter 2. The numerical simulations were carried out on five cases with different conditions as shown in Table 3.4. The subject ship was influenced by environmental disturbances such as wind and current in the numerical simulations in order to depict realistic ship motion during a voyage.

Four cases which have different combinations of wind and current were made. In these cases, a course change angle was set as 45˚. Wind velocity was selected as 0.36 m/s (Beaufort scale 3) and 1.68 m/s (Beaufort scale 6) with wind direction ν = -35˚. The velocities correspond to 8.0 knots and 36.9 knots respectively in full scale condition and the later wind speed is stipulated as an adverse condition by the IMO [60]. Current of 0.09 m/s in speed coming from α = 0˚ was assumed. It corresponds to 2.0 knots in full scale condition. The combinations of the four cases are shown in Table 3.4 as the Cases 3.1 to 3.4.

Additionally, 'S-shape' course was formed referring to a previous study [40] and a simulation carried out so as to figure out the ship's operation at various course change angles under wind (V_W = 0.36 m/s, ν = -35˚) and current (V_C = 0.09 m/s, α = 0˚). This case is shown as the Case 3.5 in Table 3.4. It was assumed that a ship traced existing course line precisely using a check helm responding to external disturbances at the beginning of the simulations. Ship's initial velocity for all of the simulations is 0.45 m/s scaled from 10.0 knots which is the normal operation speed of full scale tanker.

Table 3.4 Definitions of conditions for simulations

Case	Wind (ν = -35˚)	Current (α = 0˚)
3.1	0.36 m/s	-
3.2	0.36 m/s	0.09 m/s
3.3	1.68 m/s	-
3.4	1.68 m/s	0.09 m/s
3.5	0.36 m/s	0.09 m/s

3.4.2 Simulation results

Under the assumed virtual environmental situation, the simulations were carried out according to the cases presented in Table 3.4. Fig. 3.8 shows the result for the Case 3.1 with wind from ν = -35˚. Figures (a) and (b) show ship trajectory and the time history of rudder angle obtained by the proposed algorithm. After manipulating rudder to change course, rudder has reached to its neutral position. Then, opposite rudder was used to enter into the new course line. It means modifying rudder angle from starboard side to port side in a short moment was prevented and it helps safe operation. In Fig. 3.8(d), drift angle β and non-dimensional yaw rate r'changed according to the rudder motion. The relationship between heading error ψ_{e} and cross track error d'_{e} can be observed in Fig. 3.8(e). When the ship comes within the circle of acceptance, the absolute value of ψ_{e} increases and the cross track error d'_{e} also begins to increase simultaneously. The increment of d'_{e} value is caused by the modification of "to-waypoint" position according to Eq. 3.7. Therefore, it should be noted that actual overshoot values of d'_{e} and ψ_{e} appear around the time at which overshoot of heading angle ψ is observed. Even though the course changing angle 45˚ formed by the waypoints can be recognized as large angle to change course, used rudder angle was less than 10˚. The actual overshoot values were observed as $|\psi_{\mathrm{e}}|$ = 5.38˚ and $|d'_{\mathrm{e}}|$ = 0.33, however, they converged to zero immediately by opposite rudder.

Fig. 3.9 shows the result of the Case 3.2 with wind from ν = -35˚ and current from α = 0˚. The rudder angle used for changing course was less than 10˚ as same as that observed in the Case 3.1. But, the opposite rudder angle to keep new course line increased due to the influence of current as shown in Fig. 3.9(b). As seen from comparison with the result of the Case 3.1, it took more time to reach the last waypoint. The values of both errors increased to $|\psi_{\mathrm{e}}|$ = 9.34˚ and $|d'_{\mathrm{e}}|$ = 0.68 respectively according to Fig. 3.9(e).

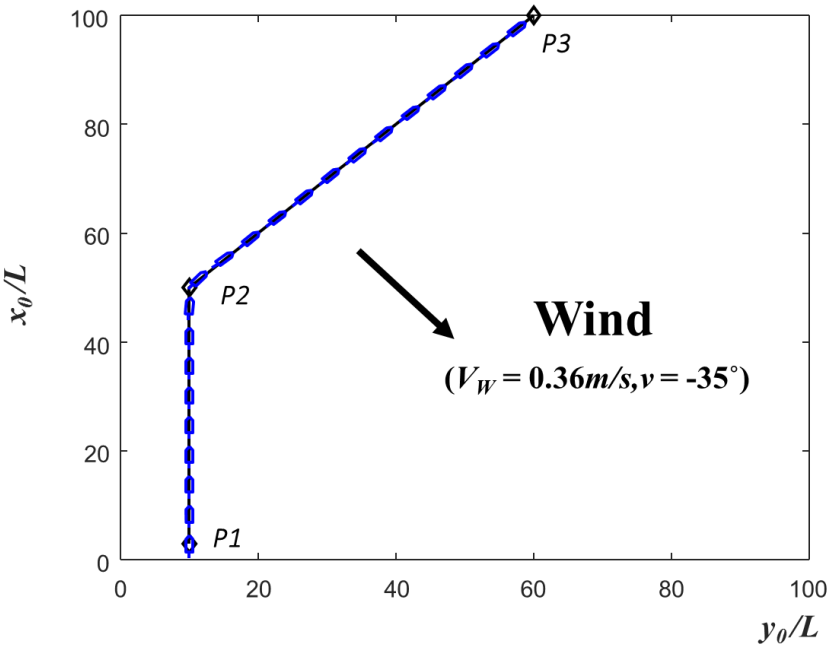
Simulation of the Case 3.3 was performed with the assumption that a ship was operated under the adverse sea condition with wind of 1.68 m/s. The magnitude of the errors appears as $|\psi_{\mathrm{e}}|$ = 3.76˚ and $|d'_{\mathrm{e}}|$ = 0.25 in Fig. 3.10(e). Since strong wind blows from the port side, yaw moment that forces the ship to turn to starboard side increased comparing with that in the Case 3.1, hence the value of ψ_{e} decreased. Even though the wind velocity has grown faster, path following shows a good result as shown in Fig. 3.10(a).

Fig. 3.11 shows the simulation corresponds to the Case 3.4 with wind of 1.68 m/s blowing from ν = -35˚ and current of 0.09 m/s from α = 0˚. Because of the large magnitude of

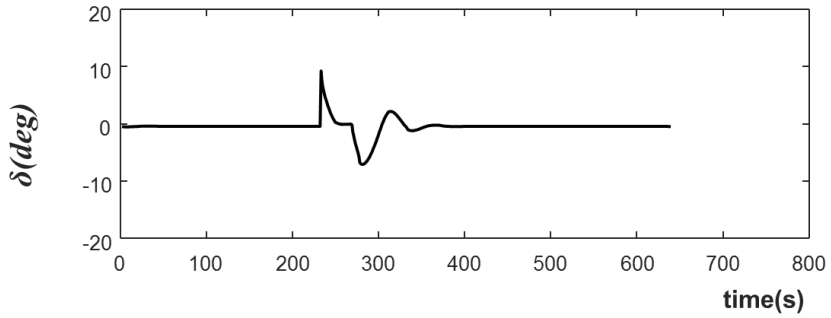
disturbances, a slightly big check helm angle was applied as an initial rudder angle. With this point of view, the opposite rudder angle is about -15˚ but substantive variation has been less than 10˚. Fig. 3.11(e) indicates that the values of both errors are $|\psi_e|$ = 6.04˚ and $|d'_e|$ = 0.33. Comparing with the results of the Case 3.3, the influence of current has resulted in a greater variation of ψ_e than d'_e.

Waypoints in the Case 3.5 were set up as 'S-shape' with various course change angles in the range from -74˚ to 55˚. The course change angles were excessively established to estimate the proposed algorithm. The simulation was performed under the influences of wind of 0.36 m/s from α = 0˚ and current of 0.09 m/s from α = 0˚. Fig. 3.12 illustrates the results of the simulation for the Case 3.5. When the course change angle is more than 60˚, the overshoots of ψ_e and d'_e increased comparing with those of the other course change angles. As for rudder angles used to change course near the waypoints, none of them exceeds 10˚.

As shown in Table 3.4, simulations of the Cases 3.1 and 3.2 were carried out with wind of 0.36 m/s and with/without current. The Cases 3.3 and 3.4 were conducted with wind of 1.68 m/s and with/without current. The results between the two pairs of cases with the same wind speed were compared to come up with the effect of current. The effects of current can be analyzed through the rate of the increase of the values of both errors ψ_e and d'_e. In case of ψ_e, the error increased in similar proportion due to the effect of current regardless of wind speed. On the other hand, the rate of increase in the value of d'_e becomes higher as wind speed becomes slower. Consequently, current has a great effect on lateral force acting on the ship rather than yaw moment when wind velocity is slow. Furthermore, all of simulation results showed the ship could closely follow desired track using realistic rudder angle because rudder was executed at proper timing or point depending on course change angle.

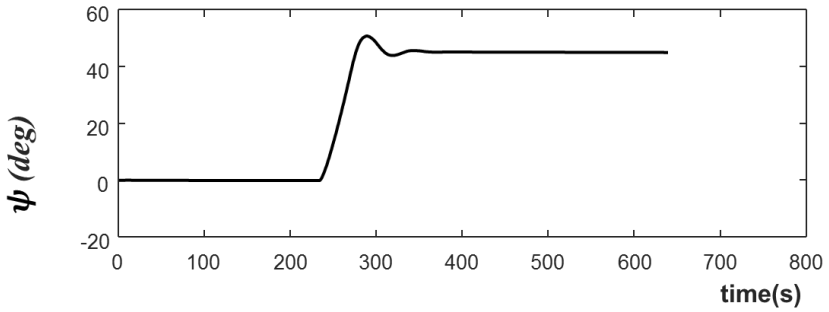


(a) Ship trajectory

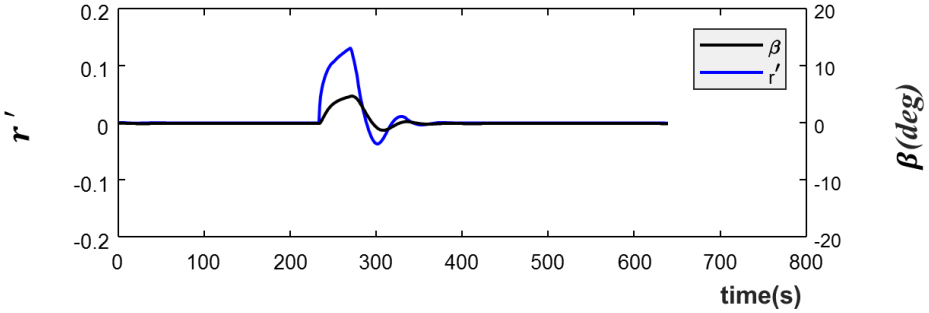


(b) Time history of rudder angle

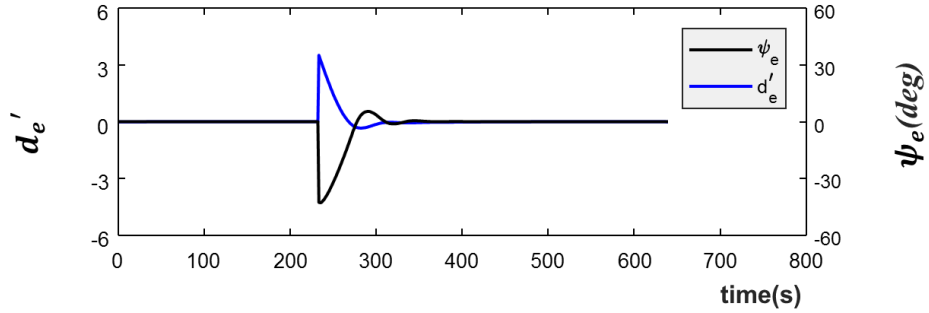
Fig. 3.8 Simulation results of the Case 3.1



(c) Time history of heading angle

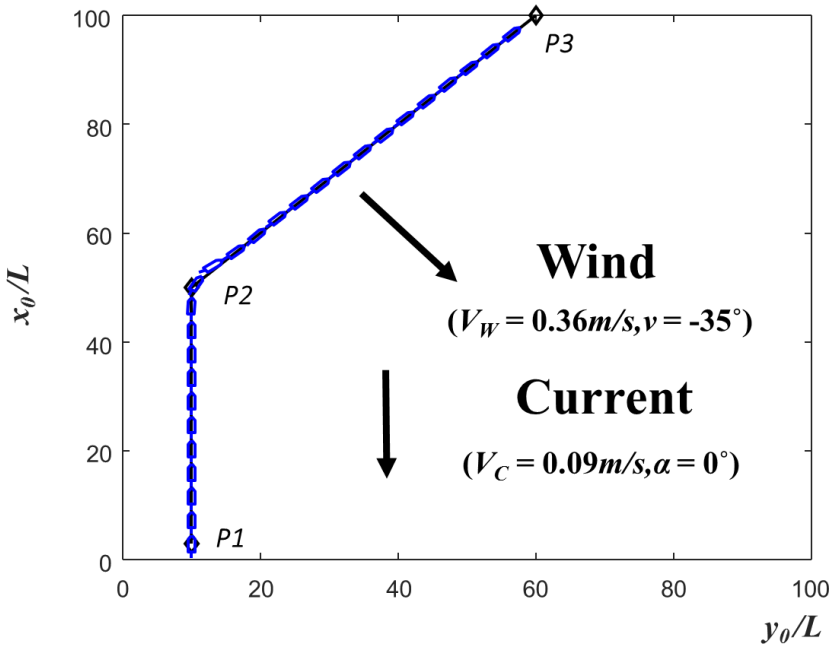


(d) Time history of drift angle and non-dimensional yaw rate

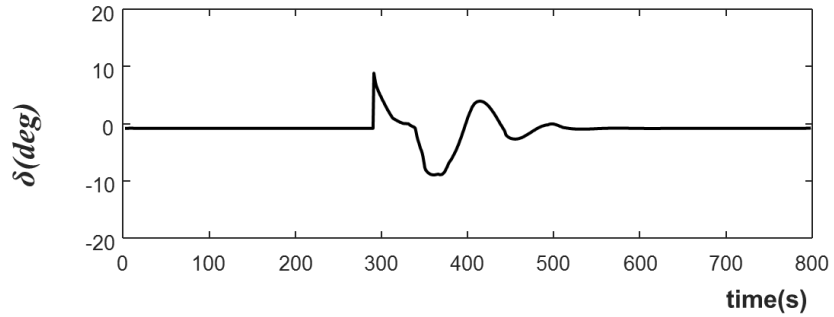


(e) Time history of heading error and non-dimensional cross track error

Fig. 3.8 Simulation results of the Case 3.1

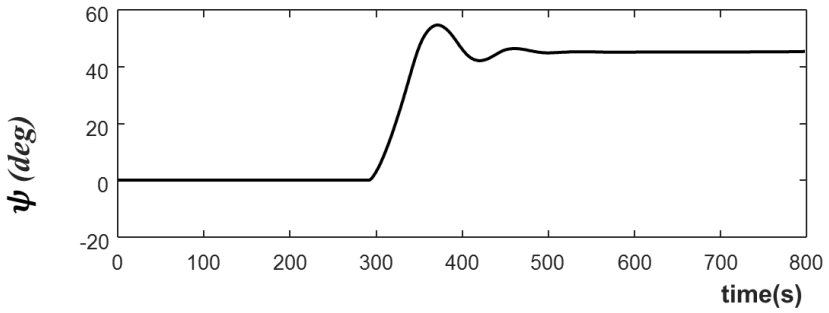


(a) Ship trajectory

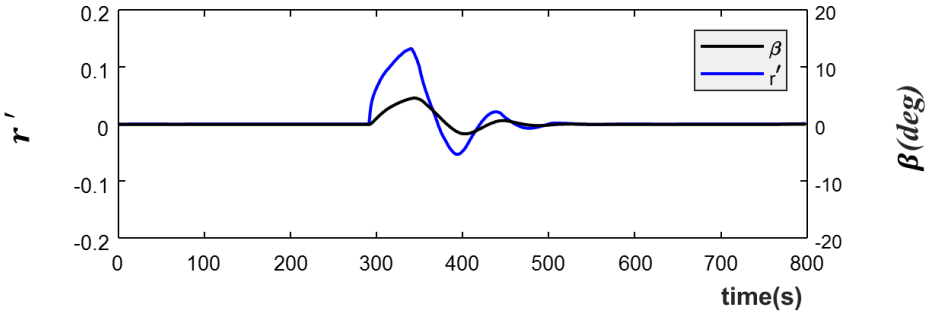


(b) Time history of rudder angle

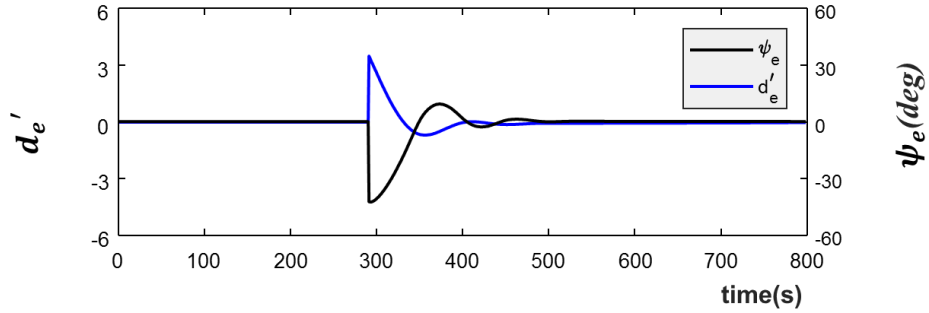
Fig. 3.9 Simulation results of the Case 3.2



(c) Time history of heading angle

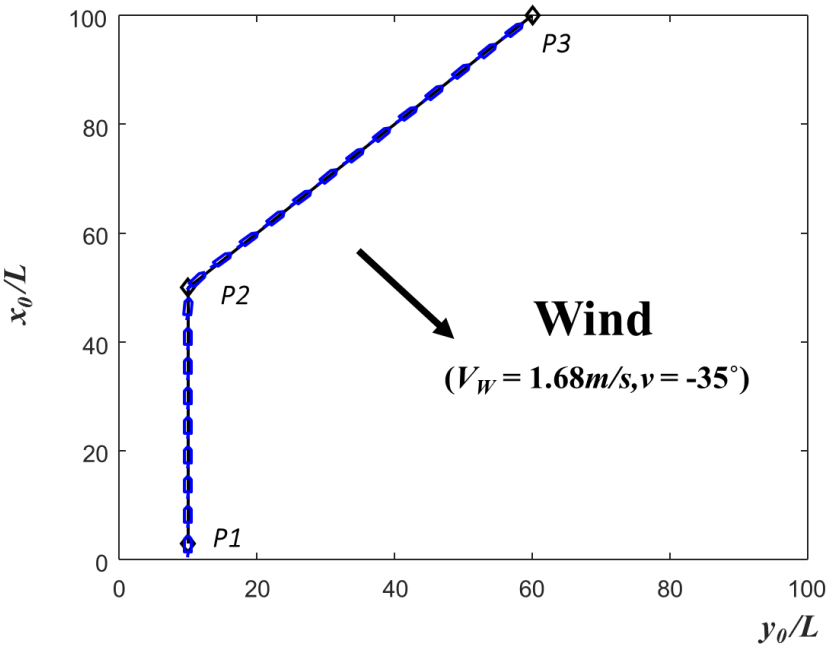


(d) Time history of drift angle and non-dimensional yaw rate

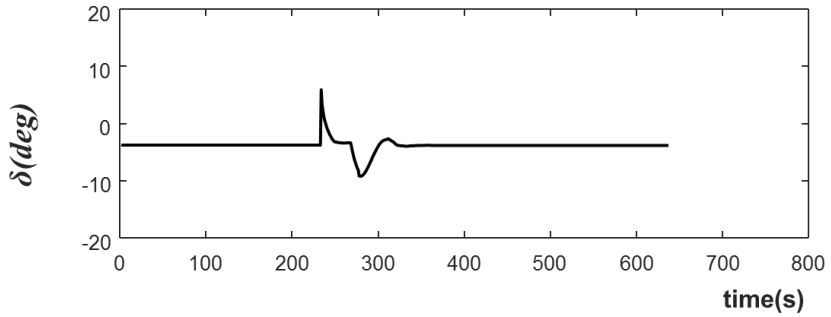


(e) Time history of heading error and non-dimensional cross track error

Fig. 3.9 Simulation results of the Case 3.2

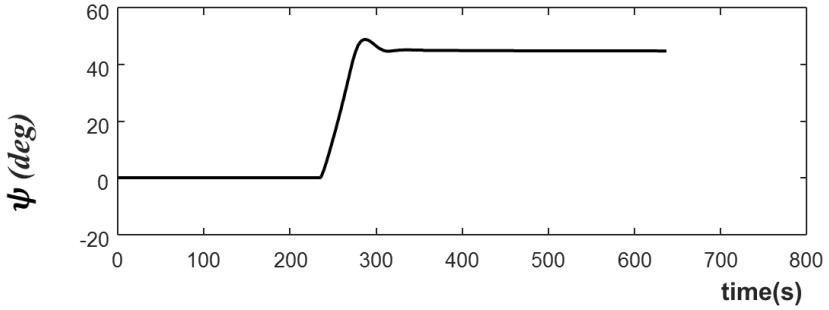


(a) Ship trajectory

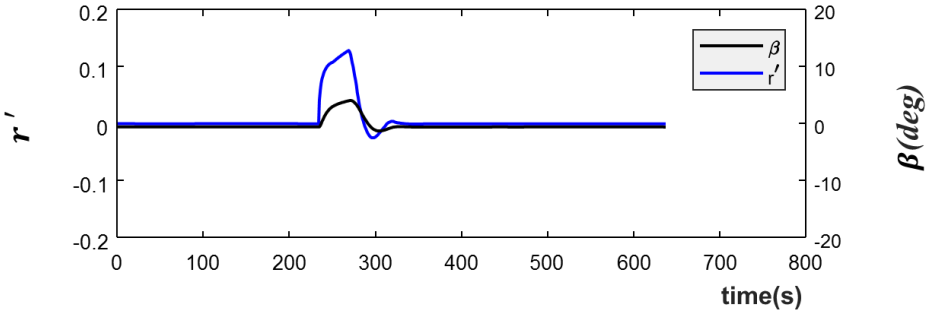


(b) Time history of rudder angle

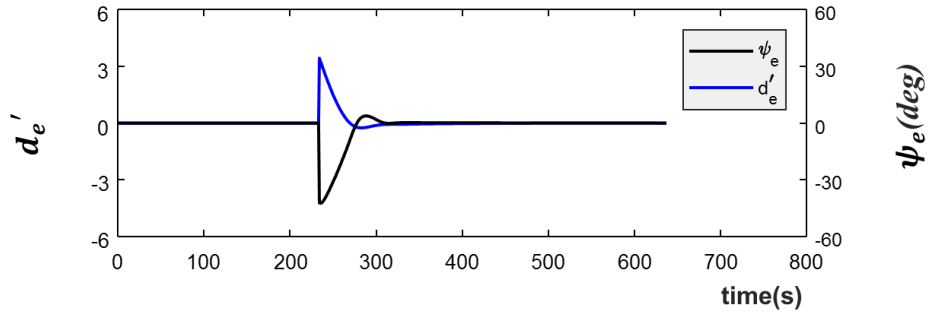
Fig. 3.10 Simulation results of the Case 3.3



(c) Time history of heading angle

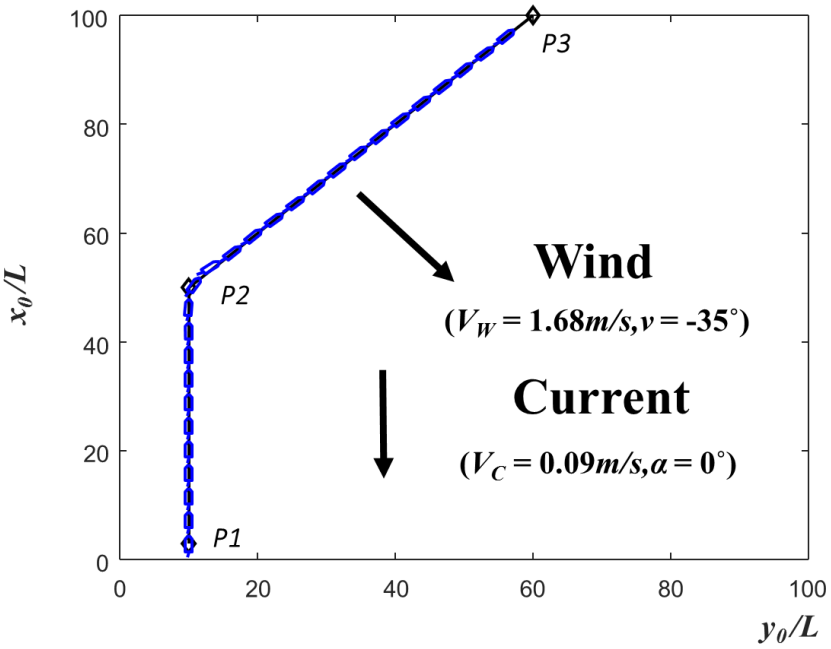


(d) Time history of rudder angle and non-dimensional yaw rate

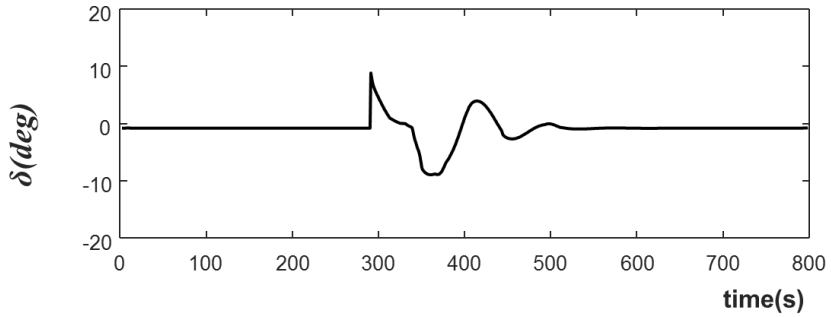


(e) Time history of heading error and non-dimensional cross track error

Fig. 3.10 Simulation results of the Case 3.3

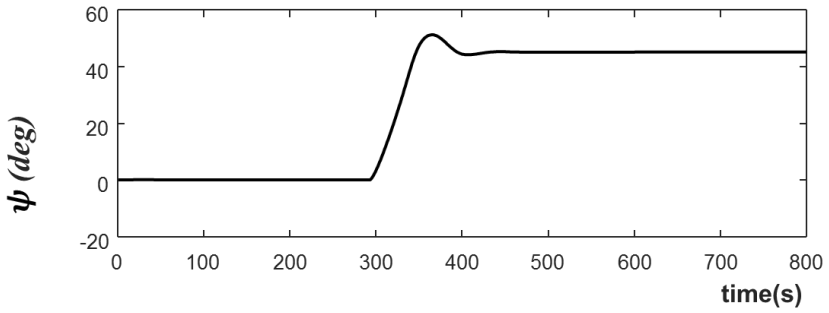


(a) Ship trajectory

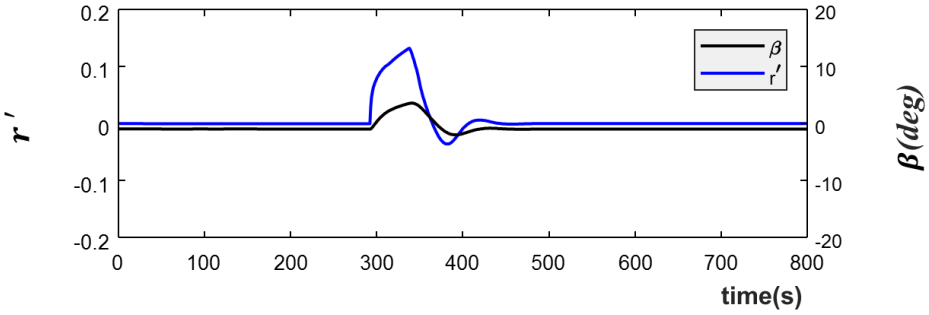


(b) Time history of rudder angle

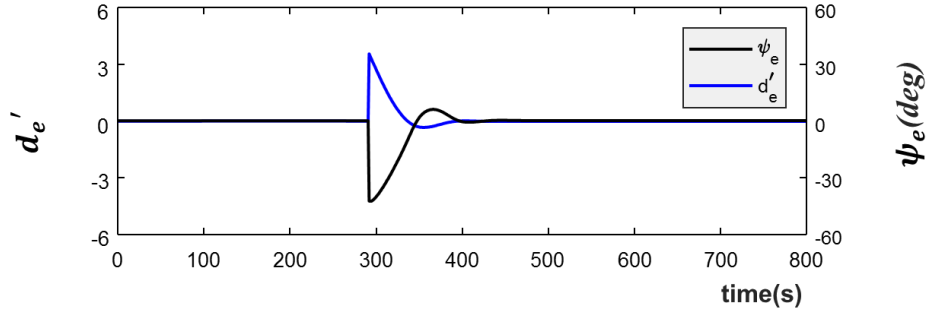
Fig. 3.11 Simulation results of the Case 3.4



(c) Time history of heading angle

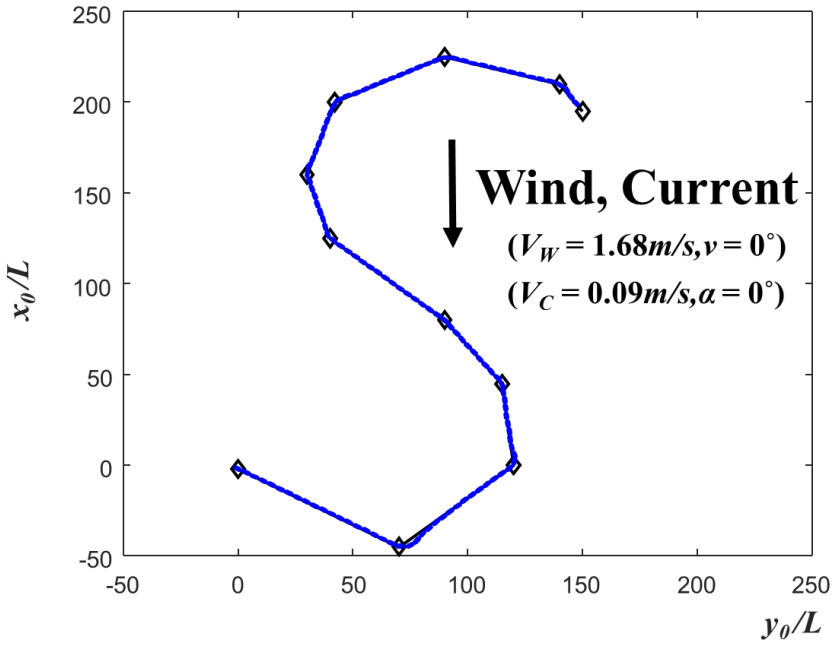


(d) Time history of drift angle and non-dimensional yaw rate

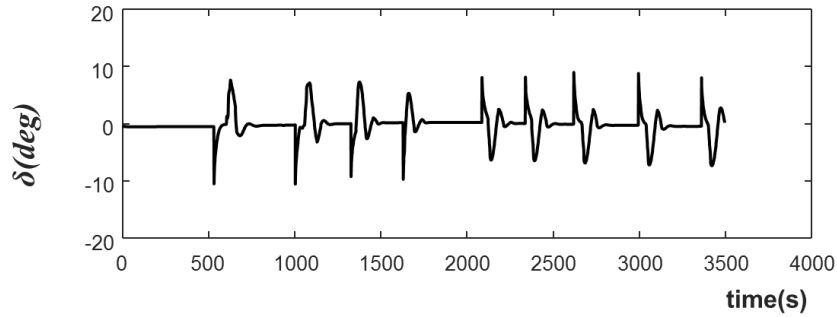


(e) Time history of heading error and non-dimensional cross track error

Fig. 3.11 Simulation results of the Case 3.4

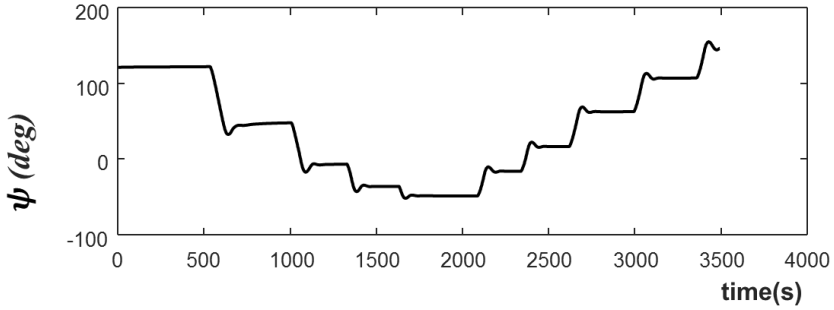


(a) Ship trajectory

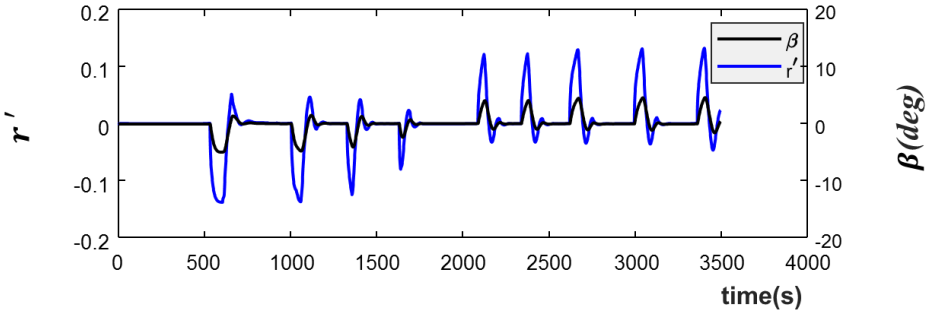


(b) Time history of rudder angle

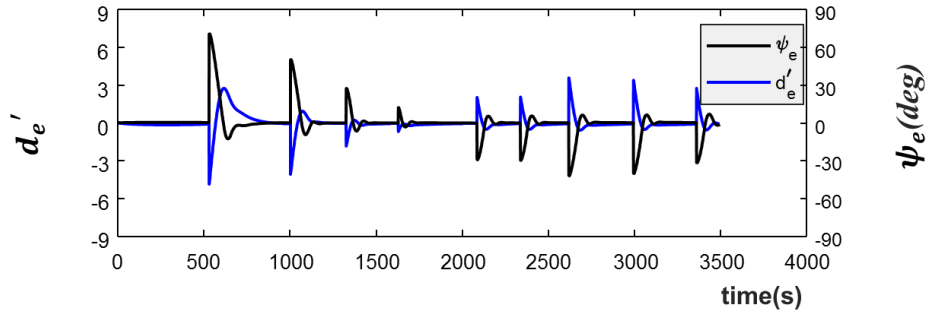
Fig. 3.12 Simulation results of the Case 3.5



(c) Time history of heading angle



(d) Time history of drift angle and non-dimensional yaw rate



(e) Time history of heading error and non-dimensional cross track error

Fig. 3.12 Simulation results of the Case 3.5

3.5 Improved path following algorithm

The improved path following algorithm is composed of two components such as a waypoints guidance system and a rudder control system. The waypoint guidance system described in the previous sections was applied as an existing method. However, a rudder control system was newly developed by adding ship speed parameters. The previously proposed rudder control system demanded three input parameters of heading error, cross track error, and yaw rate in order to bring out a rudder angle. The speed of a ship is also one of important factors to determine optimal rudder angle in that yaw rate varies depending on the magnitude of ship's velocity. Thus, unlike the previous system, current speed of vessel is taken into account additionally and the processing of the necessary elements is illustrated in Fig. 3.13. Four elements of cross track error, heading error, yaw rate and ship speed are considered as input data in the improved rudder control system. It is somewhat burden to consider four parameters in the antecedent part due to the possibility to cause the delay of response. To overcome the anxiety, the fuzzy inference of the rudder control system is dived into two phases.

