공학박사 학위논문

Simulation Method to Support Autonomous Navigation and Installation Operation of an Offshore Support Vessel

해양 작업 지원선의 자율 운항 및 설치 작업 지원을 위한 시뮬레이션 방법

2019 년 2 월

서울대학교 대학원

조선해양공학과

Luman Zhao

Simulation Method to Support Autonomous Navigation and Installation Operation of an Offshore Support Vessel

해양 작업 지원선의 자율 운항 및 설치 작업 지원을 위한 시뮬레이션 방법

지도교수 노 명 일

이 논문을 공학박사 학위논문으로 제출함

2019 년 2 월

서울대학교 대학원

조선해양공학과

Luman Zhao

Luman Zhao의 공학박사 학위논문을 인준함

2019 년 2 월

위 원 장	장 범 선 교 수 님	(인)
부위원장	노 명 일 교 수 님	(인)
위 원	윤 현 규 교 수 님	(인)
위 원	하 솔 교 수 님	(인)
위 원	구 남 국 교 수 님	(인)

Abstract

Simulation Method to Support Autonomous Navigation and Installation Operation of an Offshore Support Vessel

Luman Zhao
Department of Naval Architecture and Ocean Engineering
The Graduate School
Seoul National University

Autonomous ships have gained a huge amount of interest in recent years, like their counterparts on land–autonomous cars, because of their potential to significantly lower the cost of operation, attract seagoing professionals and increase transportation safety. Technologies developed for the autonomous ships have potential to notably reduce maritime accidents where 75% cases can be attributed to human error and a significant proportion of these are caused by fatigue and attention deficit. However, developing a high-level autonomous system which can operate in an unstructured and unpredictable environment is still a challenging task. When the autonomous ships are operating in the congested waterway with other manned or unmanned vessels, the collision avoidance algorithm is the crucial point in keeping the safety of both the own ship and any encountered ships. Instead of developing new traffic rules for the autonomous ships to avoid collisions with each other, autonomous ships are expected to follow the existing guidelines based on the International Regulations for Preventing Collisions at Sea (COLREGs). Furthermore, when using the crane on

the autonomous ship to transfer and install subsea equipment to the seabed, the heave and swaying phenomenon of the subsea equipment at the end of flexible wire ropes makes its positioning at an exact position is very difficult. As a result, an Anti-Motion Control (AMC) system for the crane is necessary to ensure the successful installation operation.

The autonomous ship is highly relying on the effectiveness of autonomous systems such as autonomous path following system, collision avoidance system, crane control system and so on. During the previous two decades, considerable attention has been paid to develop robust autonomous systems. However, several are facing challenges and it is worthwhile devoting much effort to this. First of all, the development and testing of the proposed control algorithms should be adapted across a variety of environmental conditions including wave, wind, and current. This is one of the challenges of this work aimed at creating an autonomous path following and collision avoidance system in the ship. Secondly, the collision avoidance system has to comply with the regulations and rules in developing an autonomous ship. Thirdly, AMC system with anti-sway abilities for a knuckle boom crane remains problems regarding its under-actuated mechanism. At last, the performance of the control system should be evaluated in advance of the operation to perform its function successfully. In particular, such performance analysis is often very costly and time-consuming, and realistic conditions are typically impossible to establish in a testing environment.

Consequently, to address these issues, we proposed a simulation framework with the following scenarios, which including the autonomous navigation scenario and crane operation scenario. The research object of this study is an autonomous offshore support vessel (OSV), which provides support services to offshore oil and gas field development such as offshore drilling, pipe laying, and oil producing assets (production platforms and FPSOs) utilized in EP (Exploration Production) activities.

Assume that the autonomous OSV confronts an urgent mission under the harsh environmental conditions: on the way to an imperative offshore construction site, the autonomous OSV has to avoid target ships while following a predefined path. When arriving at the construction site, it starts to install a piece of subsea equipment on the seabed. So what technologies are needed, what should be invested for ensuring the autonomous OSV could robustly kilometers from shore, and how can an autonomous OSV be made at least as safe as the conventional ship. In this dissertation, we focus on the above critical activities for answering the above questions.

In the general context of the autonomous navigation and crane control problem, the objective of this dissertation is thus fivefold:

- Developing a COLREGs-compliant collision avoidance system.
- Building a robust path following and collision avoidance system which can handle unknown and complicated environment.
- Investigating an efficient multi-ship collision avoidance method enable it easy to extend.
- Proposing a hardware-in-the-loop simulation environment for AHC system.
- Solving the anti-sway problem of the knuckle boom crane on an autonomous OSV.

First of all, we propose a novel deep reinforcement learning (RL) algorithm to achieve effective and efficient capabilities of the path following and collision avoidance system. To perform and verify the proposed algorithm, we conducted simulations for an autonomous ship under unknown environmental disturbance

to adjust its heading in real-time. A three-degree-of-freedom dynamic model of the autonomous ship was developed, and the Line-of-sight (LOS) guidance system was used to converge the autonomous ship to follow the predefined path. Then, a proximal policy optimization (PPO) algorithm was implemented on the problem. By applying the advanced deep RL method, in which the autonomous OSV learns the best behavior through repeated trials to determine a safe and economical avoidance behavior in various circumstances. The simulation results showed that the proposed algorithm has the capabilities to guarantee collision avoidance of moving encountered ships while ensuring following a predefined path. Also, the algorithm demonstrated that it could manage complex scenarios with various encountered ships in compliance with COLREGs and have the excellent adaptability to the unknown, sophisticated environment.

Next, the AMC system includes Anti-Heave Control (AHC) and Anti-Sway Control (ASC), which is applied to install a subsea equipment in regular and irregular for performance analysis. We used the proportional–integral–derivative (PID) control method and the sliding mode control method respectively to achieve the control objective. The simulation results show that heave and sway motion could be significantly reduced by the proposed control methods during the construction. Moreover, to evaluate the proposed control system, we have constructed the HILS environment for the AHC system, then conducted a performance analysis of it. The simulation results show the AHC system could be evaluated effectively within the HILS environment. We can conclude that the proposed or adopted methods solve the problems issued in autonomous system design.

Keywords: Offshore support vessel; Autonomous navigation; Collision avoidance; Path following; Deep reinforcement learning (DRL); Multibody dynamics; Hardware-in-the-Loop Simulation (HILS); Anti-Heave Control (AHC) system;

Anti-Sway Control (ASC) system; Sliding mode control

Student Number: 2014-31430

Contents

Abstract **i**

Contents **x**

List of Tables **xii**

List of Figures **xx**

Chapter 1 Introduction **1**

1.1 Background and Motivation 1
1.2 Requirements for Autonomous Operation 5
1.2.1 Path Following for Autonomous Ship 5
1.2.2 Collision Avoidance for Autonomous Ship 5
1.2.3 Anti-Motion Control System for Autonomous Ship 6
1.3 Related Work 9
1.3.1 Related Work for Path Following System 9
1.3.2 Related Work for Collision Avoidance System 9
1.3.3 Related Work for Anti-Heave Control System 13
1.3.4 Related Work for Anti-Sway Control System 14
1.4 Configuration of Simulation Framework 16
1.4.1 Application Layer 16
1.4.2 Autonomous Ship Design Layer 17
1.4.3 General Technique Layer 17

1.5 Contributions (Originality) 19

Chapter 2 Theoretical Backgrounds **20**

2.1 Maneuvering Model for Autonomous Ship 20

2.1.1 Kinematic Equation for Autonomous Ship 20

2.1.2 Kinetic Equation for Autonomous Ship 21

2.2 Multibody Dynamics Model for Knuckle Boom Crane of Autonomous Ship 25

2.2.1 Embedding Techniques 25

2.3 Control System Design 31

2.3.1 Proportional-Integral-Derivative (PID) Control 31

2.3.2 Sliding Mode Control 31

2.4 Deep Reinforcement Learning Algorithm 34

2.4.1 Value Based Learning Method 36

2.4.2 Policy Based Learning Method 37

2.4.3 Actor-Critic Method 41

2.5 Hardware-in-the-Loop Simulation 43

2.5.1 Integrated Simulation Method 43

Chapter 3 Path Following Method for Autonomous OSV **46**

3.1 Guidance System 46

3.1.1 Line-of-sight Guidance System 46

3.2 Deep Reinforcement Learning for Path Following System 50

3.2.1 Deep Reinforcement Learning Setup 50

3.2.2 Neural Network Architecture 56

3.2.3 Training Process 58

3.3 Implementation and Simulation Result 62

3.3.1 Implementation for Path Following System 62

3.3.2 Simulation Result 65
3.4 Comparison Results 83
3.4.1 Comparison Result of PPO with PID 83
3.4.2 Comparison Result of PPO with Deep Q-Network (DQN) 87

Chapter 4 Collision Avoidance Method for Autonomous OSV 89
4.1 Deep Reinforcement Learning for Collision Avoidance System . . 89
4.1.1 Deep Reinforcement Learning Setup 89
4.1.2 Neural Network Architecture 93
4.1.3 Training Process 94
4.2 Implementation and Simulation Result 95
4.2.1 Implementation for Collision Avoidance System 95
4.2.2 Simulation Result 100
4.3 Implementation and Simulation Result for Multi-ship Collision Avoidance Method 107
4.3.1 Limitations of Multi-ship Collision Avoidance Method - 1 107
4.3.2 Limitations of Multi-ship Collision Avoidance Method - 2 108
4.3.3 Implementation of Multi-ship Collision Avoidance Method 110
4.3.4 Simulation Result of Multi-ship Collision Avoidance Method 118

Chapter 5 Anti-Motion Control Method for Knuckle Boom Crane 129
5.1 Configuration of HILS for Anti-Heave Control System 129
5.1.1 Virtual Mechanical System 132
5.1.2 Virtual Sensor and Actuator 138
5.1.3 Control System Design 141
5.1.4 Integrated Simulation Interface 142
5.2 Implementation and Simulation Result of HILS for Anti-Heave Control System 145

5.2.1 Implementation of HILS for Anti-Heave Control System . 145
5.2.2 Simulation Result of HILS for Anti-Heave Control System 146
5.3 Validation of HILS for Anti-Heave Control System 159
5.3.1 Hardware Setup 159
5.3.2 Comparison Result 161
5.4 Configuration of Anti-Sway Control System 162
5.4.1 Mechanical System for Knuckle Boom Crane 162
5.4.2 Anti-Sway Control System Design 165
5.4.3 Implementation and Simulation Result of Anti-Sway Control 168

Chapter 6 Conclusions and Future Works **176**

Bibliography **178**

Chapter A Appendix **186**

국문초록 **188**

List of Tables

Table 3.1 State observation in path following system 53

Table 3.2 The hyper-parameters of our training algorithm described in Algorithm 61

Table 3.3 Environmental conditions for path following scenario. . . 64

Table 3.4 Comparison cases of path following problem according to 3 step rudder angle and 5 step rudder angle 79

Table 3.5 Environmental conditions of comparison scenarios 83

Table 3.6 Environmental conditions of comparison scenarios 87

Table 4.1 State observation in collision avoidance system 92

Table 4.2 Simulation setup with target ships. 99

Table 4.3 Environment conditions for collision avoidance scenario. . 100

Table 5.1 Data transmission 144

Table 5.2 Wave conditions for maintaining the depth of the subsea equipment at a constant lever. 146

Table 5.3 Wave conditions for maintaining the depth of the subsea equipment at a constant lever. 151

Table 5.4 Wave conditions for maintaining the depth of the subsea equipment at a constant lever. 154

Table 5.5 Data transmission. 161

Table 5.6 Dimensional principle of the autonomous OSV and subsea equipment. 169
Table 5.7 Wave conditions for controlling the sway motion of the subsea equipment. 169

List of Figures

Figure 1.1 Overview of this study. 3
Figure 1.2 An example of maritime accident caused by ship collision. 5
Figure 1.3 An example of the use of the OSV to install a subsea manifold. 6
Figure 1.4 The mechanism of the AHC system. 7
Figure 1.5 The mechanism of the ASC system. 8
Figure 1.6 Block diagram of simulation framework configuration. . . 16

Figure 2.1 Schematic representation of ship kinematic variables. . . 21
Figure 2.2 Embedding technique. 26
Figure 2.3 Pendulum example. 29
Figure 2.4 Sliding mode control theory. 32
Figure 2.5 The agent-environment interaction in MDP. 34
Figure 2.6 Bellman expectation equation of state value and state action functions. 36
Figure 2.7 The benefit of use of ROS. 44
Figure 2.8 The components of ROS. 45

Figure 3.1 Diagram of LOS guidance geometry for a straight line. . 47
Figure 3.2 A configuration of RL framework for the path following system. 52
Figure 3.3 Action space in the path following system. 54

Figure 3.4 Action space in the path following system. 55

Figure 3.5 Flow diagram of neural network architecture 57

Figure 3.6 Network architecture. The observable state is fed into two fully-connected layers (FC), the outputs of critic network and policy network are a state value function (green) and an action (orange), respectively. 57

Figure 3.7 Simulation setup in phase one: path following and the principle dimensions for the autonomous OSV and the actuator. 62

Figure 3.8 Graphs of the position of the autonomous OSV in Case 1-1. The result is calculated by the model A without environmental forces before training. 67

Figure 3.9 Graphs of the position of the autonomous OSV in Case 1-2. The result is calculated by the model A without environmental forces before training. 68

Figure 3.10 Graphs of the position of the autonomous OSV in Case 1-3. The result is calculated by the model A without environmental forces before training. 69

Figure 3.11 Graphs of rudder angle of the autonomous OSV in Case 1-3. 70

Figure 3.12 Average reward during training. 70

Figure 3.13 Graphs of the position of the autonomous OSV on a designed path in Case 1-4. The result is calculated by the model A with environmental forces before training. . 71

Figure 3.14 Graphs of rudder angle of the autonomous OSV in Case 1-4. 72

Figure 3.15 Graphs of the position of the autonomous OSV on a designed path in Case 1-5. The result is calculated by the model A with environmental forces before training. . 73
Figure 3.16 Graphs of rudder angle of the autonomous OSV in Case 1-5. 74
Figure 3.17 Graphs of the position of the autonomous OSV on a designed path in Case 1-6. The result is calculated by the model A-1 with environmental forces. 75
Figure 3.18 Graphs of rudder angle of the autonomous OSV in Case 1-6. 76
Figure 3.19 Graphs of the position of the autonomous OSV on a designed path in Case 1-7. The result is calculated by the model A-1 with environmental forces. 77
Figure 3.20 Graphs of rudder angle of the autonomous OSV in Case 1-7. 78
Figure 3.21 Comparison result of path following according to 3 step rudder angle and 5 step rudder angle. 80
Figure 3.22 Comparison result of rudder angle. 80
Figure 3.23 Comparison result of path following according to 3 step rudder angle and 5 step rudder angle. 81
Figure 3.24 Comparison result of rudder angle. 81
Figure 3.25 Continue training the 5 step rudder angle neural network model. 82
Figure 3.26 Comparison results of rudder angle. 82
Figure 3.27 Graphs of the comparison result of PPO and PID in Case 1 and Case 2. The result is calculated by the model A and B without environmental forces. 84

Figure 3.28 Graphs of the comparison result of PPO and PID in Case 2 and Case 3. 85
Figure 3.29 Graphs of the comparison result of PPO and PID in Case 3 and Case 4. 86
Figure 3.30 Graphs of the comparison result of PPO and DQN in Case 1. 87
Figure 3.31 Graphs of the comparison result of PPO and DQN in Case 2. 88

Figure 4.1 Simulation setup in phase two: collision avoidance with three target ships. 90
Figure 4.2 Target ships related state definition. 91
Figure 4.3 Network architecture. The observable state is fed into two fully-connected layers (FC), the outputs of critic network and policy network are a state value function (green) and an action (orange), respectively. 94
Figure 4.4 Target ships for COLREGs: (a) Head-on (b) Crossing (c) Overtaking. 96
Figure 4.5 Simulation setup in phase two: collision avoidance with three target ships. 97
Figure 4.6 Visualization of the simulation using unity 3D. 99
Figure 4.7 Training process of collision avoidance with three target ships. 101
Figure 4.8 Head-on and crossing simulation. 102
Figure 4.9 Graphs of positions of the autonomous OSV, and three target ships in case 2-1. The result is calculated by the model B with external forces. 103

Figure 4.10 Graphs of rudder angle of the autonomous OSV in case 2-1. 104
Figure 4.11 Graphs of positions of the autonomous OSV, and three target ships in case 2-2. The result is calculated by the model B with external forces. 105
Figure 4.12 Graphs of rudder angle of the autonomous OSV in case 2-2. 106
Figure 4.13 Average reward for collision avoidance. 106
Figure 4.14 Limitation and solution of multi-ship collision avoidance method. 108
Figure 4.15 Limitation and solution of multi-ship collision avoidance method. 109
Figure 4.16 Regions used to categorize the position of the target ships.111
Figure 4.17 Procedure of capture of target ships state observation. . 112
Figure 4.18 Ship domain of the own ship. 113
Figure 4.19 Reward function of collision avoidance. 114
Figure 4.20 Reward function of collision avoidance. 116
Figure 4.21 Reward function of collision avoidance. 117
Figure 4.22 Simulation setup based on COLREGs rule. 118
Figure 4.23 Diagram of target ships according to COLREGs. 119
Figure 4.24 Simulation result of Multi-ship collision avoidance. . . . 121
Figure 4.25 Simulation result of Multi-ship collision avoidance. . . . 122
Figure 4.26 Multi-ship collision avoidance setup, where the four ships are heading to the center point. 124
Figure 4.27 Procedure of the multi-ship collision avoidance, where the four ships are starting avoid their target ship. 125

Figure 4.28 Procedure of the multi-ship collision avoidance, where the four ships are backing to their predefined paths. . . . 126
Figure 4.29 Displacement of ship A and ship C. 127
Figure 4.30 Displacement of ship B and ship D. 128

Figure 5.1 HILS environment for AHC system. 130
Figure 5.2 Simplified geometry of the system of the OSV-knuckle boom crane. 131
Figure 5.3 Multibody system of OSV, OSV crane, and subsea equipment. 132
Figure 5.4 Rigid body models of the OSV and the crane. 137
Figure 5.5 Mechanical of dynamixel motor. 139
Figure 5.6 Procedure of HILS for AHC system. 143
Figure 5.7 Simplified geometry of the system of the OSV-knuckle boom crane. 145
Figure 5.8 Graphs of heave motion of the subsea equipment in Case 1-1. 147
Figure 5.9 Graphs of heave motion of the subsea equipment in Case 1-2. 148
Figure 5.10 Graphs of heave motion of the subsea equipment in Case 1-3. 149
Figure 5.11 Graphs of heave motion of the subsea equipment in Case 1-4. 150
Figure 5.12 Graphs of heave motion of the subsea equipment in Case 2-1. 151
Figure 5.13 Graphs of heave motion of the subsea equipment in Case 2-2. 152

Figure 5.14 Graphs of heave motion of the subsea equipment in Case 2-3. 153
Figure 5.15 Graphs of heave motion of the subsea equipment in Case 2-4. 153
Figure 5.16 Graphs of heave motion of the OSV in Case 3-1. 155
Figure 5.17 Graphs of heave motion of the subsea equipment in Case 3-1. 156
Figure 5.18 Graphs of heave motion of the OSV in Case 3-2. 156
Figure 5.19 Graphs of heave motion of the subsea equipment in Case 3-2. 157
Figure 5.20 Graphs of heave motion of the OSV in Case 3-3. 157
Figure 5.21 Graphs of heave motion of the subsea equipment in Case 3-3. 158
Figure 5.22 Hardware setup for validation of the AHC system. 159
Figure 5.23 Validation procedure of AHC system. 160
Figure 5.24 Graphs of comparison result the heave motion of the subsea equipment. 161
Figure 5.25 Simplified geometry of the system of the OSV-knuckle boom crane. 163
Figure 5.26 Block diagram of the sliding-mode control system. 165
Figure 5.27 Graphs of sway angle of the subsea equipment in Case 1. 170
Figure 5.28 Graphs of displace error in the x coordinate of the subsea equipment in Case 1. 171
Figure 5.29 Trajectory of the subsea equipment in Case 1. 172
Figure 5.30 Graphs of sway angle of the subsea equipment in Case 2. 172
Figure 5.31 Graphs of displace error in the x coordinate of the subsea equipment in Case 2. 173

Figure 5.32 Trajectory of the subsea equipment in Case 2. 173
Figure 5.33 Graphs of sway angle of the subsea equipment in Case 3. 174
Figure 5.34 Graphs of displace error in the x coordinate of the subsea equipment in Case 3. 174
Figure 5.35 Trajectory of the subsea equipment in Case 3. 175

Chapter 1

Introduction

This chapter, which serves as a brief introduction to the topics presented in this thesis, starts with a short description of background and motivation for the main topics. Next, we present the research necessities for each application along with a short discussion. After that, we thoroughly review the related work of the discussed topics and discuss the configuration of the proposed simulation framework in detail. Finally, we highlight the originality and elaborate the contributions of this thesis.

1.1 Background and Motivation

The autonomous ship is an inevitable part of the future of the maritime industry. The main advantage of the autonomous ship is that it could reduce the cost of crew and transport. The reduction of the crew will significantly reduce the accommodation spaces and all the crew-related equipment. The cargo capacity will be increased as a consequence. Moreover, technologies developed for the autonomous ships have potential to notably reduce maritime accidents attributed to human error and a significant proportion of these are caused by fatigue and attention deficit. Therefore, the autonomous ship is considered as a key element of a competitive and sustainable ship industry in future. The increasing use of the autonomous ship will also lead to the creation of highly skilled jobs in areas such as integration and planning of autonomous ships. It will boost the development of the maritime industry.

With the increasing demand for automation and self-governance of certain ship operations, autonomous systems with various applications have been explored in the maritime industry for many years. However, developing a high-level autonomous system which can operate in unstructured and unpredictable environment is still challenging task. Most of the autonomous system design are formulate to accommodate some specified situation, which has relatively low ability to deal with complex and unknown environmental conditions. In addition, these methods generally need the accurate dynamic system model. As for the collision avoidance system, such methods can not scale well to handle dense traffic and multiple highly dynamic obstacles due to the limitation to integrate the dynamics model of the ship and the effect of environmental disturbances (e.g. wave, wind and current) into the collision avoidance system. Meanwhile, the recent development of artificial intelligence area has profound effects on the industrial world, which brings researchers powerful algorithms to characterize and control the extremely complex system under the changing environment. Motivated by all of these theories and realistic reasons, the application of path following and collision avoidance system focuses on the field of advanced artificial intelligence approaches, deep reinforcement learning (DRL).

A control system for the crane is similarly important for the autonomous ship design. For instance, when transferring and installing a subsea equipment using a crane on an autonomous ship, the heave and swaying phenomenon of the subsea equipment at the end of flexible wire ropes make its positioning at an exact position difficult due to the ship motion. As a result, Anti-Motion Control (AMC) system is absolutely necessary to ensure installation operations safety. In order to successfully perform operation functions, the AMC system should have a suitable control algorithm. Moreover, the performance of the control system must be evaluated in advance of the operation. Performance analysis of the control system requires complicated testing procedures and a great deal of associated equipment. In particular, such analysis is often very costly and time-consuming, and realistic conditions are typically impossible to establish in a testing environment. To solve this problem, the Hardware-In-the-Loop Simulation (HILS) concept can be used as an effective method to test an control system prior to its final installation.

Fig. 1.1 illustrates the overall research scope including three scenarios based on the aforementioned research necessities. An OSV departs from a port heading to an imperative offshore construction site even though the harsh environment. The OSV has to avoid target ships while following a predefined path. When arriving at the construction site, complete some missions such as installing subsea equipment on the seabed.

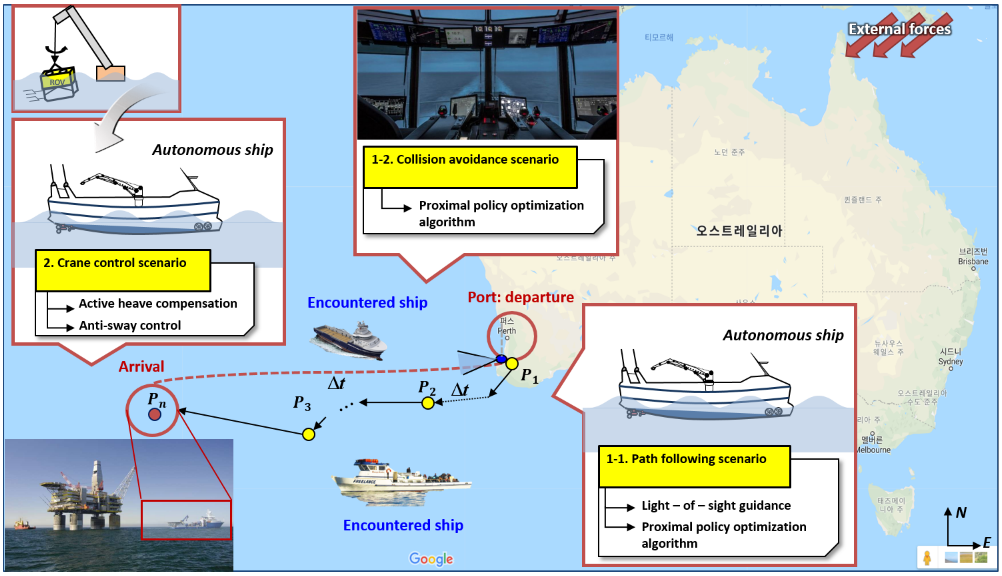


Figure 1.1: Overview of this study.

To complete the above scenarios, we conduct research on maneuvering an autonomous OSV to follow a predefined planar path, and to avoid collisions with target ships. For the AMC system design, it can be achieved through the use of the control system such as PID control law, and sliding mode control law based on the system dynamics on suited dynamic stability theory. To evaluate the proposed control system, we construct an elegant and efficient HILS environment for an AHC system.

1.2 Requirements for Autonomous Operation

1.2.1 Path Following for Autonomous Ship

To plan and control the motion of a marine vessel, path following system is required. Traditionally, path following system is functionally divided into three subsystems that must be implemented on board Fossen (2011): guidance, navigation, and control system (GNC). To accomplish an autonomous operation one needs to know where to go (guidance system), where it is (navigation system) and what to do (control system).

Typically, for an autonomous ship with a path following task, the guidance system consists of guidance laws for heading and surge velocity that, if satisfied, ensure convergence to the desired path. The control system calculates rudder angle to track the reference states delivered by the guidance system.

1.2.2 Collision Avoidance for Autonomous Ship

Marine collision can cause extreme harm to human life and a huge financial loss to the shipowner, and it can also lead to destructive environmental effects. As shown in Fig. 1.2, there are several examples of marine accidents caused by marine collision.

Figure 1.2: An example of maritime accident caused by ship collision.

According to a recent report, which indicates that more than 75% of the marine collision accidents are caused by human decision failure Rothblum (2002). Therefore, the implementation of intelligent collision avoidance system making capabilities in navigation could reduce maritime accidents and its respective causalities and represent long-term economical benefits.

1.2.3 Anti-Motion Control System for Autonomous Ship

Fig. 1.3 shows an example of the operation scenario including installation of a subsea manifold using the OSV crane mounted on its deck. Meanwhile, waves induce motion of the OSV, which induces similar effects on the subsea equipment suspended by the crane.

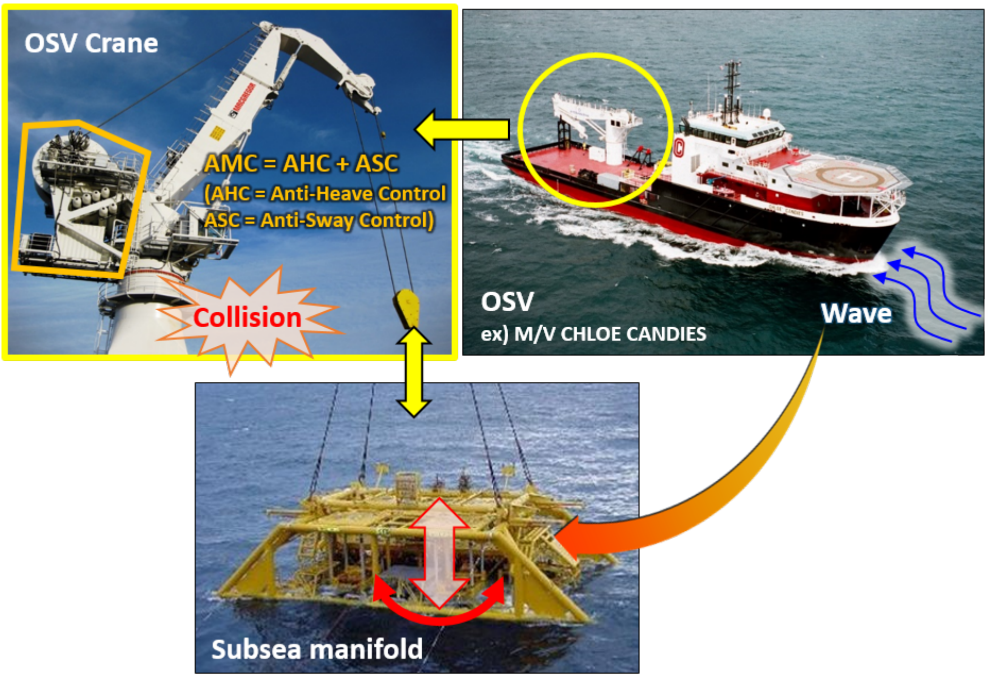


Figure 1.3: An example of the use of the OSV to install a subsea manifold.

Hence, a special equipment called AMC system including AHC and ASC system on an OSV plays a pivotal role in offshore installation operations by minimizing the heave and sway motion of suspended subsea equipment regardless of the OSV motion.

The AHC system is used to reduce the heave motion of the suspended subsea equipment by controlling the length of the associated wire rope. Fig. 1.4 describes the control mechanism of the AHC system.

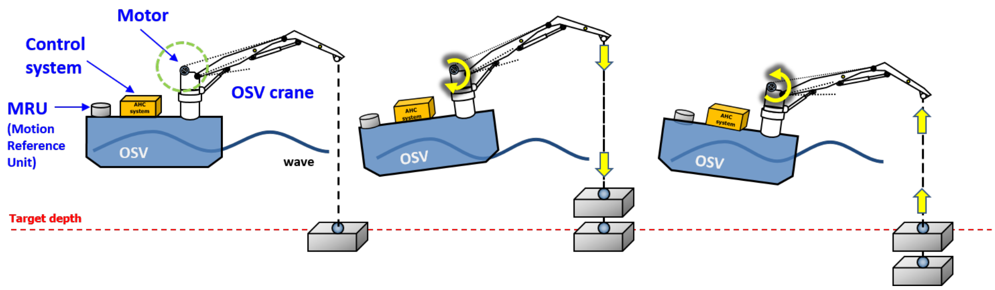


Figure 1.4: The mechanism of the AHC system.

At first, a Motion Reference Unit (MRU) measures the position and orientation of the OSV. This information is transferred to an AHC control system, in order to control the motor. The AHC system calculates the depth of the subsea equipment according to the position and orientation of the OSV and the OSV crane. If the depth of the subsea equipment is less than the target depth, the AHC system sends a lowering signal to extend the length of the wire rope. Otherwise, the AHC system sends a hoisting signal to shorten the length of the wire rope.

Similarly, during the transferring operation, the sway motion of the suspended load (e.g., subsea equipment) is inevitable. In addition, the wind at sea can intensify the sway motion. Meanwhile, the waves induce the OSV motion, and the OSV motion induces motions of the suspended load directly or indirectly. In such a situation, the installation operation by the crane some-times pauses under harsh environmental conditions to avoid above-deck collisions, those between the suspended subsea equipment and the ship structure, or those between the suspended subsea equipment and the crane. boom Jeong et al. (2016).

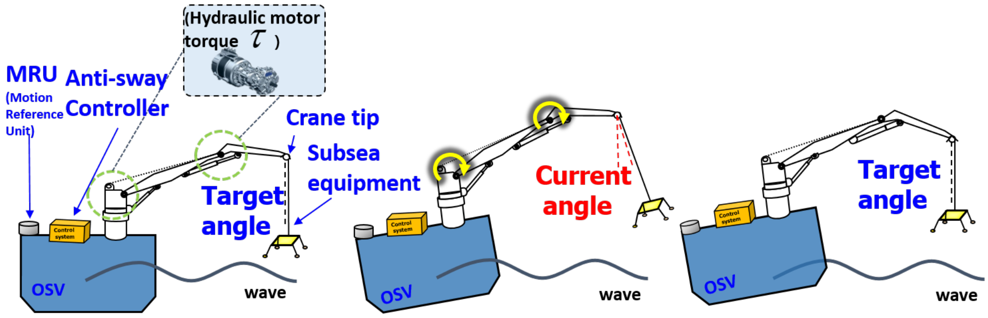


Figure 1.5: The mechanism of the ASC system.

The control mechanism of the ASC is described in Fig. 1.5. First, MRU measures the position and the orientation of the OSV and the subsea equipment, and then the ASC system calculates the control force acting on each joint of the knuckle boom crane according to the current positions of the subsea equipment. Consequently, the positions of the subsea equipment can be stabilized by the controlling of the joint angles.

Therefore, a suitable AMC system is necessary to significantly reduce the residual heave and sway motion of the suspended subsea equipment.

1.3 Related Work

1.3.1 Related Work for Path Following System

In the field of path following, an effective method, look-ahead Line-of-Sight (LOS) guidance law, is used to achieve convergence to the desired path. An overview of LOS guidance law for marine craft can be found in Fossen et al. (2003). Lekkas (2014) mainly concerned with the path planning in combination with the LOS guidance law to solve various kinds of curves. Oh and Sun (2010) proposed a Model Predictive Control (MPC) method, which integrated the LOS guidance law with a path following control for a surface vessel. An alternative adaptive control approach corresponding to the LOS guidance law have been investigated by Fossen and Lekkas (2017).

1.3.2 Related Work for Collision Avoidance System

In this study, we considered not only path following problem, but collision avoidance integrated with it. Although a wide range of researches associated with this have been extensively conducted in various approaches, several of them do not accurately take into consideration the moving obstacles, environmental disturbances, and COLREGs rules, or do not solve the problems simultaneously. Collision avoidance mainly raises two problems: one is motion planning problem, and the other is corresponding control forces computation.

Large amounts of literature only focused on the motion planning. The motion planning for the collision avoidance problem aims to find an admissible collision-free path between the initial and goal configurations, given the environmental conditions with obstacles, an initial and a goal configuration. The motion planning encompasses a wide range of algorithms, which can be distinguished into two categories: local and global methods. The local method such as dynamic window (DW) methods only consider solutions optimal at the current time step, while the global method considers the full configuration space. Examples of global method are A*, rapidly-exploring random trees (RRTs) and hybrid-state A*. In the context of this, to increase the path's optimality and reduce unpredictability, A* algorithm, one of global motion planning method, can be used to guide the RRTs. As a result, Loe (2008) proposed a hybrid approach with the A* guided RRTs for global motion planning and the DW algorithm for local collision avoidance. Stenersen (2015) utilized the velocity obstacle (VO) algorithm as a collision avoidance strategy for the surface vessel to determine safe velocity ranges for avoiding motion obstacle, then applied a proportion differentiation controller to complete several scenarios under COLREGs requirements.

Although the VO approach has the advantage of guaranteeing the safe navigation, the reactive actions of the encountered vessel are neglected. Hence, the path generated by the approach may limited in practice. To address this issue, a research conducted by Zhao et al. (2016) presented a collision avoidance strategy based on the Optimal Reciprocal Collision Avoidance (ORCA) algorithm, which is an extensional formulation of the VO concept. The work revealed that the ORCA algorithm has a better performance than VO. However, the environment conditions had not been taken into consideration. Chiang and Tapia (2018) proposed an RRT-based motion planning method for collision avoidance system on an autonomous surface vessel with COLREGs compliance, but similarly, the research did not consider the environmental disturbances caused by waves and ocean currents.

After generating the collision-free path, the next step is adopting a control system. Much research has been done for collision avoidance of unmanned ships and a number of different approaches for solving this problem have emerged.

Methods relevant for comparison in this case are VO and dynamic window (DW). Other technologies include set-based methods, potential fields and inevitable collision states (ICS). However, these methods generally do not scale well to handle dense traffic and multiple highly dynamic obstacles due to the limitation to integrate the dynamics model of the ship and the effect of environmental disturbances (e.g. wave, wind and current) into the collision avoidance system.

There have been several studies on using analytic control methods or DRL methods for collision avoidance. model predictive control (MPC), one of the popular analytic control algorithm, can compute an optimal trajectory based on predictions of obstacles' motion accounting for their uncertainty. The use of MPC allows the possibility to explicitly include models of relevant components that influences the autonomous ship's dynamics. Within this framework, it is also possible to incorporate models of the obstacles' motion, the evolution of the dynamic environment and different operational constraints Chen et al. (2018). This introduces a design flexibility and performance gains superior to other collision avoidance approaches. However, MPC can be limited by the prediction capabilities of inaccurate dynamics model and excessive computational burden associated with the optimization problem. MPC employs a dynamic model as a cost function and constraints in an optimization problem Johansen et al. (2016), while some of literatures do not consider of the use of MPC for collision avoidance with COLREGs compliance. However, there are two significant drawbacks of the MPC formulation: exorbitant online computational requirement and inability to consider the uncertainties in the optimal control calculation.

Ernst et al. (2009) compared the MPC method and the fitted Q iteration-based RL method. Simulation results showed that MPC was slightly less robust than RL from the numerical point of view, but had a slight advantage in terms of accuracy. Although analytic control methods have shown that it performs well in some specific domains, the performance of this method is often limited due to the complex dynamic systems, which are too complicated to be properly modeled in practical applications. Moreover, the rapid development of artificial intelligence has spurred greater interest in various applications of autonomous tasks. The path following and collision avoidance for the autonomous ship is one of those tasks.

Instead of designing the collision-free path and control system separately, several approaches have used RL to model the complex interactions between the ship and encountered ships. RL has excellent capacity to adapt sophisticated system while it is capable of self-learning, which provides researcher with powerful algorithms to handle an extremely complex system under the unknown environment. A value-based RL method, Q-network, for collision avoidance system has been developed Cheng and Zhang (2018). However, the research only focused on static obstacles and did not consider the environmental disturbances. The existing studies seldom integrate the path following with collision avoidance of moving obstacles simultaneously, to overcome this limitation, we focused on the problem of integrating the path following with collision avoidance for an autonomous ship.

To achieve this, we utilized a policy-based RL methodology for the design of motion planning combining with control system for autonomous ship operating in an unknown ocean environment, by considering the target ships respecting COLREGs compliance. The advantage of the proposed method is that it is extensible and easy to operate in terms of various environmental condition and COLREGs regulations compliance.

1.3.3 Related Work for Anti-Heave Control System

The general concept of HILS is comprehensively summarized in Schlager et al. (2006) and Pedersen et al. (2013). HILS technology has been widely used in the defense and aerospace industries as early as the 1950s. At that time, in spite of the high cost of HILS technology, these industries benefitted greatly from its safety. During the past decade, the advancement of computer technology has led to the adoption of HILS to automotive systems in the 1990s. Isermann et al. (1999) performed an efficient real-time HILS for Electronic Control Unit design and verification. Similar work has recently been done by Fathy et al. (2006), in providing an overview of HILS for an engine system and its prospects in the automotive area.

For the offshore industry, ships and offshore structures are equipped with advanced control systems for Dynamic Positioning and Power Management. Unlike the mass production of the automation industry, the controller is unique for each vessel or offshore structure. In this context, HILS seems indispensable in this field. Several works proposed by Pedersen et al. (2013) and Kaliappan et al. (2012) were dedicated to providing experience to implement HILS for DP and PM systems. Furthermore, HILS for the AHC system is an important field. Hu et al. (2009) performed HILS with mathematical models and physical testing of an AHC system on a pipeline lifting mining system. The simulation is based on a controller (PXI) from LabVIEW. Another common tool (Simulink) is also widely used as a connection interface between controller and virtual model. This was implemented in Muraspahic et al. (2012). The virtual model was modeled by commercial software (Simulation X), without considering environmental conditions. The Siemens PLC (Programmable Logic Controller) was utilized as a controller regulating the virtual model. The visualization model only displayed a simple winch model. An attempt to build the virtual model based on the Bond Graph method was presented by Aarseth et al. (2014). The virtual model was described as the energy flow by using 20-sim. Even though the software can display 3D-animation for development, there are some limitations in terms of the visualization model based on VR.

1.3.4 Related Work for Anti-Sway Control System

The cranes on an offshore support vessel (OSV) are used for offshore transportation and the installation of subsea equipment at sea Hong et al. (2016). The knuckle boom crane can perform various tasks, as it is characterized by the design of a folded knuckle that is attached to an extension rod.

During the installation operation, the sway motion of the suspended load (e.g., subsea equipment) is inevitable. In most cases, the reduction of the sway motion is achieved by the crane control. Abdel-Rahman et al. (2003) provided a well-classified review of the crane control. Gjelstenli (2012) used the cascade control method to solve the antisway problem for offshore crane. More recently, Ramli et al. (2017) also conducted a comprehensive review of the control strategies for different crane types. Most of the researchers concentrated on the overhead crane, tower crane, and boom crane. Abe (2011) used radial basis function networks for the trajectory planning of overhead cranes and reduced the payload sway motion. In addition, an experiment was conducted to verify the proposed controller. Le et al. (2013) presented the sliding-mode control method for a tower crane to suppress the load sway motion and the tracking of a trolley to the desired position. An antisway controller for overhead crane based on multi-sliding mode method was investigated by Xu et al. (2012). Duong et al. (2012) considered the antisway control for a tower crane by using a recurrent neural network. Also, Wu et al. (2016) applied an adaptive fuzzy controller, which is a nonlinear method, to a tower crane without the requirement of a detailed mathematical model of the crane. However, the knuckle boom crane is not the subject of most of the previous studies.

The knuckle boom crane, however, exhibits underactuated behavior, since the number of actuators is fewer than those of the systemic state variables. Therefore, the control mechanism becomes more complex, leading to a more difficult controller design. Only a few works of the literature have focused on this crane type. Bak et al. (2011) performed the tool-point control for a hydraulically actuated knuckle boom crane. proposed an additional auxiliary system to reduce the load oscillations. The additional system can directly force the hoist wire rope to control the payload sway angle. Chu et al. (2015) established a multidomain system for the knuckle boom crane and implemented an ASC. In these studies, the antisway controller was designed by the controlling of the movement of the crane tip. In the present study, the antisway controller for a knuckle boom crane on an OSV is considered.

1.4 Configuration of Simulation Framework

According to the requirements defined in the previous section, The necessary techniques are included in the simulation framework as shown in Fig. 1.6 The proposed framework comprises three layers: application layer, autonomous ship design layer, and general techniques layer. The following sections will explain the role of each layer briefly.

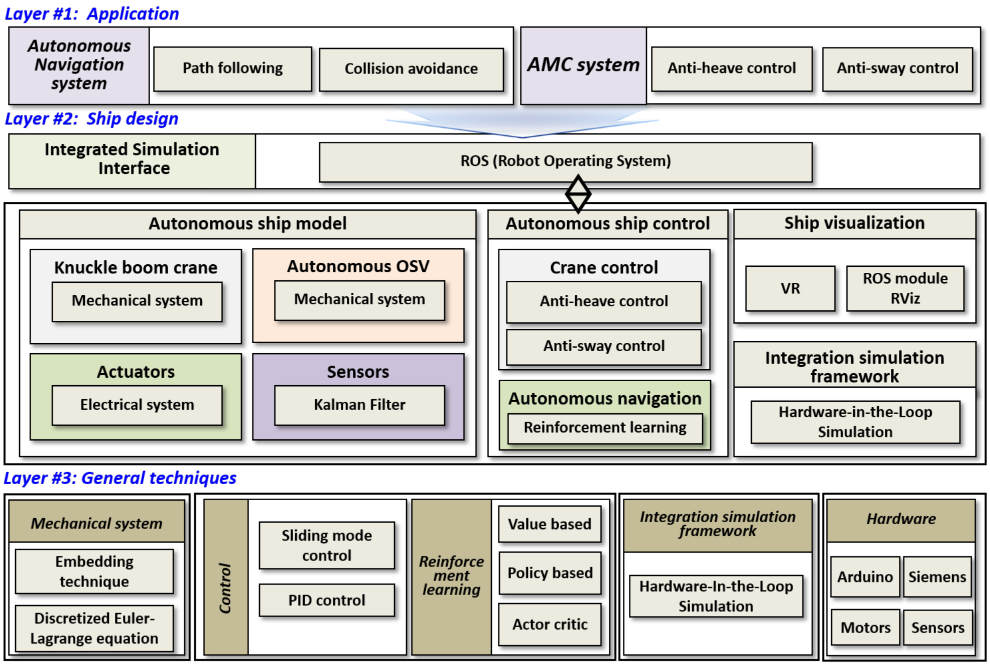


Figure 1.6: Block diagram of simulation framework configuration.

1.4.1 Application Layer

The first layer is the application layer, which exhibits two main applications including the autonomous navigation and crane control application in this study. The autonomous navigation method consists of path following and collision avoidance method, and AMC method consists of anti-heave control method and anti-sway control method.

1.4.2 Autonomous Ship Design Layer

For the specific autonomous ship design, three modules are indispensable: integrated simulation interface, autonomous ship model and autonomous control system. A visualization module is not necessary, however, it can be used to help users check the simulation results through immersive and realistic views, and can be used for training purposes.

An integrated simulation interface is used to exchange information between each module. In this study, we adopted the robot operating system (ROS) as an interface to formulate a HILS environment for AHC system. Knuckle boom crane mounted on an autonomous ship, except it, we only consider that the ship is comprised of the hull, actuation system, and sensors. We conducted two control problems: autonomous navigation and crane control.

1.4.3 General Technique Layer

General technique layer provides the main techniques we used in this study. To represent the mechanical system of the autonomous ship, the equation of motion of the ship, kunckle boom crane and the subsea equipment can be formulated as the multibody system. These can be formulated differently according to the expression of the constraint forces. For the ASC system design, the knuckle boom crane has been formulated using the embedding technique. In term of control system design, which including traditional control theory such as adaptive control, sliding mode control and so on, we selected the sliding mode control method as the control method of the ASC system. Additionally, the DRL is considered as an effective control method, which is divided into value-based, policy-based and actor-critic method. The related theories will be explained in chapter 2.

1.5 Contributions (Originality)

The original contributions of this dissertation can be divided on the theoretical side:

- We contribute to define the problem of the COLREGs-based collision avoidance for multiple autonomous ships under realistic assumptions. In terms of the COLREGs region, we defined it based on the ship domain concept.
- We present an alternative method for solving path following and collision avoidance problem simultaneously. According to the simulation results, the proposed RL scheme has the ability to find the optimal solution, even when unknown disturbances affect the ship's motion.

