博 士 学 位 論 文

Interactive Real-Time Modeling of Coupled Dynamics of Crane Ship and Suspended Load with AI-Driven Anti-Sway Control

近畿大学大学院
総合理工学研究科 東大阪モノづくり専攻
OLEG MAKAROV

博 士 学 位 論 文

クレーン船のインタラクティブな実時間連成動力学モデリングと AI を用いた吊り荷の振れ止め制御

2024 年 2 月 9 日

近畿大学大学院
総合理工学研究科 東大阪モノづくり専攻
OLEG MAKAROV

Doctoral Thesis

Interactive Real-Time Modeling of Coupled Dynamics of Crane Ship and Suspended Load with AI-Driven Anti-Sway Control

9 February 2024

Kindai University

Graduate School of Science and Engineering

Major in Innovative Engineering

OLEG MAKAROV

SUMMARY

Crane ships, or floating cranes are a special type of support vessels that perform offshore construction operations. They are a complex mechanical system and are subject to many environmental impacts. The dynamic response of the vessel to these impacts is highly non-linear and causes swaying of the cargoes it carries. Obviously, unpredictable cargo motions are a dangerous factor. Therefore, predicting the coupled dynamics of a crane ship and suspended cargo is one of the important research objectives in marine engineering. In line with the digitalization of the entire world industry, methods for analyzing crane ship dynamics are also numerical nowadays.

In this regard, this thesis is focused on the development of a numerical modeling method for the coupled dynamics of a floating crane with a suspended load in the presence of the marine environment, which is the source of the main forces acting on the crane vessel. Moreover, among the objectives of this research is not only the development of the modeling method itself. To apply it in real engineering practice, such a method must meet the requirements of reliability, computation speed and should not idealize the simulated computational cases. Therefore, an additional challenge is to make it possible to interactively control the numerical model as an operator controls a crane ship in reality. Finally, the last task of the thesis aims to test a method for swing compensation and positioning of suspended cargo. Instead of the traditional approach and linear control theory, artificial intelligence control (AI-control) has been applied, which is a promising technology for highly nonlinear systems such as a crane ship.

This thesis is divided into the following sections: Introduction, Review of existing research, Development of the numerical modeling method for multibody “vessel-cargo” system, Interactive modeling of crane vessel, AI-driven anti-sway control and positioning of suspended load, Conclusion.

The Introduction describes the relevance of the current research, the current state of the problem, the goals and objectives of the thesis, the theoretical and practical significance, the methods used and the approbation of the results.

Section 1 focuses on reviewing the research of other researchers on this topic and identifying the current state of the problem. The review is performed with respect to three areas of marine engineering science that are most essential for the methodology to be developed in this thesis: research in modeling the dynamics of floating bodies and crane ships, methods for determining hydrodynamic forces on floating objects, and modeling of cargo lifting and control of suspended loads motions.

Section 2 is focused on addressing the first task in this dissertation – developing a reliable and computationally efficient method for modeling the dynamics of a floating crane and suspended load in real-time. The section begins with a general formulation of the problem and the assumptions made in the research. Subsequently, experiments conducted in the wave basin by the team of Naruo

Technical Research Institute to determine the wave-induced motions of the crane ship model are described. The tested model is a scaled-down version of a real crane ship at a 1:50 scale. During the experiments, the displacements of the crane ship were tracked in six degrees of freedom, and the displacements and oscillations of the suspended load were also recorded.

In the concluding subsection 2.3, the developed numerical modeling method is described. Initially, the method is tested on a generalized problem and then applied to simulate the previously described laboratory experiments in the wave basin. The algorithm for constructing the numerical model using MATLAB, WEC-Sim, and Capytaine is outlined, with the former being a commercial software. The final part of this section involves an assessment of the parametric sensitivity of the modeling methodology, taking into account linear and nonlinear terms of the equation of motion. As a result, the developed method demonstrated high convergence with experiments and computation speed exceeding real-time.

Section 3 focuses on expanding the developed numerical modeling method by implementing interactive control of various components of the crane ship. Interactive control is achieved by connecting manipulators to a computer, with signals being read by MATLAB tools. The hoisting system was initially designed using the built-in tools of Simscape, but it was later enhanced with a custom-developed cable block using the Simscape programming language. This block allows not only modeling the tensioned and loosened states of the cable but also reads the signals from hardware for controlling the lifting velocity of the suspended load. As a result, the numerical model enables interactive control of the crane's base, crane boom, and the velocity of lifting and lowering the cargo. Additionally, it ensures accurate modeling of the ship's hydrodynamics through nonlinear calculation of wave forces.

Section 4 summarizes the results of the development of a swing control and cargo positioning system realized as an artificial intelligence. Its application was possible because of the results achieved in the past chapters – fast and accurate computations, as well as the ability to read control signals in real time. The section presents the methodology of training AI using reinforcement machine learning, describes the structure of neural networks of policy approximators and the results of training the RL Agent – the component responsible for controlling the crane boom and cable. As a result, the SAC algorithm showed high performance in positioning and balancing a suspended load in randomly generated target positions.

The Conclusion summarizes the research findings, describes the limitations of the developed methodology, and outlines directions for future research. The following directions of further research are focused on the improvement of the developed method: conducting additional experiments in the wave basin with irregular and wave-spreading conditions, improving the methodology for modeling cables and contacts, exploring more optimal settings for AI training, and expanding the method by modeling the stage of crossing the free water surface by a suspended load.

List of published articles

No.	Manuscripts	Authors	Journal (year, volume, pages)	Dissertation Chapter
1	Parametric sensitivity of crane ship numerical model with experimental verification in a wave basin	O. Makarov T. Harada	Ocean Engineering, 2023, 286, 115554.	Chapter 2 Chapter 3 (3.2)
2	Implementation of interactive control of a crane ship model in MATLAB/Simulink	O. Makarov H. Yase T. Harada	ROBOMECH Journal, 2023, 10, 19.	Chapter 3

Declaration

I hereby declare that this thesis entitled “Interactive real-time modeling of coupled dynamics of crane ship and suspended load with ai-driven anti-swing control” was written solely by me, and it has not been submitted, in whole or in part, to any university for any degree previously.

I am aware of and understand the university's policy regarding plagiarism, and I confirm that this dissertation is my own work, except were indicated in the references.

Oleg Makarov

9 February 2024

Preface

This dissertation is my original work conducted during my doctoral studies at the Graduate School of Science and Engineering, Kindai University, Osaka, Japan. Pursuing a doctoral degree has been my goal since the beginning of my master's studies at the Far Eastern Federal University, Vladivostok, Russia, and I have finally achieved it. My journey has been challenging, as my studies at Kindai began remotely due to the COVID-19 pandemic, continued on-site at Kindai University, but were complicated by changes in the global political situation. Additionally, alongside my studies at Kindai, I continued my Ph.D. studies at the Far Eastern Federal University, dedicating all my time to research and science. I faced numerous challenges, and I am grateful to any reader of this thesis for their interest in my research.

My research was conducted in collaboration with the Naruo Technical Research Institute (TOYO CONSTRUCTION CO., LTD), Kobe, Japan, aiming to the problem of ensuring the safety of offshore operations conducted by crane vessels during the construction of marine structures. As human activities increasingly extend into the maritime environment and exploit its technical capabilities, unfavorable natural conditions often complicate construction processes. The impact of strong winds, significant waves, and currents complicates the operation of crane ships, thereby increasing construction timelines and costs.

To predict the dynamics of crane ships in such complex conditions, the use of modern modeling methods is necessary, and the development of these methods is the focus of this thesis. In summary, the results can be generalized into three achievements: the development of a reliable and accessible real-time modeling method for crane ship dynamics, the implementation of interactive control of the numerical model, and the integration of artificial intelligence as a swing control and load positioning system for the crane ship. The outcomes of my work can serve as an example in the development of various modern smart solutions for crane ships and in shipbuilding as a whole.

This dissertation is the end point of my difficult but exciting life in Japan. I am very grateful to all those people who supported me during this challenging period. The experiences and memories gained throughout my studies at Kindai University I will remember as one of the most important and memorable chapters in my life.

Acknowledgement

I express my sincere gratitude, first and foremost, to my academic supervisor, Professor Takashi Harada, for his invaluable support throughout my studies at Kindai University. He not only guided me with his knowledge, ideas, and experience but also provided invaluable help to make me feel like at home in a foreign country. He showed understanding towards my academic and life difficulties, and I felt his support throughout my challenging journey. I thank him for his generosity, kindness, as well as his contribution to the development of my personality as a researcher.

I would also like to thank Dr. Hayato Yase for his interest in my research topic and his friendly warm attitude toward me. He contributed to my work by assisting in the hardware setup of the interactive control system, and I appreciate his willingness to offer support with his knowledge and experience. I am sincerely glad to meet Dr. Yase and wish him interesting projects and success in his research.

I extend my gratitude to the Naruo Research Institute team for their technical assistance and the provision of valuable experimental data that forms on which part of my research is based. Additionally, their scholarship played a pivotal role, providing the opportunity to study in Japan at Kindai University without concerns about tuition and living expenses. I am immensely thankful for this support.

I express my gratitude to my academic supervisor at the Far Eastern Federal University, Alexander Becker, for his persistent assistance and support, despite my absence in Russia during my two years of research at Kindai University. He continued to support me throughout my studies in Japan, providing knowledge and expertise not only on the subject of my Russian thesis but also on the topic of current research.

Finally, I express my immeasurable gratitude to my loved ones, my mother, and grandmother, for my upbringing and support. Their life guidance instilled in me the qualities that led me to this current moment. I conclude this thesis with profound thanks to the endless support of my friend Anastasia Tikhonova and my entire family, who believed in me and provided the strength to move forward despite the difficulties.

Contents

Introduction

Relevance of the topic

Maritime operations are an integral part of the process of constructing marine hydraulic structures. A significant portion of the offshore construction process is carried out using offshore support vessels (OSVs) or crane ships capable of transporting and installing heavy structural components of constructions, such as molded blocks and caissons. The operation of a crane ship during construction is highly dependent on environmental conditions, such as wave parameters, wind speed, and current velocity. The cumulative impact of natural forces on a floating crane leads to complex and unpredictable movements of the entire vessel, an increase in the amplitudes of suspended load oscillations, and consequently, a reduction in installation accuracy and overall process safety. In conditions of excessive wave height or wind speed, heavy lifting operations may become entirely impossible. Hence, strict limits on acceptable values of natural impact parameters exist to ensure the safe execution of maritime operations. In the presence of adverse conditions, there is a forced downtime of marine equipment, leading to increased construction costs. Since predicting weather conditions for the entire construction period is impossible, significant time margins for weather conditions are incorporated into the construction projects. These margins take into account the limited "weather window" for maritime operations and increase the whole period of construction of structures from the sea.

The increasing pace of offshore construction is increasingly raising the issue of improving the accuracy and safety of lifting operations carried out by crane ships. This brings new engineering challenges and increases the need for research in this area. In this regard, researchers turn their attention to the problems of analyzing the dynamic response of crane ships to external natural impacts during the process of equipment installation and construction of structures. Although the problem itself is not new, more and more effective methods of its solution are appearing, which allow developing new technologies for implementation in real engineering practice. The modern approach to solving the set tasks in the field of offshore construction is the use of mathematical modeling and numerical analysis tools.

The development of numerical simulation techniques (computational mathematics and mechanics) of physical processes and phenomena has led to the creation of many specialized software. These software programs use mathematical modeling techniques to obtain an understanding of the behavior of the sea-ship-cargo system in the frequency and time domains, which in general are the methods of multibody dynamics (MBD) and computational fluid dynamics (CFD). The range of available numerical methods and techniques itself is quite extensive. Researchers have access to both fairly accurate modeling methods, which are time and computationally intensive, and simpler and

faster ways to compute the dynamics of a floating body. The applicability of each method should be evaluated depending on the task at hand.

The task of developing a method for numerical modeling of crane ship dynamics, which should be both fast and have an appropriate calculation accuracy, is in great demand today. The optimal balance of speed and accuracy is in demand for the design of digital twins of the vessels, automated control systems and other smart solutions and applications aimed at robotization of the offshore construction industry. This will help to expand the range of safe environmental parameters at which offshore construction operations are permissible. In modeling, it is important to consider not only the cargo handling stage, but also the lift-off and installation stages, which require consideration of contact interactions and changing environmental conditions. The impact of several ambient media (sea and air), as well as the influence of suspended load swing on the hydrodynamics of the vessel, puts the problem of analyzing offshore cranes at a level above their land-based counterparts. The methodology of numerical modeling of crane ship dynamics should also be widely available and should not be highly dependent on the use of various high-cost software, which is a common problem in interdisciplinary research.

Computational speed and accuracy are critical points in solving another problem addressed in this thesis. It consists in developing a method of interactive modeling of the dynamics of a floating crane, in which manual control of the elements of the numerical model during simulation becomes possible. The implementation of such a technique will significantly increase the validity of the numerical experiments performed compared to the idealized scenarios used today. Interactivity and the user (operator) experience opens the way to the development of other intelligent systems in the field of crane ship applications.

Hence, various systems for compensating suspended cargo swing are being created and tested to improve the safety of offshore construction operations. This problem is also considered in this work, but its solution for the present time is not conventional, and consists in the application of machine learning techniques. This area is relatively new, but potentially has significant advantages over existing control theory methods. Machine learning-based control can handle highly nonlinear phenomena and take into account a large space of possible system states. Moreover, unlike typical swing compensation systems, AI control can not only minimize the swing of the load, but also automatically position it at a user-specified location.

Thus, the relevance of this thesis is explained by the existing necessity of the development of a methodology for fast and sufficiently accurate numerical analysis of the dynamic response of a crane ship with a suspended load, taking into account the mutual influence of their motions, capable of processing user-defined scenarios in real time and implementing technologies for minimizing the swing and positioning with the help of intelligent systems.

Current state of research.

Floating cranes primarily fall within the realm of the study of the dynamics of floating bodies. Significant contributions to this field have been made by researchers such as Cummins W. E., Newman J. N., Faltinsen O. M., Fossen T. I. While hydrodynamic methods based on the Navier-Stokes equations account for all flow nonlinearities, in practical seakeeping modeling, the potential flow theory is often employed. This theory is based on the Bernoulli equation and linear wave theory.

At the same time, crane ships are technical systems with moving crane components, and thus, they are additionally explored by researchers in areas like three-dimensional mechanics, rigid body dynamics, and structural mechanics. In recent decades, numerical analysis methods have taken a predominant position in the study of dynamics, driven by the development of digital technologies and the increased computing power. Several in-house and commercial software solutions have been developed in this direction. Researchers like Cha J.-H., Roh M.-I., Lee K.-Y., Ku N, Ha S., Jeong D.-H., Ham S.-H., Chen H. have developed their own numerical models of crane ships, enabling the modeling of hydrodynamics and maritime operations. Notable potential flow solvers are NEMOH, HAMS, Capytaine, Ansys AQWA, Hydrostar, Tdyn SeaFem, and WAMIT.

Regarding the control of cargo sway, comprehensive research has been conducted. Works by such authors as Sun N., Sun M., Fang Y., Chen H., and Æsøy F. have delved into standard control theory. In addition, there is active research in a relatively new direction – machine learning and artificial intelligence. Contributions in this area have been made by Ramli L., Qian Y., Kim G.-H., and their colleagues.

Taking into account the scientific world experience of various authors, it can be said that studies of the coupled dynamics of crane ships and payloads are carried out with respect to many interdisciplinary tasks and in modern times are mainly aimed at solving two main problems: ensuring the accuracy and functionality of numerical models of floating cranes, as well as the search for effective methods, schemes and systems of suspended cargo control to improve the safety of marine operations.

Goals and objectives.

The goal of this thesis is to develop a method of multiphysics interactive numerical simulation of coupled dynamics of the “vessel-cargo” system with anti-sway control of the payload using artificial intelligence. In order to achieve the set goal, the following tasks should be solved:

- to study the current state of research and "state of art" research in the field of modeling of floating bodies, mechanical systems and crane vessels with suspended load;
- select and apply a numerical method for modeling the hydrodynamics of a floating crane and calculating hydrodynamic forces;
- to develop an algorithm for constructing a numerical model of a crane ship and modeling of its motions taking into account hydrodynamic forces and effects;

- to conduct numerical experiments of the developed model in order to evaluate the reliability and performance of the crane ship dynamics modeling method by comparing the results of numerical simulation with experimental data;
- to develop a mechanism of interactive control of the numerical model elements in real time to expand the practical applicability of the developed method;
- investigate and apply the method of reducing the sway of payload on floating crane using artificial intelligence.

The object of the study is the coupled motion of a suspended load and a crane vessel.

The subject of the study is the methodology of numerical modeling of the coupled dynamics of a suspended load and a crane vessel.

Theoretical and practical importance.

The theoretical significance of the study consists in evaluating the capabilities of numerical methods used to solve the problem at hand, identification of their advantages and disadvantages, as well as ways to enhance the methodology of numerical modeling. The study is interdisciplinary and has theoretical value in terms of combining the fields of mechanics, hydrodynamics and advanced computer technologies.

The practical significance of the research lies in the development of a methodology for numerical modeling of the dynamics of a floating crane and suspended cargo. The methodology proves to be efficient in terms of calculation speed and accuracy, enabling real-time prediction of the vessel and suspended cargo movements. Additionally, the proposed algorithm for developing and utilizing the numerical model involves the combination of open-source CFD software and the commercially available MBD solver Simscape within the MATLAB/Simulink environment. Such a software set is characterized by relatively low cost and broad scalability in terms of simulated physics and mechanisms. The application of artificial intelligence for positioning and controlling anti-sway of cargo is in line with current trends in computer science. The testing of reinforcement learning technique is highly relevant to the development of smart crane ships and unmanned vessels.

Applied methods.

This thesis work is interdisciplinary with the application of methods and practices from different engineering disciplines. The primary numerical modeling methods for the dynamics of a floating crane with suspended cargo include computational fluid dynamics based on potential flow theory and three-dimensional mechanics (rigid-multibody dynamics). Within these numerical methods, different approaches to constructing the numerical model were used, such as geometric modeling, discretization (decomposition) of model components into boundary elements, and the composition of the overall numerical model in a single modeling environment. Additionally, an internal programming language in MATLAB/Simulink – Simscape, was applied for constructing individual elements of the model. The development of the cargo swing compensation mechanism was

implemented using reinforcement learning technique in the MATLAB programming language. The software used for calculations in this thesis is also interdisciplinary. Software such as NEMOH, HAMS, Capytaine, BEM Rosetta, ANSYS Fluent, MATLAB/Simulink, and WEC-Sim were used.

Among the theoretical methods of the study the following were used: review and analysis of existing research on the topic, deductive decomposition of the problem of numerical modeling of the dynamics of a floating crane, synthesis of the general methodology of numerical experiments, axiomatic method of creating initial numerical models, qualitative and mathematical comparison of the results. The final modeling methodology and numerical model have been verified through a comparison of the results with experimental data.

Contributions.

- methodology for building a numerical model of a floating crane in accessible and optionally upgradeable software;
- methodology for modeling the dynamic response of a floating crane with the capability of real-time interactive control of the crane elements;
- methodology for modeling the sway suspension and positioning system of suspended load on the basis of reinforcement machine learning technique;
- pointing out issues and bugs in the WEC-Sim software, an open-source code for modeling the dynamics of floating bodies, resulting in source code changes.

Approbation.

The main results of the study have been published in the form of scientific articles in the ROBOMECH Journal (Q2) and Ocean Engineering journal (Q1) and in the proceedings of the following conferences:

- 5th Jc-IFToMM International Symposium and the 28th Jc-IFToMM Symposium on Theory of Mechanism and Machines (Jc-IFToMM 2022, Kyoto, Japan);
- 9th International Conference on Manufacturing, Machine Design and Tribology (ICMDT 2023, Jeju, Korea);
- Robotic Symposia 2023 (Shirahama, Japan);
- IFToMM World Congress 2023 (Tokyo, Japan);
- The 10th International Conference on Electrical Engineering, Control and Robotics (EECR, Guangzhou, China).

Additionally, the work was presented in the form of a poster presentation at the Autumn Conference of The Japan Society of Naval Architects and Ocean Engineers (JASNAOE Autumn meeting) in 2022 (Kobe, Japan) and 2023 year (Nagasaki, Japan) and meetings with Naruo Technical Research Institute (TOYO CONSTRUCTION CO., LTD).

Thesis structure.

The thesis comprises 160 pages, 4 chapters, 124 figures and 12 tables.

1. Review on existing research

1.1. Dynamics of crane ships with suspended load

Crane vessels constitute a distinct class of maritime vehicles, serving as auxiliary vessels for marine loading/unloading and construction operations. Floating cranes are considered complex systems, comprising a floating hull, superstructure, the crane itself, and the carried loads. These vessels are subject to the influence of various natural forces, including waves, wind, and currents, adding complexity to their analysis compared to their land-based counterparts. Therefore, the dynamic analysis of the operation of such vehicles involves theoretical and practical methods from various disciplines, such as hydrodynamics, ship theory, multibody dynamics, mechanics, control theory, and more.

To understand the dynamics of both the vessel and the suspended load, modern research methods rely on numerical modeling, which has become the standard approach in solving problems in the field of maritime engineering. The increasing pace of nearshore and offshore construction, coupled with the active development of digital technologies and the growing computational power of computers, contributes to the development and implementation of new scientific and technical foundations and methodologies for numerical modeling of the dynamic response of floating cranes with suspended loads. In this modeling, it is crucial to consider not only the stage of load movement but also the stages of lift-off and installation, which require the incorporation of additional physics, such as contact interactions. Furthermore, the ability to simulate in real-time is essential for the application of such models in real engineering practice, for instance, in creating digital twins of vessels or interactive simulators.

First and foremost, crane vessels are floating technical systems. When analyzing the dynamic response of floating objects on a macro scale, deformations of their structural elements are typically not considered. Accordingly, the equation of their motion can be generalized in the form of Newton's second law:

$$\mathbf{M}\ddot{\mathbf{x}}(t) = \mathbf{F}_e(t), \tag{1.1}$$

where $\mathbf{M}$ is the mass matrix of the multibody system, $\ddot{\mathbf{x}}(t)$ is the acceleration vector, $\mathbf{F}_e(t)$ represents time-varying external forces.

The external forces$\mathbf{F}_e$ in this system are caused by various environmental impacs and physical phenomena, such as wave and wind actions, current effects, forces arising from wave diffraction, added mass, mooring loads, etc. By separating the components of external loads, in the absence of constraining equations, the equation of motion for the floating body can be expressed in the form of ordinary differential equations (ODE):

$$\mathbf{M}\ddot{\mathbf{x}}(t) = \mathbf{F}_{hs}(t) + \mathbf{F}_{exc}(t) + \mathbf{F}_{rad}(t) + \mathbf{F}_{v}(t) + \mathbf{F}_{ext}(t), \tag{1.2}$$

where $\mathbf{F}_{hs}(t)$ is hydrostatic restoring force vector; $\mathbf{F}_{exc}(t)$ is the wave excitation and diffraction force vector; $\mathbf{F}_{rad}(t)$ is the force vector resulting from wave radiation; $\mathbf{F}_{v}(t)$ is the viscous force vector; $\mathbf{F}_{ext}(t)$ is externally applied forces vector (including other terms of wave forces).

When analyzing the dynamic response of a floating body to external force actions, it is conventional to consider 6 degrees of freedom (DOF), as depicted in Fig. 1.1.

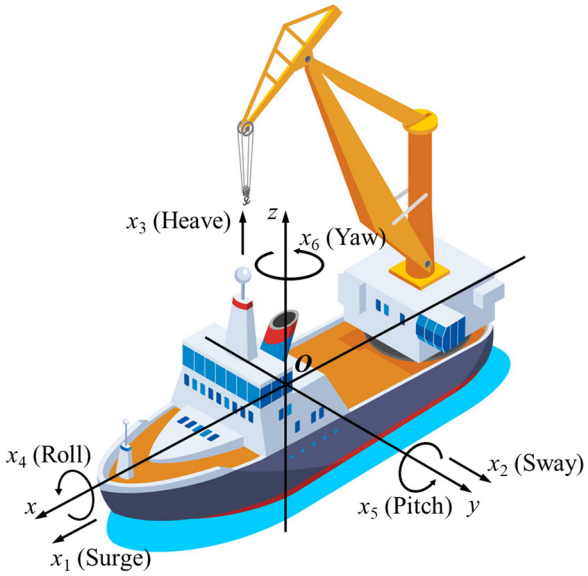


Fig. 1.1 Body motions in six degrees of freedom as in Newman J. N. [1, p. 298]

In this case, the mass matrix has dimension 6x6, the vector of accelerations and external forces are defined by 6 elements:

$$\begin{bmatrix} m & 0 & 0 & 0 & mz_G & -my_G \\ 0 & m & 0 & -mz_G & 0 & mx_G \\ 0 & 0 & m & my_G & -mx_G & 0 \\ 0 & -mz_G & my_G & I_{11} & I_{12} & I_{13} \\ mz_G & 0 & -mx_G & I_{21} & I_{22} & I_{23} \\ -my_G & mx_G & 0 & I_{31} & I_{32} & I_{33} \end{bmatrix} \begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \\ \ddot{x}_3 \\ \ddot{x}_4 \\ \ddot{x}_5 \\ \ddot{x}_6 \end{bmatrix} = \begin{bmatrix} F_1^{hs} + F_1^{exc} + F_1^{rad} + F_1^{v} + F_1^{ext} \\ F_2^{hs} + F_2^{exc} + F_2^{rad} + F_2^{v} + F_2^{ext} \\ F_3^{hs} + F_3^{exc} + F_3^{rad} + F_3^{v} + F_3^{ext} \\ F_4^{hs} + F_4^{exc} + F_4^{rad} + F_4^{v} + F_4^{ext} \\ F_5^{hs} + F_5^{exc} + F_5^{rad} + F_5^{v} + F_5^{ext} \\ F_6^{hs} + F_6^{exc} + F_6^{rad} + F_6^{v} + F_6^{ext} \end{bmatrix} \tag{1.3}$$

If the position of the center of gravity of the floating body coincides with the origin of the coordinate system, then $x_G = y_G = z_G = 0$.