Firstly, the calculator of desired yaw rate as the first step will be demonstrated in detail. The proper r'_{d} is inferred using 7×7 different combinations of two inputs ψ_{e} and d'_{e}. Related fuzzy rules and membership functions are shown in Table 3.5 and Fig. 3.14.

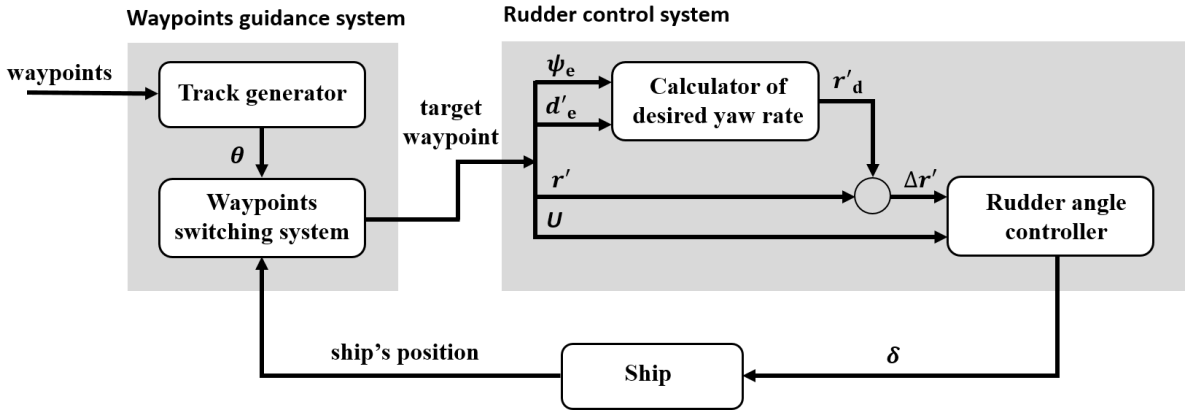
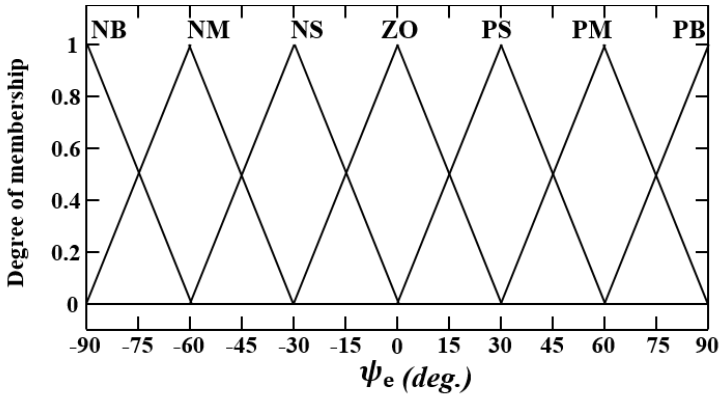


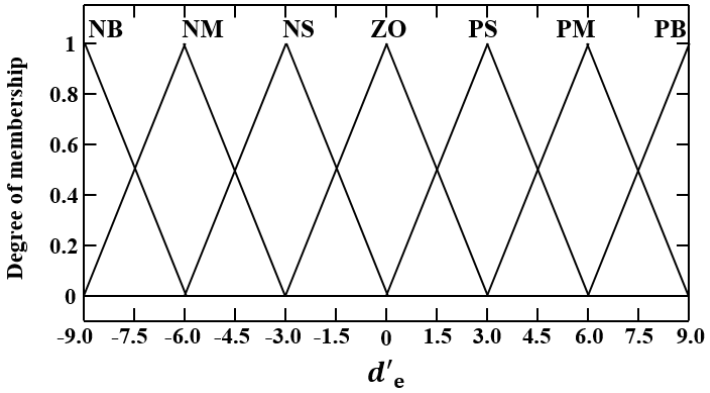
Fig. 3.13 Block diagram of improved path following algorithm

The range of ψ_e is offered from -90˚ to 90˚ in common with course change angle θ, because when a ship travels on an existing track with pinpoint accuracy as ψ_e = 0˚, ψ_e would be similar angle to θ. d'_e is established referring the study shown in the previous section, however, the study did not take disturbance forces into account, so the range of d'_e is expanded further. ψ_e and d'_e are subdivided into seven classes; NB, NM, NS, ZO, PS, PM, and PB. Here, N and P is abbreviation for "Negative" and "Positive" signifying the sign of them. As for heading error ψ_e, clockwise movement is described as positive. If a ship locates at the right-hand side of the target track, cross track error d'_e has positive sign. S, M, and B represent abbreviation of quantity such as "Small", "Medium", and "Big" respectively. The rudder angle is assumed to be able to manipulate up to 30˚ in case of requiring urgent veering. r'_d as consequent part is formed between -0.6 and 0.6 applying r' = 0.6 based on the result of spiral test carried out by Kijima et al. at δ = 30˚ [25]. r'_d is also classified into seven classes; NB, NM, NS, ZO, PS, PM, and PB.

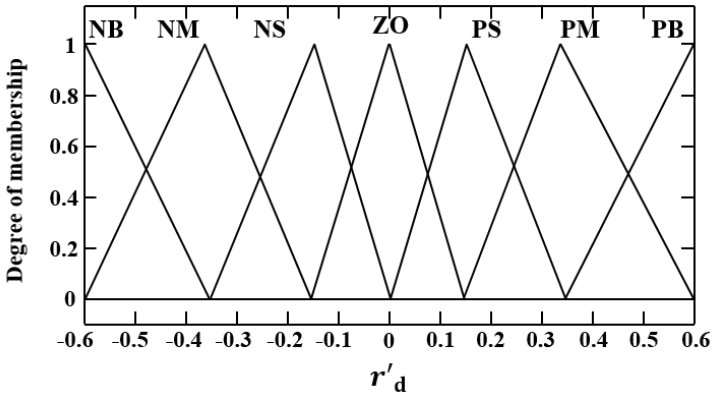
Table 3.5 Rules of fuzzy inference for the calculator of the calculator of desired yaw rate

		ψ_e						
		NB	NM	NS	ZO	PS	PM	PB
d'_e	NB	PB	PB	PB	PM	PS	PS	NS
	NM	PB	PB	PB	PS	PS	NS	NS
	NS	PB	PM	PS	ZO	NS	NS	NM
	ZO	PB	PM	PS	ZO	NS	NM	NB
	PS	PM	PS	PS	ZO	NM	NM	NB
	PM	PM	PS	NS	NS	NB	NB	NB
	PB	PS	NS	NS	NM	NB	NB	NB





(a) antecedent parts



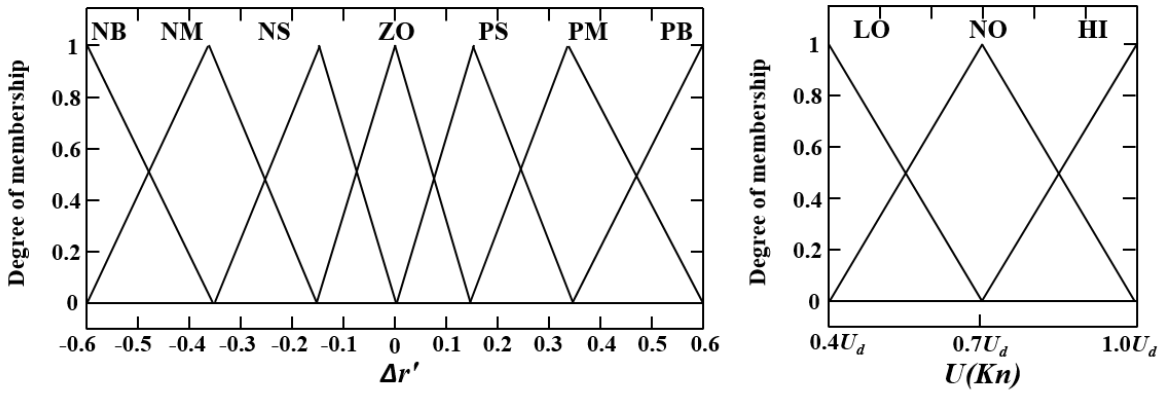
(b) consequent part

Fig. 3.14 Membership functions of fuzzy inference for the calculator of desired yaw rate

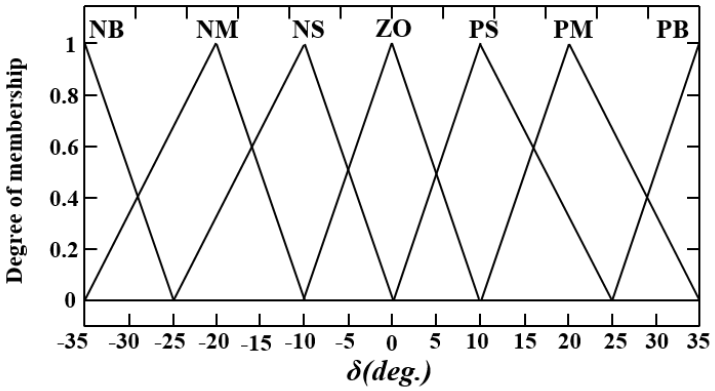
As the second step in the rudder angle controller, the author employed relationship between $\Delta r'$ and U in order to obtain δ which makes $\Delta r'$ reach to zero. The rudder angle controller determines how smoothly and swiftly make a ship enter the new desired track. 21 (= 7 × 3) fuzzy rules and membership functions that have introduced $\Delta r'$ and U as input data are presented in Table 3.6 and Fig. 3.15. $\Delta r'$ is specified with seven classes by the same membership function of r', and U is subdivided into three classes; LO, NO, and HI. The linguistic labels abbreviated "Low speed", "Normal speed", and "High speed" respectively. In general, a ship sails the ocean at 70% ~ 80% of design speed U_{d} for an energy efficiency operation. Thus, the shape of membership function of normal speed is appointed to triangle having 70% of U_{d} as an apex, and the maximum speed is limited to 100% of U_{d}. δ is arranged between -35˚ and 35˚. It is known that if the ship travels at high speed, yaw rate increases. If the velocity of ship is fast, ψ reveals substantial variations even though using small δ. The rudder angle controller is designed to use less δ as U increases, thereby avoiding a large rolling motion is expected.

Table 3.6 Rules of fuzzy inference for the the rudder angle controller

		$\Delta r'$						
		NB	NM	NS	ZO	PS	PM	PB
U	LO	NB	NB	NB	ZO	PB	PB	PB
	NO	NB	NM	NS	ZO	PS	PM	PM
	HI	NM	NS	NS	ZO	PS	PS	PM



(a) antecedent parts



(b) consequent part

Fig. 3.15 Membership functions of fuzzy inference for the rudder angle controller

3.6 Numerical simulations of the improved path following algorithm

3.6.1 Simulation conditions

A 1:128 scale model of the KVLCC2 was adopted as a subject ship for numerical simulations to verify the effectiveness of the improved path following algorithm. The principle particulars of the ship are shown in Table 2.2 of Chapter 2. The numerical simulations were carried out on four cases with different conditions as shown in Table 3.7. The subject ship was influenced by environmental disturbance such as wind and current in the numerical simulations in order to depict ship motion during a voyage. Three cases in which each ship speed was different were set using the same conditions of external disturbances and desired tracks so as to compare the variation of control performance depending on ship speed. Low, normal and high speeds were selected as 0.32 m/s, 0.50 m/s, and 0.68 m/s which belong to the specified range shown in Fig. 3.15. They coincide with 7 knots, 11 knots, and 15 knots in full scale condition. Current velocity was supposed to be V_C = 0.05 m/s coincides with 1 knot and it came from the direction of α = 0˚. Wind speed V_W was assigned to 0.36 m/s corresponding to 8 knots (Beaufort scale 3) and it blew from the direction of ν = 20˚. The desired track created by three waypoints owns course change angle θ = 30˚ and a veering action took place once near the mid-waypoint of the given waypoints. In the fourth case, a ship was required to follow a S-shape course formed by fourteen waypoints designated from a previous study [2] in order to validate the availability of the present algorithm at various course change angle θ under external disturbances coming from arbitrary directions. At that time, normal speed was applied to substantiate a general operation of the ship. In the all cases, it was assumed that a ship departed from the first waypoint position and stopped moving when the ship arrived at the final waypoint.

Table 3.7 Definitions of conditions for simulations

Case	U	V_W	ν	V_C	α
3.6	0.32 m/s	0.36 m/s	20 ˚	0.05 m/s	0 ˚
3.7	0.50 m/s	0.36 m/s	20 ˚	0.05 m/s	0 ˚
3.8	0.68 m/s	0.36 m/s	20 ˚	0.05 m/s	0 ˚
3.9	0.50 m/s	0.36 m/s	-25 ˚	0.05 m/s	0 ˚

3.6.2 Simulation results

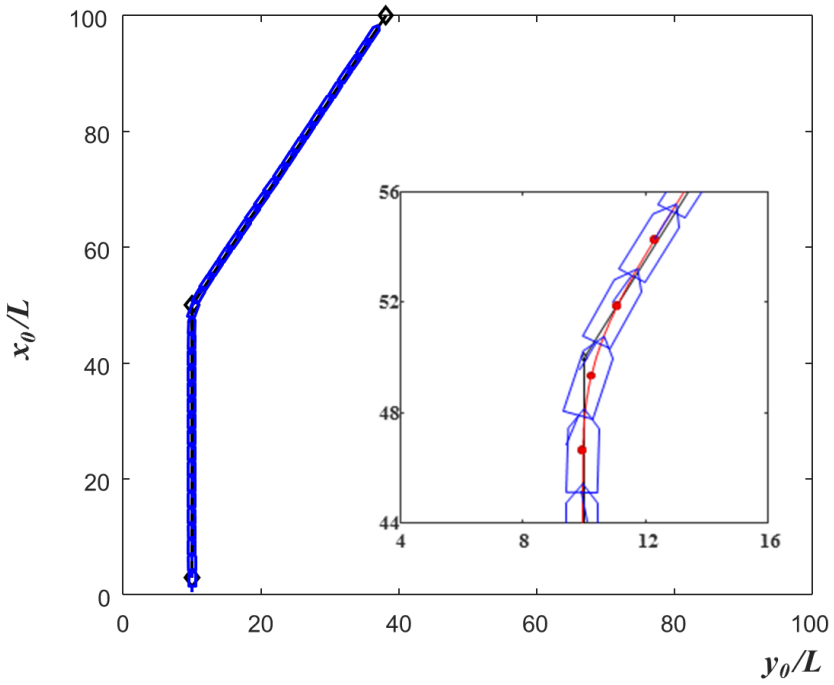
The initial states of a ship in the Cases 3.6, 3.7, and 3.8 are set as $\delta = 0^{\circ}$, $\psi = 0^{\circ}$, $r' = 0^{\circ}$, and $\beta = 0^{\circ}$. Fig. 3.16 shows the results of the Case 3.6 with low speed $U = 0.32$ m/s. Figures (a) and (b) present ship trajectory and the time history of rudder angle gained by the proposed algorithm. The shape of the ship on its trajectory is surfaced every 25 seconds and waypoints are indicated as diamond patterns. In order to figure out motions of the ship in the vicinity of a waypoint at where veering action is conducted, enlarged figure around the waypoint is presented. It is observed from Fig. 3.16(b) that rudder angle δ to veer the heading angle has been reached to 14.68° as the maximum angle, and the rudder was manipulated toward the opposite direction to inhibit the ship from deviating from the new path due to the influence of the inertial motion. It is also observed that heading angle ψ became 30° which is equal to target course change angle $\theta = 30^{\circ}$ after manipulating a rudder as shown in Fig. 3.16(c). The relationship between heading error ψ_{e} and cross track error d'_{e} is demonstrated in Fig. 3.16(e). When the ship enters into the circle of acceptance area, the absolute values of ψ_{e} and d'_e begin to increase unconditionally due to the modification of "to-waypoint" position. Therefore, it should be noted that actual overshoot values of ψ_{e} and d'_{e} appear around 450 seconds when the overshoot of ψ is observed. In this case, the values of $|\psi_{\mathrm{e}}|$ and $|d'_{\mathrm{e}}|$ were negligibly small and the ship could follow the assigned desired track precisely.

In Fig. 3.17, the ship traveled at normal speed $U = 0.50$ m/s as defined in the Case 3.7. In comparison with the results of the Case 3.6, the maximum angle decreased to $\delta = 9.23^{\circ}$ despite having the same course change angle $\theta = 30^{\circ}$. However, steerage used in port side caused the overshoot of heading angle ψ, then additional steerage to starboard side to restrain the overshoot is found after about 300 seconds as shown in Fig. 3.17(b). The actual maximum errors appeared in Fig. 3.17(e) are $|\psi_{\mathrm{e}}| = 2.47^{\circ}$ and $|d'_{\mathrm{e}}| = 0.28$. Even if we compare the enlarged figure on Fig. 3.17(a), it can be confirmed that the positions of the vessel represented by the red dots are further away from the planned path than those in the previous case.

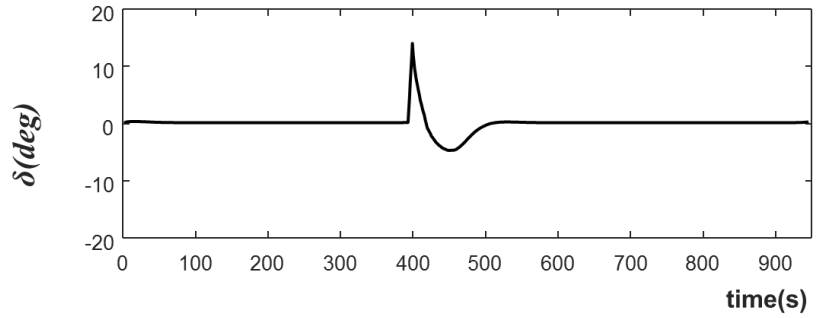
Fig. 3.18 shows the operation of a ship in the Case 3.8 with the fastest speed $U = 0.68$ m/s that is similar with the design speed 15.5 knots. Since the ship speed U in this case is the fastest among the three cases, the smallest rudder angle $\delta = 8.80^{\circ}$ was realized as shown in Fig. 3.18(b). The enlarged figure in Fig. 3.18(a) describes that the ship deviated from the desired track immediately after using a rudder to change her course. At that time, the values

of actual maximum errors are $|\psi_e|$ = 3.21˚ and $|d_e'|$ = 0.40 as shown in Fig. 3.18(d). Namely, it was proven that the ship operated at high speed uses the less rudder angle to modify the passing track. Furthermore, based on the aforementioned results of the three cases, it is confirmed that rudder was executed differently according to ship velocity.

Finally, simulation based on the fourth case was carried out on S-shape desired track consisting of diverse course change angle θ in the range from -47˚ to 47˚. A ship can be affected by wind and current coming from various directions as sailing on the track. Fig. 3.19 shows the simulated responses of the ship in the Case 3.9. The initial states of parameters are U = 0.50 m/s, δ = 0˚, ψ = 180˚, r = 0˚, and β = 0˚. The maximum $|\delta|$ used to change her course is 11.21˚ as represented in Fig. 3.19(b). It seems that large errors are observed in Fig. 3.19(e), but the value of $d_e{}'$ calculated by Eq. (3.1) is temporarily increased due to the transition of the target waypoint position. Thus, the actual maximum errors are $|\psi_e|$ = 4.30˚ and $|d_e'|$ = 0.37 which are appeared at the same time when the overshoot of ψ occurs. All of simulation results of the four cases demonstrate that the ship equipped with the proposed algorithm can trace the intended track very closely. $|d_e'|$ as an actual cross track error does not exceed one ship length since the rudder was manipulated at optimal timing obtained from the waypoint switching system depending on course change angle θ.

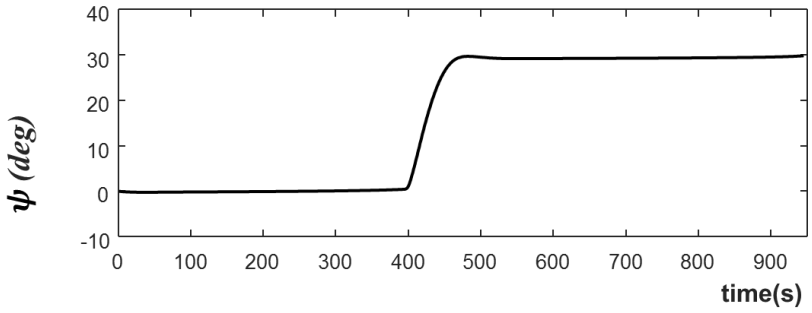


(a) Ship trajectory

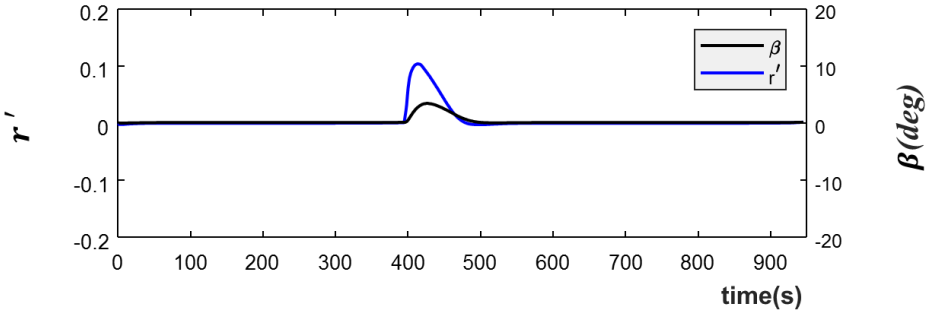


(b) Time history of rudder angle

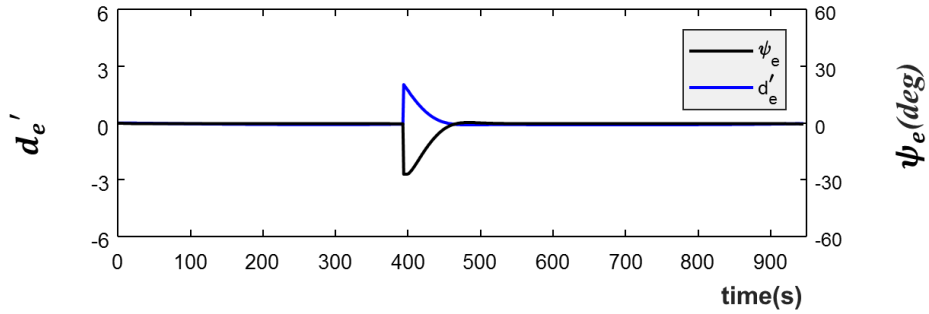
Fig. 3.16 Simulation results of the Case 3.6



(c) Time history of heading angle

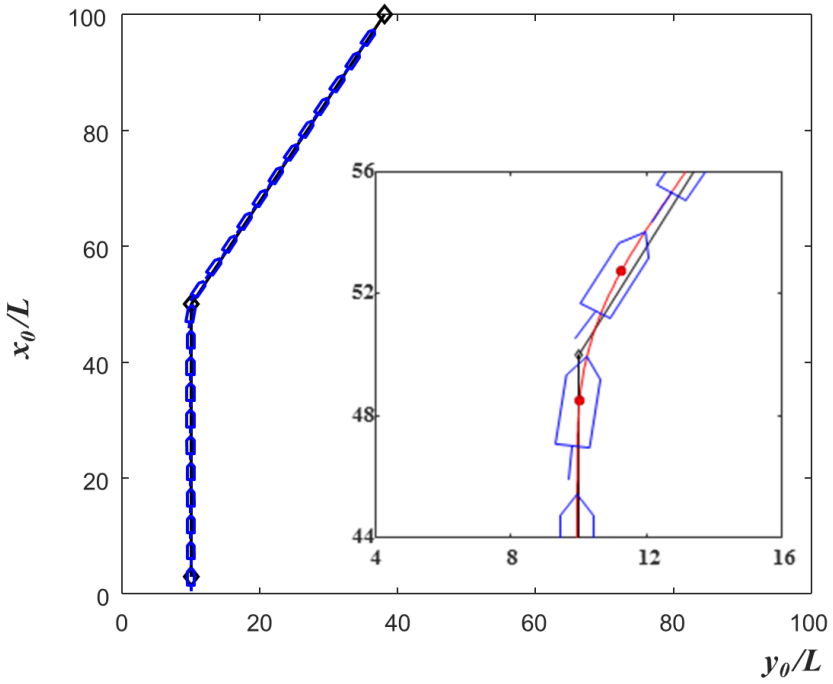


(d) Time history of rudder angle and non-dimensional yaw rate

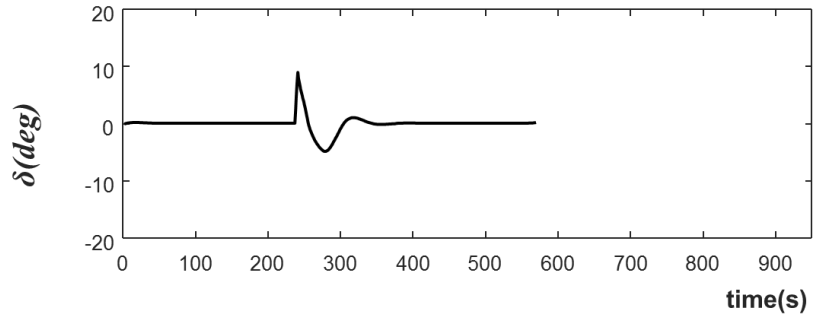


(e) Time history of heading error and non-dimensional cross track error

Fig. 3.16 Simulation results of the Case 3.6

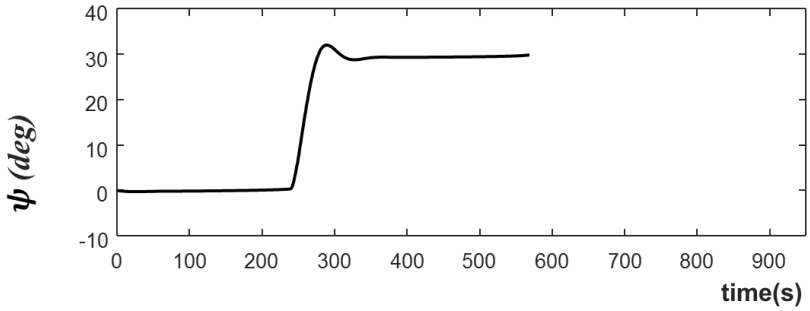


(a) Ship trajectory

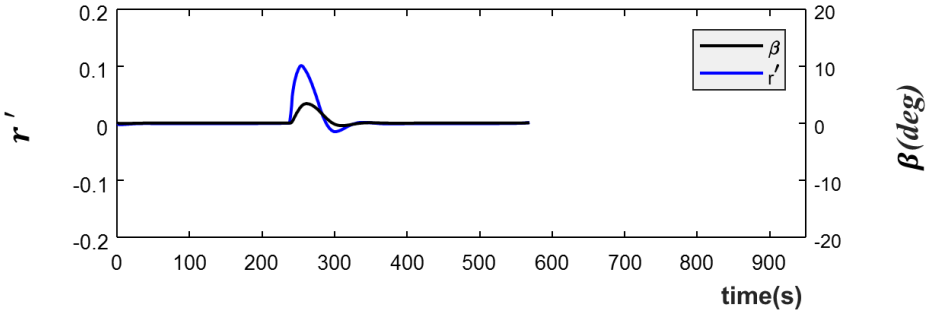


(b) Time history of rudder angle

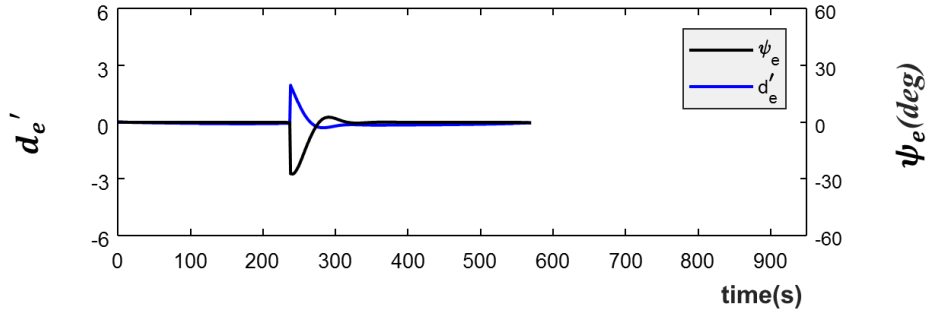
Fig. 3.17 Simulation results of the Case 3.7



(c) Time history of heading angle

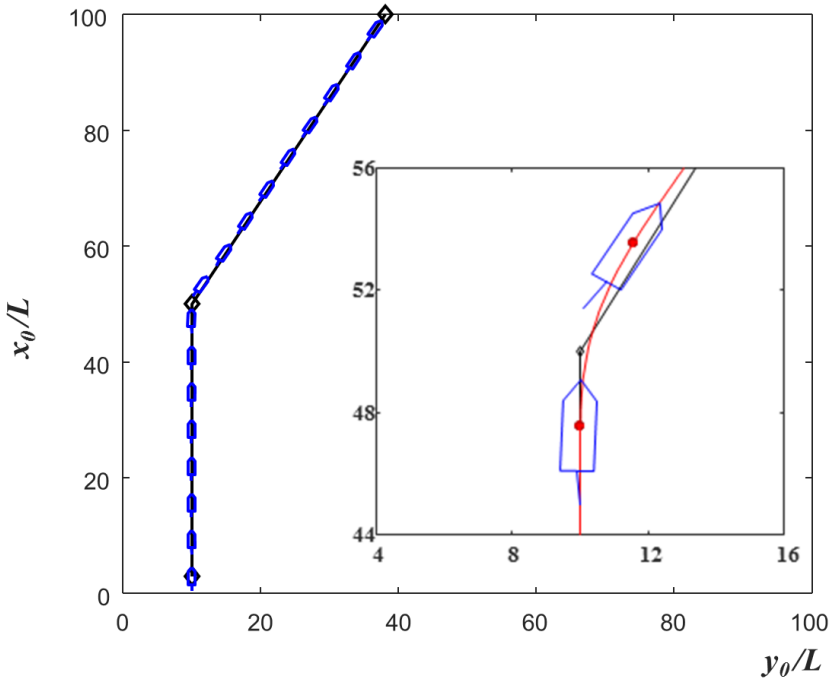


(d) Time history of rudder angle and non-dimensional yaw rate

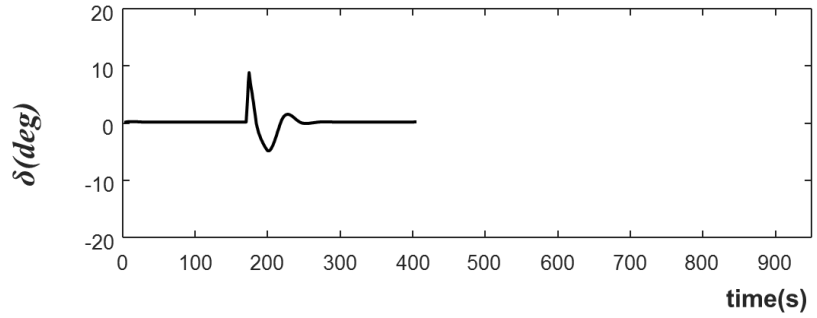


(e) Time history of heading error and non-dimensional cross track error

Fig. 3.17 Simulation results of the Case 3.7

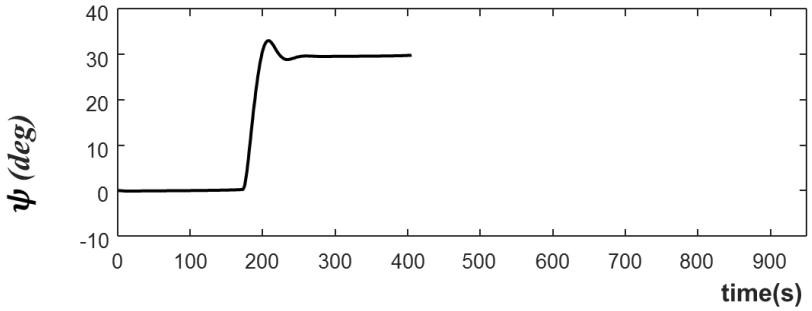


(a) Ship trajectory

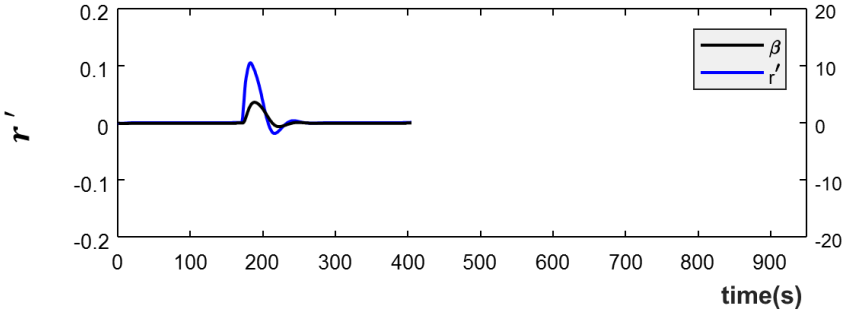


(b) Time history of rudder angle

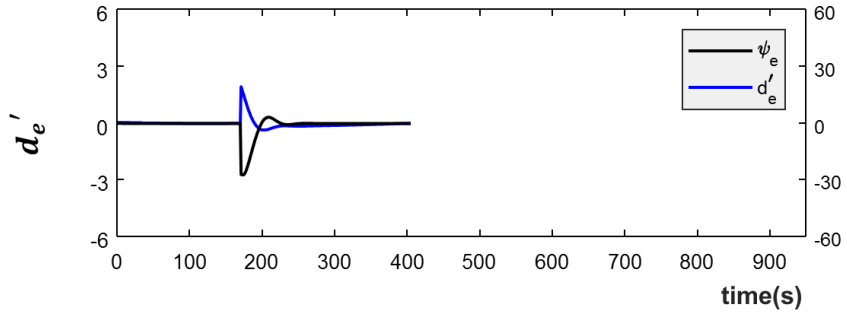
Fig. 3.18 Simulation results of the Case 3.8