On the practical side, the main original contribution is that:

- In order to validate the proposed AHC system, we used the ROS as the simulation interface, which helps to construct the HILS framework for AHC system. In addition, we experimentally show that our approach makes the validation process becomes time-consuming and easy handle.

Chapter 2

Theoretical Backgrounds

2.1 Maneuvering Model for Autonomous Ship

This section presents the autonomous OSV model and the related assumptions that are considered in this section. Two references frames, the North-East-Down (NED) coordinate system $\mathbf{n} = (x_n, y_n, z_n)$ and the body-fixed reference frame $\mathbf{b} = (x_b, y_b, z_b)$ are utilized in this study to characterize the location and orientation of the autonomous OSV. The NED frame is defined as a tangent plane on the surface of Earth moving with the ship. The x axis is parallel to lines of constant longitude, the y axis is parallel to lines of constant latitude, and z axis pointing towards the center of the Earth. The body-fixed coordinate is moving with the ship. The center of gravity of the ship is defined as the origin.

2.1.1 Kinematic Equation for Autonomous Ship

In this study, the simplified three-degree-of-freedom (3-DOF) vessel dynamic model is used to describe the autonomous ship motions in the horizontal plane (surge, sway, and yaw) (17). The rigid body kinematic equation is

$$\dot{\eta} = \mathbf{R}(\psi)\mathbf{v}, \tag{2.1}$$

Where $\eta = [x, y, \psi]^T$ represents the earth-fixed position and heading angle,

$\mathbf{v} = [u, v, r]$ represents the vessel-fixed velocities, $\mathbf{R}(\psi)$ refers to the rotation matrix from the earth-fixed frame to the vessel-fixed frame. With the ship speed $V = \sqrt{(u^2 + v^2)}$, we define the course angle $\chi = \psi + \beta$ and the sideslip angle $\beta = \arcsin(v/V)$, which are illustrated in Fig. 2.1. Note that the heading angle and course angle are equal when there is no sideslip.

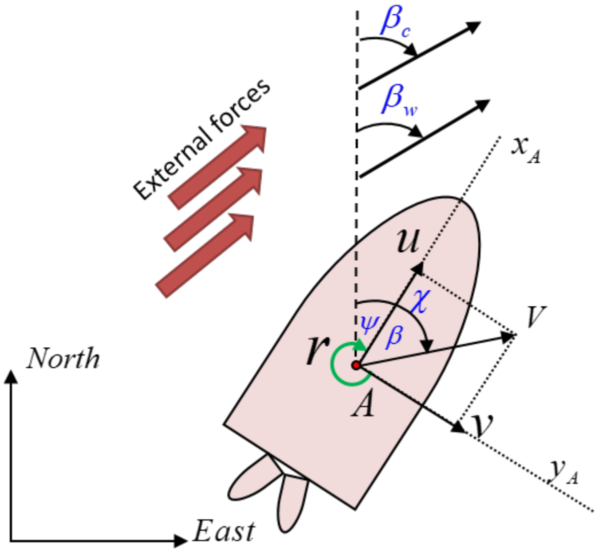


Figure 2.1: Schematic representation of ship kinematic variables.

2.1.2 Kinetic Equation for Autonomous Ship

The dynamic equation for the autonomous ship can be modelled using the following form:

$$\mathbf{M}\dot{v} + \mathbf{C}(v)v + \mathbf{D}(v)v = \tau + \tau_{external\_force}, \tag{2.2}$$

Where $\mathbf{M} = \mathbf{M}_{RB} + \mathbf{M}_A$ is the mass matrix consisting of rigid-body mass and hydrodynamic added mass, $\mathbf{C}(v) = \mathbf{C}_{RB} + \mathbf{C}_A$ is the Coriolis and centripetal matrix.

$$\mathbf{M}_{RB} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{C}_{RB} = \begin{bmatrix} 0 & 0 & -mv \\ 0 & m & mu \\ mv & -mu & 0 \end{bmatrix} \tag{2.3}$$

where m is the ship mass and I_z denotes the moment of inertia about the z-axis. The hydrodynamic added mass and added Coriolis matrices are expressed as

$$\mathbf{M}_A = \begin{bmatrix} -X_{\dot{u}} & 0 & 0 \\ 0 & -Y_{\dot{v}} & -Y_{\dot{r}} \\ 0 & -N_{\dot{v}} & -N_{\dot{r}} \end{bmatrix}, \mathbf{C}_A = \begin{bmatrix} 0 & 0 & Y_{\dot{v}}v + Y_{\dot{r}}r \\ 0 & 0 & -X_{\dot{u}}u \\ -Y_{\dot{v}}v - Y_{\dot{r}}r & -X_{\dot{u}}u & 0 \end{bmatrix} \tag{2.4}$$

where $\mathbf{D}(v)$ is the nonlinear damping matrix, which can be defined as the sum of linear damping and nonlinear damping, $\mathbf{D}(v) = \mathbf{D}_L + \mathbf{D}_{NL}$, where

$$\mathbf{D}_L = \begin{bmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & -Y_r \\ 0 & -N_v & -N_r \end{bmatrix}, \mathbf{D}_{NL}(v) = \begin{bmatrix} -X_{|u|u}\,|u| & 0 & 0 \\ 0 & -Y_{|v|v}\,|v| & -Y_{|v|r}\,|v| \\ 0 & -N_{|v|v}\,|v| & -N_{|v|r}\,|v| \end{bmatrix} \tag{2.5}$$

We assume that the ship has only one control input, rudder angle *delta*. The reason is that we maintained a constant propeller speed which is more realistic for most ship maneuvering operating conditions. The control force τ takes the following form:

$$\tau = \begin{bmatrix} X_\delta \delta \\ Y_\delta \delta \\ N_\delta \delta \end{bmatrix} \tag{2.6}$$

where X_τ, Y_τ, and N_τ are the rudder coefficients associated with surge force, sway force, and yaw moment, respectively. Furthermore, environmental forces disturbances act upon the ship that affects the behavior of the ship motion. In this study, the environmental forces are split into three parts: wind force, current force and wave force. The longitudinal wind force $F_{x_{wind}}$, lateral wind force $F_{y_{wind}}$, and wind moment $M_{z_{wind}}$, which the wind exerts on the autonomous OSV can be computed by Journée and Massie (2001):

$$F_{x_{wind}} = \frac{1}{2}\rho_{air} A_T C_{wx}(\alpha_{rwave}) V_{rw}^2, \tag{2.7}$$

$$F_{y_{wind}} = \frac{1}{2}\rho_{air} A_L C_{wy}(\alpha_{rwave}) V_{rw}^2, \tag{2.8}$$

$$M_{z_{wind}} = \frac{1}{2}\rho_{air} A_L L C_{wN}(\alpha_{rwave}) V_{rw}^2 \tag{2.9}$$

where ρ_{air} is the density of air, A_T and A_L are the transverse projected wind area and lateral projected wind area, respectively. L is the length of the ship. The wind speed and direction determine the longitudinal and lateral wind forces and the yawing moment on the ship. The wind load coefficients C_{wx}, C_{wy} and C_{wN} are parameterized in terms of relative wind direction. The relative wind direction speed V_{rw} are defined as follows, with the wind direction β_w and wind speed V_w:

$$\alpha_{rw} = \psi - \beta_w \tag{2.10}$$

$$V_{rw} = \sqrt{u_{rw}^2 + v_{rw}^2} \tag{2.11}$$

Where the components of relative velocity in the x and y directions are:

$$u_{rw} = u - u_w = u - V_w \cos\alpha_{rw} \tag{2.12}$$

$$v_{rw} = v - v_w = v - V_w \sin\alpha_{rw} \tag{2.13}$$

Similarly, based on the velocity vector synthesis method, the relative current velocity exerted by the current on the ship can be calculated from:

$$u_{rc} = u - u_c = u - V_c \cos\psi - \beta_c \tag{2.14}$$

$$v_{rc} = v - v_c = v - V_c \sin\psi - \beta_c \tag{2.15}$$

Where β_c is the current direction, and V_c is the speed of the ocean current. The influence of the wave interference is mainly divided into the first-order wave force and the second-order wave force, which can be seen as a linear wave superimposed by a large number of regular waves of different frequencies and wave height. The study only considers the second-order wave drift force that affects the autonomous ship's position and orientation. The wave force and moment can be calculated as follows:

$$F_{x_{\text{wave}}} = 1/2\, \rho_{\text{water}}\, \xi_{\text{wave}}^2\, g\, L\, C_{\text{wavex}}(\alpha_{\text{rwave}}), \tag{2.16}$$

$$F_{y_{\text{wave}}} = 1/2\, \rho_{\text{water}}\, \xi_{\text{wave}}^2\, g\, L\, C_{\text{wavey}}(\alpha_{\text{rwave}}), \tag{2.17}$$

$$M_{z_{\text{wave}}} = 1/2\, \rho_{\text{water}}\, \xi_{\text{wave}}^2\, g\, L^2\, C_{\text{waveN}}(\alpha_{\text{rwave}}) \tag{2.18}$$

Where ξ_{wave} is the wave height and α_{rwave} is the relative wave direction. C_{wN}, C_{wN} and C_{wN} represent the coefficients of the second order wave drift force and yawing moment respectively. The dynamic equation of the autonomous ship motion can be rewritten by relative velocities as:

$$\mathbf{M}\dot{v}_{rc} + \mathbf{C}(v_{rc})v_{rc} + \mathbf{D}(v_{rc})v_{rc} = \tau + \tau_{\text{wind}} + \tau_{\text{wave}} \tag{2.19}$$

2.2 Multibody Dynamics Model for Knuckle Boom Crane of Autonomous Ship

In this section, some methodologies which are required to perform the anti-sway control simulation are described. Basically, the Newton's 2Nd law could be applied to describe the motion of the autonomous OSV. An autonomous OSV mounted crane can be regarded s a multibody system which consists of interconnected rigid bodies with joints and springs like wire ropes. The equations of motion based on multibody system dynamics is required to analyze the motion of a knuckle crane system including a suspended subsea equipment. The relative motion that is permitted between bodies in the multibody system is often constrained by connections between those bodies.

2.2.1 Embedding Techniques

In the case of constrained multibody dynamics, in the case of the knuckle boom crane mounted on an autonomous ship, different numbers of coordinates can be selected, leading to different forms of the dynamic equation.

Newton's equation of motion for the the object on the ramp can be stated as:

$$m\ddot{\mathbf{r}} = \mathbf{F}^e + \mathbf{F}^c \tag{2.20}$$

The vectors in Eq. 2.20 are represented in terms of the Cartesian coordinates. m is the mass and the mass moment of inertia matrices, and $\mathbf{r}$ is the position vector of the center of gravity of the object with respect to the Cartesian coordinates. The resultant force consists of the external force $\mathbf{F}^e$ and the constraint force $\mathbf{F}^c$ caused by kinematic constraints.

The virtual work, which states that the work done by a set up forces acting

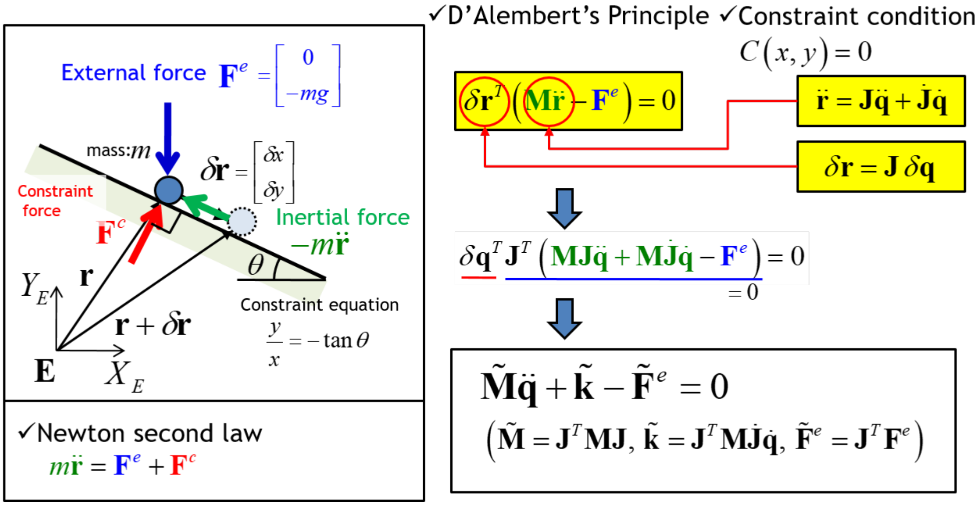


Figure 2.2: Embedding technique.

on a static system vanishes in a virtual displacement. Let δr denote the virtual displacement of the object on the slope where forces including external force $\mathbf{F}^e$, constraint force $\mathbf{F}^c$ and inertial force acts. Adding the virtual work done by forces gives

$$\mathbf{F}^e + \mathbf{F}^c - m\ddot{\mathbf{r}} = 0 \tag{2.21a}$$

$$\delta W = \delta\mathbf{r} \cdot (\mathbf{F}^e + \mathbf{F}^c - m\ddot{\mathbf{r}}) = 0 \tag{2.21b}$$

Here, the symbol δ is used because it is not deal with the increment that actually occurs.

Since the constraint force $\mathbf{F}^c$ is perpendicular to the direction of the motion, then

$$\delta\mathbf{r} \cdot \mathbf{F}^c = 0 \tag{2.22}$$

Correspondingly, the work done by the resultant force becomes

$$\delta W = \delta\mathbf{r} \cdot (\mathbf{F}^e - m\ddot{\mathbf{r}}) = 0 \tag{2.23}$$

$$\delta \mathbf{r} \cdot (m\ddot{\mathbf{r}} - \mathbf{F}^e) = 0 \tag{2.24a}$$

$$(\delta x, \delta y) \cdot ((m\ddot{x}, m\ddot{y}) - (0, -mg)) = 0 \tag{2.24b}$$

Based on the matrix formulation, like $a \cdot b = a^T b$, Eq. 2.24b can be rewritten as

$$\delta \mathbf{r}^T (\mathbf{M}\ddot{r} - \mathbf{F}^e) = 0 \tag{2.25}$$

$$\begin{bmatrix} \delta x \\ \delta y \end{bmatrix}^T (\begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} - \begin{bmatrix} 0 \\ -mg \end{bmatrix}) = 0 \tag{2.26}$$

A constraint acts on the above system through application of constraint forces. The constraint force are restrict the velocity of the system to be tangent to the surface at all times. The constraint equation in the above problem can be written in a functional form below

$$C(x, y) = 0 \tag{2.27a}$$

$$x \tan\theta + y = 0 \tag{2.27b}$$

$$\dot{x} \tan\theta + \dot{y} = 0 \tag{2.27c}$$

To describe the motion of the system, we can select a smaller set of variables that completely describes the configuration of the system.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \dot{x} \\ -\dot{x}\tan\theta \end{bmatrix} \tag{2.28}$$

$$= \begin{bmatrix} 1 \\ -\tan\theta \end{bmatrix} \dot{x} \tag{2.29}$$

As a result, we can obtain the following constraint equation $\dot{\mathbf{r}} = \mathbf{J}\dot{\mathbf{q}}$, where $\mathbf{r}$ is the system variables, $\mathbf{J}$ denotes the velocity transform matrix, and the $\mathbf{q}$ is the generalized coordinates for the system.

$$\dot{\mathbf{r}} = \mathbf{J}\dot{\mathbf{q}} \tag{2.30}$$

$$\ddot{\mathbf{r}} = \mathbf{J}\ddot{\mathbf{q}} + \dot{\mathbf{J}}\dot{\mathbf{q}} \tag{2.31}$$

Substituting Eq. 2.31 into Eq. 2.26, we can obtain the equation

$$\mathbf{M}\mathbf{J}\ddot{\mathbf{q}} + \mathbf{M}\dot{\mathbf{J}}\dot{\mathbf{q}} = \mathbf{F}^e + \mathbf{F}^c \tag{2.32}$$

Multiplying both sides of Eq. 2.32 by $\mathbf{J}^T$ yields

$$\mathbf{J}^T\mathbf{M}\mathbf{J}\ddot{\mathbf{q}} + \mathbf{J}^T\mathbf{M}\dot{\mathbf{J}}\dot{\mathbf{q}} = \mathbf{J}^T\mathbf{F}^e + \mathbf{J}^T\mathbf{F}^c \tag{2.33a}$$

$$\tilde{\mathbf{M}}\ddot{\mathbf{q}} + \tilde{\mathbf{M}} = \tilde{\mathbf{F}}^e \tag{2.33b}$$

where $\tilde{\mathbf{M}} = \mathbf{J}^T\mathbf{M}\mathbf{J}$, $\tilde{\mathbf{K}} = \mathbf{J}^T\mathbf{M}\dot{\mathbf{J}}$, and $\tilde{\mathbf{F}}^e = \mathbf{J}^T\mathbf{F}^e$; $\tilde{M}$ is the mass and the generalized mass moment of the inertia matrix, $\tilde{\mathbf{K}}$ is the generalized Coriolis and centrifugal force, $\tilde{\mathbf{F}}^e$ is the generalized external force, $\mathbf{J}$ is the velocity transformation matrix, and $\dot{\mathbf{J}}$ is the acceleration transformation matrix. Eq. 2.33b is the final form of the equations of motion of the multibody system based on the multibody system dynamics. If we use this equation, we can get dynamic motion of the autonomous OSV, the knuckle boom crane, and the suspended subsea equipment, including constraint forces among them.

Next, we take a 2 degree of freedom pendulum as an example using the induced embedding techniques, see Fig.2.3.

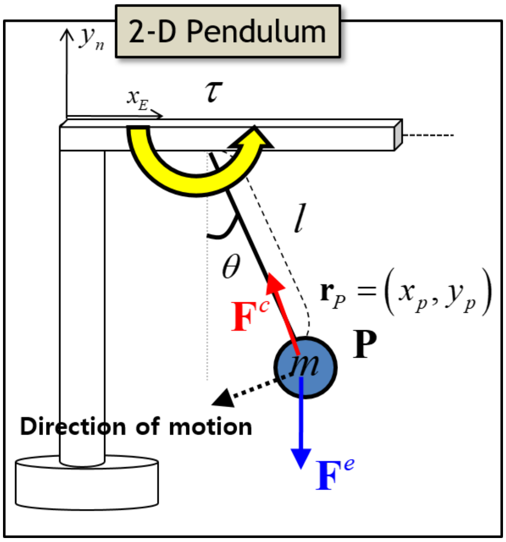


Figure 2.3: Pendulum example.

The equation of motion of the 2D pendulum gives

$$\begin{bmatrix} x_P \\ y_p \end{bmatrix} = \begin{bmatrix} \sin\theta \\ -\cos\theta \end{bmatrix} l \tag{2.34}$$

$$\begin{bmatrix} \dot{x_P} \\ \dot{y_P} \end{bmatrix} = \begin{bmatrix} \cos\theta l \\ \sin\theta l \end{bmatrix} \dot{\theta} \tag{2.35}$$

$$\mathbf{J} = \begin{bmatrix} \cos\theta l \\ \sin\theta l \end{bmatrix} \tag{2.36}$$

From the above constraint equation, we can obtain the velocity transform matrix $\mathbf{J}$. The equations of the systemic motions can be written in a matrix form by inserting $\mathbf{J}$, as follows:

$$\tilde{\mathbf{M}}\ddot{\mathbf{q}} + \tilde{\mathbf{K}} -^{E} \tilde{\mathbf{F}}^{e} = 0 \tag{2.37}$$

where

$$\tilde{\mathbf{M}} = \mathbf{J}^T\mathbf{M}\mathbf{J} = \begin{bmatrix}\cos\theta l & \sin\theta l\end{bmatrix}\begin{bmatrix}m & 0\\ 0 & m\end{bmatrix}\begin{bmatrix}\cos\theta l\\ \sin\theta l\end{bmatrix} = ml^2 \tag{2.38a}$$

$$\tilde{\mathbf{K}} = \mathbf{J}^T\mathbf{M}\dot{\mathbf{J}} = \begin{bmatrix}\cos\theta l & \sin\theta l\end{bmatrix}\begin{bmatrix}m & 0\\ 0 & m\end{bmatrix}\begin{bmatrix}-\sin\theta\dot{\theta}l\\ \cos\theta\dot{\theta}l\end{bmatrix}\dot{\theta} = 0 \tag{2.38b}$$

$${}^{E}\tilde{\mathbf{F}}^{e} = \mathbf{J}^T\,{}^{E}\mathbf{F}^{e} = \begin{bmatrix}\cos\theta l & \sin\theta l\end{bmatrix}\begin{bmatrix}0\\ -mg\end{bmatrix} = -lmg\sin\theta \tag{2.38c}$$

We can get the final equation of the 2D pendulum

$$ml^2\ddot{\theta} = -lmg\sin\theta \tag{2.39}$$

2.3 Control System Design

2.3.1 Proportional-Integral-Derivative (PID) Control

For AHC system, the position of the suspended subsea equipment is controlled by using a PID (Proportional-Integral-Derivation Control) controller that adjusts the motor's speed to the desired speed. A mathematical description of the PID controller is shown in the following equation:

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt} \tag{2.40}$$

where, $u(t)$ is the control signal, and $e(t)$ is the error signal. K_p, K_i and K_d denote the coefficients for the proportional, integral, and derivative terms, respectively.

2.3.2 Sliding Mode Control

Sliding mode control law is based on defining exponentially stable sliding surfaces as a function of the output errors Utkin (1977). design is based on making the value of sliding surface s to be equally zero. The sliding mode surface is defined as follows

$$s = \lambda x_1 + x_2 = 0(\lambda > 0) \tag{2.41}$$

$$\dot{x_1} = x_2 \tag{2.42}$$

To reduce the sway angle of the 2D pendulum illustrated in Fig. 2.3, the control force τ can be calculated by using the sliding mode control law.

At first, the Eq. 2.39 can be formulated as follows

$$ml\ddot{\theta} = -mg\sin\theta + \tau \tag{2.43}$$

$$\ddot{\theta} = -\frac{g}{l}\sin\theta + \frac{1}{ml}\tau \tag{2.44}$$

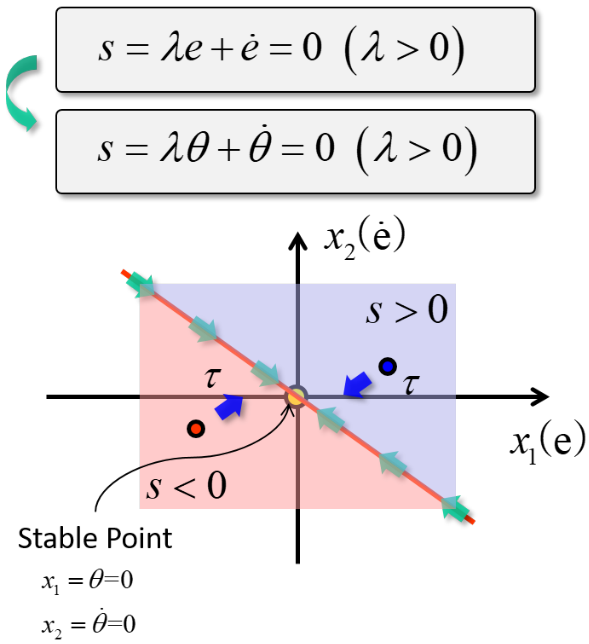


Figure 2.4: Sliding mode control theory.

Let $x_1 = \theta$ and $x_2 = \dot{\theta}$, Eq. 2.44 can be defined

$$\dot{x_1} = x_2 \tag{2.45}$$

$$\dot{x_2} = -\frac{g}{l}\sin x_1 + \frac{1}{ml^2}\tau \tag{2.46}$$

$\dot{s}$ has to satisfy Eq. 2.47. Sliding mode control always result in chattering around the surface due to the signum function. To reduce chattering, $sgn(s)$ function should be replaced by a saturation function.

$$\dot{s} = -K_1 s - K_2 sat(s) \tag{2.47}$$

Dynamical system stability has be evaluated by the use of many different methods, for example by using methods based on the eigenvalues in linear systems, which are measures of energy dissipation, or through methods using functions that represent the energy in the system, such as Lyapunov functions, which is usually the way of assuring stability when handling non-linear dynamics.

Consider a Lyapunov function given by $V = \frac{1}{2}s^T s$. To make the system stability

$$\dot{V} = s^T \dot{s} <= 0 \tag{2.48}$$

Define the sliding surface plan $s = \dot{\theta} + \lambda\theta$ and then differentiating the surface s with respect to time, one can obtain:

$$\mathbf{s} = \dot{\mathbf{e}} + \lambda\mathbf{e} = \dot{\theta} + \lambda\theta \tag{2.49}$$

$$\dot{\mathbf{s}} = \ddot{\mathbf{e}} + \lambda\dot{\mathbf{e}} = \ddot{\theta} + \lambda\dot{\theta} \tag{2.50}$$

As a result, we can get

$$\ddot{\theta} = \lambda\dot{\theta} - \dot{s} \tag{2.51}$$

$$\dot{\theta} = \lambda\theta - s \tag{2.52}$$

The control force τ

$$\tau = ml\ddot{\theta} + mg\sin\theta \tag{2.53}$$

$$\tau = ml(\lambda\dot{\theta} - \dot{s}) + mg\sin\theta \tag{2.54}$$

where $\dot{s} = -K_1 s - K_2 sat(s)$, and K_1 and K_2 are constant parameters.

2.4 Deep Reinforcement Learning Algorithm

In this study, we used deep reinforcement learning (DRL) algorithm to solve the autonomous navigation problem. The learning in RL problem discovers which action yield the most reward by trial and error search Sutton et al. (1998). In most RL topics, mathematical frameworks are created to tackle problems. The mathematical framework is referred to the Markov Decision Process (MDP), which can produce an easy framework to model a complex problem. As illustrated in Fig. 2.5. The process of the agent observing the environment output consisting of a reward and the next state, and then acting upon that. This whole process is an MDP for short.

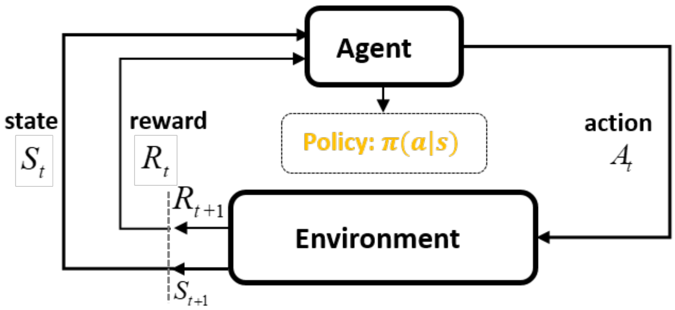


Figure 2.5: The agent-environment interaction in MDP.

MDP is meant to be a straightforward framing of the problem of learning from interaction to achieve a goal. The agent and the environment interact at each of a sequence of discrete time steps. The agent selecting actions and the environment responding to these actions and presenting new situations to the agent. The decision maker is called the agent, for example, in the path following and collision avoidance problem, agent is the controller of the autonomous ship. Environment in RL means the outside (including the environmental condition, target ships, and so on) except the agent. Rewards are given out but they may be infrequent and delayed. Very often, the long-delayed rewards make it extremely hard to untangle the information and traceback what sequence of actions contributed to the rewards.

The agent makes the decisions on which actions to take at each time step. The agent makes these decisions based on the scalar reward R_t receives and the observed environment S_t. The environment receives the action A_t from the agent and emits a new observation state R_{t+1} and scalar reward R_t. What happens next to the environment depends on the history. The action taken in a state is drawn from a stochastic policy $\pi(a|s)$, which is the probability that $A_t = a$ if $S_t = s$. The goal of the RL agent is to find the policy $\pi(a|s)$ which maximizes the expected discounted sum of rewards.

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \tag{2.55}$$

Where $\gamma \in [0, 1]$ denotes the reward discount factor. During learning, the agent has to estimate how good it is for the agent to be in a given state (state value function) and how good it is to perform a given action in a state (state-action function), see Fig. 2.6. The notation of "how good" here is defined in terms of future rewards that can be expected Sutton et al. (1998). Accordingly, the state value function

$$V_\pi(s) = E[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s] \tag{2.56}$$

Similarly, the state action function can be defined as follows:

$$Q_\pi(s, a) = E[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s, A_t = a] \tag{2.57}$$

RL agent can be categorised into the following types:

- **Value based method**: take actions greedily based on state value function or state action value function.

- **Policy based method**: use the policy function directly to take actions.
- **Actor Critic method**: use both the policy and value functions. Where a critic neural network is used to tell an actor neural network how good the action is and how it should adjust, then the actor neural network decides which actions to take.

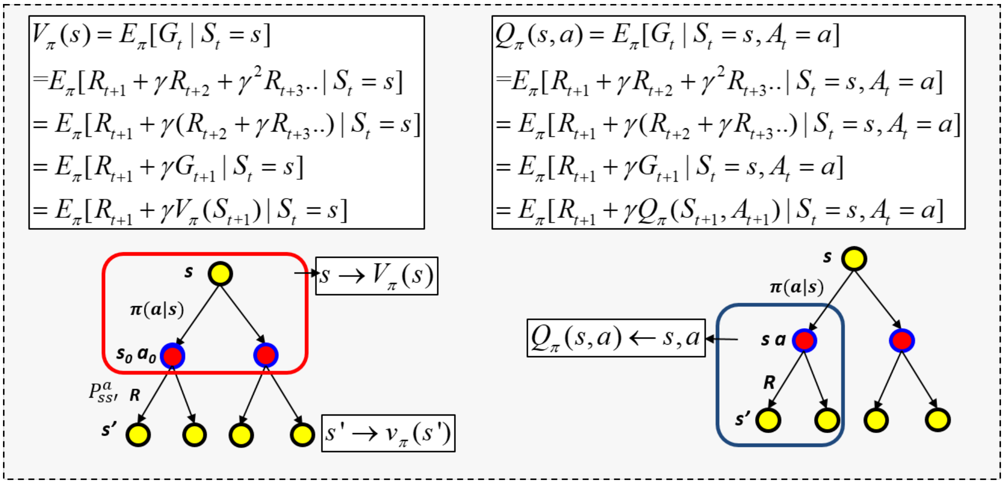


Figure 2.6: Bellman expectation equation of state value and state action functions.

2.4.1 Value Based Learning Method

TD error is the error in the estimate made at each time, which depends on the next state and next reward. It measures the difference between the estimate value of V_t and the better estimate $R_{t+1} + \gamma V(S_{t+2})$.

$$\delta_t = R_{t+1} + \gamma V(S_{t+2}) - V(S_t) \tag{2.58}$$

Value based learning method, such as Q-learning, uses a state action value function $Q(s, a)$, which is defined as the expected return the agent would get by starting in state s, taking action a and then following a policy π.

$R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_{t+1}, A_t)$ is the difference between the updated (target) Q value and the predict Q value, which takes the form of a TD error. This equation tells us that the maximum future reward is the reward the agent received for entering the current state S_t plus the maximum future reward for the next state S_{t+1}. The Q learning is defined by Eq. 2.59

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)] \tag{2.59}$$

Based on the above equation, the policy strategy is taking actions which result in highest value of value action function Q. The policy is called Greedy-Policy.

$$\pi(s, a) = \begin{cases} 1 & \text{if } a = \text{argmax}_a Q(s, a) \\ 0 & \text{otherwise} \end{cases} \tag{2.60}$$

2.4.2 Policy Based Learning Method

In policy based learning, the policy, probability distribution of action, is the objective of learning Sutton et al. (2000). We have our policy $\pi_\theta(a|s)$ that has a parameter θ. This policy outputs a probability distribution of actions. So how can we know whether our policy good or not? The answer is that the policy can be seen as an optimization problem. For policy objective function, θ is the variable and $J(\theta)$ is the objective function. We can find the best θ to maximize a objective function.

The objective function of the policy gradient can be defined as follows Eq. 2.61.

$$J(\theta) = \sum_s d(s) \sum_a \pi_\theta(s, a) R_s^a \tag{2.61}$$

Here, $d(s)$ is probability distribution of state on policy $\pi_\theta(s,a)$, $J(\theta)$ can tell us how good our policy is; Policy gradient ascent can find the best policy parameters to maximize the sample of good actions.

RL problem involves manipulating probabilities, which are most often represented as log-probabilities. The log-ratio trick can helps us to solve stochastic optimization problem.

$$\nabla \log f(x) = \frac{1}{f(x)} \nabla f(x) \tag{2.62}$$

$$\nabla f(x) = f(x) \nabla \log f(x) \tag{2.63}$$

In a continuous environment, we can use the average reward for each time step to formulate the objective function, the idea is that we want to get the most reward at each time step.

$$\begin{aligned}
\nabla_\theta J(\theta) &= \nabla_\theta \sum_s d(s) \sum_a \pi_\theta(s,a) R_s^a && (2.64)\\
&= \sum_s d(s) \sum_a \nabla_\theta \pi_\theta(s,a) R_s^a && (2.65)\\
&= \sum_s d(s) \sum_a \pi_\theta(s,a) \frac{\nabla_\theta \pi_\theta(s,a)}{\pi_\theta(s,a)} R_s^a && (2.66)\\
&= \sum_s d(s) \sum_a \pi_\theta(s,a) \nabla_\theta \log \pi_\theta(s,a) R_s^a && (2.67)\\
&= E_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s,a) R_s^a] && (2.68)
\end{aligned}$$

Where, $\nabla_\theta \ln \pi_\theta(s,a)$ means the direction of policy update; R_s^a indicates the size of policy update, which is the immediate reward. From the aforementioned, we can conclude that:

- If R_s^a is high, it means that on average we took actions that lead to high rewards under the high positive reward condition.

- On the contrary, if R_s^a is low, we want to decrease the probabilities of the actions seen under the high negative reward condition. We want to increase the probability of taking these actions.

However, the previous formulation can not handle the long-term reward. So that it can be replaced the instantaneous reward R_s^a with state action value function $Q(s, a)$. The equation turns to be

$$\nabla_\theta J(\theta) = E_{\pi_\theta}[\nabla_\theta \log \pi_\theta(s, a) Q(s, a)] \tag{2.69}$$

$$\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta) \tag{2.70}$$

In policy gradient learning, we rely on full trajectories of an agent acting within an episode of the environment to compute the reinforcement signal. Given a trajectory, we produce a value estimate R_s^a for each step in the path by calculating a discounted sum of future rewards G_t for each step in the trajectory. The problem is that the policies we are learning are stochastic, which means there is a certain level of noise to account for. This stochasticity leads to variance in the rewards received in any given trajectory. Consequently, policy gradients suffer from high variance and low convergence Peters and Schaal (2006). The stochastic policy takes different action in different episodes. To reduce the variance caused by actions, we have to reduce the variance for the sampled rewards.

One of the most common approaches to reducing the variance of an estimate is to employ a baseline which is subtracted from the reward to produce a more stable value.

$$A_\pi(S_t, A_t) = Q_\pi(S_t, A_t) - V_\pi(S_t) \tag{2.71}$$

The advantage $A_\pi(s, a)$ is the difference between two estimates: the stable learned value function $V_\pi(s)$ and the discounted sum of future rewards G_t, or

$Q_\pi(s,a)$. Note that the learned value function $V_\pi(s)$ can be learned from a critic neural network. It is covered in detail in the next section.

Another advantage estimate approach, called the Generalized Advantage Estimate (GAE) Schulman et al. (2015), allows for an interpolation between Temporal difference (TD) learning and Monte-Carlo (MC) sampling using a lambda parameter. Roughly speaking, MC sampling wait until the return $G(S_t)$ can be obtained, until the end of the episode to determine the increment to V, while TD need to wait only until the next time step Sutton et al. (1998).

$$TDlearning : GAE(\gamma, 0) : \hat{A}_t = R_t + \gamma V(S_{t+1}) - V(S_t) \tag{2.72a}$$

$$MCsampling : GAE(\gamma, 1) : \hat{A}_t = \sum_{l=0}^{\infty} \gamma^l R_{t+l} - V(S_t) \tag{2.72b}$$

However, policy gradient involves a second-order derivative matrix which makes it not scalable for large scale problems. The computational complexity is too high for real tasks. Intensive research is done to reduce the complexity by approximate the second-order method.

In the study, we selected the proximal policy optimization (PPO) method, which is an extension of the policy gradient method. Instead of imposing a hard constraint, it formalizes the constraint as a penalty in the objective function. By not avoiding the constraint at all cost, we can use a first-order optimizer like the Gradient Descent method to optimize the objective. Even we may violate the constraint once a while, the damage is far less and the computation is much simple. We repurpose it to measure the difference between the two policies. We don't want any new policy to be too different from the current one. In its implementation, we maintain two policy networks. The first one $\pi_{\theta_{old}}(a_t|s_t)$ is the current policy that we want to refine. The second $\pi_{\theta(a_t|s_t)}$ is the policy that we last used to collect samples.

With the idea of importance sampling, we can evaluate a new policy with samples collected from an older policy Schulman et al. (2017). This improves sample efficiency. But as we refine the current policy, the difference between the current and the old policy is getting larger. The variance of the estimation will increase and we will make bad decision because of the inaccuracy. As a result, we synchronize the second network with the refined policy again. It will be introduced in the next section.

We use a ratio Eq. 3.14 between the old policy and new policy to measure how difference between two policies.

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} \tag{2.73}$$

We construct a new objective function to clip the estimated advantage function if the new policy is far away from the old policy. Our new objective function becomes:

$$L^{PPO}(\theta) = \hat{E}[\min(r_t(\theta)\hat{A}_t, \mathrm{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)] \tag{2.74}$$

If the probability ratio between the new policy and the old policy falls outside the range (1-ϵ) and (1-ϵ), the advantage function will be clipped.

2.4.3 Actor-Critic Method

So far we has focused on value based method such as Q-learning, and policy based method such as policy gradient. Actor-critic method combines the benefits of both methods. The learning agent of Actor-critic method has been split into two separate entities: the actor (policy) and the critic (value function). Roughly speaking, actor-critic method, utilize both an actor which defines the policy, and a critic (often a parameterized value estimate) which provides a more reduced variance reward signal to update the actor.

Actor-critic method uses TD method that has a separate memory structure to explicitly represent the policy independent of the value function. The policy structure is known as the actor, because it is used to select actions; and the estimated value function is known as the critic, because it criticizes the actions made by the actor.

Typically, the critic is a state-value function. After each action selection, the critic evaluates the new state to determine whether things have gone better or worse than expected. That evaluation is the TD error:

$$\delta_t = R_{t+1} + \gamma V(S_{t+1}) - V(S_t) \tag{2.75}$$

where $V(S_t)$ is the current value function implemented by the critic. If the TD error is positive, it suggests that the tendency to select should be strengthened for the future, whereas if the TD error is negative, it suggests the tendency should be weakened.

The actor is only responsible for generating an action, given the current state S_t.

2.5 Hardware-in-the-Loop Simulation

Hardware-in-the Simulation (HILS) is a technique, where physical part of a real system are substituted by a simulation. It is used for validating embedded real-time systems prior to their actual deployment.

The development of of a safety-related real-time system involves validation techniques ensuring sufficient trust in the reliability of the system. As part of validation, dynamic testing aims to remove errors within a system prior to its final deployment. However, it is often difficult to test a real-time system performing control tasks in its natural environment because the conditions of the environment circumvent an accurate observation of the real-time systems interface. Also, it may be very costly and time consume to establish certain test conditions in the physical environment. A comprehensive summary on concepts of regarding the design and analysis of real time system can be found in. With regard to the real-time system, a fundamental study on the concepts of HILS is presented in .

In this study, HILS environment for our system includes an actual control system and a simulation of the environment. HIL tesing is performed by connecting the simulator to the target control system, so that testing can be conducted in a controlled environment. The simulator acts as a virtual world for the control system by simulating the necessary equipment including system dynamics, sensors and actuators. Consequently, the simulator will respond in the same way as the actual system would in real operation on board.

2.5.1 Integrated Simulation Method

The HILS communication with the control system by sending and receiving signals. So that the communication interface is necessary. In this study, we adopt the robot operating system (ROS), which is widely used to develop hardware and control algorithms rapidly.

ROS is an open-source, meta-operating system for your robot. It provides the services you would expect from an operating system, including hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management. `http://wiki.ros.org/ROS/Introduction`

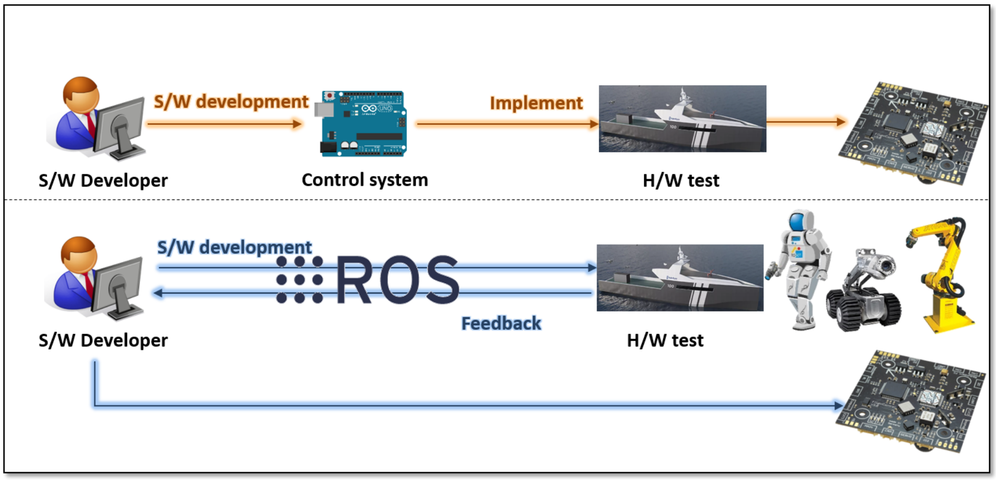


Figure 2.7: The benefit of use of ROS.

For the communication between the control system and the HIL simulator, an integrated simulation environment was constructed using the data communication interface of the ROS. In our application, the use of ROS allows us to easily integrate the control algorithms with the HIL simulator and with the real hardware in the ships. As shown in Fig. 2.7, the advantage of ROS is that it is possibility to directly develop algorithms on the control system and safely test them in a simulated scenario.

Fig. 2.8 describes the communication and features of ROS configuration.

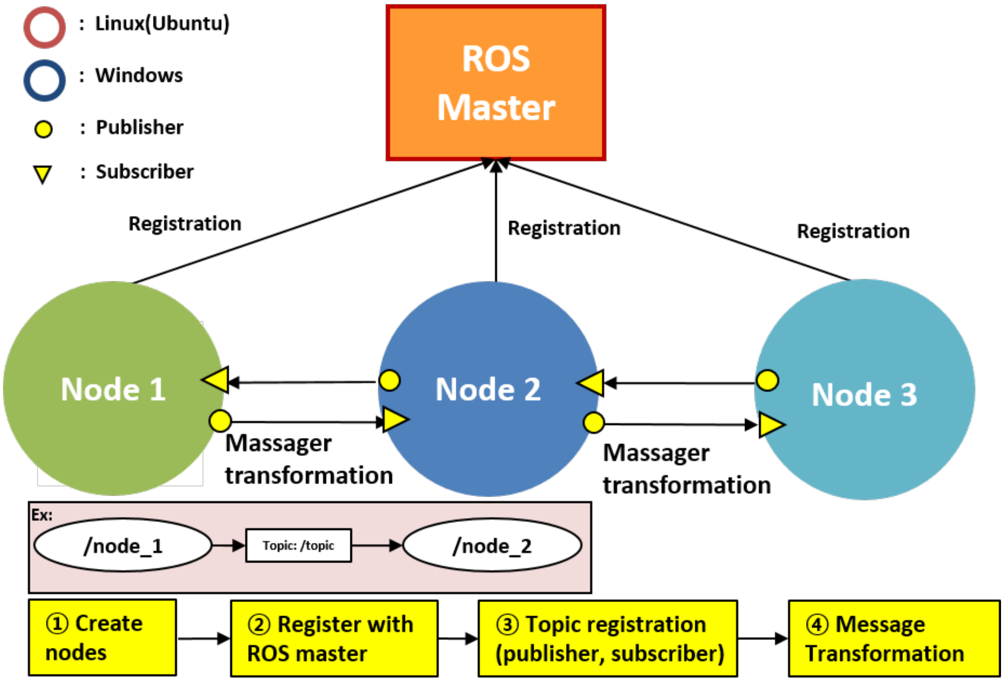


Figure 2.8: The components of ROS.

One of the primary purposes of ROS is to facilitate communication between ROS modules called nodes. These nodes represent the executable code and the code can reside entirely on one computer, or nodes can be distributed between computers or between computers and robots. With this simulation environment, the HIL simulator can be easily replaced by actual hardware, which is supported by the ROS. As a result, an experiment can be easily conducted in the real environment to validate the control algorithms.

Chapter 3

Path Following Method for Autonomous OSV

Autonomous ships rely heavily on guidance system in order to accomplish desired motion control scenarios such as object tracking, path following. Breivik and Fossen (2005).

3.1 Guidance System

3.1.1 Line-of-sight Guidance System

This section considers a 2-D control scenario where the autonomous OSV is assigned to follow a predefined path. Path following is a task of following the predefined path which is usually specified in terms of waypoints. Each waypoint is defined using coordinates (x_k, y_k) for $k = 1, 2, 3..$ For the autonomous OSV, this means it should pass through waypoint (x_i, y_i) with desired heading angle. A frequently used method for the path following is LOS guidance. To avoid large drift when switching at the desired heading angle, and to provide a proper desired heading angle to the controller, the commanded LOS heading is fed through a reference model. The diagram of the LOS guidance system is given in Fig. 3.1, where the LOS position P_{LOS} is the point along the path the the vessel should point at. It is located somewhere along the straight line connecting the current waypoint $P_k((x_k, y_k))$ and the next $P_k + 1((x_{k+1}, y_{k+1}))$. Let the ship's current position be located at the center of a circle with the radius of n times the ship length. The circle intersects the current straight line at two points where P_{LOS} is selected as the point closest to the next waypoint P_{k+1}.