The most well-known special form of equation (1.2) in solving problems of floating body dynamics is the Cummins equation [2], developed under the assumption of linear motions of the floating body. Expressing the pressures in the linear wave formulation on the ship hull as a function of the momentum potential in terms of the displacement, he derived the following integral-differential equation, which is still widely used for describing spatial movements of floating bodies in various seakeeping software:

$$\big(\mathbf{M}+\mathbf{A}(\omega)\big)\ddot{\mathbf{x}}(t)+\mathbf{K}_{hs}\mathbf{x}(t)+\big(\mathbf{C}_v+\mathbf{B}(\omega)\big)\dot{\mathbf{x}}(t)=\Re\left[\frac{h}{2}\mathbf{F}_{exc}(\omega,\theta)e^{i\omega t}\right]+\mathbf{F}_{ext}(t), \quad (1.4)$$

where $\mathbf{A}(\omega)$ is a frequency-dependent added mass matrix; $\mathbf{B}(\omega)$ is a frequency-dependent radiation damping matrix; $\mathbf{K}_{hs}$ is the matrix of hydrostatic stiffness coefficients of the floating body; $\mathbf{C}_v$ is the viscous damping matrix; $\boldsymbol{F}_{exc}(\omega,\theta)$ is the complex wave-excitation amplitude vector obtained from the frequency domain solution, h is a wave height.

For a particular frequency and wave direction, Eq. (1.3) can be expanded as follows:

$$\begin{bmatrix} m+A_{11} & A_{12} & A_{13} & A_{14} & A_{15} & A_{16} \\ A_{21} & m+A_{22} & A_{23} & A_{24} & A_{25} & A_{26} \\ A_{31} & A_{32} & m+A_{33} & A_{34} & A_{35} & A_{36} \\ A_{41} & A_{42} & A_{43} & I_{11}+A_{44} & I_{12}+A_{45} & I_{13}+A_{46} \\ A_{51} & A_{52} & A_{53} & I_{21}+A_{54} & I_{22}+A_{55} & I_{23}+A_{56} \\ A_{61} & A_{62} & A_{63} & I_{31}+A_{64} & I_{32}+A_{65} & I_{33}+A_{66} \end{bmatrix} \begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \\ \ddot{x}_3 \\ \ddot{x}_4 \\ \ddot{x}_5 \\ \ddot{x}_6 \end{bmatrix} +$$

$$+ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & K_{33} & K_{34} & K_{35} & 0 \\ 0 & 0 & K_{43} & K_{44} & K_{45} & K_{46} \\ 0 & 0 & K_{53} & K_{54} & K_{55} & K_{56} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} +$$

$$+ \begin{bmatrix} C_{v,11}+B_{11} & C_{v,12}+B_{12} & C_{v,13}+B_{13} & C_{v,14}+B_{14} & C_{v,15}+B_{15} & C_{v,16}+B_{16} \\ C_{v,21}+B_{21} & C_{v,22}+B_{22} & C_{v,23}+B_{23} & C_{v,24}+B_{24} & C_{v,25}+B_{25} & C_{v,26}+B_{26} \\ C_{v,31}+B_{31} & C_{v,32}+B_{32} & C_{v,33}+B_{33} & C_{v,34}+B_{34} & C_{v,35}+B_{35} & C_{v,36}+B_{36} \\ C_{v,41}+B_{41} & C_{v,42}+B_{42} & C_{v,43}+B_{43} & C_{v,44}+B_{44} & C_{v,45}+B_{45} & C_{v,46}+B_{46} \\ C_{v,51}+B_{51} & C_{v,52}+B_{52} & C_{v,53}+B_{53} & C_{v,54}+B_{54} & C_{v,55}+B_{55} & C_{v,56}+B_{56} \\ C_{v,61}+B_{61} & C_{v,62}+B_{62} & C_{v,63}+B_{63} & C_{v,64}+B_{64} & C_{v,65}+B_{65} & C_{v,66}+B_{66} \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \quad (1.5)$$

$$= \begin{bmatrix} \Re\left(\frac{h}{2}F_{exc,1}e^{i\omega t}\right) \\ \Re\left(\frac{h}{2}F_{exc,2}e^{i\omega t}\right) \\ \Re\left(\frac{h}{2}F_{exc,3}e^{i\omega t}\right) \\ \Re\left(\frac{h}{2}F_{exc,4}e^{i\omega t}\right) \\ \Re\left(\frac{h}{2}F_{exc,5}e^{i\omega t}\right) \\ \Re\left(\frac{h}{2}F_{exc,6}e^{i\omega t}\right) \end{bmatrix} + \begin{bmatrix} F_{ext,1} \\ F_{ext,2} \\ F_{ext,3} \\ F_{ext,4} \\ F_{ext,5} \\ F_{ext,6} \end{bmatrix}$$

Thus, the response of the floating body to wave actions from the marine environment is determined by its frequency ω, wave direction θ, and wave height h. The wave excitation forces, in turn, include two effects – Froude-Krylov force due to undistributed pressure field and diffraction force occurred because the floating structure changes this pressure field [3, p. 59].

The terms added mass and radiation damping represent steady-state hydrodynamic forces and moments due to forced harmonic rigid body motions, which generate outgoing waves [3, p. 41]. As seen from the above equation, there are a total of 36 coefficients of added mass and 36 coefficients of radiation damping. For the submerged part of the ship's hull with a vertical plane of symmetry parallel to the forward direction (which is true for conventional hulls and floating cranes), half of the coefficients are equal to zero [3, p. 42].

The restoring forces characterized by the hydrostatic stiffness matrix come from the hydrostatic pressure of water and displaced volume (mass).

The above formulation assumes that the flow of the fluid around a vessel is quasi-steady. Cummins, on the other hand, included a time history of motion in the radiation terms. This physical phenomenon is called the fluid memory effect and very important for proper dynamic time-domain analysis of floating bodies in irregular waves:

$$(\mathbf{M} + \mathbf{A}_\infty)\ddot{\mathbf{x}}(t) + \mathbf{K}_{hs}\mathbf{x}(t) + \int_0^t \mathbf{K}_{\boldsymbol{r}}(t-\tau)\dot{\mathbf{x}}(\tau)\mathrm{d}\tau + \mathbf{C}_v\dot{\mathbf{x}}(t) = \mathbf{F}_{exc}(t) + \mathbf{F}_{ext}(t), \qquad (1.6)$$

where $\mathbf{A}_\infty$ is added mass matrix at infinite frequency; $\mathbf{K}_{\boldsymbol{r}}(t-\tau)$ is a retardation function, which captures the fluid memory effect:

$$\mathbf{K}_r(t) = \frac{2}{\pi}\int_0^\infty \mathbf{B}(\omega)\cos(\omega t)d\omega, \qquad (1.7)$$

If the sea state is characterized by irregular waves, which are a composition of random waves of different height and frequency, it is usually described by a wave spectrum. In such a case, the wave excitation forces $\mathbf{F}_{exc}$ are determined from the linear superposition of the set of regular wave components:

$$\mathbf{F}_{exc}(t) = \Re\left[\sum_{j=1}^{N} \mathbf{F}_{exc}(\omega_j, \theta)e^{i(\omega_j t + \phi_j)}\sqrt{2S(\omega_j)d\omega_j}\right], \qquad (1.8)$$

where $S(\omega_j)$ is wave spectral density for wave frequency ω_j; ϕ_j is the randomized phase angle; N is the number of frequency bands selected to discretize the wave spectrum.

Accordingly, the calculation of floating bodies is performed not only in the time domain but also in the frequency domain. Irregular waves, which are a typical real case, result in a complex dynamic response of the floating body. For accurate calculations in real conditions, it is necessary to determine the hydrodynamic parameters of the submerged part for the entire range of wave

frequencies, which, according to the superposition principle, form the resulting irregular wave that directly affects the ship's hull.

The above hydrostatic and hydrodynamic forces are commonly considered in marine engineering, and the equation of motion is the de facto standard in analyzing the dynamic response of floating objects caused by wave action. However, when considering crane vessels specifically, the dynamics of such technical entities cannot be solely described by the motion of the floating part, as the movements of all structural elements are not entirely constrained.

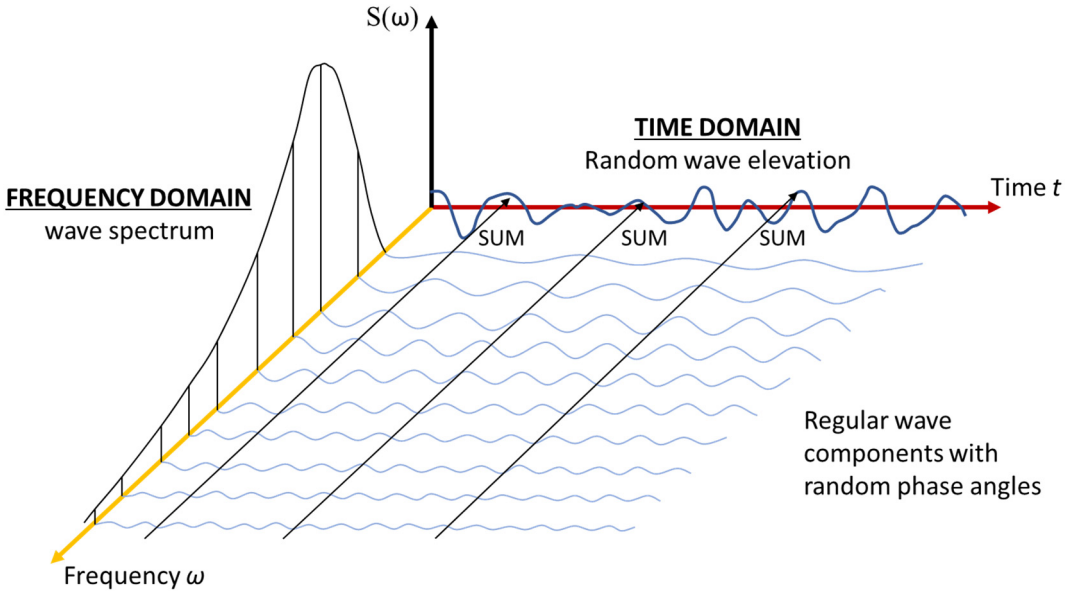


Fig. 1.2 The connection between a frequency domain and time domain representation of waves in a long-crested short term sea state according to Faltinsen [3]

The influence of the crane moving parts and the suspended load on the dynamics of the entire system must be considered. In a simple case, the suspended load can be represented as a pendulum (Fig. 1.3), the equation of motion of which includes an algebraic constraint C_x:

$$\begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix} \begin{bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{bmatrix} = \begin{bmatrix} mg - T\cos\theta \\ -T\sin\theta \end{bmatrix} \tag{1.9}$$

$$C_x = x_1^2 + x_2^2 - l^2 = 0 \tag{1.10}$$

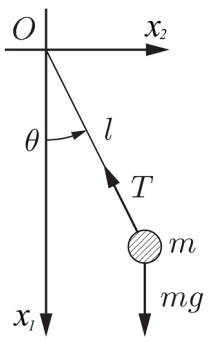


Fig. 1.3 Simple pendulum

Translating this example to a crane vessel, there can be multiple such constraints. Since the motion of the crane vessel is described by both algebraic equations and ordinary differential equations, the crane vessel as a system of rigid bodies can be described by the following conventional differential-algebraic equation (DAE), widely used in multibody dynamics [4]:

$$\begin{bmatrix} \mathbf{M} & \mathbf{C}_\mathbf{x}^\mathrm{T} \\ \mathbf{C}_\mathbf{x} & 0 \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{x}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_e \\ \boldsymbol{\gamma} \end{bmatrix}, \tag{1.11}$$

where $\mathbf{M}$ is the mass and inertia matrix of the multibody system; $\ddot{\mathbf{x}}$ is acceleration vector; $\mathbf{F}_e$ and $\mathbf{C}_\mathbf{x}$ are vectors of external forces and constraint equations respectively; $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers; $\boldsymbol{\gamma}$ is a vector that can be obtained by differentiating the constraint equations with respect to time twice:

$$\boldsymbol{\gamma} = \mathbf{C}_\mathbf{x}\ddot{\mathbf{x}} \tag{1.12}$$

Eq. (1.6) offers a linear solution that well describes the dynamics of floating cranes over a wide range of operating conditions. However, the motion of a floating crane is influenced in one way or another by different nonlinearities, which introduce complexities in determining the dynamic response. The nonlinearities are caused by various phenomena such as nonlinear waves, influence of mooring systems, kinematic coupling between the hull and the load, viscous friction in fluid-structure interaction and many other sources. Ellermann et al. [5] investigated the nonlinear dynamic responses of moored crane vessels to regular waves. Their study consists of an experimental and a theoretical parts. In the experimental part, the authors used a moored crane vessel model and investigated its motions in a wave tank (Fig. 1.4). The experiments indicated that subharmonic motion occur in the dynamics of crane vessels [5].

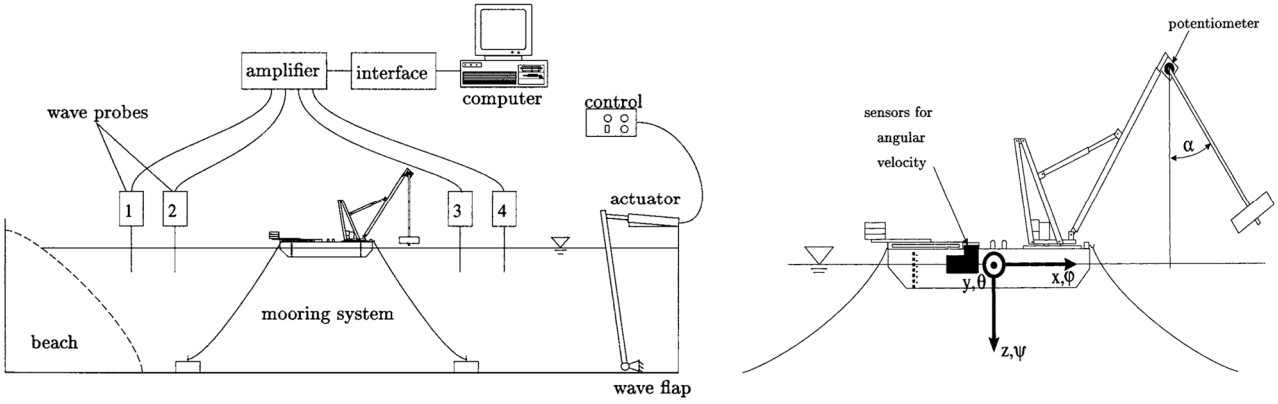


Fig. 1.4 Experimental setup in Ellermann et al. [5]

The theoretical part consisted in mathematical modeling of floating cranes using two models. The authors derived the equation of motion of the moored vessel and took into account hydrodynamic forces such as the restoring forces, the mooring line forces, forces due to viscous drag, the frequency-

dependent wave excitation forces and the radiation forces calculated as for state space model. The results of modeling showed that the large resonant oscillations (one-periodic motion) may occur if the wave frequency approaches the system eigenfrequency corresponding to a strongly coupled surge-swing motion [5]. Also, authors stated that for perfectly balanced vessels, the heave motion is coupled with other degrees-of-freedom only through nonlinear terms.

The development of numerical modeling techniques (computational mathematics and mechanics) of physical processes and phenomena has led to the creation of many specialized commercial and in-house software that use methods of mathematical modeling to obtain an understanding of the behavior of crane ships. For instance, Cha et al. [6] developed their own framework for modeling crane ship dynamics and shipbuilding process planning. This software includes modules for dynamics calculation and collision detection based on the Open Dynamics Engine (ODE) [7] and a visualization module based on Open Scene Graph (OSG) [8]. The authors developed the integrated module DSG, which combines ODE and OSG into a unified system as illustrated in Fig. 1.5. Additionally, the authors of this work implemented the ability to model the cable as a damped mass-spring element. However, regarding the hydrodynamics of the crane vessel, hydrostatic and hydrodynamic forces were not fully considered, as in the methods mentioned earlier. They were approximated as forces of a damped mass-spring system. Although the authors did not provide a full description of the equations, the approach described is a significant simplification that does not allow for the automatic consideration of various sea states.

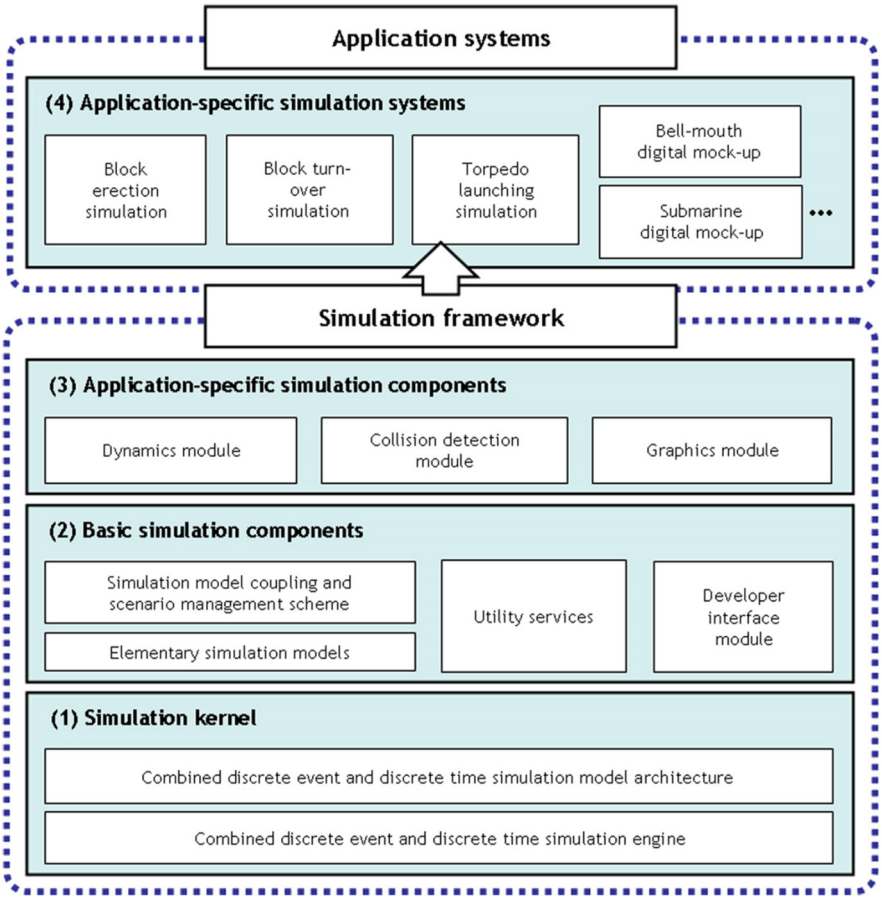


Fig. 1.5 Configurations of integrated simulation framework proposed in Cha et al. [6]

In their subsequent work, Cha et al. [9] modeled a real floating crane Daewoo as a MBD system with 12 degrees of freedom. The model is illustrated in Fig. 1.6. In this paper, the authors considered hydrodynamic and hydrostatic forces in an extended form. The radiation terms were computed tacking into account the fluid memory effect with added mass and radiation damping coefficients calculated in commercial software. The nonlinear hydrostatic force was calculated at every time step by using the pressure integration technique. The wave excitation force was calculated by considering the response amplitude operator (RAO) also obtained from a commercial system [9]. They also used a constraint equation similar to Eq. (1.10) to model the constraint forces of the cable. The simulation results, such as motions of the floating crane and the heavy cargo, and the cable tension force, were compared with the results obtained in the commercial software MOSES, showing good convergence. An obvious challenge of the applied method is the need to determine hydrodynamic coefficients in external commercial software.

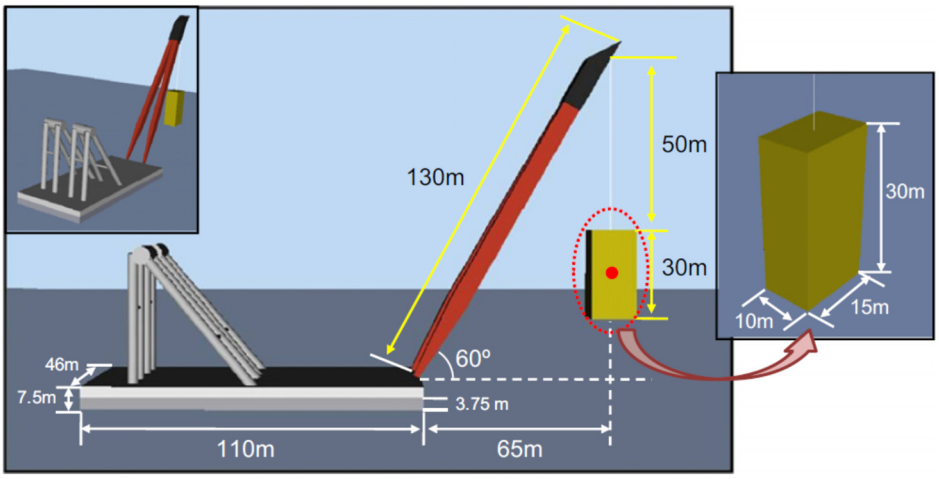


Fig. 1.6 The principal dimensions of the floating crane and the cargo in the dynamic response simulation conducted by Cha et al. [9]

Another MBD kernel for dynamic analysis of floating cranes was developed and verified by Ku and Ha [10]. They introduced the recursive MBD formulation for computation of dynamic motions of rigid body systems. The developed kernel was verified by comparison of simulation results with the commercial programs such as RecurDyn. The simulated cases included abstract MBD models and demonstrated full convergence with the results obtained in commercial solvers. The software provides tools for modeling predefined scenarios of cargo handling by floating and land-based cranes (in the example, the authors mentioned the Goliath crane). In this software, hydrostatic and hydrodynamic loads are considered more comprehensively than in previous examples. Hydrostatic forces are calculated by integrating buoyancy forces over the volume of the immersed mesh of the floating body, the hydrodynamic forces are calculated in time domain using 3D Rankine panel method via Laplace equation and incident wave potential. The authors also provided examples of simulations using their software, one of which is shown in Fig. 1.7, demonstrating the process of loading a ship with two floating cranes.

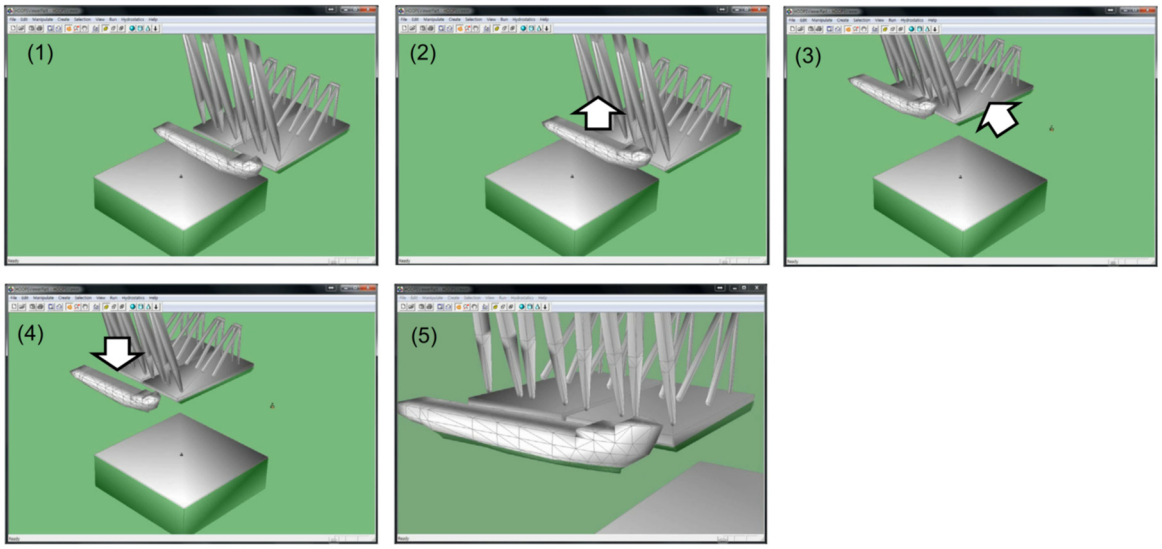


Fig. 1.7 The result of the simulation of the lunching process of the ship using two floating cranes demonstrated by Ku and Ha [10]

Further this MBD simulator was modified by Ha et al. [11]. In this paper authors raised the problem of the absence of specific items in the field of naval architecture and ocean engineering, such as hydrostatics, hydrodynamics, and mooring forces in existing MBD tools. The improved software included the kernels for multibody system dynamics, force calculation, the numerical analysis, hybrid simulation kernel, scenario management and collision detection as presented in Fig. 1.8. Also, it has a GUI and database layer for configuration of geometric properties of the modeling objects. The methods of computation the hydrodynamic forces remain the same as in previous version.