(c) Time history of heading angle

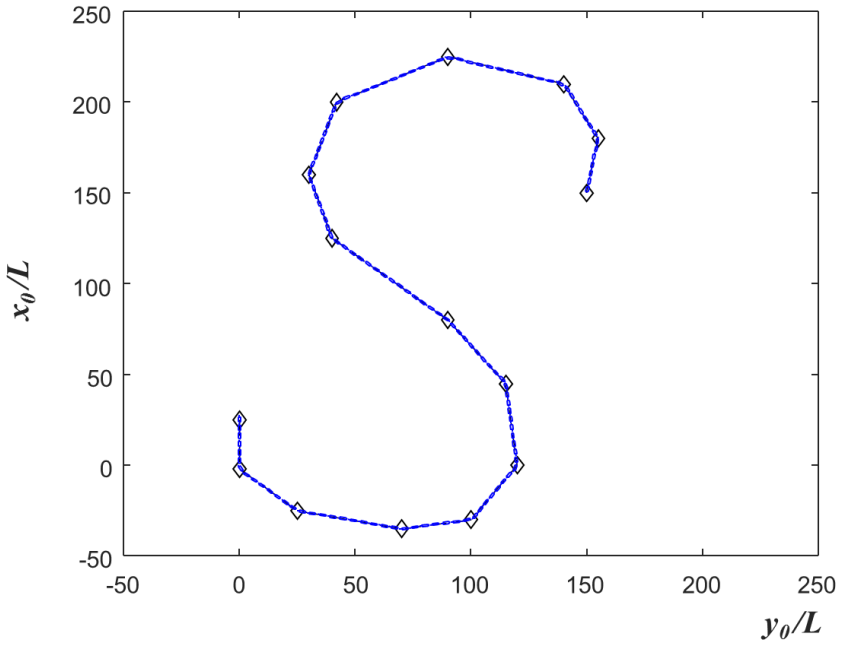


(d) Time history of rudder angle and non-dimensional yaw rate

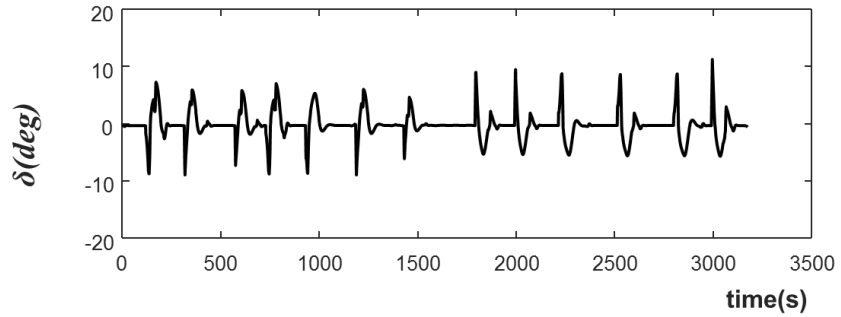


(e) Time history of heading error and non-dimensional cross track error

Fig. 3.18 Simulation results of the Case 3.8

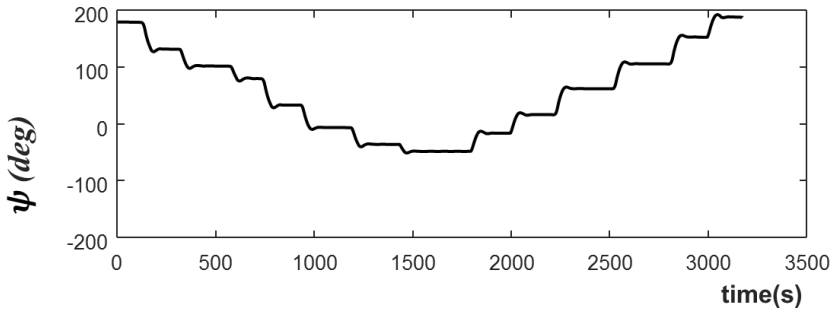


(a) Ship trajectory

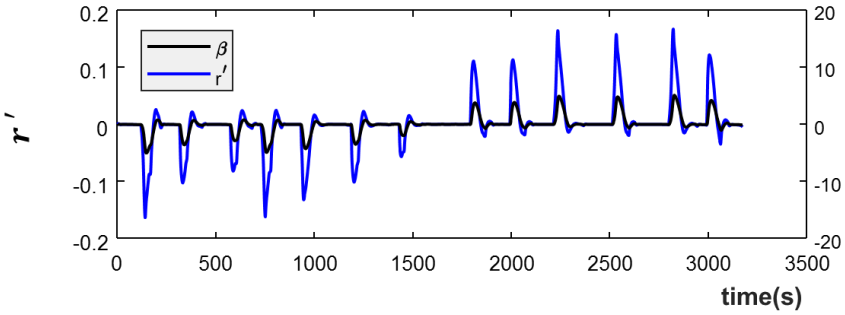


(b) Time history of rudder angle

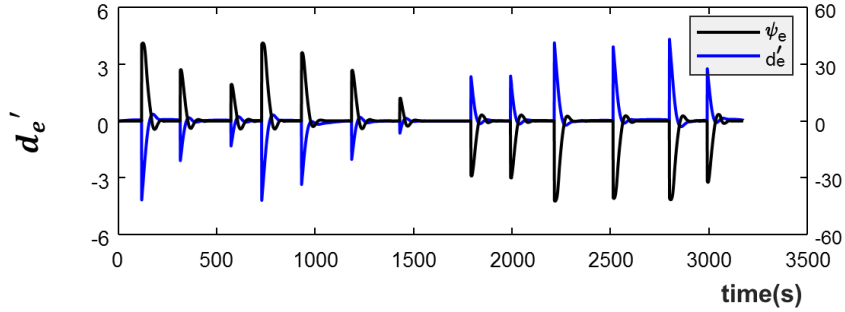
Fig. 3.19 Simulation results of the Case 3.9



(c) Time history of heading angle



(d) Time history of drift angle and non-dimensional yaw rate



(e) Time history of heading error and non-dimensional cross track error

Fig. 3.19 Simulation results of the Case 3.9

3.7 Conclusions

In this chapter, automatic path following algorithm applying fuzzy inference is proposed. Conclusions drawn from this work can be summarized as follows:

- The path following algorithm consists of two components, waypoints guidance system and rudder control system, were developed. In the waypoints guidance system, a desired track which a ship should follow was built by feeding waypoints positon data. Optimal timing to use rudder was derived from the waypoint switching system according to a course change angle owned by a target waypoint.

- Two kinds of path following algorithm were developed. The algorithm is distinguished depending on the performance of rudder control system which provides suitable rudder angle to change ship's course. They were named as the basic path following algorithm and the improved path following algorithm respectively.

- The basic path following algorithm has rudder control system taking account of three parameters: heading error, a cross track error, and a yaw rate. On the other hand, rudder control system implemented in the improved path following algorithm employed four parameters by adding ship speed to the existing three parameters.

- Numerical simulations were carried out assuming external disturbance such as wind and current in virtual situation to verify the effectiveness of the developed algorithm. The simulation results show that a ship equipped with the proposed systems can arrive at her destination with little overshoots of a heading error and a cross track error.

The performance of the proposed algorithm was proved in the situation assuming external disturbances. Further investigation will be needed for adverse condition as well as in various environmental conditions considering shallow water effect, the influence of wave, and so on.

Chapter 4. Application of Developed Automatic Path Following Algorithm in Realistic Situations

4.1 Introduction

Most of the simulations of path following systems presented in the past researches were carried out in virtual situations that ships moved on a geometrical track consisted of waypoints under arbitrary external disturbances such as wind, current, and wave. The arbitrary external disturbances have been often assumed to have constant direction and speed over the entire simulated area. As far as the actual disturbances concerned, they have irregular characteristics and let ships swerve from a desired course. Simulations conducted under the assumed environmental conditions are expected to have limitation to assess the effectiveness of the developed algorithm of path following. Hence, using the automatic path following algorithms suggested in Chapter 3, numerical simulations will be performed to verify the capability in realistic environmental situation in this chapter. From the above-mentioned point of view, route plans that were actually used in operations to create desired tracks are utilized as the scenarios of the simulations in this study. Irregular environmental disturbances reproduced by using actual data of wind and current at sea are applied depending on ship's position in real time.

4.2 Realistic environmental condition

Environment data related to wind and current on the specific date (7th August, 2017) were acquired from National Oceanic Earth and Atmospheric Administration (NOAA) [61] and Japan Meteorological Agency (JMA). Raw data of wind were measured every three hours by NOAA and current data were estimated every 15 minutes by JMA. To create more detailed conditions of the external disturbances, the obtained data were interpolated based on time, latitude, and longitude respectively using MATLAB. As a result, more detailed and closely spaced data of current velocity V_C, wind speed V_W, current direction α, and wind direction υ were generated. It makes it possible to simulate the motion of a model ship affected by the disturbances considering time and her position. Fig. 4.1 and Fig. 4.2 demonstrate the views of

distribution of wind and current every 30 minutes from 21:00 to 23:30 on 7th August, 2017, respectively. The blue arrow bars indicate speed vectors of wind and current.

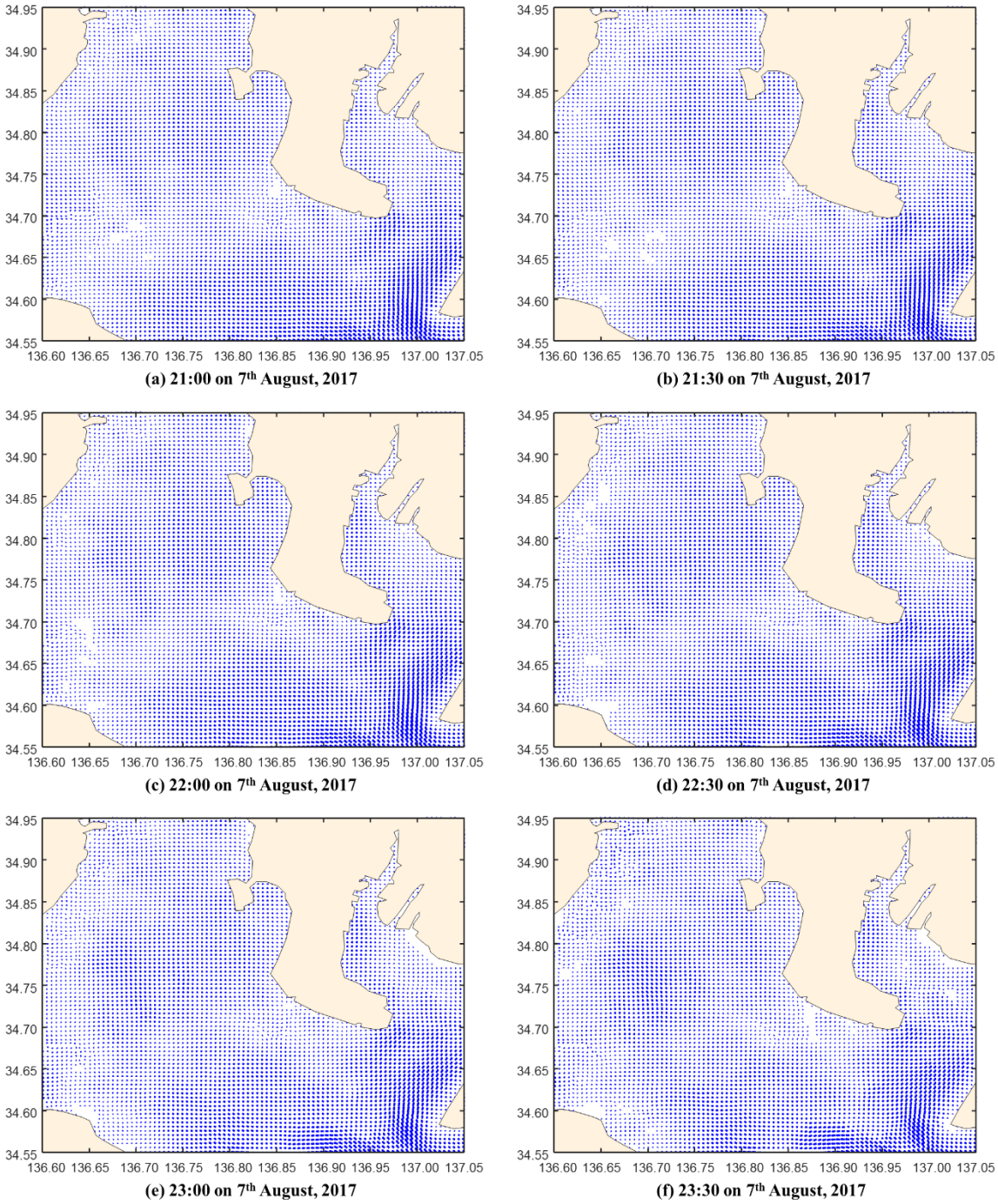


Fig. 4.1 Distribution of curernt speed and direction

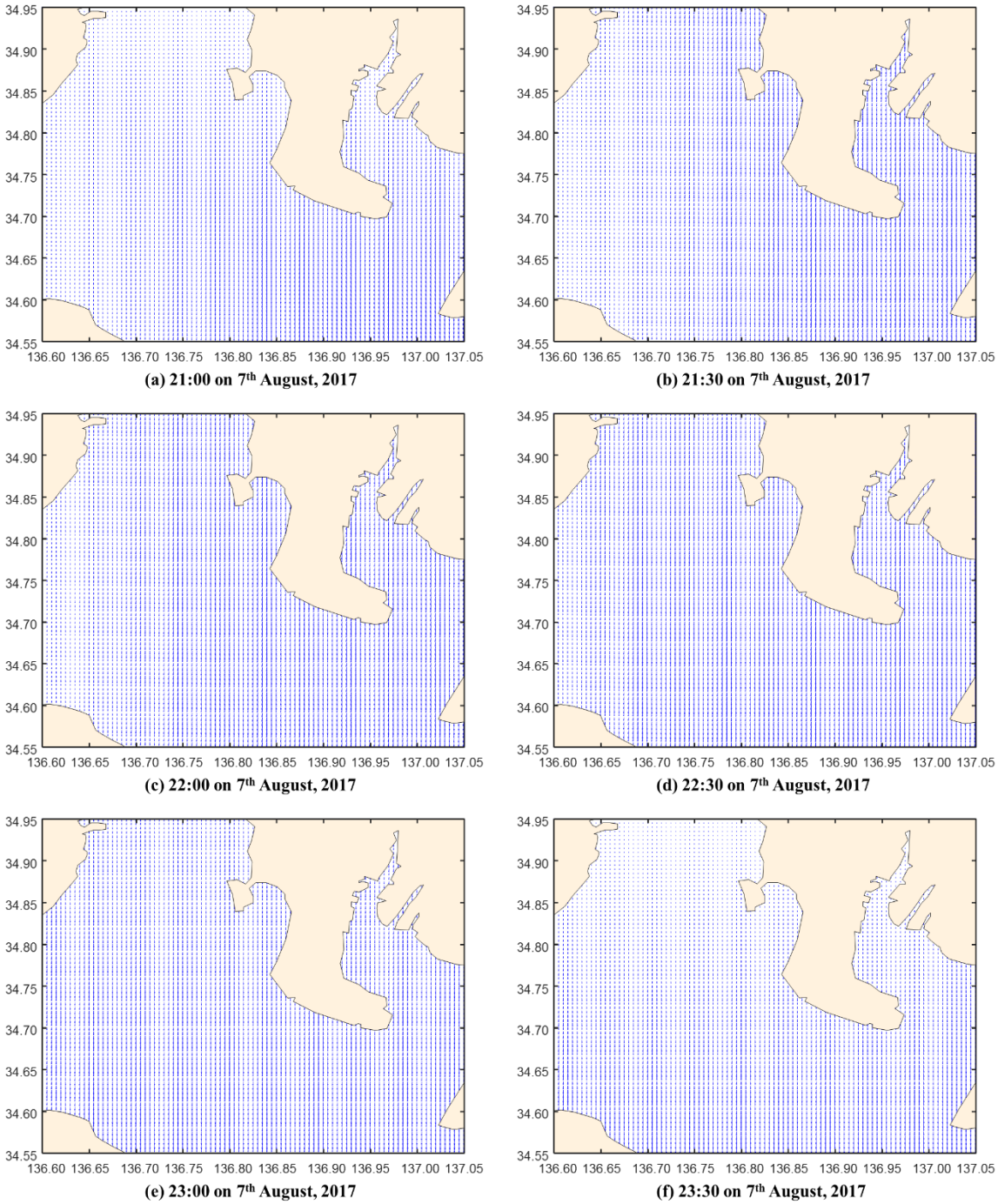


Fig. 4.2 Distribution of wind speed and direction

4.3 Numerical simulations with basic path following algorithm

4.3.1 Simulation conditions

The subject ship was selected as KVLCC2, which is the same with the ship used in the previous chapter. However, the simulations were carried out with actual-sized ship instead of scaled model ship so as to verify the performance in realistic environment. The principle particulars of the full-scaled ship are shown in Table 2.2 in Chapter 2.

Ships are generally affected by external disturbance such as wind and current during voyage. In order to reproduce the realistic navigation situation, a route plan made by operators and actual disturbance data in the Ise Bay, Nagoya, Japan were applied to set simulation scenarios. The position data of the waypoints are named as P1, P2, P3, and P4 as presented in Table 4.1 and the desired track consisted of these four waypoints is shown in Fig. 4.1 and Fig. 4.2. However, port domain is not included in the scenarios because following the track at the inside of ports should be the function of auto berthing systems which is not considered in this research. The simulation scenarios cover situations that a ship enters to or departs from the port at two kinds of speed: normal speed 10 knots and high speed 15 knots. In other words, totally four scenarios are able to be summed up as shown in Table 4.2. Departure situation means that the ship heads to the P4 from the P1 and the ship moves to the opposite direction at arrival situation, namely from the P4 to the P1.

Table 4.1 Waypoint position data for simulations

Waypoints	Latitude	Longitude
P1	34.93˚	136.76˚
P2	34.80˚	136.76˚
P3	34.97˚	136.84˚
P4	34.59˚	136.97˚

Table 4.2 Definition of scenarios for simulations

Scenario	Situation	Speed
4.1	Departure	10 knots
4.2	Arrival	10 knots
4.3	Departure	15 knots
4.4	Arrival	15 knots

The ship applied to the simulations is assumed to start her navigation at 21 o'clock on 7th August, 2017. Table 4.3 shows the initial conditions of external disturbance that the ship receives at that moment. The external disturbances in the departure situation (the Scenarios 4.1 and 4.3) have smaller effect comparing with those in the arrival situation (the Scenarios 4.3 and 4.4) at the beginning of the simulations.

Table 4.3 Definition of initial environmental condition for simulations

Scenario	Wind		Current	
	V_W	ν	V_C	α
4.1	1.58 m/s	0.71 ˚	0.11 m/s	-169.0 ˚
4.2	9.03 m/s	4.07 ˚	0.44 m/s	178.4 ˚
4.3	1.58 m/s	0.71 ˚	0.11 m/s	-169.0 ˚
4.4	9.03 m/s	4.07 ˚	0.44 m/s	178.4 ˚

4.3.2 Simulation results

Simulations for the four scenarios were carried out under the realistic environmental conditions as described above and they were updated in real time. The black line connected with four diamond-shaped waypoints is defined as the desired track. A ship starts navigation at the first waypoint and stops at the moment when she reaches the last waypoint. During the voyage, the target waypoint is switched twice near the waypoints P2 and P3 and course change angles around the waypoints are θ = 26.46˚ and θ = 26.97˚ respectively.

Fig. 4.3 shows the simulation results for the Scenario 4.1 that a ship departs from the port at 10 knots. Figs. 4.3(a)-(b) present ship's trajectory and the time history of rudder angle obtained by the proposed path following algorithm. The shape of ship on the trajectory appears every 5 minutes. A rudder was manipulated in order to change course before the ship reaches the waypoints P2 and P3. Thereafter, the rudder was used in the opposite direction to prevent the ship from passing through new track due to the influence of inertial motion. All of used rudder angle are less than 10°. Information regarding to wind and current as external disturbances is observed in Figs. 4.3(d)-(e). V_W and ν gradually increased with progress of the time. V_C changes between 0.05 m/s and 0.31 m/s. After the ship changed her course near the waypoint P3, we can know that the lateral force caused by wind coming from about $\nu = 5°$ makes cross track error d'_e continuously increase as presented in Fig. 4.3(g). The relationship between heading error ψ_e and cross track error d'_e is also shown in Fig. 4.3(g). During the voyage, observed maximum values of actual overshoot were $|\psi_e| = 5.13°$ and $|d'_e| = 0.23$.

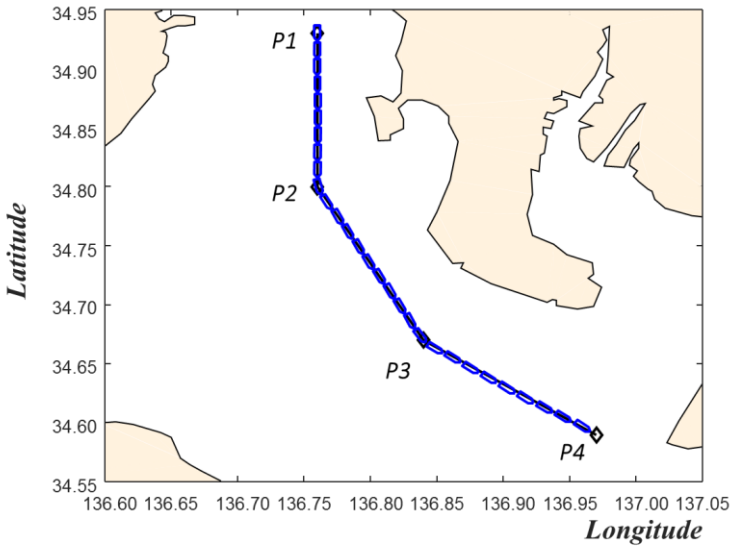
Simulation results for the Scenario 4.2 were illustrated in Fig. 4.4 when a ship enters into the port at 10 knots. Even though the ship traveled the same distance at the same speed with the previous simulation, she reached the final waypoint faster than that due to the influence of external disturbances. That is because V_W gradually decreased and the ship is in fair current condition as shown in Fig. 4.4(d)-(e). According to Fig. 4.4(b), used rudder angles are less than 10°. The maximum values $|\psi_e| = 6.38°$ and $|d'_e| = 0.36$ appear immediately after the veering near the waypoint P3 as shown in Fig. 4.4(g).

Fig. 4.5 describes simulation results for the Scenario 4.3 performed when a ship departs from the port with 15 knots which is faster than that of the Scenario 4.1. When the ship uses rudder to change her course, yaw rate becomes higher as her speed becomes faster. As shown in Fig. 4.5(b), larger rudder was executed in opposite direction so as to restrain the yaw rate which is increased in comparison with the Scenario 4.1. The magnitude of errors appear as $|\psi_e| = 6.78°$ and $|d'_e| = 0.37$ in Fig. 4.5(g).

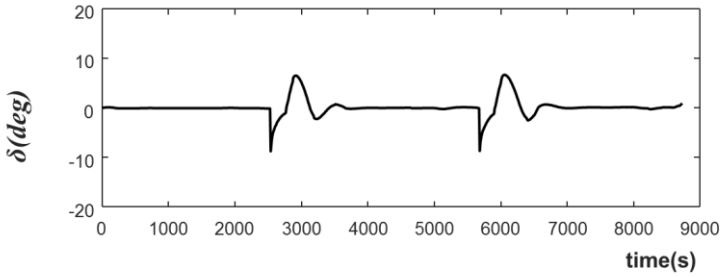
Fig. 4.6 shows the simulation results for the Scenario 4.4 carried out when a ship enters into the port at the speed of 15 knots. The ship in this simulation arrived at her destination earliest among the four simulation cases. However, the rudder angle in Fig. 4.6(b) oscillates more as seen from comparison with the results of other simulations. It means that it takes more time to attain the ψ_d directly after rudder was manipulated for changing course. Nevertheless, used

rudder angles did not exceed 10°. As the maximum errors, $|\psi_e| = 8.27°$ and $|d'_e| = 0.61$ are obtained from Fig. 4.6(g).

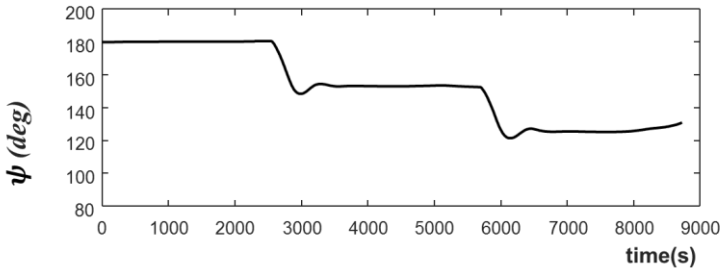
All of simulation results of the four scenarios show that a ship can closely follow the planed route using realistic rudder angle. Actual cross track error does not exceed one ship length because rudder was used at appropriate timing depending on course change angle. When a ship turns in the vicinity of a waypoint to approach the new course, rudder is used to change her heading toward desired heading angle and yaw rate increased. Then, the rudder is executed in opposite direction so as to keep the heading by making yaw rate zero. Given that velocity of a ship influences on the magnitude of yaw rate, larger opposite rudder angle is generated in order to control a yaw rate if ships move with high speed.



(a) Ship trajectory

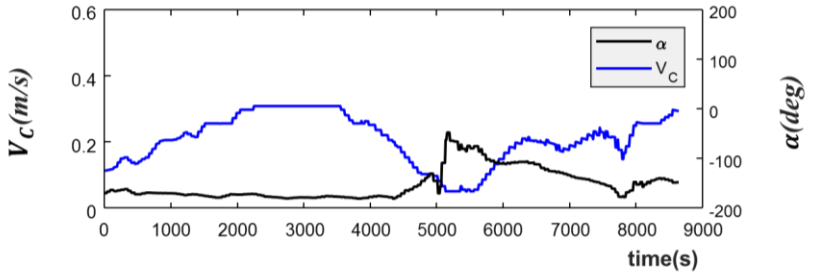


(b) Time history of rudder angle

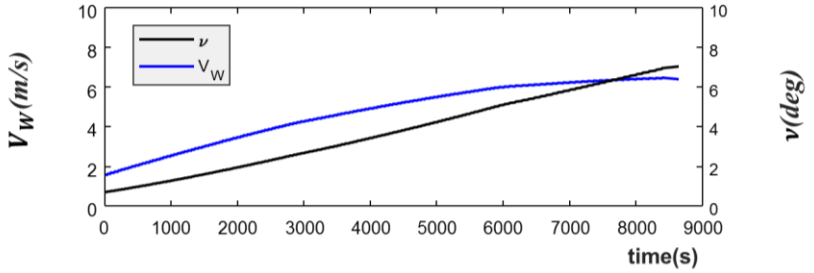


(c) Time history of heading angle

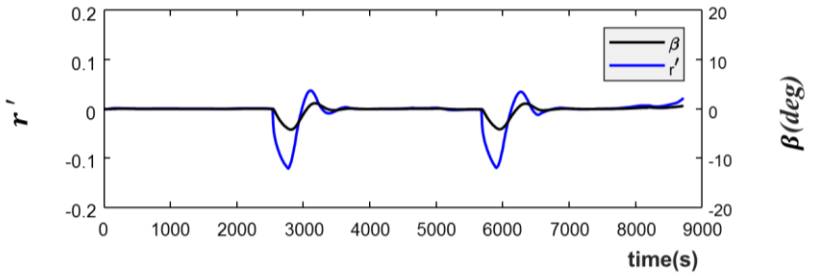
Fig. 4.3 Simulation results for the Scenario 4.1



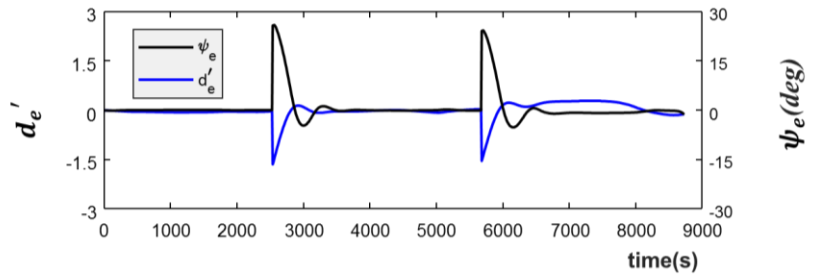
(d) Time history of speed and direction for current



(e) Time history of speed and direction for wind

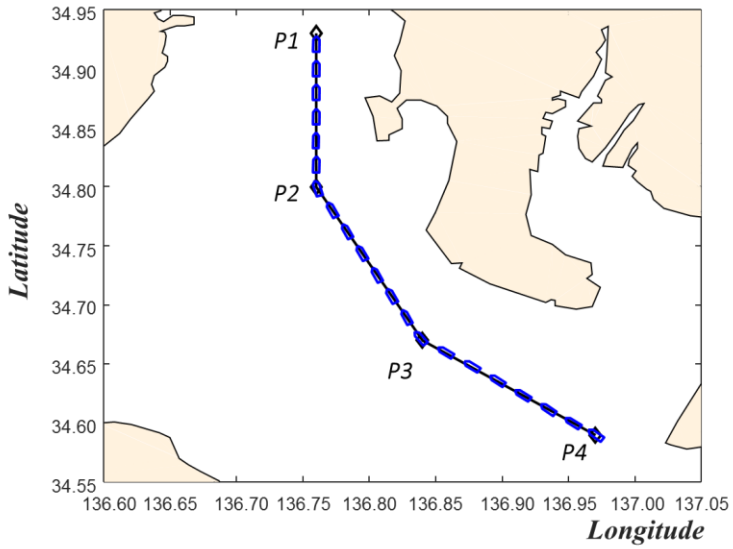


(f) Time history of drift angle and non-dimensional yaw rate

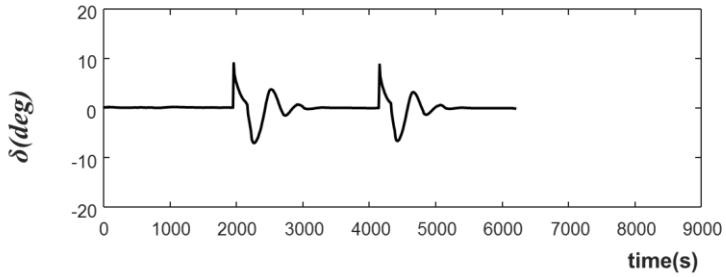


(g) Time history of heading error and non-dimensional cross track error

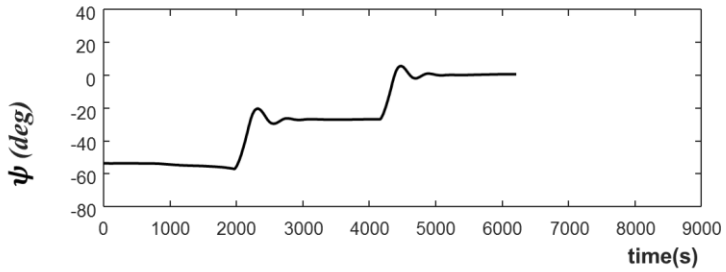
Fig. 4.3 Simulation results for the Scenario 4.1