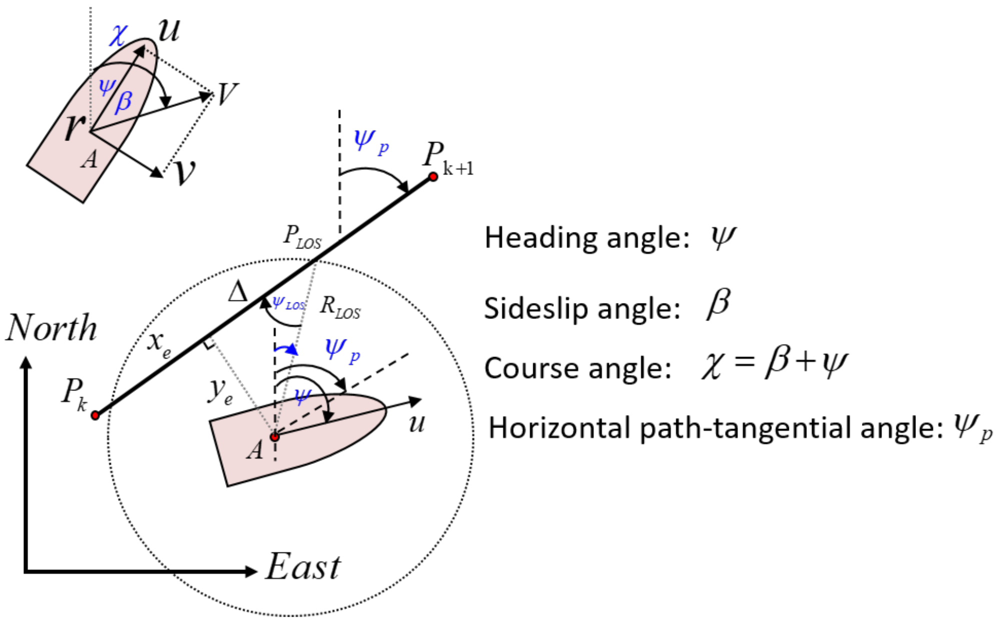


Figure 3.1: Diagram of LOS guidance geometry for a straight line.

Consider a straight line path defined by the two waypoints $P_k(x_k, y_k)$ and $P_{k+1}(x_{k+1}, y_{k+1})$, respectively; the path-tangential angle can be adjusted as follows:

$$\psi_p = \arctan\left(y_{k+1} - y_k, x_{k+1} - x_k\right) \tag{3.1}$$

Hence for a ship that is specified regarding the positions (x, y), the along-track error and cross-track error can be computed as the orthogonal distance to the path-tangential reference frame defined by the point P_k.

$$\begin{bmatrix} x_e \\ y_e \end{bmatrix} = \begin{bmatrix} \cos\psi_P & \sin\psi_P \\ -\sin\psi_P & \cos\psi_P \end{bmatrix} \begin{bmatrix} x - x_k \\ y - y_k \end{bmatrix} \tag{3.2}$$

One of the control objective for straight line path following becomes $\lim_{t\to\infty} y_e = 0$. Driving y_e to zero directs the velocity towards the intersection point P_{LOS} which corresponds to the desired direction. Based on the LOS guidance law, the desired course angle is separated into two parts:

$$\psi_d = \psi_P + \psi_{LOS} \tag{3.3}$$

$$= \psi_P + \arctan(\frac{-y_e}{\triangle}) \tag{3.4}$$

$$\psi_{LOS} = \arctan(\frac{-y_e}{\triangle}) \tag{3.5}$$

Where $\triangle$ represents the look-ahead distance and takes values between 1.5 and 2.5 of the ship length Fossen (2011). ψ_{LOS} ensures that the velocity is directed toward the point on the path.

In the presence of the environmental disturbances, the heading angle error ψ_e becomes:

$$\psi_e = \psi_d - \beta - \psi \tag{3.6}$$

By the above equations, the cross-track error and heading angle error can be explicitly stated by the following equation:

$$\begin{bmatrix} x_e \\ \psi_e \end{bmatrix} = \begin{bmatrix} -\sin\psi_P & \cos\psi_P & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x - x_k \\ y - y_k \\ \psi_P + \psi_{LOS} - \beta - \psi \end{bmatrix} \tag{3.7}$$

Where y_e and ψ_e are referred to the cross-track error and heading angle error respectively. The control objective of the path following is to drive the two errors to zero.

3.2 Deep Reinforcement Learning for Path Following System

This section presents the definitions and theoretical background of the controller design used in this study. The main objective of the study is to ensure that position of the ship converges to and follows the predefined path. The brain of the path following system is the controller. Generally, the controller measures the process variables concerning the analysis module of the autonomous OSV and directs to control command to the actuators to correct the error between the process variables and the desired value.

3.2.1 Deep Reinforcement Learning Setup

The path following problem is defined in the context of the sequential decision-making problem by considering the surrounding environment. During training, all the current process variables of the autonomous OSV can be observed and evaluated whether it arrive at the destination. Based on the observation space, self-play trials are conducted to determine the control strategy under various training processes. Once the training process is completed, the autonomous OSV is capable of automatically navigating predefined way and arriving at the destination under the commands of the controller.

At each time step t, the controller has access to an observation vector and computes a control command that drives the ship from the current position to the destination. Given the observation vector s_t, the autonomous OSV computes a control command, a_t, sampled from a stochastic policy $\pi_\theta(a|s)$ with the policy parameter θ.

$$\mathbf{a}_t \sim \pi_\theta(\mathbf{a}_t \mid \mathbf{s}_t), \tag{3.8}$$

As a result, the path following problem has been formulated as a sequential decision-making problem. Thus, the objective of the controller design is to find an optimal policy.

As a result, the sequential decision-making for the path following problem can be formulated as a Markov decision process (MDP) in an RL framework illustrated in Fig. 3.2.

The decision-maker (autonomous OSV), which is called an agent, executes an action in the environment, and the environment, in turn, yields a new state and reward. The terms 'agent', 'environment', and 'action' are used instead of 'autonomous OSV', 'analysis module', and 'control signal' further in this paper. More formally, the agent and environment interact at each of a sequence of time steps, $t = 0, 1, 2...,$ At each time step t, the agent receives states of the environment, $s_t \in \mathcal{S}$, where S is the set of states, $a_t \in \mathcal{A}(s_t)$, is the set of actions available in state s_t. One time step later, the agent received a numerical reward, $r_{t+1} \in \mathcal{R}$, and find itself in a new state s_{t+1}. The mapping from the states to actions is called a policy (denoted as π_θ), where $\pi_\theta(a|s)$ is the probability that $a_t = a$ if $s_t = s$ Sutton et al. (1998). Namely, the return $G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$, where $\gamma \in [0, 1]$ is the discount rate. The state value function $V_\pi(s) = E[r_t|s_t = s]$ is the expected return for following policy π_θ from the state s_t. The state action value function $Q_\pi(s, a) = E[r_t|s_t = s, a_t = a]$ is the expected return for selecting action a_t in state s_t and then following policy π_θ. (Sutton and Barto, 2015) As shown in Fig. 3.2, the policy can be formulated as a controller that observes states and applies actions to the agent (autonomous OSV). The aim of the agent is to find an optimal policy, which can maximize the sum of the rewards (return) received during the interaction with the environment. In this way, the autonomous OSV can follow the predefined path and avoid collision with the encountered OSVs.

As illustrated in Fig. 3.2, the policy can be formulated as a controller that observes states and applies actions to the autonomous OSV. which can maximize the sum of the rewards (return) received during the interaction with the environment.

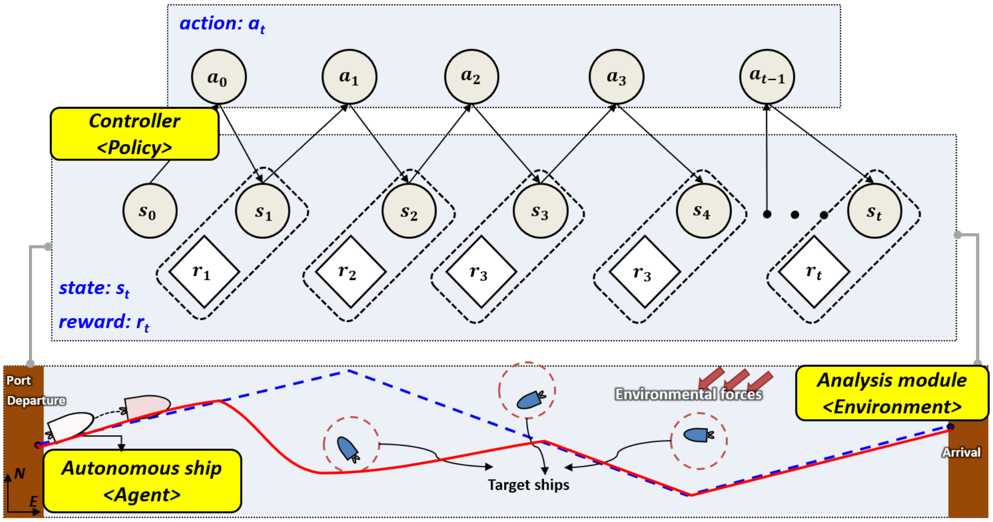


Figure 3.2: A configuration of RL framework for the path following system.

Observation Space

The observation vector of the path following system is defined as s_t^O, which denotes the autonomous OSV observation vector. We define the state as the information the agent receives about the environment at a given time step. In addition, we assume that the state space in this study are all observed. It can be expressed as:

$$s_t^O = [y_e,\, \psi_e,\, \dot{\psi}_e,\, \chi_e,\, \dot{\chi}_e,\, \|P_{goal} - P\|_2,\, \tilde{\phi},\, \delta,\, \dot{\delta},\, L] \tag{3.9}$$

The autonomous OSV state related to the path following system s_t^O, con-

Table 3.1: State observation in path following system

State	Description
y_e	Cross error
$\|P_{goal} - P\|_2$	Distance between the ship and the destination
ψ_e	Heading angle error
$\tilde{\phi}$	Relative angle between the course angle and the angle point to the destination
χ_e	Course angel error
δ	Rudder angel
$\dot{\psi}_e$	Angular velocity of heading angle error
$\dot{\delta}$	Rudder angular velocity
$\dot{\chi}_e$	Angular velocity of course angle error
L	Length of the autonomous OSV

sisting of 10 elements, which are illustrated in the Table 3.1. The y_e is the cross error, the ψ_e is the heading angle error, and $\dot{\psi}_e$ is the angular velocity of heading angle error and χ_e is the course angle error. Moreover, $\|P_{goal} - P\|_2$ represents the distance between the position of the autonomous OSV and the destination. $\tilde{\phi}$ is the relative angle between the course angle of the autonomous OSV and angle pointing to the destination from the ship. the rudder angle δ and rudder angular velocity $\dot{\delta}$ are also considered as a part of the state space. As the length of the autonomous OSV L may have a specific impact on the action space, it is also included into the state space.

Action Space

As was mentioned above, the state space consists of the autonomous OSV inertial coordinates, while the action is related to the rudder angle (Fig. 3.3).

We divide the permissible rudder angle δ into a set of three discrete values: $a \in [-20, 0, 20]$. Here, the maximum angle of the specified autonomous OSV δ_{max} is set to 20 degree. This chosen set of action is proven adequate and any given set up action can be used. As the autonomous OSV is an underactuated system which has been formulated in the previous section, the control vector can be expressed as $\tau = [\tau_x(t), \tau_y(t), \tau_N(t)]^T$.

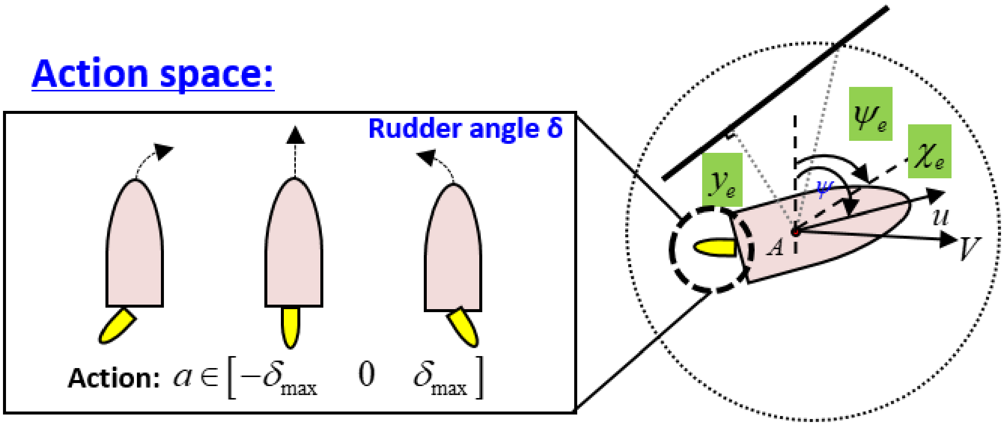


Figure 3.3: Action space in the path following system.

Reward Function Design

The reward function is computed as the sum of the rewards accumulated in each episode, where a reward is a measurement of action quality. The reward function can be specified to reward the agent for approaching its goal. It is designed to constraint the autonomous OSV to follow the predefined paths.

At first, a distance reward $R_{distance}$ is designed to guide the autonomous OSV to achieve the destination. This can be expressed mathematically as:

$$R_{\text{distance}} = -\lambda_{\text{distance}} ||P_{\text{goal}} - P||_2 \tag{3.10}$$

where P_{goal}, P are the position of the destination and current position of the autonomous OSV respectively, which can be found in Fig. 3.4. where $\lambda_{distance}$ refers to a hyper-parameter. When the ship approaches the destination, the more substantial distance reward value is imposed on the agent.

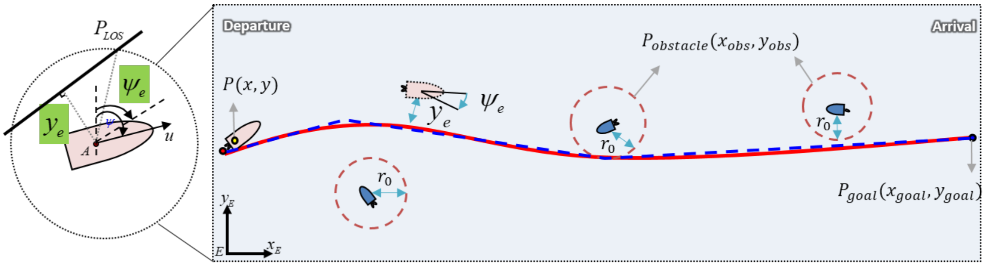


Figure 3.4: Action space in the path following system.

Moreover, to avoid the drift phenomenon, the linear velocity of sway v has to smaller than the surge velocity u. As a result, the drift reward function can be formulated as follows. Where the r_{drift} refers to the drift reward value in case of the drift phenomenon occurs.

$$R_{\text{drift}} = \begin{cases} -r_{\text{drift}} & \text{if } |u| < |v| \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

The course angle error ψ_e and the cross error y_e have been considered in another two reward function. In order to encourage the autonomous OSV to follow the predesigned path, the course angle error and the cross error between the autonomous OSV and the path have to converge to zero. When calculating the course angel error reward, within a small range $|\psi_e| < |\psi|$, we propose an exponential reward function to model it. If the heading angle error and the heading angular velocity error equal to zero, which means that there is no deviation between the autonomous OSV and path, the agent receives the maximum reward at the current time step.

$$R_{\text{heading}} = \begin{cases} \exp(-k_d({\psi_e}^2 + \dot{\psi}_e^2)) & |\psi_e| < |\psi| \\ -r_{\text{heading\_error}} & \text{otherwise} \end{cases} \tag{3.12}$$

The formation of the cross error reward function is similar to the heading angle reward function.

$$R_{\text{cross}} = \begin{cases} \exp(-k_c({y_e}^2 + \dot{y}_e^2)) & |y_e| < |y| \\ -r_{\text{cross\_error}} & \text{otherwise} \end{cases} \tag{3.13}$$

where k_d and k_c define the parameters of the exponential function which relate to its convergence speed. $r_{\text{heading\_error}}$ and $r_{\text{cross\_error}}$ are the positive values when the autonomous OSV deviates from the path at a relatively large angle.

3.2.2 Neural Network Architecture

The network consists of a critic network (value function) and a policy network (policy function). The critic network is used to predict a state value function for each state, and the policy network is used to predict the action. As shown in Fig. 3.5, to represent the policy network, we use a fully-connected (FC) multilayer perceptron with two hidden layers consisting of 64 and 32 hidden units with *tanh* nonlinearities outputting the probability over the action space. In the process of training, the state is transmitted to the neural network, and the agent selects and executes an action according to the policy with the highest probability. Training of the critic and policy networks (see Fig. 3.6) is performed by defining the surrogate loss functions for each network. Then, back-propagate gradients computed with the unified surrogate loss function are used to update the weights of the network. We refer to the network trained with this approach as Clipped PPO, as shown in Algorithm 1.

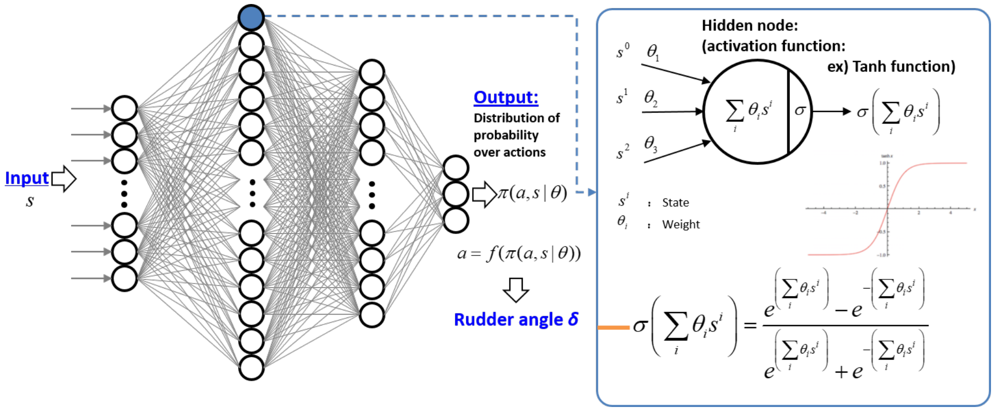


Figure 3.5: Flow diagram of neural network architecture

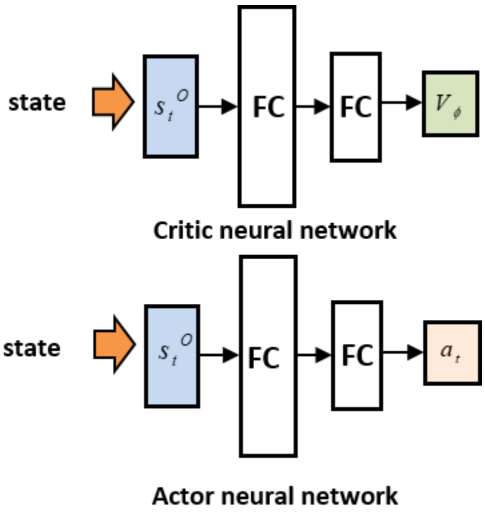


Figure 3.6: Network architecture. The observable state is fed into two fully-connected layers (FC), the outputs of critic network and policy network are a state value function (green) and an action (orange), respectively.

3.2.3 Training Process

In this section, we focus on learning path following policy which performs robustly and effectively in various scenarios. Policy gradient (PG) methods directly optimize the policy parameters θ by following the direction of the gradient of the expected return with respect of the policy parameters, which can be directly estimated from samples. However, traditional PG methods are sensitive to the choice of step size and have poor sample efficiency. In order to eliminate these disadvantages, a Proximal Policy Optimization (PPO) method is proposed to constrain the step size of the policy update during training. PPO is an extension of the policy gradient method. It uses a clipped surrogate objective function as the policy network's loss function, which is formulated as follows Schulman et al. (2017).

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} \tag{3.14}$$

$$L^{PPO}(\theta) = \hat{E}[\min(r_t(\theta)\hat{A}_t, \mathrm{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)] \tag{3.15}$$

In Eq. 3.14, $\pi_\theta(a_t|s_t)$ is the probability of the action under the current policy with the policy parameter θ, $\pi_{\theta_{old}}(a_t|s_t)$ is the probability of the action under the previous policy. Thus, $r_t(\theta)$ is the ratio of the probabilities under the current and previous policies, it is greater than 1 when the action is more probable for the current policy, while it is between 0 and 1 when the action is less probable for the current policy than the previous one.

However, if $r_t(\theta)$ trend to a big value, it means that it would lead to taking a big gradient steps. To deal with this, the objective function gains a penalty term, which $r_t(\theta)$ is clipped between $1-\epsilon$ and $1+\epsilon$. Therefore, it updates the policy conservatively by clipping the policy ratio to be within a small range around 1.0.

In Equation. 3.15, $\hat{E}_t$ denotes the empirical expectation over time steps, $\hat{E}_t$ is the estimated advantage at time t, ϵ is a hyper-parameter, which is usually set to 0.1 or 0.2. The value targets are calculated based on the generalized advantage estimation (GAE) advantages Schulman et al. (2015). It is defined as the difference between the state action value function and the state value function. As shown in Algorithm 1, at each iteration, the agent (autonomous OSV) collect T time steps of data (where T is much less than the episode length), and run the policy for T time steps. Then we construct the surrogate loss $L^{PPO}(\theta)$ on these sampled trajectories, the loss function is optimized with the Adam optimizer for E_π epochs. By taking a gradient ascent step on this loss with respect to the network parameters, the action will be led to obtain a higher reward. The state value function $V_\phi(s_t)$, used as a baseline to estimate the advantage $\hat{A}_t$, which is approximated with a neural network with parameters ϕ. We construct the mean squared error loss $L^V(\phi)$ for $V_\phi(s_t)$, and optimize it with Adam optimizer for E_v epochs. We update $\pi_\theta(a_t|s_t)$ and $V_\phi(s_t)$ independently and their parameters are not shared since we have found that using two separated networks will lead to a better results in practice. For completeness, the algorithm for iteratively updating policy and value function is given below:

Table. 3.2 presents the hyper-parameters used during the simulation runs.

Algorithm 1 PPO with an autonomous OSV.

Initialize policy network π_θ and value network $V_\phi(s_t)$.
for iteration $= 1, 2, ...,$ **do**
    Run policy π_θ for T timesteps, collecting $\{\mathbf{s}_t, r_t, \mathbf{a}_t\}$, where $t \in [0, T]$
    Estimate advantages using GAE Schulman et al. (2015), $\hat{A}_t = \sum_{l=0}^{T} (\gamma\lambda)^l \delta_t$, where $\delta_t = r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t)$
    break, if $T > T_{max}$
    $\pi_{old} \leftarrow \pi_\theta$
    // Update policy
    for $j = 1, ..., E_\pi$ **do**
        $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{old}(a_t|s_t)}$
        $L^{PPO}(\theta) = \sum_{t=1}^{T_{max}} \mathbf{min}(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)$
        Update θ with lr_θ by Adam Kingma and Ba (2014) w.r.t $L^{PPO}(\theta)$
    end for
    // Update value function
    for $k = 1, ..., E_V$ **do**
        $L^V(\phi) = -\sum_{t=1}^{T} (\sum_{t'>t} \gamma^{t'-t} r_{t'} - V_\phi(s_t))^2$
        Update ϕ with lr_ϕ by Adam w.r.t $L^V(\phi)$
    end for
end for

Table 3.2: The hyper-parameters of our training algorithm described in Algorithm

Parameter	Value
λ in line 4	0.95
γ in line 4 and 15	0.99
T_{max} in line 5	5120
E_ϕ in line 8	10
ϵ in line 10	0.2
lr_θ in line 11	$2e{-}5$
E_V in line 14	10
lr_ϕ in line 16	$1e{-}3$

3.3 Implementation and Simulation Result

3.3.1 Implementation for Path Following System

In this study, we consider an autonomous OSV that assigned to converge to a predesigned path specified by the path planner. Fig. 3.7 illustrates the implementation with the path following mission. To visualize the simulation result, the 3D visualization tool RViz was considered to be a suitable platform which provided by ROS. It is perfect for easily visualizing a simulated environmental including autonomous OSV, generated path, and other target ships. The calculation module interfaces with RViz to set the position, orientation of the autonomous OSV.

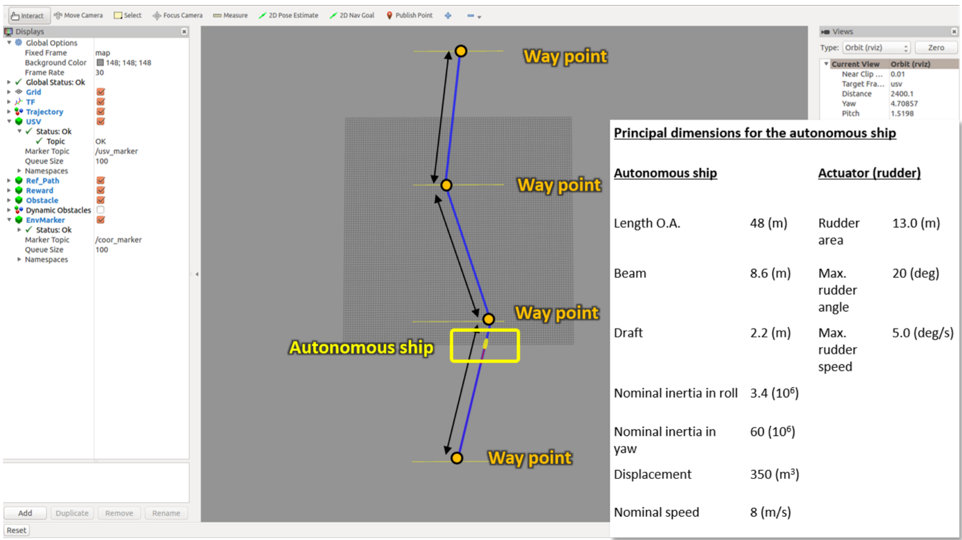


Figure 3.7: Simulation setup in phase one: path following and the principle dimensions for the autonomous OSV and the actuator.

In the real-world implementations, the environment is often dynamic and unpredictably changing. To handle this environment, the path planner is required to be able to generate various paths. We divided the training process into two phases, which accelerates the policy convergence and allows to get a higher reward.

At first, we train the autonomous OSV to follow the randomly generated path without any target ships, and this allows the autonomous OSV to improve the training speed in the presence of the target ships. The designed path considered here is composed of a collection of waypoints which randomly created as shown in Fig. 3.7. During the training process, the position and the number of waypoints are all created randomly. The principal dimensions of the autonomous OSV with its actuator is shown on the left side of Fig. 3.7. Pérez and Blanke (2002)

The objective of controlling the autonomous OSV is to follow the randomly generated path without deviation. In this task, the rudder angle applied to the autonomous OSV is limited to three choices: a positive angle of a fixed magnitude, a negative angle of the same magnitude, or zero. The rewards function, described in the previous section, are given on every time step until the destination (last waypoint) is reached, which finishing an episode. After finishing the given training iteration, the optimal policy can be obtained.

Table 3.3: Environmental conditions for path following scenario.

Case	Num. of way points	Wave condition			Current direction	Wind direction	Training & Testing		
		Direction [deg]	Height [m]	Period [sec]	[deg]	[deg]	Network model	Training iteration	Training time[h]
1-1	4							0	0
1-2		-	-	-	-	-	A	55	0.5
1-3								150	1.0
1-4		90	1.0	10	90	90	A	-	-
1-5		90	1.0	10	90	90	A-1	20	0.5
1-6		45	1.0	10	45	45	A-1	-	-
1-7	6	0	1.0	10	0	0	A-1	-	-

The proposed control algorithm was applied to path following in the real environment using the wind and current data to examine the effectiveness and practical utility of the approach. The simulation environment has been implemented using the python software package. The environment includes the dynamic model of the autonomous OSV, the randomly generated paths, and the simulated wave, wind, and current sea disturbances. The integrated time step was set to $dt = 0.1$ s. Several simulation cases considering a variety of environmental conditions were conducted, regarding the following variables: wave velocity direction: 0°, 45°, 90°, wind velocity direction: 0°, 45°, 90°, and current velocity direction: 0°, 45°, 90°. The wind and sea current velocity upper bounds are equal to 30.0 m/s and 1.0 m/s, respectively. The resultant disturbance forces and torque are collected in $\tau_{wind} = [F_{x_{wind}}, F_{y_{wind}}, F_{z_{wind}}]$ and $\tau_{wave} = [F_{x_{wave}}, F_{y_{wave}}, F_{z_{wave}}]$. The related parameters of the environmental forces used in the following cases are obtained from the Oil Companies International Marine Forum (OCIMF).

At each training iteration, the agent exploits the policy to generate trajectories until the maximum length of ($T_{max} = 5120$) time steps is reached. We then randomly select samples from the collected data. The selected sampled mini-batch (=64) is used to construct the surrogate loss function with is optimized with the Adam optimizer for E_π ($E_\pi = 10$) epochs. Average reward is computed as the sum of the rewards accumulated in each episode, where the path following reward functions follow the rules. An episode ends when the determination is reached, or the ship is too far away from the path.

3.3.2 Simulation Result

Simulation Result of 3 Step Rudder Angle

Fig. 3.8, Fig. 3.9, Fig. 3.10 illustrate the simulation results of the position of the autonomous OSV following the predefined path in Case 1-1, Case 1-2, and Case 1-3, respectively. These simulation results represent the pre-training, on-training with the iteration of 55, and post-training with the iteration of 150 respectively. The network model A was trained without the environmental forces. During the training, the paths are generated randomly by connecting four way-points. The initial heading angle of the autonomous OSV is also defined randomly. According to the simulation results, the capability of successfully following the path and reaching the destination is apparent.

The performance of the rudder angle in Case 1-3 is depicted in Fig. 3.11. It shows that when passing through each waypoint, the rudder angle is set to be maximum.

The following plot shows the average total reward during the path following training. We can find that the reward increases smoothly. The Case 1-1, Case 1-2 and Case 1-3 are signed in Fig. 4.13. However, when the environmental forces including wave, wind and sea current acting on the autonomous OSV (Case 1-4), we tested the trained model A. The simulation result of Case 1-4 is shown in Fig. 3.13, which represents that the autonomous OSV deviates from the designed path regularly so that it fails to follow the path. The corresponding result of rudder angle is depicted in Fig. 3.14.

Based on the network model A, we continuously train it under the same environmental conditions in Case 1-4. As a result, we got the trained network model A1 after 20 iterations. Fig. 3.15 and Fig. 3.16 in Case 1-5 show the simulation results using the network model A-1; the autonomous OSV can followed the designed path successfully under the environmental forces.

The environmental conditions, such as the wave amplitude, period, and direction, current and wind direction can induce different motions of the autonomous OSV. In order to evaluate the effectiveness of network model A1 under the various environmental forces, Here, we implement the Case 1-6 concerning with the following environmental conditions: wave direction (45°), current direction (45°), and wind velocity direction (45°). Fig. 3.17 and Fig. 3.18 show the simulation results of the Case 1-6 without training.

In Case 1-7, the wave, current and wind directions are equal to 0 degrees, and the training paths are more complicated than the previous Cases by adding to six waypoints. As shown in Fig. 3.17, the autonomous OSV can successfully follow the designed path without training. According to these simulation results, we can conclude that one of the advantages of the proposed method is its excellent performance of adaptation in the unknown environmental disturbance.

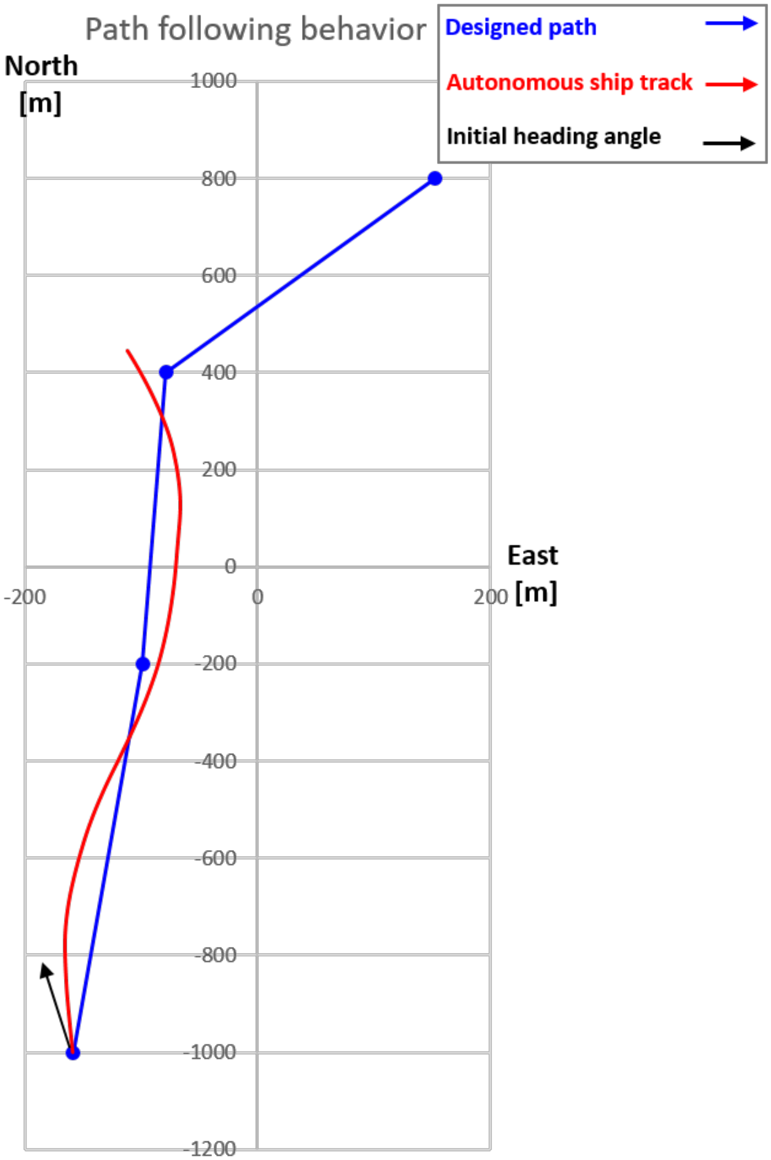


Figure 3.8: Graphs of the position of the autonomous OSV in Case 1-1. The result is calculated by the model A without environmental forces before training.

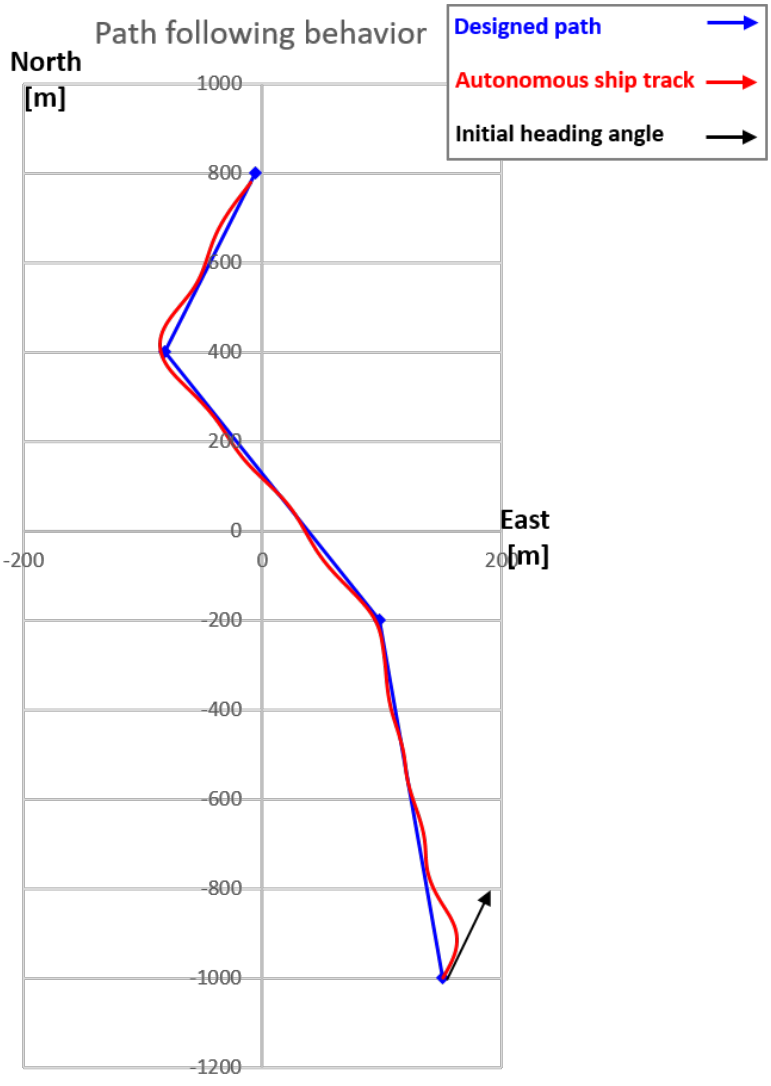


Figure 3.9: Graphs of the position of the autonomous OSV in Case 1-2. The result is calculated by the model A without environmental forces before training.

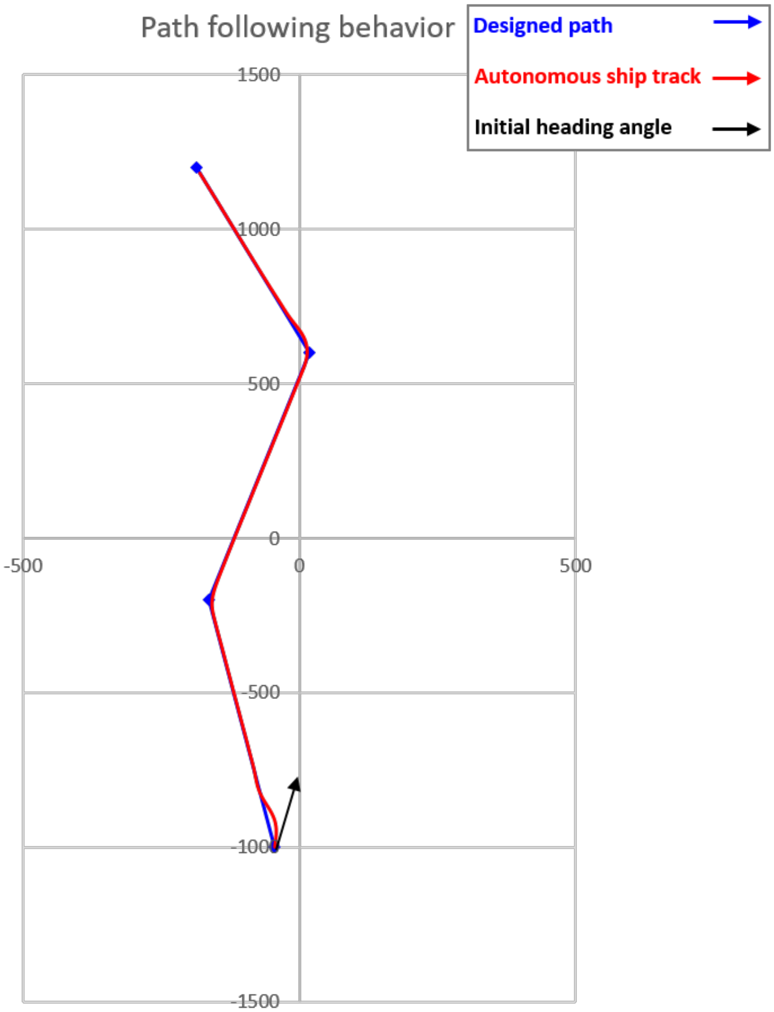


Figure 3.10: Graphs of the position of the autonomous OSV in Case 1-3. The result is calculated by the model A without environmental forces before training.

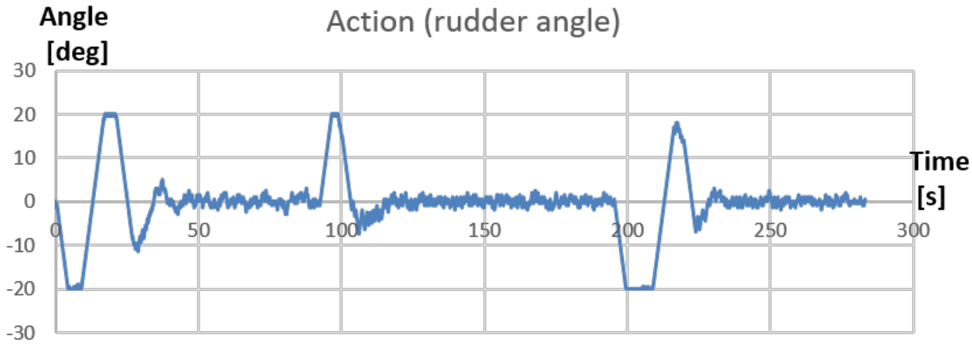


Figure 3.11: Graphs of rudder angle of the autonomous OSV in Case 1-3.

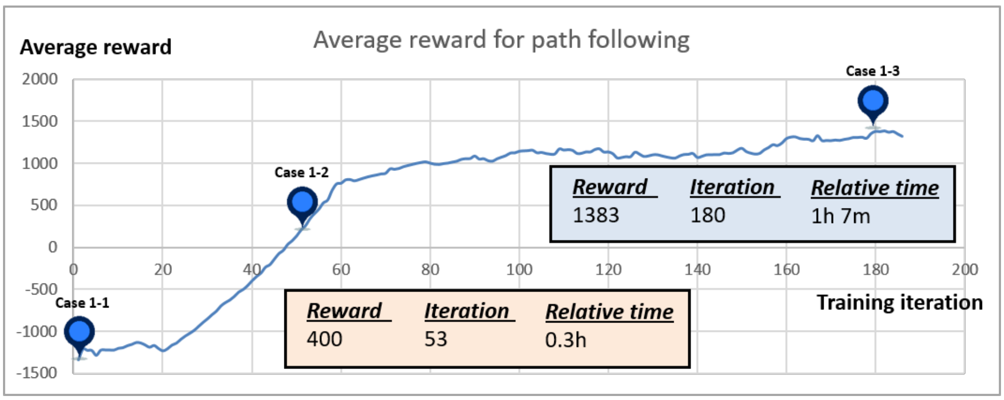


Figure 3.12: Average reward during training.

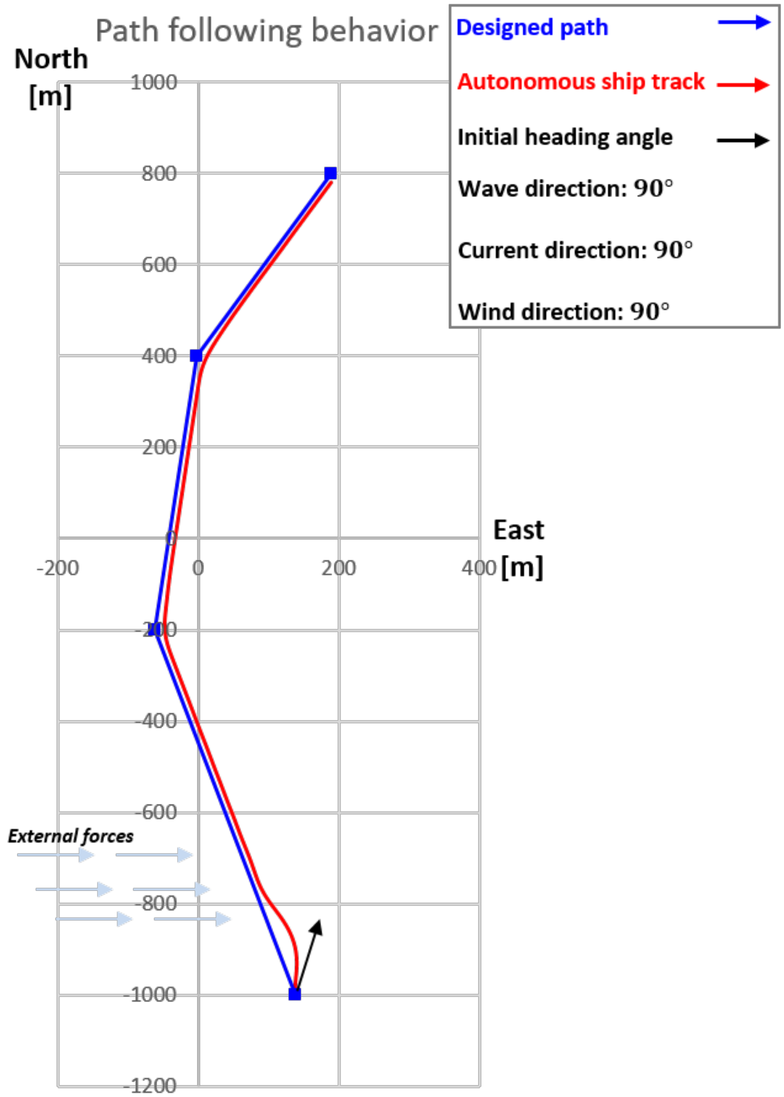


Figure 3.13: Graphs of the position of the autonomous OSV on a designed path in Case 1-4. The result is calculated by the model A with environmental forces before training.

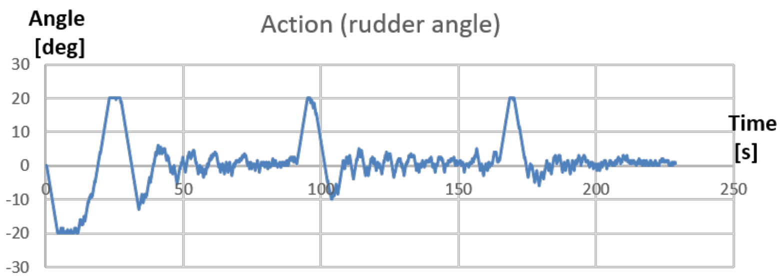


Figure 3.14: Graphs of rudder angle of the autonomous OSV in Case 1-4.

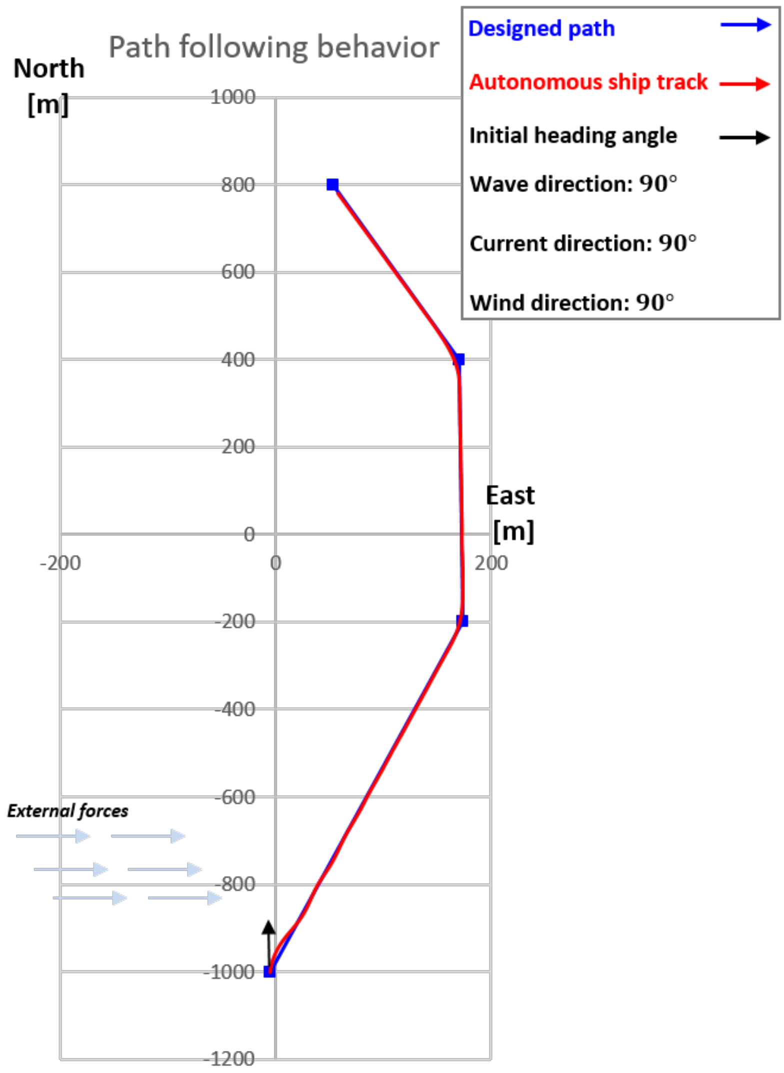


Figure 3.15: Graphs of the position of the autonomous OSV on a designed path in Case 1-5. The result is calculated by the model A with environmental forces before training.