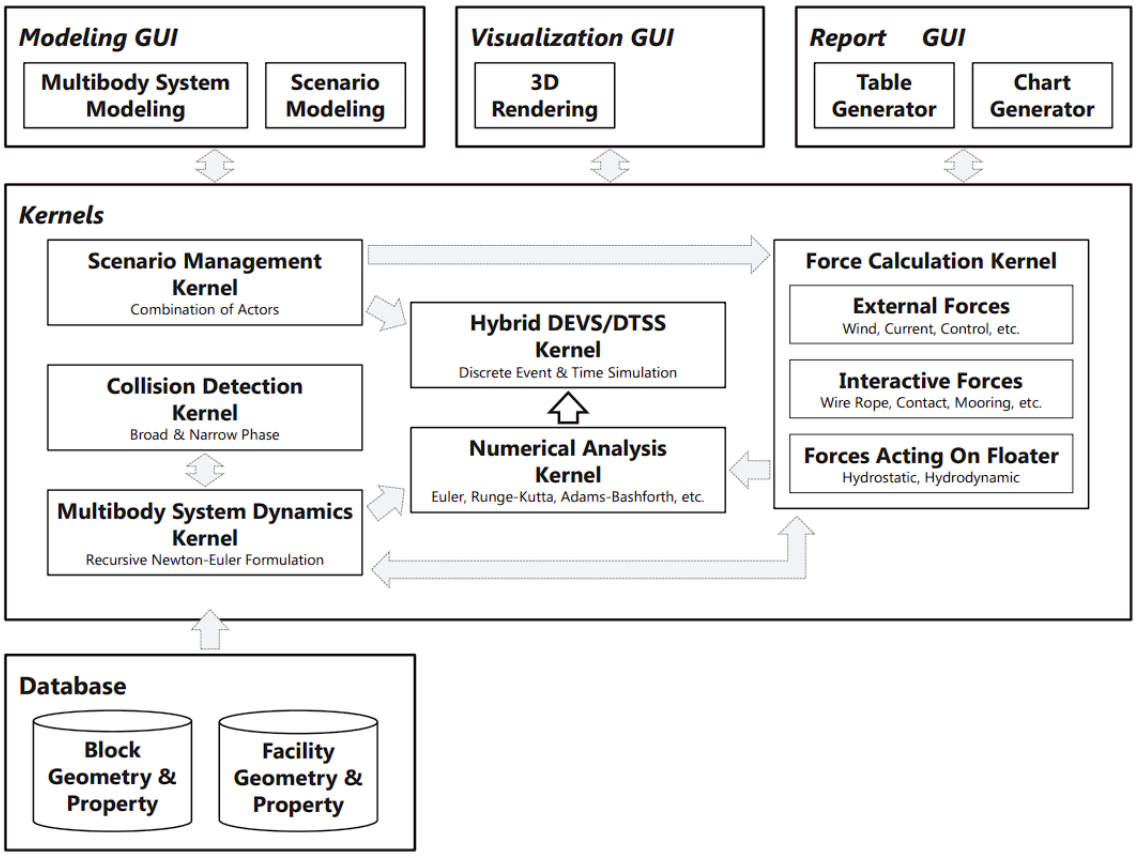


Fig. 1.8 Configuration of the multibody system dynamics simulator developed by Ha et al. [11]

Thus, the aforementioned and other relevant scientific works somehow incorporate certain terms from equations (1.4) or (1.6) to calculate the dynamic response. There are also studies with an extended approach, including, for example, structural mechanics equations to account for the deformability (elasticity) of crane structural elements. In this context, methods for analyzing the behavior of the suspended load in the "sea-ship-load" system in the time domain may be complemented by other governing equations depending on the adopted approach. One of the examples of such studies is the paper by Hong et al. [12], in which the authors simulated the hydrodynamic response of a simplified model of crane ship Skandi Arctic and considered the crane boom as a flexible body. They use finite element method for modeling the crane boom and the conventional DAE equation (same as Eq. (1.11)) but with additional terms for calculation of flexible body response. Wave excitation and radiation forces were computed in the same way as in Cha et al. [9] using commercial hydrodynamics tool WADAM by DNV. As for hydrostatic forces they were calculated in simplified way as force equal to the displaced mass of water. After the analysis of simulation results authors concluded that the flexible models had a bigger dynamic amplification factor than rigid models in all cases, suggesting that the flexibility of the crane boom should be considered to accurately estimate the dynamic effect on crane-lifting.

Here we can also highlight the problem of using simplified approaches for modeling the hydrodynamics of a crane vessel, which require the use of commercial software to initialize the model. In the understanding of the author of this thesis, there is the fact that today there are various software solutions for modeling floating objects with tools for hoisting system modeling, both commercial and free. However, they are non-extensible, limiting their ability to solve only a limited number of tasks. Consequently, the challenge of developing a methodology for modeling crane vessels in accessible and extensible software remains relevant to this day.

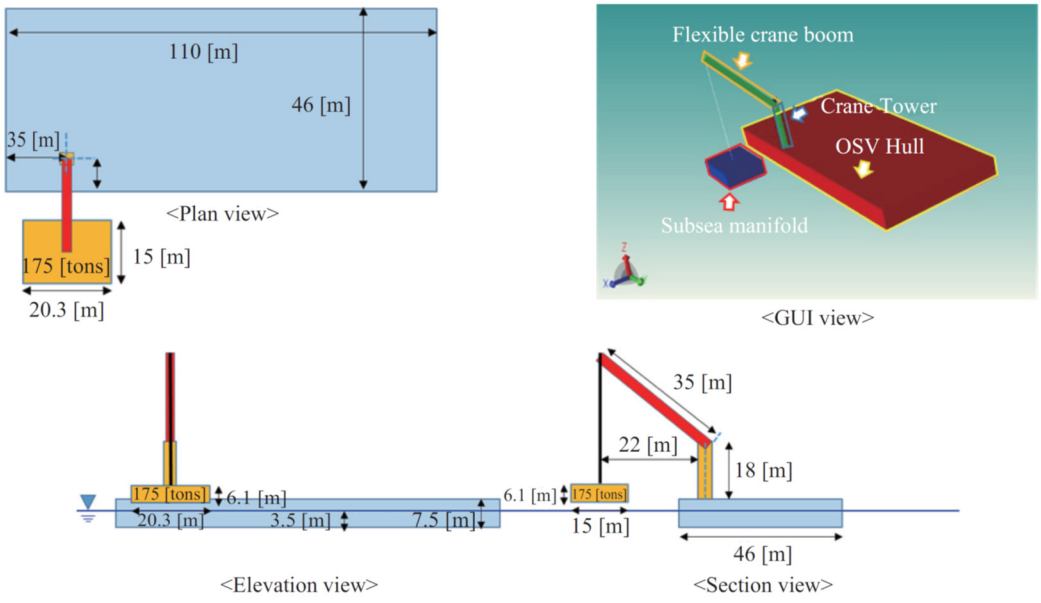


Fig. 1.9 Simulation model of Skandi Arctic and the subsea manifold used by Hong et al. [12]

1.2. Determination of hydrodynamic forces on the floating bodies

1.2.1. Brief discussion

Based on the aforementioned principles, the determination of hydrostatic and hydrodynamic forces is one of the fundamental tasks in the study of the dynamics of floating cranes and floating objects in general. Their determination is possible using various hydrodynamic approaches that are not specific to floating cranes but are applied in the modeling of any floating bodies. However, having a suspended load on a floating crane increases the complexity of the problem. The dynamics of the load and the vessel is coupled, and the degree of interaction varies depending on the current system parameters—such as the load mass, natural frequency (which, in turn, depends on the current length of the cable), wind loads on the suspended load, and other factors. Consequently, the determination of hydrodynamic forces must take into account their interdependence, leading to the need to solve a coupled multiphysics problem.

Further, the main methods of computational hydrodynamics currently used to determine the hydrodynamic parameters of floating bodies and the corresponding external forces are described. Computational Fluid Dynamics (CFD) methods have undergone significant development, encompassing various approaches to determine velocities, seawater pressures, and associated forces, each with its own set of advantages and disadvantages. The application of such methods today is carried out through mathematical modeling tools and specialized CFD solvers, allowing for the determination of the magnitude and characteristics of the marine environment's impact on a floating body. Among the many software programs available, both commercial and free programs exist, each with its algorithms and capabilities. A comprehensive review and analysis of CFD methods are provided by Zullah and Lee [13], where the authors assess and compare numerical methods and software for modeling wave energy converters (WECs). Although the authors focus on WECs, their conclusions are applicable to ships, including floating cranes.

In general, CFD methods for determination of wave impacts on floating structures can be categorized as follows.

1.2.2. Navier Stocks equations

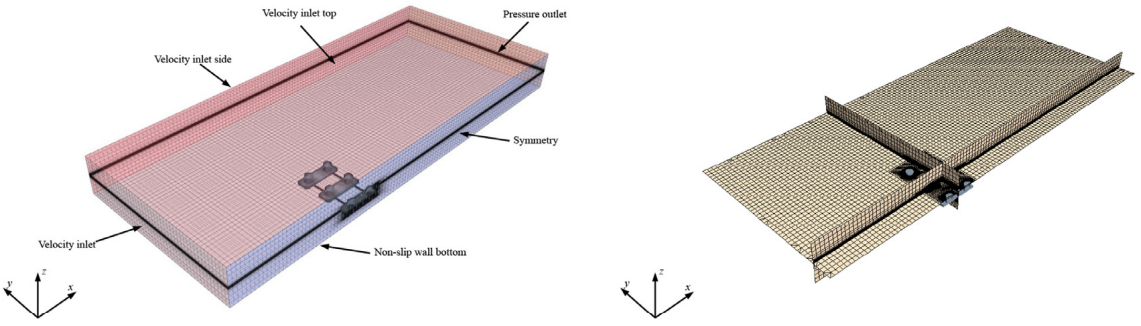
The Navier-Stokes equations are partial differential equations widely used in computational fluid dynamics. Numerical modeling using these equations involves discretizing the entire fluid domain into individual elements and computing the velocity and pressure fields at each integration point. This can be achieved, for example, using the Finite Volume numerical method (FVM). In vector form, the Navier-Stokes equations are given by:

$$\frac{\partial \boldsymbol{\upsilon}}{\partial t} = -(\boldsymbol{\upsilon} \cdot \nabla)\boldsymbol{\upsilon} + \nu \Delta \boldsymbol{\upsilon} - \frac{1}{\rho}\nabla p + \mathbf{f}, \tag{1.13}$$

where $\boldsymbol{\upsilon}$ is the velocity vector; t is time; ν is the kinematic viscosity of the fluid; ρ is the density; p is the pressure; $\mathbf{f}$ is the vector of mass forces; ∇ and Δ are the nabla and Laplace operators, respectively.

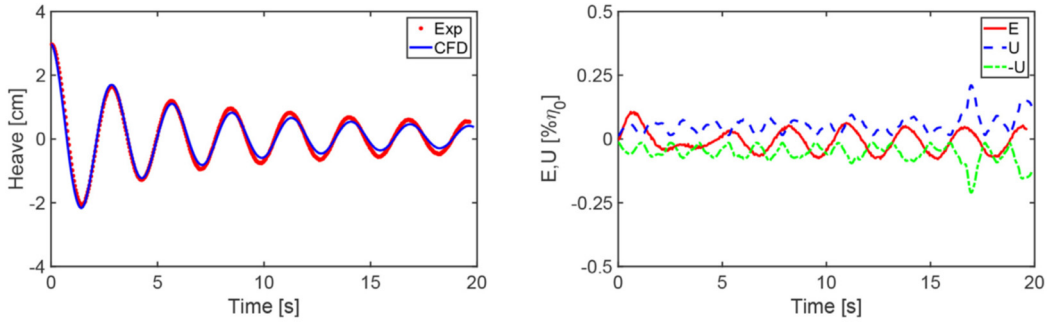
The numerical models with this method provide the most comprehensive insight into the dynamics of a floating object as they can consider the fluid as viscous and compressible and the flow as turbulent. There are four major numerical methods to simulate wave turbulence flow called the direct numerical simulations (DNS), large eddy simulations (LES), Reynolds-averaged Navier-Stokes methods (RANS) and detached eddy simulations (DES) [13]. Among them, RANS methods have found the most application due to offering the relatively acceptable computation speed. However, the computational cost of such calculations is still extremely high for modeling different sea states since, during simulation, it is possible to consider only one set of wave conditions.

For example, Wang et al. [14] performed CFD simulations of free-decay tests to determine the damping coefficients of a floating structure. The computational mesh is presented in Fig. 1.10. When comparing the results of simulation and physical experiments, the authors achieved nearly complete convergence with an error not exceeding 0.25% for the heave degree of freedom (Fig. 1.11). However, an evident drawback of this numerical method lies in its computational speed. The simulation time for a 20-second case was up to 44.4 hours. Considering the necessity to calculate different states and directions of waves, wind, and other factors, such studies would turn into very time-consuming and labor-intensive numerical experiments.



(a) 3D mesh with boundary conditions (b) Background and overset mesh at three plane sections

Fig. 1.10 Computational mesh domain in simulations performed by Wang et al. [14]



(a) Time history comparison (b) Error and uncertainty

Fig. 1.11 Validation of heave free-decay motions in physical and numerical experiments by Wang et al. [14]

Michailides [15] introduced a hybrid method for modeling the dynamic response of the WEC with velocity-dependent viscous damping (VD-PQ). The method involves creating two coupled models (Fig. 1.12) – a computational fluid dynamics model (CFDM) used to estimate the viscous damping loads iteratively in every time-step via Navier–Stokes equations and a potential flow model that computes the motions of WEC via Cummins equation. The coupling is realized through a dynamic-link library (DLL). According to the author the proposed numerical analysis method for estimating the viscous damping loads provides good accuracy compared to experimental data and, at the same time, low computational cost [15]. Although the author's approach is promising, the performance of the method has not been presented.

As a rule, computing partial differential equations for the entire mesh domain does not allow achieving high computational speed, especially real-time computing. Therefore, the implementation of Multibody Dynamics (MBD) and Computational Fluid Dynamics (CFD) co-simulation with Navier–Stokes equations in the time domain, or the so-called two-way coupling, has not found wide application in real engineering practice. Hence, in maritime engineering, alternative approaches are employed to simplify and expedite calculations.

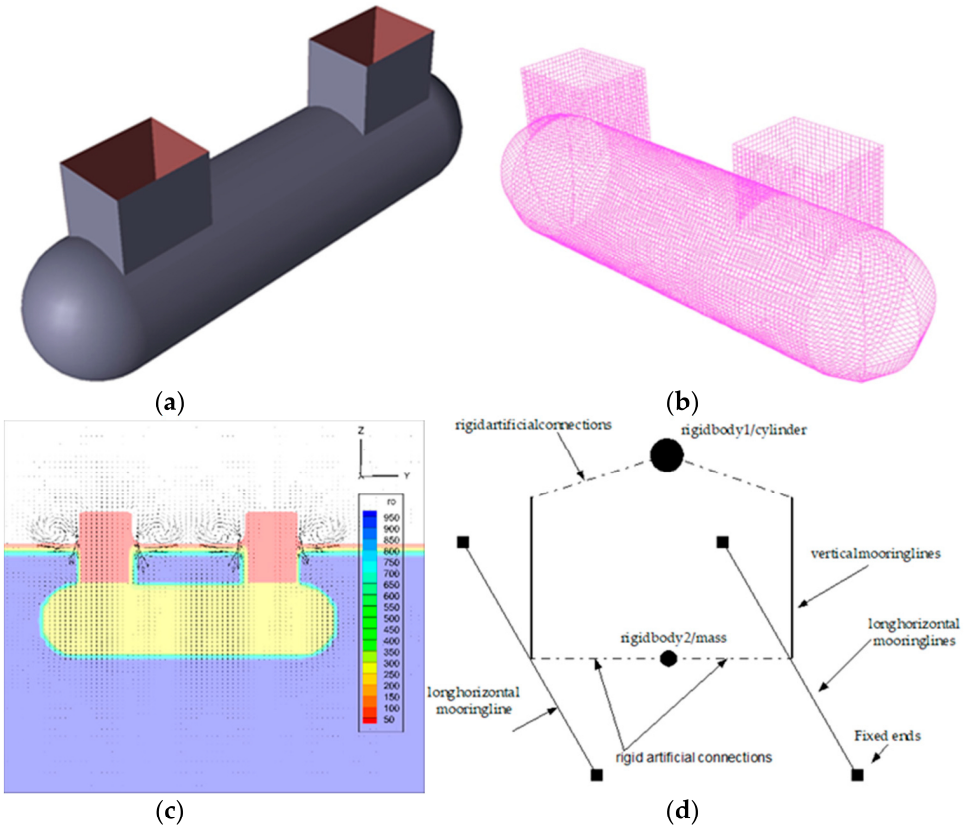


Fig. 1.12 Numerical models of wave-energy converter developed by Michailides [15]: a) potential theory (PTM) numerical model; b) panel discretization of the wet surface of the PTM model; c) computational fluid dynamics model for decay curves calculation; d) 3D model of WEC

1.2.3. Potential flow theory

Instead of the Navier–Stokes equations, simplified CFD methods use the Bernoulli equation to couple the velocity and pressure of the fluid [3, p. 14]:

$$p + pgz + \rho\frac{\partial\varphi}{\partial t} + \frac{\rho}{2}\boldsymbol{\upsilon}\cdot\boldsymbol{\upsilon} = f(t) \tag{1.14}$$

where p is a pressure; g is gravity acceleration; z is vertical coordinate (z-axis is positive upwards); ρ is fluid density; φ is the fluid velocity potential; $\boldsymbol{\upsilon}$ is the fluid velocity vector, and $f(t)$ is an arbitrary function of time.

The velocity vector and velocity potential are coupled as follows:

$$\boldsymbol{\upsilon} = \nabla\varphi = \nabla = \frac{\partial\varphi}{\partial x}\mathbf{i} + \frac{\partial\varphi}{\partial y}\mathbf{j} + \frac{\partial\varphi}{\partial z}\mathbf{k} \tag{1.15}$$

The Eq. (1.14) is valid for unsteady, irrotational and inviscid fluid motion, so in such a case the velocity potential satisfy the Laplace equation:

$$\nabla^2\varphi = \Delta\varphi = \frac{\partial^2\varphi}{\partial x^2} + \frac{\partial^2\varphi}{\partial y^2} + \frac{\partial^2\varphi}{\partial z^2} = 0 \tag{1.16}$$

The complete mathematical problem of finding a velocity potential of irrotational, incompressible fluid motion consists of the solution of the Laplace equation with relevant boundary conditions on the fluid [3]. In such a way the derivation of linear wave theory (Airy theory) can be performed. This theory is then used to estimate the wave-induced forces acting on the floating structure. Therefore, instead of computing the fluid state at each point of computational domain this theory assumes finding the solution only on the boundary of diffracting surface (vessel hull).

The hydrodynamic parameters involved in the Cummins equation (Eq. (1.4) and Eq. (1.6)) are determined much faster compared to modeling the Navier–Stokes equations. However, the assumption of inviscid and irrotational flow is an obvious drawback, reducing the accuracy of the calculation and requiring additional tools to achieve an acceptable result (Fig. 1.13 Of course, there are methods to overcome these assumptions, for example, by calculating viscous damping coefficients, as in the previously mentioned papers [14, 15]. In general, there are two approaches using potential flow theory.

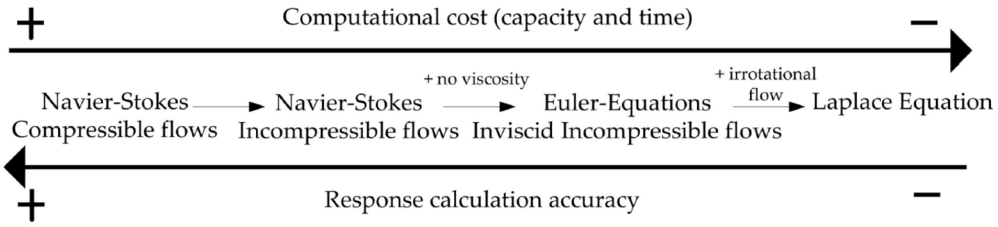


Fig. 1.13 Contradiction between response calculation accuracy and computational cost with different CFD methods [15]

Strip theory.

The description of strip theory is provided here according to Faltinsen [3, p. 50]. Strip theory implies in dividing the submerged part of the vessel into a number of strips as presented in Fig. 1.14. Then, two-dimensional hydrodynamic coefficients are calculated for each strip and combined according to which added mass and damping coefficient is wanted. Using a strip theory assumes that the variation of the flow in the cross-sectional plane is much larger than the variation of the flow in the longitudinal direction. It also should be noted that strip theory is a low Froude number theory. It does not properly account for the interaction between the steady wave system and the oscillatory effects of ship motions [3, p. 57]. The advantage of strip theory is high computation speed. However, this theory is outdated and is almost no longer applied in practice due to the development of another more modern method – panel methods.

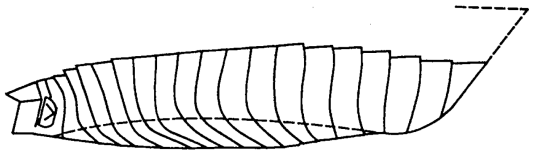


Fig. 1.14 Illustration of strip theory for ships [3]

Panel method.

The panel method is a numerical approximation method that implies on using discrete elements on the surface of a simulated object as illustrated in Fig. 1.15. The boundary problem on finding the fluid velocity is solved with the help of the Green function in each collocation point on the panel. This method allows for an optimal balance between accuracy and computational speed, efficiently determining the hydrodynamic parameters of a floating object for various frequencies and wave directions. As a result, it is widely used today in various seakeeping codes. For example, the previously mentioned studies [6, 9-12] specifically utilize the panel method to determine hydrodynamic forces.

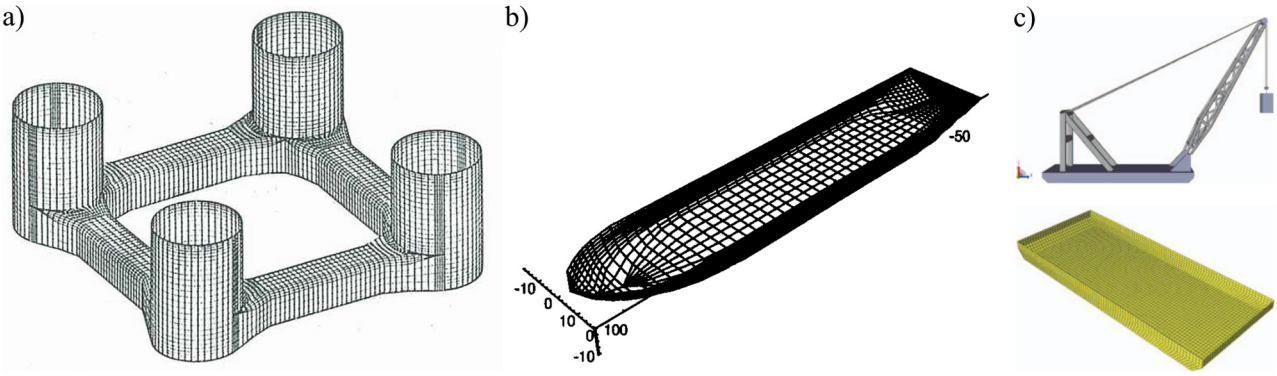


Fig. 1.15 Discretization of the submerged surface of floating body: a) The Snorre Tension Leg Platform with 13952 panels [16]; b) wetted surface of crane barge in [5]; c) the model of crane ship and its wet surface mesh in [17]

The panel method, like any other numerical method, has its limitations, as mentioned, for instance, in Peña and McDougall [18]. However, its ability to model complex geometry of the

submerged part has facilitated its application not only in marine engineering but also in aerospace engineering and other fields. Numerous commercial and open-source programs have been developed to perform Computational Fluid Dynamics (CFD) computations using the panel method. Some of them are discussed in Section 2.3.2 of this thesis. Practically all of these programs use the Boundary Element Method (BEM) as a discretization method for transforming the surface into discrete panels. The BEM method is well-suited for implementing the panel method because it embodies the same idea of solving a boundary problem.