(a) Ship trajectory

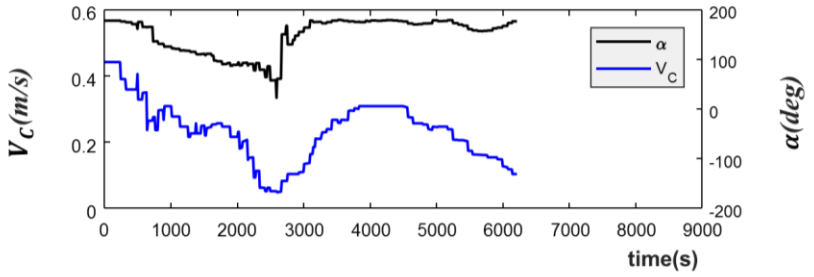


(b) Time history of rudder angle

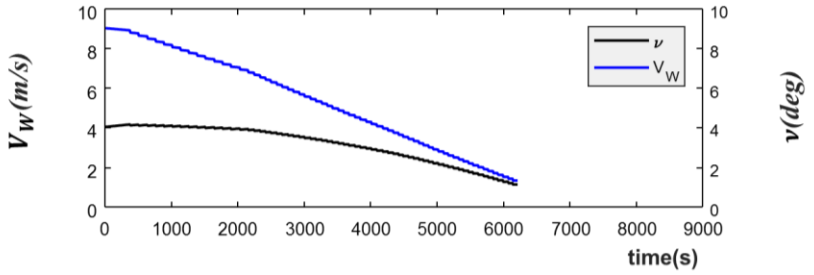


(c) Time history of heading angle

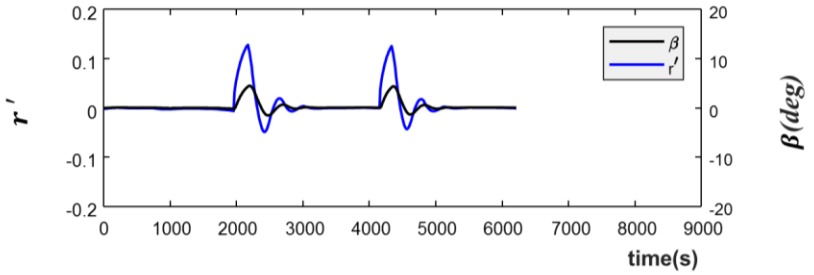
Fig. 4.4 Simulation results for the Scenario 4.2



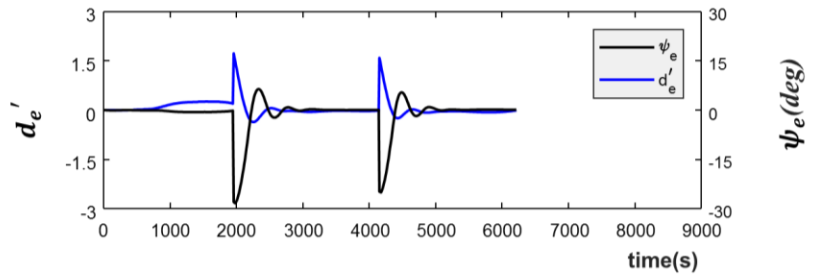
(d) Time history of speed and direction for current



(e) Time history of speed and direction for wind

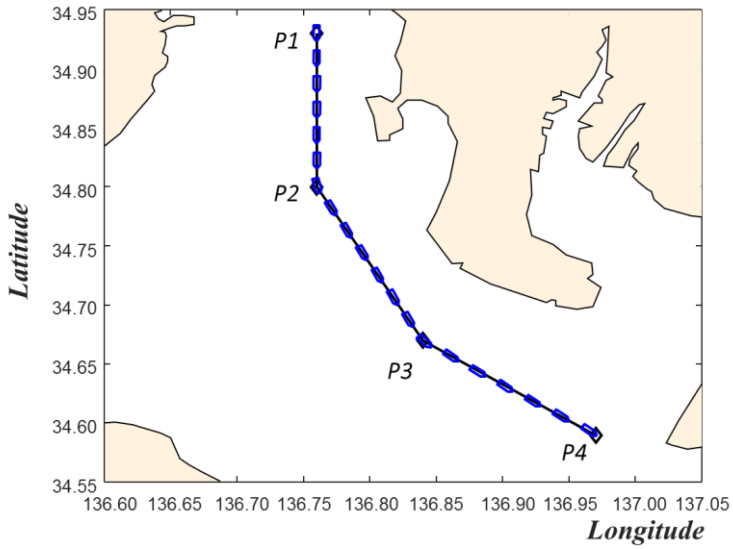


(f) Time history of drift angle and non-dimensional yaw rate

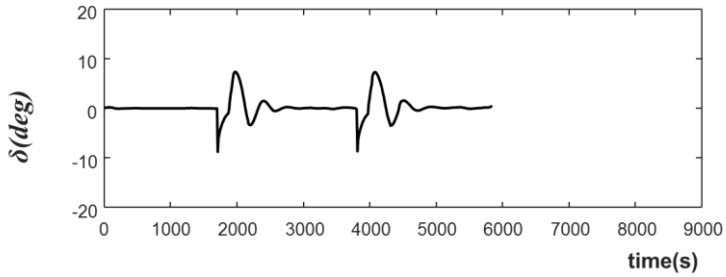


(g) Time history of heading error and non-dimensional cross track error

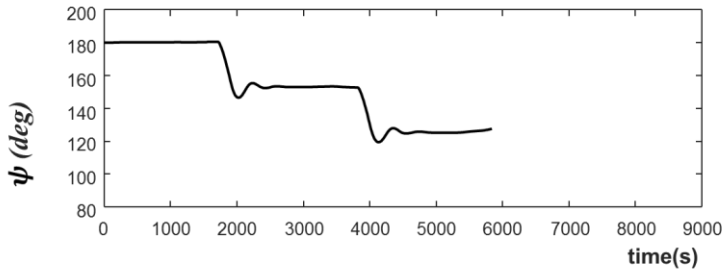
Fig. 4.4 Simulation results for the Scenario 4.2



(a) Ship trajectory

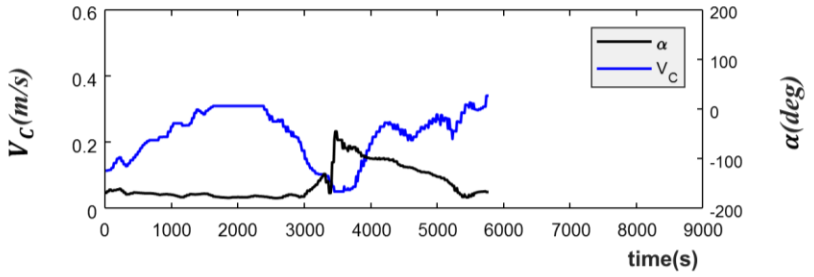


(b) Time history of rudder angle

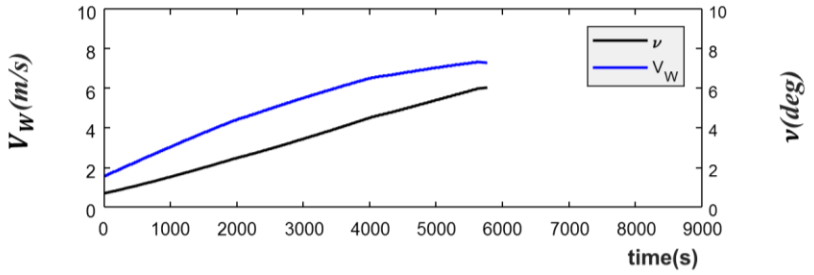


(c) Time history of heading angle

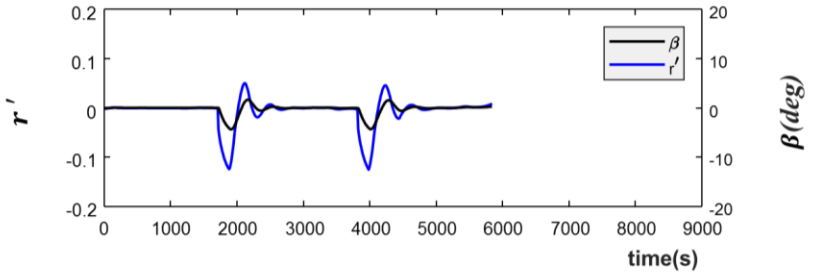
Fig. 4.5 Simulation results for the Scenario 4.3



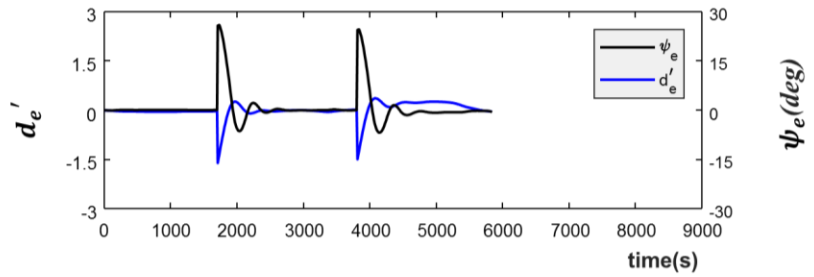
(d) Time history of speed and direction for current



(e) Time history of speed and direction for wind

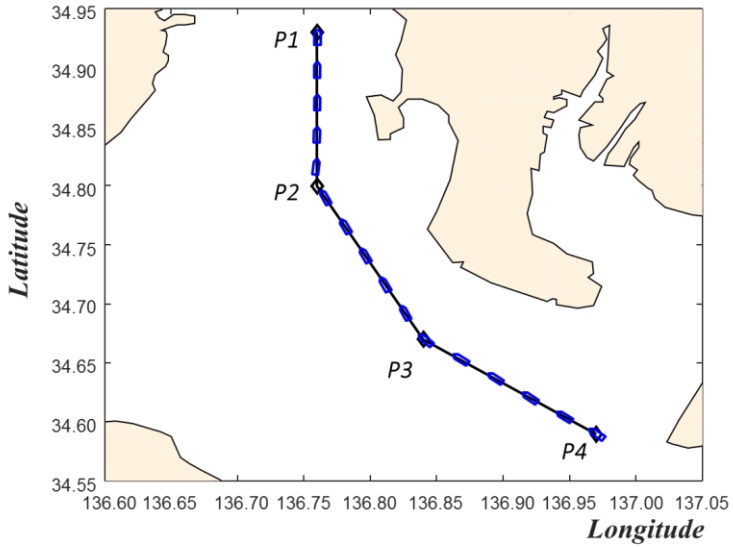


(f) Time history of drift angle and non-dimensional yaw rate

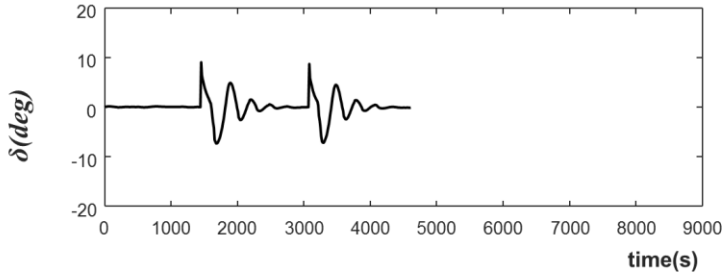


(g) Time history of heading error and non-dimensional cross track error

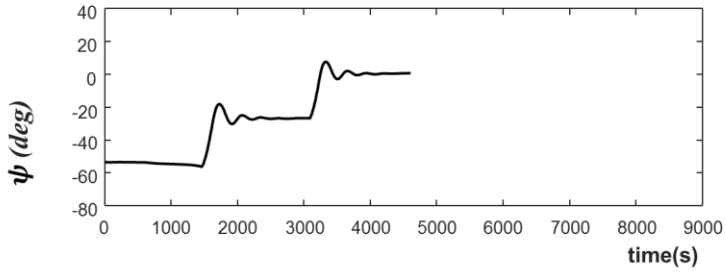
Fig. 4.5 Simulation results for the Scenario 4.3



(a) Ship trajectory

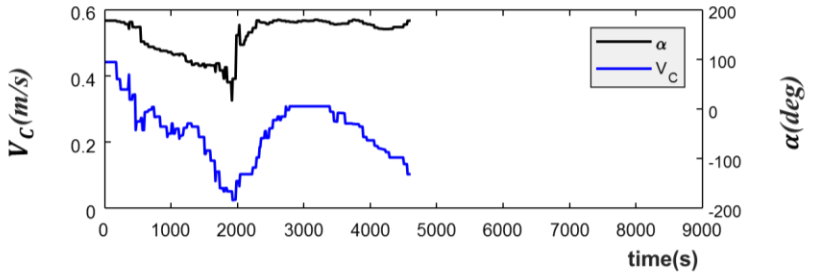


(b) Time history of rudder angle

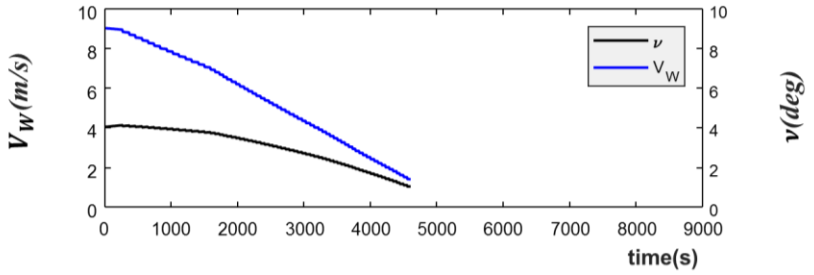


(c) Time history of heading angle

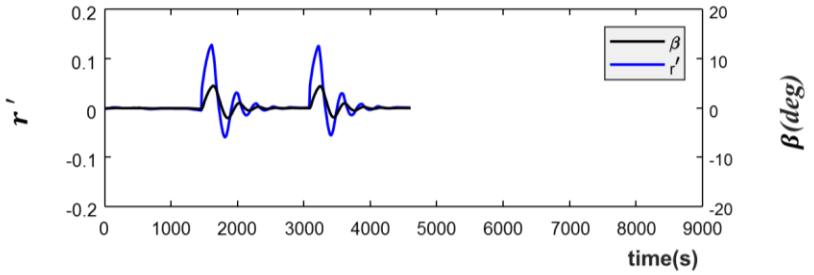
Fig. 4.6 Simulation results for the Scenario 4.4



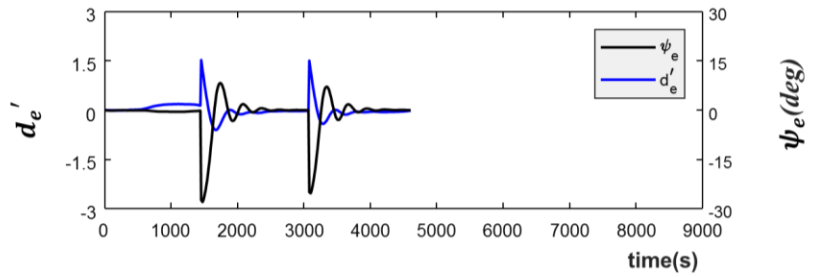
(d) Time history of speed and direction for current



(e) Time history of speed and direction for wind



(f) Time history of drift angle and non-dimensional yaw rate



(g) Time history of heading error and non-dimensional cross track error

Fig. 4.6 Simulation results for the Scenario 4.4

4.4 Numerical simulations of the improved path following algorithm

4.4.1 Simulation conditions

The subject ship for simulations is same as the ship of previous section, full scaled KVLCC2, and the principle particulars of the ship is indicated in Table 2.2 in Chapter 2. The numerical simulations were carried out to verify the improved path following algorithm introduced in Chapter 3 under the realistic situation. The improved path following algorithm takes the ship speed into account additionally, unlike the basic path following algorithm validated in the previous section. Therefore, scenarios with different speeds were established as shown in Table 4.4. The speeds are distinguished as low speed 7 knots, normal speed 10 knots, and high speed 15 knots. To compare the two algorithms, the Scenario 4.4 of Table 4.2, one of the scenarios implemented in the previous section, was selected. In order to confirm the difference of performance according to the speed, normal speed and high speed were set to the same value with the speed condition of the Scenario 4.4. As shown in Table 4.3, The ship in the Scenario 4.4 is most influenced by external forces such as wind and current at the beginning of her navigation. To be short, the ship moved to the waypoint P1 from the waypoint P4 as the arrival situation and is affected by disturbance forces V_W = 9.03 m/s, ν = 4.07˚, V_C = 0.44 m/s, α = 178.4˚ at the beginning of the simulations. The waypoints P1 and P4 are illusterated in Fig. 4.1.

Table 4.4 Definition of scenarios for simulations

Scenario	Speed
4.5	7 knots
4.6	10 knots
4.7	15 knots

4.4.2 Simulation results

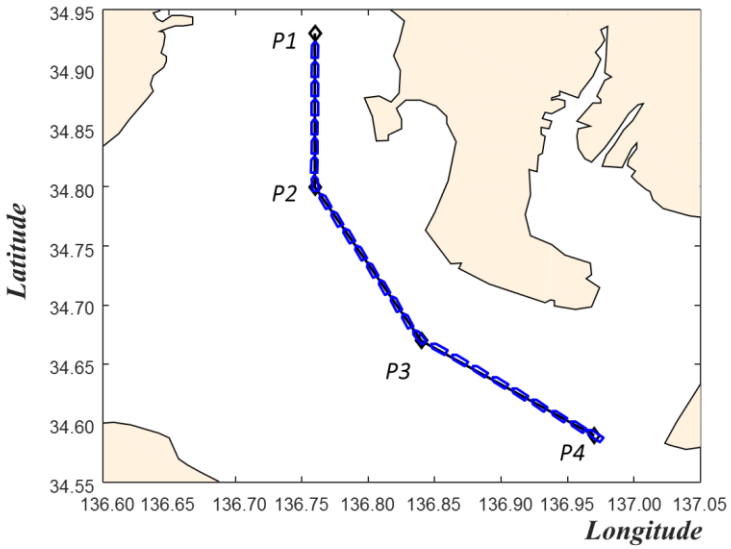
Simulations were carried out under the realistic environmental conditions as described above and they were updated in real time. A ship starts navigation at the first waypoint P4 and stops at the moment when she reaches the last waypoint P1. During the ship is travelling, course angles are modified twice around the waypoints P2 and P3. Namely, the ship employed rudder angle twice for entering into the desired course lines consisting of the waypoints P3 and P4 and the waypoints P1 and P2.

The simulation results of the Scenario 4.5 in which the ship departs from the port at low speed 7 knots are demonstrated in Fig. 4.7. The shapes of the ship are drawn every 5 minutes on the trajectory as shown in Fig. 4.7(a). Fig. 4.7(b) and Fig. 4.7(c) present the time histories of rudder angle and heading angle respectively. It can be observed in Fig. 4.7(b) that rudder angle is adjusted to change the direction of the ship before she reaches the positions of the waypoints P3 and P2. The largest rudder angle manipulated until reaching the final waypoint is 11.7˚. The ship started her navigation with the initial heading angle ψ = -53.32˚ and the heading angle is shifted to -26.42˚ and 0.04˚ as much as course change angle around the waypoints P2 and P3 which are θ = 26.46˚ and θ = 26.97˚. Fig. 4.7(d) and Fig. 4.7(e) show external environmental conditions of current and wind varied according to the time and the ship's position. The black lines present current angles α and wind angle υ and the blue lines express current velocity V_C and wind speed V_W. V_C changed in the range between 0.05 m/s and 0.45 m/s. On the other hand, V_W and ν gradually increased as time goes by. The overshoots ψ_e and d'_e are presented in Fig. 4.7(g). The obtained maximum actual values in Fig. 4.7(g) revealed as $|\psi_e|$ = 1.87˚ and $|d'_e|$ = 0.20.

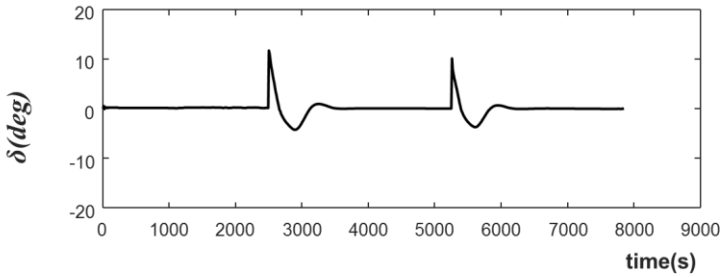
Fig. 4.8 shows simulation results for the Scenario 4.6 with the normal speed 10 knots. The ship traveled on the same route with the previous scenario. In order to compare the difference between the path following algorithm considering speed and not considering, we took note of results shown in Fig. 4.8 and Fig. 4.4. In the rudder motion shown in Fig. 4.8(b) and Fig. 4.4(b), the most notable point is motion of the rudder that used in an attempt to suppress inertial motion. Despite the ship sailed on the same track at the same speed, the rudder of the ship equipped with the algorithm that does not consider the speed shows more oscillation. The maximum overshoots were observed as $|\psi_e|$ = 2.78˚ and $|d'_e|$ = 0.29 immediately after changing orientation around the waypoint P3 according to Fig. 4.8(g).

Simulations displayed in Fig. 4.9 were carried out at high speed 15 knots which is faster than that in aforementioned simulations. In Fig. 4.9(b), the rudder motion appeared to oscillate more than those in Fig. 4.7(b) and Fig. 4.8(b) which were simulated with slower speed. However, it is restrained than that in Fig. 4.6(b) which shows the result of the basic path following algorithm in which the speed effect is not considered. The magnitude of errors observed in the Scenario 4.7 are $|\psi_e| = 6.78^\circ$ and $|d'_e| = 0.37$ as indicated in Fig. 4.9(g).

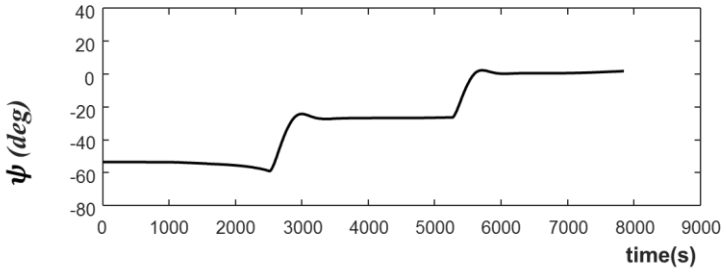
In conclusion, the ship equipped with the improved path following algorithm including the speed effect could follow the pre-planned track with small deviation in the all simulations. In case that the ship executed a rudder to change her direction, the value of yaw rate became higher as her speed became faster. However, through the rudder motion in the simulations performed in this section, the improved path following algorithm showed stable rudder action despite operation at high speed.



(a) Ship trajectory

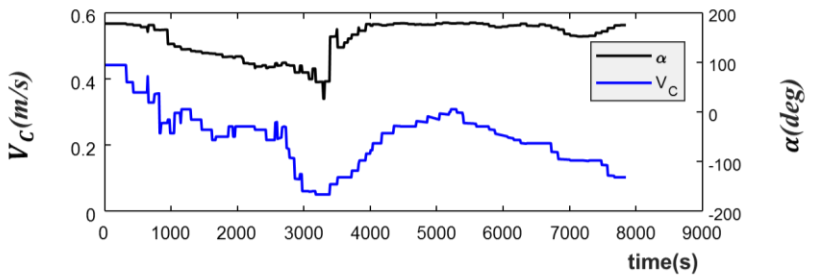


(b) Time history of rudder angle

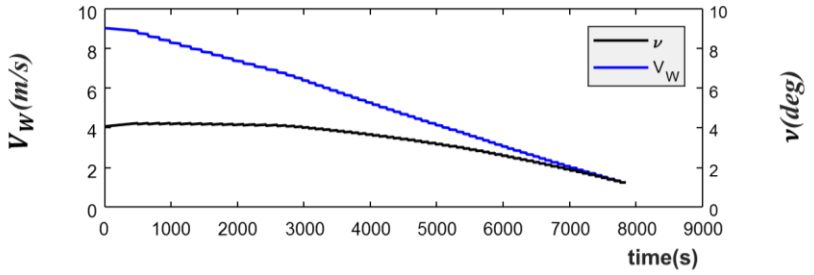


(c) Time history of heading angle

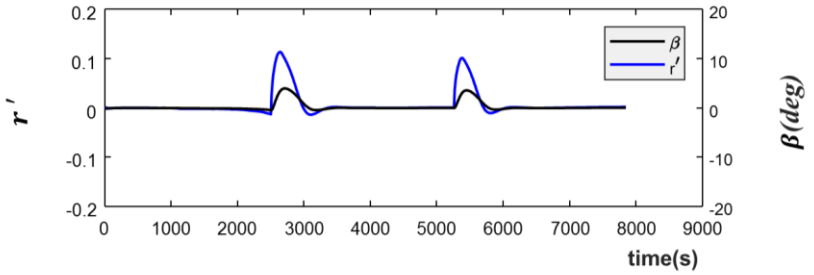
Fig. 4.7 Simulation results for the Scenario 4.5



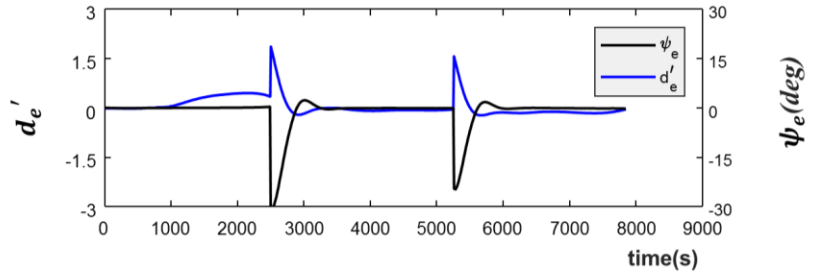
(d) Time history of speed and direction for current



(e) Time history of speed and direction for wind

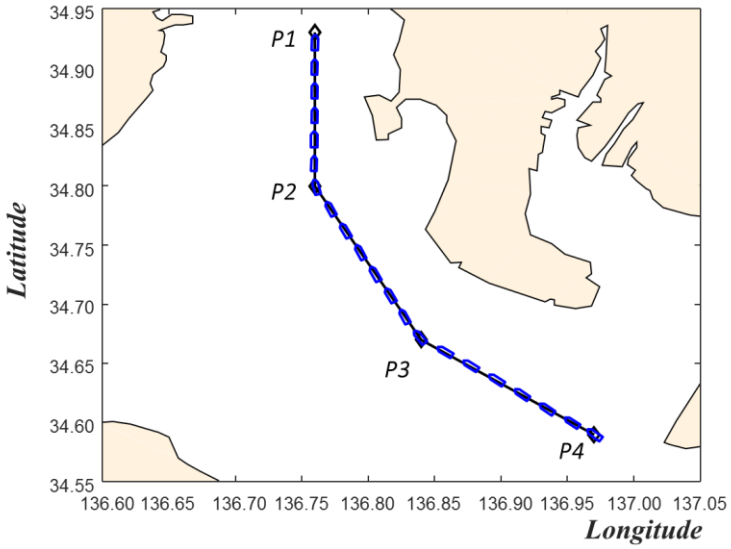


(f) Time history of drift angle and non-dimensional yaw rate

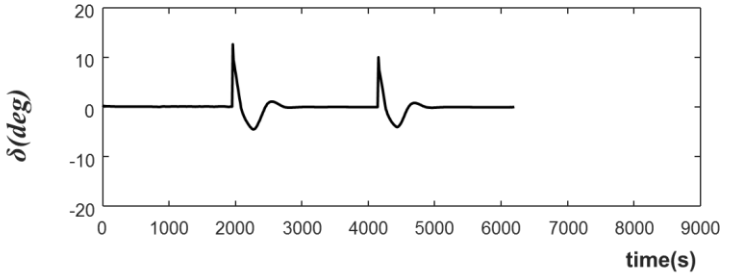


(g) Time history of heading error and non-dimensional cross track error

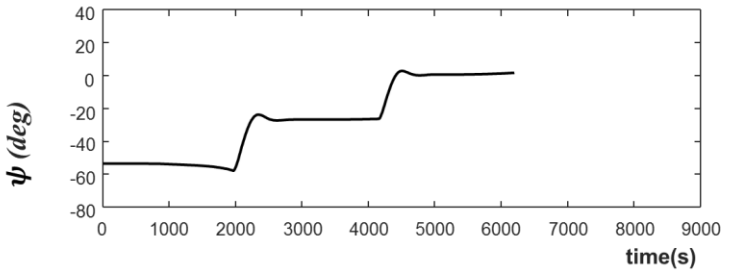
Fig. 4.7 Simulation results for the Scenario 4.5



(a) Ship trajectory

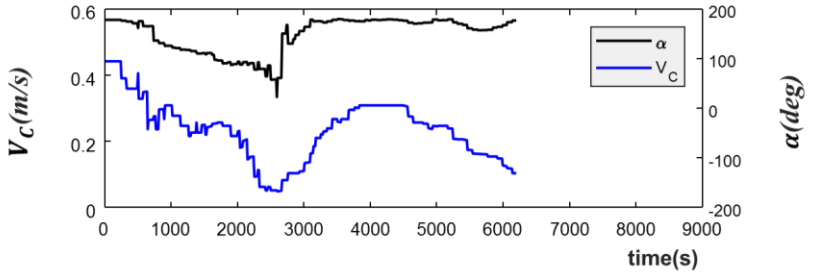


(b) Time history of rudder angle

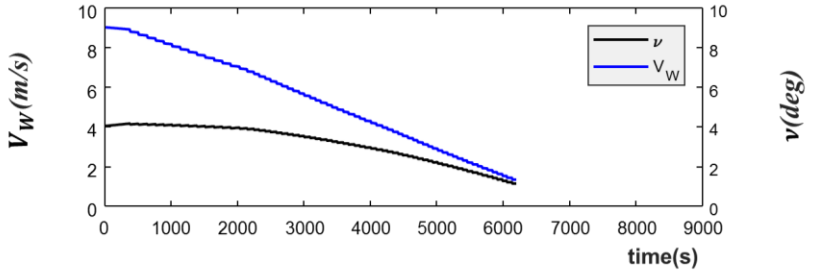


(c) Time history of heading angle

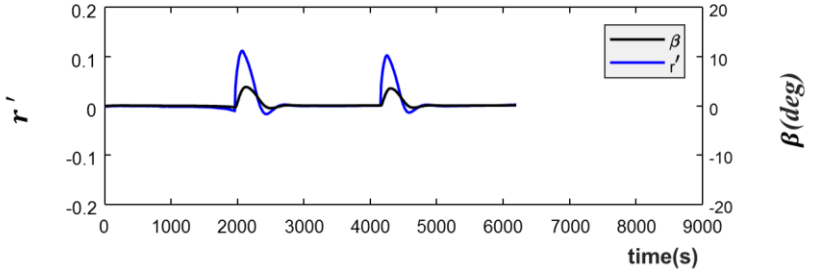
Fig. 4.8 Simulation results for the Scenario 4.6



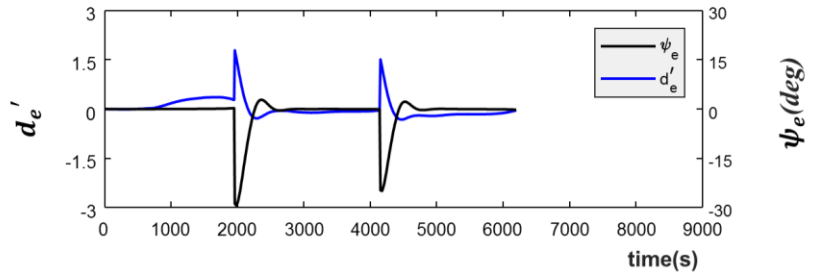
(d) Time history of speed and direction for current



(e) Time history of speed and direction for wind

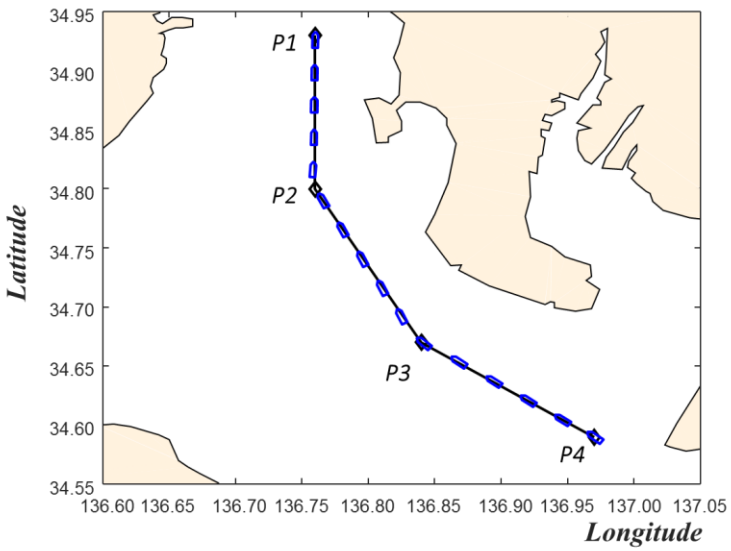


(f) Time history of drift angle and non-dimensional yaw rate

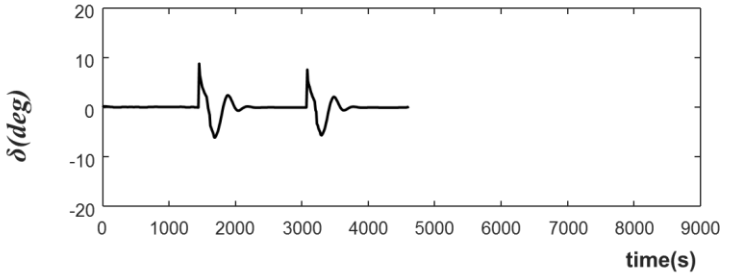


(g) Time history of heading error and non-dimensional cross track error

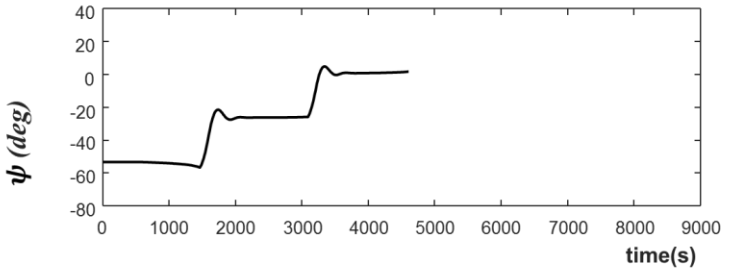
Fig. 4.8 Simulation results for the Scenario 4.6



(a) Ship trajectory

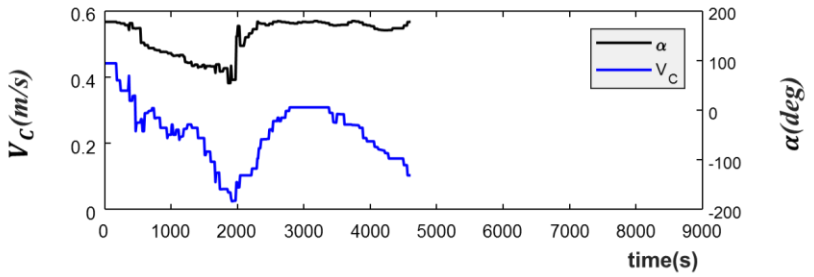


(b) Time history of rudder angle

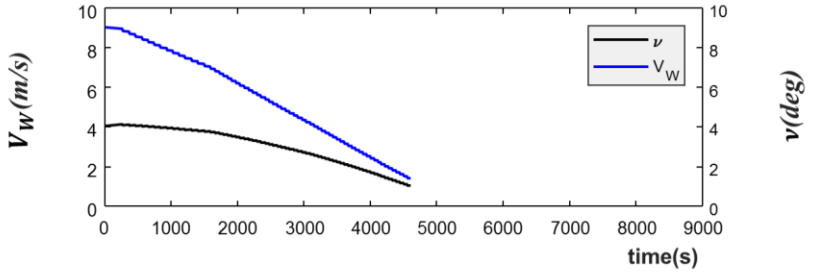


(c) Time history of heading angle

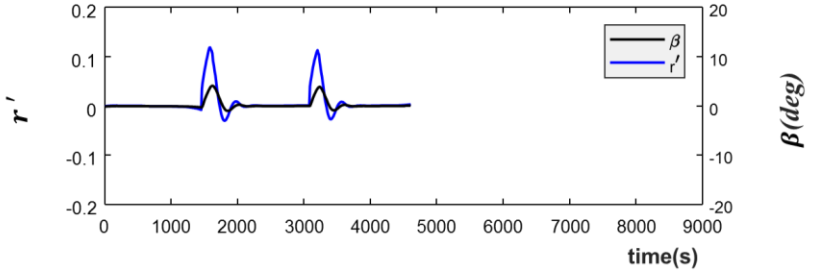
Fig. 4.9 Simulation results for the Scenario 4.7



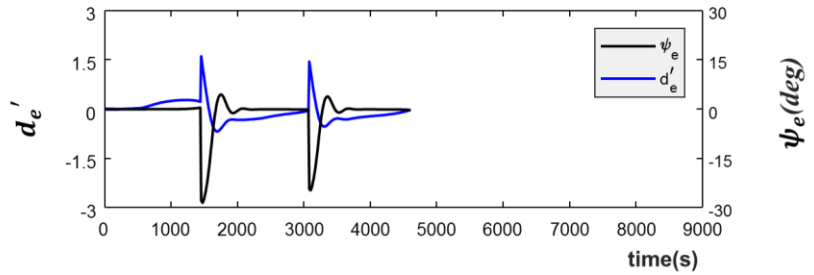
(d) Time history of speed and direction for current



(e) Time history of speed and direction for wind



(f) Time history of drift angle and non-dimensional yaw rate



(g) Time history of heading error and non-dimensional cross track error

Fig. 4.9 Simulation results for the Scenario 4.7

4.5 Conclusions

The proposed basic and improved path following algorithm composed of both waypoints switching system and rudder control system were verified in this chapter through numerical simulations in realistic operation environment. The main conclusions of this chapter can be summarized as follows:

- Realistic external disturbances such as wind and current were reproduced using velocity vectors in real time based on actual measured data from the official organization. In addition, the pre-planned track was designed with the position data of waypoints actually used by ship's operators.