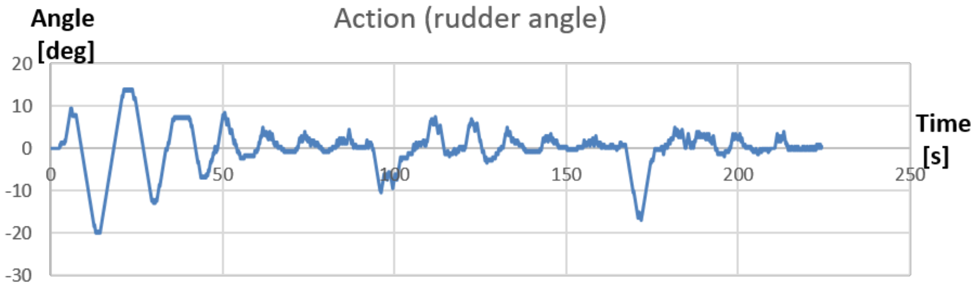


Figure 3.16: Graphs of rudder angle of the autonomous OSV in Case 1-5.

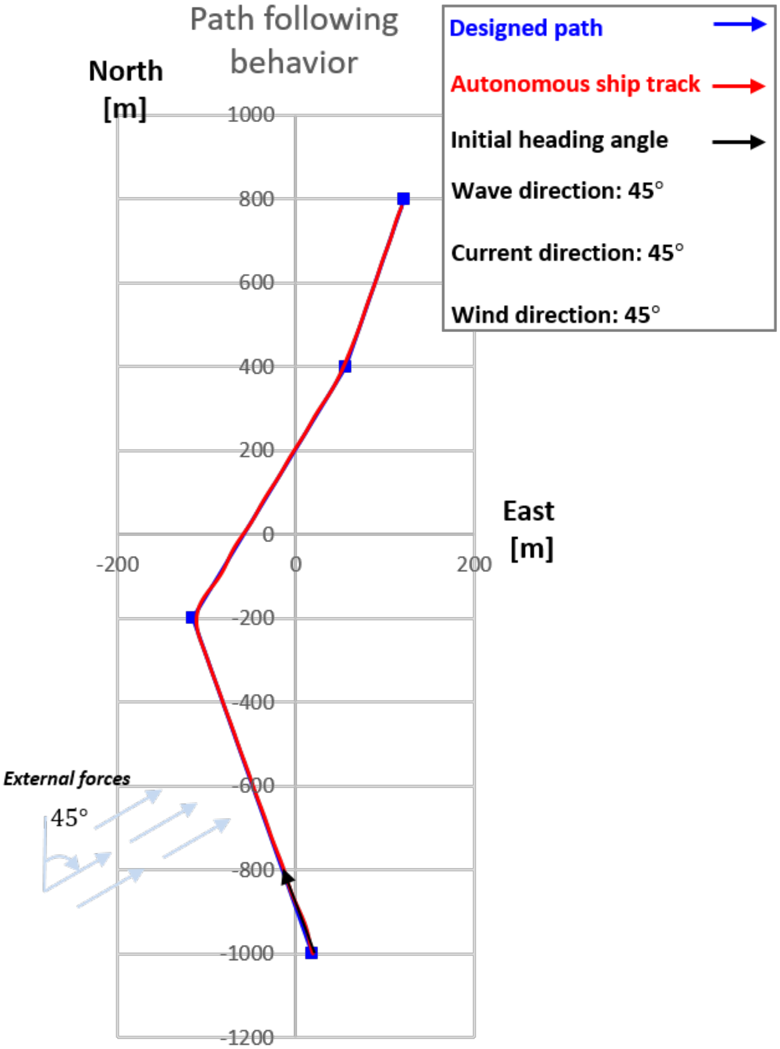


Figure 3.17: Graphs of the position of the autonomous OSV on a designed path in Case 1-6. The result is calculated by the model A-1 with environmental forces.

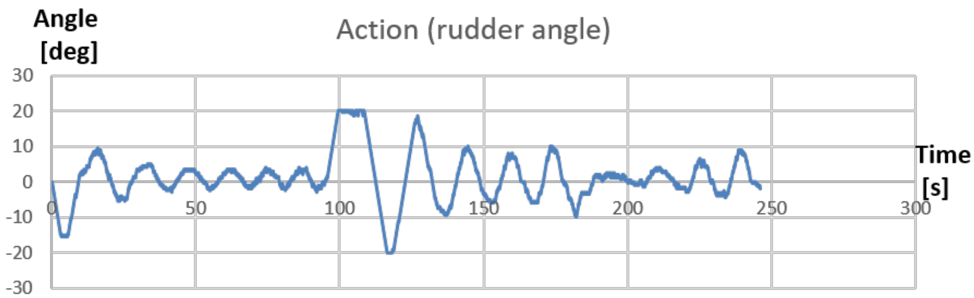


Figure 3.18: Graphs of rudder angle of the autonomous OSV in Case 1-6.

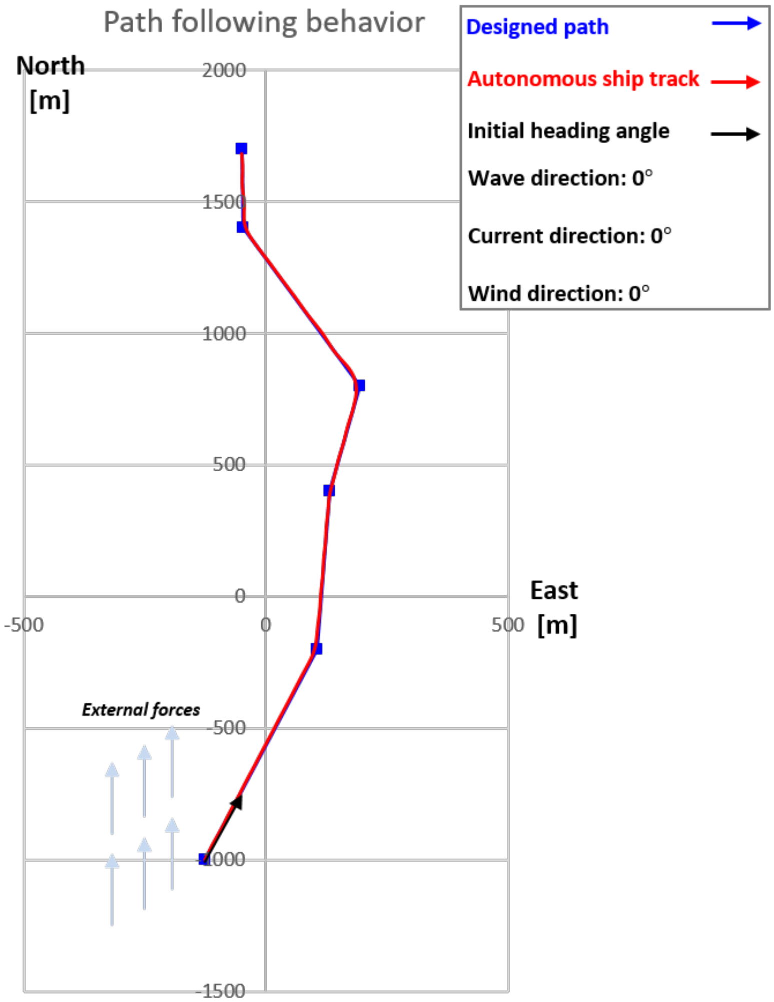


Figure 3.19: Graphs of the position of the autonomous OSV on a designed path in Case 1-7. The result is calculated by the model A-1 with environmental forces.

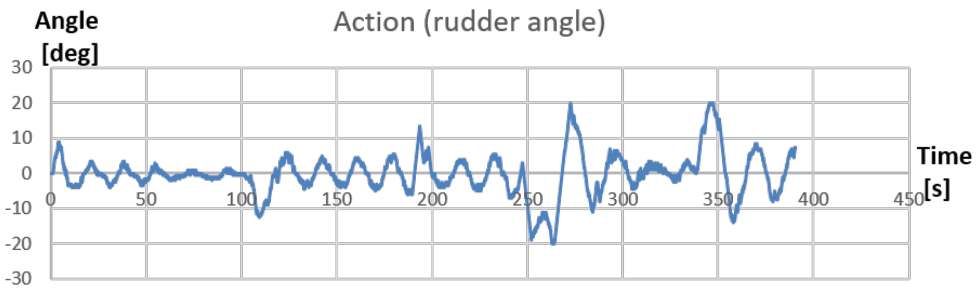


Figure 3.20: Graphs of rudder angle of the autonomous OSV in Case 1-7.

Table 3.4: Comparison cases of path following problem according to 3 step rudder angle and 5 step rudder angle

Case	Num. of way points	Training & Testing	
		Mean cross error [m]	
1	4	1.12	1.26
2	6	1.76	1.91
3	6	1.76	1.81

Simulation Result of 5 Step Rudder Angle

To evaluate the 3 step action performance, we subdivided the action command from a set of three discrete values (3 step) into a set of five discrete values (5 step). The action that is related to the rudder angle becomes a set of three discrete values $a \in [-20, -10, 0, 10, 20]$. Furthermore, we compared the simulation results of path following according to 3 step rudder angle and 5 step rudder angle in Table. 3.4.

Fig. 3.21 and Fig. 3.23 illustrate the simulation results of the position of the autonomous OSV following the predefined path in Case 1 and Case 2. These simulation results represent the 3 step rudder angle and 5 step rudder angle, respectively.

However, the mean cross error of the 3 step rudder angle result is smaller than the 5 step rudder angle, that is mean the performance of the 3 step rudder angle result is better than 5 step rudder angle. Fig. 3.22 and Fig. 3.24 represent the corresponding rudder angle tendency.

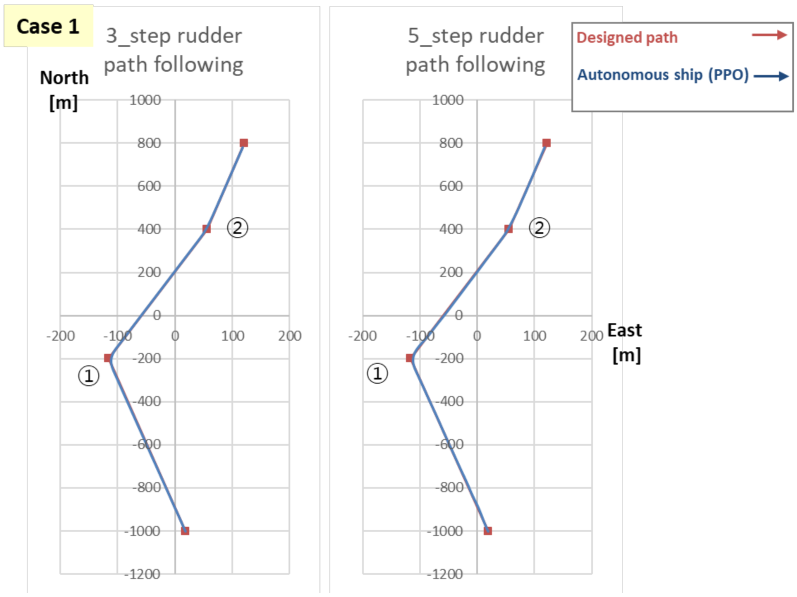


Figure 3.21: Comparison result of path following according to 3 step rudder angle and 5 step rudder angle.

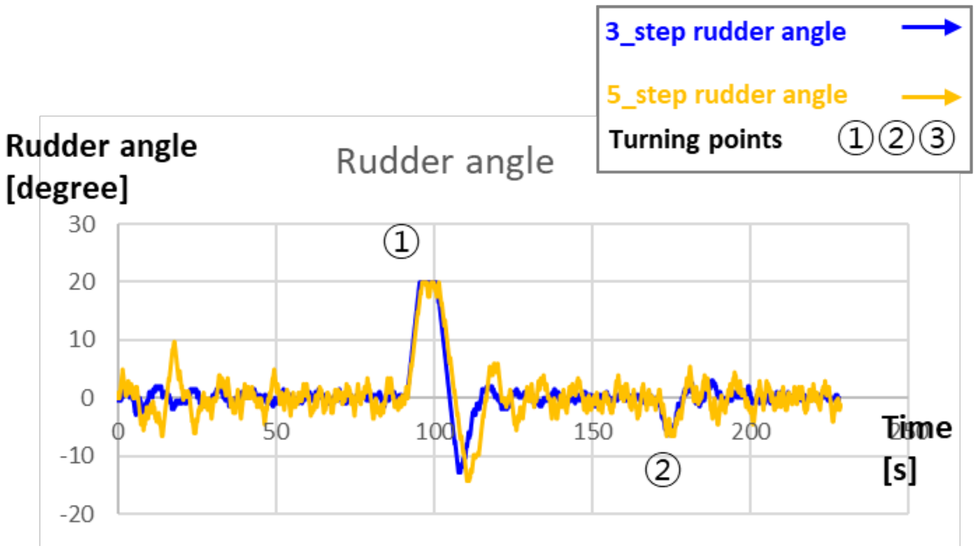


Figure 3.22: Comparison result of rudder angle.

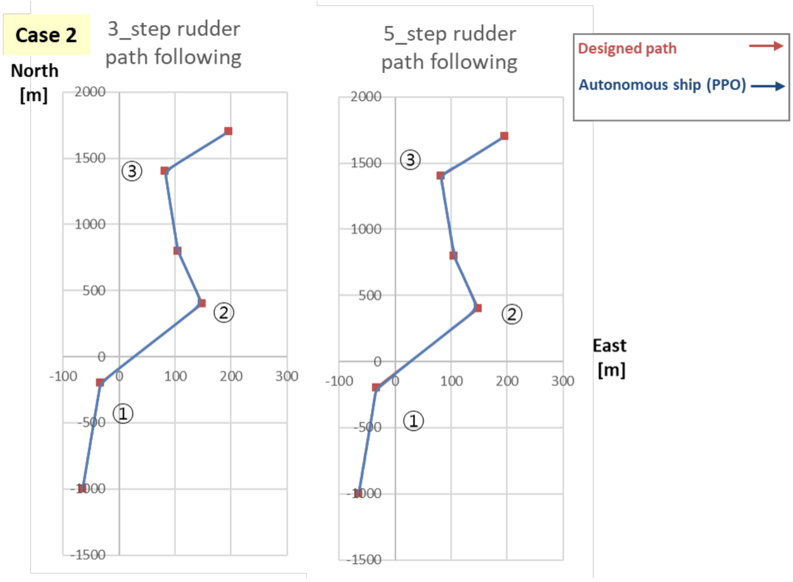


Figure 3.23: Comparison result of path following according to 3 step rudder angle and 5 step rudder angle.

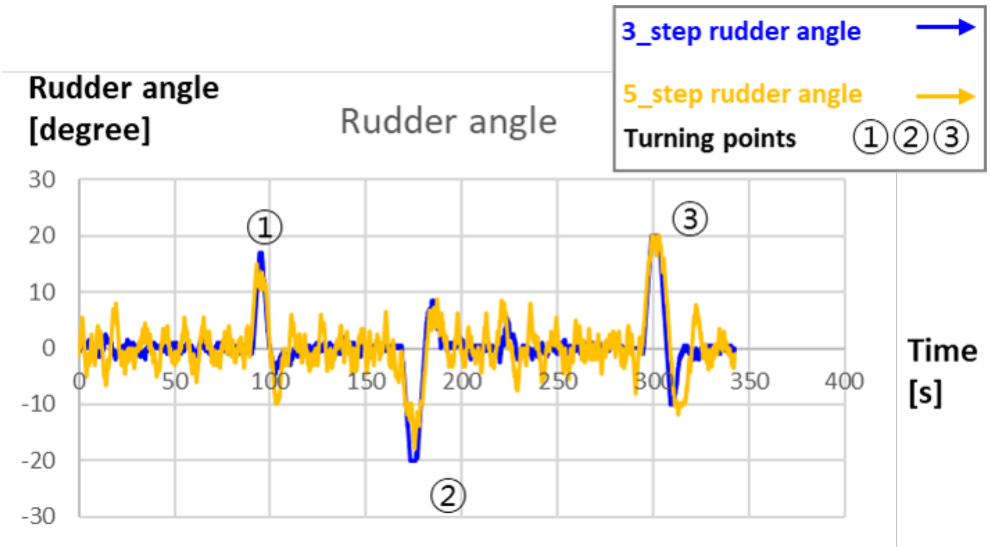


Figure 3.24: Comparison result of rudder angle.

Since the training iteration between the 3 step rudder angle and 5 step rudder angle has the difference, thus, we continually train the neural network model of 5 step rudder angle. As shown in Fig. 3.25, the average reward converge to a constant value eventually. Using the retrained neural network model, we compare the rudder angle between 3 step rudder angel result and 5 step rudder angel result. According to the Fig. 3.26, we can conclude that the rudder angel tendency of 3 step rudder angel and 5 step rudder is similar to each other when the training iteration is the same.

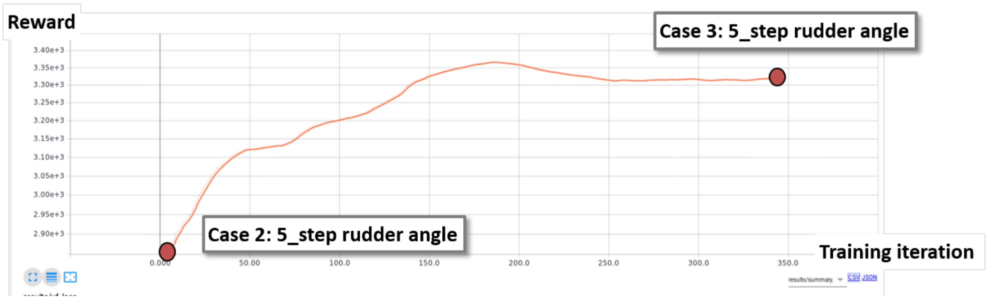


Figure 3.25: Continue training the 5 step rudder angle neural network model.

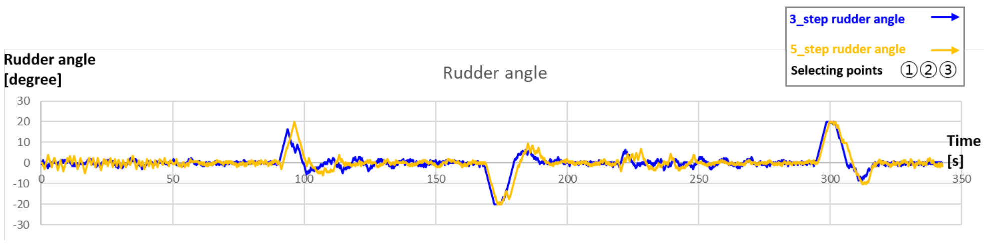


Figure 3.26: Comparison results of rudder angle.

Table 3.5: Environmental conditions of comparison scenarios

Case	Path type	Wave condition			Current direction	Wind direction	PPO		PID (fully tuned)	
		Direction [deg]	Height [m]	Period [sec]			Model	Mean cross error	Model	Mean cross error
1	CURVE	-	-	-	-	-	A	1.14	B	3.26
2		45	1.0	10	45	45	A	6.11	B	9.37
3		45	1.0	10	45	45	A_1	0.91	B_1	7.03
4		90	1.0	10	90	90	A_1	0.95	B_1	7.28

3.4 Comparison Results

3.4.1 Comparison Result of PPO with PID

Path following system for autonomous OSV are predominately implemented using the Proportional, Integral, Derivative (PID) control systems. PID have demonstrated close-to-ideal performance in many circumstances.

Case 1 is the simulation result of PPO model A and PID model B under the condition that there are no environmental forces(Fig. 3.27). As a result of simulation with PPO model A, the mean cross error between the autonomous navigation ship and the set path was 1.14 meters, and the simulation with the fully tuned PID model B resulted in a mean cross error of 3.26 meters.

However, the trained PPO model A and tuned PID model B were used for the simulation of path following when the environment forces, including wind, wave, and sea current, were taken into account. The simulation of Case 2. As a result of simulating with PPO model A, the mean cross error is 6.11 meters and the simulation result of PID model B shows that the mean cross error is 9.37 meters. As shown in the simulation results, the learned / tuned PPO and PID models in the absence of environmental external forces performed not well

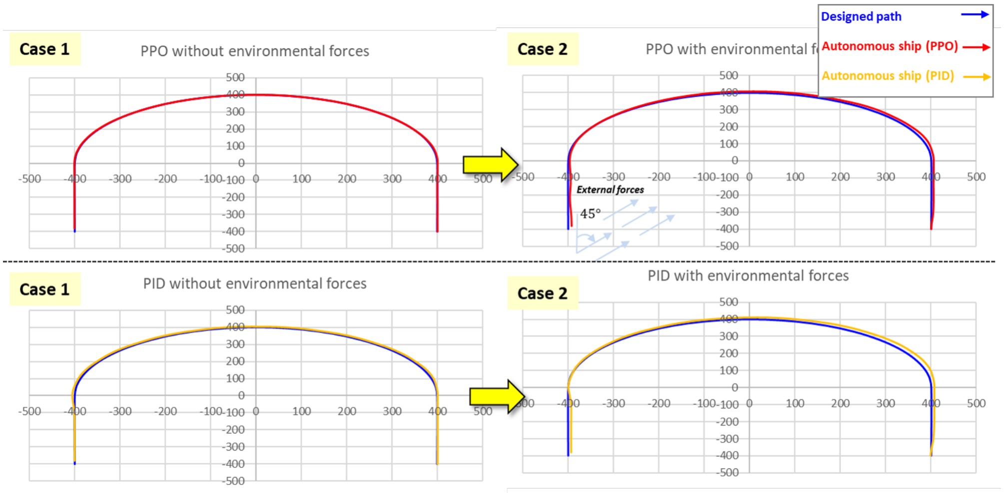


Figure 3.27: Graphs of the comparison result of PPO and PID in Case 1 and Case 2. The result is calculated by the model A and B without environmental forces.

followed in Case 2 given the environmental external force. It can be seen that the autonomous OSV deviates from the predefined path and all failed to follow the path.

Based on training model A and tuned model B, we continuously trained / tuned it under similar environmental conditions in Case 3. As a result, we obtained the trained PPO model A-1 and PID model B-1. Fig. 3.28 shows the simulation results using PPO model A-1 and PID model B-1 in Case 3; one can see that the autonomous OSV follows the designed path successfully under the environmental forces. In the context of the 45° environmental forces (Case 3), the mean cross error of PPO model A-1 is 0.91 meters and the mean cross error of PID model B is 7.03 meters.

To evaluate the effectiveness of PPO model A-1 and PID model B-1 under various environmental forces, the following environmental conditions were set in

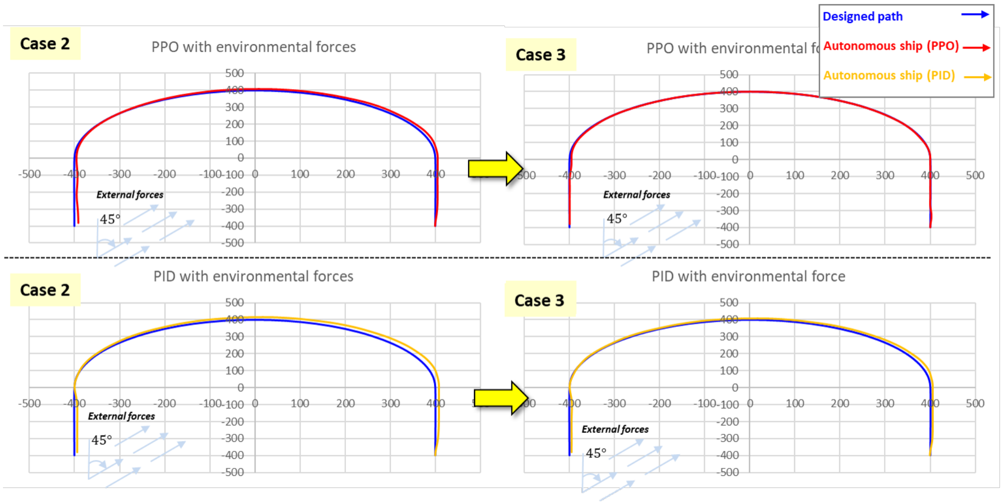


Figure 3.28: Graphs of the comparison result of PPO and PID in Case 2 and Case 3.

Case 4: wave direction (90°), current direction (90°), and wind velocity direction (90°).

According to these simulation results, we can conclude that one of the advantages of the proposed method is its excellent performance in the unknown environmental disturbances. When the objective exposed to unperdictable and harsh environments (e.g. wind, current, and wave, etc). However, a PID controller can be far from optimal Maleki et al. (2016) Koch et al. (2018).

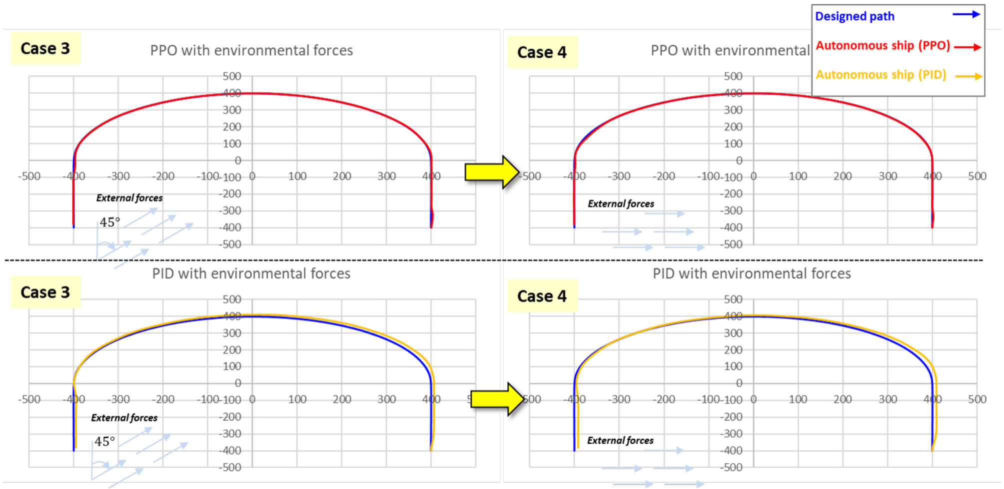


Figure 3.29: Graphs of the comparison result of PPO and PID in Case 3 and Case 4.

3.4.2 Comparison Result of PPO with Deep Q-Network (DQN)

The development of intelligent way point navigation control is an active area of research. In RL, an agent is given a reward for every action it makes in an environment with the objective to maximize the rewards over time.

We compare the performance of PPO controller with that of value based RL algorithm DQN controller. Table 3.6 shows the environmental conditions of comparison scenarios.

Table 3.6: Environmental conditions of comparison scenarios

Case	Num. of way points	Wave condition			Current direction	Wind direction	PPO		DQN	
		Direction [deg]	Height [m]	Period [sec]			Model	Mean cross error	Model	Mean cross error
1	4	45	1.0	10	45	45	A_1	1.14	C	3.26
2	6	0	1.0	10	0	0	A_1	6.11	C	9.37

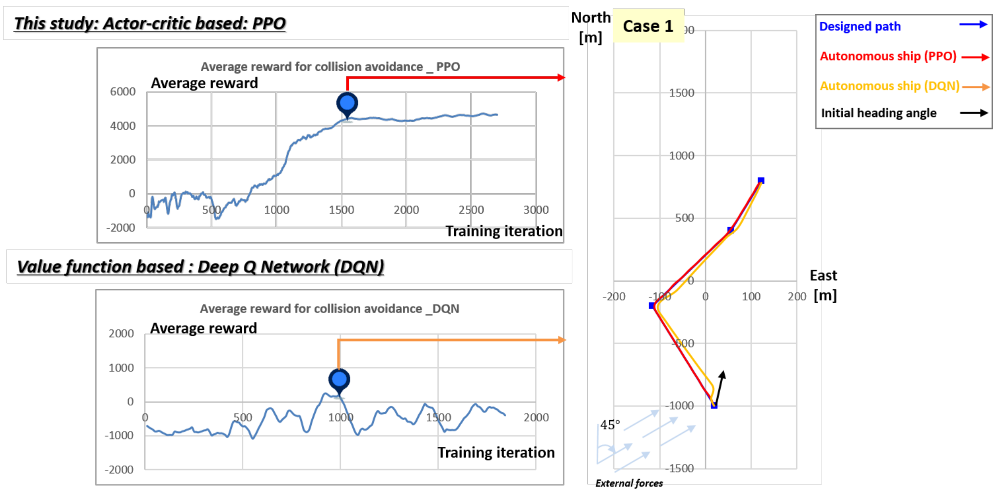


Figure 3.30: Graphs of the comparison result of PPO and DQN in Case 1.

Fig. 3.30 shows the average reward of PPO and DQN for path following in

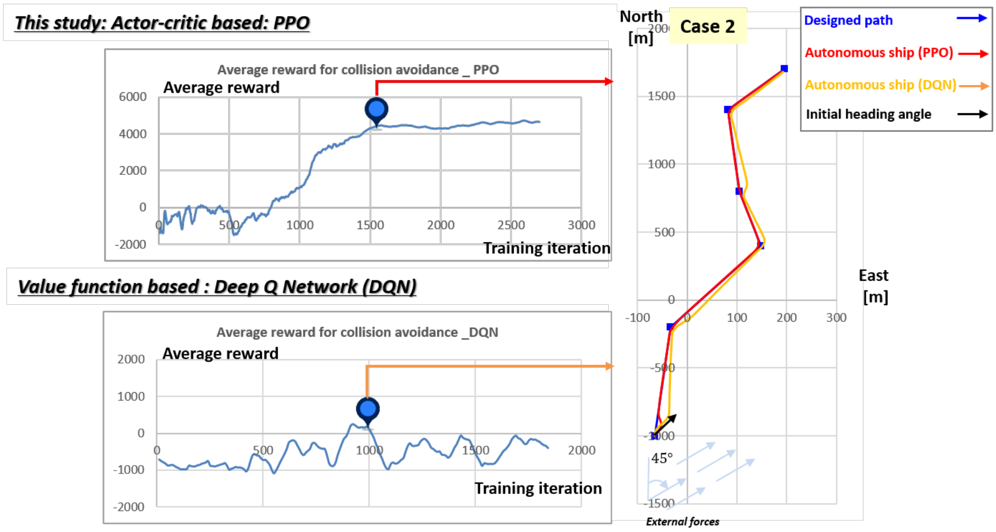


Figure 3.31: Graphs of the comparison result of PPO and DQN in Case 2.

Case 1. The average reward converges to a maximum value until 1580 iterations. Fig. 3.31 shows the average reward of PPO and DQN for path following in Case 2. The average reward converges to a maximum value until 1000 iterations.

In this study, we study the accuracy and prediction of navigation control provided by intelligent controller trained using state-of-the art RL algorithm: PPO and DQN. We then compare the performance of the RL controller with that of a PID controller. Our evaluation finds that 1) controller trained using PPO outferform PID controller and capable of exceptional performance. 2) controller trained using PPO outperform value-based DQN algorithm.

Chapter 4

Collision Avoidance Method for Autonomous OSV

4.1 Deep Reinforcement Learning for Collision Avoidance System

4.1.1 Deep Reinforcement Learning Setup

As illustrated in Fig. 4.1, the agent (autonomous OSV) and environment interact at each of a sequence of discrete time steps. At each time step t, the agent has access to the observation vector and computes the collision-free control command that drives the ship from the current position to the destination.

In this study, we define the action as the rudder angle, which will change the direction of the ship. One time step later, the agent of the ship receives a reward and find itself in a new state.

The policy can be formulated as a controller that observes states and applies actions to the agent. The aim of the agent is to find an optimal policy, which can maximize the sum of the rewards (return) received during the interaction with the environment. In this way, the autonomous OSV can follow the predefined path and avoid collision with the encountered ships.

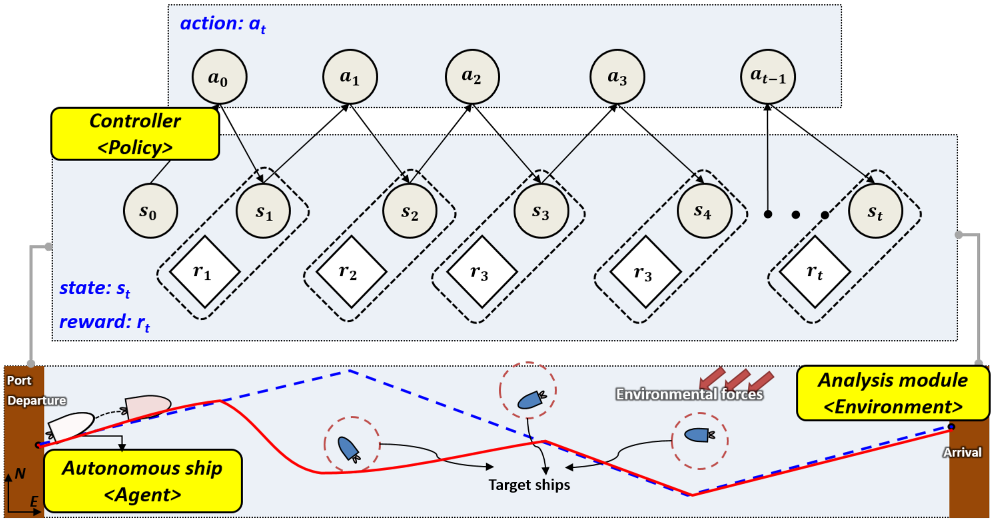


Figure 4.1: Simulation setup in phase two: collision avoidance with three target ships.

Observation Space

We define a state as the information the agent about the environment at a given time step. The observation vector of the system is divided into two parts: s_t^O and s_t^T. where s_t^O denotes the autonomous OSV observation vector and s_t^T is the observation vector related to the target ships.

As mentioned in the previous chapter, the state s_t consists of the state of the autonomous OSV observed itself s_t^O and s_t^T relevant to the target ships. It can be expressed as:

$$s_t^O = [y_e,\, \psi_e,\, \dot{\psi}_e,\, \chi_e,\, \dot{\chi}_e,\, \|P_{goal} - P\|_2,\, \tilde{\phi},\, \delta,\, \dot{\delta},\, L] \tag{4.1}$$

$$s_t^T = [P_{obstacle_i},\, V_{obstacle_i},\, \|P - P_{obstacle_Pi}\|_2,\, \|\chi - \chi_{obstacle_Pi}\|_2,\, l_i] \tag{4.2}$$

The path following related state s_t^O has been introduced in the previous chapter. The observation vector of the target ships s_t^T, consist of 5 elements as shown in Fig. 4.2. It contains positions $P_{obstacle_i}$ and velocities $V_{obstacle_i}$ of the target ships in the local frame attached to the autonomous OSV. The relative distances between the autonomous OSV and the target ships, $\|P - P_{\text{obstacle}_P i}\|_2$, and the relative angles between the autonomous OSV and the target ships $\|\chi - \chi_{\text{obstacle}_P i}\|_2$ have been considered. In addition, the lengths of the target ships l_i are also contained in the state space, where i represents the number of the target ships l_i. The state s_t are summarized in the following Table. 4.1.

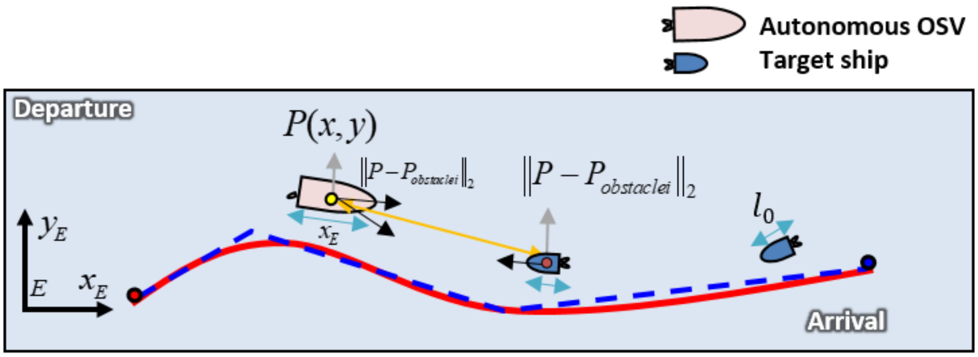


Figure 4.2: Target ships related state definition.

Action Space

We have considered that state space consists of the autonomous OSV inertial coordinates, while the action is related to the direction of the rudder angle. We divide the permissible rudder angle δ into a set of three discrete values: $a \in [-20, 0, 20]$. This chosen set of action is proven adequate and any given set up action can be used.

Table 4.1: State observation in collision avoidance system

State	Description
y_e	Cross error
$\|P_{goal} - P\|_2$	Distance between the ship and the destination
ψ_e	Heading angle error
$\tilde{\phi}$	Relative angle between the course angle and the angle point to the destination
χ_e	Course angel error
δ	Rudder angel
$\dot{\psi}_e$	Angular velocity of heading angle error
$\dot{\delta}$	Rudder angular velocity
$\dot{\chi}_e$	Angular velocity of course angle error
L, l_i	Length of the autonomous OSV and the target ships' length
$P_{obstacle_i}$	Target ship's position
$V_{obstacle_i}$	Target ship's velocity
$\|P - P_{obstacle_P i}\|_2$	Relative distance between the autonomous OSV and the target ships
$\|\chi - \chi_{obstacle_P i}\|_2$	Relative course angles between the autonomous OSV and the target ships
$\|P - P_{obstacle_P i}\|_2$	Relative distance between the autonomous OSV and the target ships

Reward Function Design

The objective of autonomous OSV is to follow the predefined path while avoids collision with the target ships. So the reward functions for path following have to be taken into consideration.

Besides the path following reward functions, when the autonomous OSV collides with the other target ships in the range of a circle with radius $r0$, it is penalized by the collision reward $r_{collision}$. Where $P_{obstacle_i}$ the current position of the target ships.

$$R_{\text{collision}} = \begin{cases} -r_{\text{collision}} & \text{if } |P - P_{obstacle_i}| < r_0 \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

As the autonomous OSV has to avoid the target ships in compliance with COLREGs, the related reward function R_{COLREGs} has to be added:

$$R_{\text{COLREGs}} = \begin{cases} r_{\text{COLREGs}} & \text{if turn right} \\ -r_{\text{COLREGs}} & \text{otherwise} \end{cases} \tag{4.4}$$

4.1.2 Neural Network Architecture

Similar with the path following problem, the network of collision avoidance scenario consists of a critic network (value function) and a policy network (policy function). The critic network is used to predict a state value function for each state, and the policy network is used to predict the action. To represent the policy network, we used a fully-connected (FC) multilayer perceptron with two hidden layers, consisting of 64, 32 hidden units with tanh nonlinearities, outputting the probability over the action space. When training, the state inputs to the neural networks, and the agent selects and executes an action according to the policy with the highest probability. As shown in Fig. 4.3, training both the critic network and policy network by defining surrogate loss functions for each network. Then back-propagate gradients computing with the unified surrogate loss function are used to update the weights of the network. We refer to the network trained with the approach as Clipped PPO.

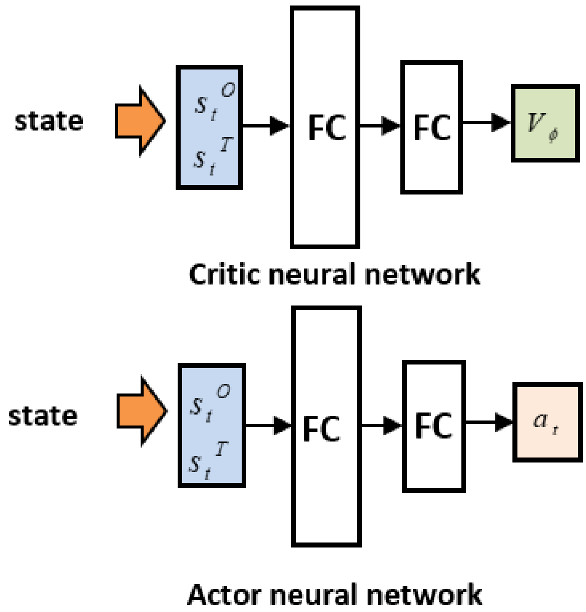


Figure 4.3: Network architecture. The observable state is fed into two fully-connected layers (FC), the outputs of critic network and policy network are a state value function (green) and an action (orange), respectively.

4.1.3 Training Process

Here, we focus on learning a path following and collision avoidance policy simultaneously which performs robustly and effectively in various scenarios. We also adopted clipped PPO for the collision avoidance problem.

4.2 Implementation and Simulation Result

4.2.1 Implementation for Collision Avoidance System

COLREGs compliance

An important factor in the implementation for the collision avoidance system is ensuring all the ships comply with the COLREGs rules. The detailed description of COLREGs are in the appendix. A diagram of target ships as defined by COLREGs Commandant (1999) is shown in Fig. 4.4: the autonomous OSV is requested to be the give-way vessel, and the target ships are designed to be the stand-on vessels. If the distance between the give-way vessel and the stand-on vessel is in a dangerous range, COLREGs is considered to apply. As suggested in Fig. 4.4(a), in the case of head-on scenario, the autonomous OSV should change course to starboard and pass with the target ship on its port side to avoid it, then return to the original path when confirming safety. Similarly, the diagram of the crossing scenario is shown in Fig. 4.4(b), the optimal strategy corresponds to a course offset toward starboard side until the target ships are passed at a safe distance on the autonomous OSV's port side. Finally, the overtaking scenario is shown in Fig. 4.4(c), the autonomous OSV can either pass starboard or port of the target ship, depending on COLREGs. In these scenarios, the target ships do not respect its responsibility to keep away.

Implementation

To visualize the simulation results, the 3D visualization tool RViz provided by Robot Operating System was used. It allows visualizing the simulated environment, including the autonomous OSV, generated path, and target ships. The calculation module interacts with RViz to set the position and orientation of the autonomous OSV.

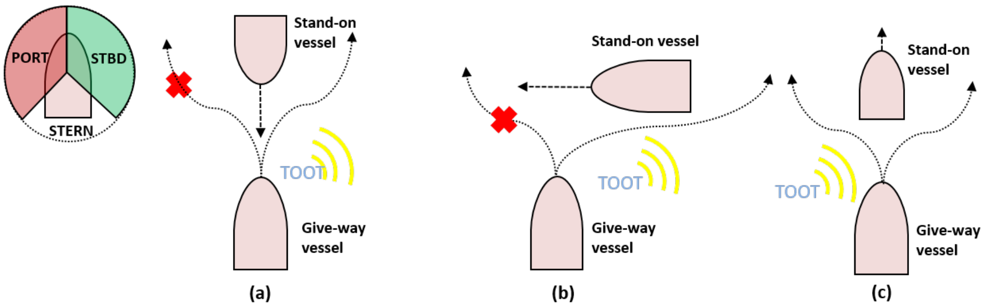


Figure 4.4: Target ships for COLREGs: (a) Head-on (b) Crossing (c) Overtaking.

In the real-world implementations, the environment is often dynamic and may change unpredictably. To handle this environment, the path planner is required to be able to generate various paths. We divided the training process into two phases, which accelerates the policy convergence and allows to get a higher reward.

In the first phase, we train the autonomous OSV to follow the randomly generated path without any target ships. It allows the autonomous OSV to improve the training speed in the presence of the target ships.

The objective of controlling the autonomous OSV is to follow the randomly generated path without deviation. In this task, the rudder angle of the autonomous OSV is limited to three choices: a positive angle of a fixed magnitude, a negative angle of the same magnitude, or zero. The reward functions described in previous section, except the collision reward function and COLREGs related reward function, are given on every time step until the destination (last waypoint) is reached. That means that the episode is finished. After completing the given training iteration, the optimal policy can be obtained.

When the autonomous OSV achieves reliable performance, we save the trained policy and proceed to the second phase. Based on the trained neural network, the policy is further updated in the second phase, when the autonomous OSV is assigned to follow the randomly generated path with the target ships. To simplify the problem, we assume that there are three target ships, which represent different scenarios at each segment path: head-on scenario, crossing scenario, and overtaking scenario. During following the path, the autonomous OSV encounters these three types of ships. Fig. 4.5 illustrates the training process setup of the second phase.

In each episode, the autonomous OSV follows the path and avoids the target ships. Rudder angle is applied by the autonomous OSV until the destination is reached. Then, the autonomous OSV is restored to its initial position, and the new episode begins.

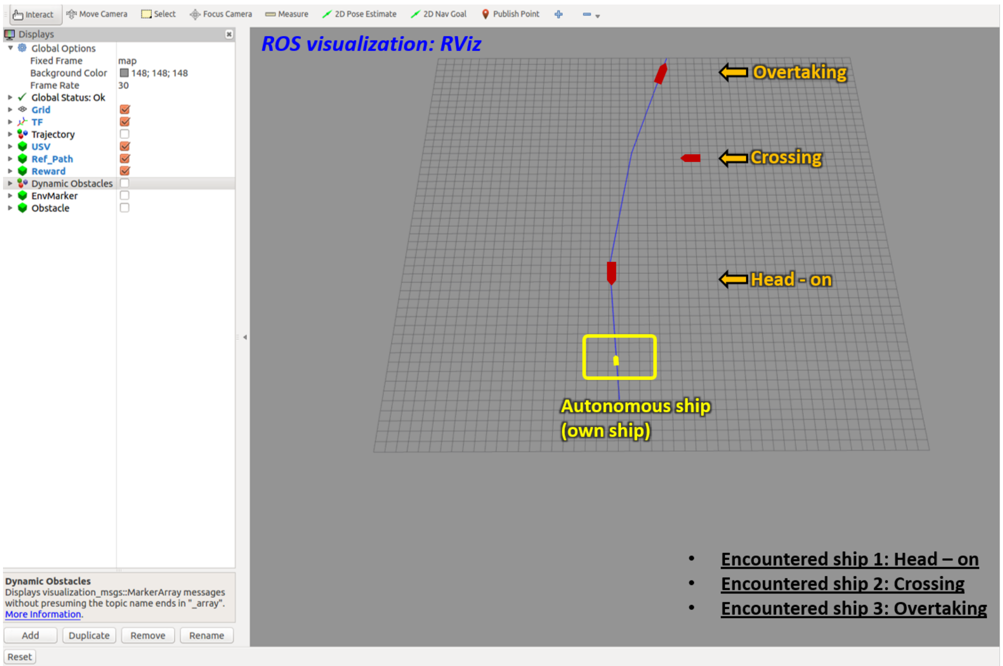


Figure 4.5: Simulation setup in phase two: collision avoidance with three target ships.