Shigunov et al. [19] conducted a comprehensive benchmark study of numerical methods for the prediction of time-average wave-induced forces and motions of ships in waves. Authors stated that 3D panel methods are applicable to a wide variety of practical hull forms and environmental conditions, but they modestly depend on the qualification and background of the software user. Liu et al. [20] performed numerical and experimental study on linear and nonlinear seakeeping phenomena for the DTC containership in regular waves. Authors used 3D panel method in NEWDRIFT+ software for simulation and compared the modeling results with corresponding experimental data obtained from tests at the towing tank and ocean basin facilities of MARINTEK for various wave headings at low and moderate speeds. Based on the results, one can say that in general the 6DOF vessel motions were satisfactory except for the cases of oblique wave action. The strong divergence with experiments in this case is due to the varying and asymmetric wetted surface of the ship and the resulting highly nonlinear response of the experimental model.

Thus, the panel method continues its development due to the lack of alternatives in terms of speed and accuracy of calculations. Its application is constrained by the potential flow theory, specifically for small wave heights and small displacements of floating bodies. Nevertheless, new approaches are being developed to overcome the limitations of the potential flow theory. An interesting approach was implemented in the work of Eskilsson et al. [21], where the authors introduced a hybrid LPF-ML (Linear Potential Flow Machine Learning) approach. This method involves determining a nonlinear correction factor as the difference between the forces from high-fidelity CFD and LPF models. Through multiple simulations and machine learning, the authors obtained a Long Short-Term Memory (LSTM) network, which can apply additional corrective forces at each calculation step to improve accuracy while maintaining computational efficiency.

1.3. Modeling of cargo lifting and control of suspended loads motions

As marine construction and loading operations are the primary tasks performed by floating cranes, focusing solely on modeling the dynamics of the floating structure will not be sufficient. Clauss and Riekert [22] emphasized that the vertical motion of suspended load is the most critical parameter in crane vessel operation. When transitioning to the modeling of the dynamics of the

suspended load itself, several works by other researchers address this aspect. In these studies, the authors have implemented stages beyond lifting in the air, such as lifting off, splash zone crossing, submerging, and installation on the seabed. For example, the previously mentioned software for modeling crane ships [6, 9, 11] not only provides tools for modeling the hydrodynamics of floating cranes but also offers the possibility of controlling the position of the cargo. Fig. 1.16 illustrates a scenario of lifting a ship block with a floating crane in the software developed by Cha et al. [6]. All movements planned in the simulation are specified in a separate scenario file. When the simulation starts, the event generator gets events from the scenario manager and transmits them to the floating crane simulation model.

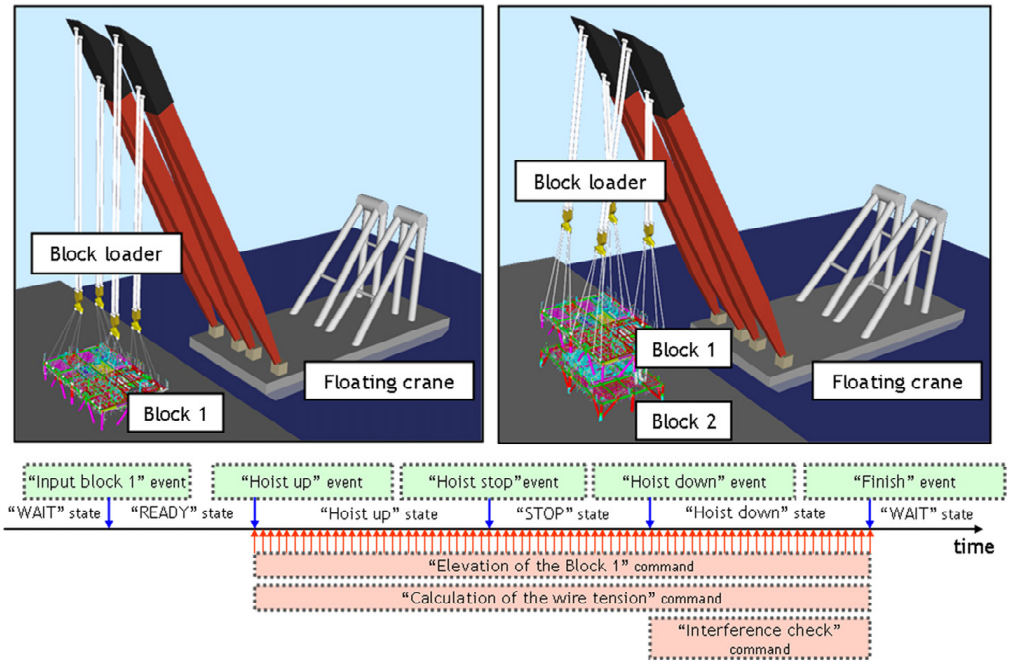


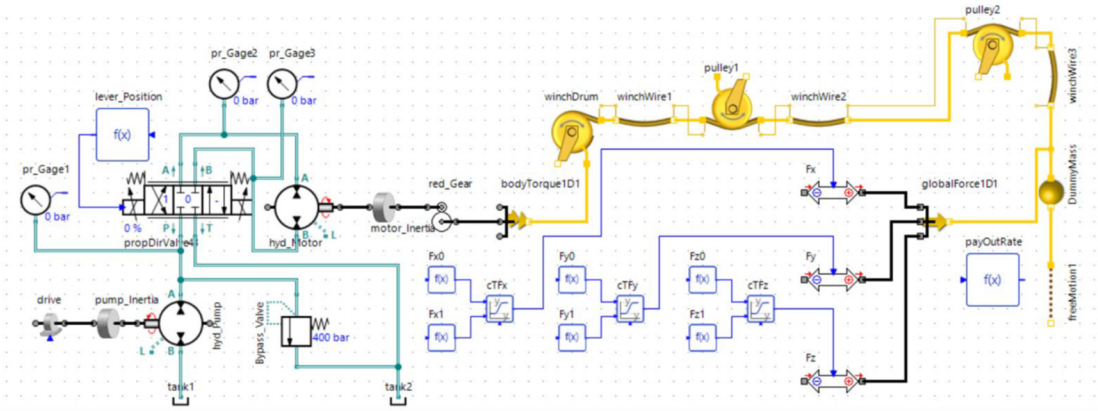
Fig. 1.16 Scenario of the block lifting simulation by Cha et al. [6]

Jeong et al. [23] conducted a simulation of a lifting operation where an Offshore Support Vessel (OSV) installs offshore structures, such as subsea equipment (Fig. 1.17). The authors incorporated collision detection into the model to alter the motions of collided objects by adjusting their velocities. They analyzed the movements of the suspended load, including cable tension, during lifting off, lifting in the air, and splash zone crossing operations. The buoyancy force acting on the suspended load was calculated based on the submerged volume using Archimedes' principle. Additionally, the model included a slamming impact force to account for the impulsive response during splash zone crossing.

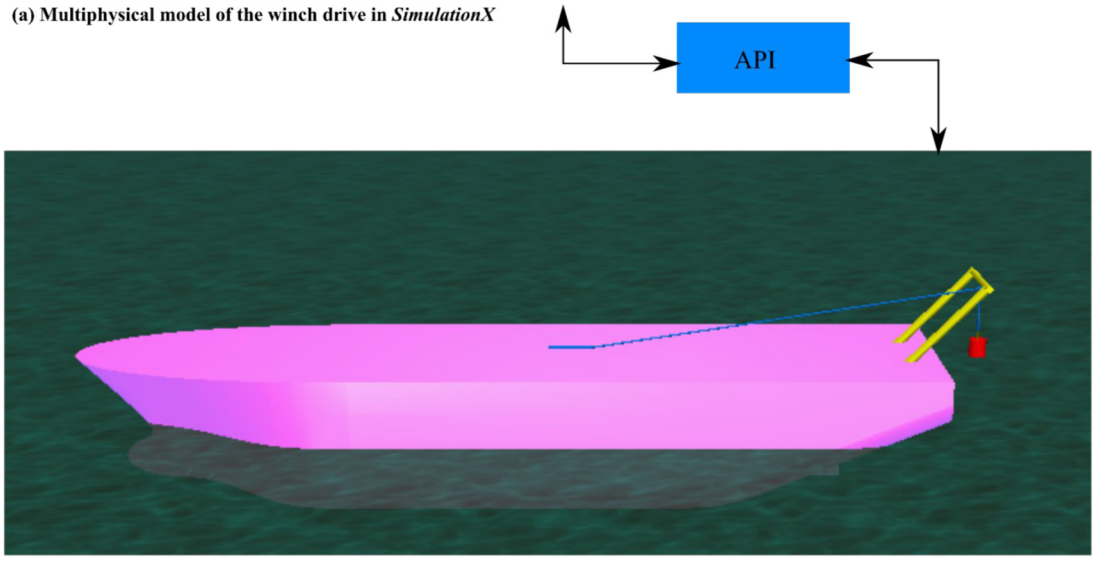
Viswanathan et al. [24] implemented a coupling between Orcaflex and the MBD solver SimulationX through their own developed application programming interface (API) to link the computation of hydrodynamics and cargo dynamics between the different software (Fig. 1.18). Such multiphysics co-simulation approach eliminates the dependence of mechanics on commercial software but has obvious drawbacks from a technical point of view. If the CFD and MBD solvers and

the API are implemented in different programming languages, the obvious effect of this connection will be a slowdown in the computation speed. According to the authors the simulation of 260-s event took approximately one hour. In addition, upgrading a version of one software cannot guarantee reliable functioning with an old version of another solver and may require an update of the API linking them. Therefore, to ensure the reliability and speed of calculations, minimizing the number of links between separate software is necessary.

Fig. 1.17 Actual and simplified models for the simulation as presented in Jeong et al. [23]: a) actual models; b) simplified models.



(a) Multiphysical model of the winch drive in *SimulationX*



(b) Model of the OSV with payload in *OrcaFlex*

Fig. 1.18 The models used by Viswanathan et al. [24] in co-simulation via SimulationX and OrcaFlex.

Landsverk et al. [25] simulated a rigid-body model of crane vessel with 250-ton active heave compensated knuckle boom crane and payload. The final model consists of ten rigid bodies. The hydrodynamic coefficients were determined in VERES/ShipX software and then usen in multibody simulation conducted using Marine Systems Simulator (MSS) MATLAB toolbox [26]. To simulate wave impact authors used the JONSWAP frequency spectrum with significant wave height of 2 m, and peak wave period 8 sec. A total of three cases with predefined rotations of the crane and strokes of the cylinders were considered. The practical significance of the work was the successful construction of a fully coupled model of a crane ship, which was fast computable and had the potential for improvement due to the fact that it was based on MATLAB. In our opinion, the approach to the use of popular and easily extensible software, realized in this paper, is promising. In this case, improvement of the numerical model and introduction of new functionality is greatly simplified due to the capabilities of MATLAB programming language and a large community of engineers using this software.

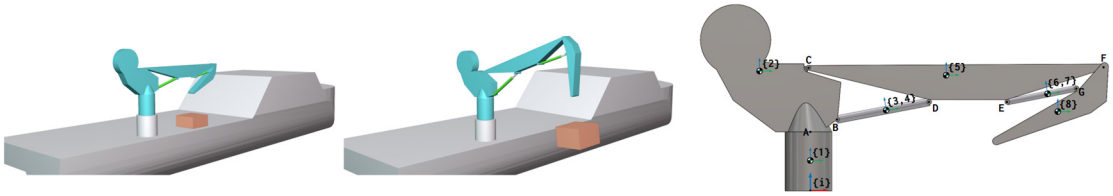


Fig. 1.19 Model of floating crane with knuckle boom crane developed by Landsverk et al. [25]

Vorhölter et al. [27] carried out a time-domain analysis of lifting operations conducted by three different vessels under two different load variations. One of the simulated cases is presented in Fig. 1.20. Authors used JONSWAP-type wave spectrum with directional cosine spreading to model the propagating waves. In the conclusion, they highlighted that, without additional measures to control the motion of the load, like tugger lines or active motion compensation of the crane the sway and surge motions of the load in simulations were too large in order to perform a safe lifting operation [27].

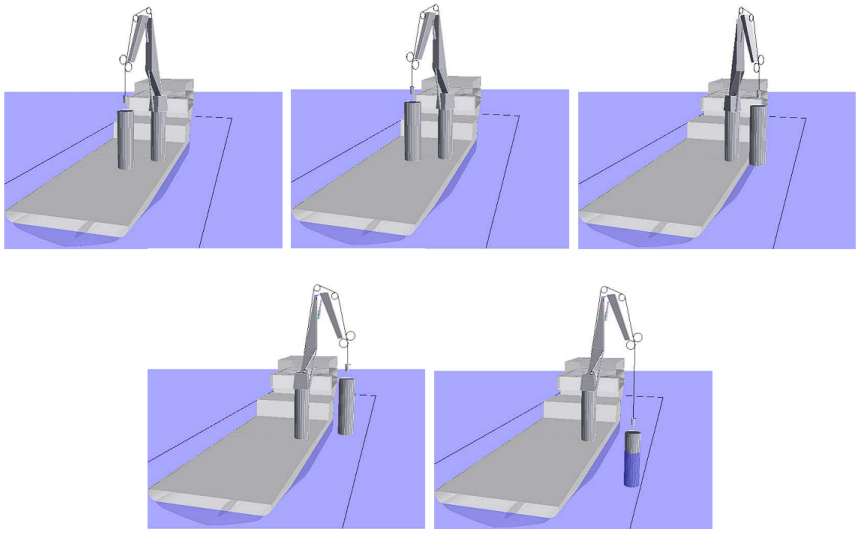


Fig. 1.20 Crane operation by Offshore Construction Vessel presented in Vorhölter et al. [27]

The implementation of cargo motion control in the aforementioned papers relies on predefined events (or scenarios), where the motion of crane parts and the suspended load lifted by the hoisting system is specified by the user before model initialization. They do not allow for changes in position on demand during the computation process. Thus, the capabilities of these software systems may not always meet the demands of the task at hand. While such software can be extremely useful, for example, in shipbuilding process planning, it may not be applicable in real-time applications. This underscores a crucial requirement for a modeling system – the ability to consider not only idealized pre-defined scenarios but also to directly and interactively control the model during the calculation process.

Li et al. [28] developed an integrated method for a collaborative simulation that allows multiple operators to control cranes together in the virtual world as presented in Fig. 1.21. This system consisted of multibody dynamics solvers with virtual reality visualization and control devices (joystick and head mount display) for handling and tracking crane wire and viewpoints. This paper focuses on a detailed description of the approach to design of the control system using available and self-developed modules for dynamics simulation, visualization and data exchange. To demonstrate the capabilities of the developed approach, the authors performed simulations of representative marine operations – block turn-over (Fig. 1.21) and offshore module installation (Fig. 1.22). The developed approach has a high technological level and, in the future, can be used in interactive training simulators for crane operators or remote control of real cranes.

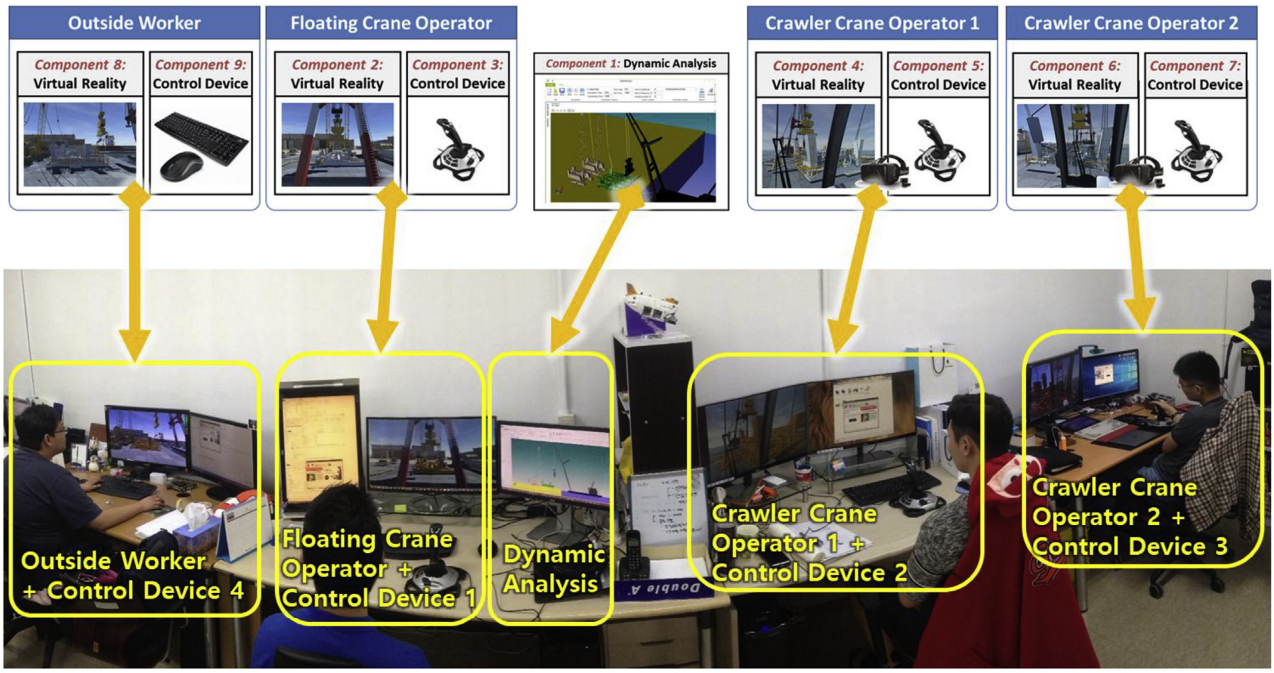


Fig. 1.21 Implementation of the collaborative simulation of the block turn-over presented by Li et al. [28]

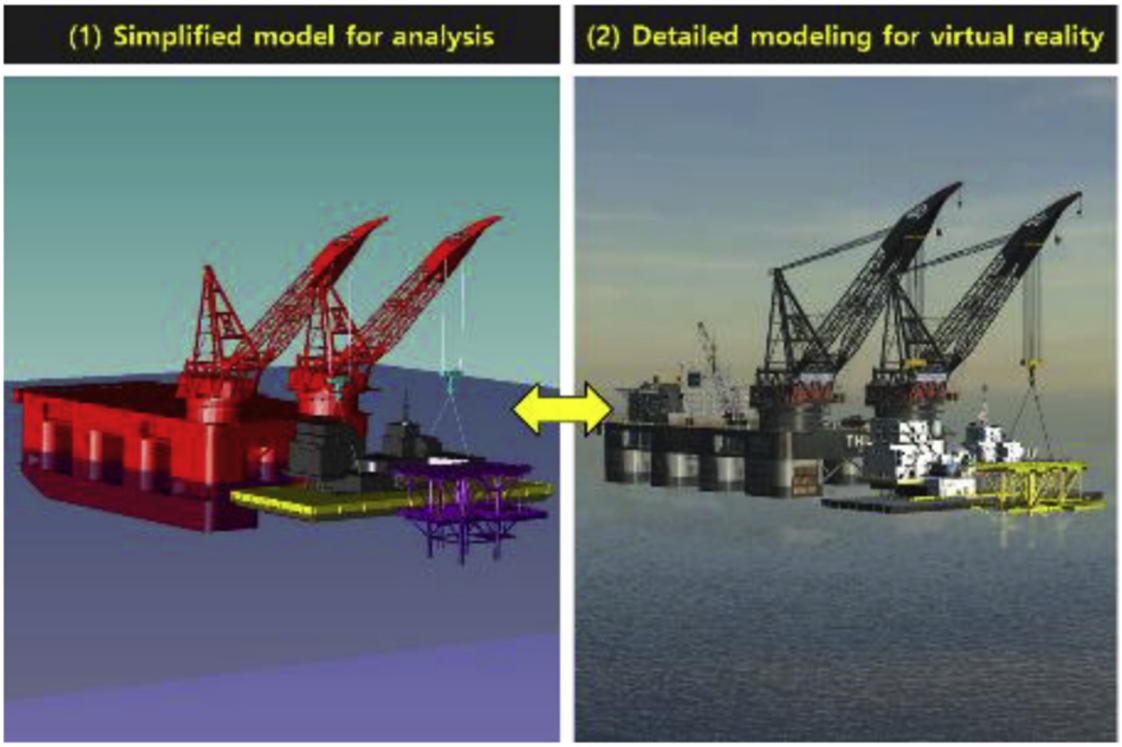


Fig. 1.22 The simplified dynamic-analysis model and the detailed VR model of offshore module installation process presented by Li et al. [28]

Chu et al. [29, 30] presented a numerical model of a crane manipulator with a compensation system of load swaying implemented through hydraulic cylinders (Fig. 1.23). During the simulation, the cargo position was controlled by joysticks. The authors managed to achieve a good decrease in cargo oscillation by using PID controllers that correct the control signal of the hydraulic cylinders (Fig. 1.24). The approach showed excellent results but is applicable to cranes without a complex cable system. The authors also mentioned that to achieve a generic solution for modeling and real-time simulation of complex multi-domain systems, an interfacing framework for integration of the sub-modules needs to be developed [12].

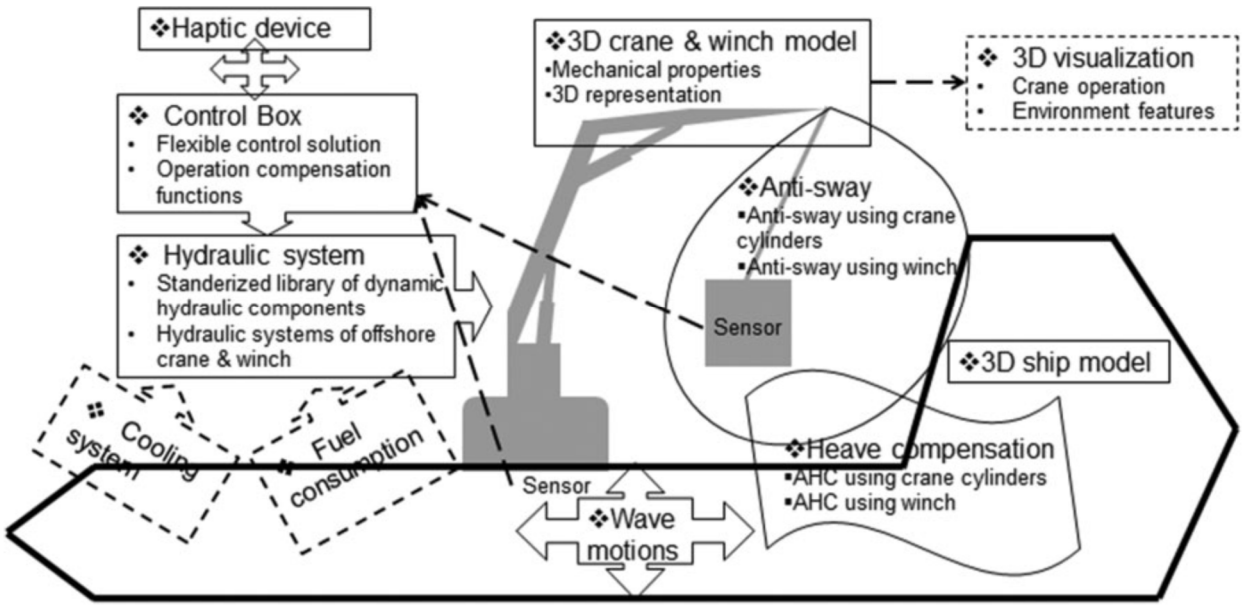


Fig. 1.23 Offshore hydraulic crane modelling structure in paper of Chu et al. [30]

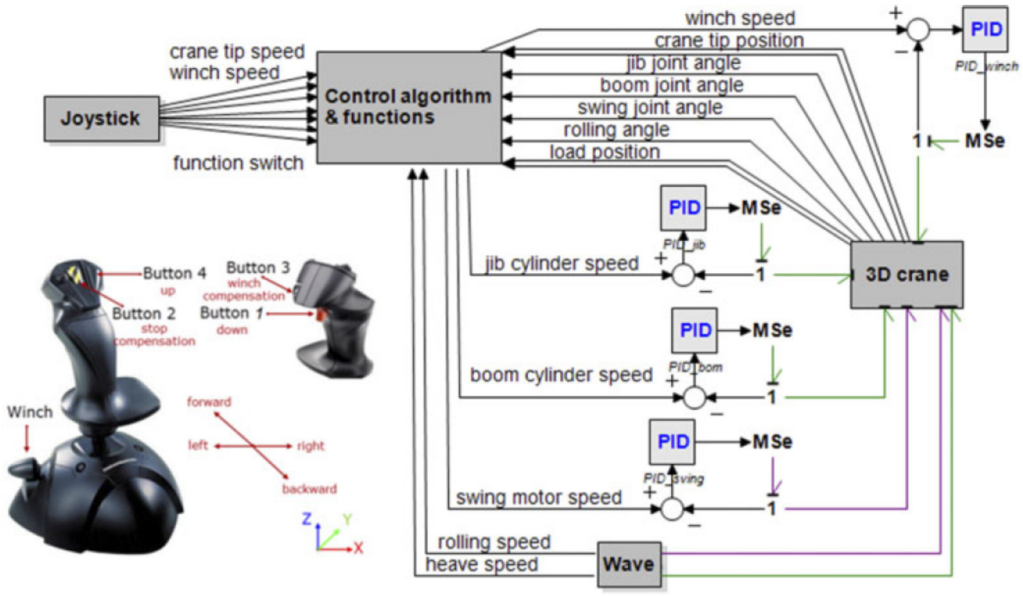


Fig. 1.24 Crane control model with compensation functions without hydraulics developed by Chu et al. [30]

Ye et al. [31] presented the design of a nonlinear control scheme for stabilizing the heavy load position during offshore operations of a crane vessel. The authors' contribution is to consider the problem of load control taking into account the influence of the dynamic positioning system of the vessel itself. The authors' crane vessel model is illustrated in Fig. 1.25(a). The authors formulated a 3DoFs dynamic model of the load motion and derived a reduced-order model, taking into account the vessel position controlled by dynamic positioning system. The design of an underactuated nonlinear controller is based on backstepping and command filtering approach. The control system adjusts the hoist force in the crane wires in order to keep the load in a desired position. Simulation results of the application of the designed control scheme to system are presented in Fig. 1.25(b) as well. The designed control method has demonstrated good performance in keeping the load in a desired position. A total of 3 simulated cases with different load masses – 500 tons, 1000 tons and 2000 tons – were considered. As a result, the authors observed that the performance of the controller improves as the load decreases, which is a reasonable result due to the decreasing effect of the load dynamics on the ship dynamics when the load mass decreases.