- Numerical simulations were carried out to verify the effectiveness of the proposed algorithm, the basic path following algorithm and the improved path following algorithm, under realistic environmental conditions.

- In the simulations, information of wind and current obtained from real sea was applied depending on ship's location and time. As a result, the ship equipped with the proposed algorithms can travel on desired track using realistic rudder angle. However, it has been found that including the speed effect in path following algorithm makes the rudder action taken to keep track more stable.

Chapter 5. Development of Automatic Collision Avoidance Algorithm

5.1 Introduction

Ships usually encounter other ships while following their pre-planned path to arrive at their destinations during voyage. Introducing the path following algorithm suggested in Chapter 3 alone is insufficient to complete a fully autonomous ship because the ship cannot avoid automatically colliding with other ships. Therefore, collision avoidance algorithm has further been studied in this chapter in order to achieve the development of fully autonomous ships. Additionally, accomplishment of the automatic collision avoidance algorithm is expected to contribute the safety at sea by reducing marine accidents.

Marine accidents cause property losses, environmental pollution due oil spill, and even casualties. Table 5.1 summarizes the types of marine accidents that occurred in Japan during last 5years, which were investigated by Japan Transport Safety Board [62]. Collisions between ships or between a ship and obstacles accounts for about 30.3% of all marine accidents at highest rate and takes place more than 200 every year. Hence, researches on automatic collision avoidance which is the key element of automatic navigation have been conducted in an effort to improve the safety of ship navigation.

When a ship encounters other ships at sea, she should decide when she starts to take an action to avoid collision and what kind of manoeuvring motion to be taken. Studies on inferring the timing to start collision avoidance action can be classified into two methods. One is the application of ship safety domain and the other is the evaluation of collision risk index using the Closest Point of Approach (CPA).

The ship domain means the minimum virtual safety zone around a ship that enables ship operators to take an action to avoid potential collision. Namely, if obstacles or other ships invade the present ship domain, collision avoidance action is performed immediately. Beginning with the first concept of the ship domain which is an ellipse centered on ship's position based on geometrical model created by Fujii and Tanaka [19], the ship domain has been briskly researched since 1970s. The configuration of the domain has been evolved into a variety of outlines such as a combination of three sectors, a semi-ellipse and a circle, taking

into account the International Regulations for Preventing Collision at Sea (COLREGs), ship speed, traffic pattern, and so on [20][24][63][64][65][66].

On the other hand, the evaluation of collision risk was achieved though assembling Distance to the Closet Point Approach (DCPA) and Time to the Closest Point Approach (TCPA), which came from conception developed by Iwasaki and Hara [25]. When the degree of the risk of a ship reaches a threshold value which deemed unsafe, she should take an action to avoid collision. Fuzzy logic on the basis of the knowledge and experience of ship operators has been typically utilized by inputting DCPA and TCPA as antecedent part in order to obtain collision risk index as consequent part [26][67].

Table 5.1 Statistical data of marine accidents in Japan (2014~2018) [62]

Year	2014	2015	2016	2017	2018
Collision	265	244	217	200	240
Contact	116	102	94	96	88
Grounding	213	202	163	181	169
Sinking	7	5	5	13	21
Flooding	11	12	19	22	26
Capsizing	61	56	46	55	51
Fire	35	38	26	27	24
Explosion	1	3	3	3	2
Vessel missing	0	0	0	0	0
Facility damage	37	20	21	23	24
Fatality/ Injury	150	122	144	143	179
Others	3	1	0	0	0
Total	899	805	738	763	821

The advantage of the ship domain is tolerance for flexible consideration of various influencing factors. However, requiring many training samples which should be given by experienced navigators to make proper shapes is the disadvantage of the ship domain [68]. Whereas, collision risk index is evaluated according to fuzzy rules that already reflect expertise for inferring the starting point to take an action. Therefore, the collision avoidance algorithm to be developed in this chapter applies the collision risk indicator for inferring the starting point to take an action.

Furthermore, reinforcement learning which is a branch of machine learning is employed in a bid to settle a course of action to evade crash. In order to implement the collision avoidance algorithm, current circumstances around a ship should be modelled. However, it is generally difficult to model every complex situation caused by the combination of various variables such as the speed and heading of other ships and traffic volume which changes every moment during ship operation. The reinforcement learning is a method that an agent can learn process to achieve own goal by repeating trial and error similar to a human being. Thus, if the reward for the achievement and failure of the objective of collision avoidance problem is set properly, the ship can judge the situation by herself like a human and take an action to avoid crash when the collision risk was recognized.

The reinforcement learning is receiving a lot of attention because it is expected to solve problems similar to human thinking. In case of playing Atari games, controlling 3D locomotion, human level performance has already been achieved by Deep Reinforcement Learning (DRL) [69][70][71]. Google DeepMind has shown for the first time that a computer can solve problems as much as human level thorough working out the Atari game using Deep Q-network (DQN) [69] which is one of the algorithm of DRL. DQN can be apply to problems with high-dimensional observation space and low-dimensional action space defined as discrete space. However, most of tasks in real world such as physical control have high-dimensional action space defined as continuous space. Therefore, Deep Deterministic Policy Gradient (DDPG) [70] has been introduced as another algorithm of DRL to resolve problems with high-dimensional observation space and action space. Since the reinforcement learning has just begun to be utilized in the marine filed, DQN has been usually applied to solve problems of collision avoidance problem [32]. In this research, to find which method of DQN and DDPG is more appropriate on collision avoidance problem, decision making problem of collision avoidance to prevent collisions will be resolved by using DQN and DDPG.

Actions can be taken by a ship is assumed to change her course angle from -30˚ to 30˚ around current course angle using the rudder controller described in Chapter 3. Since DQN can handle only discontinuous action space, the action spaces has divided for applying continuous and high dimensional domains such as ship control. Therefore, the course angles of the actions range were discretized at 10˚ intervals. To complement of the disadvantages of DQN, DDPG was developed for resolving problems on continuous and high dimensional action space. DDPG can intactly exercise the continuous action space for course angle. Numerical simulations will be carried out to verify the collision avoidance algorithm proposed in this Chapter. Through the simulations, it will be turned out that which method of DQN and DDPG is more suitable for collision avoidance problem.

5.2 Reinforcement learning

5.2.1 Basic concept of reinforcement learning

Machine learning algorithms, an area of artificial intelligence that allows computers to learn by their experience, are divided into three classes: supervised learning, unsupervised learning, and reinforcement learning. Supervised learning trains a model on known input and output data that mean fully labelled examples so that it can predict future outputs. In real world, however, it is difficult to know exact output values required for supervised learning and dynamic states information changes over time. For the reason, it is not suitable for application in predicting ship motion. Unsupervised learning finds hidden pattern or intrinsic structures in input data. The data given to unsupervised learning are not labelled. It represent that the input variables are only given with no corresponding output variables.

Reinforcement learning has slightly different attributes from supervised learning and unsupervised learning because it does not use any data set. In reinforcement learning, a learner or decision-maker called “agent” tries behaviour without prior knowledge in “environment” that means physical world in which the agent operates. The agent learns through acquiring “reward” as an evaluation value of the action taken. This process has the same structure as trial and error learning, which is the basic learning method of human being and animals through experience. Fig. 5.1 indicates relationship between the agent and the environment. At each time step t, the agent receives a reward r_t for current state s_t and take an action a_t according to the trained model. One time step later, the environment changes s_t

to the next state s_{t+1} in response to the action a_t, and then the agent receives a new reward r_{t+1}. As aforementioned, a problem of reinforcement learning can be solved by repeating sequential process in order to achieve maximum sum of rewards.

The agent implements a mapping called agent's policy, which is a kind of function that outputs an action when a state comes in as input. The policy is nominated π, where $\pi(s, a)$ is the probability that $a_t = a$ if $s_t = s$. Objective of reinforcement learning is finding optimal policy π^* and maximizing the total amount of reward which the agent receives over the long run.

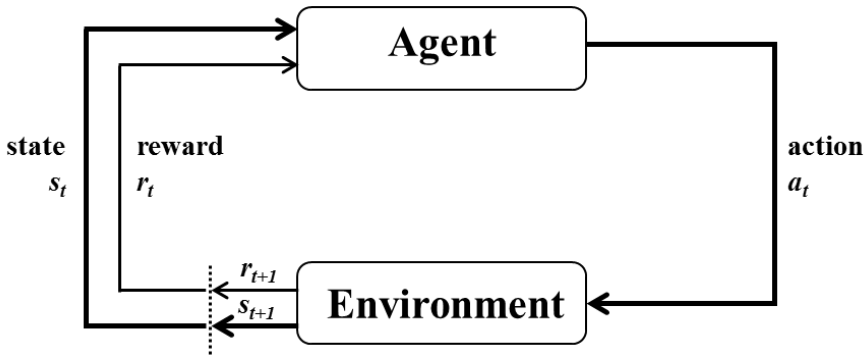


Fig. 5.1 Interaction between agent and environment in reinforcement learning

5.2.2 Markov decision process

Various algorithms such as Q-learning and SARSA has been developed to solve the reinforcement learning problem, which is defined based on the MDP. However, the classic reinforcement learning algorithms are difficult to apply to real world with high dimensional space. To resolve the problem, Deep Reinforcement Learning (DRL) was devised by applying Deep Neural Networks (DNN) that provides a general framework for approximating non-linear functions from training examples. This thesis employed Deep Q-network (DQN) for discrete action space and Deep Deterministic Policy Gradient (DDPG) for continuous action space.

Markov decision process (MDP) devised by Bellman and Howard [72] at least in the 1950s provides a mathematical framework for modelling decision-making. MDP is applied to

formulate a reinforcement learning problem that needs to determine actions sequentially. MDP can be represented as 4-tuple (S, A, P, R) and the each element is defined as follows:

- S : a finite set of states
- A : a finite set of actions
- P : a state transition probability
- R : a reward function

State transition probability means the possibility of the next state s' to be reached when a specific action a is performed in a specific state s. It is expressed as Eq. (5.1).

$$P_{ss'}^{a} = P\{s_{t+1} = s' | s_t = s, a_t = a\} \tag{5.1}$$

If any current state and action, s and a, are given together with next state s', the expected reward is demonstrated as shown in Eq. (5.2).

$$R_{ss'}^{a} = E\{r_{t+1} | s_t = s, a_t = a, s_{t+1} = s'\} \tag{5.2}$$

The goal of reinforcement learning is to choose a policy that maximizes the expected sum of future reward value. The discount factor nominated as γ reduces the value of reward as time goes by and it has a value between 0 and 1. The cumulative reward is called as the return G_t, which is given as,

$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \tag{5.3}$$

Where, k denotes the end of an episode and t is the time index.

The agent utilizes value function so as to decide what kind of policy is how good. A way of measuring the "goodness" in a particular state is to use value function and it can be classified as state-value function and action-value function. State-value function denoted as $V^{\pi}(s)$ is defined as the sum of rewards expected to be received when the agent follows the corresponding policy π from current state s. Action-value function indicated as $Q^{\pi}(s, a)$ is the expected sum of rewards starting from current state s, if the agent takes an action a and then follows a policy π. State-value function and action-value function are expressed as the following equations:

$$V^{\pi}(s) = E_{\pi}\{G_t | s_t = s\} = E_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \, | s_t = s\right\} \tag{5.4}$$

$$Q^{\pi}(s,a) = E_{\pi}\{G_t | s_t = s, a_t = a\} = E_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \, | s_t = s, a_t = a\right\} \tag{5.5}$$

Solving a reinforcement learning task means detecting an optimal policy π^* that achieves a lot of reward over the long run. The Bellman equations formulate the problem of maximizing the expected sum of reward in terms of a recursive relationship. Namely, it represents the relationship between the value function of a particular state and the value function of the following state. A policy π is deemed to be better than or equal to that of other policy π' if the expected return following π is greater than that following π' for all states, which implies $V^{\pi}(s) \geq V^{\pi'}(s)$. Thus, optimal value functions $V^*(s)$ is defined as,

$$V^*(s) = \max_{\pi} V^{\pi}(s) \tag{5.6}$$

Similarly, the optimal action-value function $Q^*(s,a)$ expressed as follows:

$$Q^*(s,a) = \max_{\pi} Q^{\pi}(s,a) \tag{5.7}$$

This function gives the expected return for taking an action a in state s and thereafter following an optimal policy. Thus, we can write Q^* in terms of V^* as follows:

$$Q^*(s,a) = E\{r_{t+1} + \gamma V^*(s_{t+1}) | s_t = s, a_t = a\} \tag{5.8}$$

Bellman optimality equations for value functions formulas, V^* and Q^*, are demonstrated as follows:

$$V^*(s) = \max_{a} \sum_{s'} P_{ss'}^{a} \left[R_{ss'}^{a} + \gamma V^*(s')\right] \tag{5.9}$$

$$Q^*(s,a) = \max_a \sum_{s\prime} P^a_{ss\prime} \left[R^a_{ss\prime} + \gamma Q^*(s',a')\right] \tag{5.10}$$

If we know the state transition function $P^a_{ss\prime}$, V^* and Q^* are found through the Bellman optimality equations and π^* can be figured out. It means solving the reinforcement learning problem.

5.2.3 Deep Q-network

A basic knowledge of Q-learning is necessary to understand DQN. Q-learning introduced by Watkins [73] finds the optimal policy π^* by learning the action-value function and has been widely used as off-policy temporal difference algorithm. An action is selected with respect to ε-greedy policy, which initially allows the agent to do a lot of exploration by choosing action randomly and learned action comes to be chosen as the episodes are repeated. Action-value function also called Q-function is updated at each time step according to the following equation. Where, α is learning rate and has a value from 0 to 1.

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha[r_{t+1} + \gamma \cdot \max_a Q(s_{t+1},a) - Q(s_t,a_t)] \tag{5.11}$$

DQN was developed to supplement Q-learning limited to task with small state spaces. Mnih et al. [69] introduced two new methods in Q-learning: experience replay memory and target network. The experience replay means storing the agent's experience and then randomly extracting a part of the stored experience data to train the network. In general, the sampled trajectories from the environment are temporally correlated and if these trajectories are used to train the network it would lead to over fitting in the network and the network would not be able to learn effectively [74]. Therefore, the experience replay is used in order to break the temporal correlation of data points while training. The agent's experience (s_t, a_t, r_t, s_{t+1}) is saved in replay buffer every time step and a fixed number of samples are randomly extracted from replay buffer to train the network. In DQN, loss function *Loss* for updating the neural network with weight θ uses Mean Squared Error (MSE). The network is trained by minimized loss function $Loss$ and its equation is given as:

$$Loss = E\left[\left(y - Q(s,a,\theta)\right)^2\right],$$
$$y = E\left[\left(r_{t+1} - \gamma \max_{a'} Q(s,a',\theta)\right)^2\right]. \quad (5.12)$$

As shown in Eq. (5.12), target network y computed via the Bellman equations is non-stationary because y uses same Q-function with prediction network $Q(s,a,\theta)$ if the networks was updated. To overcome the problem, the weight of target network is fixed, and the target network is only periodically updated by Q-network. The two key aspects, experience replay memory and target network, make the network to stable and better. The completed algorithm is given below.

Table 5.2 Algorithm of Deep Q-Network (DQN) [69]

Algorithm 1 Deep Q-learning with experience replay

Initialize replay memory D to capacity N

Initialized action-value function Q with random weights θ

Initialize target action-value function $\hat{Q}$ with weights $\theta^- = \theta$

for episode = 1, M **do**

Initialize sequence $s_1 = \{x_1\}$ and preprocessed sequenced $\phi_1 = \phi(s_1)$

for $t = 1$, T **do**

With probability ε select a random action a_t

Otherwise select $a_t = \max_a Q(\phi(s_t), a; \theta)$

Execute action a_t in emulator and observe reward r_t and image x_{t+1}

Set $s_{t+1} = s_t, a_{t+1} = a_t, x_{t+1}$ and preprocess $\phi_{t+1} = \phi(s_{t+1})$

Store transitions $(\phi_t, a_t, r_t, \phi_{t+1})$ in D

Sample random minibatch of transitions $(\phi_j, a_j, r_j, \phi_{j+1})$ in D

Set $y_j = \begin{cases} r_j & \text{if episode teminates at step } j+1 \\ r_j + \gamma \max_{a'} Q\left(\phi_{j+1}, a'; \theta^-\right) & \text{otherwise} \end{cases}$

Perform a gradient descent step on $(y_j - Q(\phi_j, a_j; \theta))^2$ with respect to the network parameters θ

Every C steps reset $\hat{Q} = Q$

end for

end for

5.2.4 Deep deterministic policy gradient

DQN achieved to solve task with high dimensional observation spaces, however, it cannot handle continuous and high dimensional action spaces. In other words, DQN suits only discrete and low dimensional action spaces. Many problem of interest, especially physical control tasks, has continuous action space. If the action space is discretized too finely, the action space becomes too large and it is also hard to converge. Lillicrap et al. [70] presented Deep Deterministic Policy Gradient (DDPG) that accomplished well on continuous action space. DDPG is actor-critic approach based on Deterministic Policy Gradient (DPG) [75].

The actor-critic composes two structures, actor neural network that adjusts the stochastic policy and critic neural network that estimates the value function, Q-function. Namely, the action to be taken is selected by the actor network and the critic network judges how good each action is through approximating the Q-function. Actor and critic networks work together and are individually trained for the purpose. When the stochastic policy is applied to the continuous action space, the action with the highest probability is needed to be searched. However, since the process consumes too much time to be practical with large domain, a deterministic policy should be used for continuous action space. Therefore, the actor-critic algorithm applying a deterministic policy function using policy gradient called DPG has been developed by Silver et al [76].

DDPG modifies DPG inspired by insights from the DQN, replay memory and target network, which allow it to use neural network function approximators to learn in large state and action spaces. Target networks are two separate networks which cope with the actor and critic networks. They are employed to calculate the target y in the temporal difference error according to Eq. (5.11). The difference between DQN and DDPG on the target network is update method. The target network is updated periodically copying weights from DQN, whereas, DDPG used "soft" target updates, rather than directly copying the weights. The weights of target networks in DDPG are updated slowly by accumulating the learned networks. A major challenge of learning in continuous action spaces is exploration. DDPG constructed an exploration policy as adding noise sampled from a noise process N which is Ornstein-Uhlenbeck process [77]. The completed DDPG algorithm is described below.

Table 5.3 Algorithm of Deep Deterministic Policy Gradient (DDPG) [70]

Algorithm 2 DDPG algorithm

Randomly initialize critic network $Q(s,a|\theta^Q)$ and actor $\mu(s,|\theta^\mu)$ with weights θ^Q and θ^μ

Initialized target network Q' and μ' with weights $\theta^{Q'} \leftarrow \theta^Q, \theta^{\mu'} \leftarrow \theta^\mu$

Initialize replay buffer R

for episode = 1, M **do**

Initialize a random process $\mathcal{N}$ for action exploration

Receive initial observation state s_1

for $t = 1$, T **do**

Select action $a_t = \mu(s_t,|\theta^\mu) + \mathcal{N}_t$ according to the current policy and exploration noise

Execute action a_t and observe reward r_t and observe new state s_{t+1}

Store transitions (s_t, a_t, r_t, s_{t+1}) in R

Sample a random minibatch of $\mathcal{N}$ transitions (s_i, a_i, r_i, s_{i+1}) from R

Set $y_i = r_i + \gamma Q'\left(s_{t+1}, \mu(s_{i+1}\middle|\theta^\mu)|\theta^{Q'}\right)$

Update critic by minimizing the loss: $L = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i|\theta^\mu))^2$

Update the actor policy using the sampled policy gradient:

$$\nabla_{\theta^\mu} J \approx \frac{1}{N}\sum_i \nabla_a Q(s,a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu}\mu(s|\theta^\mu)|_{s_i}$$

Update the target networks:

$$\theta^{Q'} =\leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$$
$$\theta^{\mu'} =\leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$$

end for

end for

5.3 Collision avoidance algorithm

5.3.1 Overview

Collision avoidance algorithm suggested in this study consists of three parts: collector and analyser, decision maker, and rudder controller as shown in Fig. 5.2. In collector and analyser, the own ship that is controlled directly obtains information of all other ships approaching around own ship which is called the target ships such as position data, speed, heading and ship's type from Automatic Identification System (AIS) receiver and radar. Using the collected data of the target ships, DCPA and TCPA are calculated and the value of collision risk (CR) can be gained. In addition, if a situation that own ship is expected to crash with the target ships or obstacles is detected by the computed collision risk, the role of the give-away ship or the stand-on ship is ordained and the target ship's dynamic data is communicated to the decision maker.

Through the decision maker developed based on DRL, the action to be taken by own ship is judged. Namely, the own ship can decide whether she keeps the existing route or switch her heading direction toward starboard / port side and when she starts and ends the selected action. In general, collision avoidance behaviour is achieved through altering ship's heading to starboard side according to COLREGs in most cases, but it is recommended to choose useful veering direction in the case of overtaking. Therefore, the action given by the decision maker is composed of path following, avoidance toward starboard side, and avoidance toward port side. The threshold value of collision risk determining the start and end timing of avoidance was set as CR = 40 through computer simulations performed many times in advance. If the value of CR was greater than 40 such as CR = 50 or 60, the collision avoidance action was conducted too late, so that the ship collided with other ships in all episodes and the training was not completed. On the other hand, when the value of CR was less than 40 such as CR = 20 or 30, the ship could take avoidance measures in early stage but it lets the ship swerve too much far from the desired track. Due to the above, the threshold value for CR was set as 40.

When the obtained collision risk value is less than preset threshold value, $CR \leq 40$, the own ship continues to follow the original course, pre-planned track, utilizing the automatic path following algorithm described in Chapter 3. However, in case that the degree of collision risk is greater than the threshold value, $CR > 40$, the own ship is able to change her course as well

as maintain current track by adjusting her heading angle. Function to veer the heading angle is implemented by using the rudder controller explained in detail in Fig. 3.2

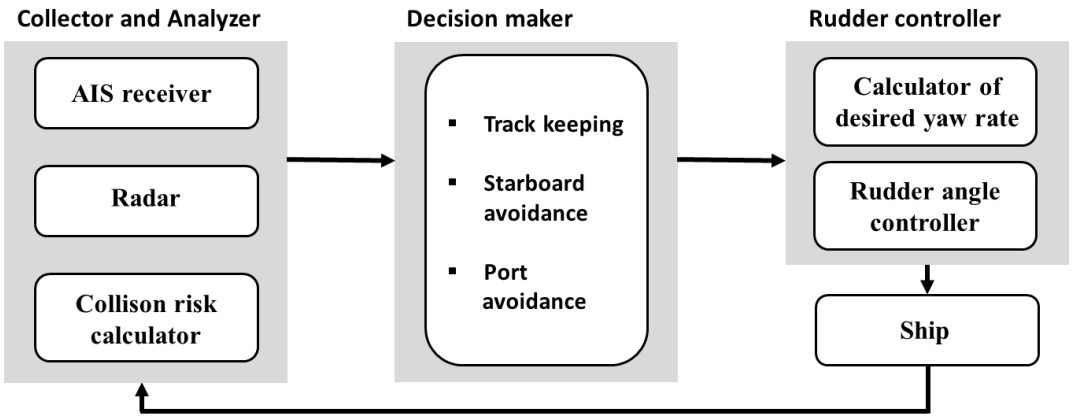


Fig. 5.2 Block diagram of collision avoidance algorithm

5.3.2 International regulation for preventing collisions

In order to reduce collision accidents at sea, International Maritime Organization (IMO) established International Regulations for Preventing Collisions at sea (COLREGs) in 1972 [78]. COLREGs is composed of five parts, and the collision avoidance algorithm suggested in this research is developed by utilizing Part B of COLREGs which gives the rules related to steering and sailing. According to Part B, encounter situations are classified into three types: head on, crossing, and overtaking. In each situation, role of ships is assigned to either the give-way ship which takes an action to avoid collision or the stand-on ship which maintains her course and speed. The main rules of COLREGs applied to this study are described as follows.

Rule 8

Action to avoid a collision

(a) Any action taken to avoid collision shall be taken in accordance with the Rules of this Part and shall, if the circumstances of the case admit, be positive, made in ample time and with due regard to the observance of good seamanship.

(b) Any alteration of course and/or speed to avoid collision shall, if the circumstances of the case admit, be large enough to be readily apparent to another vessel observing visually or by radar; a succession of small alterations of course and/or speed should be avoided.

(c) lf there is sufficient sea room, alteration of course alone may be the most effective action to avoid a close-quarters situation provided that it is made in good time, is substantial and does not result in another close-quarters situation.

(d) Action taken to avoid collision with another vessel shall be such as to result in passing at a safe distance. The effectiveness of the action shall be carefully checked until the other vessel is finally past and clear.

(e) If necessary to avoid collision or allow more time to assess the situation, a vessel shall slacken her speed or take all way off by stopping or reversing her means of propulsion.

(f)

(i) A vessel which, by any of these rules, is required not to impede the passage or safe passage of another vessel shall, when required by the circumstances of the case, take early action to allow sufficient sea room for the safe passage of the other vessel.

(ii) A vessel required not to impede the passage or safe passage of another vessel is not relieved of this obligation if approaching the other vessel so as to involve risk of collision and shall, when taking action, have full regard to the action which may be required by the rules of this part.

(iii) A vessel, the passage of which is not to be impeded remains fully obliged to comply with the rules of this part when the two vessels are approaching one another so as to involve risk of collision.

Rule 13

Overtaking

(a) Notwithstanding anything contained in the Rules of Part B, Sections I and II, any vessel overtaking any other shall keep out of the way of the vessel being overtaken.

(b) A vessel shall be deemed to be overtaking when coming up with another vessel from a direction more than 22.5 degrees abaft her beam, that is, in such a position with reference

to the vessel she is overtaking, that at night she would be able to see only the sternlight of that vessel but neither of her sidelights.

(c) When a vessel is in any doubt as to whether she is overtaking another, she shall assume that this is the case and act accordingly.

(d) Any subsequent alteration of the bearing between the two vessels shall not make the overtaking vessel a crossing vessel within the meaning of these Rules or relieve her of the duty of keeping clear of the overtaken vessel until she is finally past and clear.

Rule 14

Head-on situation

(a) When two power-driven vessels are meeting on reciprocal or nearly reciprocal courses so as to involve risk of collision each shall alter her course to starboard so that each shall pass on the port side of the other.

(b) Such a situation shall be deemed to exist when a vessel sees the other ahead or nearly ahead and by night she could see the masthead lights of the other in a line or nearly in a line and/or both sidelights and by day she observes the corresponding aspect of the other vessel.

(c) When a vessel is in any doubt as to whether such a situation exists she shall assume that it does exist and act accordingly.

Rule 15

Crossing situation

When two power-driven vessels are crossing so as to involve risk of collision, the vessel which has the other on her own starboard side shall keep out of the way and shall, if the circumstances of the case admit, avoid crossing ahead of the other vessel.

Rule 16

Action by give-way vessel

Every vessel which is directed to keep out of the way of another vessel shall, so far as possible, take early and substantial action to keep well clear.

Rule 17

Action by stand-on vessel

(a)

(i) Where one of two vessels is to keep out of the way the other shall keep her course and speed.

(ii) The latter vessel may however take action to avoid collision by her manoeuvre alone, as soon as it becomes apparent to her that the vessel required to keep out of the way is not taking appropriate action in compliance with these Rules.

(b) When, from any cause, the vessel required to keep her course and speed finds herself so close that collision cannot be avoided by the action of the give-way vessel alone, she shall take such action as will best aid to avoid collision.

(c) A power-driven vessel which takes action in a crossing situation in accordance with sub-paragraph (a)(ii) of this Rule to avoid collision with another power-driven vessel shall, if the circumstances of the case admit, not alter course to port for a vessel on her own port side.

(d) This Rule does not relieve the give-way vessel of her obligation to keep out of the way.

In summary of the above rules, once the role of the two ships was determined, the give-away ship should veer to starboard side preferentially. Encounter situations can be identified according to relative angle as shown Fig. 5.3.