Before proposing the collision avoidance formulation, some assumptions should be made to simplify the training process. There are three target ships which represent the different scenarios respectively at each segment path: head-on scenario, crossing scenario, and overtaking scenario. The specifications of the target ships are listed in Table. 4.2. Each target ship is regarded as a circle with radius R, R is randomly selected between 25 and 150 meters. Also, it is assumed that if the autonomous OSV did not take avoidance actions, it will collision with target ships. As a result, the initial position of the target ships should be well-designed.

Since all the target ships are designed to be the stand-on vessels, the velocities of them are set to be constant. During the training process, the head-on ship is located on the first segment path with a random velocity. When the autonomous OSV successfully avoids the first head-on ship and passes the first turning waypoint, the crossing ship set sail with a constant velocity V_C. Similarly, the overtaking ship start moving with a relatively slow velocity V_O as soon as the autonomous OSV passes the second turning waypoint.

When evaluating the simulation effect, it is worth to take into account an additional visualization tool to help users check the results through a more immersive and realistic view. In this study, we used the Unity 3D which has many advantages such as low price, good performance, and various effects. Especially, the environment such as the sky with clouds, ocean with reflection and waves around the autonomous OSV was modeled for realistic visualization. The graphical user interface of the simulation result is shown in Fig. 4.6.

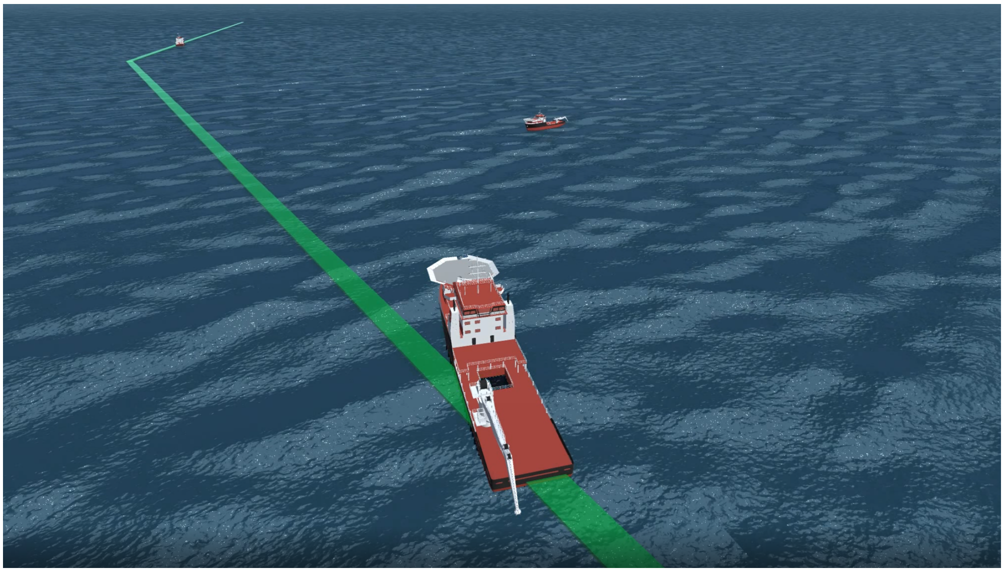


Figure 4.6: Visualization of the simulation using unity 3D.

Table 4.2: Simulation setup with target ships.

Target Ship Types		**Details**		
		Dimension (radius)	**Initial Position**	**Velocity**
Ship 1	Head-on	$R \in [25, 150]$	P_H	V_H
Ship 2	Crossing	$R \in [25, 150]$	P_C	V_C
Ship 3	Overtaking	$R \in [25, 150]$	P_O	V_O

Table 4.3: Environment conditions for collision avoidance scenario.

Case	Wave condition			Current direction [deg]	Wind direction [deg]	Training & Testing		
	Direction [deg]	Height [m]	Period [sec]			Network model	Training iteration	Training time [h]
2-1	90	1.0	10	90	90	B	1,580	27
2-2	45	1.0	10	45	45	B	-	-

4.2.2 Simulation Result

The performance analysis of the collision avoidance system was conducted by using the proposed control algorithm for various environmental conditions. The objective is to control the autonomous OSV to avoid the target ships with respect to COLREGs compliancy, while ensuring following the predefined path.

To demonstrate COLREGs compliance, we trained the RL agent to avoid target ships using the clipped PPO algorithm. According to the previous section, the state input s_t consists of the state of the autonomous OSV observed itself s_t^O and s_t^T relevant to the target ships. The output of the network is the rudder angle and the reward function consist of the collision avoidance reward functions, and the path following reward functions.

Cases in Table. 4.3 suggest that the RL agent is trained including the external forces. Case 2-1 shows the simulation is trained using the network model B under the following environmental conditions. It takes 27 hours approximately by 1,580 iterations. Case 2-2 is the other simulation case using the network model B when changing the environmental conditions, particularly, which is tested without additional training.

Fig. 4.7 illustrates a training process of collision avoidance with three target ships. Training starts with a head-on scenario, where the black arrow and the green arrow represent the initial heading angle of the autonomous OSV and head-on ship respectively. When in the head-on stage, the autonomous OSV (red) pass with the head-on vessel (green) on its port side, then back to the original path (blue). The first waypoint arrival of the autonomous OSV will trigger the crossing scenario. At this time, the crossing OSV (light blue arrow) starts approaching to the path, the autonomous OSV (red) makes a course change to avoid it. The course change is to starboard (in compliance with COLREGs) since a course change to port might increase the hazard. Stage 3 illustrates the overtaking scenario: while the autonomous OSV is overtaking a slower target ship (orange arrow), the autonomous OSV then changes course to starboard to keep away from the overtaking ship. Finally, one trajectory finishes until the autonomous OSV reaches its destination.

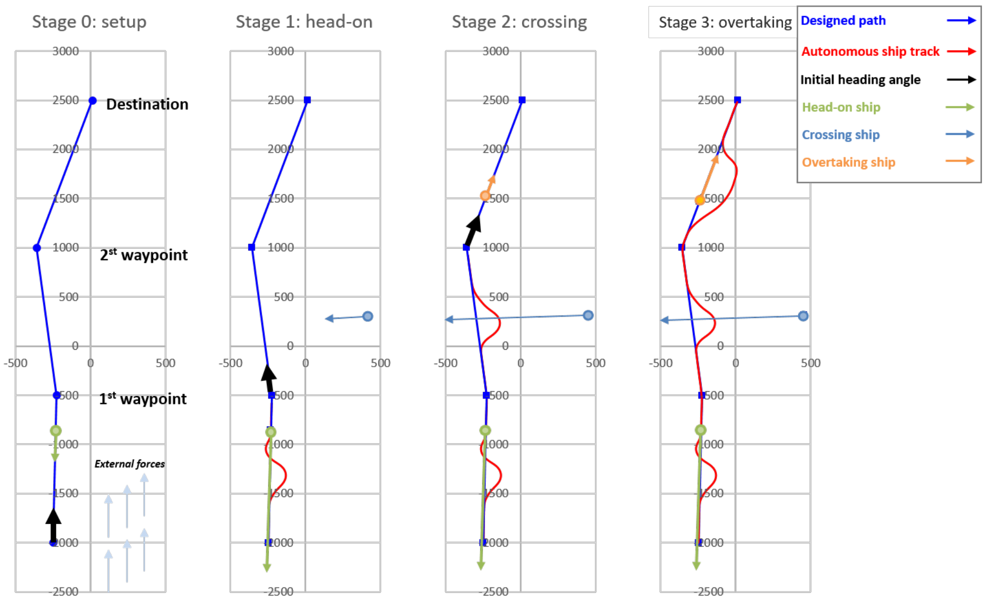


Figure 4.7: Training process of collision avoidance with three target ships.

During the head-on and crossing training process, it is not clear whether the autonomous OSV collision with the target ships from the above graphs as lack of the time coordinate. Therefore, we added Fig. 4.8 to illustrate the middle process when two ships meet at the same time. In the head-on scenario, when the head-on ship arrives at the P_{1_t}, the autonomous OSV has made a maneuver to avoid collision and crosses abaft of the head-on ship. In the second case, when the crossing ship arrives from the starboard side, and the autonomous OSV changes course to starboard and passes with crossing ship on her port side.

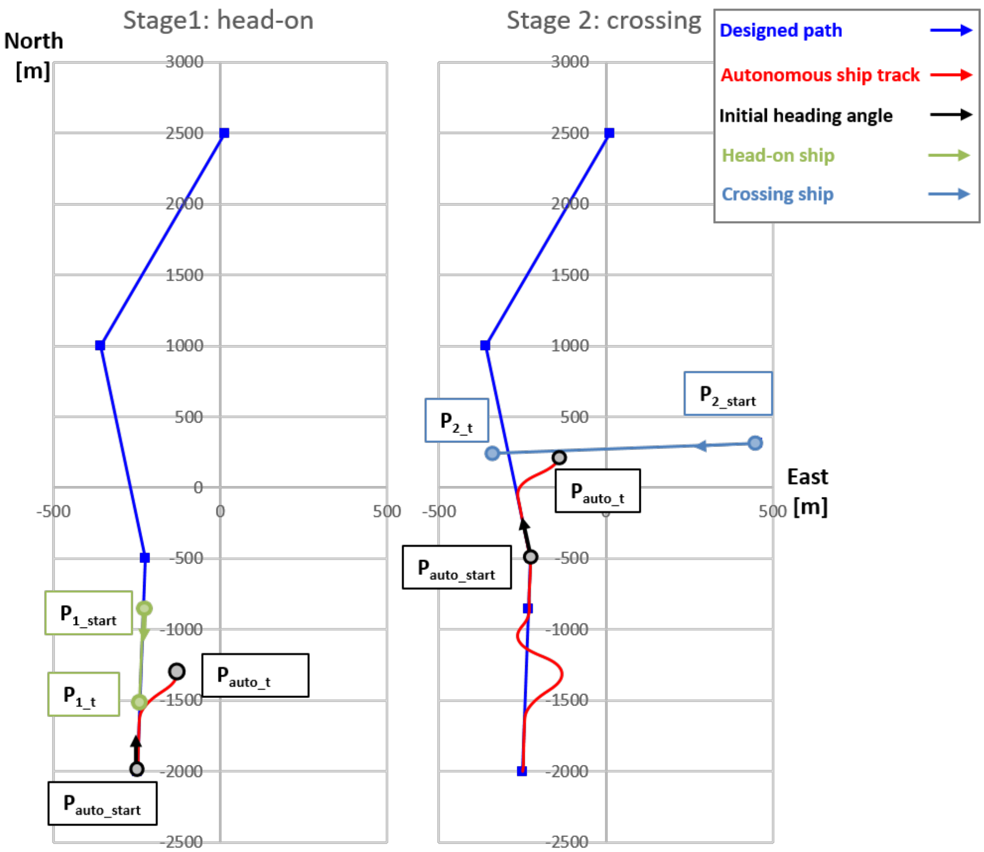


Figure 4.8: Head-on and crossing simulation.

Fig. 4.9 illustrates the simulation result of collision avoidance with three type of target ships using the network model B. In Fig. 4.10, a corresponding graph of rudder angle of the autonomous OSV is presented. The control behavior corresponds to a course offset toward the starboard side until all the target ships pass at a safe distance on the autonomous OSV's port side.

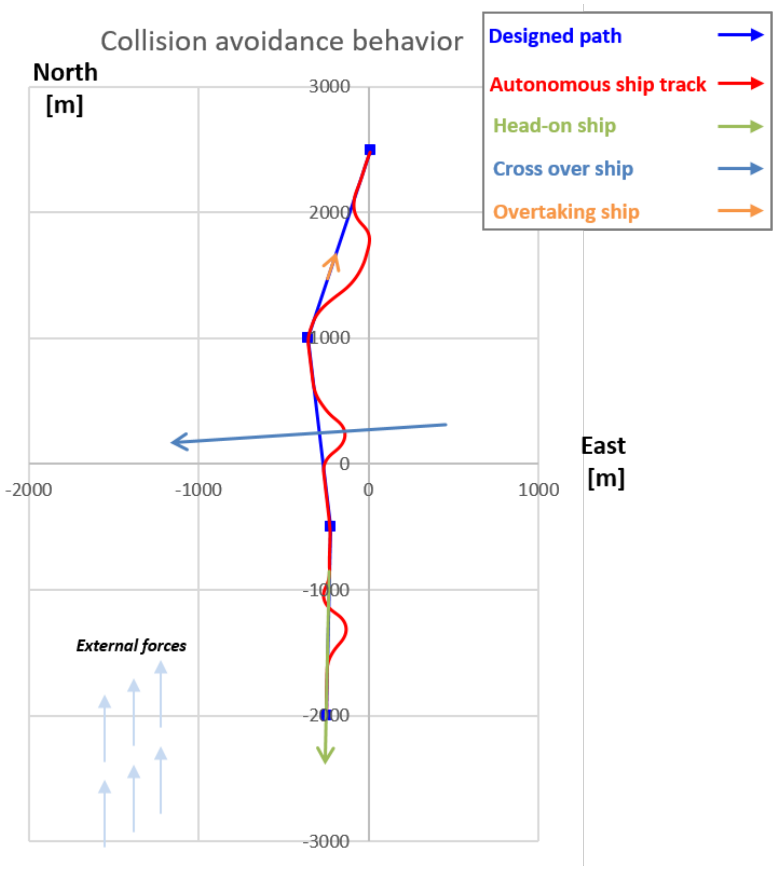


Figure 4.9: Graphs of positions of the autonomous OSV, and three target ships in case 2-1. The result is calculated by the model B with external forces.

Therefore, we can conclude that the performance of the proposed algorithm for controlling the autonomous OSV to avoid various target ships. Also, based

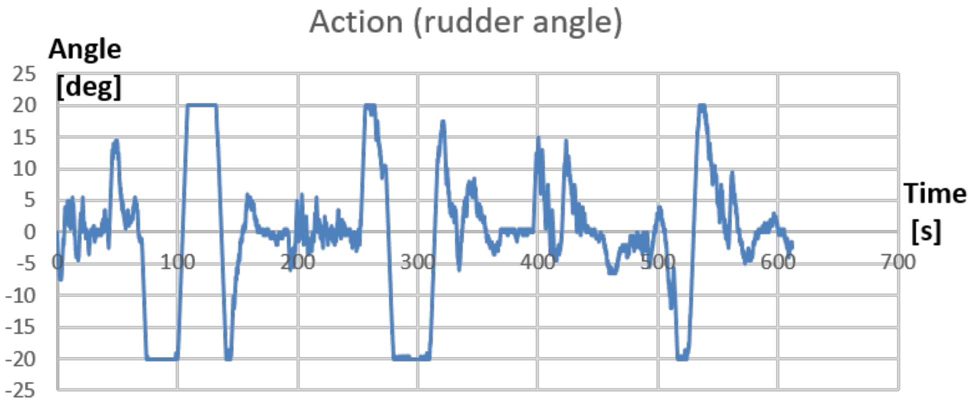


Figure 4.10: Graphs of rudder angle of the autonomous OSV in case 2-1.

on the network model B, we changed the target conditions. Simulation results in Fig. 4.11 and Fig. 4.12 illustrate that the proposed algorithm is practical and can safely manage complex scenarios with various environmental disturbances.

Fig. 4.13 shows the average reward for collision avoidance in case 2-1. The average reward converges to a maximum value until 1580 iterations, and it takes approximately 27 hours.

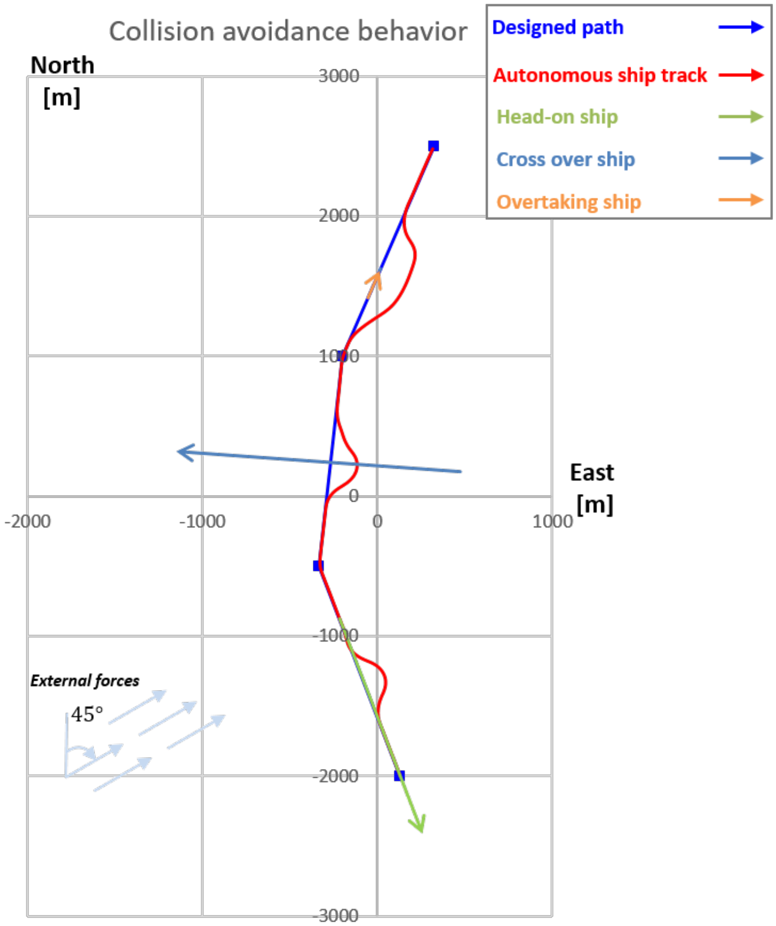


Figure 4.11: Graphs of positions of the autonomous OSV, and three target ships in case 2-2. The result is calculated by the model B with external forces.

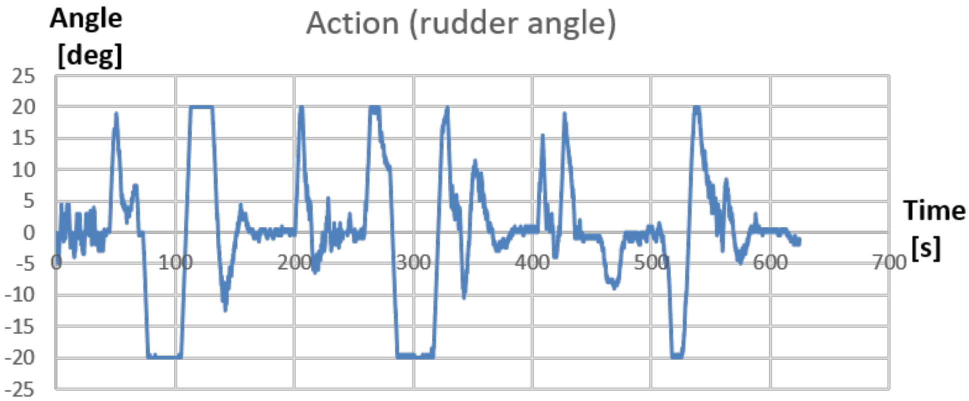


Figure 4.12: Graphs of rudder angle of the autonomous OSV in case 2-2.

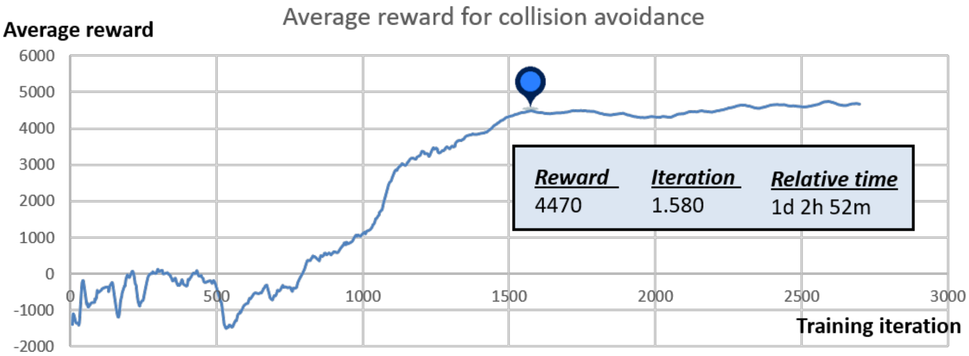


Figure 4.13: Average reward for collision avoidance.

4.3 Implementation and Simulation Result for Multi-ship Collision Avoidance Method

4.3.1 Limitations of Multi-ship Collision Avoidance Method - 1

In the previous section, the single-ship collision avoidance system has taken the first person view, where the own ship is the only manoeuvring party while the target ships remain their heading and velocity. Only the own ship is assumed to comply with COLREGs. According to the it, the own ship is requested to avoid all of the target ships. Moreover, it is assumed that there is only one target ship which the own ship has to make decision to avoid collision at some point.

However, whether it is possible to let the own ship avoid multiple target ships at the same time? Fig. 4.14 illustrates a case when the own ship encounter with multiple target ships. For example, when three target ships (ship 4, ship 5, ship 6) approaching to the own ship at the same time, the own ship has to avoid all of them. That is mean, the own ship has to make the optimal decision by considering the situation of the three target ships. While meeting the three target ships, the state space of the three target ships has to input to the neural network. The input is consist of not only the own ship related state, but the target related states (red). As a result, when the own ship encounters several target ships at the same time, the input size of neural network will be changed.

To solve this problem, we divide the input into two regions: the own ship related state region and the target ships related state region. Furthermore, the target ships related state region is divided into four sub-regions: head-on, crossing_port, crossing_starboard, and overtaking region. For example, the target ship 4, which is determined be the head on ship with respect the the own ship, ones state has to be input to the head-on region. Based on the section partition method, the input size of the neural network can be fixed.

One of the key points to the multi-ship collision avoidance method is to categorize the type of the target ships in terms of COLREGs.

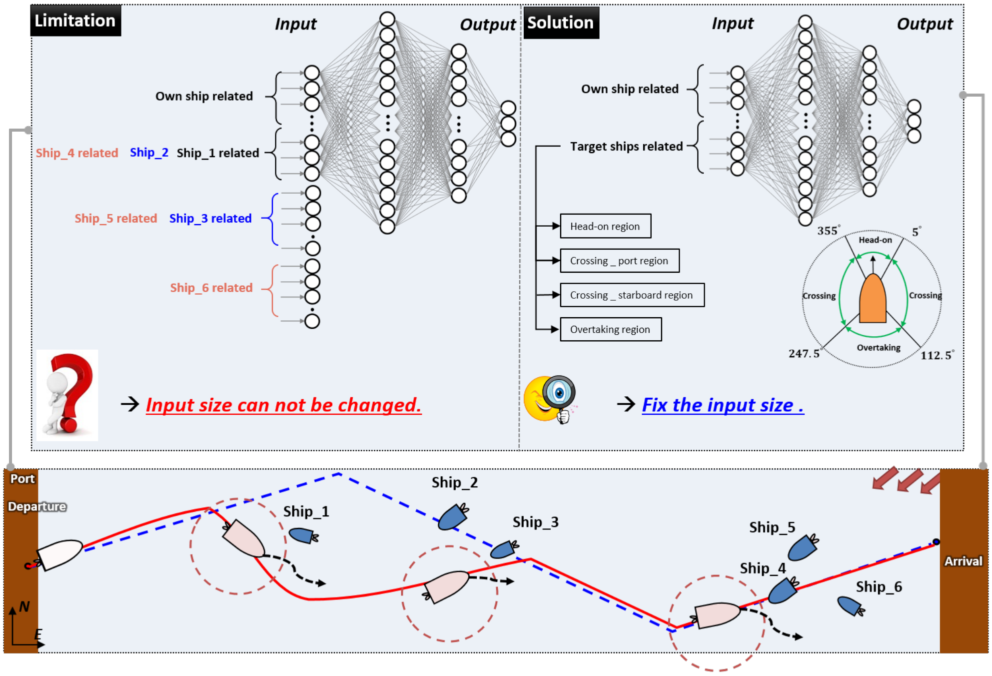


Figure 4.14: Limitation and solution of multi-ship collision avoidance method.

4.3.2 Limitations of Multi-ship Collision Avoidance Method - 2

The other limitation of the training process is that when all the ships state observation inputted to the neural network, its input size will be changed correspondingly. Here, Fig. 4.15 illustrates a case that the own ship encounters with multiple target ships.

Instead of inputting the state observation of all the ships into the neural network, we copy one network model, which is fully trained at the beginning of the simulation, to all the ships. Using this strategy, each ship not only own ship, but target ships can make decision to avoid each other.

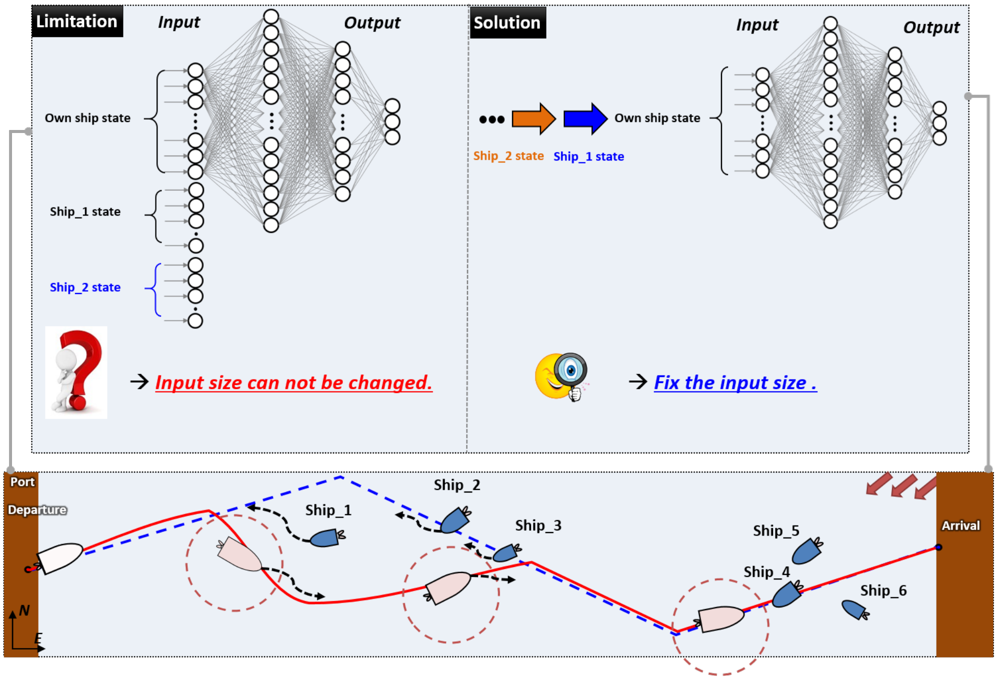


Figure 4.15: Limitation and solution of multi-ship collision avoidance method.

By overcoming the two limitations, we aim to let all the ships have the capability to make decisions from the first person perspective according to the state observation receives from the surrounding ships. The multi-ship collision avoidance method has the capability to manoeuvre not only so called the own ship, but the target ships can autonomously avoid each other according to COLREGs.

4.3.3 Implementation of Multi-ship Collision Avoidance Method

Target Ships Category

Each ship treats itself as the own ship (OS) and its surrounding ships as target ships (TSs). In order to categorize the type of the target ships with respect to the own ship, the input to the neural network is divided into four regions including head-on, crossing_port, crossing_starboard, and overtaking. As shown in Fig. 4.16, the diagram centers on the the own ship is divided into four regions with head-on (Ship 3), crossing_port (Ship 4), crossing_starboard (Ship 2), and overtaking (Ship 1). It can be used to determine whether the own ship has to avoid the target ships. Here, the instantaneous velocity of the own ship is defined in v_0, and the instantaneous velocities of the target ships are defined in v_{T_i}. i is the number of the target ships. Each target ship approaching to the own ship is categorized with respect to the heading of the own ship v_0.

Fig. 4.17 illustrates the overall process involved in determining the type of target ships based on COLREGs. The process involves computing the dimensions of the safety area for the own ship, which is considered the region R where target ships should not enter. It can be explained in three steps:

- The first step involves determining the whether the target ship enter the safety area or not,
- If that, the second step involves categorizing the type of the target ships. The target ships are categorized based on its instantaneous position with respect to the heading and position of the own ship according to the regions defined in Fig. 4.16: head-on, crossing_port, crossing_starboard, and overtaking,
- In the third step, the target ship is further categorized based on its relative

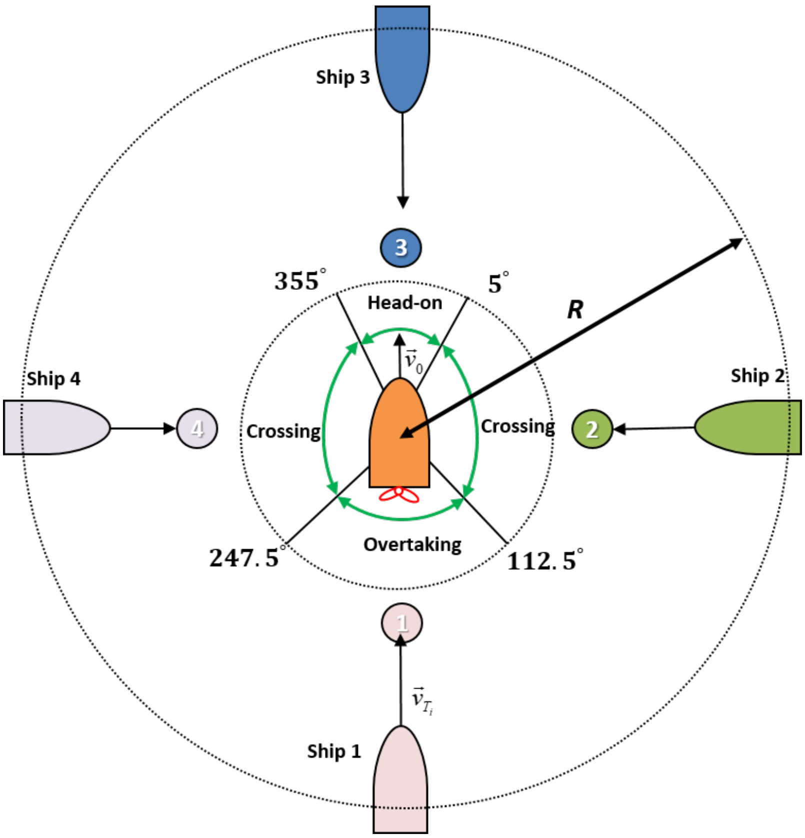


Figure 4.16: Regions used to categorize the position of the target ships.

heading with respect to the heading of the own ship.

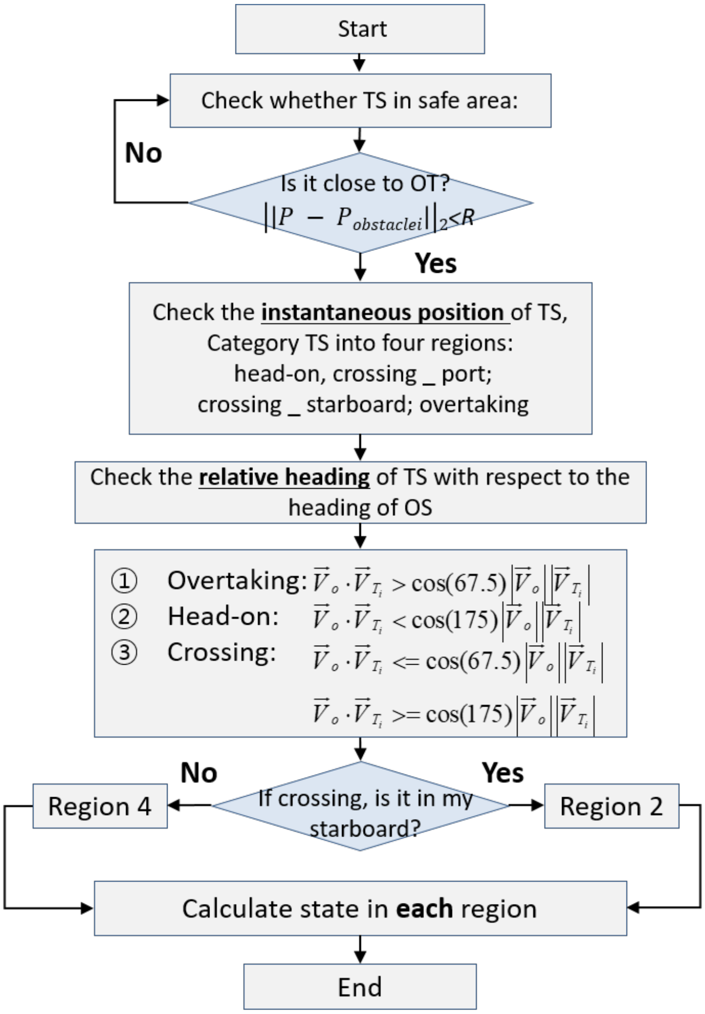


Figure 4.17: Procedure of capture of target ships state observation.

Reward Function of Multi-ship Collision Avoidance

The state category procedure for the multi-ship collision avoidance method is presented in the previous subsection, so that the next step is to design the training process, which means how to implement the reward functions for the multi-ship collision avoidance method.

One problem has to solved first is how to judge which reward function should be used. It can be determined by the relative position and velocity between the own ship and target ships. Suppose the current positions of the own ship is P and the target ships is set to $P_{obstacle}$, respectively.

The ship domain is treated as a circle with radius R and any intrusion of other target ships must be avoided. The ship domain of own ship R depends on the ship length and the maximum rudder angle. Fig. 4.18 illustrates a critical scenario: while the own ship is head on a target ship, whose principle dimension is same as the own ship. The own ship keeps on turning course to starboard with its maximum rudder angle in order to keep away from the target ship. The most minimum distance between the two ships is obtained. Correspondingly, the safe distance R is set to 400 meters in this study.

domain.png

Principal dimensions of the autonomous OSV	
Length O.A	48 [m]
Beam	8.6 [m]
Draft	2.2 [m]
Max. rudder angle	20 [deg]
Max. rudder speed	4.6 [deg/s]

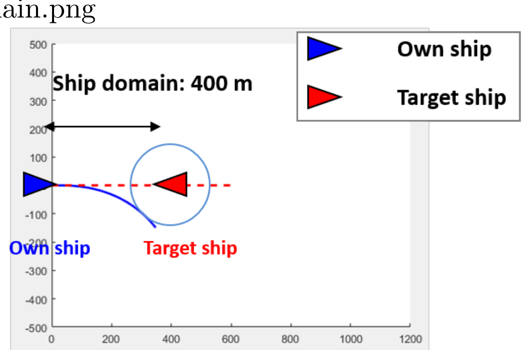


Figure 4.18: Ship domain of the own ship.

In real situation, there may be more than one target ship, and the own ship has to judge which one has to avoid first at one time. It can be judged by the ship domain R. If the distance between the own ship and the target ship is larger than the radius of the ship domain R, the path following reward function is adopted.

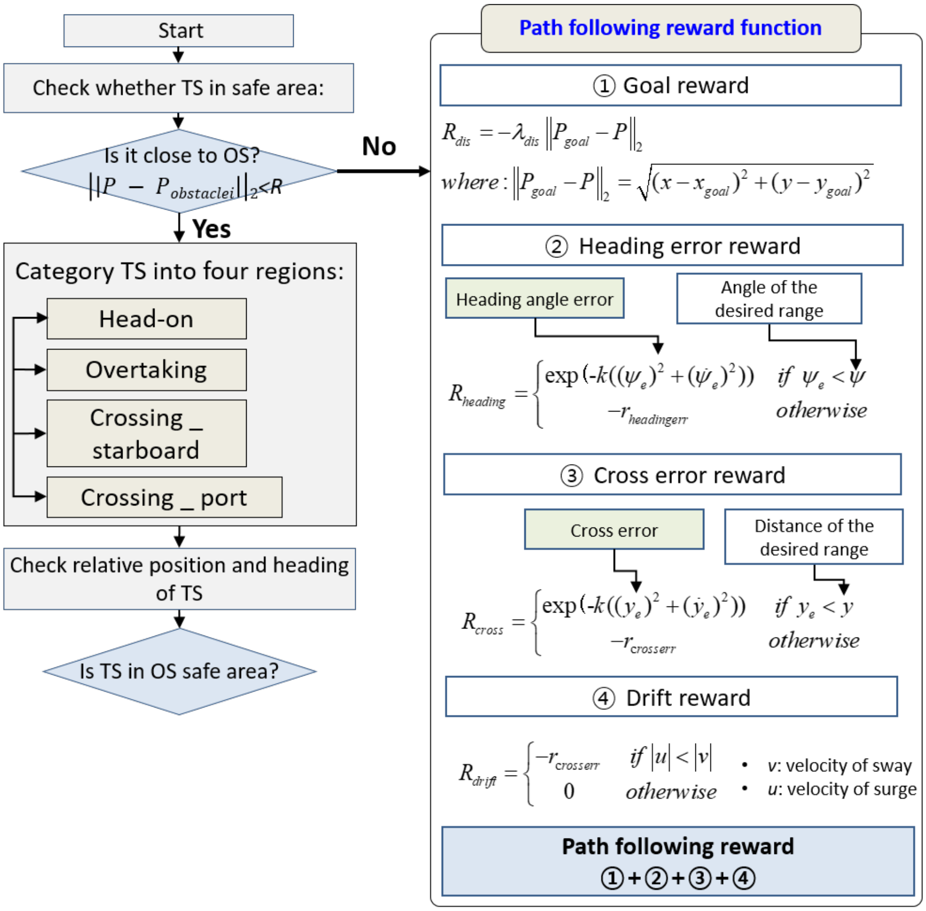


Figure 4.19: Reward function of collision avoidance.

If not, let's move to the next step: category the target ships into four regions and calculate the state observation respect to each target ship. Once the target ships in the following region: head-on, crossing_port, and overtaking region, it can trigger the collision avoidance reward function.

It should be noted that when the target ship is approaching from the port side of the own ship, the target ship is regarded as the crossing_port target ship. At this time, the crossing_port target ship should take appropriate operations to give way to the own ship, while the own ship makes no attempt to keep away, and keeps its way to the destination. In this situation, the path following reward function has to be continuous implemented.

As shown in Fig. 4.20, the collision avoidance reward function should be implemented until all the target ships are passed at a safe distance on the own ship's port side. It can be judged by checking the relative position and heading angle between the own ship and the target ships.

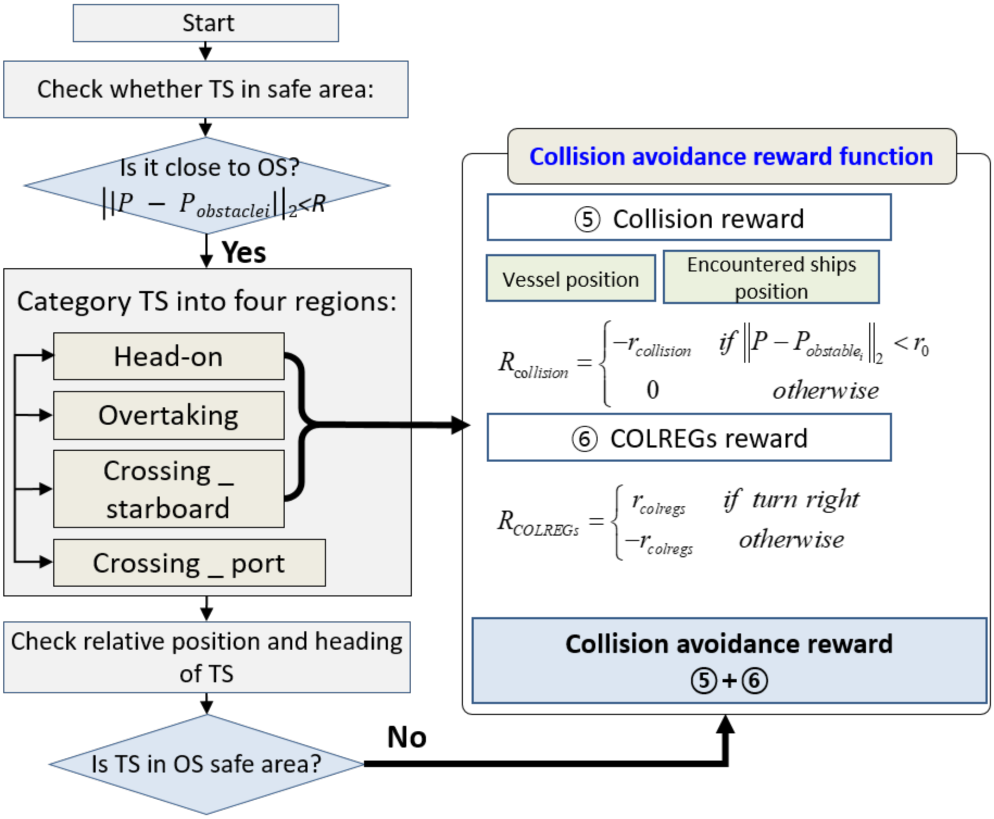


Figure 4.20: Reward function of collision avoidance.

If the own ship have been successfully avoid all the target ships, it is requested to converge to the predefined path. Consequently, the path following reward function is switched to implement.

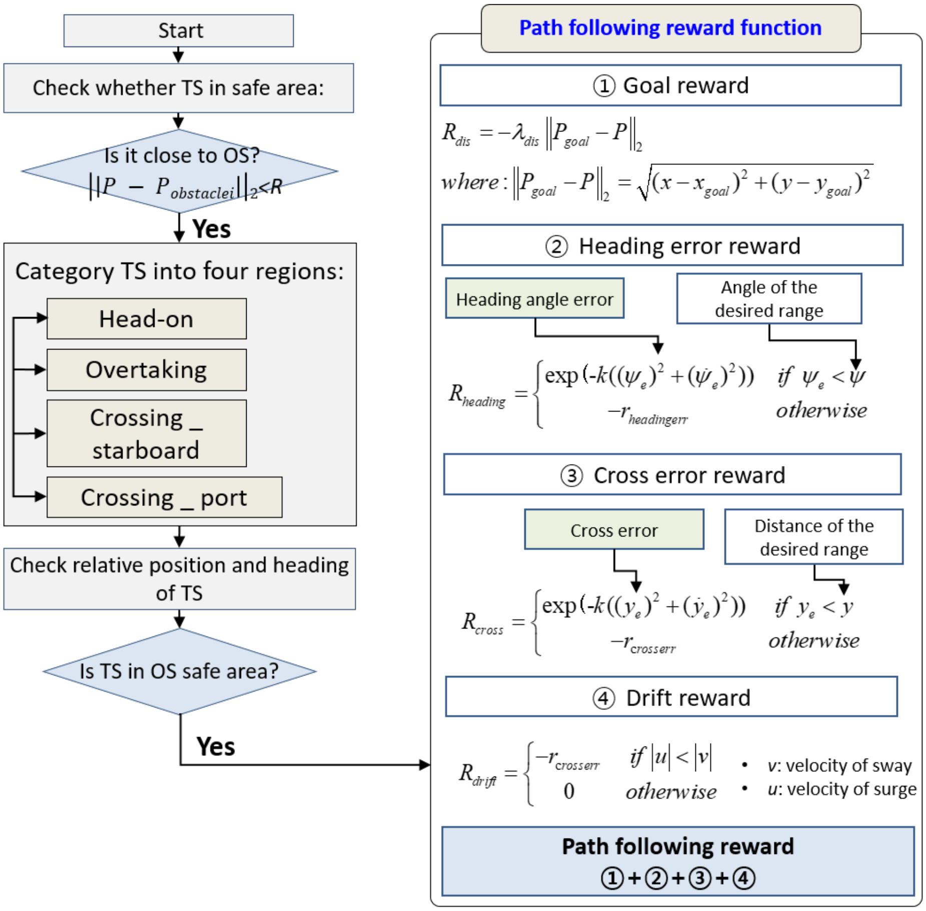


Figure 4.21: Reward function of collision avoidance.

4.3.4 Simulation Result of Multi-ship Collision Avoidance Method

Simulation Case 1

The main information used to evaluate COLREGs compliance at a given future point in time, on a predicted ship trajectory generated by the control behavior, is illustrated in Fig. 4.22, the detailed information can be found in the appendix.

According to COLREGs, two ships have four types of encounter situations, which are crossing_port, head-on, overtaking, and crossing_starboard. As we have explained previously, the purpose of this study is to make a decision concerning collision avoidance that is COLREGs compliant, with the reactive action of the surrounding target ships being taken into account. According to the target ships stipulated in COLREGs, the anticipated action of both the ships potentially involved in a collision is defined in Fig. 4.22, with the corresponding regions using different colors.

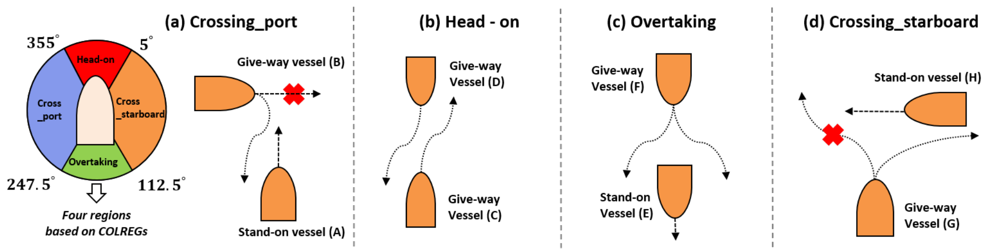


Figure 4.22: Simulation setup based on COLREGs rule.

Simulation setup with multiple target ships is given in Fig. 4.23. The specific procedure for this simulation case is presented as follows. During training process, all the ships (from ship A to ship H) use the same neural network. In addition, the initial position of the ships are fixed at the same point, and the range of relative heading of the ships are randomly specified according to

COLREGs.