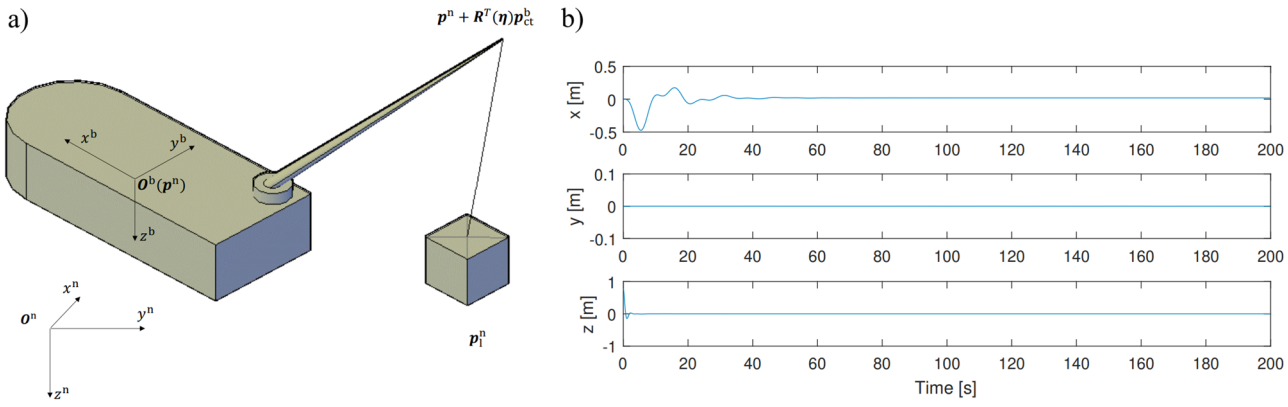


Fig. 1.25 Crane ship model (a) and the distance from the 1000 t load to the desired point (b) controlled using the method presented by Ye et al. [31]

Sun et al. [32] developed an anti-swing control scheme for underactuated cranes subject to unmodeled uncertainties. They introduced a sliding-modelike crane control method based on a newly developed sliding surface and control algorithm and performed a series of hardware experiments (Fig. 1.26) to verify the performance. They compared the developed method with the LQR (linear quadratic regulator) control method, the gantry kinetic energy coupling (GKEC) control law, the end-effector motion-based controller, and the SMC (sliding-mode control) method. As a result, authors stated that their nonlinear method eliminates the payload swing much better than others and can achieve superior control performance.

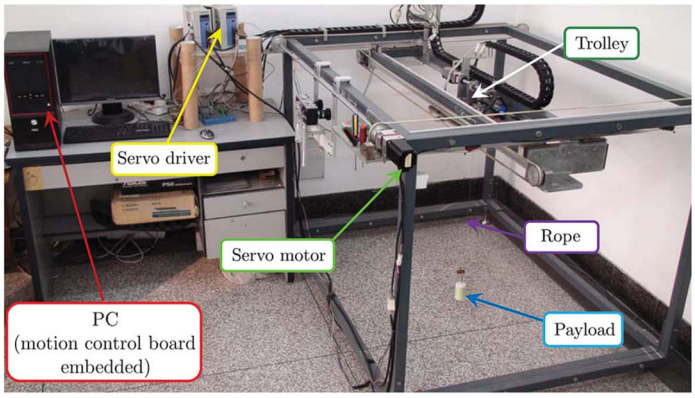


Fig. 1.26 Setup of the experiment conducted by Sun et al. [32]

Wang et al. [33] considered the 2DOF anti-swing control method for underactuated double-pendulum maritime jib cranes to reduce the swing of payload. They designed a precise numerical model (Fig. 1.27, left) of the anti-swing mechanism in MATLAB/Simulink environment and studied the influence of several model parameters on swing suspension. Also, they constructed the physical model of this mechanism (Fig. 1.27, right) and performed several experiments to validate the dynamical model. When comparing these two models' authors obtained good convergence. But in addition, Wang et al. mentioned that there was a little discrepancy between acquired data values of swing angles and the theoretical values. They captured this issue by measuring the swing with a high-frequency camera.

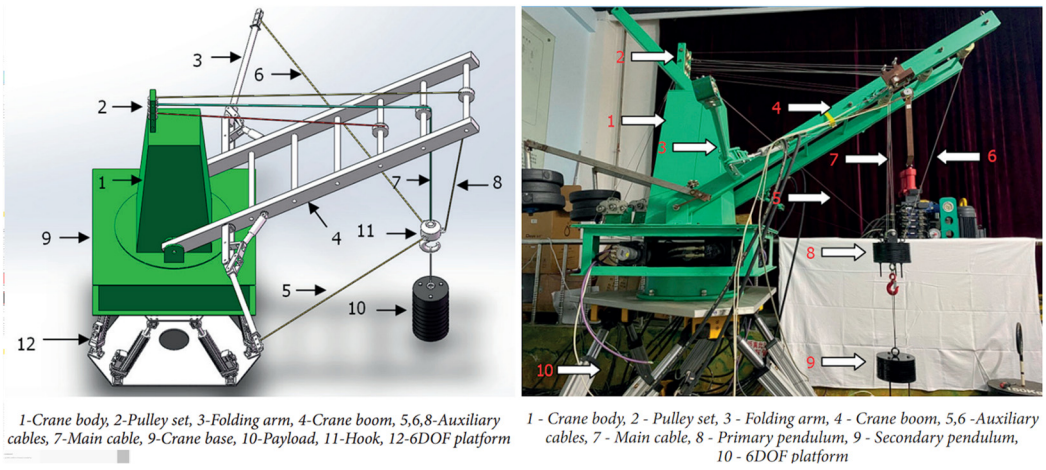


Fig. 1.27 The numerical model of double-pendulum anti-swing device (left) and experimental setup (right) designed and constructed by Wang et al. [33]

Chen et al. [34] constructed the wave compensation system using a permanent magnet synchronous machine (PMSM) as the actuator and fractional-order PID (FOPID) as a controller for the position loop. FOPID parameters were optimized by genetic algorithm-particle swarm optimization (GAPSO) algorithm and final verification of anti-swing mechanism performance was conducted in MATLAB/Simulink environment.

Another interesting mechanism for swing compensation was proposed by Sun et al. [35]. They developed the dynamical model (Fig. 1.28) of a triple-tagline anti-swing system (TTAS) with the possibility to represent motions under irregular waves. This mechanism is to some extent looks similar to the one studied by Wang et al. [33]. They simulated that model in MATLAB/Simulink, and reported that under irregular environment excitation, the average amplitude of the in-plane angle was reduced by 63%, and that of the out-of-plane angle was reduced by 82% using TTAS. Also, the two-dimensional trajectory of the payload was reduced by 92% [35].

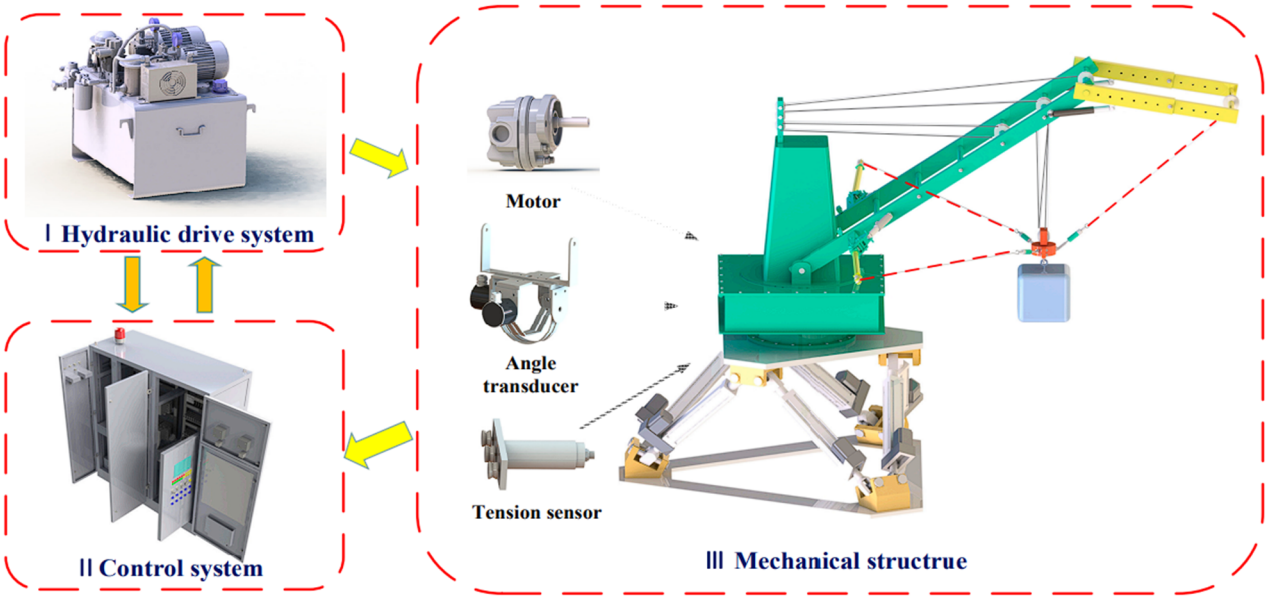


Fig. 1.28 Schematic diagram of the TTAS in Sun et al. [35]

1.4. Section conclusions

In this section, a review was conducted in three key research areas related to the topic of this dissertation. The first area pertains to methods describing the dynamics of floating cranes and floating objects in general. As widely known, the dynamic response of floating bodies in maritime conditions is frequency-dependent, requiring calculations not only in the time domain but also in the frequency domain. The de facto standard for the equations of motion for floating bodies is Cummins' equation, which takes into account various hydrodynamic and hydrostatic characteristics of the vessel, as well as the fluid memory effect. Recently, specialized software has been developed for modeling loading operations performed by floating cranes, although not all of them allow for the construction of a comprehensive numerical model. A common situation is the need to calculate hydrodynamic

parameters in third-party software. Korean researchers have made a significant contribution to this field among the works reviewed in this regard. A comparison of the software used by other authors in related is presented in Table 1.1.

Table 1.1 The software used in the related studies

Study	Calculation and implementation of		
	Hydrodynamic coefficients	Hydrodynamic forces and MBD	Control
Cha et al. [6]	–	In-house (self-developed)	In-house, scenario-based (pre-defined) control
Cha et al. [9]	Commercial software	In-house (Cha et al. [6])	–
Ku and Ha [10] and Ha et al. [11]	–	In-house (self-developed)	In-house, scenario-based
Hong et al. [12]	WADAM	In-house (Cha et al. [6])	–
Jeong et al. [23]	Commercial software	In-house (Ha et al. [11])	In-house, scenario-based
Viswanathan et al. [24]	ANSYS AQWA	OrcaFlex coupled with SimulationX via API	SimulationX, scenario-based, slow computation via API
Landsverk et al. [25]	VERES/ShipX	MSS MATLAB toolbox [26]	MATLAB, scenario-based
Vorhölter et al. [27]	E4	E4	E4, scenario-based
Li et al. [28]	–	In-house (self-developed)	In-house, VR interactive collaborative control
Chu et. al [30]	–	20-Sim	20-Sim, real-time interactive control with anti-sway system

The second part of the section is focused on reviewing methods for determining hydrodynamic loads on floating bodies. Two fundamental equations of computational hydrodynamics are prominently featured today – the Navier-Stokes equation and the Bernoulli equation. The former comprehensively describes all nonlinear processes that can occur in a fluid flow, including viscosity and turbulence. However, modeling based on the Navier-Stokes equations is extremely computationally intensive. Therefore, the second approach (the Bernoulli equation) is more commonly applied in maritime engineering for simulating the dynamics of vessels in the time domain. This method approximates the flow as inviscid and non-turbulent but allows for very fast determination of flow velocity at the boundary of the floating structure. Among the numerical modeling methods within the framework of potential flow theory, the panel method has gained popularity. In this method, the complex geometry of the submerged part can be divided into finite-size elements, and the boundary problem can be solved.

The third part of the section was dedicated to reviewing research focused on modeling loading operations and control systems for load swinging. This relevant area has been actively developing in recent times. The standard approach for controlling load movements in models is pre-defined

(scenario-based) control, involving pre-specifying crane part movements before initializing the model. Among advanced digital solutions, researchers are developing sophisticated interactive models of crane ships that allow control of the model using manipulators and incorporate PID controllers to compensate for oscillations. There are also several works on developing physical and numerical models of new mechanisms for anti-sway systems.

Thus, the problem of dynamic analysis of crane ships is multi-component and interdisciplinary, requiring a comprehensive approach. While computational hydrodynamics methods for floating bodies are well-established and undergo slow modifications, interactive modeling methods and load sway control methods actively acquire new solutions. This underscores the relevance of the goals and tasks outlined in this dissertation.

2. Development of the numerical modeling method for multibody “vessel-cargo” system

2.1. Problem formulation

A crane vessel and suspended load is a mechanical system consisting of several structural elements and joints (Fig. 2.1). The initial stage of the research in this dissertation involves developing an approach to construct a numerical model of the “ship-load” system, taking into account the dynamic characteristics of such a system and the interaction of its components. In line with the thesis stated goal, the development of this approach is focused on achieving an optimal balance between computation speed and accuracy. This will enable linking the model to the physical object in real-time and effectively handling the dynamics of the crane ship and its components under various environmental conditions (waves, wind, currents, etc.).

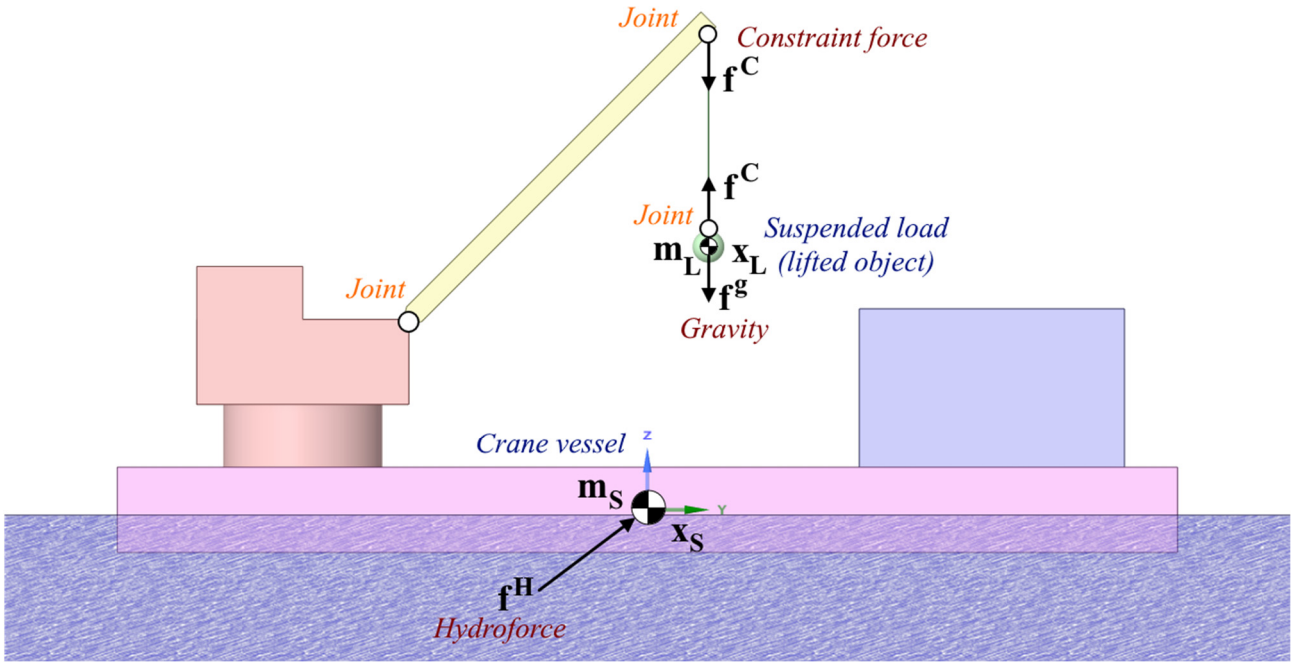


Fig. 2.1 Crane vessel as multibody system [37]

The goal of this thesis is to develop a method of multiphysics interactive numerical simulation of coupled dynamics of the “vessel-cargo” system with anti-sway control of the payload using artificial intelligence.

In order to achieve the set goal, the following tasks should be solved:

To achieve the stated goal, the problem can be decomposed into the following tasks:

- Ship dynamics modeling: Develop a mathematical model describing the motion of the crane ship under the influence of external forces.
- Suspended load modeling: Create a numerical model component that simulates the cable and suspended load, taking into account its mass and inertia.
- Environmental impact consideration: Integrate into the model the effects of hydrodynamic forces on the crane ship from the marine environment.

- Assessment of modeling method reliability: Evaluate the effectiveness and accuracy of the developed numerical model, conducting an analysis of the influence of various parameters on the system's dynamics.
- Development of control method: Devise an approach for real-time control of the crane ship parts, ensuring stability and accuracy during lifting operations.
- Development of load swing (sway) control system: Investigate and apply methods to reduce oscillations in the suspended load on the floating crane.

In accordance with the previously mentioned objectives, tasks, and the intended application area of the developed methods, this study introduces a set of assumptions. These assumptions not only contribute to solving the defined tasks but also make the developed approaches practically applicable in engineering practice. Here, a quote from Box and Draper (1987) [38] can be cited, stating: “Essentially, all models are wrong, but some are useful.”

Assumptions made in this study are as follows:

1. **The structural elements of the floating crane are considered as rigid bodies** (except for the cable). While the literature review conducted earlier includes several works by other authors taking into account the stiffness of crane components, deformations are relatively small. Therefore, this dissertation does not consider the possible flexibility of the crane's boom and other elements. This will significantly speed up calculations and avoid entering the realm of structural analysis when formulating governing equations.
2. **When determining loads, the condition of an incompressible and non-turbulent flow is assumed.** Thus, the potential flow theory described in Section 1 is implemented, and hydrodynamic loads are determined in accordance with the Cummins equation (1.6). This assumption is also introduced to extend the applicability of the research results and increase calculation speed. Although hydrodynamics based on the Navier-Stokes theory undoubtedly has greater accuracy, its application in real engineering practice is highly debatable due to significant computational costs. The method of hydrodynamic coefficients, calculated using BEM solvers (panel method), allows accounting for a variety of sea states and avoiding the solution of complex differential equations in partial derivatives when calculating dynamics. At the same time, there is still the possibility of independently considering the viscosity of the fluid through viscosity coefficients.
3. **Hydrodynamic coefficients do not depend on the velocity of the vessel.** Conducting maritime operations implies the absence of significant relative speeds of the vessel, especially when dynamic positioning systems are in place. Accordingly, it is permissible to disregard changes in the hydrodynamic characteristics of the "vessel-cargo" system with changes in the vessel's speed.

4. **Second-order wave drift forces are not considered in detail.** As known, the contribution of hydrodynamic forces of the second and higher orders can also be significant. However, the scope of the study pertains to the operation of crane vessels. While the research contributes to expanding the permissible parameters of the external environment, maritime operations are still conducted under conditions of relatively low sea wave heights. Under such circumstances, the contribution of second-order wave forces to the overall impact on the floating body is quite negligible, accounting for less than 10% (closer to 1%) [36].
5. **Investigations into the methodology of modeling the dynamics of a crane ship and suspended load in the time domain are not conducted at the real scale.** This assumption is related to the fact that the reliability and accuracy of the dynamic modeling method in this dissertation are assessed by comparing the simulation results with data from laboratory experiments. The laboratory experiments, described in Section 2.2, involved testing an experimental model of a floating crane in a wave basin, built to resemble a real vessel at a 1:50 scale. Undoubtedly, the transition from an intermediate scale to the full scale is not linearly proportional. However, this assumption does not limit the applicability, as the described approach can be applied at the actual scale by creating a CAD model with corresponding dimensions. At the same time, this allows for the assessment and verification of the numerical model parameters by comparing the results with laboratory experiments in the wave basin.
6. **Only hydrodynamic forces from the water environment are considered among external forces.**

2.2. Experimental studies

2.2.1. Experiment setup

To make the development of the numerical method valid and feasible, numerical experiments must be accompanied by experiments replicating the real physics of the phenomenon. This dissertation is a joint research effort with the Naruo Technical Research Institute, a subdivision in the Civil Construction Division of TOYO CONSTRUCTION CO., LTD (Japan), engaged in research and development in the field of marine construction and geotechnics. The Naruo Technical Research Institute, located near the city of Kobe in the Kansai region, has a wave basin facility [39], allowing for medium-scale experiments (Fig. 2.2 and Fig. 2.3). The basin has dimensions of 30x19x1.5 m and is equipped with two different wave makers capable of producing regular, irregular waves, and tsunami waves.

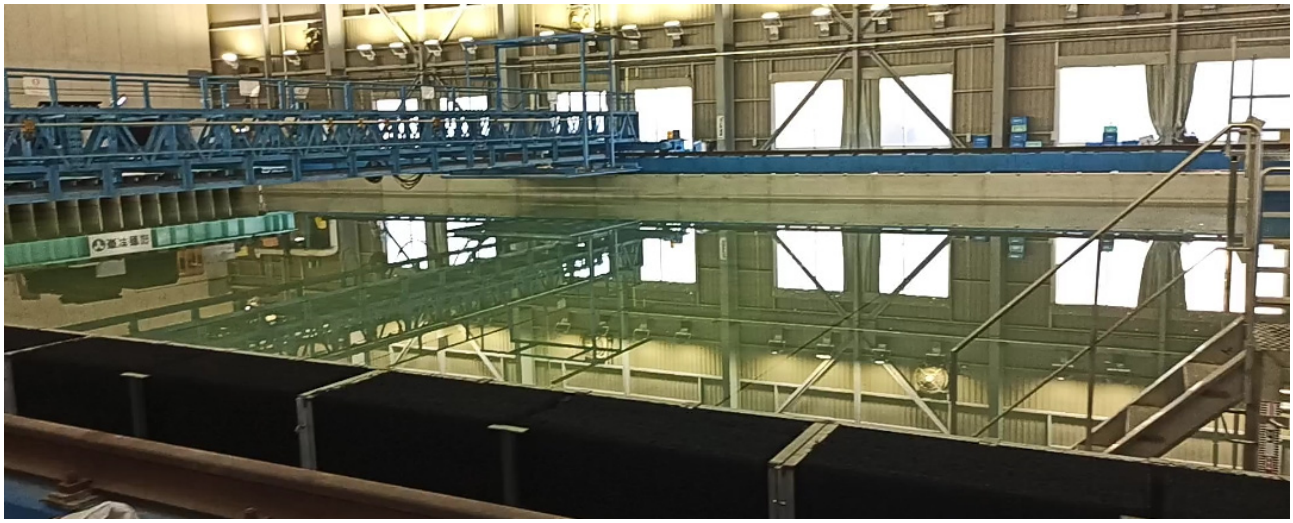


Fig. 2.2 Wave basin at Naruo Technical Research Institute

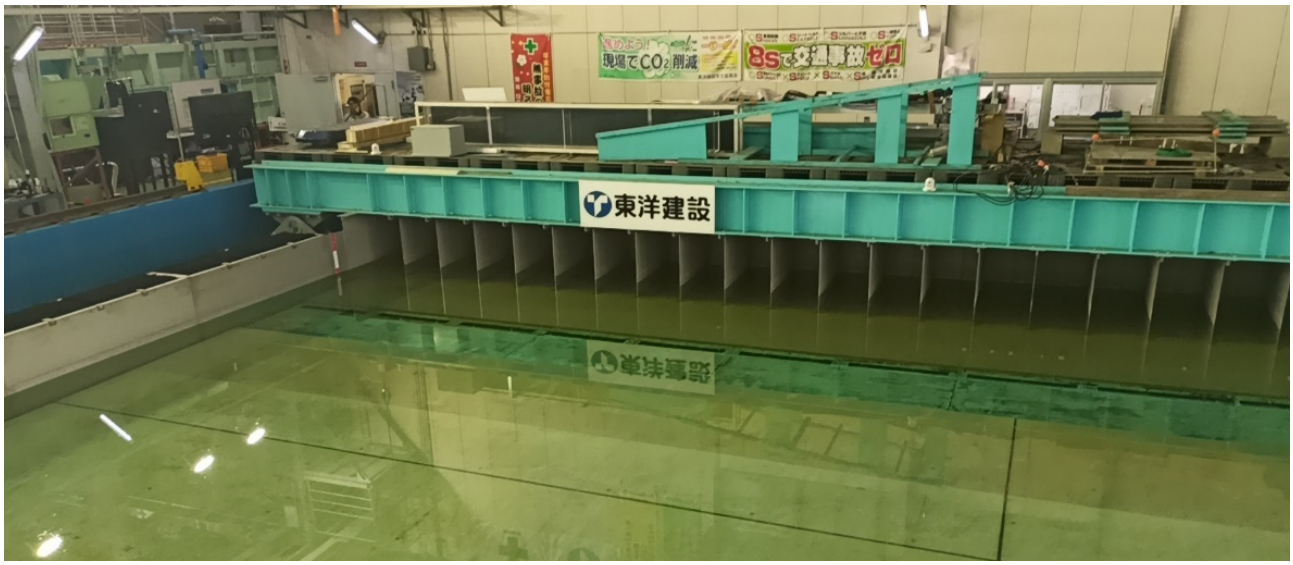


Fig. 2.3 View of the wave makers in the wave basin

The capabilities of the wave basin were used in this thesis to verify the reliability of the developed numerical method in simulating the hydrodynamics of the crane vessel. For this purpose, a series of experiments were conducted by the specialists of Naruo Technical Research Institute. The experiments aimed to measure the wave-induced motions of the crane ship model, which is a scaled-down reproduction of a real crane vessel at a 1:50 scale. The physical model of the crane vessel installed in the wave basin is shown in Fig. 2.4 and Fig. 2.5. A schematic representation of the model with dimensions is provided in Fig. 2.6.