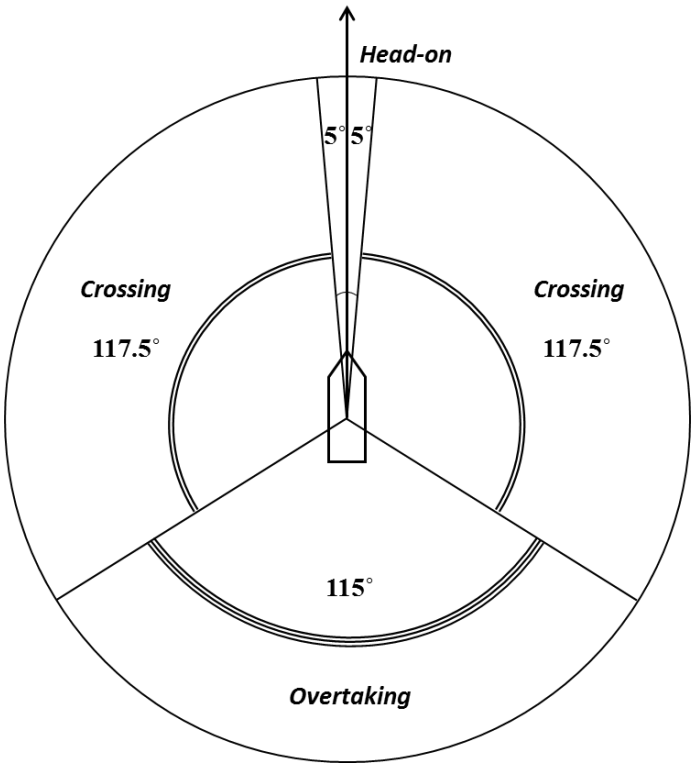


Fig. 5.3 Encounter situation accoring to COLREGs

5.3.3 Calculation of collision risk

When building a collision avoidance algorithm, it is important to infer the time to start an action for collision avoidance. In this study, we used the collision risk based on fuzzy inference to determine the start and terminate positions of an action to be taken to avoid crash. If the collision risk is bigger than a certain value, the give-away ship should perform collision avoidance action. Conversely, when the collision risk is less than the present value, the taken action must be stopped and then the ship should return to her original course. Collision risk is calculated by inspiring from fuzzy inference presented by Kjima and Furukawa [26]. As parameters for the fuzzy inference, DCPA and TCPA are employed and the elements are shown in Fig. 5.4. The equations regarding DCPA and TCPA are expressed in Eq. (5.13).

$$
\left.
\begin{aligned}
V_{xr} &= V_1 \cos\theta_1 - V_0 \cos\theta_0 \\
V_{yr} &= V_1 \sin\theta_1 - V_0 \sin\theta_0 \\
R &= \sqrt{(y_1 - y_0)^2 + (x_1 - x_0)^2} \\
\theta_{01} &= \tan^{-1}\frac{y_1 - y_0}{x_1 - x_0} \\
\theta_r &= \tan^{-1}\frac{V_{yr}}{V_{xr}} \\
V_r &= V_0 \cos(\theta_0 - \theta_r) - V_1 \cos(\theta_1 - \theta_r) \\
DCPA &= R \sin(\theta_{01} - \theta_r) \\
TCPA &= \frac{R \sin(\theta_{01} - \theta_r)}{V_r},
\end{aligned}
\right\} \qquad (5.13)
$$

Where, V_0 is own ship's velocity and V_1 is target ship's speed. R indicates the distance from the own ship to the target ship and θ_{01} is bearing angle of R. θ_0 and θ_1 are angles between x_0 -axis and the course of own ship and target ship, respectively. V_{xr} and V_{yr} are components of relative speed V_r between two ships in x_0- and y_0-axes and θ_r is bearing angle of the two components.

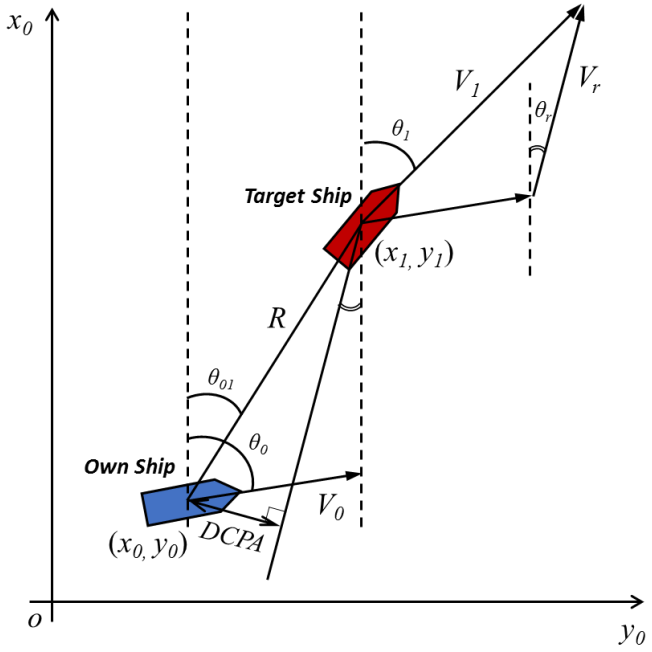


Fig. 5.4 Determination of DCPA and TCPA in an approach situation

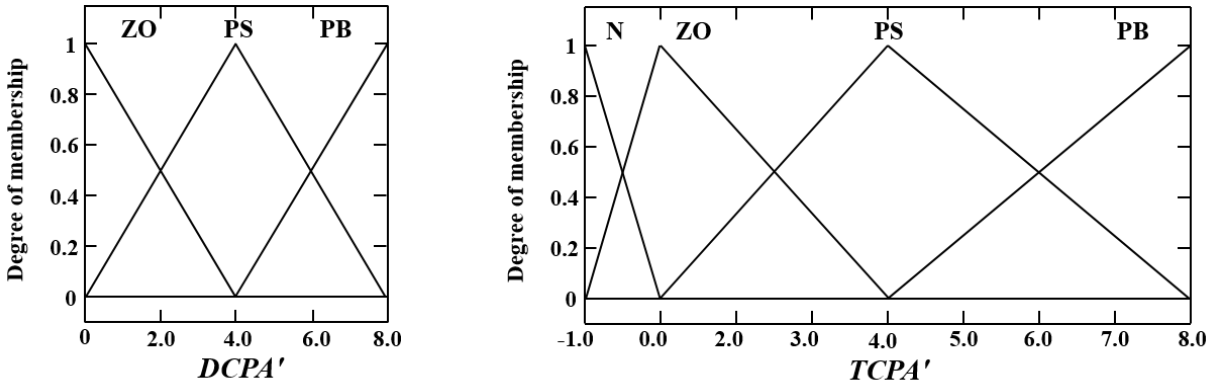
For fuzzy inference of collision risk, DCPA and TCPA are non-dimensionalized using ship's length L and speed U as shown in Eq. (5.14).

$$DCPA' = \frac{DCPA}{L}, \quad TCPA' = \frac{TCPA}{L/U}. \tag{5.14}$$

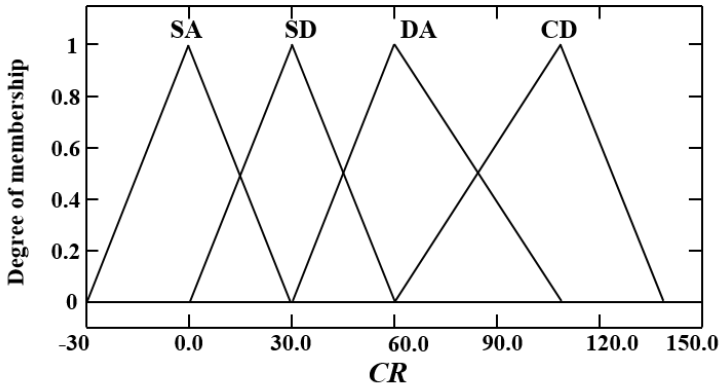
The non-dimensional DCPA denoted by DCPA′ is considered as the antecedent part and the non-dimensioanl TCPA specifed as TCPA′ is the consequent part of the membership function of fuzzy inference respectively. Fuzzy rules and membership functions for the collision risk inference are indicated in Table 5.4 and Fig. 5.5. DCPA′ is described by three classes: Near Zero (ZO), Positive Small (PS), and Positive Big (PB). TCPA′ has four classes: Negative (N), ZO, PS, and PB. Collision risk degree denoted by CR, is also constituted of four classes: Safe (SA), Slightly Dangerous (SD), Dangerous (DA), and Considerably Dangerous (CD). In the collsion avoidance algorithm, the threshold value of CR is detremend by 40. If CR > 40, the give-way ship takes collision avoidance action and, on the contrary to this, CR ≤ 40, the own ship stops the taking action and then returns to the previous track.

Table 5.4 Rules for the collision risk of fuzzy inference

		TCPA′			
		N	ZO	PS	PB
DCPA′	ZO	SA	CD	DA	SA
	PS	SA	DA	SD	SA
	PB	SA	SA	SA	SA



(a) antencedent part



(b) consequent part

Fig. 5.5 Membership functions for the collision risk fuzzy inference

5.3.4 Definition of markov decision process for collision avoidance

In the decision maker as shown in Fig. 5.2, reinforcement learning was utilized to decide action to be taken in order to evade collision. The reinforcement learning is one of ways to solve problems that need to determine behaviour sequentially. In order for a computer to solve any sequential problems, the problems should be mathematically defined through MDP. MDP is composed of state, action, reward, and policy. In the collision avoidance problem, the own ship as an agent moves along a desired course, original course, and if collision risk is detected in approaching situation with other ships, the own ship takes an action to avoid crash.

Depending on the assigned reward, policy is determined to conform with the taken action at each state. This section will describe state, action, and reward of the collision avoidance problem to make MDP model.

State

So as to make an accurate judgment at the decision maker depicted in Fig. 5.2, the own ship as the agent needs to fully understand the present situation. Thus, the state set S is established by position and dynamic data of two ships, own ship and target ship, to investigate the information of the current encounter situation. Eq. (5.15) expresses a set S of the collision avoidance problem.

$$S = \{x_o, y_o, \psi_o, U_o, r_o, \delta_o, x_t, y_t, \psi_t, U_t, r_t, CR\} \tag{5.15}$$

Where, subscript 'o' and 't' mean own ship and target ship, respectively. The data relating to the target ship is obtained from AIS and radar and present CR value is handed over from the collision risk calculator indicated in Fig. 5.2. x and y represent position of the ship. ψ is heading angle and U is ship's speed. r and δ signify the yaw rate and the rudder angle of own ship. The dynamic data of two ships, the own ship and the target ship, in the state set S can be represented as shown in Fig. 5.6.

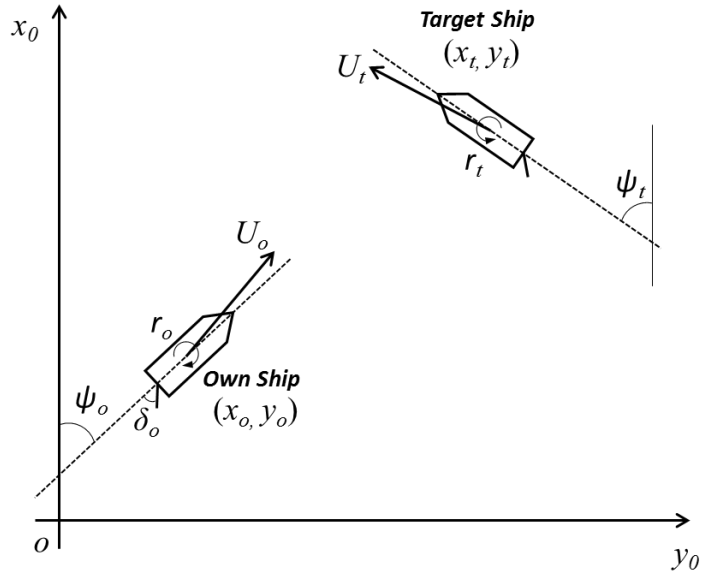
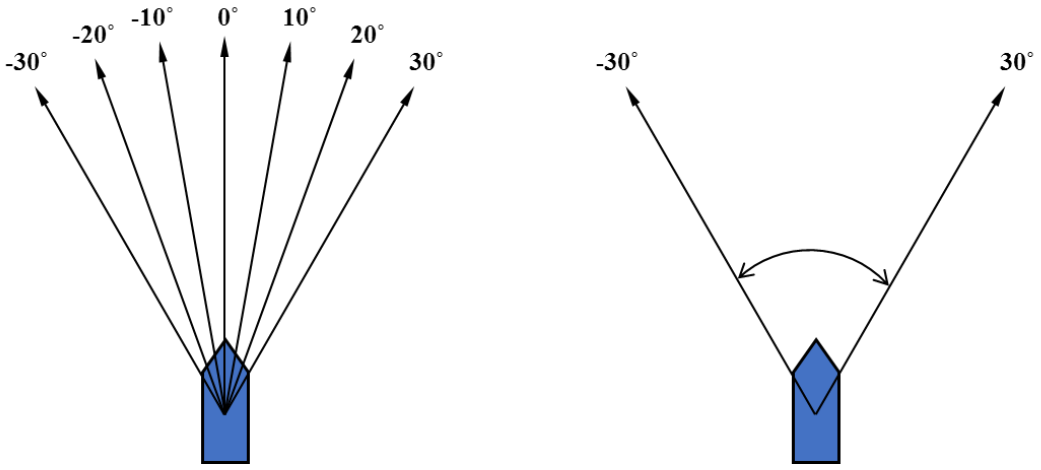


Fig. 5.6 State space of collision avoidance problem

Action

The own ship as the agent is able to have three kinds of action in the decision maker of the collision avoidance algorithm as shown in Fig. 5.2. If a collision possibility between a target ship approaching and own ship following an existing track defined by waypoints is recognized , the own ship assumed as a give-away ship needs to select an action to change her heading angle toward starboard side or port side. In the collision avoidance problem, the action indicates course angle to be changed from the current heading direction and the designated action space is allocated from -30˚ to 30˚. On the basis of the current heading angle of the own ship, the heading angle can be veered 30˚ each toward both starboard and port directions. As mentioned previously, numerical simulations using two methods of DRL that are DQN and DDPG are performed in this research. Since the behaviour that an agent can implement in DQN must be set up in discrete space, the action is split into seven at 10˚ intervals. On the other hand, DDPG can treat any action value within continuous action space between -30˚ and 30˚. The action spaces of DQN and DDPG are illustrated in Fig. 5.7.



(a) DQN (b) DDPG

Fig. 5.7 Action space of collision avoidance problem

Reward

Whenever an agent performs an action, the agent receives a reward value from the environment as shown in Fig. 5.1. The reward informs the agent whether the current action was good or bad. The reward function is important because the goal of reinforcement learning is to maximize the cumulative reward values. Thus, the reward function R in the collision avoidance problem is defined as follows:

$$R = \begin{cases} \exp(-d_e{}') & , \quad d_e \geq 0 \\ \exp(d_e{}') - 1 & , \quad d_e < 0 \\ -100 & , \quad \text{collision} \end{cases} \tag{5.16}$$

Where, $d_e{}'$ is non-dimensional cross track error calcualted by Eq. (3.1). When $d_e{}'$ has positive value ($d_e > 0$), it can be considered that the ship has moved to starboard side. On the other hand, if $d_e{}'$ has negative value ($d_e < 0$), the ship has been expected to alter direction to port side. In case of $d_e = 0$, it means that the ship follows the target track. COLREGs recommends a ship to move toward stardboard side primarily to avoid conflicts. Thus, when a ship moves toward the starboard direction accoridng to the international rules, reward is positve value. If the ship head for the port side, the agent ,the own ship, acquires negative value which can be regarded as penalty.

Exponential function is used to apply the reward value differently depending on the distance from the desired track. Accuratly following the desired track, it makes possible to minimize energy loss during operation and reach a destination in a short time because of the best route planned by skilled opertors considering water depth, geomatric condition, weather condition, and so on. Therefore, when the ship travells on the desired track, the reward becomes the maxmum value. On the other hand, when the own ship veers to starboard side, as the ship moves farr away from the target track due to altering starboard, she obtains less amount of the positive reward. In case that the own ship changes heading angle to port side in order to avoid the potential collision, the ship receives negative value as far as the distance from the target path.

Lastly, if the own ship collides with other ship, it receives -100. The reward value has the largest negative value in order for the ship to deem that the behavior which caused the

collision was the worst. Collision is established as when the distance between two ships, the own ship and the target ship, is less than four times of the ship's length to prevent passing each other too close.

5.4 Numerical simulations

5.3.1 Simulation conditions

For the collision avoidance simulations, it is necessary to introduce two ships as an own ship and a target ship. The KVLCC2 tanker scaled by 1/128 was selected as the both ships. The principle particulars of the subject ship are represented in Table 2.2 in Chapter 2. The numerical simulations were carried out on head-on situation in order to verify the effectiveness the developed collision avoidance algorithm. In spite of head-on situation, the roles of ships are assigned as the stand-on ship and the give-away ship respectively. The initial position and heading angle of the target ship is changed randomly whenever episode starts within the range of the head-on situation as shown in Fig. 5.3. The target ship treated as a give-way ship was assumed to travel her existing track without any action such as rudder manipulation. The own ship departs from fixed position in every episode. The initial conditions of two ships on each episode are defined as shown in Table 5.5. The model ship's speed 0.4547 m/s corresponds to 10 knots of full scale ship.

Table 5.5 Definition of initial conditions

	Own ship	**Target ship**
Position	$x_0/L = 0$ $y_0/L = 0$	$x_0/L = 40.0$ $-3.5 \leq y_0/L \leq 3.5$
Heading angle	0°	$175^\circ \leq \psi \leq 185^\circ$
Speed	0.4547 m/s	0.4547 m/s

5.3.2 Simulation results using DQN and DDPG

DQN and DDPG are implemented using Python and TensorFlow library [79] is employed to build deep neural networks. The learning rate in DQN is 0.0005 and the number of hidden layers is two. The network in DQN has 16 units for each hidden layer. In DDPG, the learning rate for the actor network and the critic network are 0.0001 and 0.001 respectively. The both networks are also consisted of two hidden layers. The actor network and the critic network have 400 units for the first hidden layer and 300 units for the second hidden layer respectively. The discount factor for two methods, DQN and DDPG, is set to 0.99.

Using the abovementioned hyperparameters, the collision avoidance problem of head-on situaion according to starting state as shown in Table 5.5 was learned and the training results of DQN and DDPG are presented in Fig. 5.8 and Fig. 5.9. The figures show the total accumulated reward per episode. In Fig. 5.8, the collision avoidance problem seems to have been resolved after about 900 episodes through DQN. On the other hand, since DDQP began to converge as soon as learning was started as shown in Fig. 5.9, it can be seen that training is completed faster than DQN.

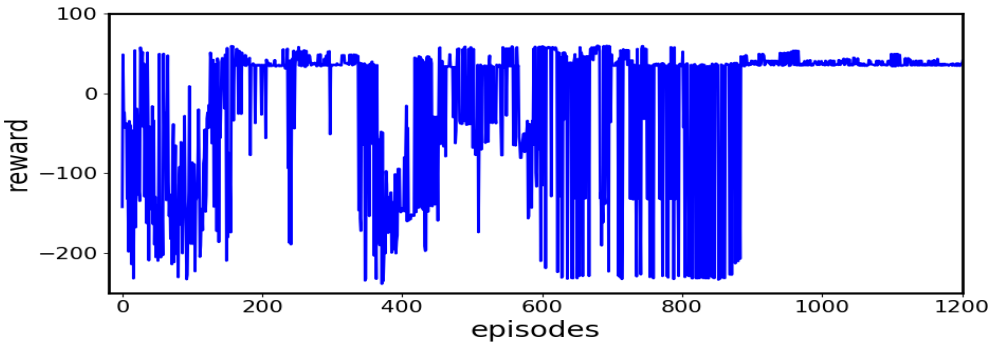


Fig. 5.8 Training results of DQN

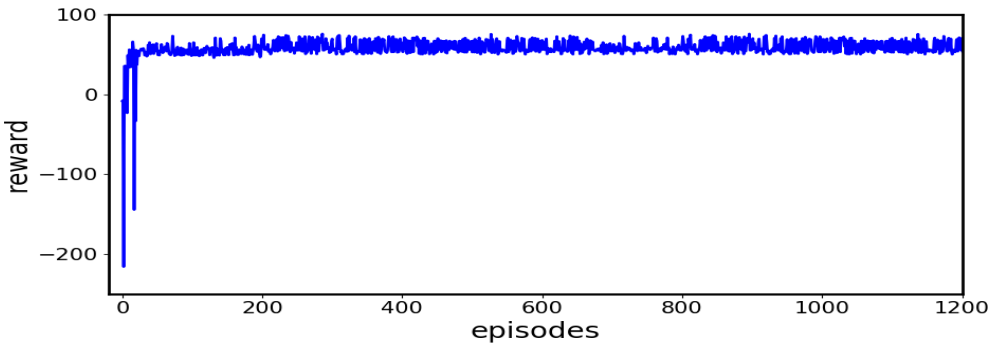


Fig. 5.9 Training results of DDPG

Applying the trained model of DQN and DDPG, the own ship will move according to optimal policy while coping with the target ship. Numerical simulations are carried out using the trained model in several cases of head-on situation set up depending on the location of a target ship. In order to investigate whether difference in manoeuvring performance as well as learning speed exist between the two methods, simulations are conducted for same encounter situation. Established target ship information in each encounter case is shown in Table 5.6. The relative angles selected within the range shown in Fig. 5.3 are 0˚, -5˚ and 5˚. It is assumed that the target ship approaches own ship from the front with the greatest relative angle on the starboard side or port side.

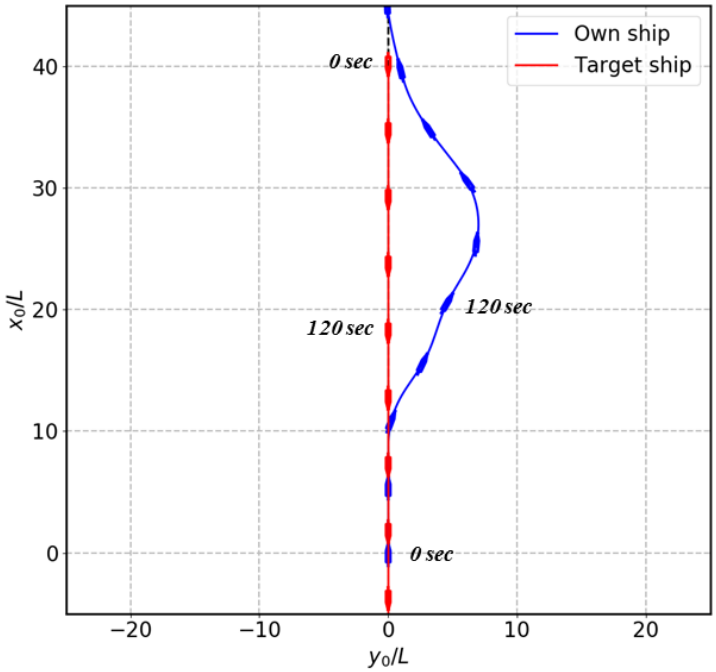
The simulation results for DQN are observed from Fig. 5.10 to Fig. 5.12. Fig.5. 10 shows the results of Case 1 when the target ship comes from relative angle 0˚. In Fig. 5.10, the blue line means the own ship's data and the red line indicates the target ship's data. The shapes of a ship in Fig. 5.10(a) depicts the positions of ships every 30 seconds and the black dotted line signifies a desired track that is the original route. The positions of the two ships were nearest at 110.0 seconds and the distance at that time was 4.08L. The own ship competed collision avoidance without invading the distance of 4.0L which was set to be considered as collision. Fig. 5.10(b) and Fig. 5.10(c) demonstrate time histories of rudder angle and heading angle. The rudder angle started to change after degree of collision risk, CR, exceed the threshold value CR = 40. Namely, an action to be taken was decided after 36.0 seconds when CR exceeded 40 as shown in Fig. 5.10(f). During the voyage, the own ship used the maximum absolute rudder angle as $|\delta|$ = 12.2˚. Fig. 5.10(d) and Fig. 5.10(e) show time histories of DCPA′ and TCPA′ used for obtaining CR. The CR in Fig. 5.10(f) increase at the beginning of the simulation, but the value decrease rapidly as soon as TCPA′ goes to zero. If TCPA′ is equal to zero or negative value, it means the own ship have passed the target ship.

Table 5.6 Definition of initial conditions for a target ship

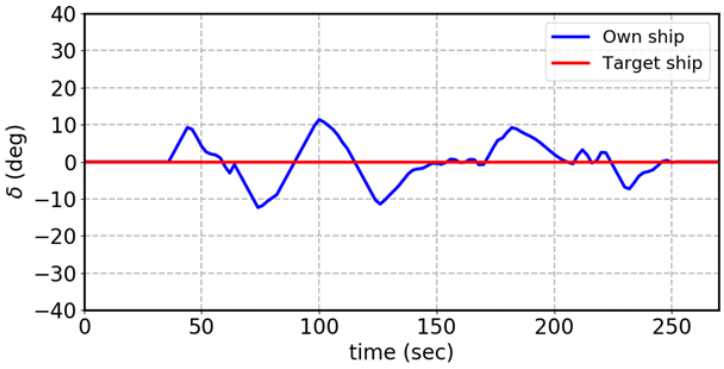
	Heading angle (˚)	Position (x_0/L, y_0/L)
Case 1	180	(40.0, 0.0)
Case 2	175	(40.0, -3.5)
Case 3	-175	(40.0, 3.5)

Fig. 5.11 presents the results of simulation in case that the target ship approaches own ship from port side of the own ship. As shown in Fig. 5.11(a), the two ships approached closest at 108.0 seconds as distance of $4.16L$. The rudder angle used in this simulation is between δ = -12.8˚ and δ = 11.8˚ as shown in Fig. 5.11(b). In order to obtain the maximum reward, the own ship traveled close to the original course in the early stages. The heading angle seen in Fig. 5.11(c) changed up to 39.6˚ to evade a crash and reached 0˚ after 250.0 seconds. In figures (d), (e), and (f), TCPA′ is zero at 110.6 seconds and after then CR decreased even through DCPA′ increased.

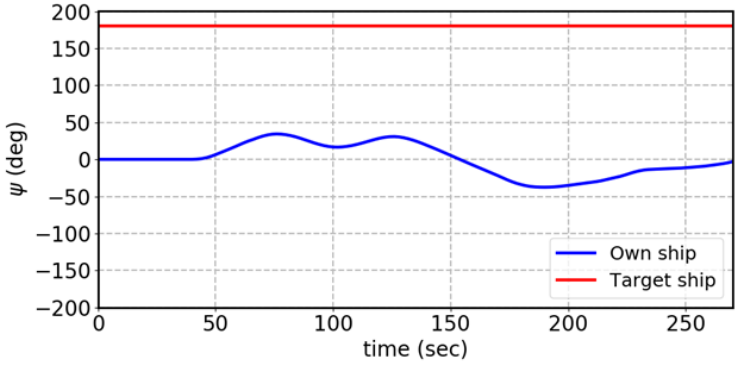
In Fig. 5.12, the target ship moves toward the own ship with relative angle 5˚ as Case 3. As shown in Fig. 5.12(a), the own ship performed the evasive motion with bigger turning radius comparing with other cases since the target ship comes from starboard side where the own ship should take action preferentially according to COLREGs. Therefore, the nearest distance from the own ship to the target ship is the longest among three cases as $4.57L$. However, the maximum magnitude of used rudder angle in Fig. 5.12(b) is the smallest as $|\delta|$ = 9.6˚. The CR has 61.5 as the maximum value when DCPA′ and TCPA′ are 0.02 and 4.09, respectively. Based on the simulation results of three cases, it can be said that the decision maker of the collision avoidance algorithm developed by DQN showed good performance to accomplish avoidance without causing collision.



(a) Ship trajectories

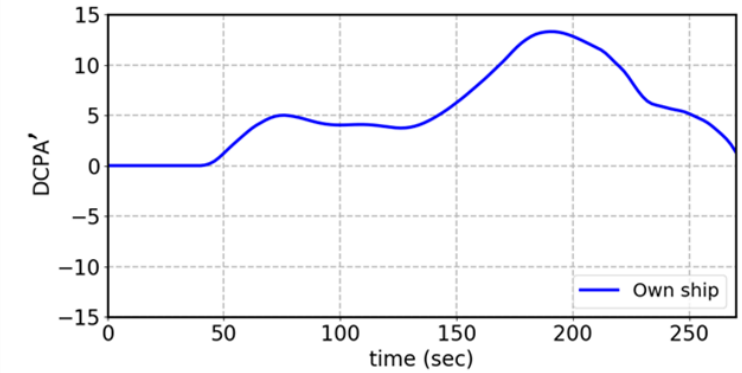


(b) Time history of rudder angle

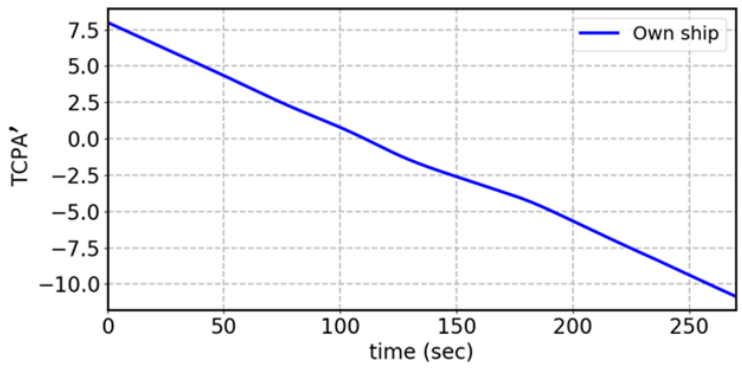


(c) Time history of heading angle

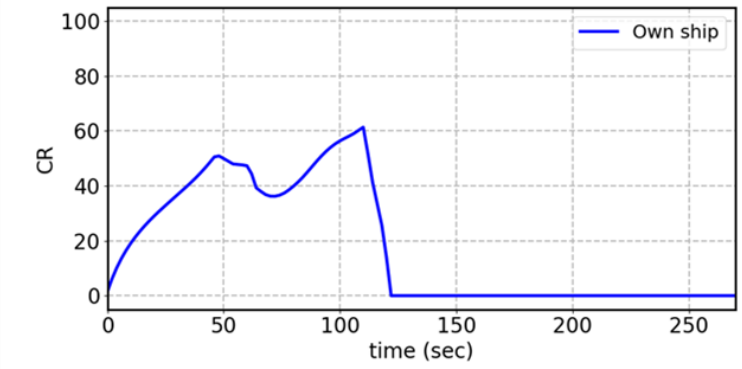
Fig. 5.10 Simulation results using DQN of the Case 1



(d) Time history of DCPA'

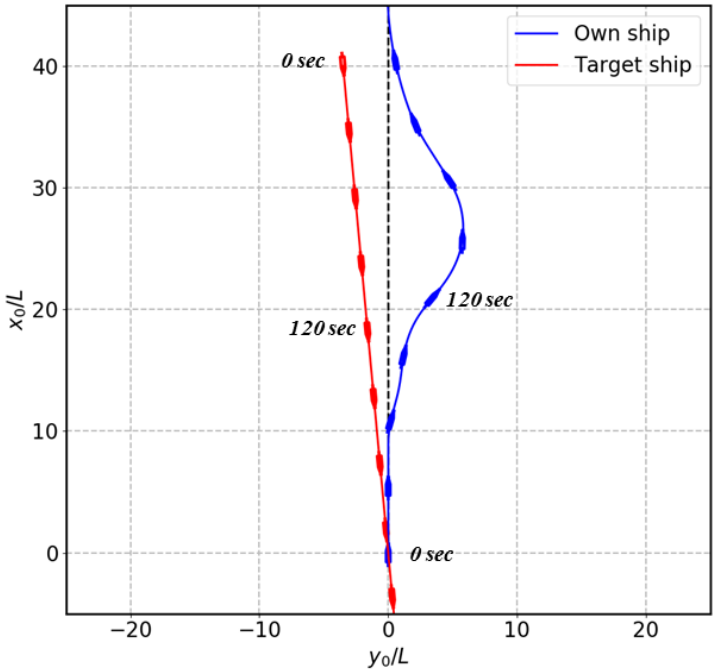


(e) Time history of TCPA'

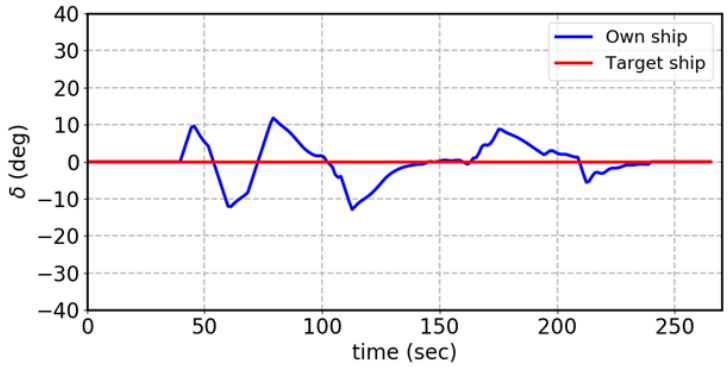


(f) Time history of CR

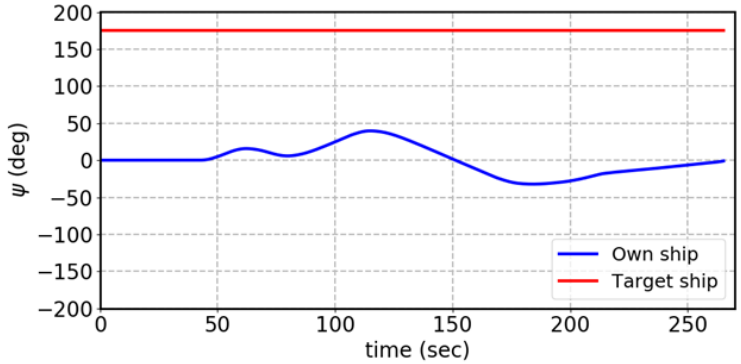
Fig. 5.10 Simulation results using DQN of the Case 1