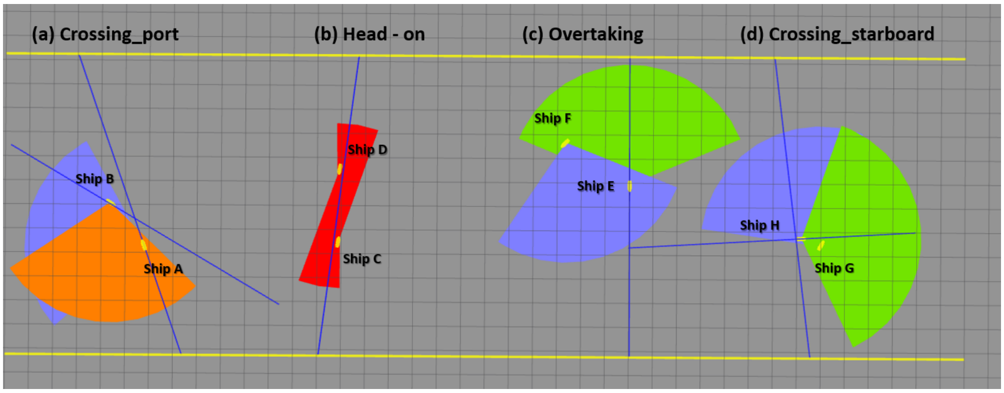


Figure 4.23: Diagram of target ships according to COLREGs.

- In case (a), a crossing_port situation happens if the other ship (ship B) coming from more than 22.5 degrees abaft the beam. When the two ships are involved in a dangerous ship domain, their surrounding region is painted with the corresponding color. For example, as ship B are in the crossing_port region of ship A, from the view of ship A, its color of the region becomes purple. On the contrary, ship A is regarded to be located in the crossing_starboard region of ship B, so that its color of the region becomes orange.

- In case (b), a head-on situation happens if two ships (ship C and ship D) are meeting on reciprocal courses in the range of 10 degrees. In this situation, ship C and ship D are found to be located in the other party's head-on region, the corresponding region are all painted with red color.

- In case (c), a overtaking situation happens if the other ship (ship F) is overtaking ship E in the range of 135 degrees.

- In case (d), a crossing_starboard situation situation happens if the other ship (ship H) coming from the starboard side of ship G.

Since all the ships manoeuvre at the same velocity, they will collision with each other if contrary to the COLREGs rules. The objective of this test is to evaluate the introduced method in interpreting COLREGs, which dictates that the ships should pass each other port to port.

The simulation result of case (a) and (b), as shown in Fig 4.24, which illustrates the procedure of the four ships collision avoid with each other and then converge to the predefined paths. When ship A and ship B move to the configuration shown in case (a), the two ships form a crossing_port situation, after the target ship's type is detected in terms of the procedure illustrated in Fig. 4.17. In case (a), when ship B arrive from the port side from the first person view of ship A, ship A has the right to stay on and ship B makes a course change to avoid it.

In case (b), when ship C and ship D are meeting in the head-on situation, both the ships have to alter course to starboard so that each pass on the port side of each other.

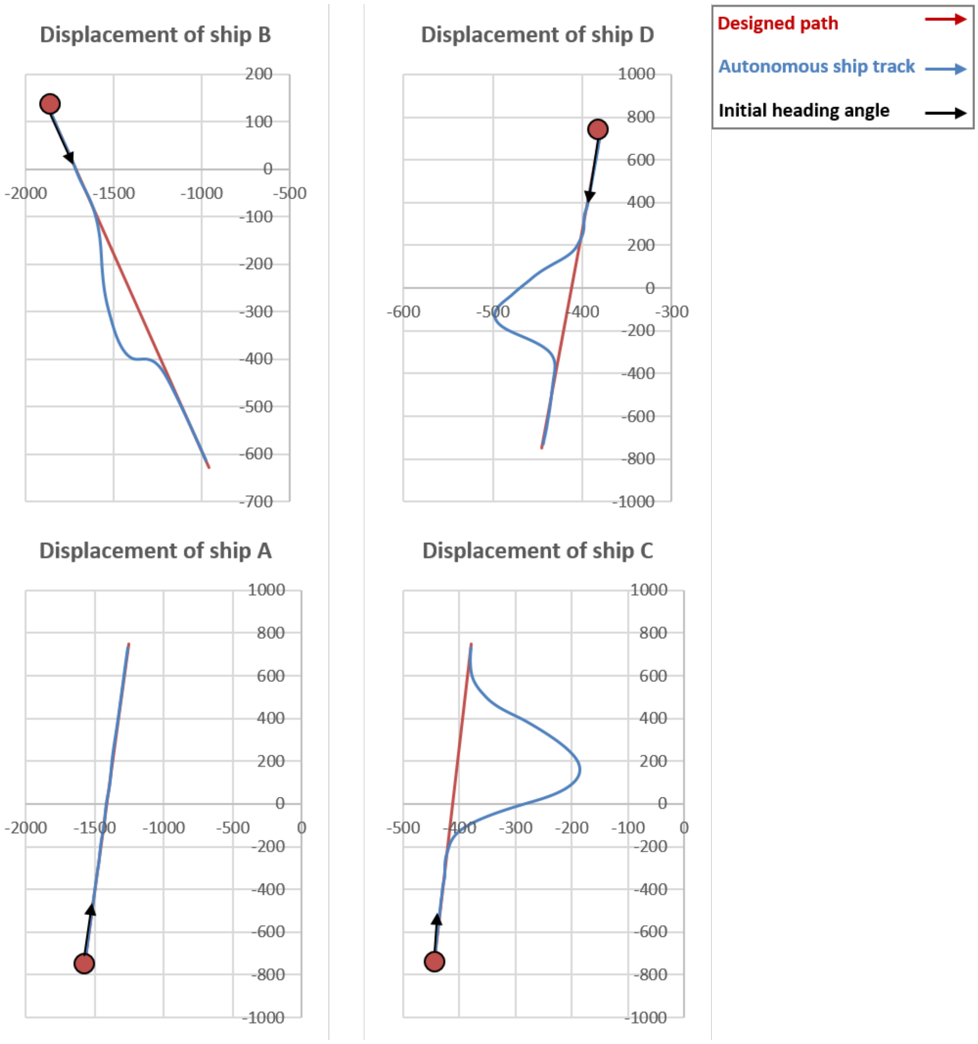


Figure 4.24: Simulation result of Multi-ship collision avoidance.

The simulation result of case (c) and (d), as shown in Fig 4.25. In case (c), ship E is overtaking ship F, and keeps out of the way with alternative control behaviors. In case (d), when ship H arrive from the starboard side from the first person view of ship G, ship H has the right to stay on and ship G makes a course change to avoid it.

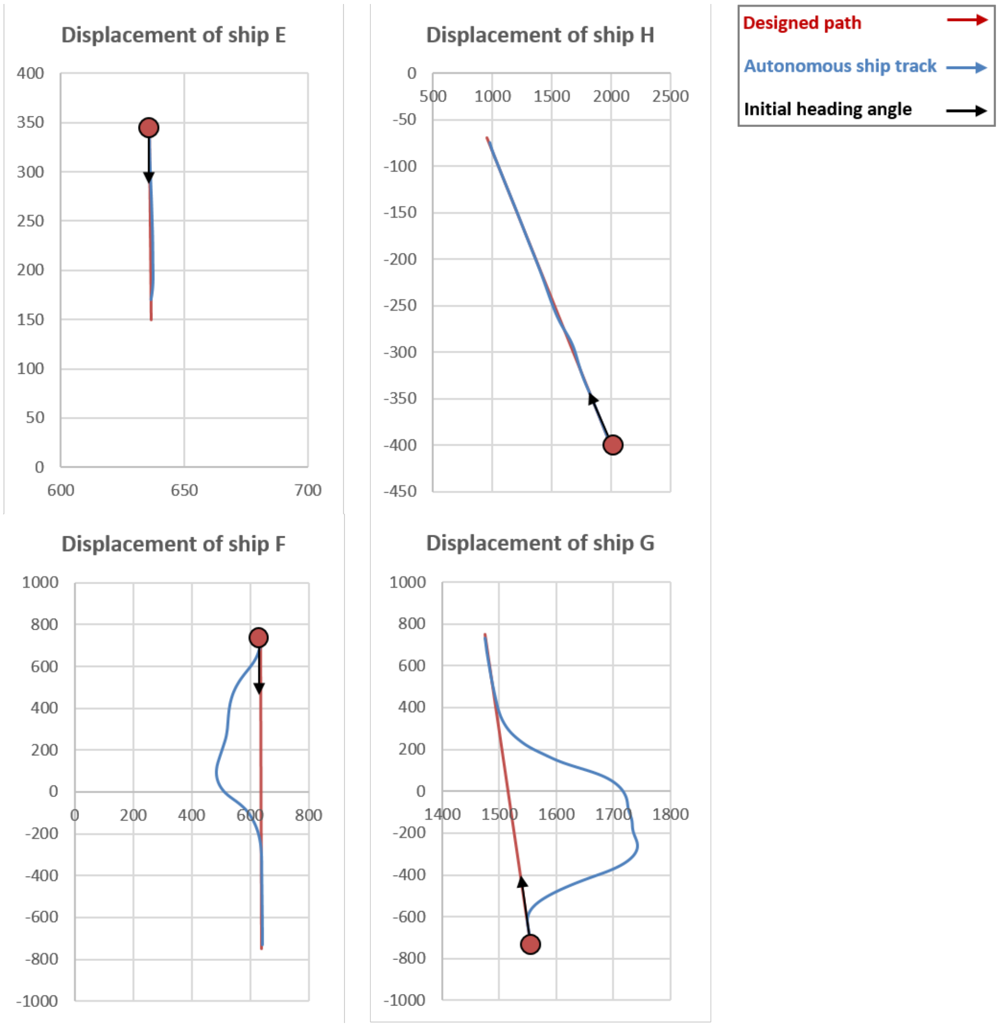


Figure 4.25: Simulation result of Multi-ship collision avoidance.

In the following subsections, a more complex scenarios is simulated and the results are analysed to test the performance of the proposed anti-collision decision making procedure.

Simulation Case 2

Simulation with four ships (from ship A to ship D) scenario is given in Fig. 4.26. Since the four ships manoeuvre with the same velocity, they will collision with each other if contrary to COLREGs.

From ship A's point of view, there are three target ships approaching: ship D (crossing_port), ship C (head-on), ship B (crossing_starboard), which means that ship A needs to manoeuvre to ship B (crossing_starboard)'s port side to avoid collision with respect to COLREGs.

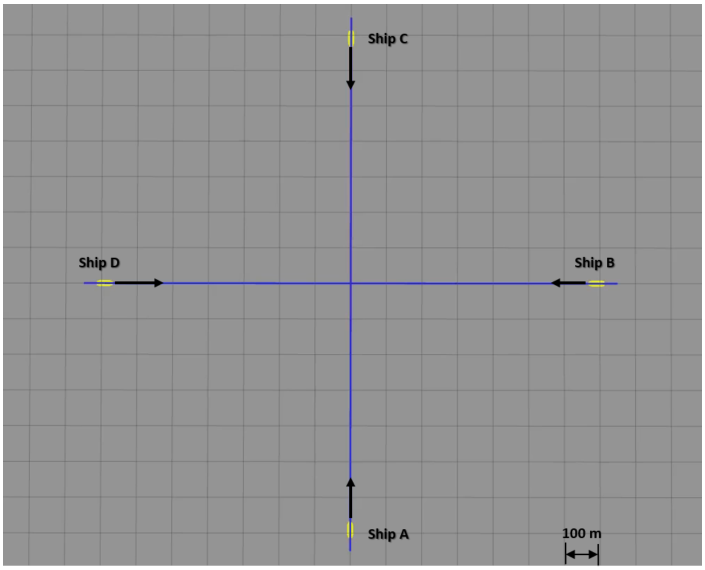


Figure 4.26: Multi-ship collision avoidance setup, where the four ships are heading to the center point.

With respect to ship A and ship B, it forms a crossing_starboard situation; for ship A and ship C, it forms a head-on situation. Ship A can ignore ship D as it is located in its crossing_port region. Therefore, ship A is responsible to change course to starboard in order to keep away from ship B and ship C.

Similarly, for the first person view of ship C, it has to make a course change to starboard in order to keep away and pass with ship D and ship A. The four ships will go round and round as long as the path does not deviate too much from the predefined path.

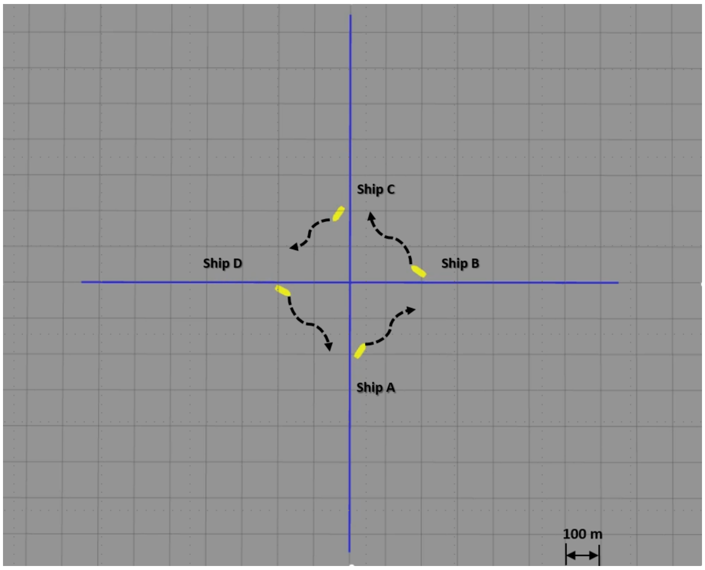


Figure 4.27: Procedure of the multi-ship collision avoidance, where the four ships are starting avoid their target ship.

Finally, the four ships converge to its predefined path until passing the other target ships.

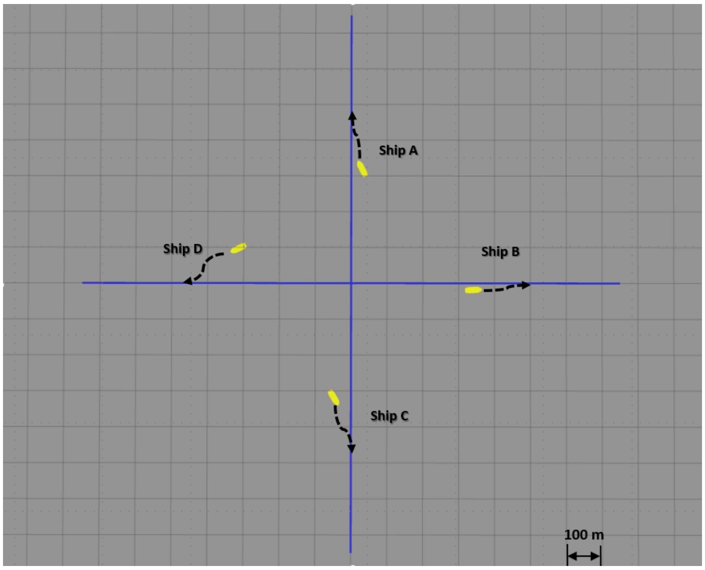


Figure 4.28: Procedure of the multi-ship collision avoidance, where the four ships are backing to their predefined paths.

Fig. 4.29 and Fig. 4.30 show the simulation result of the proposed multi-ship collision avoidance method. As explained previous part, ship A gives way to ship C while ship C gives way to ship A.

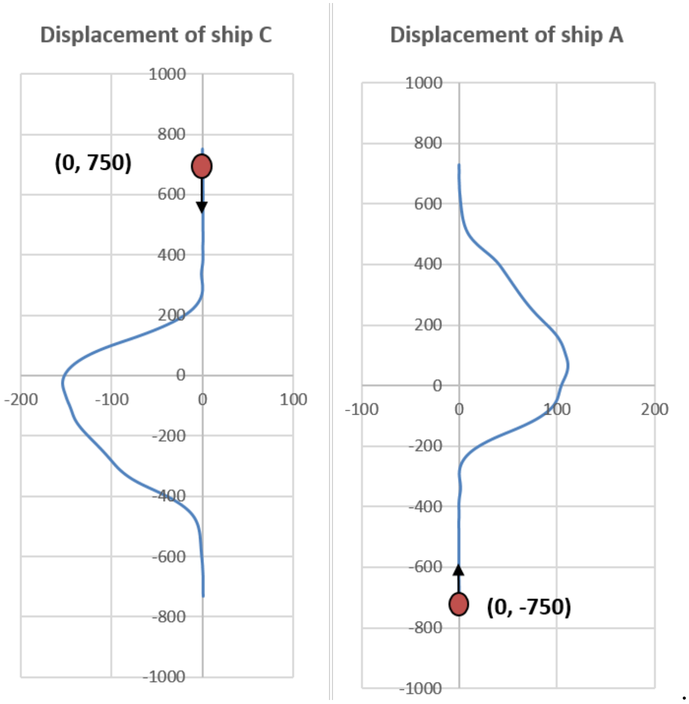


Figure 4.29: Displacement of ship A and ship C.

ship C and ship D have responsibility to stay away from each other in compliance with COLREGs. We believe that the proposed multi-ship collision avoidance method can deal with more complex situations. The method can be refined further by considering by reducing the unnecessary action command and adding the ship velocity as the action command.

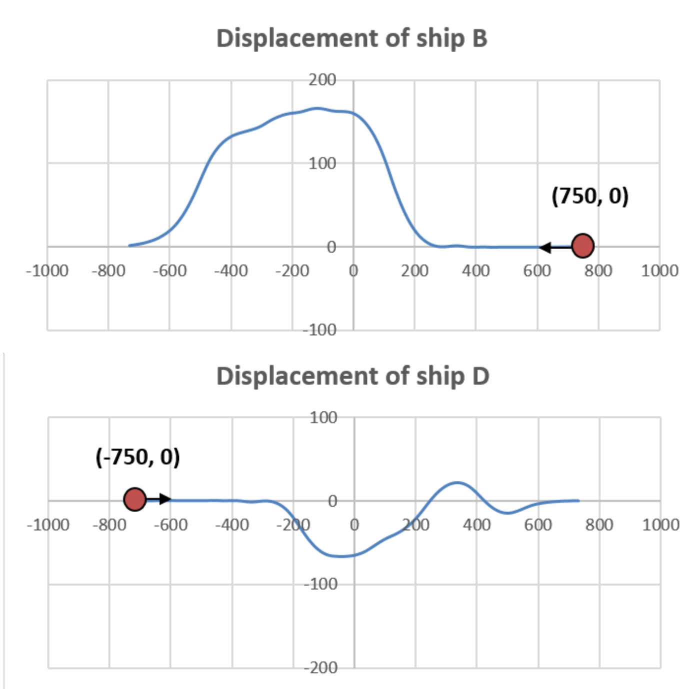


Figure 4.30: Displacement of ship B and ship D.

Chapter 5

Anti-Motion Control Method for Knuckle Boom Crane

5.1 Configuration of HILS for Anti-Heave Control System

Since the AHC system is responsible for the successful operation of the crane, the AHC system must have a suitable control algorithm, and its performance should be evaluated in advance. Performance analysis of the AHC system requires a complicated test procedure, and a great deal of equipment.

This real model requires real sensors, actuators, and mechanical systems to generate true sensor measurements, and receive control signals. This test environment is very costly, and it is nearly impossible to guarantee completely safe conditions for testing. Moreover, it is time consuming to test while at sea, and particularly difficult to identify bugs before testing. Therefore, to overcome these defects, the idea of replacing real sensors, actuators, and mechanical system by virtual ones is introduced, as shown in Fig. 5.1.

This testing environment is called Hardware-In-the-Loop Simulation (HILS), and represents a technique that is used in the development and testing of a complex control system interacting with a virtual model. Zhao et al. (2018). This virtual model is included in the testing and development by adding mathe-

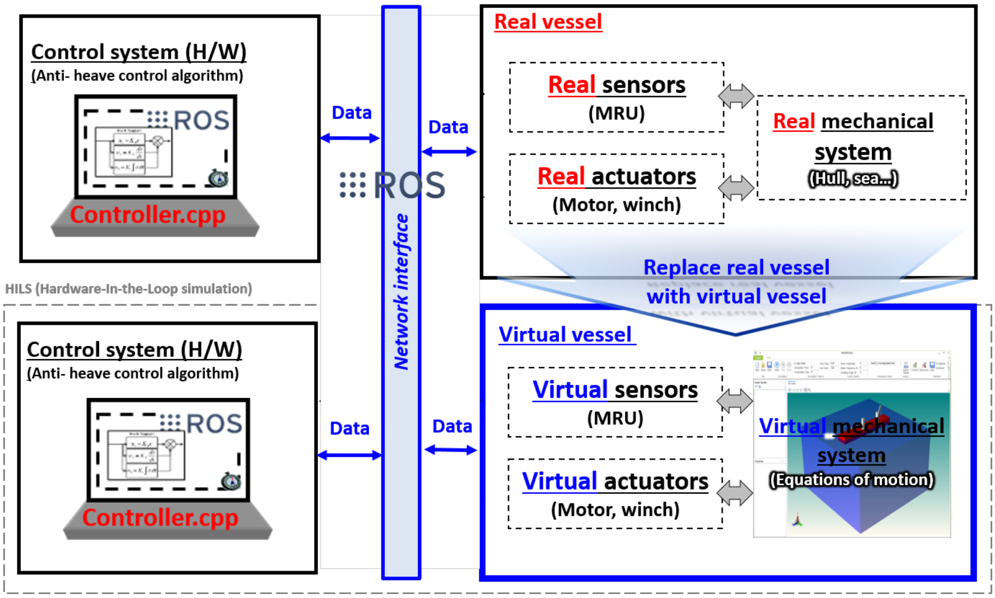


Figure 5.1: HILS environment for AHC system.

matical representation of all related dynamic systems. The test in the HILS environment is less expensive than the real test, because all parts, with the exception of the controller, can be replaced by software. It can also generate any environmental conditions, takes less time for testing, and allows bugs to be found in advance.

Fig. 5.2 shows the procedure to control the length of the wire rope with the sensors and actuator. the virtual model of the OSV can consist of a multibody system, which can represent realistic motion in waves. The HILS environment did not include the virtual sensor, which is assumed to be ideal. The virtual measures the current position of the suspended subsea equipment calculated by the virtual mechanical system. The difference between the desired position and current position of the equipment is sending to the AHC system. The AHC system, having a control algorithm, calculate the control signal - the desired angular velocity to actuate a virtual actuator. Consequently, the motor actuation will change the wire length to keep the equipment at a desired position.

The virtual sensor is implemented by simply receiving the motion data calculated from the virtual mechanical system. The virtual actuator calculates the rotating angle of the winch by integrating the rotating speed from the controller. Finally, the winch changes the length of the wire rope according to the rotating angle.

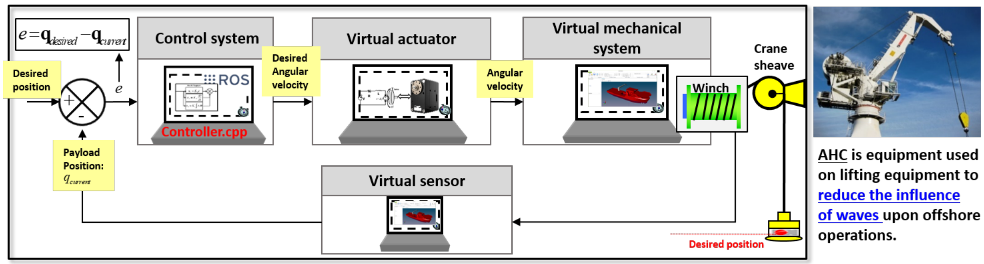


Figure 5.2: Simplified geometry of the system of the OSV-knuckle boom crane.

To develop the HILS environment for the proposed AHC system, a virtual model of the OSV was first created from a multibody system that can represent realistic motion in waves. Then, a controller of the AHC system with a control algorithm for heave compensation was implemented on real hardware. Next, an integrated simulation interface was implemented to efficiently connect the virtual model and the controller, and a visualization model was developed to verify simulation results by immersive and realistic views. Finally, a performance analysis of the AHC system was conducted within the proposed HILS environment.

5.1.1 Virtual Mechanical System

To represent the virtual mechanical system, the equations of motion of the OSV, the OSV crane, and the subsea equipment, including wave loads as an external force, must be formulated. They should be solved simultaneously, because these bodies are closely related to each other by joints or wire ropes, as shown in Fig. 5.3. Hence, this is referred to as a multibody system Cha et al. (2010).

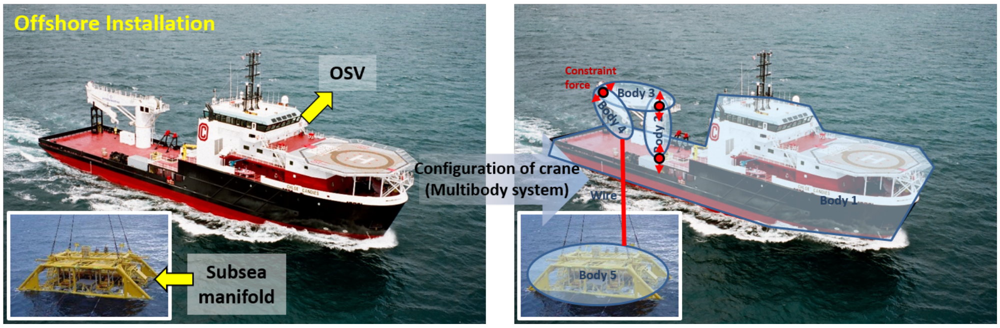


Figure 5.3: Multibody system of OSV, OSV crane, and subsea equipment.

It is very difficult to apply a Newton-Euler equation directly to the multibody system in the case of existing constraints and constraint forces. Therefore, we change the form of the Newton-Euler equation to the formulation for multibody system dynamics. Here, we choose the discrete Euler-Lagrange equation Ham et al. (2015). A variational principle is introduced in Fowles and Cassiday (1999). To compute a discrete trajectory of a body, the concept of virtual displacement and virtual work can be considered. According to Hamiltons's principle, which addresses the expenditure of energy in the system during motion, the action integral J can be defined as:

$$J = \int_{t_1}^{t_2} L \mathrm{d}t = \int_{t_1}^{t_2} (T - V) \mathrm{d}t \tag{5.1}$$

where, T and V are the kinetic and potential energy of the particle, respectively, and L is called the Lagrangian. During a time interval from t_1 to t_2, the actual motion minimizes the above integral. This can be expressed mathematically as:

$$\delta J = \delta \int_{t_1}^{t_2} (T - V)\mathrm{d}t = \delta \int_{t_1}^{t_2} L\mathrm{d}t \tag{5.2}$$

From Eq. 5.2, we can deduce the Euler-Lagrange equation, which is known to be:

$$\frac{d}{\mathrm{d}t}(\frac{\partial L}{\partial \dot{q}}) - \frac{\partial L}{\partial q} = 0 \tag{5.3}$$

A discrete Euler-Lagrange equation is utilized in Lew (2003), and Marsden and West (2001). The action integral of the Lagrangian of Eq. 5.1 can be represented by the sum of an infinitesimal area with time divided into small time steps h. Approximating each infinitesimal area as a rectangular shape, and the velocity $\dot{q}_k$ as $\frac{q_{k+1}-q_k}{h}$, the discrete action integral of the Lagrangian J_d can be expressed by:

$$J_d = \sum_{k=0}^{N-1} L_d(\mathbf{q}_k, \mathbf{q}_{k+1}, h) \tag{5.4}$$

where, h is the time step, and q_k is the position of the particle at time $t_1 + kh$. According to the variational principle, the particle moves along the trajectory where ∂J is zero. Thus, the formula can be expressed as:

$$\delta J_d = \sum_{k=0}^{N-1} L_d(\mathbf{q}_k, \mathbf{q}_{k+1}, h) = \sum_{k=0}^{N-1} [D_2 L_d(\mathbf{q}_{k-1}, \mathbf{q}_k, h) + D_1 L_d(\mathbf{q}_k, \mathbf{q}_{k+1}, h)]\delta \mathbf{q}_k = 0 \tag{5.5}$$

where, D_i is the partial differential operator, which means the partial differentiation by the i^{th} variable. As a result, Eq. 5.5 can be expressed in the form of a discrete Euler-Lagrange equation:

$$[D_2 L_d(\mathbf{q}_{k-1}, \mathbf{q}_k, h) + D_1 L_d(\mathbf{q}_k, \mathbf{q}_{k+1}, h)] = 0 \tag{5.6}$$

Considering the system with constraints $L = L + \sum_{j=1}^{m} \lambda_j g_j(q)$ and external forces, the discrete Euler-Lagrange equations for the system can be obtained, namely:

$$\begin{aligned}[D_2 L_d(\mathbf{q}_{k-1}, \mathbf{q}_k, h) D_1 L_d(\mathbf{q}_k, \mathbf{q}_{k+1}, h)]& \\ &+ \sum_{j=1}^{m} h\lambda_{jk}(\frac{\partial \mathbf{q}_k}{\partial g_j}) \\ &+ f_d^{\alpha+}(\mathbf{q}_{k-1}\mathbf{q}_k) \\ &+ f_d^{\alpha-}(\mathbf{q}_k\mathbf{q}_{k+1})(\mathbf{q}_{k+1} \\ &- \mathbf{q}_k) = 0 \qquad (5.7) \\ & \qquad (5.8)\end{aligned}$$

According to a Taylor series expansion, the constraints can be expressed by:

$$\sum_{j=1}^{m} g_j(\mathbf{q}_{k+1}) = \sum_{j=1}^{m} [(g_j \mathbf{q}_k) + (\frac{\partial g_j}{\partial \mathbf{q}_k})(\mathbf{q}_{k+1} - \mathbf{q}_k)] \tag{5.9}$$

From Eq. 5.7 and the constraints (Eq. 5.9), we can derive the discrete Euler-Lagrange equations in matrix form:

$$\begin{bmatrix} \mathbf{M} & -\mathbf{G}_k^T \\ \mathbf{G}_k & 0 \end{bmatrix} \begin{bmatrix} \mathbf{q}_k + 1 \\ h^2\lambda_k \end{bmatrix} = \begin{bmatrix} \mathbf{M}(2\mathbf{q}_k - \mathbf{q}_k - 1) - h^2 \frac{\partial V}{\partial \mathbf{q}_k} + h^2 \mathbf{f}(q_k, \frac{q_k - q_{k-1}}{h}) \\ -\mathbf{g}(\mathbf{q}_k) + \mathbf{G}_k \mathbf{q}_k \end{bmatrix} \tag{5.10}$$

Now, the motion of the OSV, including the OSV crane and the subsea equipment, can be calculated according to Eq. 5.10, which considering wave loads as an external force.

When the simulation starts, the equations of motion of the OSV can be solved to find its acceleration. Then the velocity and position are successively calculated by time integration (Ham et al., 2015b; Hong et al., 2015). Based on the obtained velocity and position, the external forces, including the radiation, diffraction, Froude-Krylov, and restoring forces, are updated for the next time step. The hydrodynamic force can be divided into two parts: the wave exciting force, which is exerted by the incident wave; and the radiation force from the wave, generated by the motion of the floater itself. This is expressed as:

$$\mathbf{F}_{\text{hydrodynamic}} = \mathbf{F}_{\text{exciting}} + \mathbf{F}_{\text{ratiation}} \tag{5.11}$$

$\mathbf{F}_{\text{exciting}}$ is calculated by multiplying the force Response Amplitude Operator (RAO) with the sinusoidal function at a given frequency. The force RAO can be obtained from a commercial solver, such as Wave Analysis by Diffraction and Morison (WADAM) by Veritas (2002). The Cummins equation Cummins (1962) can be used to calculate Fradiation, which considers the impulse response of the floater in the time domain. The frequency-dependent added mass coefficient $a_{ij}(\omega)$ and the frequency-dependent damping coefficient $b_{ij}(\omega)$ at a given frequency ω can also be obtained from the commercial solver. Using the frequency-dependent coefficients $a_{ij}(\omega)$ and $b_{ij}(\omega)$, the added mass A and retardation function $\mathbf{B}(\tau)$ can be determined. The infinite added mass $\mathbf{A}_{\infty}$, which is a constant matrix, is often used, rather than calculating the above integral. In the case of a regular wave, only one wave frequency should be chosen according to a single wave amplitude (a). The motions RAOs of OSV can be obtained from a commercial solver like WADAM.

However, in the case of an irregular wave, there are a number of N wave frequencies and amplitudes, according to the given wave spectrum.

$$\mathbf{F}_{exciting} = \sum_{m=1}^{N} a_m \mathbf{F}(\omega_m) \cos(\omega_m t + \epsilon_m) \tag{5.12}$$

Based on the exciting force theory proposed by Journée and Massie (2001), Eq. 5.12 is the linear superposition, which can be calculated in irregular wave. Since the first order wave force is a linear phenomenon, time history of the first order wave loads in a certain sea state can be obtained from frequency domain calculations by using the frequency characteristics of the first order wave loads, and the wave spectrum, by using the superposition principle. So that the time history of the first order wave loads becomes Eq. 5.12 with chosen phase shifts.

The principal dimensions of the target OSV (M/V CHLOE CANDIES) are 85.25 m in length, 18 m in breadth, and 7.4 m in depth, while its deadweight is 3,500 tons. The principal dimensions of the subsea equipment are 20.3 m in length, 15 m in breadth, and 6.1 m in depth. Its weight is 175 tons. Fig. 5.5 shows that the OSV and the crane are composed of several rigid bodies.

The pedestal of the crane (Body2) is attached to the deck of the OSV barge (Body1). The knuckle boom has upper and lower arms, which are depicted as Body3 and Body4. Those rigid bodies are connected by revolute joints, which can allow only one rotational motion perpendicular to the joint axis. The subsea equipment (Body5) is suspended by a single wire rope from the tip of the upper arm. As an initial condition, the wire length is 197.1 m below the sea surface. Because those bodies are connected by joints and the wire rope, the motions are closely related to the others. The OSV barge is the only part upon which the hydrodynamic force is exerted.

In this study, the wire rope is modeled by the incompressible spring, which adds the force only when it is extended. It is calculated as follows:

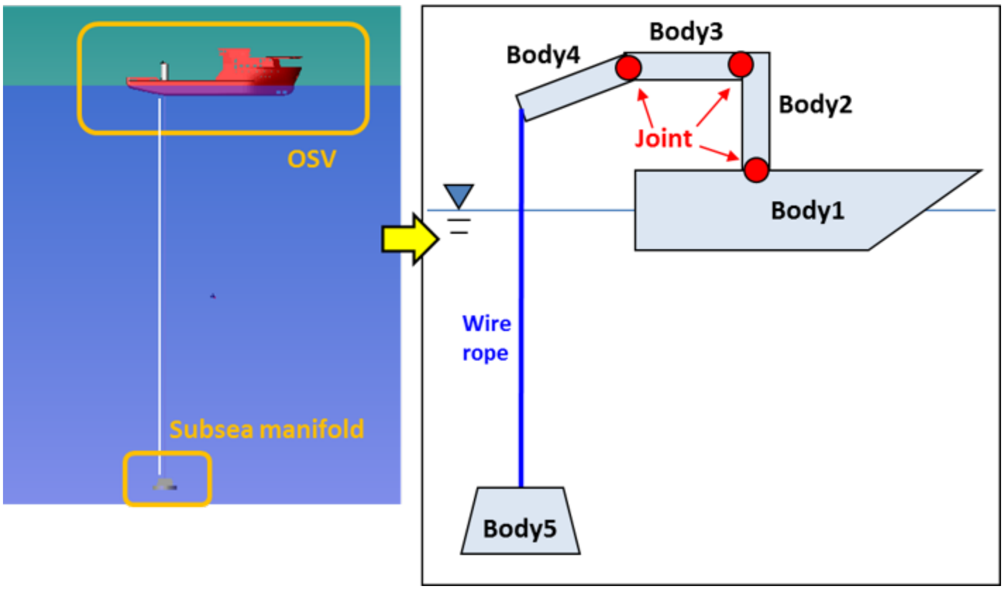


Figure 5.4: Rigid body models of the OSV and the crane.

$$\mathbf{F}_{wirerope} = k(x - x_0) \quad \text{if } x > x_0 \tag{5.13}$$

where, k is the spring constant, which measures how stiff and strong the spring is; and x–x_0 is the distance when the spring is stretched from its equilibrium position.

The added mass of the subsea equipment, which is immersed inside the seawater, should be considered. Because the vertical motion of the equipment is dominant during the operation, we can estimate the added mass roughly by the rectangular plate (DNV, 2011). The vertical added mass of the subsea equipment is obtained from the following equation:

$$\begin{aligned}\mathbf{M}_{\text{added,virtical}} &= \rho C_A V_R \\ &= 1.025 \times 0.66 \times (20.3 \times 15) \\ &= 206[\text{ton}] \end{aligned} \tag{5.14}$$

where, C_A is the added mass coefficient, and V_R is the reference volume. It is interpolated by the ratio of the length and breadth of the subsea equipment Veritas (2011).

5.1.2 Virtual Sensor and Actuator

An MRU is a motion reference device that is capable of measuring pitch, roll, heave, and heading to a high degree of accuracy, and is suitable for any maritime operation. In the real OSV, the measurements are transferred to the control system. Then the control system sends voltage to regulate the rotating speed of the motor. Finally, the torque causes the winch to rotate, and controls the wire rope length. In the HILS environment, the real sensor and actuator should be replaced by a virtual sensor and actuator. As described before, the virtual sensor is implemented by simply receiving the motion data calculated from the virtual mechanical system. The virtual actuator calculates the rotating angle of the winch by integrating the rotating speed from the control system. Finally, the winch changes the length of the wire rope according to the rotating angle.

Electrical motor

The virtual actuator we chose in this study is a electric motor(Robotics Dynamixel XM-430 servo actuator), which can be controlled by ROS directly.

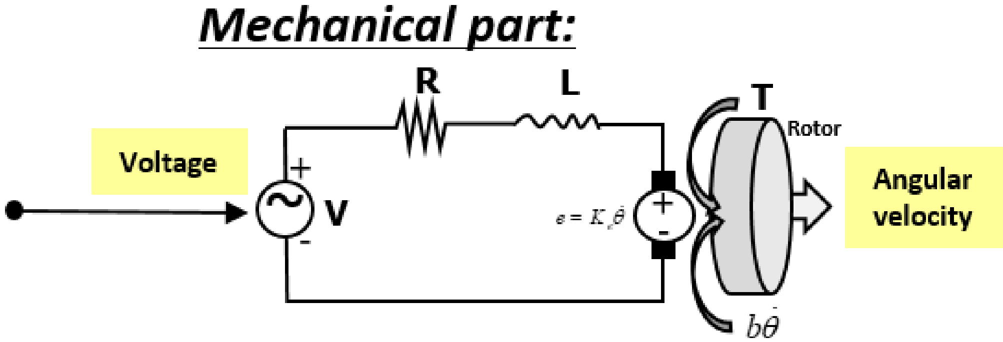


Figure 5.5: Mechanical of dynamixel motor.

The motor torque is proportional to the armature current i by a constant factor K_t.

$$T = K_t i \tag{5.15}$$

The back electromagnetic fields is proportional to teh angular velocity of the shaft by a constant factor K_e.

$$e = K_e\dot{\theta} \tag{5.16}$$

Equations derived from Newton's second law and Kirchhoff's voltage law. where b is the motor viscous friction constant.

$$V = Ri + L\frac{di}{dt} + e \tag{5.17}$$

$$J\ddot{\theta} = T - b\dot{\theta} \tag{5.18}$$

In state-space form, the governing equations above can be expressed by choosing the rotational angel, speed and electric current as the state variables.

$$\frac{d}{dt}\begin{bmatrix}\theta\\ \dot{\theta}\\ i\end{bmatrix} = \begin{bmatrix}0 & 1 & 0\\ 0 & -\frac{b}{J} & -\frac{K}{J}\\ 0 & -\frac{K}{L} & -\frac{R}{L}\end{bmatrix}\begin{bmatrix}\theta\\ \dot{\theta}\\ i\end{bmatrix} + \begin{bmatrix}0\\ 0\\ \frac{1}{L}\end{bmatrix}V \tag{5.19}$$

Hydraulic motor

The servo-valve is a device that uses mechanical motion to deliver a measured amount of fluid power to the hydraulic motor. The mechanical motion is induced by the electrical current that changes in proportion to the displacement of the spool attached on the valve Eq. 5.20.

$$x_v \approx \frac{k_t i}{k(r+b)} \tag{5.20}$$

where, x_v is the spool position changed by the current input i. As a result, the movement of the spool can change the load flow into the two chambers of the hydraulic motor. The load flow q_L represents the average of the flows in the lines connected between the servo-valve and the hydraulic motor, which can be linearized as follows:

$$q_L = K_q x_v - K_c p_L \tag{5.21}$$

where,

$$K_q = C_d b\sqrt{\frac{1}{\rho}(p_s - sgn(x_v)p_L)} \tag{5.22}$$

$$K_c = \frac{C_d b x_v \sqrt{\frac{1}{\rho}(p_s - sgn(x_v)p_L)}}{2(p_s - sgn(x_v)p_L)} \tag{5.23}$$

In addition, the dynamic model of the hydraulic motor can be derived based on the following equations:

$$\frac{V_t}{4\beta}\dot{p}_L = -C_{tm}p_L - D_m\omega_m + q_L \tag{5.24}$$

$$J_t\dot{\omega}_m = -B_m\omega_m + D_m p_L - T_L \tag{5.25}$$

Consequently, the speed of the motor shaft can be controlled by the electrical current. By combining the above Eq. 5.20 with Eq. 5.24, the linear dynamic model of the servo valve controlled hydraulic motor can be formulated by inserting the linearized valve characteristics into the model (Merritt, 1967).

$$\begin{bmatrix} \frac{V_t}{4\beta} & 0 & 0 \\ 0 & J_t & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{p}_L \\ \dot{\omega}_m \\ \dot{\theta}_m \end{bmatrix} = \begin{bmatrix} -K_{cei} & -D_m & 0 \\ D_m & -B_m & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} p_L \\ \omega_m \\ \theta_m \end{bmatrix} + \begin{bmatrix} K_p \\ 0 \\ 0 \end{bmatrix} i + \begin{bmatrix} 0 \\ -T_L \\ 0 \end{bmatrix} \tag{5.26}$$

where, V_t and D_m are the total volume and displacement of the hydraulic motor chambers, respectively; T_L is the load torque on the hydraulic motor, J_t is the moment of inertia of the hydraulic motor, and θ_m denotes the motor shaft angle; and K_{cei} is the leakage coefficient of the hydraulic motor and servo-valve, B_m is the viscous friction coefficient, and β is the bulk modulus. These parameters listed above represent the valve characteristics.

5.1.3 Control System Design

A mathematical description of the PID controller is shown in the following equation:

$$u(t) = K_P[e(t) + \frac{1}{T_I}\int_0^t e(\tau)\mathrm{d}\tau + T_D\frac{\mathrm{d}e(t)}{\mathrm{d}t}] \tag{5.27}$$

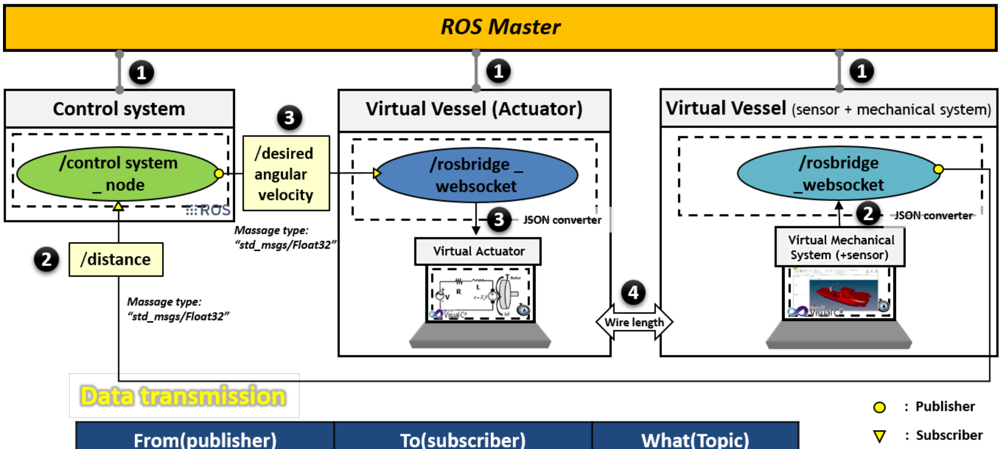
where, $u(t)$ is the control signal, and $e(t)$ is the error signal. The control algorithm of the AHC system is used to minimize the heave motion of the subsea equipment. Therefore, the set-point in the control algorithm is the target depth of the subsea equipment. The difference between the process variable and the set point is the error signal e(t); K_P is the proportional gain, T_I and T_D are the integral and derivative time constants, respectively; and $K_I = \frac{K_P}{K_I}$ and $K_D = K_P T_D$ are the integral gain and derivative gain respectively. Thus these PID gain values (K_P, T_I, and T_D) can be tuned through simulation to achieve the desired AHC performance of the system. In this study, the position of the equipment, which is calculated in real time, is sent to the controller, and is regarded as the process variable input. The control algorithm continuously calculates the difference between the set-point and the current depth of the subsea equipment, to minimize the difference over time, by adjusting the rotating speed of the motor. To accomplish this, it transmits to the virtual actuator, and the process repeats, until the subsea equipment can maintain the target depth.

5.1.4 Integrated Simulation Interface

The three-component system, consisting of the virtual model, the AHC system, and the vritual actuator model, is difficult to integrate, as each piece possesses its own development purposes and environment. Therefore, a simulation interface is necessary. Fig. 5.6 shows the integrated simulation procedure for HILS of the AHC system.

The procedure of HILS for AHC system is described as follows:

1. Start the ROS master in Linux, and create three nodes. Every node register with the ROS master to be able to communicate with the rest of the network.



From(publisher)	To(subscriber)	What(Topic)
/rosbridge _ websocket	/control system _ node	/distance
/control system _ node	/rosbridge _ websocket	/desired angular velocity

Figure 5.6: Procedure of HILS for AHC system.

2. Current position of the payload is published to the control system node through the rosbridge.

3. Based on the transferred data, desired angular velocity is calculated. The control system node use a publisher to send the data to the topic (/desired angular velocity) and the rosbridge websocket node will use a subscriber to receive the same topic. Using the transferred data, virtual actuator calculates the rotating angle.