The crane model comprises the following structural elements: hull, bridge, crane base, crane boom, and an additional lead weight required for balancing the model in a horizontal position. The length of the hull (floating part) is 1.77 m, the width of the hull is 0.54 m, and the waterplane area is 0.861 m² (measured further by CAD software). The total weight of the model is approximately 44.15 kg (corresponding to 5519 tons in full scale). Special markers (orange color) were installed on the ship model to track the motions – six on the hull and one on the crane boom. The suspended load of 0.16 kg (corresponds to 20 tons in full scale) also served as a marker.

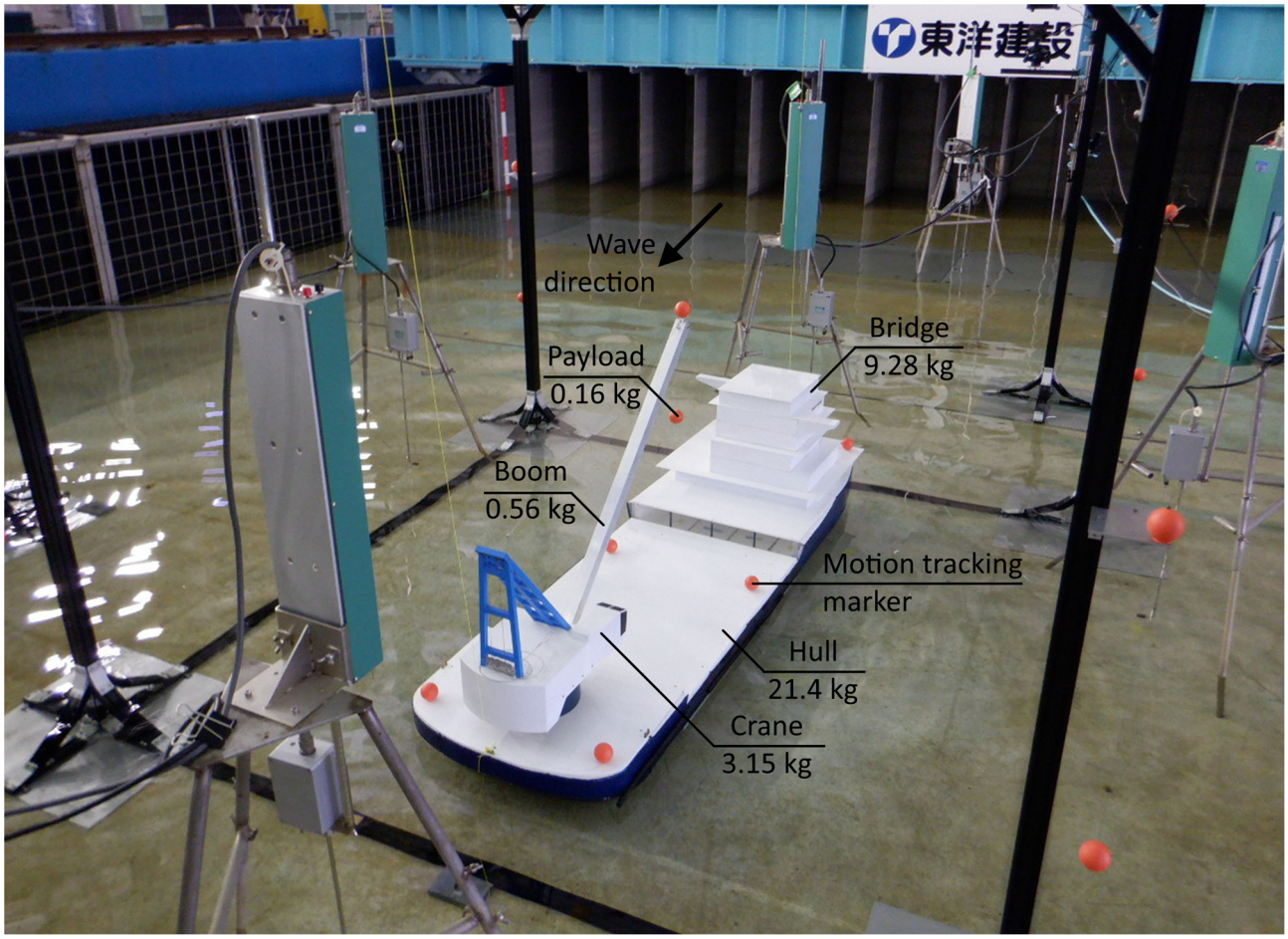


Fig. 2.4 Crane ship model, front view

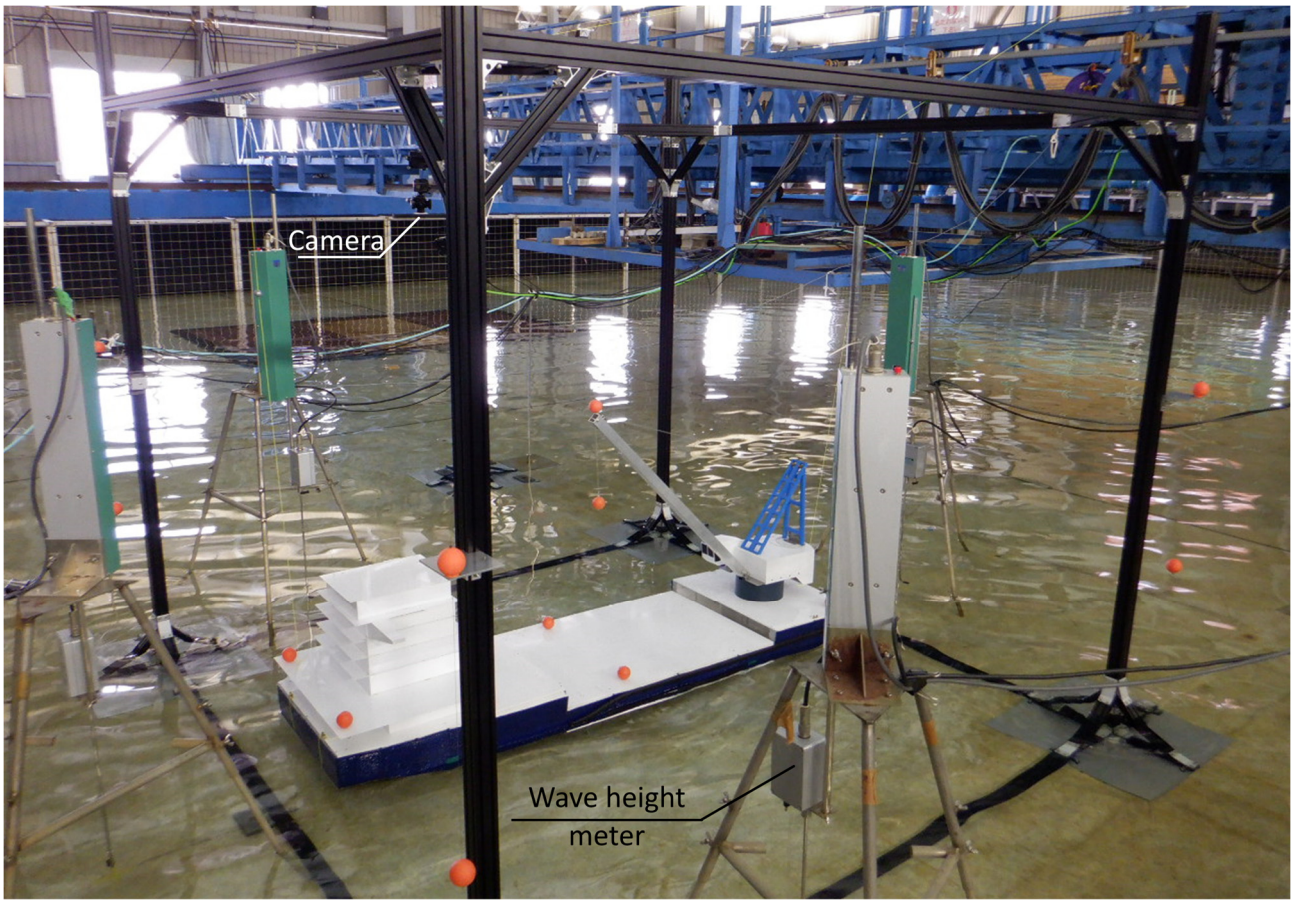


Fig. 2.5 Crane ship model, side view

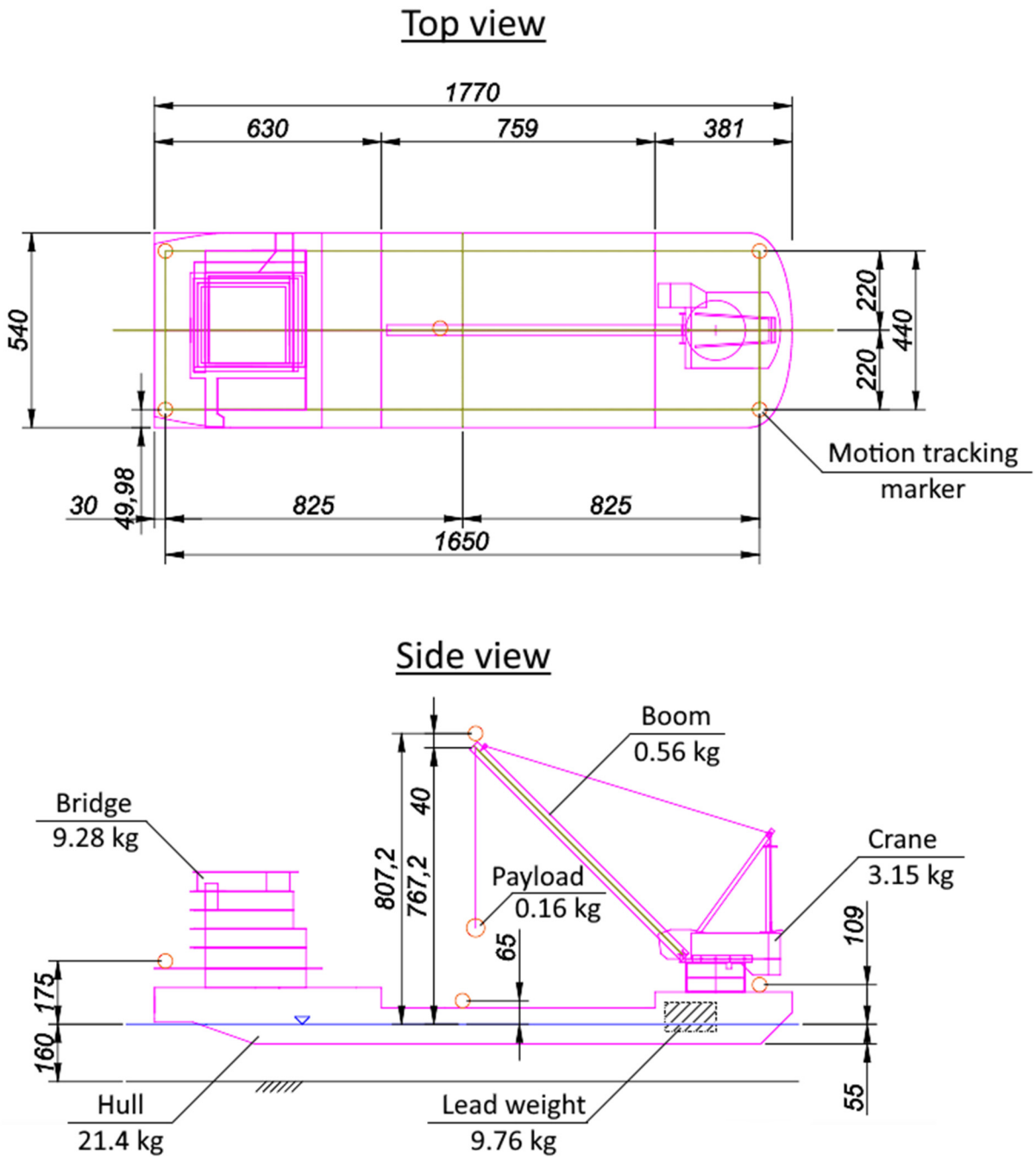


Fig. 2.6 Schematic drawing of crane ship model

When planning the experiment, the model parameters X_m were determined through the scaling factor α:

$$X_m = \alpha X_a \tag{2.1}$$

where X_a is the parameter of actual crane ship.

The scale factor of geometric dimensions was taken equal to $\alpha = 1/50 = 0.02$, the scale factor of mass $\alpha = 1/50^3 = 0.000008$ (since mass has an inverse cubic dependence on dimensions at density $\rho = [kg]/[m^3]$). The correspondence between the parameters of the real ship and the physical model in the experiment is given in Table 2.1.

Table 2.1 The correspondence between the parameters of the real ship and the physical model

Parameter	Actual ship (1:1)	Model (1:50)
Draft (m)	2.76	0.0552
Total mass calculated from draft (kg)	5518750	44.15
Ship's hull with lead weight	unknown	31.16
Bridge (kg)	unknown	9.28
Crane base mass (kg)	unknown	3.15
Crane boom mass (kg)	70040	0.56
Crane boom length (m)	40.6	0.812
Suspended load mass (kg)	20000	0.16

The displacements of each marker were tracked by two GoPro HERO7 Black cameras installed near the test area and recorded with special software. The test area represented an equipped space in the wave basin near the wave makers, with dimensions of approximately 2.5x2.5 m (Fig. 2.7). Additionally, in the test area, KENEK wave height meters (fast response type) with measurement accuracy 0.5% were installed. The wave height meters were mounted on tripods to measure the free surface elevation at different positions. Two wave height meters were installed near the wave makers (WL-OFF1 and WL-OFF2 in Fig. 2.7), and four around the crane ship (WL1, WL2, WL3, and WL4 in Fig. 2.7). The distance from the wave makers to the geometric center of the ship's hull was 6.45 m. The measuring equipment and a screenshot from the software for recording displacements are shown in Fig. 2.8 and Fig. 2.9.

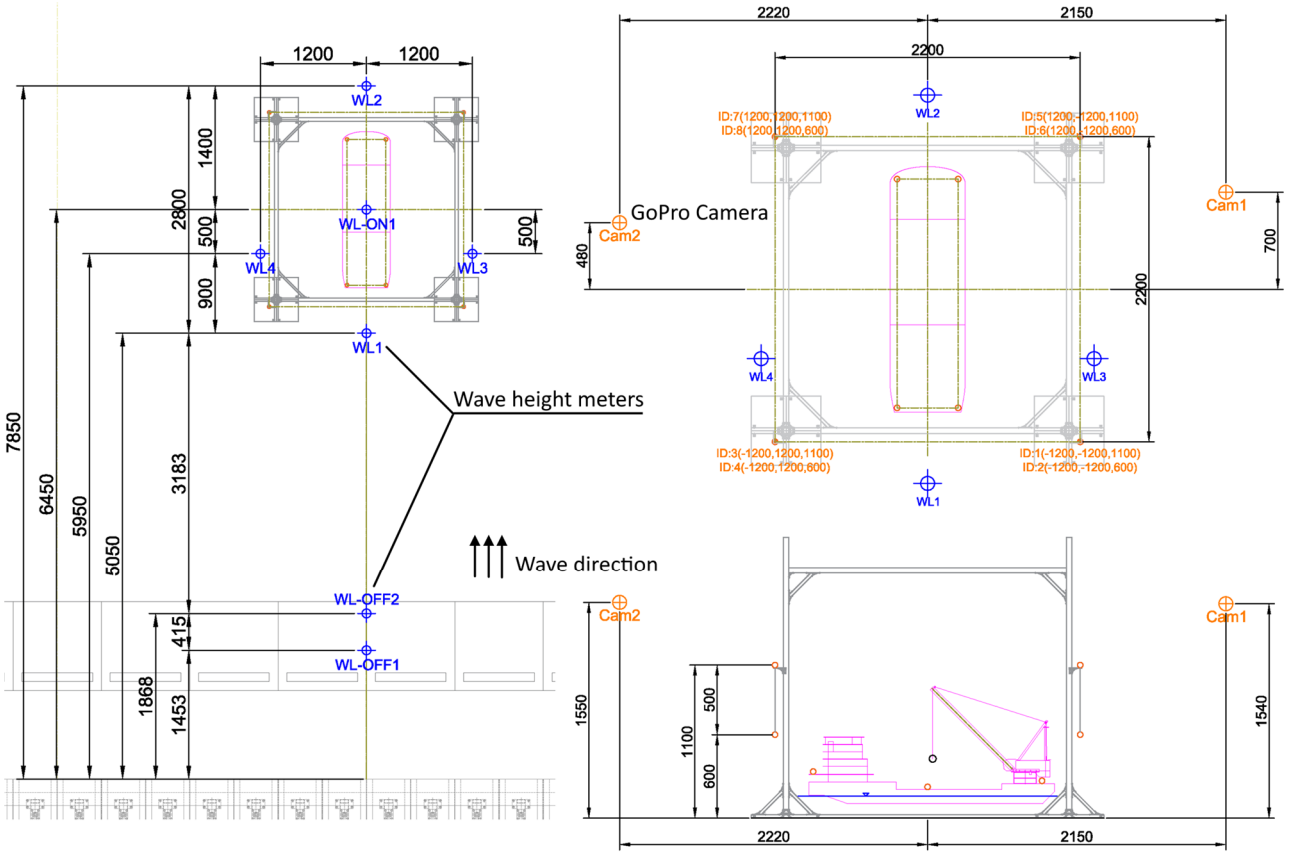


Fig. 2.7 Schematic drawing of experimental setup (test area)

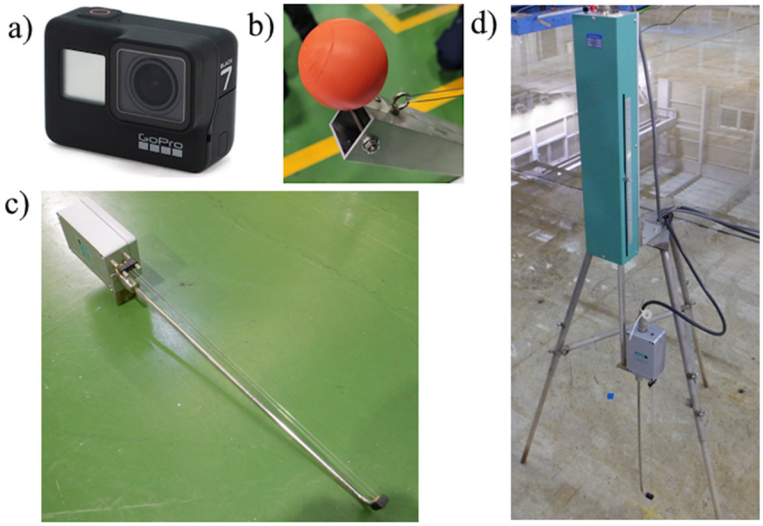


Fig. 2.8 Measurement equipment: a) GoPro HERO7 Black camera; b) Motion tracking marker; c) KENEK wave height meter; d) Wave height meter on tripod

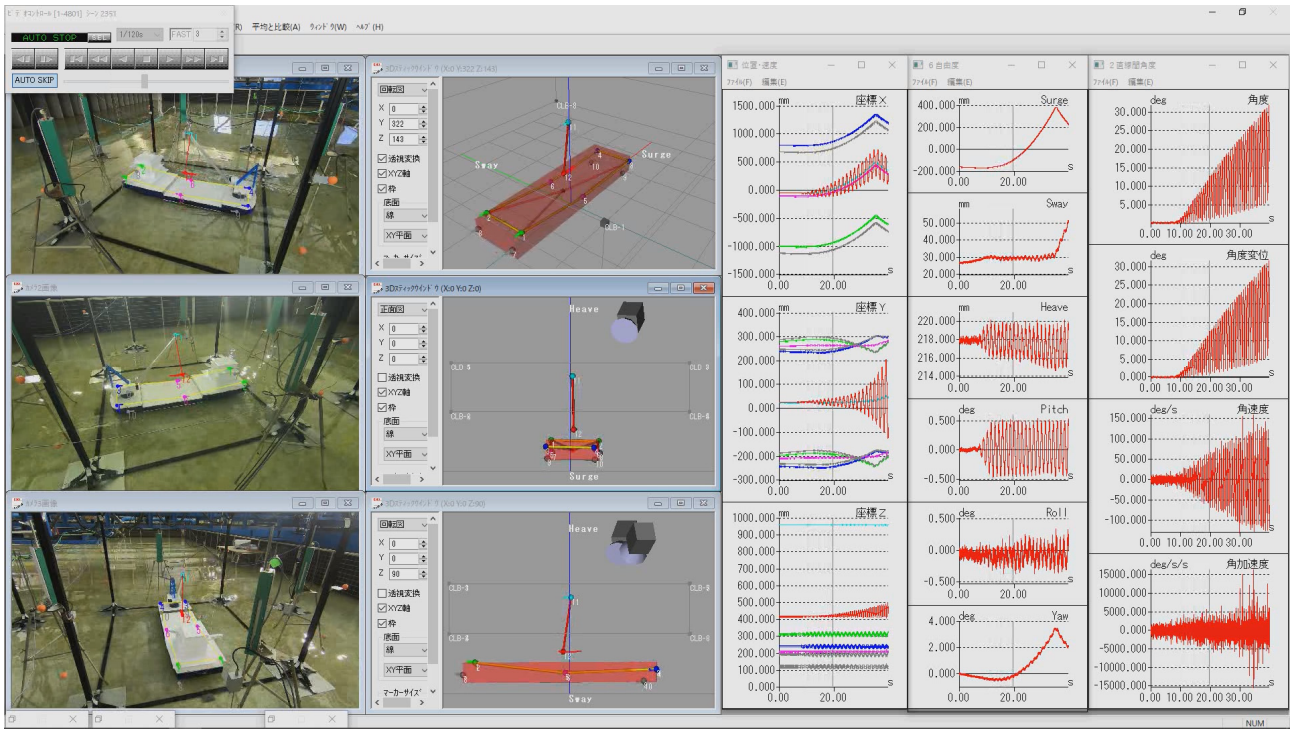


Fig. 2.9 Software used for motion tracking

The experimental conditions implied the action of a regular wave with a height of 0.02 m and a period of 1.4 s with heading angle of 0 degrees. The experimental series included measuring the displacements of markers (including the suspended load) at different cable lengths. The planned distances from the pendulum's "joint" to the center of the suspended load were 0.125 m (cable length 103.5 mm), 0.25 m (cable length 228.5 mm), and 0.5 m (cable length 478.5 mm). However, during the experiment, it was found that the first two situations were unrepresentative because, during swinging, the cable came into contact with an opening in the crane boom, through which the cable passed. Therefore, the suitable option for use was the case with a distance to the load of 0.5 m. Nevertheless, this case is the most interesting in terms of verification since in such a scenario the wave frequency corresponds to the natural frequency of the suspended load (approximately 0.7 Hz).