(a) Ship trajectories

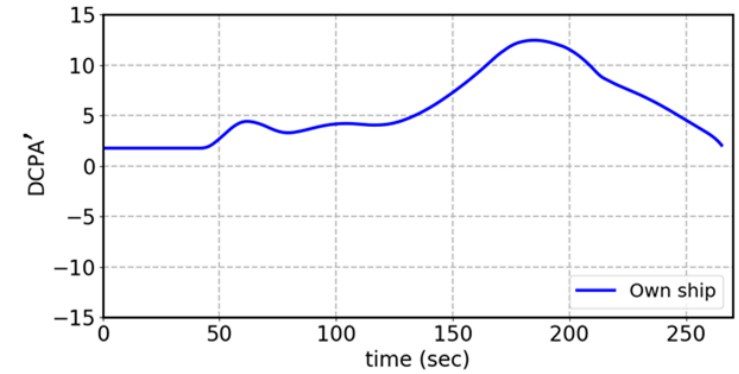


(b) Time history of rudder angle

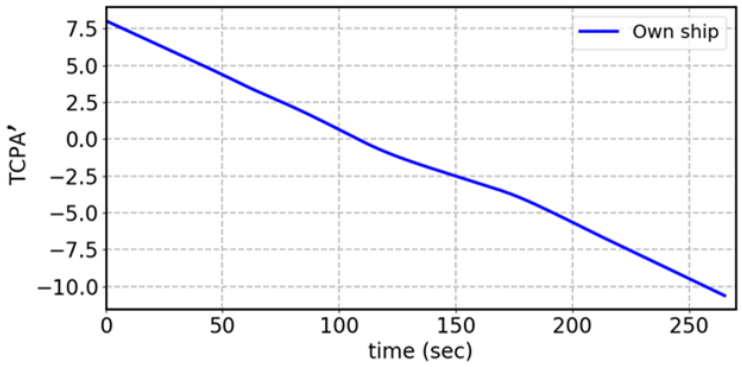


(c) Time history of heading angle

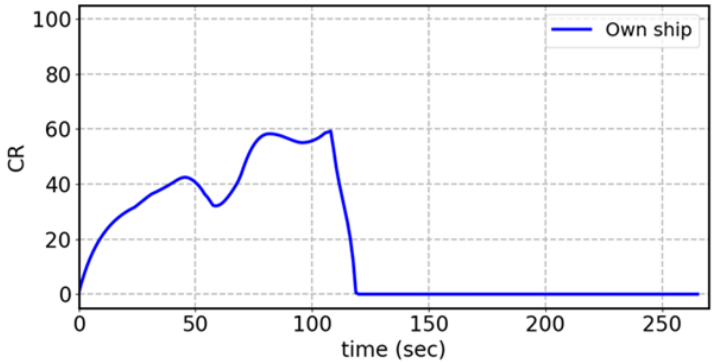
Fig. 5.11 Simulation results using DQN of the Case 2



(d) Time history of DCPA'

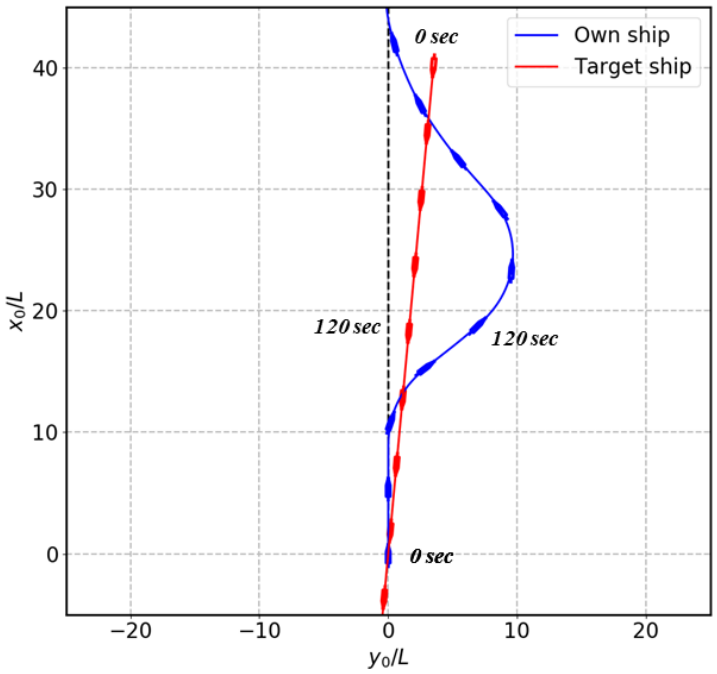


(e) Time history of TCPA'

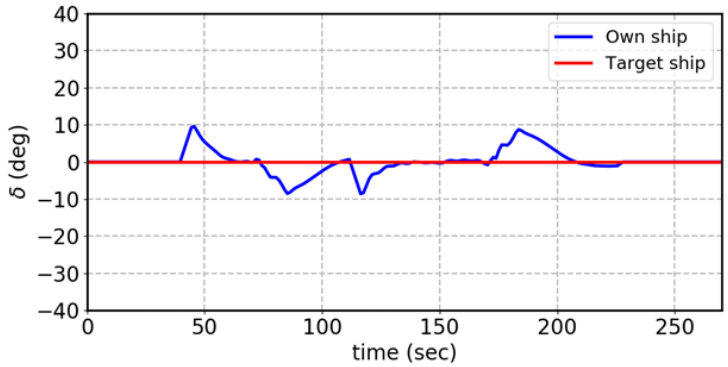


(f) Time history of CR

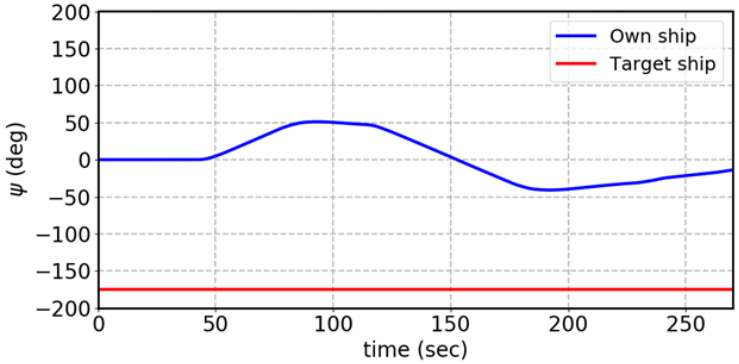
Fig. 5.11 Simulation results using DQN of the Case 2



(a) Ship trajectories

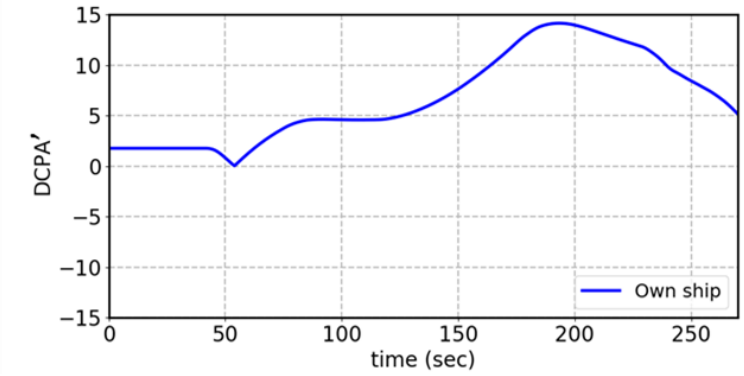


(b) Time history of rudder angle

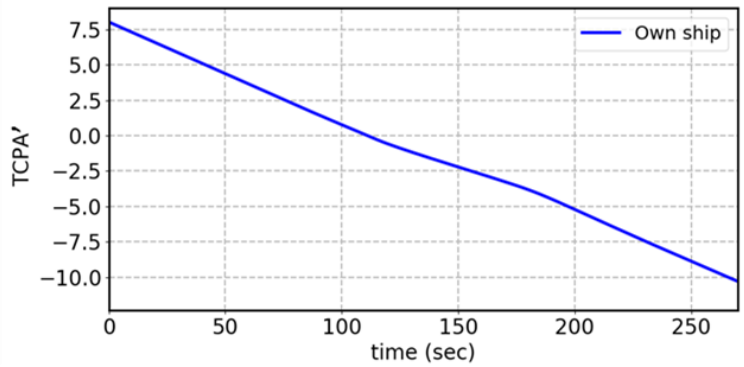


(c) Time history of heading angle

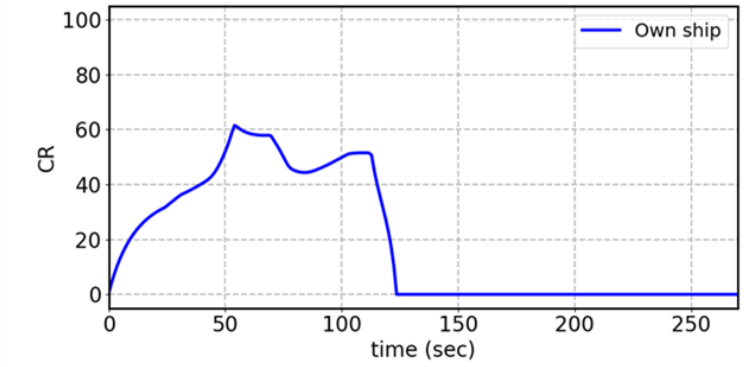
Fig. 5.12 Simulation results using DQN of the Case 3



(d) Time history of DCPA'



(e) Time history of TCPA'



(f) Time history of CR

Fig. 5.12 Simulation results using DQN of the Case 3

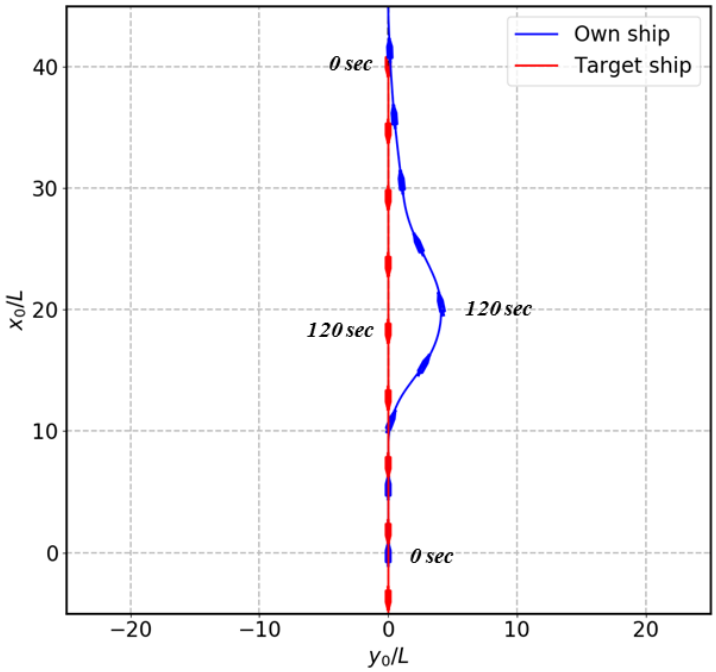
Simulation results for the collision avoidance algorithm using DDPG are shown in Fig. 5.13 to Fig. 5.15. Fig. 5.13 is the results of Case 1 when the target ship having the heading angle of 180° approached the own ship. While the own ship travels, the target ship marked in blue in Fig. 5.13(a) neared the own ship as much as distance of $4.06L$. Fig. 5.13(b) and Fig. 5.13(c) represent time histories of rudder angle and heading angle, respectively. The rudder is started to execute after 36.0 seconds, when CR exceeded 40. The heading angle toward the starboard side was modified to port side in order to return the original route after the collision action was stopped by getting CR which is less than the threshold value 40. Fig. 5.13(f) signifies time history of CR derived from DCPA' and TCPA' shown in Fig. 5.13(d) and Fig. 5.13(e). TCPA' became zero at 114.0 seconds and the maximum value of CR was 61.4.

The results of Case 2 in which the target ship comes from the port side of own ship were demonstrated in Fig. 5.14. As shown in Fig. 5.14(a), the target ship was located closest as distance of $4.06L$ at 118.8 seconds. As displayed in Fig. 5.14(b), The maximum rudder angle used in this case is $|\delta| = 13.1^\circ$ and the rudder angle reached zero after about 150 seconds. The heading angle returned to the desired heading angle $\psi = 0^\circ$ as shown in Fig. 5.14(c), because the risk of collision disappeared as CR ≤ 40. The time history of CR for Case 2 is observed in Fig. 5.14(g).

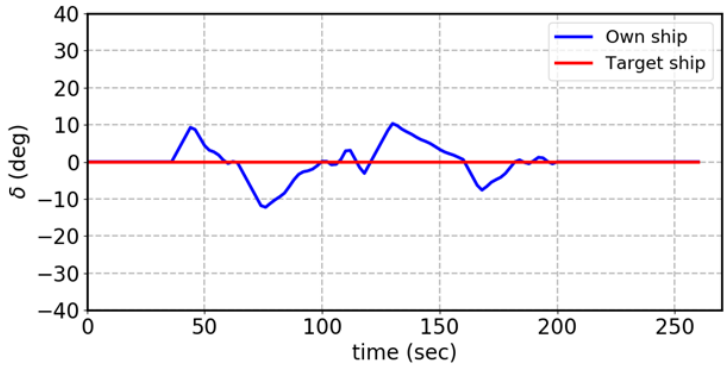
Fig. 5.15(a) depicted trajectory of the Case 3, the own ship moves forward and the target ship approaches the own ship from starboard side with relative angle 5°. Despite the fact that the target ship comes from starboard side, the own ship avoided the target ship by taking starboard avoidance according to international regulations. At this time, the two ships encountered each other at a nearest distance of $4.1L$. Maximum rudder angle used in Fig. 5.15(b) is between $\delta = -12.5^\circ$ and $\delta = 11.8^\circ$. Given that the heading angle reached to zero in Fig. 5.15(c), it can be known that the own ship finished the collision avoidance action and returned to the original course which is shown with the black dotted line in Fig. 5.15(a). The effectiveness of the decision maker devised by using DDPG in collision avoidance problem was also verified .

As seen from comparison between the results of DQN and DDPG, the closest distance between the two ships for DQN and DDPG had similar value which is approximately $4.0L$. If the value set to be deemed as a collision is changed, it seems that the own ship can travel closer or farther than in this algorithm. The maximum rudder angles in both method were also

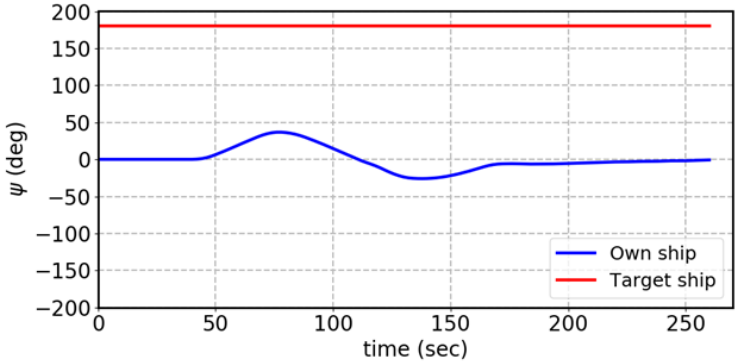
similar, but the distance from the original path, a cross track error, showed significant difference.



(a) Ship trajectories

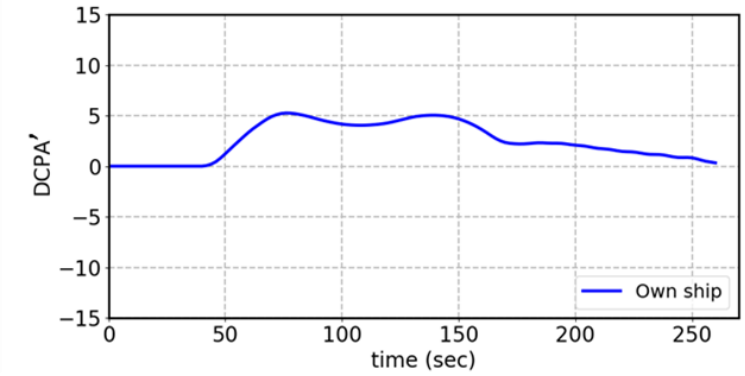


(b) Time history of rudder angle

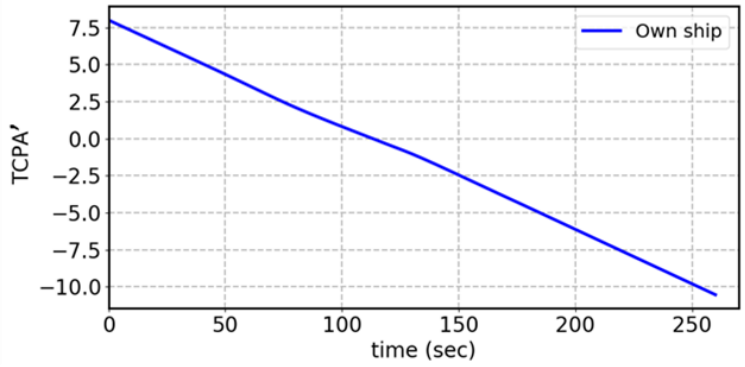


(c) Time history of heading angle

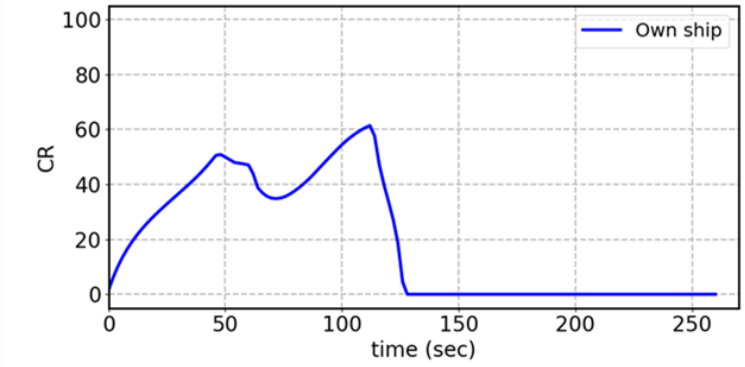
Fig. 5.13 Simulation results using DDPG of the Case 1



(d) Time history of DCPA'

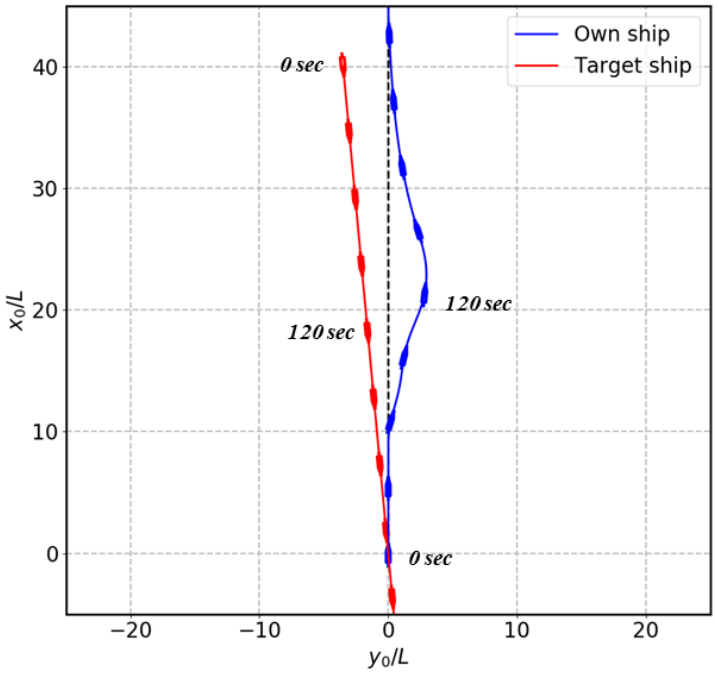


(e) Time history of TCPA'

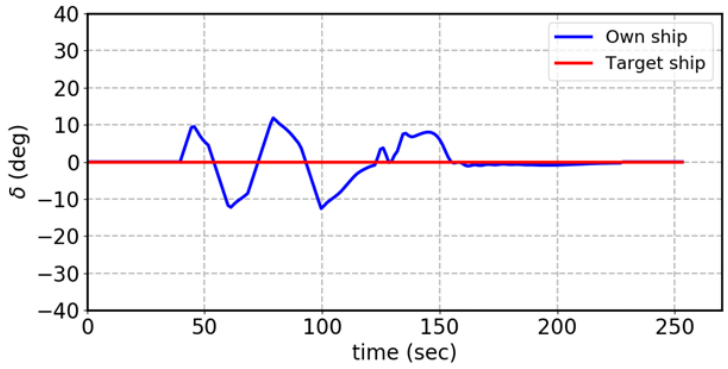


(f) Time history of CR

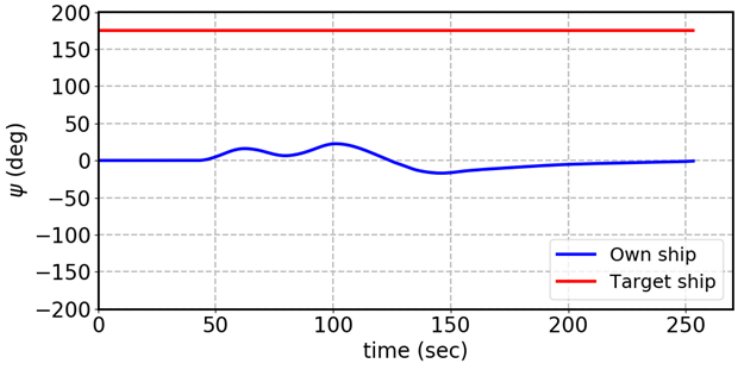
Fig. 5.13 Simulation results using DDPG of the Case 1



(a) Ship trajectories

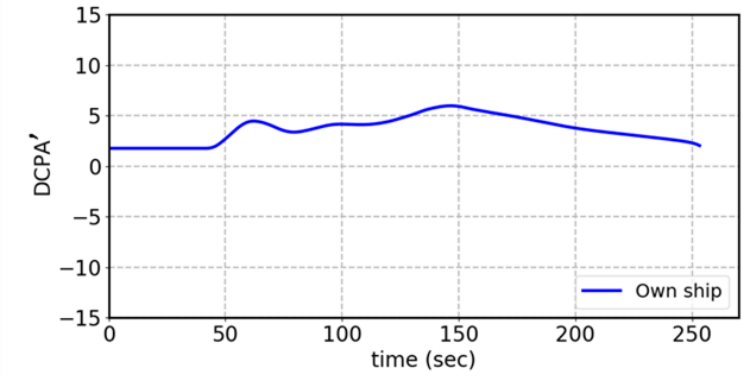


(b) Time history of rudder angle

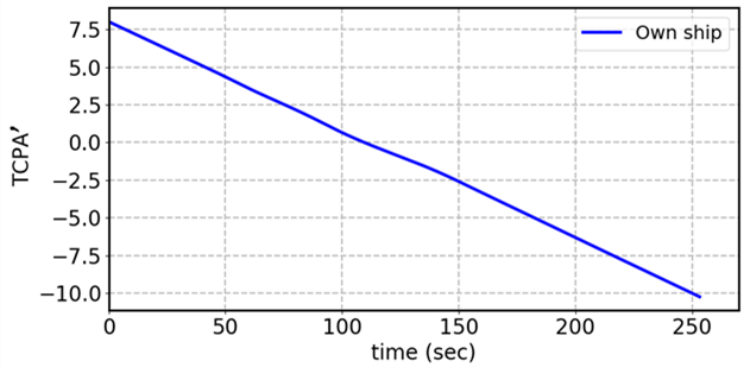


(c) Time history of heading angle

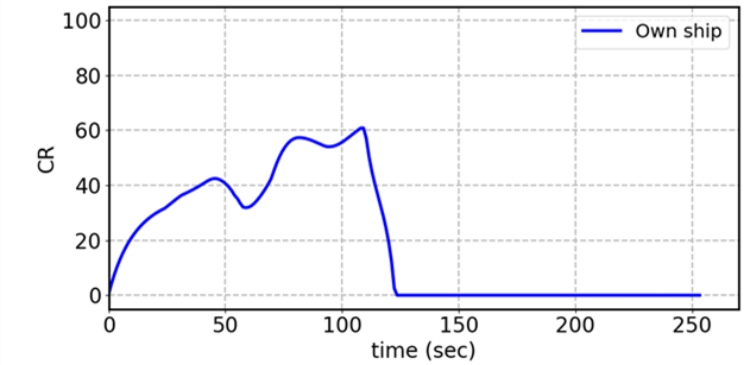
Fig. 5.14 Simulation results using DDPG of the Case 2



(d) Time history of DCPA'

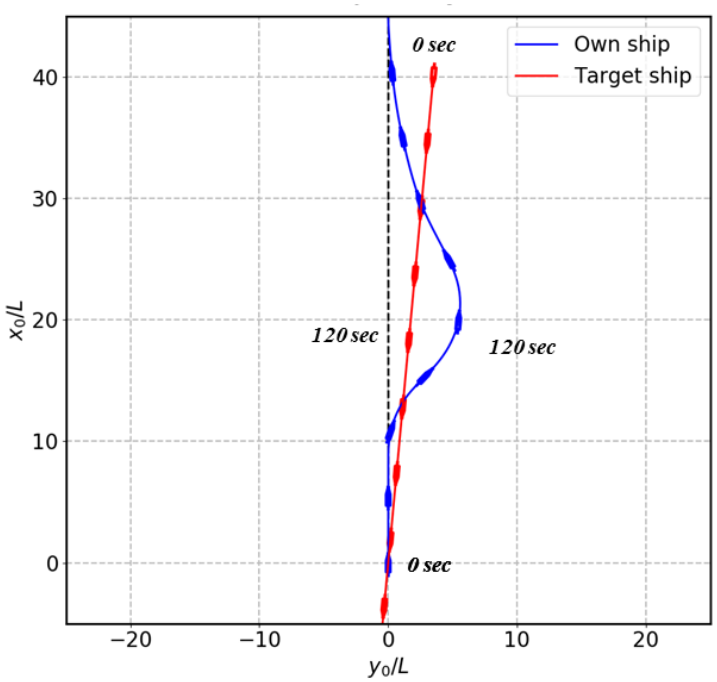


(e) Time history of TCPA'

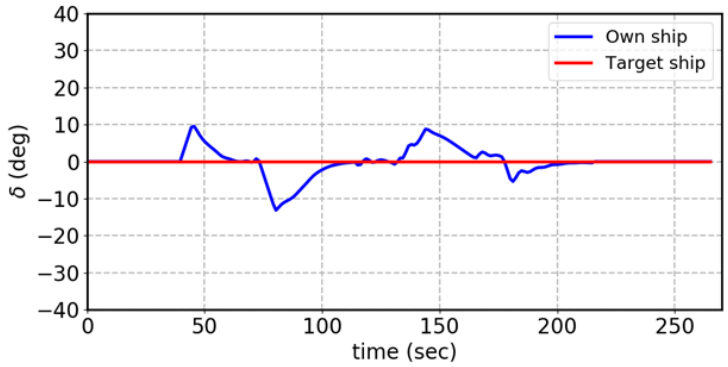


(f) Time history of CR

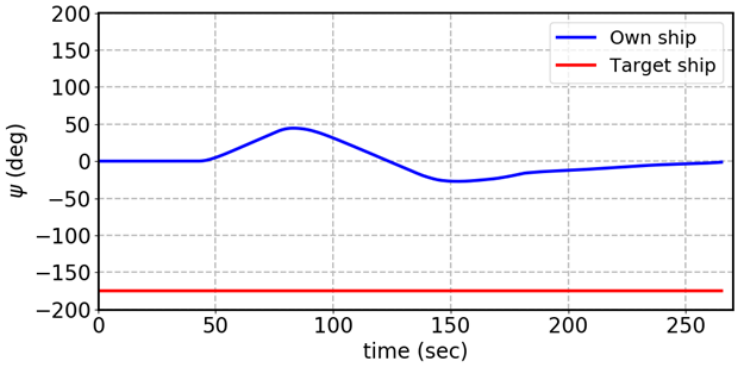
Fig. 5.14 Simulation results using DDPG of the Case 2



(a) Ship trajectories

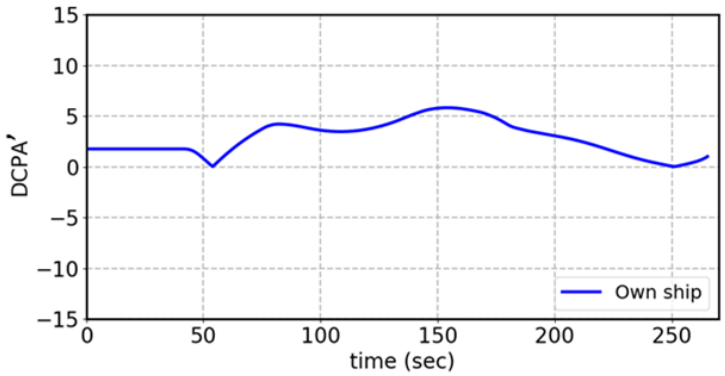


(b) Time history of rudder angle

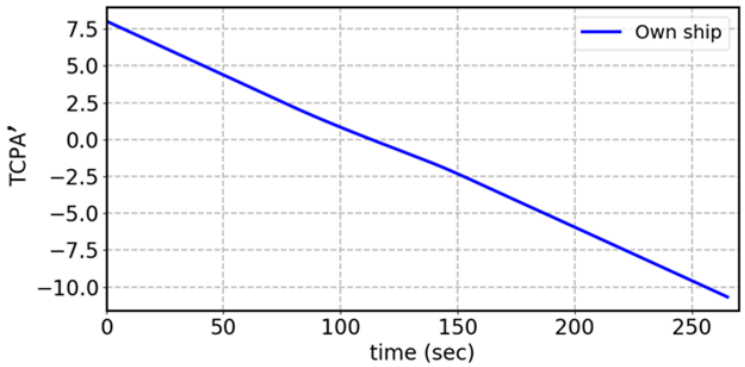


(c) Time history of heading angle

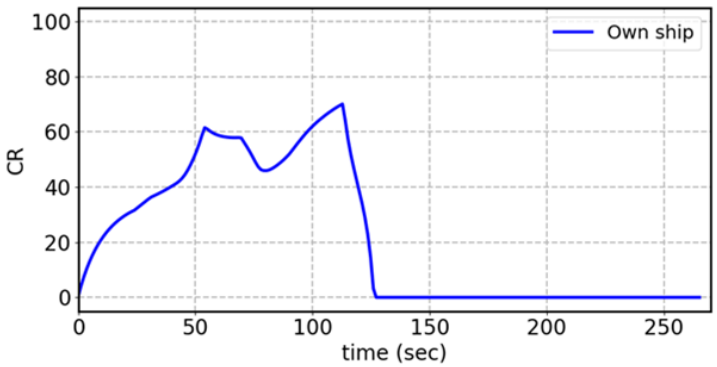
Fig. 5.15 Simulation results using DDPG of the Case 3



(d) Time history of DCPA'



(e) Time history of TCPA'



(f) Time history of CR

Fig. 5.15 DDPG Simulation results using DDPG of the Case 3

Therefore, the trajectories obtained for the same cases are expressed in one figure to facilitate comparison. The trajectories of three simulation cases using DQN and DDPG are displayed in Fig. 5.16 to Fig. 5.18. The blue lines are trajectories obtained by using DQN and the red lines are trajectories obtained through DDPG. The black lines indicate trajectories of the target ship and the black dotted lines show the original tracks of the own ship. Trajectories using DQN and DDPG in Case 1 are represented in Fig. 5.16. Focusing on the longest distance between x_0-axis and ship position in y_0-axis, is $7.00L$ in case of DQN and DDPG has $4.10L$. In Fig. 5.17 for Case 2 and Fig. 5.18 for Case 3, the maximum distances for DQN are $5.82L$ and $9.67L$, respectively. On the other hand, they are $2.97L$ and $5.58L$ for DDPG, individually. When CR, degree of collision risk, exceeds 40, the ship starts to take collision avoidance action. Therefore, the trajectories of DQN and DDPG show similar feature at the beginning of the voyage. However, since DQN, unlike DDPG, cannot select an action in a continuous action space, the own ship equipped with the algorithm developed by using DQN seems to be not able to take detailed action. It made the own ship to navigate further away from the current route in order to evade a target ship. Hence, it can be known that DDPG is more suitable for collision avoidance problem than DQN.