4. Virtual mechanical system solve the equations of motion and changes current position of the payload.

Table 5.1: Data transmission

From (publisher)	**To(subscriber)**	**What(Topic)**
/rosbridge _ websocket	/control system _ node	/distance
/control system _ node	/rosbridge _ websocket	/desired angular velocity

5.2 Implementation and Simulation Result of HILS for Anti-Heave Control System

5.2.1 Implementation of HILS for Anti-Heave Control System

Fig. 5.7 shows the implementation of the HILS environment for AHC system. The virtual mechanical system module displays the outside view of the OSV and the subsea equipment. One PC is in charge of the virtual model including the virtual actuator, virtual sensor, and virtual mechanical system. The other one is a control system capable of AHC control algorithm. An integrated simulation interface is working in the background.

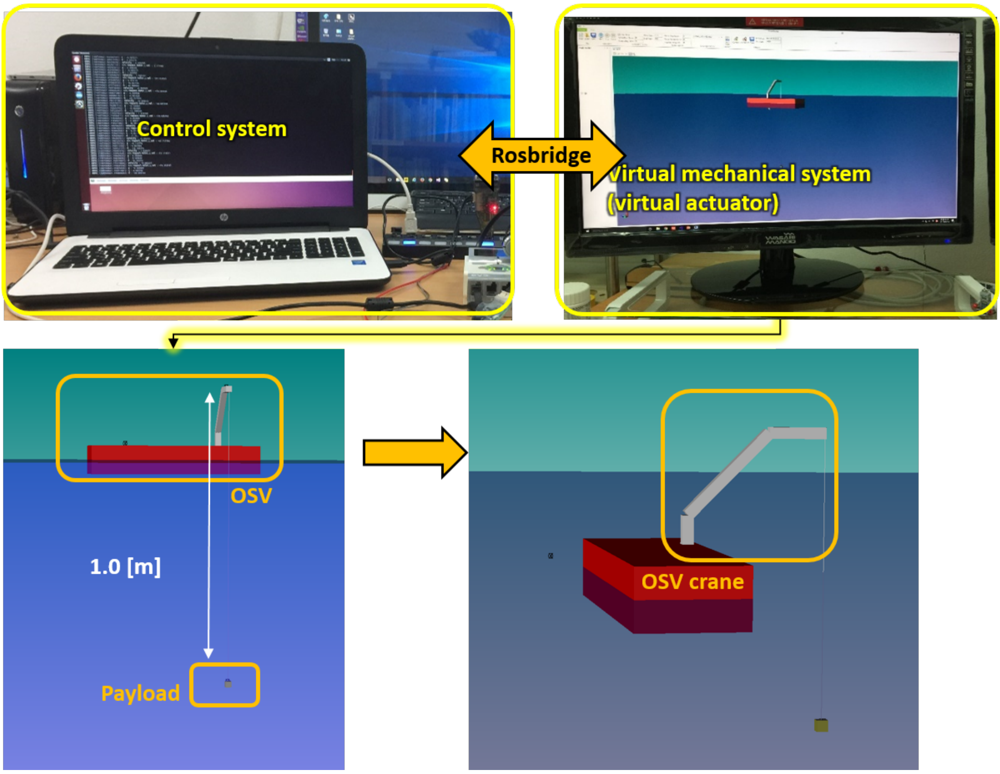


Figure 5.7: Simplified geometry of the system of the OSV-knuckle boom crane.

Table 5.2: Wave conditions for maintaining the depth of the subsea equipment at a constant lever.

Case	Motion type	Wave height [m]	Wave period [s]	Wave direction [deg]	Target depth [m]
1-1	Regular wave	1.0	10	0	-197.1
1-2				45	
1-3				90	
1-4		4.0	10	0	

5.2.2 Simulation Result of HILS for Anti-Heave Control System

Table. 5.2 lists the simulation cases according to the regular wave conditions. Wave conditions, such as wave height, period can induce different motions of the OSV.

Fig. 5.8 shows graphs of the heave motion of the OSV in Case 1-1. According to the wave direction, the OSV crane uses the wire rope to maintain and lower the subsea equipment from the sea surface to the seabed. In other words, the motion of the OSV is affected not only by the wave force, but also by the spring force. As a result, the motion responses of the OSV show irregular phenomena, in spite of the regular wave.

Fig. 5.9 Fig. 5.11 show the graph of the heave motion of the suspended subsea equipment in Case 1-1, Case 1-2, Case 1-3, and Case 1-4. The error of heave motion are reduced by 83.7 % approximately from 0.43 m to 0.07 m in Case 1-1. Therefore, we can conclude that the performance of the AHC system for maintaining of the depth of the subsea equipment at a constant level has an error reduction ratio of more than 80 %.

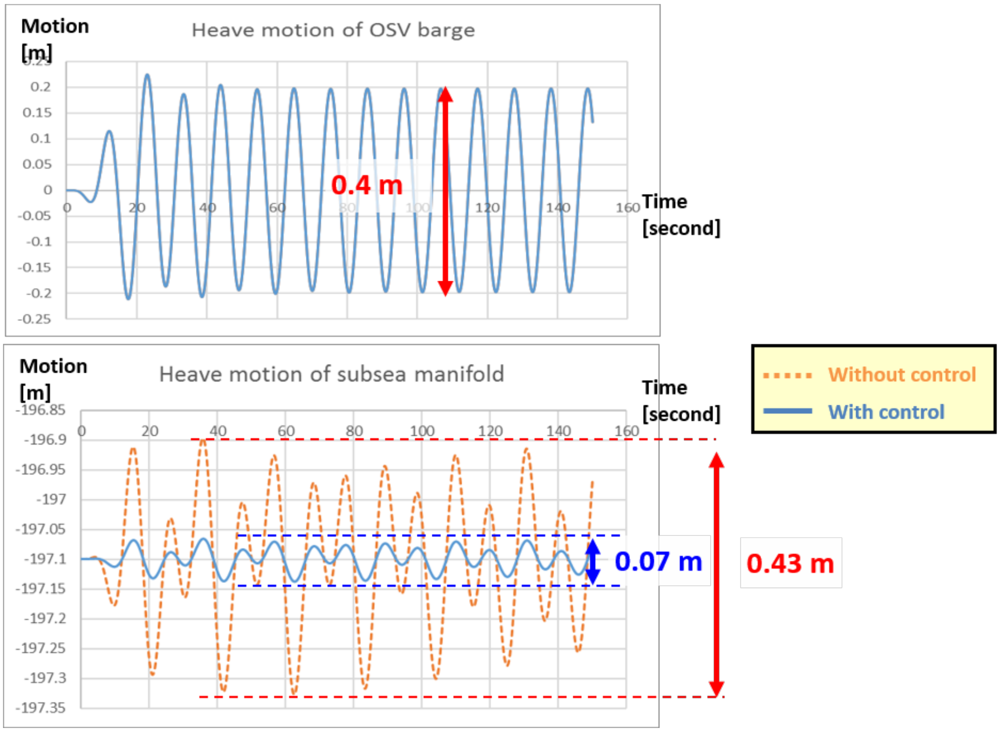


Figure 5.8: Graphs of heave motion of the subsea equipment in Case 1-1.

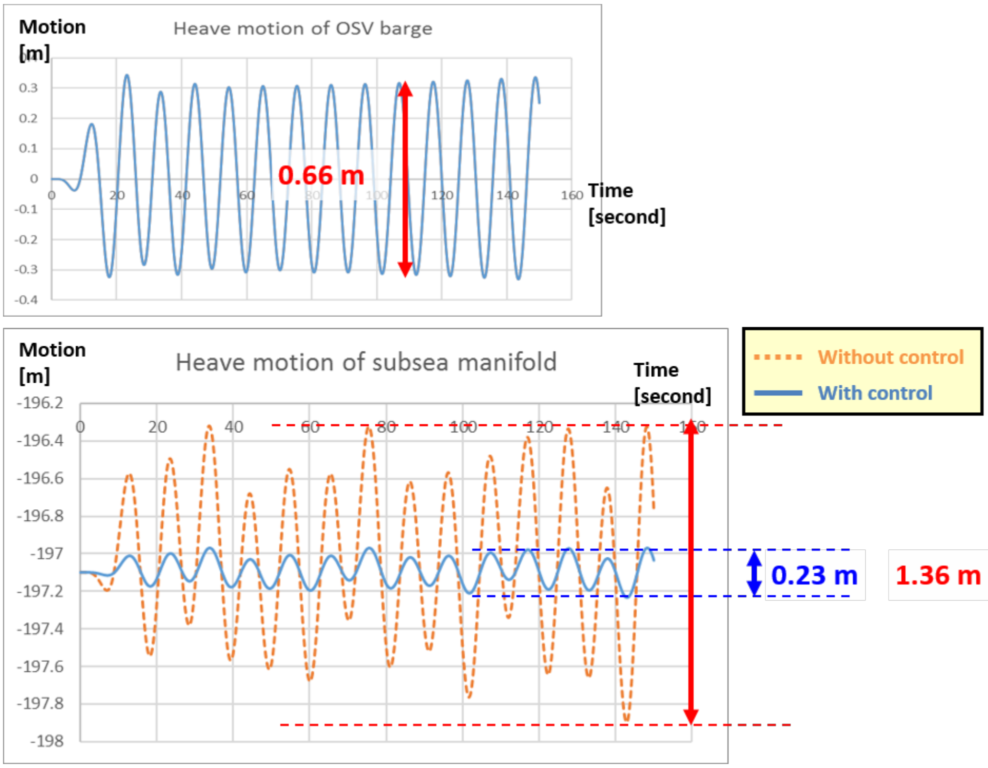


Figure 5.9: Graphs of heave motion of the subsea equipment in Case 1-2.

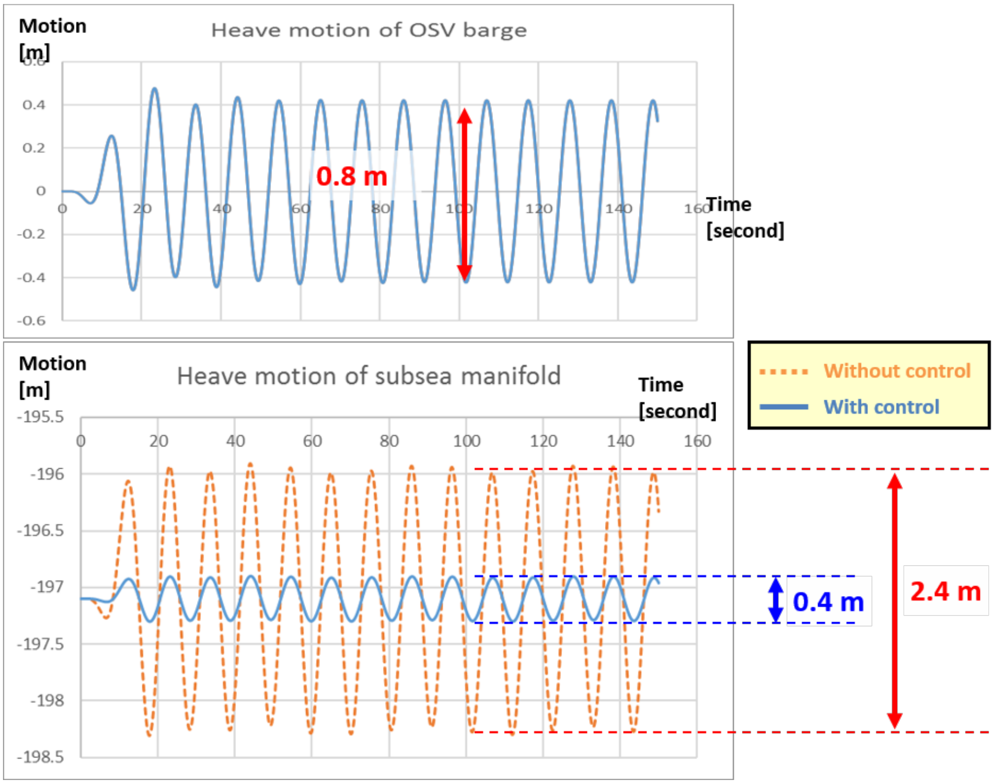


Figure 5.10: Graphs of heave motion of the subsea equipment in Case 1-3.

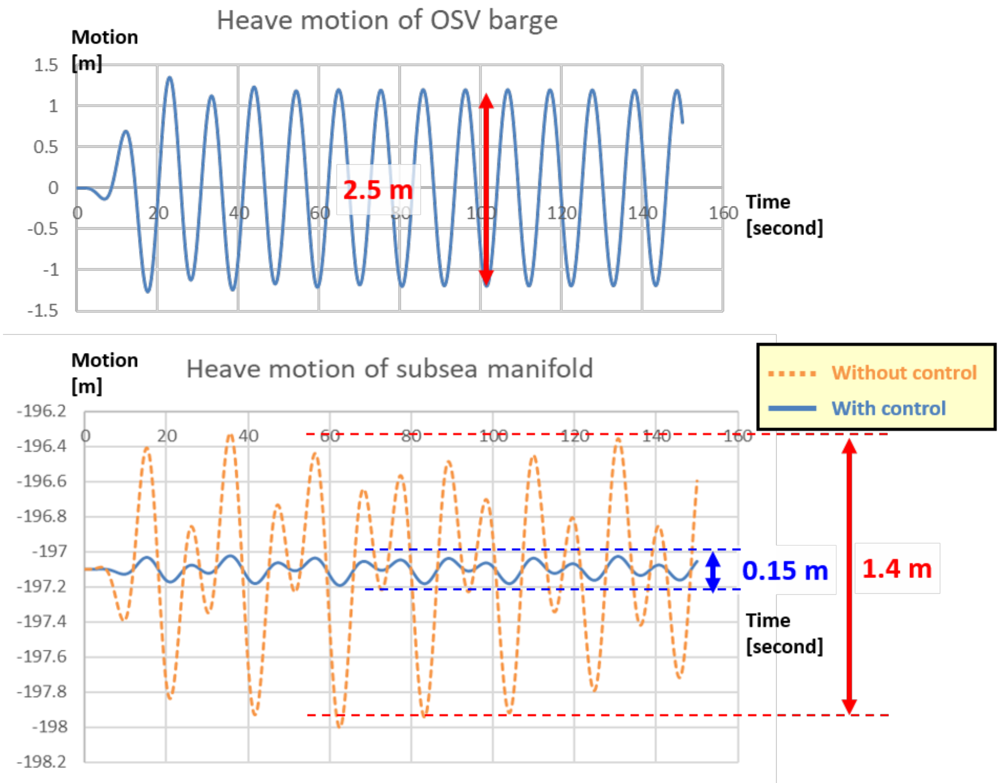


Figure 5.11: Graphs of heave motion of the subsea equipment in Case 1-4.

Table 5.3: Wave conditions for maintaining the depth of the subsea equipment at a constant lever.

Case	Motion type	Wave height [m]	Wave period [s]	Wave direction [deg]	Target depth [m]
2-1	Irregular wave (JONSWAP)	1.0	10	0	-197.1
2-2				45	
2-3				90	
2-4		4.0	10	0	

Table. 5.3 lists the simulation cases according to the irregular wave conditions. Wave conditions, such as wave height, period can induce different motions of the OSV.

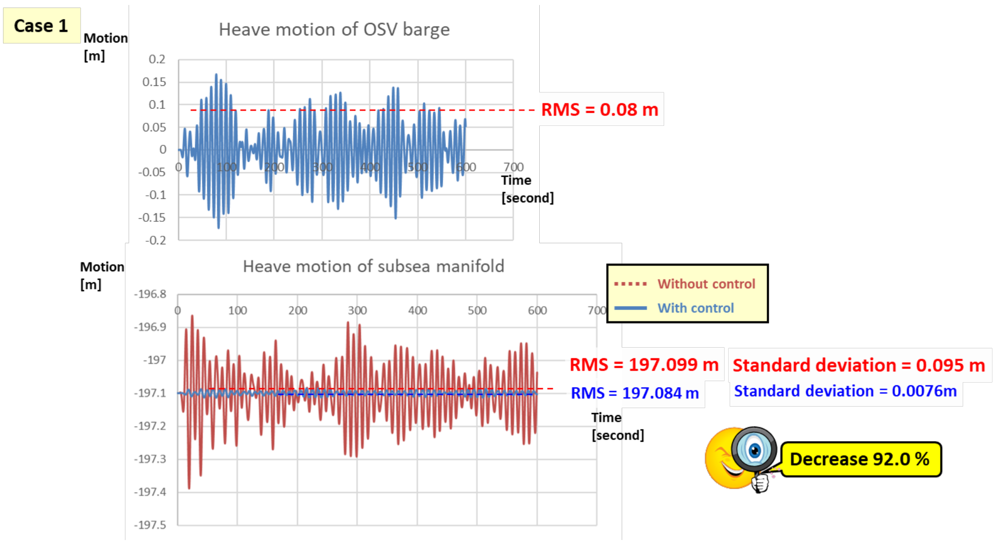


Figure 5.12: Graphs of heave motion of the subsea equipment in Case 2-1.

Fig. 5.12 Fig. 5.15 show the graph of the heave motion of the suspended subsea equipment in Case 2-1, Case 2-2, Case 2-3 and Case 2-4. The error of the standard deviation of heave motion is reduced by 92 % approximately

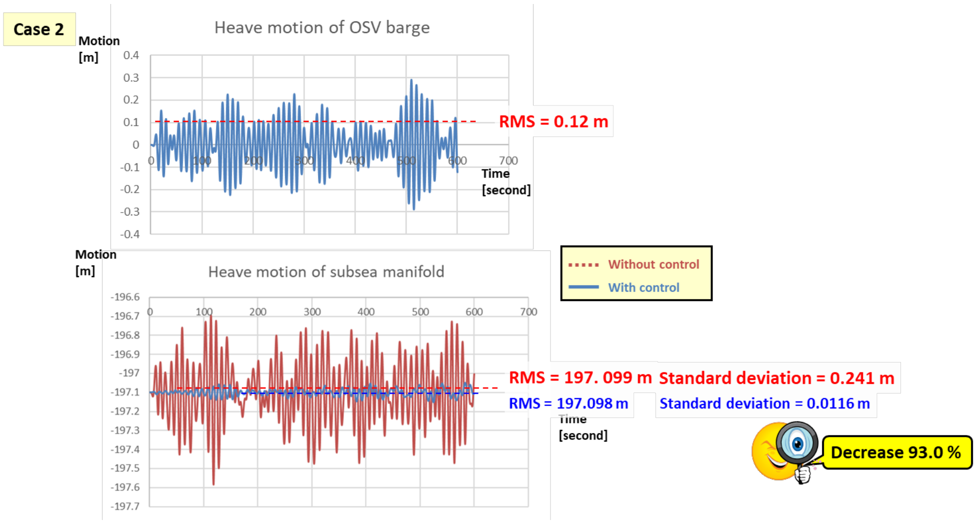


Figure 5.13: Graphs of heave motion of the subsea equipment in Case 2-2.

from 0.095 m to 0.0076 m in Case 2-1. Therefore, we can conclude that the performance of the AHC system for maintaining of the depth of the subsea equipment at a constant level has an error reduction ratio of more than 90 %.

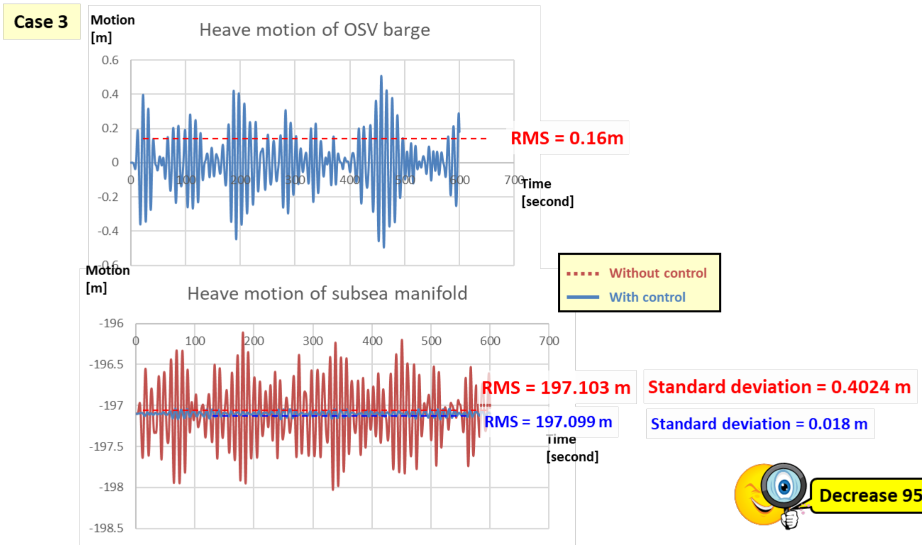


Figure 5.14: Graphs of heave motion of the subsea equipment in Case 2-3.

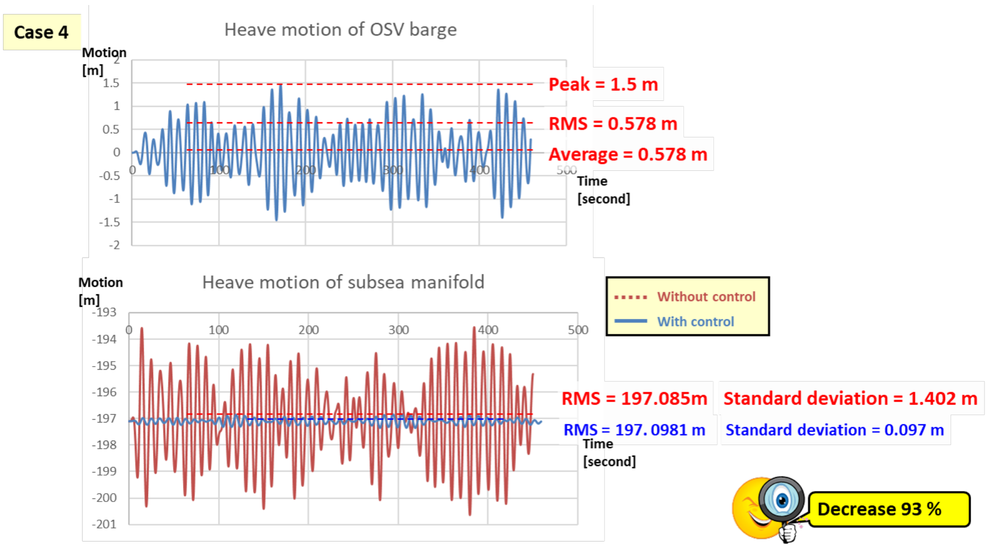


Figure 5.15: Graphs of heave motion of the subsea equipment in Case 2-4.

Table 5.4: Wave conditions for maintaining the depth of the subsea equipment at a constant lever.

Case	**Motion type**	**Wave height**	**Period**	**Direction**	**Target depth**
3-1	Regular wave	0.01 [m]	6.0 [s]	0 [deg]	-0.569 [m]
3-2		0.01 [m]	8.0 [s]	0 [deg]	
3-3		0.01 [m]	10.0 [s]	0 [deg]	

Using the same PID controller as illustrated in Table. 5.2 and Table. 5.3. We applied it to the small scale OSV model which is same to the hardware simulation model. Table. 5.4 lists the simulation Cases according to the wave conditions. Wave conditions, such as wave height, period can induce different motions of the OSV.

Fig. 5.16 shows graphs of the heave motion of the OSV in Case 3-1. According to the wave direction, the OSV crane uses the wire rope to maintain and lower the subsea equipment from the sea surface to the seabed. In other words, the motion of the OSV is affected not only by the wave force, but also by the spring force. As a result, the motion responses of the OSV show irregular phenomena, in spite of the regular wave.

Fig. 5.17 shows the graph of the heave motion of the suspended subsea equipment in Case 3-1. The error of heave motion are reduced by 81.81 % approximately from 0.011 m to 0.002 m in Case 3-1. Therefore, we can conclude that the performance of the AHC system for maintaining of the depth of the subsea equipment at a constant level has an error reduction ratio of more than 80 %.

In Case 3-2, the wave height is 0.01 m, wave period is 8.0 s. Fig. 5.18 shows the graph of the heave motion of the OSV. Fig. 5.19 shows the graph

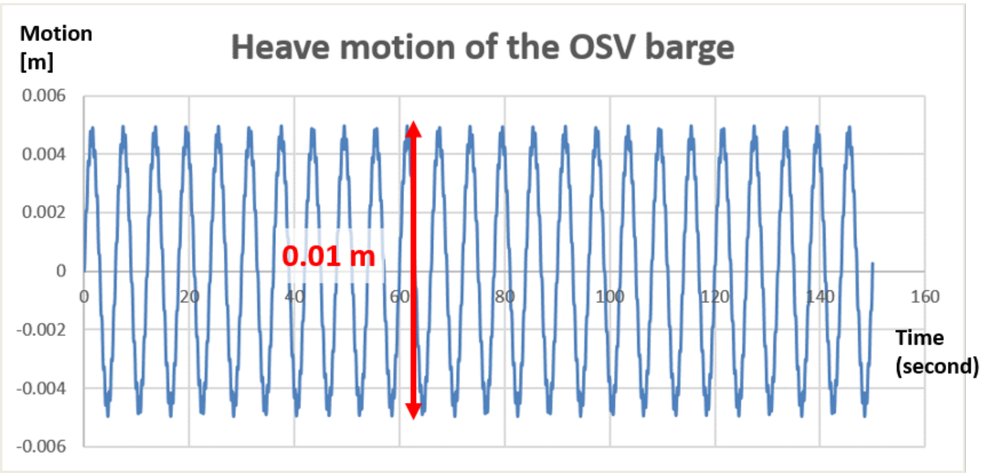


Figure 5.16: Graphs of heave motion of the OSV in Case 3-1.

of the heave motion of the subsea equipment in Case 3-2. The heave motion with control is reduced by approximately 80.3 % from (0.01 m to 0.31 m) in the regular wave.

In Case 3-3, the wave height is 0.01 m, wave period is 10.0 s. Fig. 5.20 shows the graph of the heave motion of the OSV. Fig. 5.21 shows the graph of the heave motion of the subsea equipment in Case 1-2. The heave motion with control is reduced by approximately 70 % from (0.01 m to 0.31 m) in the regular wave.

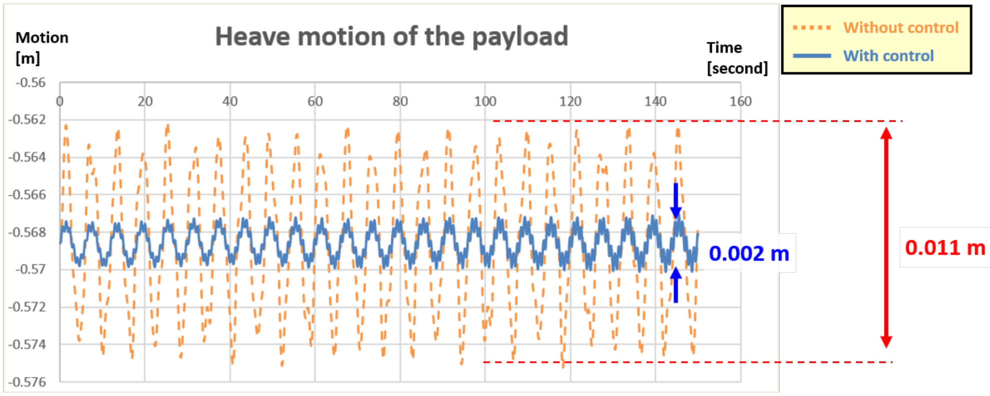


Figure 5.17: Graphs of heave motion of the subsea equipment in Case 3-1.

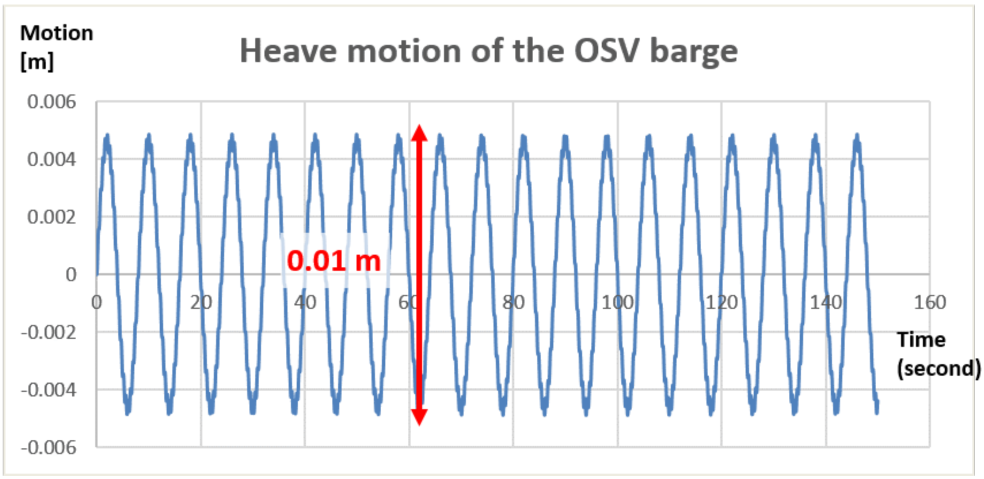


Figure 5.18: Graphs of heave motion of the OSV in Case 3-2.

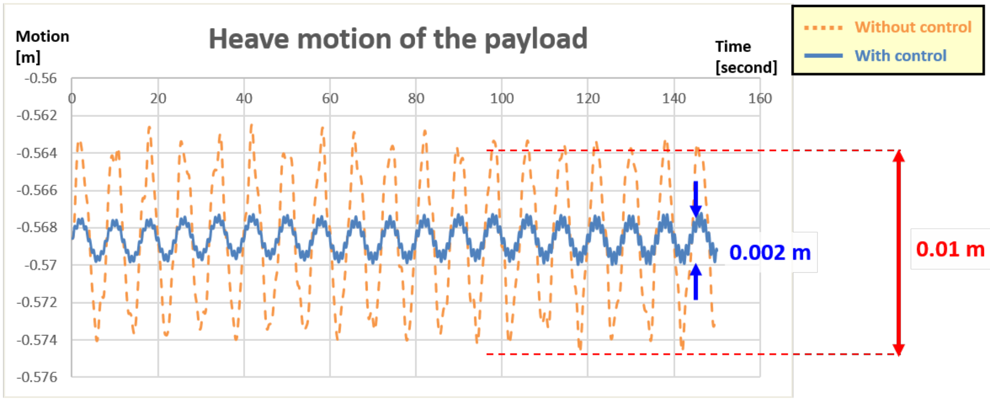


Figure 5.19: Graphs of heave motion of the subsea equipment in Case 3-2.

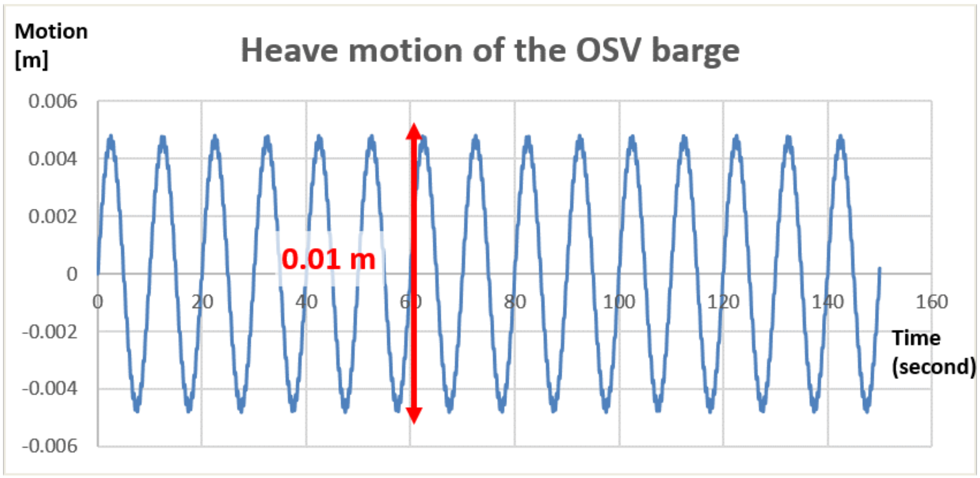


Figure 5.20: Graphs of heave motion of the OSV in Case 3-3.

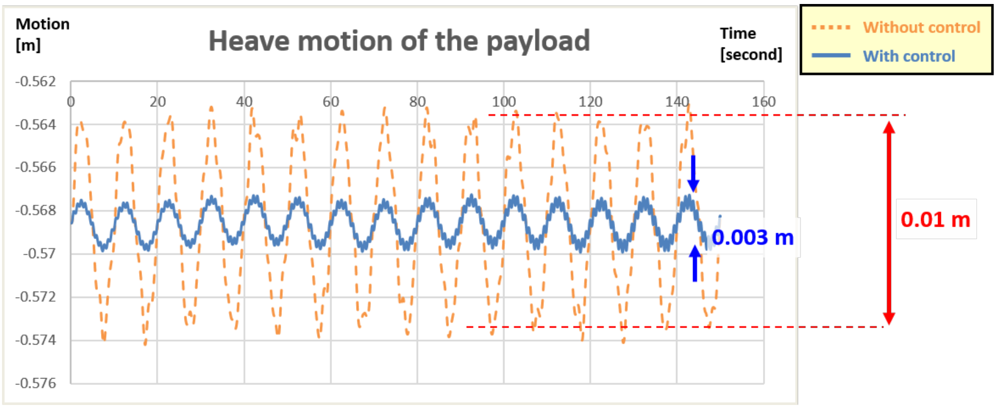


Figure 5.21: Graphs of heave motion of the subsea equipment in Case 3-3.

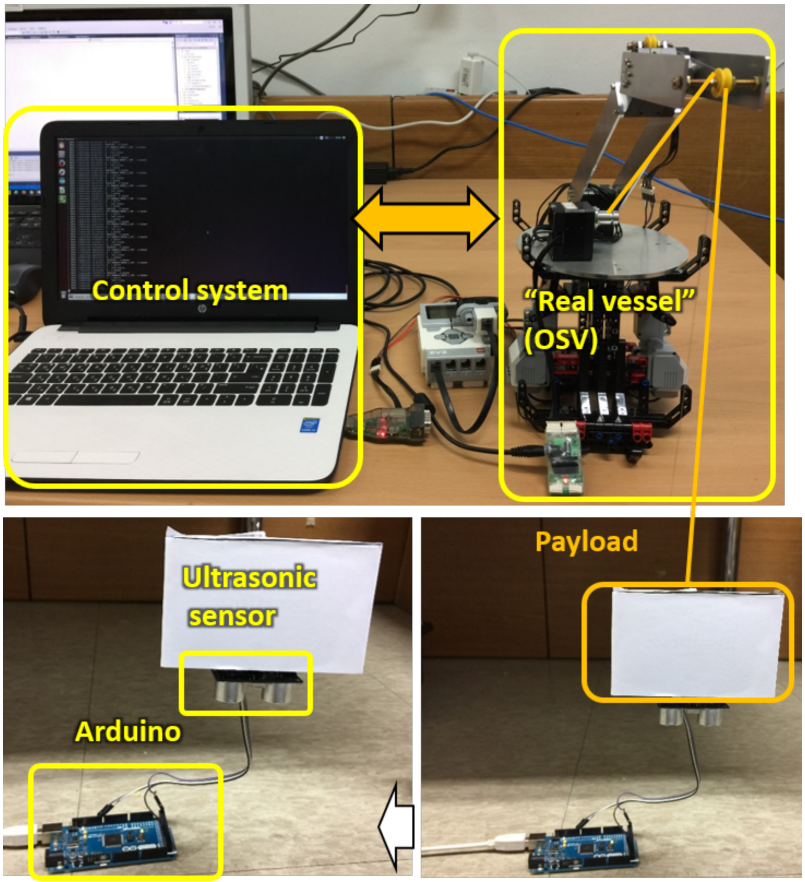


Figure 5.22: Hardware setup for validation of the AHC system.

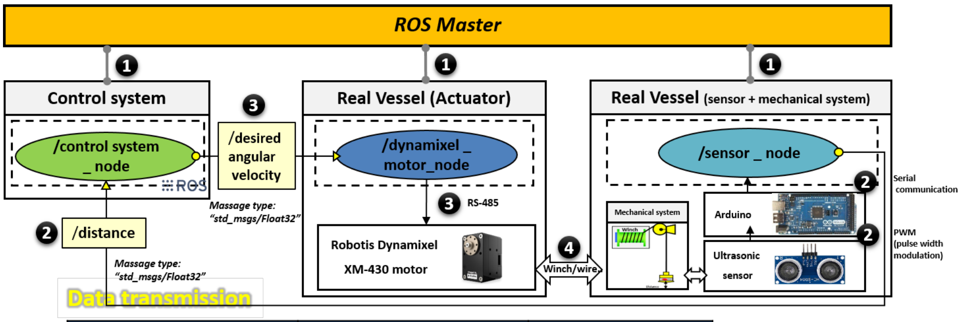
5.3 Validation of HILS for Anti-Heave Control System

5.3.1 Hardware Setup

To validate the effectiveness of the AHC system, we conduct an experiment with a real dynamixel motor and sensor, which is represented in Fig. 5.22.

The control system, proposed in the HILS environment, was used to control the dynamixel motor. The dynamixel motor connect with a payload by the wire rope. We used a ultrasonic sensor, which is controlled by an Arduino controller, to measure the current position of the payload. Moreover, we used a motion platform to simulate the wave motion.

Compared with HILS for AHC system, we replaced the virtual actuator and motor with a real ones. The whole procedure is shown in Fig. 5.23.



From(publisher)	To(subscriber)	What(Topic)
Ultrasonic sensor	Arduino	Electrical current
Arduino	/sensor _ node	Distance
/sensor _ node	/control system _ node	/distance
/control system _ node	/dynamixel _ motor _ node	/desired angular velocity
/dynamixel _ motor _ node	Dynamixel motor	Angular velocity

Figure 5.23: Validation procedure of AHC system.

The validation procedure for AHC system is summed up as follows:

1. Start the ROS master in Linux, and create three nodes. Every node register with the ROS master to be able to communicate with the rest of the network.

2. The ultrasonic sensor measures the current position of the payload, then the sensor node use a publisher to send current position data to the topic (/distance) and the control system node will use a subscriber to receive the same topic.

3. Based on the transferred data, desired angular velocity to control dynamixel motor is calculated. The control system node publish the topic

(/desired angular velocity) to the dynamixel motor node. As a result, angular velocity of the dynamixel motor will be changed.

4. Finally, the position of the payload will reach the desired position.

Table 5.5: Data transmission.

From (publisher)	**To(subscriber)**	**What(Topic)**
Ultrasonic sensor	Arduino	Electrical current
Arduino	/sensor _ node	/desired angular velocity
/sensor _ node	/control system _ node	/distance
/control system _ node	/dynamixel _ motor _ node	/desired angular velocity
/dynamixel _ motor _ node	Dynamixel motor	Angular velocity

5.3.2 Comparison Result

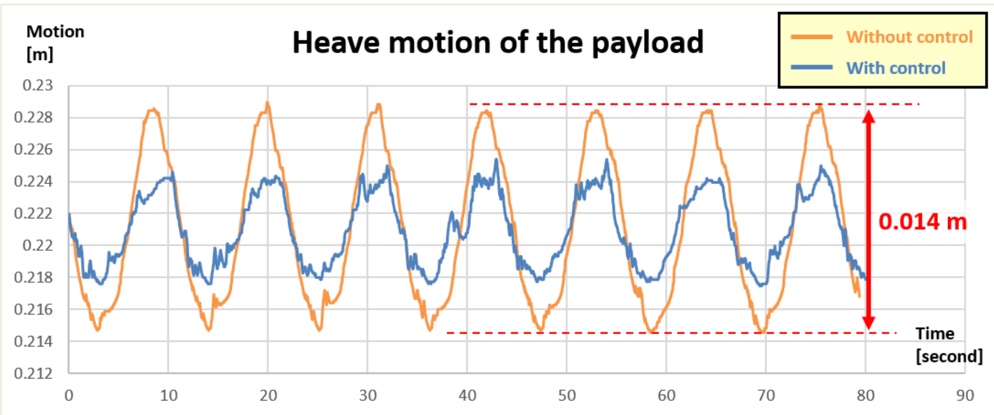


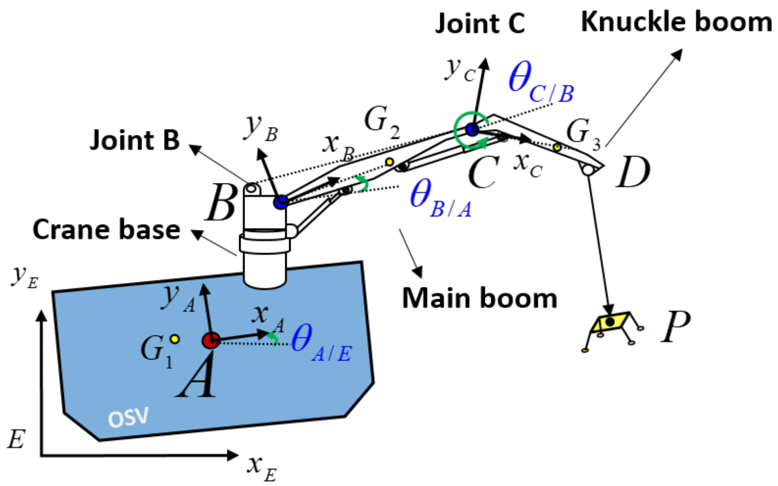
Figure 5.24: Graphs of comparison result the heave motion of the subsea equipment.

5.4 Configuration of Anti-Sway Control System

5.4.1 Mechanical System for Knuckle Boom Crane

The motion of the OSV-mounted knuckle boom crane in an inertial coordinate system is depicted in Fig. 5.25. The formulation can be found in Zhao et al. (2017). To establish the whole mechanical system, the OSV equation, knuckle boom crane, and subsea equipment including the wave loads as an external force must be formulated. They should be calculated simultaneously because these bodies are closely related to each other by joints and wire rope. It is difficult, however, to directly apply the Newton–Euler equation to the multibody system because of the existing constraints and constraint forces, so an embedding technique that leads to the elimination of the constraint forces is chosen Shabana (2009). A number of equations that are equal to the number of the systemic degrees of freedom can be obtained.

To obtain the differential equations, it is necessary to use a velocity transformation matrix. Here, the chosen independent coordinates are $q = [{}^{E}x_{B/A}$, ${}^{E}y_{B/A}$, $\theta_{A/E}$, $\theta_{B/A}$, $\theta_{C/B}$, ${}^{E}x_{P}$, ${}^{E}y_{P}]^{T}$, as these represent the positions and angular positions of the mechanical systems. Since mechanical systems are utilized for the antisway-controller design, only the heave, sway, and roll motions of the vessel that play a pivotal role in the pendulum motion of the subsea equipment are considered. The OSV with the crane can be modeled as a planar mechanism using revolute joints, as shown in Fig. 5.25. According to the kinematic structure, the knuckle boom crane can be treated as three links that are connected by two revolute joints. The bottom of the crane is fixed at the OSV, and the points B and C represent the revolute joints that are actuated by two hydraulic-motor systems. By controlling the two revolute-joint angles $\theta_{B/A}$ and $\theta_{C/B}$, the points P and D can be kept at the same vertical line. As a result, the swaying motion of the subsea equipment can be reduced.



$$\theta_{G_1/E} = \theta_{A/E}$$

$$\theta_{G_2/E} = \theta_{A/E} + {}^{E}\theta_{B/A}$$

$$\theta_{G_3/E} = \theta_{A/E} + {}^{E}\theta_{B/A} + {}^{E}\theta_{C/B}$$

Figure 5.25: Simplified geometry of the system of the OSV-knuckle boom crane.

To derive the motion equations, the coordinate systems must be clearly defined. The A-Frame, B-Frame, and C-Frame are placed at the points A, B, and C, respectively. G_i is the center of the mass of each body. The notation of $i = 1$ represents the OSV hull; $i = 2$ and 3 represents the two crane links. The position vectors from the inertial coordinate system, the E-Frame, to the points A, B, and C are defined as ${}^{E}r_A$, ${}^{E}r_B$, and ${}^{E}r_c$, respectively. Therefore, the position vector ${}^{E}r_{G_i}$ can be expressed by Eqs. 5.28 to 5.30, as follows:

$$ {}^{E}\mathbf{r}_{G_1} = {}^{E}\,\mathbf{r}_A + {}^{E}\,\mathbf{R}_A\,{}^{A}\mathbf{r}_{G_1} \tag{5.28}$$

$$ {}^{E}\mathbf{r}_{G_2} = {}^{E}\,\mathbf{r}_A + {}^{E}\,\mathbf{R}_A\,{}^{A}\mathbf{r}_B + {}^{E}\,\mathbf{R}_B\,{}^{B}\mathbf{r}_{G_2} \tag{5.29}$$

$$^{E}\mathbf{r}_{G_3} =^{E} \mathbf{r}_A +^{E} \mathbf{R}_A \,^{A}\mathbf{r}_B +^{E} \mathbf{R}_B \,^{B}\mathbf{r}_C +^{E} \mathbf{R}_C \,^{C}\mathbf{r}_{G_3} \tag{5.30}$$

where the matrices $^{E}\mathbf{R}_A$, $^{E}\mathbf{R}_B$, and $^{E}\mathbf{R}_C$ are the respective rotational transformations. It is convenient to determine the coordinates of the mass centers that correspond to the inertial coordinate system using the rotational transformations. Using these relationships, the velocity-transformation matrix $\mathbf{J}$ can be obtained, as shown in the appendix.

The equations of the systemic motions can be written in a matrix form by inserting $\mathbf{J}$, as follows:

$$\tilde{\mathbf{M}}\ddot{q} + \tilde{\mathbf{K}} =^{E} \tilde{\mathbf{F}}^{e} + \tau \tag{5.31}$$

where

$$\tilde{\mathbf{M}} = \mathbf{J}^{T}\mathbf{M}\mathbf{J} \tag{5.32}$$

$$\tilde{\mathbf{K}} = \mathbf{J}^{T}\mathbf{M}\dot{\mathbf{J}} \tag{5.33}$$

$$^{E}\tilde{\mathbf{F}}^{e} = \mathbf{J}^{T}\,^{E}\mathbf{F}^{e} \tag{5.34}$$

$$^{E}\mathbf{F}^{e} = \mathbf{F}_{hydrodynamic} + \mathbf{G} + \mathbf{F}_{s} \tag{5.35}$$

The first term in Eq. 5.31 represents the inertial forces, the second term represents the Coriolis and centrifugal forces, $^{E}\mathbf{F}^{e}$ represents the external forces including the hydraulic dynamic force acting on the OSV, and $\mathbf{G}$ and $\mathbf{F}_s$ represent the gravity force of the bodies and the wire-rope tension, respectively. τ is the applied torques on the joints B and C that are driven by the hydraulic-motor system and can be calculated by the antisway controller (Murray et al. 1993). In addition, the hydrodynamic force can be divided into two parts: the wave-exciting force Fexciting that is exerted by the incident wave, and the radiation force Fradiation from the wave that is generated by the motion of the vessel itself. The Cummins equation(Cummins (1962)) can be used to calculate the radiation force, for which the floater impulse response in the time domain is considered:

$$\mathbf{F}_{hydrodynamic} = \mathbf{F}_{exciting} + \mathbf{F}_{radiation} \tag{5.36}$$

The motion equations for the subsea equipment are given by Eq. 5.37. The position coordinates of the subsea equipment that are relative to the inertial coordinate system can be obtained as follows:

$$\mathbf{M}^{E}\ddot{\mathbf{r}}_{P} = \mathbf{F}_{s} + \mathbf{G}_{P} \tag{5.37}$$

where the matrix M is the inertia matrix of the subsea equipment and G_P is the gravitational force. It should be noted that F_s is the tension vector acting at both ends of the wire rope that is of the opposite direction.