The duration of the pure experiment was approximately 30 seconds. After this time, due to surge movements of the crane ship, there was tension in the positioning ropes, which restricted further movement. This duration is more than sufficient for a qualitative verification of subsequent numerical experiments.

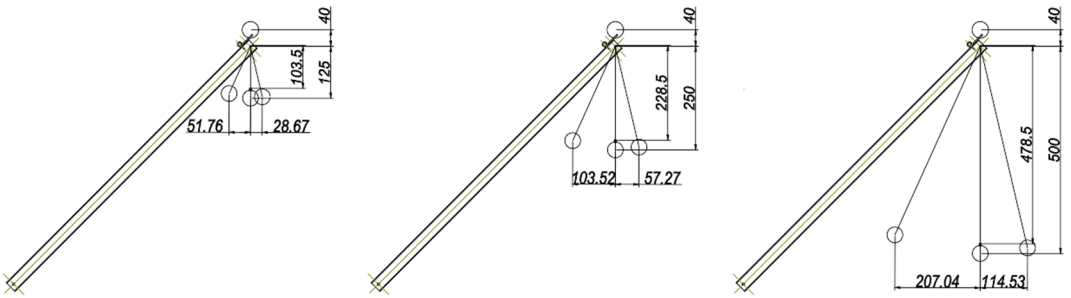


Fig. 2.10 Experimental cases with different cable length (mm)

2.2.2. Free surface elevation

Wave height measurements were taken at 6 points (Fig. 2.7) – 2 near the wave makers and 4 around the crane ship. The sampling rate during the experiment was 50 Hz, so 70 measurements were made per wave period. The measurement results are presented in Fig. 2.11-2.13.

Fig. 2.11 presents the results of measurements from wave height meters WL-OFF1 and WL-OFF2, which were installed near the wave generators. Dotted lines represent the average values of the lower and peak wave points across all extrema after 6 seconds. Overall, it can be stated that the condition of a regular wave was observed in the area of its generation. At the same time, the wave height was more stable (better preserved its amplitude) at point WL-OFF2, which can be explained by the stronger influence of the wave maker in the WL-OFF1 position. The average wave height (the difference between the values of the dashed lines) was 19.5 mm and 21.5 mm for WL-OFF1 and WL-OFF2, respectively, with measurement accuracy of ±0.1 mm (0.5% according to the equipment specification).

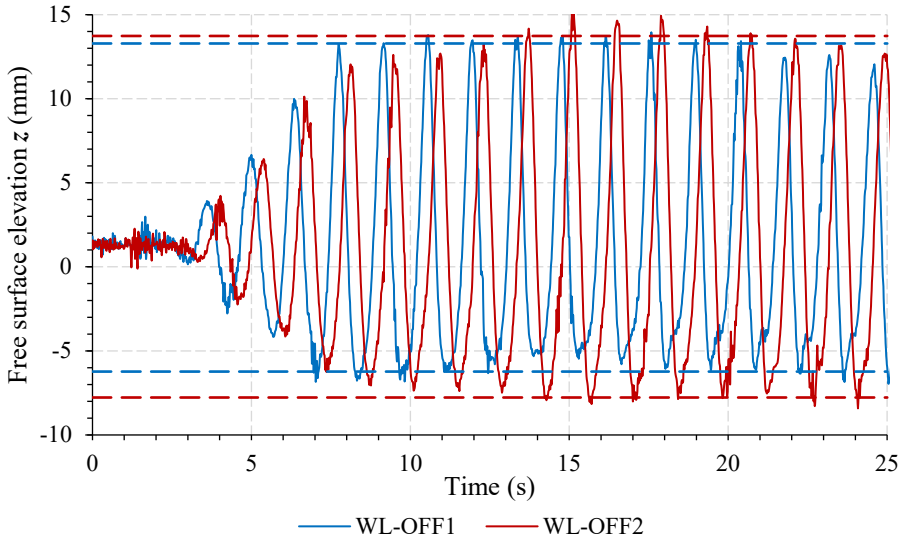


Fig. 2.11 Free surface elevation measured at positions WL-OFF1 and WL-OFF2

Fig. 2.12 illustrate the results of measurements from wave height meters WL1 and WL2, which were installed in the direction of wave propagation in front and behind the model of the crane ship. In this case, an ambiguous picture is observed. At the beginning of the experiment, the incoming wave at point WL1 was regular with an increasing amplitude. However, after 12 seconds of the experiment, additional local peaks in the free surface elevation appear. Obviously, this is explained by the reflection of the wave from the hull of the ship and its superposition on the main incoming component. Moreover, after 15 seconds, a gradual decrease in amplitude (wave height) at point WL1 is observed. Dashed lines in Fig. 2.12 similarly represent the average values of the lower and peak wave points at point WL1 across all extrema after 10 seconds. Despite the apparent unevenness of the wave height, the average value is 18.7 mm, which is close to the planned wave height of 20 mm. As for the position of WL2, a regular wave of approximately equal height is observed throughout the experiment. However, due to wave attenuation by the ship's hull, the average height at this point is about 12 mm.

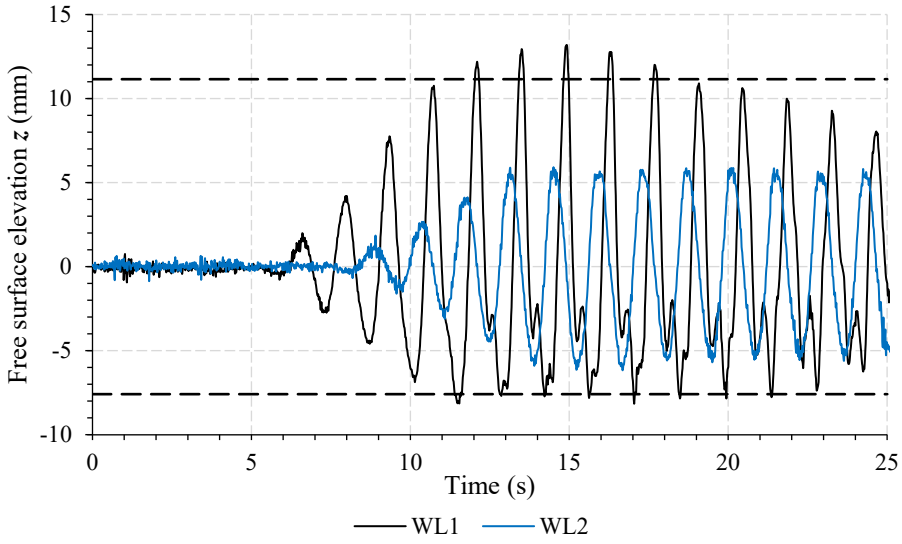


Fig. 2.12 Free surface elevation measured at positions WL1 and WL2

Fig. 2.13 presents the results of measurements from wave height meters WL3 and WL4, which were installed on the lateral sides of the crane ship model, where a mismatch in the average wave height in these points is observed. However, the minimum values of the wave elevation are almost identical. The most probable reason for such a result is the imperfect alignment of the wave propagation vector with the planned direction in the experiment. Another possible cause could be the uneven reflection of the wave from the ship model in the direction of these wave height meters, although the crane ship model was almost symmetrical, making this scenario less likely. The average values of the minima and maxima of the wave elevation at these points are also shown on Fig. 2.13 as dashed lines. The average wave height when averaging over points WL3 and WL4 was 19 mm.

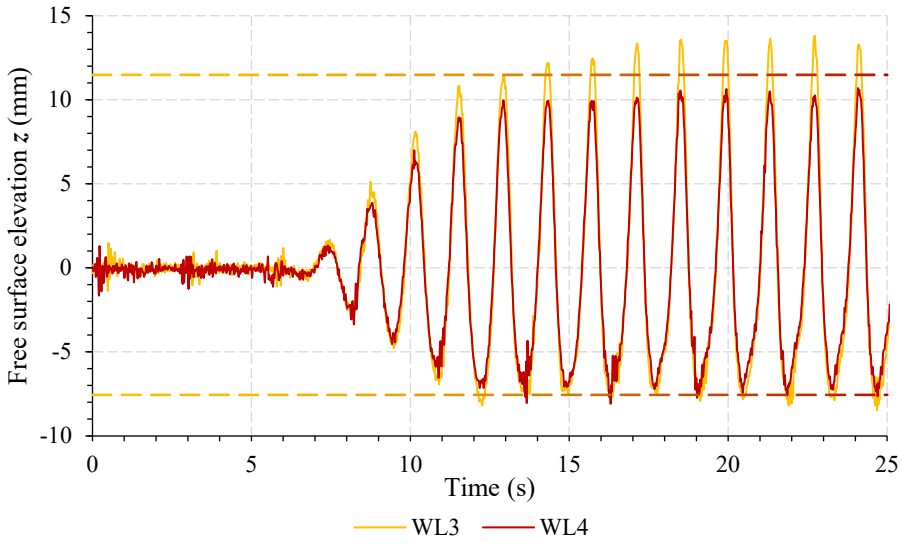


Fig. 2.13 Free surface elevation measured at positions WL3 and WL4

Analyzing the above-mentioned results, it can be concluded that the hull of the crane ship was subjected to the influence of almost regular waves throughout the experiment. Considering the measurements at points WL2, WL3, and WL4, local peaks at point WL1 can be disregarded since they arise due to the reflection of the wave from the model and correspond to the movement of the wave in the opposite direction to the incoming wave.

2.2.3. Crane vessel response

The displacements of the crane ship were measured at 12 points using motion tracking markers. For precise positioning of the ship relative to the global coordinate system, an additional 8 markers were installed on the frame, as illustrated in Fig. 2.5. As a result, displacement, velocity, and acceleration histories were obtained at 12 points, including the suspended load on the cable. The sampling rate was 0.008 s. Since deformations of structural elements can be neglected, for the convenience of using the experimental results, these displacements were recalculated into rigid body motions with 6 degrees of freedom relative to the center of gravity of the ship's hull. The displacement data for the first 25 seconds of the experiment are presented in Fig. 2.14.

Analyzing the measurement results, it can be stated that the impact of the incoming wave on the physical model of the crane ship began approximately at 6.5 seconds into the experiment. Before the direct wave impact, there are small free movements of the ship. There is a slight roll of the ship due to the asymmetry of the upper parts of the model. As a result of the wave impact, the ship shows a nearly regular response. However, due to the asymmetry and coupling of degrees of freedom, there is a displacement of the ship in the Sway and Yaw directions.

The displacements of the suspended load (marker 12) relative to the point where the cable is attached to the crane boom (point 11) are presented in Fig. 2.15. The largest amplitude of the load swing occurs predictably along the *x*-axis due to the pitch motions of the crane vessel. There are also minor displacements along the *y*-axis, which are approximately 10 times smaller in magnitude. The

displacements in all directions are continually increasing, as the natural frequency of the pendulum coincides with the frequency of the regular wave in the wave basin.

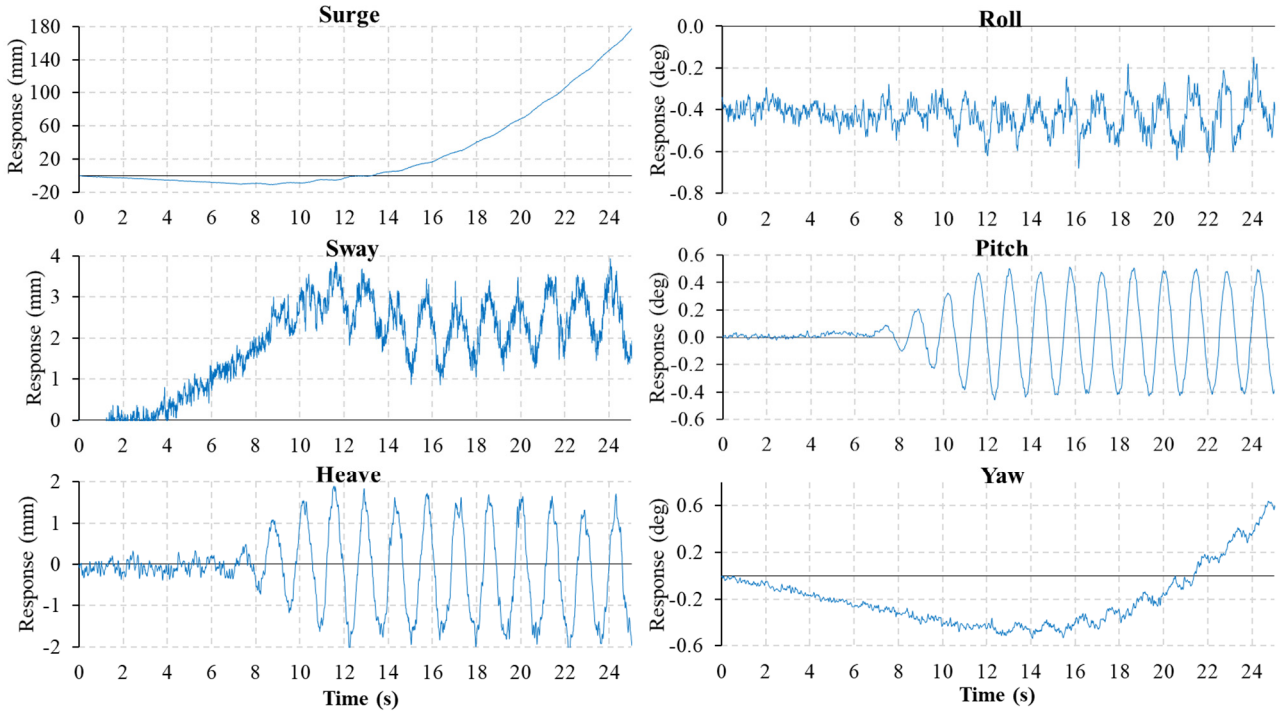


Fig. 2.14 Crane vessel hydrodynamic response

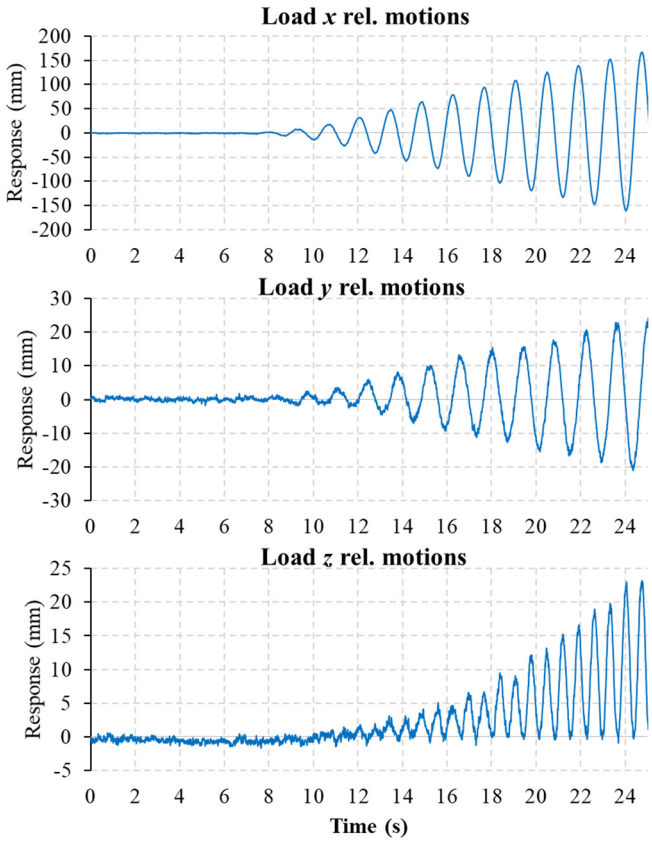


Fig. 2.15 Suspended load displacements relative to crane boom

In subsequent numerical experiments during the verification of the hydrodynamics modeling method of the crane vessel, the displacements of the numerical model (hull and suspended load) are compared precisely with the aforementioned results of laboratory experiments.

2.3. Numerical experiments

2.3.1. General description of the crane ship dynamics modeling method

In this subsection, a description of the basic principles of the applied approach to modeling the floating crane vessel in the time domain is provided. As mentioned earlier, this study is focused on several goals:

1. Reliability of modeling physics (accuracy);
2. Computational speed;
3. Capability to expand functionality and simulate various physical phenomena (multiphysics);
4. Minimization of reliance on multiple commercial solutions, a commitment to openness, and accessibility of the approach.

The decision has been made to use WEC-Sim software [40, 41] as a multibody dynamics solver in this thesis, following an assessment of available software options based on the principles mentioned above. WEC-Sim (Wave Energy Converter SIMulator) is an open-source software developed by National Renewable Energy Laboratory (NREL) and Sandia National Laboratories (Sandia) in MATLAB/Simulink using the multibody dynamics solver Simscape Multibody. WEC-Sim was created for simulating wave energy converters (WECs), but it also can be applied to any floating bodies since governing equations are the same.

The mention of WEC-Sim as "open-source" here requires clarification, as WEC-Sim is a library designed in the MATLAB/Simulink environment, providing custom blocks and functions. The library itself is free, but its usage is intended within the commercial MATLAB software. Therefore, users are required to have a MATLAB license and the Simscape Multibody solver. Nevertheless, the cost of this software is significantly lower compared to existing commercial solvers in the market, such as ANSYS AQWA (USA) [42], Tdyn SeaFEM (Spain) [43] and OrcaFlex (UK) [44]. Thomsen et al. [45] provided a comparison and description of the capabilities of commercial software designed for dynamic analysis of wave energy converters, which is also true for other types of floating structures.

WEC-Sim adheres to the conventional approach for defining hydrodynamic forces based on the Cummins equation and frequency-dependent coefficients. The simulation inputs consist of linear coefficients obtained from external frequency-domain potential flow Boundary Element Method (BEM) solvers using the BEMIO tool. Simulation parameters are manually defined using the "*wecSimInputFile.m*" file written in the MATLAB programming language. Subsequently, the calculation of hydrodynamic forces and the dynamics of the floating body occurs within the Multibody Dynamics (MBD) loop using the built-in MATLAB Simscape solver. Thus, a one-way weak-coupled time-domain hydrodynamic analysis can be performed (Fig. 2.16).

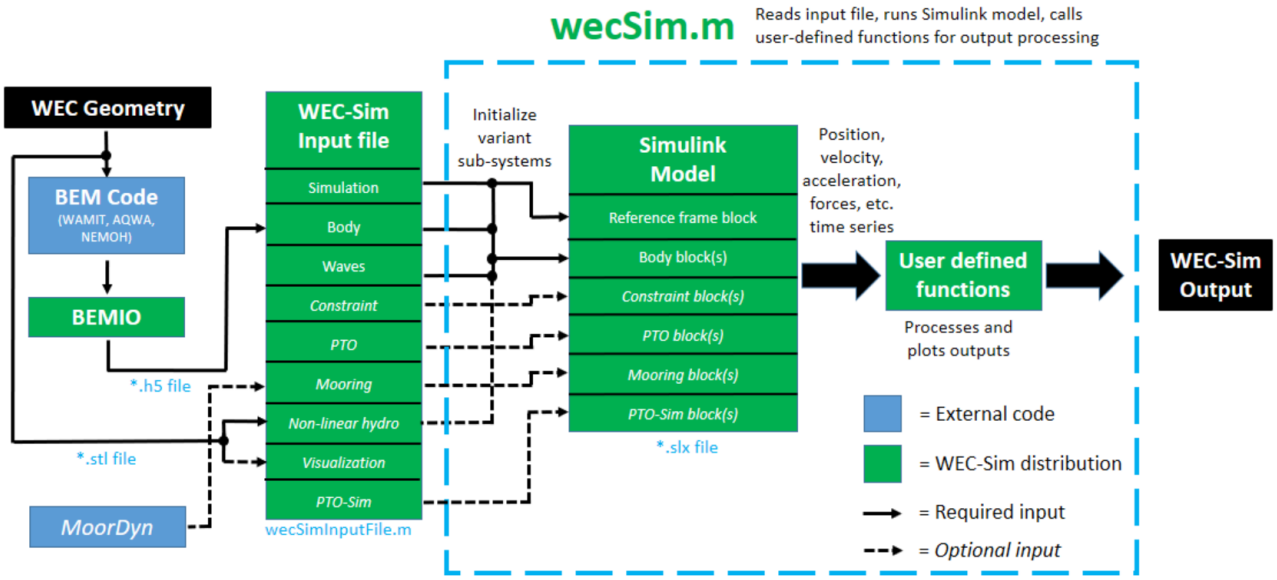


Fig. 2.16 Workflow of analysis using WEC-Sim [46]

In total, there are 3 main modeling stages involving multi-step processes:

I. Preparation stage

- modeling the geometry of the floating system: determination of mass and inertial properties of bodies, creation of a discretized model of the submerged part (hull);
- hardware assembly: assembling a PC for high-performance computing, connecting controllers, expansion boards and manipulators (for further interactive control discussed in Section 3);
- setting up the software part: installation of necessary software and libraries, development of a convenient file system for the project, setting up permissions and environmental variables.

II. Determination of hydrodynamic coefficients in the frequency domain:

- the creation of a model in BEM software (if necessary);
- configuration of the computation (solver selection, selection of output data format), definition of initial conditions (range of wave frequencies and directions, water depth, water density);
- performing the computation and analyzing the results.

III. Modeling the dynamics in the MBD solver in time domain:

- designing the model in MBD software (MATLAB/Simulink + Simscape + WEC-Sim): assigning properties of rigid bodies, import of the geometry, setting up the connections (joints), organizing the data output;
- development of the software part of the model: importing calculation results from BEM solver, setting parameters of WEC-Sim blocks and interactive control system (discussed in Section 3);

- modifying the model with necessary physics and additional features (discussed in Sections 3 and 4 of the thesis): setting up synchronization of PC hardware with the model (interactive control), developing new Simscape custom blocks, introducing artificial intelligence into the model.
- running the dynamic model of the floating vessel (including AI training discussed in Section 4);

The mentioned approach has the following significant advantages over other similar codes and corresponding methods:

- The end user is not limited by the functionality of the current software release. If desired and necessary for expansion (e.g., implementation of multi-physics calculations), extensive opportunities for user customization of the numerical model become available. Additionally, the integration of MATLAB with other third-party software can be implemented. There are also a large number of publicly available examples for solving relevant engineering problems using MATLAB.
- The weak coupling of dynamics and hydrodynamics adds convenience by eliminating the need to recalculate the frequency solution when changing the geometry and inertia of the above-water parts of the numerical model. Thus, it is possible to precompute hydrodynamic coefficients in external software for various configurations of the crane vessel, for example, in the case of modeling loading operations with different loads.
- Simscape offers a wide range of dynamic solvers, both explicit and implicit, allowing the selection of the most optimal option in terms of performance and convergence of calculations. It is reasonable to use a PC or workstation with a high-end processor with a large number of cores. It is also possible to use a high-performance GPU to speed up calculations.
- Full compatibility with MATLAB/Simulink libraries ensures easy development of complex engineering models, including real-time applications and AI-driven control.

WEC-Sim allows importing results from different Boundary Element Method (BEM) solvers such as AQWA, WAMIT, NEMOH, or Capytaine. The BEMIO tool facilitates the direct conversion of output data from the mentioned software into a format supported by WEC-Sim. AQWA [42] and WAMIT [47] are popular commercial software that is frequently used in solving hydrodynamics problems. NEMOH [48] and Capytaine [49] are open-source codes that also find their application in related research studies, also due to their accessibility. Based on the experience of using NEMOH, AQWA, and Capytaine, which is described in the Section 2.3.2, it was decided to use the Python-based software Capytaine to determine hydrodynamic coefficients due to a number of advantages.

To test the algorithm for constructing and computing the model, a simple model of arbitrary configuration was used, visually resembling the physical model in the conducted laboratory experiments (Fig. 2.17). BEM coefficients were calculated using the Capytaine software.

The first model building step is to create a geometry of the crane ship and calculate the hydrodynamic coefficients in the BEM solver. Geometric modeling of the floating crane can be performed in any CAD software (ANSYS SpaceClaim, SOLIDWORKS, Autodesk Fusion, etc.). It is necessary that the selected CAD software has the capability to export the geometry in the formats required for further calculations in Capytaine and MATLAB solvers. Since WEC-Sim has a modular structure, crane ship model can be assembled from various blocks, with each block representing an individual part of the ship (hull, bridge, crane, boom). In addition, it is sufficient to consider only the hull in the frequency-domain calculation with the coordinates of the center of gravity (CoG) of exactly this part and not the whole vessel. WEC-Sim uses STL format, which also can be processed by Capytaine. The geometry of structural parts of the ship were created in this file format and are presented in Fig. 2.18. Parameters of present BEM problem are presented in Table 2.2.