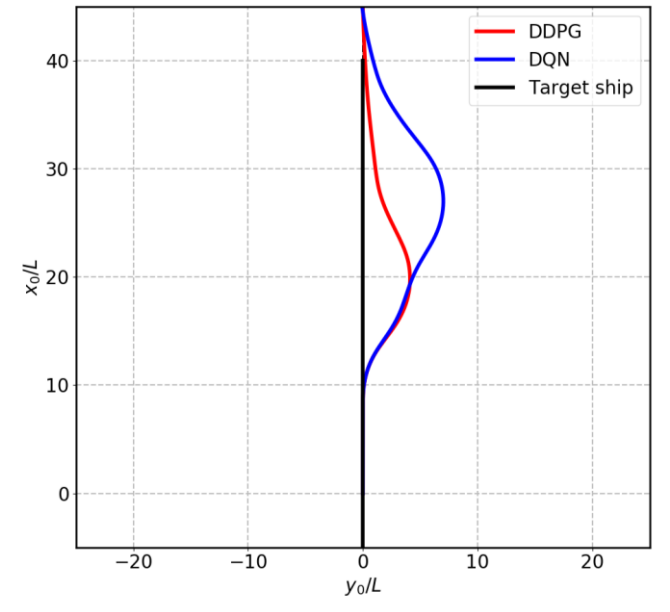


Fig. 5.16 Comparison of trajectories for Case 1

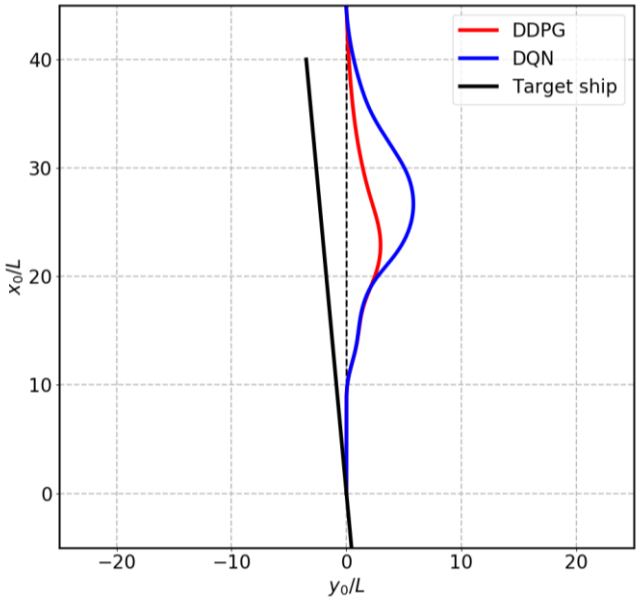


Fig. 5.17 Comparison of trajectories for Case 2

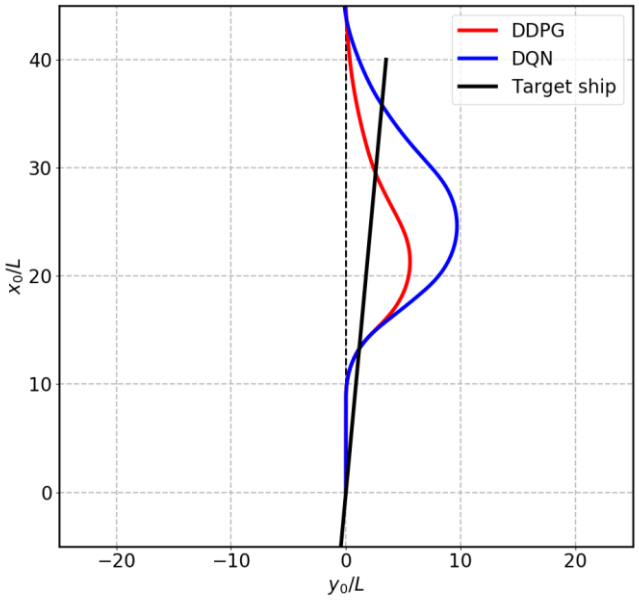


Fig. 5.18 Comparison of trajectories for Case 3

Finally, so as to further verify the proposed automatic collision avoidance algorithm in this chapter, several cases in which the target ship approaches the own ship from various positions with various heading angles were simulated. The simulations were also carried out through the trained model of DQN and DDPG, and the trajectories are shown in Fig. 5.19 and Fig. 5.20.

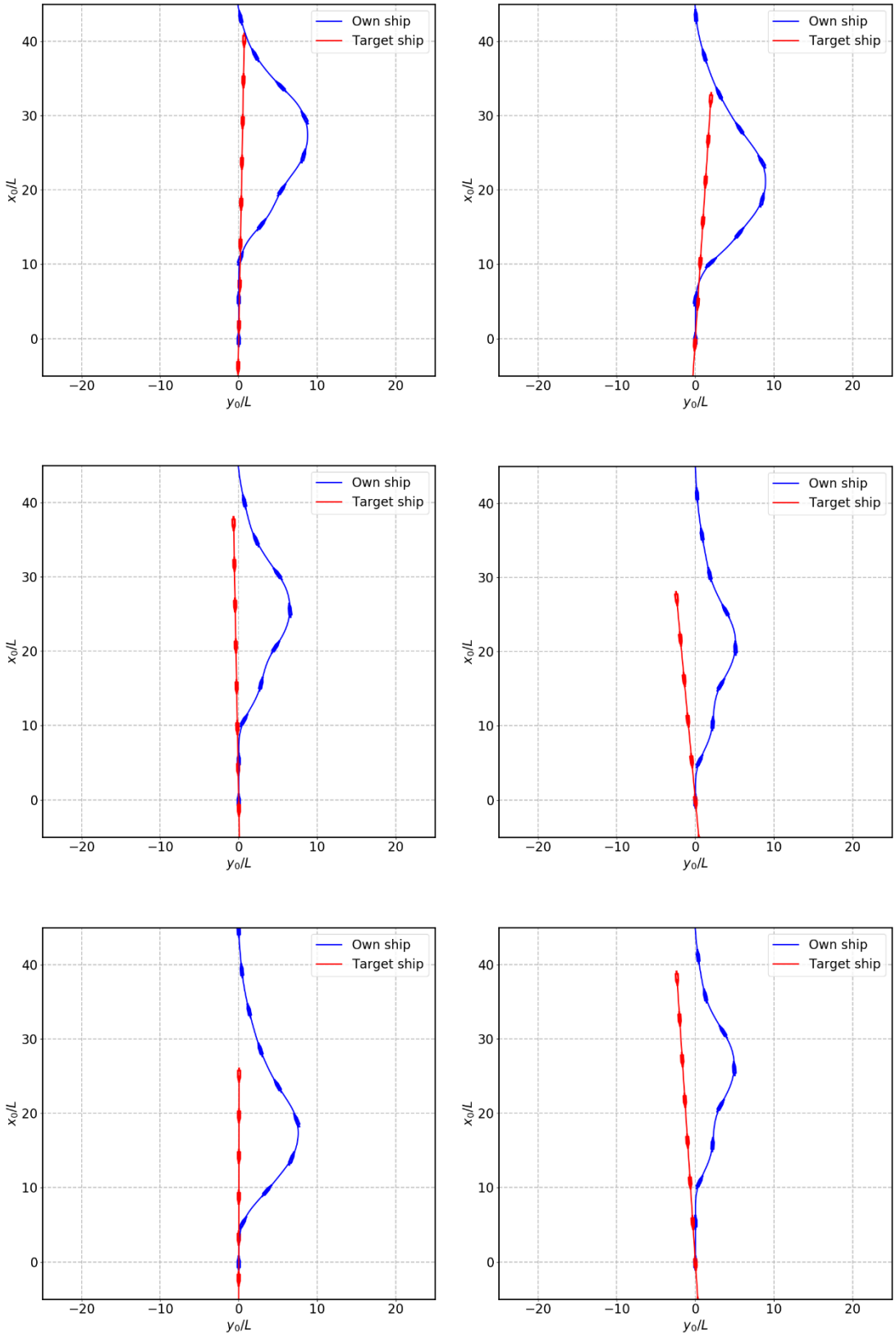


Fig. 5.19 The simulation results using DQN for random states of the target ship

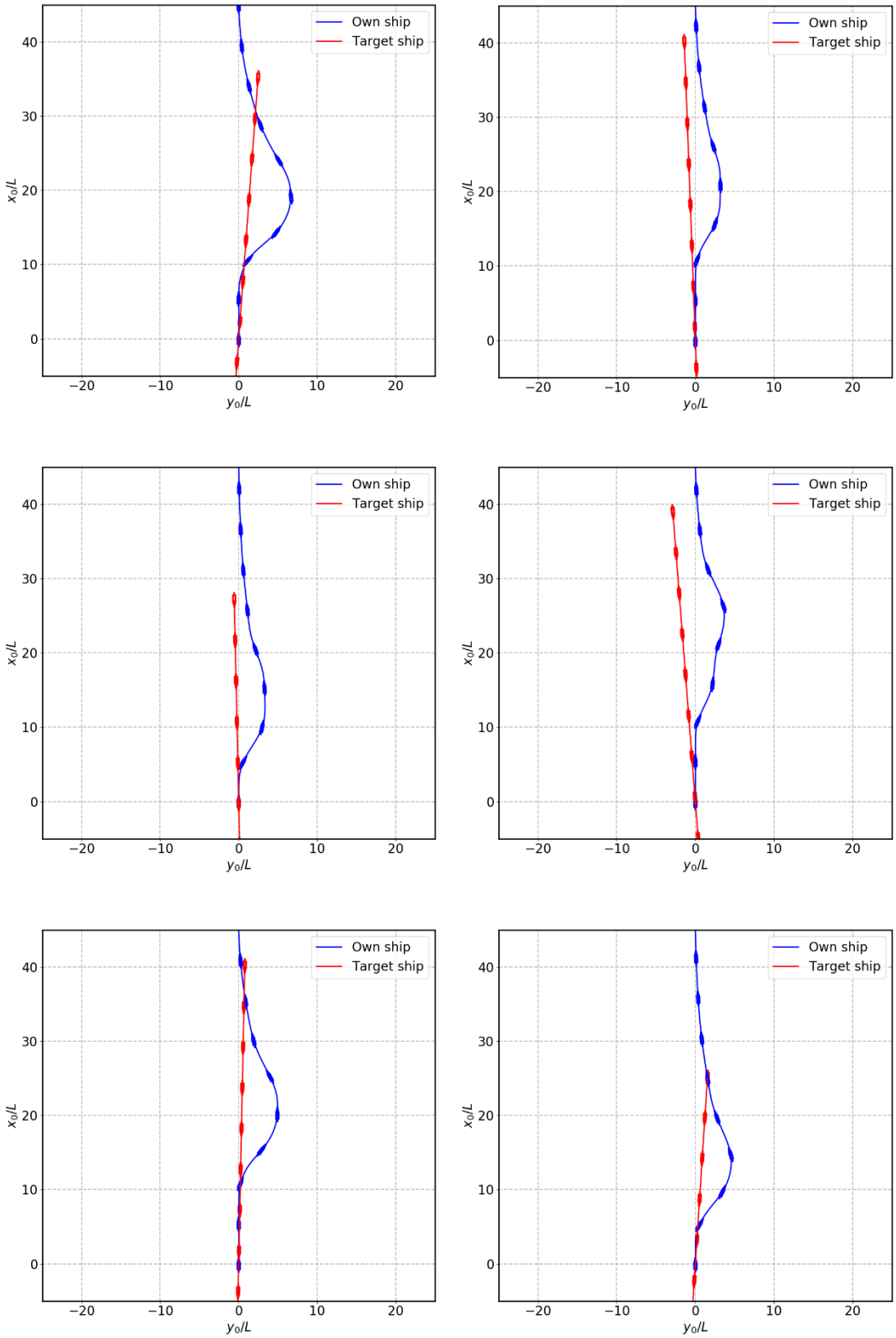


Fig. 5.20 The simulation results using DDPG for random states of the target ship

Through the figures indicating trajectories in Fig. 5.19 and Fig. 5.20, it is proven that the ship equiped with the automatic collision avoidacne algorithm is able to cope with a variety of encounter situations, and the algorithm based on DDPG can gives smooth evasive action to avoid collision.

5.5 Conclusions

In this chapter, in order to achieve a fully autonomous ship, an automatic collision avoidance algorithm along with the path following algorithm is designed utilizing reinforcement learning. Conclusions in this chapter can be drawn as follows:

- The automatic collision avoidance algorithm consits of three functions: collector and analyser, decision maker, and rudder controller was designed. In the collector and analyser, a controlled ship acquires data of approaching ships around her from AIS and radar. Using the obtained information, degree of collision risk is calculated in the controller and analyser.

- Through the decision maker designed by using two kinds of algorithm for deep reinforcement learning which are DQN and DDPG, the controlled ship can decide her collision avoidance action to be taken in order to evade crash with other ships or obstacles. The action space for collision avoidance problem is defined by course angles which changes within the range from -30˚ to 30˚. While DQN has a discrete action space, DDPG can have a continuous action space. The rudder controller introduced in Chapter 3 was applied to this algorithm.

- A collision avoidance problem was learned by DQN and DDPG. It was shown that training of DDPG finished earlier than that of DQN. Numerical simulations were carried out with the models trained by DQN and DDPG. The both methods could make the ship to safely avoid the approaching ships. However, the ship equipped with algorithm using DDPG can be operated closer to the original track comparing with the results of DQN. Therefore, it was confirmed that DDPG is better suited to solve the collision avoidance problem.

The suggested algorithm was performed in only head-on situation. Thus, verification is required in a variety encounter situations such as crossing and overtaking. In addition,

numerical simulations should be carried out in various environmental conditions including effects of wind, current and wave.

Chapter 6. Conclusions

In this thesis, the automatic path following algorithm and the automatic collision avoidance algorithm were developed in order to contribute to an accomplishment of the autonomous ship as an intelligent ship, and the algorithm has been verified through numerical simulations.

In Chapter 1, the background and purpose of this paper were introduced.

In Chapter 2, the mathematical model of ship motion was introduced. The main conclusions of this chapter are drawn as follows:

- Two kinds of coordinate systems are used to demonstrate the ship motion. The relationship between the earth-fixed coordinate system and the body-fixed coordinate system was introduced from the equations of manoeuvring motion.

- KVLCC2 was adopted as a subject ship and mathematical model based on the MMG model was selected in order to predict ship dynamic motion in numerical simulations. The forces and moment acting on a hull, the forces generated by a propeller and the forces and moments due to a rudder as parts of the MMG model were described respectively.

- Since the effects of wind and current were applied to numerical simulations, the relevant mathematical models were explained.

In Chapter 3, automatic path following algorithm applying fuzzy inference was proposed. Conclusions drawn from this work can be summarized as follows:

- The path following algorithm consists of two components, waypoints guidance system and rudder control system, were developed. In the waypoints guidance system, a desired track which a ship should follow was built by feeding waypoints positon data. Optimal timing to use rudder was derived from the waypoint switching system according to a course change angle owned by a target waypoint.

- Two kinds of path following algorithm were developed. The algorithm is distinguished depending on the performance of rudder control system which provides suitable rudder angle to change ship’s course. They were named as the basic path following algorim and the improved path following algorithm respectively.

- The basic path following algorithm has rudder control system taking account of three parmeters: heading error, a cross track error, and a yaw rate. On the other hand, rudder control system implemented in the improved path following algorithm employed four parmeters by adding ship speed to the existing three parameters.

- Numerical simulations were carried out assuming external disturbance such as wind and current in virtual siuation to verify the effectiveness of the developed algorithm. The simulation results showed that a ship equipped with the proposed systems could arrive at her destination with little overshoots of a heading error and a cross track error.

In Chapter 4, the proposed basic and improved path following algorithm composed of both waypoints switching system and rudder control system were verified through numerical simulations in realistic operation environment. The main conclusions of this chapter can be summarized as follows:

- Realistic external disturbances such as wind and current were reproduced using velocity vectors in real time based on actual measured data obtained from the official organization. In addition, the pre-planned track was designed with the position data of waypoints actually used by ship's operators.

- Numerical simulations were carried out to verify the effectiveness of the proposed algorithm, the basic path following algorithm and the improved path following algorithm, under realistic environmental conditions.

- In the simulations, information of wind and current obtained from real sea was applied depending on ship's location and time. As a result, the ship equipped with the proposed algorithms can travel on desired track using realistic rudder angle. However, it has been found that including the speed effect in path following algorithm makes the rudder action taken to keep track more stable.

In Chapter 5, in order to achieve a fully autonomous ship, an automatic collision avoidance algorithm along with the track keeping algorithm is designed utilizing reinforcement learning. Conclusions in this chapter can be drawn as follows:

- The automatic collision avoidance algorithm consisting of three functions: collector and analyser, decision maker, and rudder controller was designed. In the collector and

analyser, a controlled ship acquires data of approaching ships around her from AIS and radar. Using the obtained information, degree of collision risk is calculated in the controller and analyser.

- Through the decision maker designed by using two kinds of algorithm for deep reinforcement learning which are DQN and DDPG, the controlled ship can decide her collision avoidance action to be taken in order to evade crash with other ships or obstacles. The action space for collision avoidance problem is defined by course angles which changes within the range from -30˚ to 30˚. While DQN has a discrete action space, DDPG can have a continuous action space. The rudder controller introduced in Chapter 3 was applied to this algorithm.

- A collision avoidance problem was learned by DQN and DDPG. It was shown that training of DDPG finished earlier than that of DQN. Numerical simulations were carried out with the models trained by DQN and DDPG. The both methods could make the ship to safely avoid the approaching ships. However, the ship equipped with algorithm using DDPG can be operated closer to the original track comparing with the results of DQN. Therefore, it was confirmed that DDPG is better suited to solve the collision avoidance problem.

Although algorithm related on automatic path following and collision avoidance has been proposed in this research, some topics that need to be covered by future works still remain in order to achieve a fully autonomous ship. As for the path following algorithm, numerical simulations were conducted in realistic environmental situation, but it is necessary to verify in various environmental conditions considering shallow water effect, the influence of wave, and so on. The collision avoidance algorithm was performed in only head-on situation. Thus, verification is required in a variety encounter situations such as crossing and overtaking. The performance of two algorithm has been investigated through numerical simulation. However, experiment using model ship should be carried out for applying the developed algorithm to a real ship. In addition, if algorithm that is automatically able to determine waypoints is developed and then it combines with the algorithm devised in this study, it is expected that one step closer to completing a fully intelligent ship.

Acknowledgement

I would like to express my deep gratitude to my supervisor, Professor Yoshitaka Furukawa, for his invaluable guidance and encouragement on this thesis. He always helped me to move forward one-step in my research. He also provided me a great of various support during stay in Kyushu University. I will never forget his help and kindness.

I am grateful to Professor Takeshi Shinoda and Professor Satoru Yamaguchi at Department of Marine Systems Engineering in Kyushu University. They gave me valuable advice to complete this thesis. They carefully examined my thesis and notified me of what I did not realize.

I am deeply indebted to Professor Nam-Kyun Im at Division of Navigation Science in Mokpo National Maritime University. When I decided to study in Kyushu University, he gave me valuable and helpful advice as well as encouragement.

I express many thanks to all members of Marine Dynamics and Control Laboratory. I was able to adapt easily to life in Japan with their help. I want to express my gratitude to all my friends who came to Japan in order to see me. Especially, I am grateful to my friend Jisu Park who encouraged me whenever I had a hard time.

Finally, I would like to thank my father Young-Su Choe, my mother Misun Kim, my sister Boram Choe and my brother Jin-Gyeong Choe. I would like to share the accomplishment of this thesis with my family who supported my research with endless sacrifice for three years.

References

[1] Turing, A.M., 1950. Computing Machinery and Intelligence, MIND, Vol. 59, No. 236, pp. 443-460.

[2] Kongsberg, 2016. https://www.kongsberg.com/maritime/about-us/news-and-media/news-archive/2016/automated-ships-ltd-and-kongsberg-to-build-first-unmanned-and-fully-autonomous/

[3] Kongsberg, 2017. https://www.kongsberg.com/maritime/about-us/news-and-media/news-archive/2017/yara-and-kongsberg-enter-into-partnership-to-build-worlds-first-autonomous-and/

[4] Rolls-Royce, 2017. https://www.rolls-royce.com/media/press-releases/2017/03-10-2017-rr-joins-forces-with-google-cloud-to-help-make-autonomous-ships-a-reality.aspx

[5] Japan Ship Technology Research Association, 2017. https://www.jstra.jp/conference/2017/08/post-1.html

[6] IMO, 2017. http://www.imo.org/en/MediaCentre/MeetingSummaries/MSC/Pages/MSC-98th-session.aspx

[7] IMO, 2018. http://www.imo.org/en/MediaCentre/PressBriefings/Pages/08-MSC-99-MASS-scoping.aspx

[8] IMO, 2018. http://www.imo.org/en/MediaCentre/MeetingSummaries/MSC/Pages/MSC-99th-session.aspx

[9] Sperry, E., 1922, Automatic Steering, 13th Meeting of the Society of Naval Architects and Marine Engineers, pp. 53-61.

[10] Minorsky, N., 1922. Directional Stability and Automatically Steered Bodies, Naval Engineers Journal, Vol. 34, Issue 2, pp. 280-309.

[11] Kallstorom, C.G., Astrom, K.J., 1981. Experience of system identification applied to ship steering, Automatica, Vol.17, pp. 187-198.

[12] Juntong, Q., Yan, P., He, W., Jianda, H., 2007. Design and Implement of a Trimaran Unmanned Surface Vehicle System, International Conference on Information Acquisition (ICIA), Seogwipo, South Korea, pp. 361-365.

[13] Kumar, V., Nakara, B.C., Mittal, A.P., 2011. A Review on Classical and Fuzzy PID controllers, International Journal of Intelligent Control and Systems, Vol. 16, No. 3, pp. 170-181.

[14] Breivik, M., Hovstin, V.E., Fossen, T.I., 2008. Straight-Line Target Tracking for Unmanned Surface Vehicles, Journal of Modeling, Identification and Control, Vol. 29, No. 4, pp. 131-149.

[15] Jin, C., Jianquiang, Y., Dongbin, Z., 2006. A New Fuzzy Autopilot for Way-point Tracking Control of Ships, IEEE International Conference on Fuzzy Systems, pp. 451-456

[16] Fozzen, T.I., Brivik, M., Skjetne, R., 2003. Line-of-sight Path Following of Underactuated Marine Craft, IFAC Proceedings, Girona, Spain, Vol. 36, Issue 21, pp. 211-216.

[17] Velagic, J., Vukic Z., Omeridic E., 2003. Adaptive Fuzzy Ship Autopilot for Track-keeping, Control Engineering Practice, Vol. 11, Issue 4, pp. 433-443.

[18] Mcgookin, E.W., Murray-Simith, D.J., Li, Y., Fossen, T.I., 2000. Ship Steering Control System Optimization using Genetic Algorithms, Control Engineering Practice, Vol. 8, Issue 4, pp. 429-443.

[19] Fuji, Y., Tanaka, K., 1971. Traffic capacity, Journal of Navigation, Vol. 24, pp. 543-552.

[20] Tam, C., Bucknall, R., 2010. Path-planning Algorithm for Ships in Closerange Encounters, Journal of Marine Science and Technology, Vol. 15, No. 4, pp. 395-407.

[21] Tom, C., Bucknall, R., 2010. Collision Risk Assessment for Ships, Journal of Marine Science and Technology, Vol. 15, No. 3, pp. 257-270.

[22] Wang, N., Meng, X., Xu, Q., Wang, Z., 2009. A Unified Analytical Framework for Ship Domains, Journal of Navigation, Vol. 62, No. 4, pp. 643-655.

[23] Goodwin, E.M., 1975. A Statistical Study of Ship Domains, Journal of Navigation, Vol. 28, pp. 329-341.

[24] Pietrzykowski, Z., Uriasz, J., 2009. The Ships Domain – A Criterion of Navigational Safety Assessment in an Open Sea Area, Journal of Navigation, Vol. 62, No. 1, pp. 93-108.

[25] Iijima Y., Hagiwara, H., Kasai, H., 1991. Results of Collision Avoidance Manouvre Experiments Using a Knowledge-based Autonomous Piloting System, Journal of Navigation, Vol. 44, No. 2, pp. 194-204

[26] Kijima, K., Furukawa, Y., 2002. Development of Collision Avoidance Algorithm Using Fuzzy Inference. Proceedings of ISOPE Pacific/Asia Offshore Mechanics Symposium, Daejeon, South Korea, pp. 123-130.

[27] Kao, S.K., Lee, K.T., Chang, K.Y., Ko, M.D., 2007. A Fuzzy Logic Method for Collision Avoidance in Vessel Traffic Service, Journal of Navigation, Vol. 60, No. 1, pp. 17-31.

[28] Son, N.S., 2009. On the Development of an Autonomous Collision Avoidance System for Ship using Changeable Action Space, PhD thesis, Kyushu University, Japan.

[29] Cummings, M.L., Buchin, M., Carrigan, G., Donmez, B., 2010. Supporting Intelligent and Trustworthy Maritime Path Planning, International Journal of Human-Computer Studies, Vol. 68, No. 10, pp. 616-626.

[30] Blaich, M., Rosenfelder, M., Schuster, M., Bittel, O., Reuter, J., 2012. Fast Grid Based Collision Avoidance for Vessels Using A* Search Algorithm, Proceedings of International Conference on Methods and Models in Automation and Robotics, Miezyzdroje, Poland, pp. 385-390.

[31] Tuyen, L.P., Layek, A., Vien, N.A., Chung, T.C., 2017. Deep Reinforcement Learning Algorithms for Steering an Underacuated Ship, IEE International Conference on Multisensor Fusion and Integration for Intelligent System, Daegu, South Korea, pp. 602-607.

[32] Kim, D.H., Lee, S.W., Nam, J.H., Furukawa, Y., 2019. Determination of Ship Collision Avoidance Path using Deep Deterministic Policy Gradient Algorithm, Journal of the Society of Naval Architects of Korea, Vol. 56, No. 1, pp. 58-65.

[33] Nomoto, K., Tauchi, T., Honda, K., Hirano, S., 1957. On the Steering Qualities of Ships, International Shipbuilding Progress, Vol. 4, No. 35, pp. 354-370.

[34] Abkowitx, M.A., 1980. Measurement of Hydrodynamic Characteristics from Ship Maneuvering Trials by System Identification. SNAME Transactions 88, pp. 283-318.

[35] Fujino, M., Kijima, K., Hamamoto, M., 1990. Present State of the Prediction Method of Ship Manoeuvrability, MARSIM & ICSM 90, pp. 7-17.

[36] Hamamoto, M., MMG report II Bulletin of the Society of Naval Architects of Japan, No. 577, pp. 322-329.

[37] Ogawa, A., Koyama, T., Kijima, K., 1977. MMG report-I, On the Mathematical Model of Ship Manoeuvring, Bulletin of the Society of Naval Architects of Japan, Vol. 575, pp. 22-28.

[38] Kijima, K., Katsuno, T., Nakiri, Y., Furukawa, Y., 1990. On the Manoeuvring Performance of a Ship with the Parameter of Loading Condition, Journal of the Society of Naval Architects of Japan. Vol. 1990, Issue 168, pp. 141-148.

[39] Matsumoto, K., Suemitsu, K., 1980, The Prediction of Manoeuvring Performances by Captive Model Tests, Journal of the Kansai Society of Naval Architects, No. 176

[40] Fujiwara, T., Ueno, M., Nimura, T., 1998. Estimation of Wind Forces and Moments Acting on Ships, Journal of the Society of Naval Architects of Japan, Vol. 1998, Issue 183, pp. 77-90.

[41] Li, Z., Sun, J., Oh, S., 2009. Design, Analysis and Experimental Validation of a Robust Nonlinear Pathfollowing Controller for Marine Surface Vessels, Automatica, Vol. 45, pp. 1649-1958.

[42] Ahmed, Y.A., Hasegawa, K., 2016. Fuzzy Reasoned Waypoint Controller for Automatic Ship Guidance, 10th IFAC Conference on Control Applications in Marine System, Vol. 49, Issue 23, pp. 604-609.

[43] Wang, J.Q., Wang, T., 2019. Path Following of a Surface Ship Sailing in Restricted Waters under Wind Effect using Robust H^{∞} Guaranteed Cost Control, International Journal of Naval Architecture and Ocean Engineering, Vol. 11, Issue 1, pp. 606-623.

[44] Lekkas, A., Fossen, T.I., 2013. Line-of-sight Guidance for Path Following of Marine Vehicles, in Advanced in Marine Robotics, Saarbrcken (Ed.), Lambert Academic Publishing, Germany.

[45] Breivik, M., Fossen, T.I., 2009. Guidance Laws for Autonomous Underwater Vehicles, in: Intelligent Underwater Vehicles, Inzartsev, A.V. (Ed.), I-Tech Education and Publishing, Vienna.

[46] Velasco, F.J., Revestido, E., Lopez, E., Moyono, E., Casado, M., 2008. Autopilot and Track-keeping Simulation of an Autonomous In-scale Fast-ferry Model, 12th WSEAS International Conference on Systems, pp. 649-654.

[47] Guo, W., Wang, S., Dun, W., 2015. The Design of a Control System for an Unmanned Surface Vehicle, Open Automation and Control Systems Journal, Vol. 7, pp. 150-156.

[48] Choe, B.R., Furukawa, Y., 2018. Development of Track Keeping Algorithm using Fuzzy Inference, Proceeding of 20th International Ocean and Polar Engineering Conference, Sapporo, Japan, pp. 731-737.

[49] Fossen, T.I., 1999. Recent Developments in Ship Control Systems, World Superyacht Review, Fossen, T.I. (Ed.), Sterling Publication Limited, London.

[50] Azzeri, M.N., Adnan, F.A., Zain, M.Z Md., 2015. Review of Course Keeping Control System for Unmanned Surface Vehicle, Jurnal Teknologi, Vol. 74, No. 5, pp. 11-20.

[51] Vukic, Z., Velagic, J., 1999. Comparative Analysis of Mamdani and Sugeno Type Fuzzy Autopilots for Ships, Proceeding of the 15th European Control Conference, Karlsruhe, Germany, Paper No. F0505.

[52] Wang, D., Huang, J., 2005. Neural Network-Based Adaptive Dynamic Surface Control for a Class of Uncertain Nonlinear Systems in Strict-Feedback Form, IEEE Transactions on Neural Networks, Vol. 16, No. 1, pp. 195-202.

[53] Chen, W., Jiao, L., Li, R., Li, J., 2010. Adaptive Backstepping Fuzzy Control for Nonlinearly Parameterized Systems with Periodic Disturbances, IEEE Transactions on Fuzzy Systems, Vol. 18, Issue 4, pp. 674-685.

[54] Liu, Z., Jin, H., Grimble M.J., Katebi, R., 2016. Ship Forward Speed Loss Minimization using Nonlinear Course Keeping and Roll Motion Controllers, Ocean Engineering, Vol. 113, pp. 201-207.

[55] Zadeh, L.A., 1965, Fuzzy Set, Information and Control, Vol. 8, pp. 338-353

[56] Mamdani, E.H., Assilian, S., 1975. An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller, International Journal of Man-Machine Studies, Vol. 7, No. 1, pp. 1-13.

[57] Sugeno, M., 1985. Industrial Applications of Fuzzy Control, Sugeno, M. (Ed.), Elsevier Science Inc., New York.

[58] IMO resolution MSC.74(69), 1998. Adoption of New and Amended for Performance Standards, International Marine Organization (IMO).

[59] IMO resolution MSC.137(76), 2002. Standards for Ship Manoeuvrability, International Marine Organization (IMO).

[60] IMO resolution MEPC.232(65) (2013). 2013 Interim Guidelines for Determing Minimum Propulsion Power to Maintain the Manoeuvrablity of Ships in Adverse Conditions, International Maritime Organization (IMO).

[61] National Oceanic and Atmospheric Administration (NOAA), ftp://polar.ncep.noaa.gov/pub/history/waves/

[62] Japan Transport Safety Board, http://www.mlit.go.jp/jtsb/statistics_mar.html

[63] Davis, P.V., Dove, M.J., Stockel, C.T., 1980. A Computer Simulation of Marine Traffic using Domains and Arenas, Journal of Navigation, Vol. 33, Issue 2, pp. 215-222.

[64] Coldwell, T.G., 1983. Marine Traffic Behaviour in Restricted Waters, Journal of Navigation, Vol. 36, Issue 3, pp. 431-444.

[65] Zhu, X., Xu, H., Lin, J., 2001. Domain and its Model Based On Neural Networks. Journal of Navigation, Vol. 54, Issue 1, pp. 97-103.

[66] Kijima, K., Furukawa, Y., 2003. Automatic Collision Avoidance System using the Concept of Blocking Area, International Federation of Automatic Control Proceedings, Girona, Spain, Vol. 36, No. 21, pp. 223-228.

[67] Hasegawa, K., Kouzuki, A., 1987. Automatic Collision Avoidance System for Ship using Fuzzy Control, Journal of the Kansai Society of Naval Architects, pp. 1-10

[68] Tu, E., Zhang, G., Rachmawati, L., Rajabally. E., 2018. Exploiting AIS Data for Intelligent Maritime Navigation: A Comprehensive Survey From Data to Methodology, IEEE Transactions on Intelligent Transportation Systems, Vol. 19, No. 5, pp. 1559-1582

[69] Mnih. V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves. A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D., 2015. Humanlevel Control through Deep Reinforcement Learning, Nature, Vol. 518, pp. 529-533.

[70] Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D., 2016. Continuous Control with Deep Reinforcement Learning, arXiv: 1509.02971, pp. 1-14.

[71] Schulman, J., Moritz, P., Levine, S., Jordan, M., Abbeel, P., 2016. High-Dimensional Continuous Control Using Generalized Advantage Estimation, arXiv: 1506.02438, pp. 1-9.

[72] Bellman, R.E., 1957. A Markovian Decision Process, Journal of Mathematics and Mechanics, pp. 679-684.

[73] Watkins, C.J.C.H., 1989, Learning from Delayed Rewards, PhD thesis, King's College, England.

[74] Divyam R., 2017. Deep Reinforcement Learning for Bipedal Robots, Master thesis, Delft University of Technology, Nederland.

[75] Silver, D., Lever, G., Heess, N., 2014. Deterministic Policy Gradient Algorithms, Proceedings of the 31st International Conference on Machine Learning (ICML-14), Beijing, China, pp. 387-395.

[76] Degris, T., White, M., Sutton, R.S., 2012. Off-policy Actor-critic, arXiv: 1205.4839, pp. 457-464.

[77] Uhlenbcdk, G.E., Ornstein, L.S., 1930. On the Theory of the Brownian Motion, Physical Review Journals Archive, Vol. 36, No. 5, pp. 823-841.

[78] IMO, 1972. Convention on the International Regulations for Preventing Collisions at Sea.

[79] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D.G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zheng, X., 2016. Tensorflow: A System for Large-scale Machine Learning, arXiv: 1603.04467, pp. 265-283.