5.4.2 Anti-Sway Control System Design

We applied the sliding-mode control to the underactuated system in this study. Fig. 5.26 shows a block diagram of the selected control law that will ensure the convergence of the four controlled state variables of $q = [\theta_{B/A}, \theta_{C/B}, {}^{E}x_{P}, {}^{E}y_{p}]^T$ to the desired values of $q = [\theta_{B/A_d}, \theta_{C/B_d}, {}^{E}x_{P_d}, {}^{E}y_{p_d}]^T$.

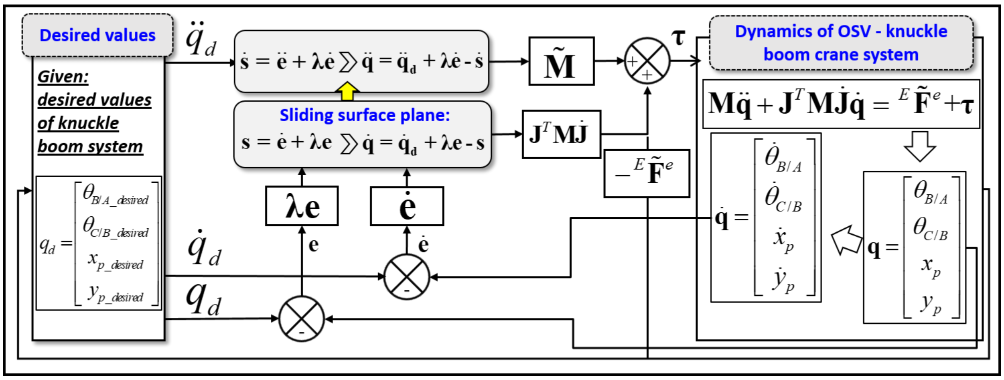


Figure 5.26: Block diagram of the sliding-mode control system.

The objective of the controller is to force all of the state variables to a sliding-surface plane on the state space. Therefore, the sliding-surface plane $\mathbf{s}$, which is given by Eq. 5.38 Bergerman and Xu (1996) and Utkin et al. (2009), is denoted as follows:

$$\mathbf{s} = \dot{\mathbf{e}} + \lambda \mathbf{e} = (\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \lambda(\mathbf{q}_d - \mathbf{q}) \tag{5.38}$$

where e is the error between the desired and current values of the state variables, and λ is the gain matrix. In terms of designing the desired state variables $\mathbf{q}_d$, the main strategy is shown as follows. At first, it is desirable to maintain the crane tip (point D) at a fixed position in space, despite the movement of the crane base. By using the kinematic relation, Eqs. 18 and 19 can be obtained as follows:

$$^{E}\mathbf{r}_{D_initial} = F_{0,3}\,^{C}\mathbf{r}_D, \tag{5.39}$$

$$F_{0,3} = \begin{bmatrix} \cos\theta_{A/E} & -\sin\theta_{A/E} & ^{E}x_A \\ \sin\theta_{A/E} & \cos\theta_{A/E} & ^{E}y_A \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \cos\theta_{B/A_d} & -\sin\theta_{B/A_d} & ^{A}x_B \\ \sin\theta_{B/A_d} & \cos\theta_{B/A_d} & ^{A}y_B \\ 0 & 0 & 1 \end{bmatrix}$$
$$\times \begin{bmatrix} \cos\theta_{C_{B_d}} & -\sin\theta_{C_{B_d}} & ^{B}x_C \\ \sin\theta_{C_{B_d}} & \cos\theta_{C_{B_d}} & ^{B}y_C \\ 0 & 0 & 1 \end{bmatrix} \tag{5.40}$$

where $^{A}r_B = [^{A}x_B, ^{A}y_B]^T$ is the local position of the crane base B in the A-Frame, and $^{B}r_C = [^{B}x_C, ^{B}y_C]^T$ is the local position of main boom C in the B-Frame. The desired joint angles $q_{d_1} = [\theta_{B/A_d}, \theta_{C/B_d}]^T$ can be obtained by solving the inverse kinematics equation. The desired position vector in the global coordinate space of the crane tip is denoted by $^{E}r_{D_{\text{initial}}} = [^{E}x_{D_{\text{initial}}}, ^{E}y_{D_initial}]^T$.

${}^{C}r_D = [{}^{C}x_D, {}^{C}y_D]^T$ is the position vector of the crane tip in the local coordinate space. Then, the lateral position between the crane tip and the subsea equipment must satisfy the following equation to stabilize the subsea-equipment position:

$$\mathbf{q}_{d_2} = \begin{bmatrix} {}^{E}x_{P_d} \\ {}^{E}y_{P_d} \end{bmatrix} = \begin{bmatrix} {}^{E}x_D \\ {}^{E}y_D - l \end{bmatrix} \tag{5.41}$$

where ${}^{E}r_D = [{}^{E}x_D, {}^{E}y_D]^T$ describes the position vector of the crane tip D in the global coordinate space. Using Eq. 5.42, ${}^{E}r_D$ is written in terms of the state variables $\mathbf{q}$, as follows:

$$\begin{aligned} {}^{E}\mathbf{r}_D &= {}^{E}\mathbf{r}_A + {}^{E}\mathbf{R}_A\,{}^{A}\mathbf{r}_B + {}^{E}\mathbf{R}_B\,{}^{B}\mathbf{r}_C + {}^{E}\mathbf{R}_C\,{}^{C}\mathbf{r}_D \\ &= {}^{E}\mathbf{r}_A + \begin{bmatrix} \cos\theta_{A/E} & -\sin\theta_{A/E} \\ \sin\theta_{A/E} & \cos\theta_{A/E} \end{bmatrix} \begin{bmatrix} {}^{A}x_B \\ {}^{A}y_B \end{bmatrix} \\ &+ \begin{bmatrix} \cos\theta_{A/E}+\theta_{B/A} & -\sin\theta_{A/E}+\theta_{B/A} \\ \sin\theta_{A/E}+\theta_{B/A} & \cos\theta_{A/E}+\theta_{B/A} \end{bmatrix} \cdot \begin{bmatrix} {}^{B}x_C \\ {}^{B}y_C \end{bmatrix} \\ &+ \begin{bmatrix} \cos\theta_{A/E}+\theta_{B/A}+\theta_{C/B} & -\sin\theta_{A/E}+\theta_{B/A}+\theta_{C/B} \\ \sin\theta_{A/E}+\theta_{B/A}+\theta_{C/B} & \cos\theta_{A/E}+\theta_{B/A}+\theta_{C/B} \end{bmatrix} \cdot \begin{bmatrix} {}^{C}x_D \\ {}^{C}y_D \end{bmatrix} \end{aligned} \tag{5.42}$$

As a result, the desired position vector of the subsea equipment P ($\mathbf{q}_{d_2}$) can be calculated. Notably, ${}^{C}\mathbf{r}_D = [{}^{C}x_D, {}^{C}y_D]^T$ is the local position of the crane tip D in the C-Frame. To analyze whether or not the controlled state variables reach the sliding-surface plane, the differentiation of the Lyapunov function $V = \mathbf{s}^T\mathbf{s}$ should be considered as follows:

$$\dot{V} = \mathbf{s}^T\dot{\mathbf{s}} < 0 \tag{5.43}$$

Furthermore, Eq. 5.44 was used to guarantee the systemic stability, as follows:

$$\dot{\mathbf{s}} = -\mathbf{K}_1\mathbf{s} - \mathbf{K}_2\text{sgn}(\mathbf{s}) \tag{5.44}$$

where $\mathbf{K}_1$ and $\mathbf{K}_2$ are the gain matrices. Moreover, $\dot{\mathbf{s}}$ can be written as

$$\dot{\mathbf{s}} = \ddot{\mathbf{s}} + \lambda\dot{\mathbf{s}} = \ddot{\mathbf{q}}_d - \ddot{\mathbf{q}} + \lambda(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) \tag{5.45}$$

By substituting $\ddot{\mathbf{q}}$ and $\dot{\mathbf{q}}$ of Eq. 5.31 with and from Eq. 5.38 and Eq. 5.44, respectively, the following equation can be obtained:

$$\begin{aligned}\tau = &\tilde{\mathbf{M}}(\mathbf{q})(\ddot{\mathbf{q}}_d - \lambda\dot{\mathbf{q}} + \lambda\dot{\mathbf{q}}_d) + \mathbf{J}^T\mathbf{M}\dot{\mathbf{J}}(\mathbf{s} + \dot{\mathbf{q}}_d - \lambda\mathbf{q} + \lambda\mathbf{q}_d)\\ &+^E\mathbf{F}^e(\mathbf{q}) + \mathbf{M}\mathbf{q}(-\mathbf{K}_1\mathbf{s} - \mathbf{K}_2\text{sng}(\mathbf{s}))\end{aligned} \tag{5.46}$$

Thus, the control force in Eq. 5.46 guarantees the convergence of the controlled state variables to their desired values.

5.4.3 Implementation and Simulation Result of Anti-Sway Control

The performance analysis of the ASC was conducted using the proposed simulation framework. The principal dimensions of the OSV and the subsea equipment that are used in this study are illustrated in Table. 5.6. When the subsea equipment and the crane tip produce a displacement difference, the control system will use these errors to calculate the corresponding voltage of each motor. The control system then sends a voltage to regulate the rotational speed of the motor so that the crane joints on the OSV crane will be actuated. As a result, the sway angle of the suspended subsea equipment, which is calculated by the displacement errors in both directions, can be reduced.

Table 5.6: Dimensional principle of the autonomous OSV and subsea equipment.

Autonomous OSV	
Length O.A	90.0 m
Length at waterline	86.6 m
Breadth	18.0 m
Depth	7.85 m
Design draft	6.30 m
Deadweight	4,213 tons
Subsea equipment	
Weight	30 tons

The simulation cases according to the wave condition are listed in Table. 5.7. The wave conditions including the wave amplitude and the wave period can induce different OSV motions.

Fig. 5.27 shows the graphs of the sway angle of the subsea equipment under different wave conditions. The simulation result demonstrates that it is suitable to utilize the sliding-mode control for a nonlinear system, as it is effective under uncertain conditions.

Table 5.7: Wave conditions for controlling the sway motion of the subsea equipment.

Case	**Wave height (m)**	**Wave period (s)**	**Heading angle (deg)**	**Heave motion of OSV barge (m)**	**Roll motion of OSV barge (deg)**
1	1.0	10	90	0.6	1.14
2	1.5	10	90	1.0	1.43
3	1.0	12	90	0.7	1.19

Similarly, Fig. 5.28 shows the comparison results of the displacement error in the x coordinate of the subsea equipment in Case 1. The displacement error in the x coordinate of the subsea equipment is a significant factor in the presentation of the effect of the proposed ASC system. It is possible to conclude that the displacement error in the x coordinate of the suspended subsea equipment can be minimized when the controller is running.

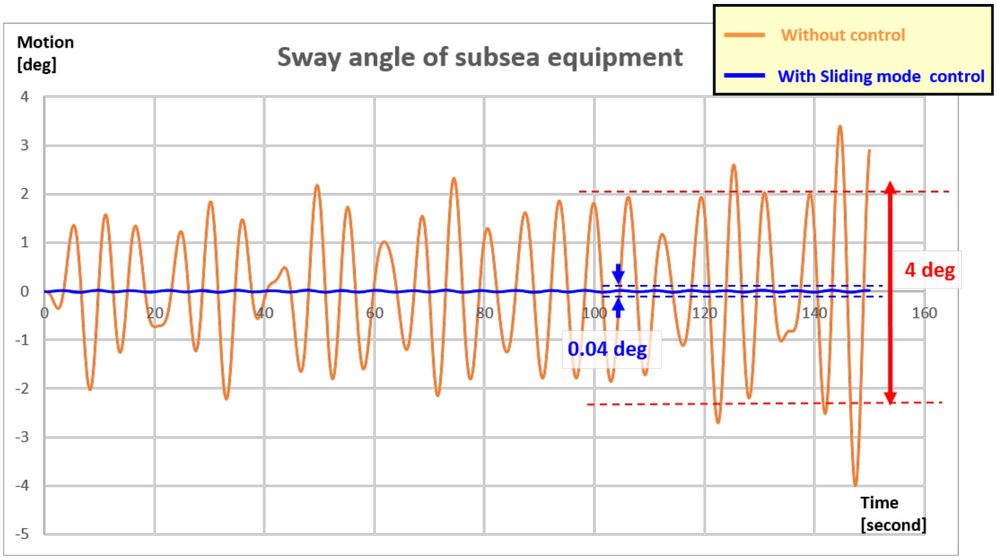


Figure 5.27: Graphs of sway angle of the subsea equipment in Case 1.

Additionally, Fig. 5.29 shows the trajectory of the suspended subsea equipment along the x and y coordinates. The subsea equipment moves like a pendulum in the absence of a control system, and the maximum range of the motion along the x coordinate is 1 m. When the sliding-mode control method works on the crane, the range along the x coordinate can be reduced to 0.008 m. At the same time, a slight oscillation occurs along the y coordinate that is caused by the movement of the crane tip.

Fig. 5.30, Fig. 5.31, and Fig. 5.32 show the comparison results of the anti-

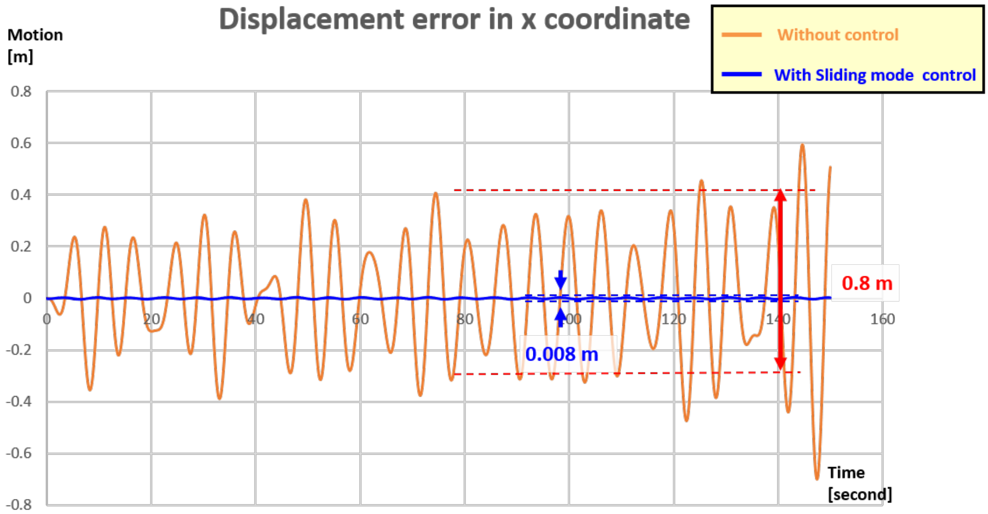


Figure 5.28: Graphs of displace error in the x coordinate of the subsea equipment in Case 1.

sway control of the subsea equipment when the sliding-mode control methods are used in Case 2, and the sway motions of the subsea equipment are reduced by approximately 98 %. The last case of the ASC, Case 3, is illustrated in Fig. 5.33, Fig. 5.34, and Fig. 5.35. The errors of the sway angle are reduced by approximately 99%.

The analysis and control design for the ASC of subsea equipment that is suspended by the OSV crane are presented in this chapter.

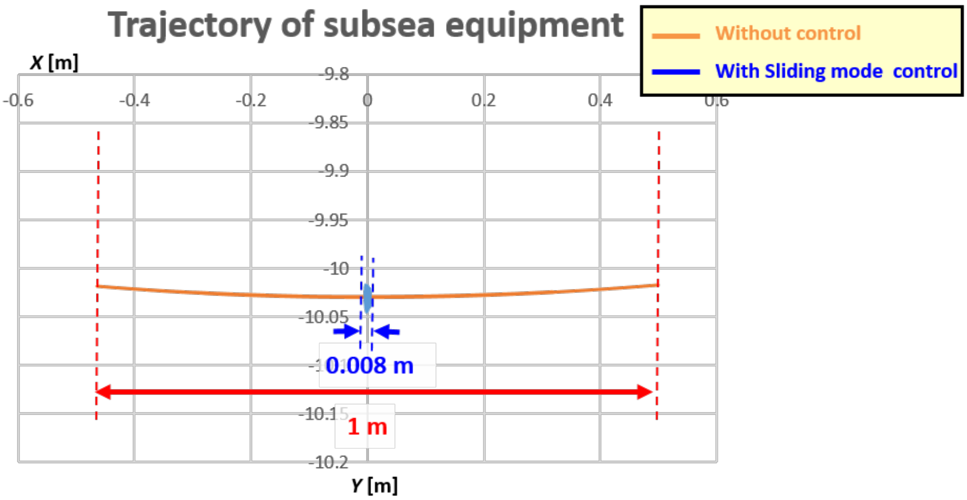


Figure 5.29: Trajectory of the subsea equipment in Case 1.

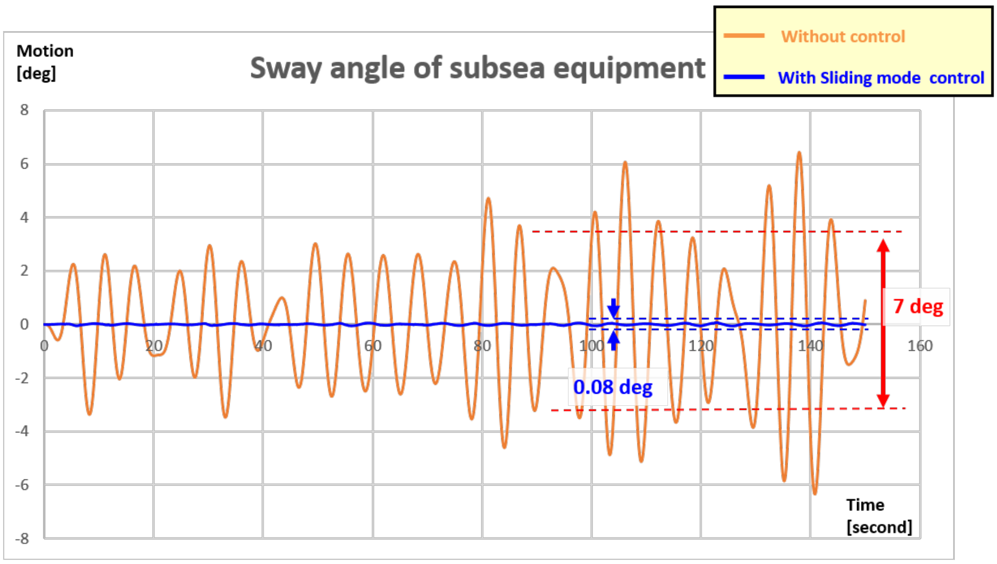


Figure 5.30: Graphs of sway angle of the subsea equipment in Case 2.

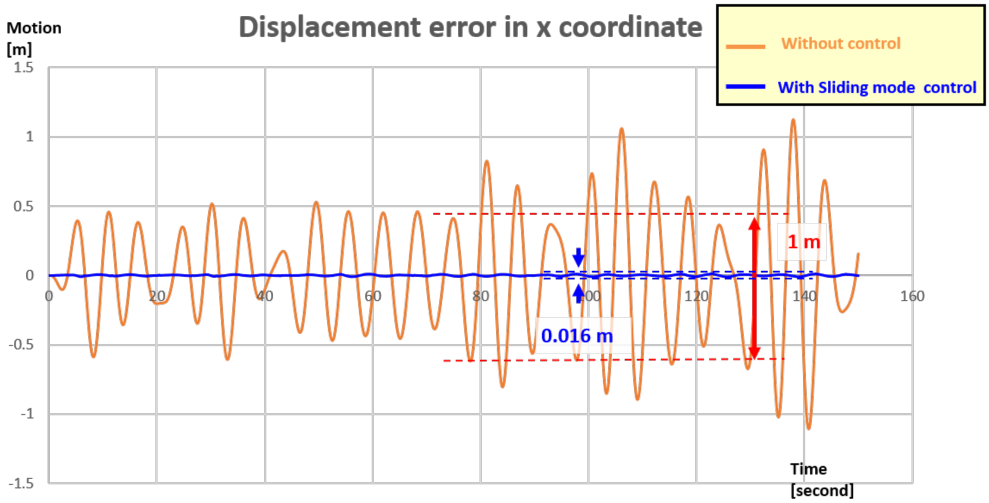


Figure 5.31: Graphs of displace error in the x coordinate of the subsea equipment in Case 2.

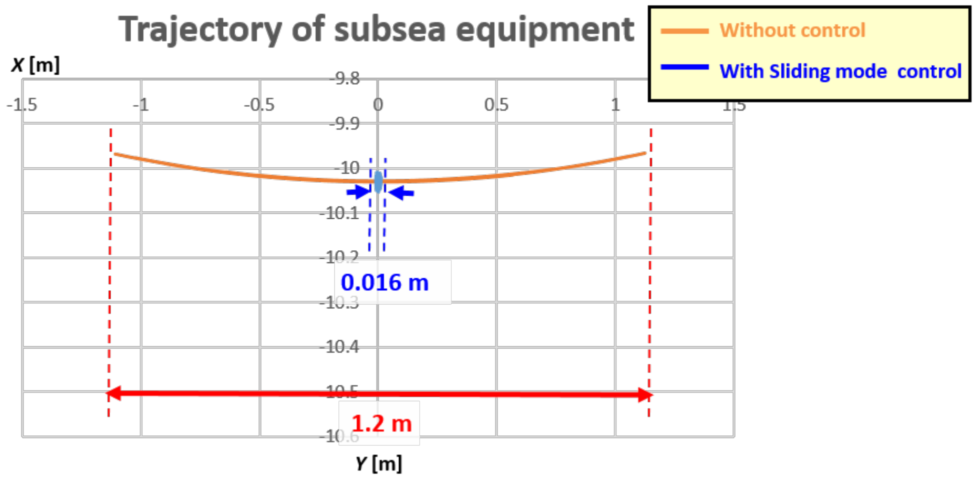


Figure 5.32: Trajectory of the subsea equipment in Case 2.

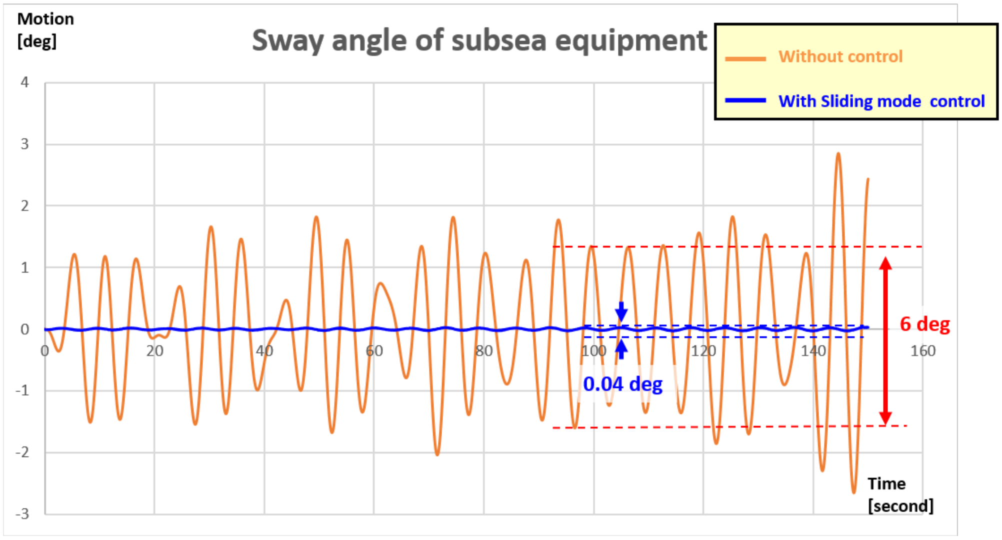


Figure 5.33: Graphs of sway angle of the subsea equipment in Case 3.

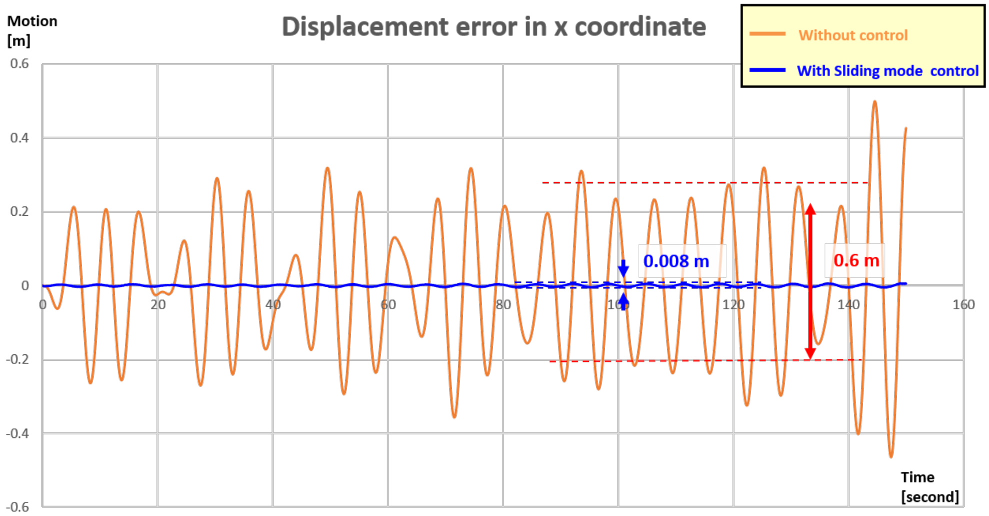


Figure 5.34: Graphs of displace error in the x coordinate of the subsea equipment in Case 3.

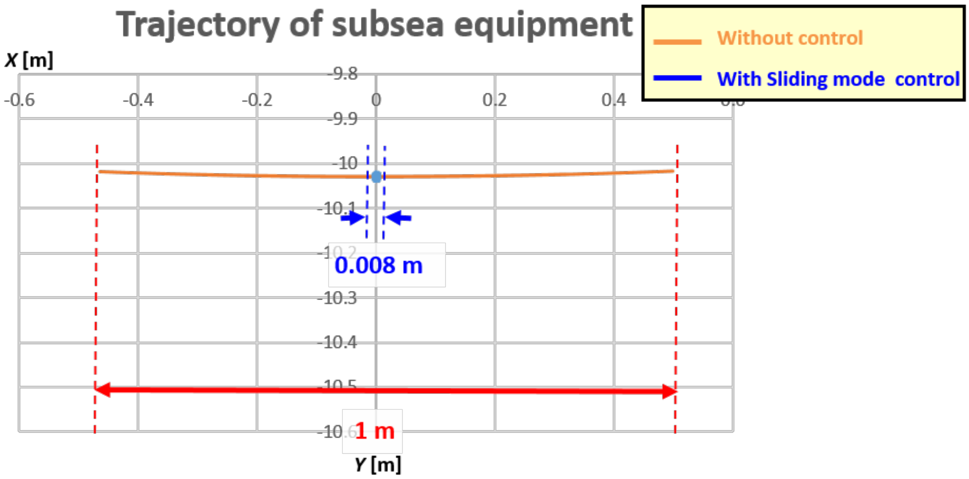


Figure 5.35: Trajectory of the subsea equipment in Case 3.

Chapter 6

Conclusions and Future Works

The Autonomous ship that operate in changing and stochastic environments must adapt to new situations and be equipped with fast decision making processes. In this study, we adopted model-free RL algorithm to solve the path following and collision avoidance problem which can compensate for the environmental disturbances. The autonomous ship is expected to follow the existing guidelines based on the International Regulations for Preventing Collisions at Sea (COLREGs), so that we present the COLREGs-based collision avoidance method, which can autonomously avoid encounter target ships followed by COLREGs. Moreover, we extend the trained RL framework to multiple ships, the simulation results show that all the ships have the ability to avoid each other and then back to the predefined paths.

When the autonomous ship arrive at a on-site construction, due to the environmental forces, the install operations will become difficult. We present the solution to AMC system to reduce the heave and sway motion. In addition, a HILS environment for AHC system is formulated, which can validate the proposed control system.

There are two main future lines of work which are currently being worked on. On the one hand, we would focus on improving the model-free RL algorithm we have adopted. This algorithms have been shown to be capable of learning a wide range of tasks, however, these suffer from very high sample complexity, often requiring millions of samples to achieve good performance. The high sample complexity of purely model-free RL algorithms has made them difficult to use for learning in the real world, where sample collection is limited by the constraints of real-time operation. In additional, model-based RL are known in general to outperform model-free RL regarding sample complexity, and in practice have been applied successfully to control robotic systems both in simulation and in the real world. As a result, we can combine the benefits of model-based and model-free RL by using the model-based agent to initialize a model-free agent. To build a robust, safety and efficient autonomous ships path following and collision avoidance system, I will focus on developing a hybrid algorithm which combines the sampling efficiency of model-based approach, with the high task-specific performance of model-free approach. I hope that the hybrid algorithm can accelerate model-free learning for sample-efficient and encourage future research in this area. Furthermore, we will implement the target tracking mission integrate with the target tracking filter.

On the other hand, a hardware experiment would be to validate the effectiveness of the proposed method for path following and collision avoidance system.

Bibliography

[1] Johnny Aarseth, Alf Helge Lien, Oyvind Bunes, Yingguang Chu, Vilmar Æsøy, Rolls-Royce Marine, and Alf Helge. A hardware-in-the-loop simulator for offshore machinery control system testing. In *ECMS*, pages 57–63, 2014.

[2] Eihab M Abdel-Rahman, Ali H Nayfeh, and Ziyad N Masoud. Dynamics and control of cranes: A review. *Modal Analysis*, 9(7):863–908, 2003.

[3] Akira Abe. Anti-sway control for overhead cranes using neural networks. *International Journal of Innovative Computing, Information and Control*, 7(7B):4251–4262, 2011.

[4] Morten Kollerup Bak, Michael R Hansen, and Hamid Reza Karimi. Robust tool point control for offshore knuckle boom crane. 2011.

[5] Marcel Bergerman and Yangsheng Xu. Robust joint and cartesian control of underactuated manipulators. *Journal of dynamic systems, measurement, and control*, 118(3):557–565, 1996.

[6] Morten Breivik and Thor I Fossen. Guidance-based path following for autonomous underwater vehicles. In *OCEANS, 2005. Proceedings of MTS/IEEE*, pages 2807–2814. IEEE, 2005.

[7] Ju-Hwan Cha, Myung-Il Roh, and Kyu-Yeul Lee. Dynamic response simu-

lation of a heavy cargo suspended by a floating crane based on multibody system dynamics. *Ocean Engineering*, 37(14-15):1273–1291, 2010.

[8] Linying Chen, Hans Hopman, and Rudy R Negenborn. Distributed model predictive control for vessel train formations of cooperative multi-vessel systems. *Transportation Research Part C: Emerging Technologies*, 92:101–118, 2018.

[9] Yin Cheng and Weidong Zhang. Concise deep reinforcement learning obstacle avoidance for underactuated unmanned marine vessels. *Neurocomputing*, 272:63–73, 2018.

[10] Hao-Tien Lewis Chiang and Lydia Tapia. Colreg-rrt: An rrt-based colregs-compliant motion planner for surface vehicle navigation. *IEEE Robotics and Automation Letters*, 3(3):2024–2031, 2018.

[11] Yingguang Chu, Vilmar Æsøy, Sören Ehlers, and Houxiang Zhang. Integrated multi-domain system modelling and simulation for offshore crane operations. *Ship Technology Research*, 62(1):36–46, 2015.

[12] UCG Commandant. International regulations for prevention of collisions at sea, 1972 (72 colregs). *US Department of Transportation, US Coast Guard, COMMANDANT INSTRUCTION M*, 16672, 1999.

[13] WE Cummins. The impulse response function and ship motions. Technical report, David Taylor Model Basin Washington DC, 1962.

[14] Sam Chau Duong, Eiho Uezato, Hiroshi Kinjo, and Tetsuhiko Yamamoto. A hybrid evolutionary algorithm for recurrent neural network control of a three-dimensional tower crane. *Automation in Construction*, 23:55–63, 2012.

[15] Damien Ernst, Mevludin Glavic, Florin Capitanescu, and Louis Wehenkel. Reinforcement learning versus model predictive control: a comparison on a power system problem. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2):517–529, 2009.

[16] Hosam K Fathy, Zoran S Filipi, Jonathan Hagena, and Jeffrey L Stein. Review of hardware-in-the-loop simulation and its prospects in the automotive area. In *Modeling and simulation for military applications*, volume 6228, page 62280E. International Society for Optics and Photonics, 2006.

[17] Thor I Fossen. *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011.

[18] Thor I Fossen and Anastasios M Lekkas. Direct and indirect adaptive integral line-of-sight path-following controllers for marine craft exposed to ocean currents. *International Journal of Adaptive Control and Signal Processing*, 31(4):445–463, 2017.

[19] Thor I Fossen, Morten Breivik, and Roger Skjetne. Line-of-sight path following of underactuated marine craft. *IFAC Proceedings Volumes*, 36 (21):211–216, 2003.

[20] Grant R Fowles and George L Cassiday. Analytical mechanics, thomson learning. *Inc*, 1999.

[21] Oddvar Gjelstenli. Anti-sway control and wave following system for offshore lattice crane. Master's thesis, Institutt for teknisk kybernetikk, 2012.

[22] Seung-Ho Ham, Myung-Il Roh, Hyewon Lee, and Sol Ha. Multibody dynamic analysis of a heavy load suspended by a floating crane with constraint-based wire rope. *Ocean Engineering*, 109:145–160, 2015.

[23] Jeong-Woo Hong, Myung-Il Roh, Seung-Ho Ham, and Sol Ha. Dynamic simulation of subsea equipment installation using an offshore support vessel based on flexible multibody system dynamics. *Journal of Marine Science and Technology*, 24(4):807–821, 2016.

[24] Qiong Hu, Shaojun Liu, et al. A hardware-in-the-loop simulation system of heave compensation of deepsea mining. In *Eighth ISOPE Ocean Mining Symposium*. International Society of Offshore and Polar Engineers, 2009.

[25] Rolf Isermann, Jochen Schaffnit, and Stefan Sinsel. Hardware-in-the-loop simulation for the design and testing of engine-control systems. *Control Engineering Practice*, 7(5):643–653, 1999.

[26] Dong-Hoon Jeong, Myung-Il Roh, and Seung-Ho Ham. Lifting simulation of an offshore supply vessel considering various operating conditions. *Advances in Mechanical Engineering*, 8(6):1687814016654633, 2016.

[27] Tor Arne Johansen, Tristan Perez, and Andrea Cristofaro. Ship collision avoidance and colregs compliance using simulation-based control behavior selection with predictive hazard assessment. *IEEE transactions on intelligent transportation systems*, 17(12):3407–3422, 2016.

[28] JMJ Journée and WW Massie. Offshore hydromechanics. 2001.

[29] Vishnu Kumar Kaliappan, Agus Budiyono, Dugki Min, K Muljowidodo, et al. Hardware-in-the-loop simulation platform for the design, testing and validation of autonomous control system for unmanned underwater vehicle. 2012.

[30] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[31] William Koch, Renato Mancuso, Richard West, and Azer Bestavros. Reinforcement learning for uav attitude control. *arXiv preprint arXiv:1804.04154*, 2018.

[32] Tuan A Le, Viet-Hung Dang, Deok H Ko, Tran N An, and Soon-Geul Lee. Nonlinear controls of a rotating tower crane in conjunction with trolley motion. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 227(5):451–460, 2013.

[33] Anastasios M Lekkas. Guidance and path-planning systems for autonomous vehicles. 2014.

[34] Adrian Jose Lew. *Variational time integrators in computational solid mechanics*. PhD thesis, California Institute of Technology, 2003.

[35] Øivind Aleksander G Loe. Collision avoidance for unmanned surface vehicles. Master's thesis, Institutt for teknisk kybernetikk, 2008.

[36] K Niki Maleki, Kaveh Ashenayi, Loyd R Hook, Justin G Fuller, and Nathan Hutchins. A reliable system design for nondeterministic adaptive controllers in small uav autopilots. In *Digital Avionics Systems Conference (DASC), 2016 IEEE/AIAA 35th*, pages 1–5. IEEE, 2016.

[37] Jerrold E Marsden and Matthew West. Discrete mechanics and variational integrators. *Acta Numerica*, 10:357–514, 2001.

[38] Sanin Muraspahic, Peter Gu, Lawk Farji, Yousef Iskandarani, Peng Shi, and Hamid Reza Karimi. Hardware-in-the-loop implementation for an active heave compensated drawworks. *Central European Journal of Engineering*, 2(2):201–211, 2012.

[39] So-Ryeok Oh and Jing Sun. Path following of underactuated marine surface vessels using line-of-sight based model predictive control. *Ocean Engineering*, 37(2-3):289–295, 2010.

[40] Tom Pedersen, Øyvind Smogeli, et al. Experience from hardware-in-the-loop testing of drilling control systems. *SPE Drilling & Completion*, 28 (04):308–315, 2013.

[41] Tristan Pérez and Mogens Blanke. *Mathematical ship modelling for control applications*. Ørsted-DTU, Automation, 2002.

[42] Jan Peters and Stefan Schaal. Policy gradient methods for robotics. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 2219–2225. IEEE, 2006.

[43] Liyana Ramli, Z Mohamed, Auwalu M Abdullahi, HI Jaafar, and Izzuddin M Lazim. Control strategies for crane systems: A comprehensive review. *Mechanical Systems and Signal Processing*, 95:1–23, 2017.

[44] Anita M Rothblum. Keys to successful incident inquiry. In *Human Factors in Incident Investigation and Analysis, 2nd International Workshop on Human Factors in Offshore Operations (HFW2002), Houston, TX*, 2002.

[45] Martin Schlager, Wilfried Elmenreich, and Ingomar Wenzel. Interface design for hardware-in-the-loop simulation. In *Industrial Electronics, 2006 IEEE International Symposium on*, volume 2, pages 1554–1559. IEEE, 2006.

[46] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

[47] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[48] Ahmed A Shabana. *Computational dynamics*. John Wiley & Sons, 2009.

[49] Thomas Stenersen. Guidance system for autonomous surface vehicles. Master's thesis, NTNU, 2015.

[50] Richard S Sutton, Andrew G Barto, Francis Bach, et al. *Reinforcement learning: An introduction*. MIT press, 1998.

[51] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

[52] Vadim Utkin. Variable structure systems with sliding modes. *IEEE Transactions on Automatic control*, 22(2):212–222, 1977.

[53] Vadim Utkin, Jürgen Guldner, and Jingxin Shi. *Sliding mode control in electro-mechanical systems*. CRC press, 2009.

[54] Det Norske Veritas. Wave analysis by diffraction and morison theory. *SESAM User Manual Wadam*, 2002.

[55] Det Norske Veritas. Dnv-rp-h103 modelling and analysis of marine operations. *Det Norske Veritas*, 2011.

[56] Tzu-Sung Wu, Mansour Karkoub, Wen-Shyong Yu, Chien-Ting Chen, Ming-Guo Her, and Kuan-Wei Wu. Anti-sway tracking control of tower cranes with delayed uncertainty using a robust adaptive fuzzy control. *Fuzzy Sets and Systems*, 290:118–137, 2016.

[57] Wei Min Xu, Bo Liu, Jian Xin Chu, and Xian Wen Zhou. An anti-swing and positioning controller for overhead cranes based on multi-sliding mode method. In *Advanced Materials Research*, volume 468, pages 328–334. Trans Tech Publ, 2012.

[58] Luman Zhao, Myung-Il Roh, Seung-Ho Ham, et al. Anti-sway control of the crane on an offshore support vessel based on the hardware-in-the-loop simulation. In *The 27th International Ocean and Polar Engineering Conference*. International Society of Offshore and Polar Engineers, 2017.

[59] Luman Zhao, Myung-Il Roh, and Seung-Ho Ham. Performance analysis of an active heave compensation system on an offshore supply vessel using the hardware-in-the-loop simulation. *Journal of Marine Science and Technology*, 26(5):678–691, 2018.

[60] Yuxin Zhao, Wang Li, and Peng Shi. A real-time collision avoidance learning system for unmanned surface vessels. *Neurocomputing*, 182:255–266, 2016.

Appendix A

Appendix

This section provides a brief overview of the main operational requirements from COLREGs Commandant (1999), relevant of our purpose:

- **Rule 6**: Safe speed. The following should be considered: Visibility, traffic density, stopping distance and turning ability, wind/waves/current, navigational hazards, draught vs. depth, radar/sensor state.

- **Rule 8**: Actions to avoid collision. Actions shall be made in ample time. If there is sufficient sea-room, alteration of course alone may be most effective. Safe distance required. Reduce speed, stop or reverse if necessary. Action by the ship is required if there is risk of collision, also when the ship has right-of-way.

- **Rule 13**: Overtaking. Any vessel overtaking any other shall keep out of the way of the vessel being overtaken. 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.

- **Rule 14**: Head-on situation. When two power-driven vessels are meeting on nearly reciprocal courses so as to involve risk for collision, then alter course to starboard so that each pass on the port side of each other.

- **Rule 14**: 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.
- **Rule 16**: Actions by give-way vessel. Take early and substantial action to keep well clear.
- **Rule 17**: Actions by stand-on vessel. Keep course and speed (be predictable) if possible. If it is necessary to take action, then the ship should try to avoid to alter course to port for a vessel on her own port side.
- **Rule 18**: Responsibilities between vessels. Except for Rules 9, 10, and 13, a power-driven vessel shall keep out of the way of: a vessel not under command, a vessel restricted in her ability to manoeuvre, a vessel engaged in fishing, and a sailing vessel.
- **Rule 19**: Conduct of vessels in restricted visibility. Avoid alteration of course to port for a vessel forward of the beam, and avoid alteration of course towards a vessel abeam or abaft the beam, if possible.

국문초록

해양 작업 지원선 (Offshore Support Vessel: OSV)의 경우 극한의 환경에도 불구하고 출항하여 해상에서 작업을 수행해야 하는 경우가 있다. 이러한 위험에의 노출을 최소화하기 위해 자율 운항에 대한 요구가 증가하고 있다. 여기서의 자율 운항은 선박이 출발지에서 목적지까지 사람의 도움 없이 이동함을 의미한다. 자율 운항 방법은 경로 추종 방법과 충돌 회피 방법을 포함한다. 우선, 운항 및 작업 중 환경 하중 (바람, 파도, 조류 등)에 대한 고려를 해야 하고, 국제 해상 충돌 예방 규칙 (Convention of the International Regulations for Preventing Collisions at Sea, COLREGs)에 의한 선박간의 항법 규정을 고려하여 충돌 회피 규칙을 준수해야 한다. 특히 연근해의 복잡한 해역에서는 많은 선박을 자동으로 회피할 필요가 있다. 기존의 해석적인 방법을 사용하기 위해서는 선박들에 대한 정확한 시스템 모델링이 되어야 하며, 그 과정에서 경험 (experience)에 의존하는 파라미터 튜닝이 필수적이다. 또한, 회피해야 할 선박 수가 많아질 경우 시스템 모델이 커지게 되고 계산 양과 계산 시간이 늘어나 실시간 적용이 어렵다는 단점이 있다. 또한, 경로 추종 및 충돌 회피를 포함하여 자율 운항 방법을 적용하기가 어렵다. 따라서 본 연구에서는 강화 학습 (Reinforcement Learning: RL) 기법을 이용하여 기존 해석적인 방법의 문제점을 극복할 수 있는 방법을 제안하였다. 경로를 추종하는 선박 (agent)은 외부 환경 (environment)과 상호작용하면서 학습을 진행한다. State S_0 (선박의 움직임과 관련된 각종 상태) 가지는 agent는 policy (현재 위치에서 어떤 움직임을 선택할 것인가)에 따라 action A_0 (움직일 방향) 취한다. 이에 environment는 agent의 다음 state S_1 을 계산하고, 그에 따른 보상 R_0 (해당 움직임의 적합성)을 결정하여 agent에게 전달한다. 이러한 작업을 반복하면서 보상이 최대가 되도록 policy를 학습하게 된다.

한편, 해상에서 크레인을 이용한 장비의 이동이나 설치 작업 시 위험을 줄이기

위해 크레인의 거동 제어에 대한 요구가 증가하고 있다. 특히 해상에서는 선박의 운동에 의해 크레인에 매달린 물체가 상하 동요 (heave)와 크레인을 기준으로 좌우 동요 (sway)가 발생하는데, 이러한 운동은 작업을 지연시키고, 정확한 위치에 물체를 놓지 못하게 하며, 자칫 주변 구조물과의 충돌을 야기할 수 있다. 이와 같은 동요를 최소화하는 Anti-Motion Control (AMC) 시스템은 Anti-Heave Control (AHC)과 Anti-Sway Control (ASC)을 포함한다. 본 연구에서는 해양 작업 지원선에 적합한 AMC 시스템의 설계 및 검증 방법을 연구하였다. 먼저 상하 동요를 최소화하기 위해 크레인의 와이어 길이를 능동적으로 조정하는 AHC 시스템을 설계하였다. 또한, 기존의 제어 시스템의 검증 방법은 실제 선박이나 해양 구조물에 해당 제어 시스템을 직접 설치하기 전에는 그 성능을 테스트하기가 힘들었다. 이를 해결하기 위해 본 연구에서는 Hardware-In-the-Loop Simulation (HILS) 기법을 활용하여 AHC 시스템의 검증 방법을 연구하였다. 또한, ASC 시스템을 설계할 때 제어 대상이 under-actuated 시스템이기 때문에 제어하기가 매우 어렵다. 따라서 본 연구에서는 sliding mode control 알고리즘을 이용하며 다관절 크레인 (knuckle boom crane)의 관절 (joint) 각도를 제어하여 좌우 동요를 줄일 수 있는 ASC 시스템을 설계하였다.

주요어: 해양 작업 지원선; 자율 운항; 충돌 회피; 경로 추종; 강화 학습; 다물체 시스템; Hardware-in-the-Loop Simulation (HILS); 상하 동요 저감 시스템; 좌우 동요 저감 시스템; 슬라이딩 모드 제어

학번: 2014-31430