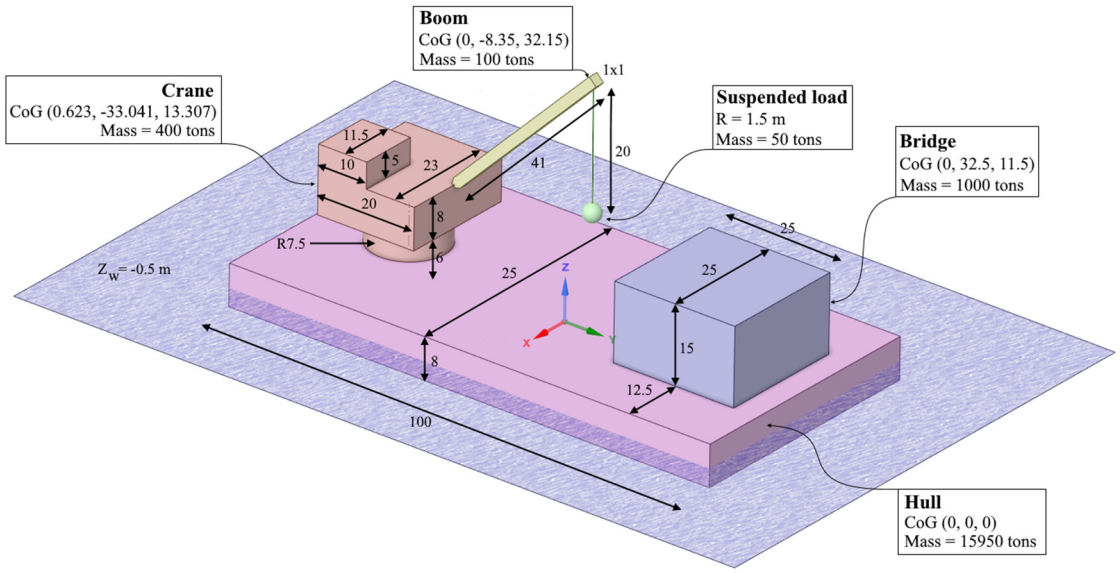


Fig. 2.17 Dimensions (m) and mass properties of the crane ship model [37]

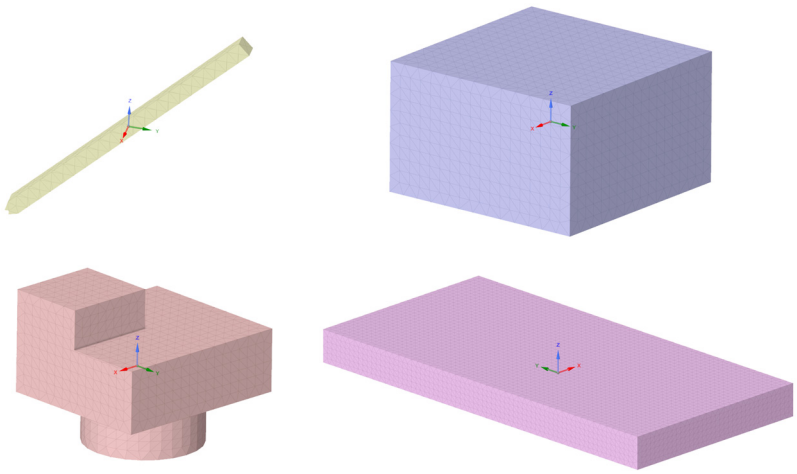


Fig. 2.18 STL models of structural elements of crane ship [37]

Table 2.2 Simulation conditions

Parameter	Units	Value
Gravity acceleration g	m/s^2	9.80665
Water density ρ	kg/m^3	1000
Water depth h	m	100
Wave period T	s	2 ~ 4 (50 frequencies)
Wave direction	deg	0 (x axis)
Crane ship draft d	m	3.5

To use previously calculated hydrodynamic coefficients and other floating body parameters, the Capytaine output files must be converted to a structured *.h5* hydrodata file for further use in time-domain analysis. This is accomplished using the BEMIO tool available in WEC-Sim. To convert the results of Capytaine calculation, the code in MATLAB language is executed. It reads Capytaine output files and writes them in the required format. Also, among the functions of BEMIO there are various commands that allow to perform different operations with hydrodata. The following script reads the Capytaine results, then calculates the normalized radiation and excitation impulse response functions and writes the data in *.h5* format:

```
hydro = struct();
hydro = readCAPYTAINE(hydro,'.\ship_Capytaine.nc');
hydro = radiationIRF(hydro,60,6001,[],[],[]);
hydro = excitationIRF(hydro,60,6001,[],[],[]);
writeBEMIOH5(hydro)
```

As an example, the normalized added mass $A(\omega)$ for the Surge, Heave, and Pitch DOFs is presented in Fig. 2.19.

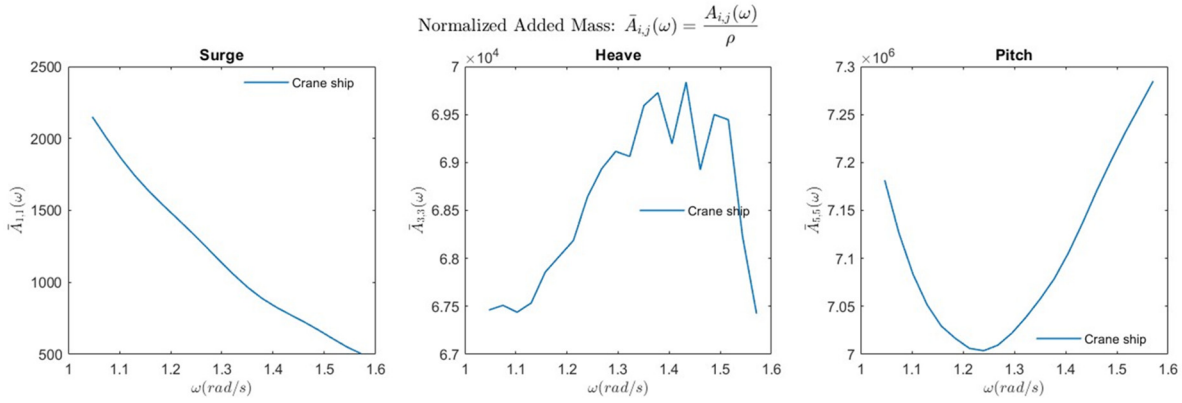


Fig. 2.19 Normalized added mass computed by Capytaine and BEMIO [37]

For calculations in WEC-Sim it is necessary to configure two files – Simulink model with crane ship components and MATLAB file *"wecSimInputFile.m"* with initial parameters of numerical model. As initial parameters of the model, the following parameters can be set: masses and inertia of all bodies in the model, frequencies, and directions of waves, initial positions of constraints and others.

Regarding the Simulink model, all advantages of WEC-Sim are revealed here. The minimum required set of blocks for calculation in WEC-Sim is Global Reference Frame block, body block from WEC-Sim library (for crane ship it is Hydrodynamic body block), and Floating Constraint block. The visualization of crane ship motions can be carried out using several tools, including the standard Simscape Mechanics Explorer and ParaView [50].

The model for the present simulation is shown in Fig. 2.20. The main structural parts of the vessel are represented by four WEC-Sim blocks. The hull of the ship is defined as a hydrodynamic body, and the other parts use non-hydrodynamic body blocks. Hydrodynamic bodies use the hydrodata structure files generated in BEMIO before to calculate forces from fluid medium acting in the CoG of these bodies. Non-hydrodynamic bodies are non-submerged parts of numerical model that act as gravity force applied to hydrodynamic body at reference frame. These bodies can be replaced by common rigid bodies of MATLAB Simscape block library. Gravitational forces for each part of the ship are automatically taken into account, and the position in space is adjusted using the standard rigid transform from the Simscape library. Since bridge, crane, and boom parts have their own mass and inertia and act as external forces, the total center of gravity in the final crane ship model has the correct position.

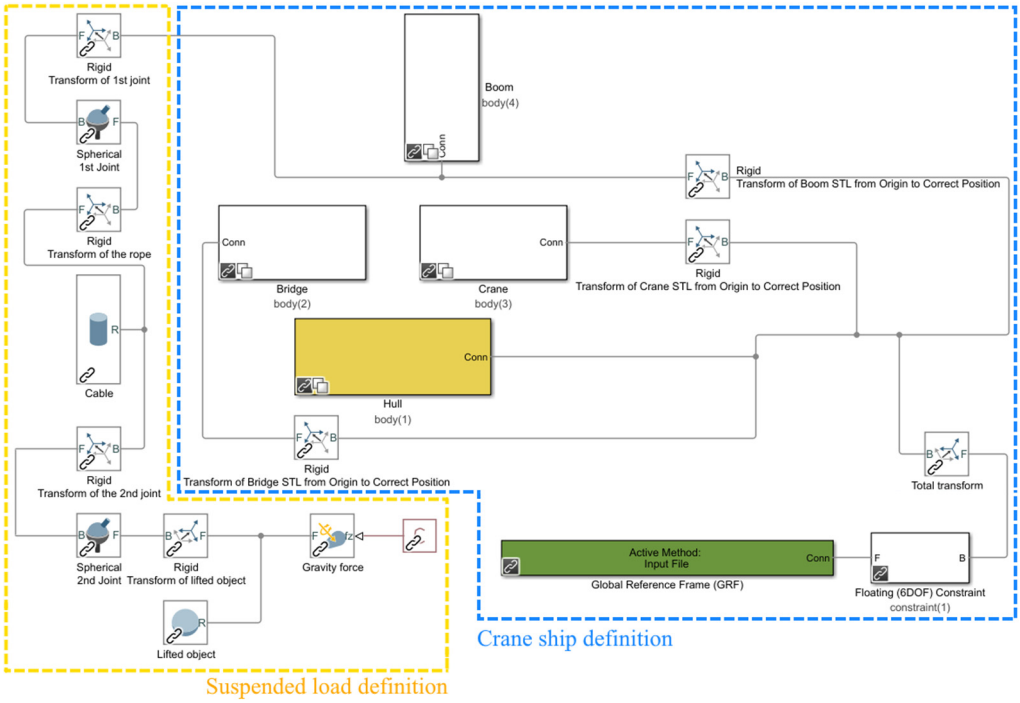


Fig. 2.20 Simulink model of crane ship [37]

As for the lifted object and the cable, they are represented as spherical and cylindrical bodies, respectively. Their connection to each other, as well as to the crane's boom, is carried out using spherical joints from the Simscape block library as well. Since in this case the lifted object body is not a non-hydrodynamic WEC-Sim body, it requires a definition of the gravity force. In current implementation the cable is rigid and further improvements of cable modeling is required to correctly account for suspended load dynamics.

The assembled model and its heave response for the direction of the regular wave are shown in Fig. 2.21 as an example of the simulation results. The wave direction coincides with the *x*-axis, the wave period is equal to 4 s, and the wave height is equal to 4 m. The described approach and modeling results are also described in Makarov and Harada (2022) [37].

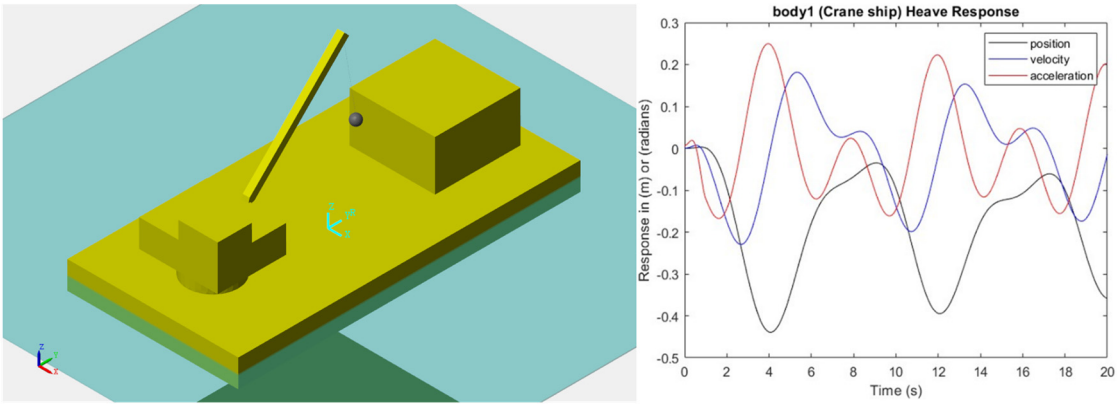


Fig. 2.21 Heave response of crane ship hull [37]

2.3.2. Selection and application of potential flow solver

As mentioned earlier, WEC-Sim has the BEMIO tool for converting output files from some solvers into a hydrodata structure file, which is further used in time-domain analysis. For comparative analysis and the selection of an optimal tool for solving the frequency-domain problem, the following open-source BEM solvers were considered: NEMOH, Capytaine, and HAMS.

NEMOH is a Boundary Element Methods (BEM) code dedicated to the computation of first order wave loads on offshore structures (added mass, radiation damping, diffraction forces). It has been developed by researchers at Ecole Centrale de Nantes (France) for 30 years [48, 51, 52]. The computation process in NEMOH assumes the use of several compiled executables for preprocessing, computation and postprocessing. The minimum required input files are geometry file, *Nemoh.cal* with simulation parameters, and *input_solver.txt* with solver settings. Regarding solvers the Gauss elimination, LU-decomposition and GMRES-iterative [54] solvers are available. Numerical implementation of the Green function is described in Delhommeau [55, 56]. The flowchart of computation in NEMOH is presented in Fig. 2.22.

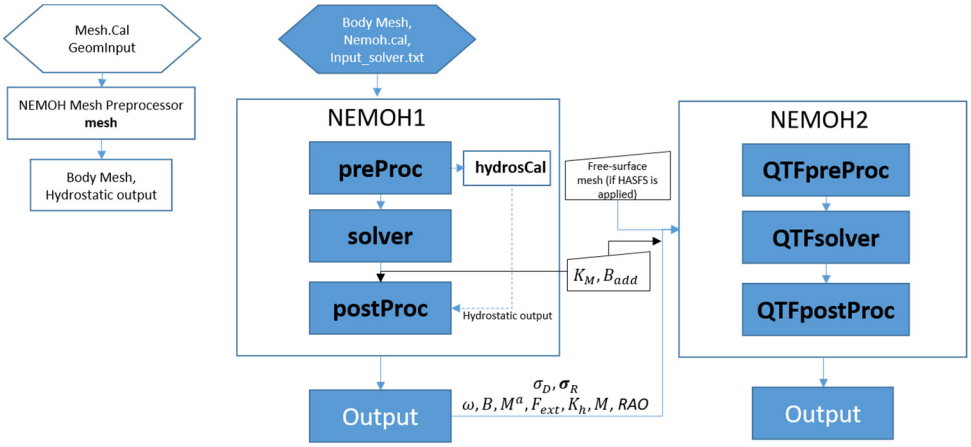


Fig. 2.22 Global flowchart of computation process in NEMOH software [57]

Despite NEMOH being, so to speak, a pioneer among open-source BEM solvers (according to developers, it is the world's first open-source BEM code), its usage imposes several limitations and inconveniences. Among such drawbacks, the following can be highlighted:

- the geometry file must be created in a specific format readable by NEMOH. Thus, for its use in conjunction with WEC-Sim, as well as other time-domain solvers, it is necessary to use multiple geometry files. Also, this format is specific and, accordingly, cannot be created in common CAD programs. Of course, it can be generated by converting popular CAD formats using, for example, a tool like BEM Rosetta [58].
- most of the solver's algorithms are written in FORTRAN. And although this programming language is a de facto standard in scientific computing due to its speed, the entry threshold is quite high, making customization of functions challenging and requiring specific programming skills from researchers.
- the computation process requires the sequential execution of various executable files. This is not a significant drawback but may pose an obstacle when dealing with specific tasks that require multi-condition runs.
- according to Penalba et al. [59], problems with the reliability of calculations are observed when computing thin structures.

Nevertheless, this software has been evolving for a considerable amount of time and has a wide range of features and tools. Recently, the new comprehensive update (NEMOH v3.0.0) has been released, which provided the possibility to compute quadratic transfer functions (QTFs) [60]. The application features of NEMOH and its comparison with codes such as WAMIT and HAMS can be studied in works [59-63], which are not explored here.

HAMS [64-66] is a free open-source software to analyze wave-structure interactions in the frequency domain. It has been developed by the author Yingyi Liu (Japan) in FORTRAN 90 programming language. HAMS requires two files to run the simulation – mesh file in special HAMS format and input file ControlFile.in with simulation parameters. The output files are generated in 3 formats – HAMS, WAMIT and Hydrostar. This feature is convenient for comparison of results obtained by several codes. Also, HAMS offers parallel processing technique via OpenMP and computation becomes faster than in NEMOH. As for the Green function this code implements two different algorithms developed by Liang et al. [67] for deep water conditions and Liu et al. [68] for the finite depth conditions. In general, the drawbacks of the code are similar to those of NEMOH – the need to generate a proprietary mesh and the difficulty of customization.

Capytaine [49] is a Python package for simulation of the interaction between water waves and floating bodies in frequency domain [69]. According to developer it is built around a full rewrite of the open-source solver NEMOH. But unlike NEMOH it requires only one customized Python script to for preprocessing, computation and postprocessing. Two formulations of Green function are

available described in [48] (same as NEMOH) and in [53]. Computations can be performed using two kinds of parallelization provided by OpenMP (same as in HAMS) or *joblib* Python package.

Since Capytaine is written in the Python programming language, which is very easy learnable and widespread among software developers, it becomes possible to use many third-party libraries, such as NumPy, SciPy, Matplotlib and VTK for customization, postprocessing and other purposes. Capytaine uses the Meshio and Meshmagic libraries for mesh operations. The latter is also a development of LHEEA for the manipulation of meshes encountered in the hydrodynamics community. Because of this convenient feature it allows to import more than 16 different mesh formats including mesh files from WAMIT, NEMOH, Hydrostar.

As it is common in BEM codes, the free water surface must be located at $z = 0$, which requires the creation of a mesh file in which the position of the vessel must correspond to its draught. However, in WEC-Sim mesh origin of each part of the vessel must be located at its center of gravity, the vertical coordinate of which is often not the same as the free surface of the water. The convenience of using Capytaine is that the orientation and position of the mesh can be modified after reading the file by the program tools (unlike in NEMOH or HAMS). For a combined calculation with WEC-Sim, one mesh model can be created in a third-party CAD program with the origin in the center of gravity. After initialization of the calculation, the mesh will be automatically moved to the required position and cut along the waterline to save computing resources. Therefore, there is no need to create many geometry files to analyze the crane ship at different drafts.

The dataset of the results from the Capytaine calculation is written in NetCDF file format. For further import into WEC-Sim, two additional files KH.dat and Hydrostatics.dat must be created (same as NEMOH output), which include the hydrostatic stiffness matrix of the body, coordinates of the buoyancy center, and the displaced water volume. These parameters can be easily calculated with Meshmagic during the calculation process and written into the files using standard Python functions. Capytaine output files can be written to any directory after the calculation. In this regard, it is more convenient to organize the export of the results into a folder containing the BEMIO tool of WEC-Sim in order to avoid manual copying of the files.

The author's understanding is that individual numerical approaches and methods within both commercial and free codes may differ. For example, different formulations of the Green function, various solvers, or simply different units of measurement may be applied. Therefore, the task of comparing resulting hydrodynamic coefficients is an interesting aspect for investigation. Some code comparisons are available for study through the previously mentioned references, but they are not considered in this research.

However, within this study, the computational efficiency of these open-source codes was explored by solving radiation and diffraction problems for the hull of the physical model tested earlier in the wave basin (Section 2.2). For this purpose, several mesh files were generated. The basic

geometry was created in the CAD editor of ANSYS [70], based on the dimensions of characteristic sections of the full-scale vessel at locations depicted in Fig. 2.23 , and then scaled down by a factor of 50.

Then, this model was discretized into elements and converted to the ".dat" format, which can be used in all the aforementioned codes. In total, 9 mesh models with different levels of discretization were created, as shown in Fig. 2.24. Test calculations for each model were conducted for 9 wave directions uniformly distributed from 0 to 2π and 100 wave frequencies ranging from 0.1 to 12.566 rad/s.

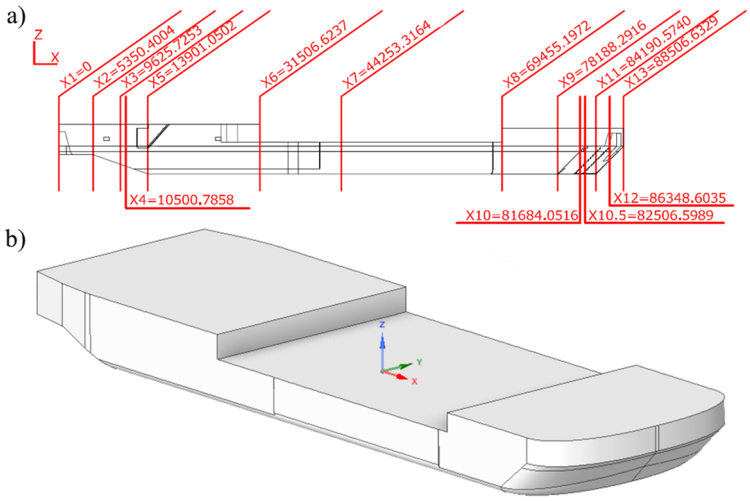


Fig. 2.23 CAD model of crane vessel hull: a) the characteristic sections; b) the final CAD model

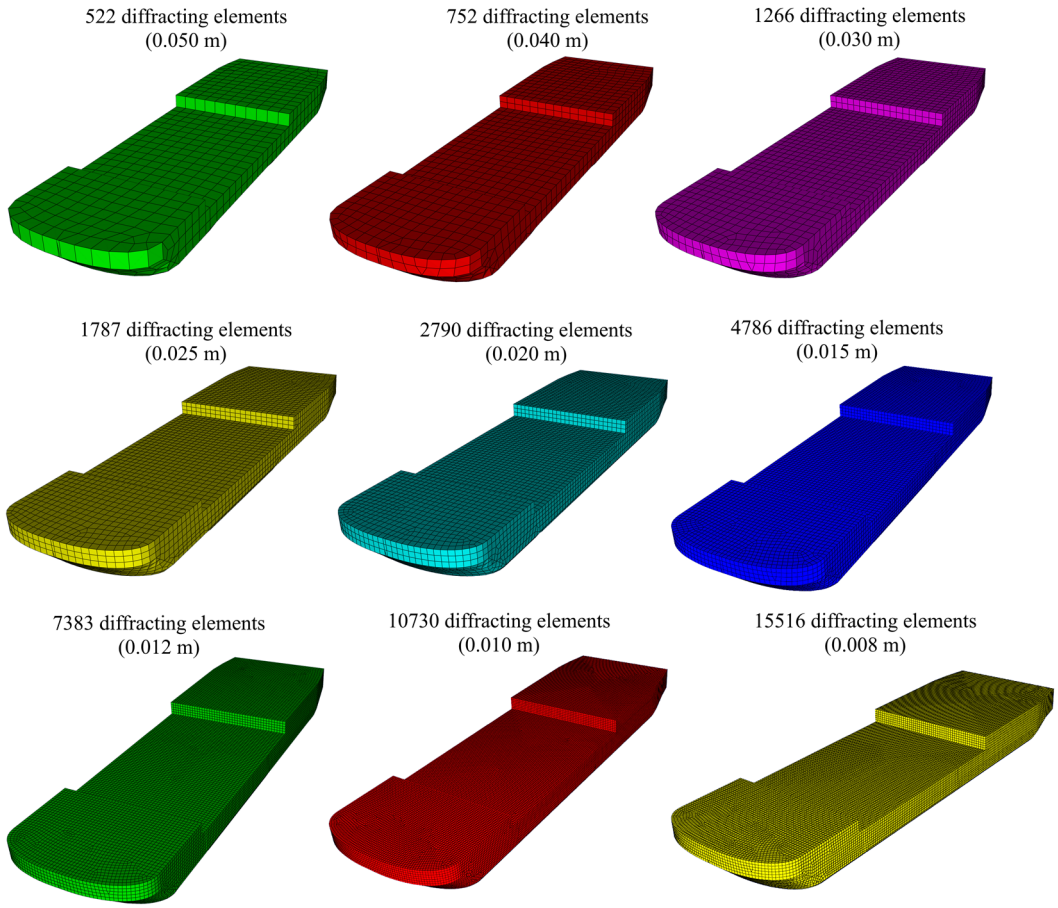


Fig. 2.24 Mesh models used in BEM computations

Results of the performance test are presented in Fig. 2.25. All calculations were performed using a direct solver with caching of the LU decomposition, which is available in all tested BEM codes. This solver offers excellent convergence at a much faster calculation speed than a simple direct solver. As seen in the figure, the NEMOH solver is significantly slower than HAMS and Capytaine even with a small number of diffracting elements. This is likely due to the lack of parallelization in the calculations. The computational complexity is nonlinear, leading to 35 hours of calculation time for 10,730 panels. The case with a model of 15,516 diffracting elements was not simulated in NEMOH because the predicted calculation time was too large.

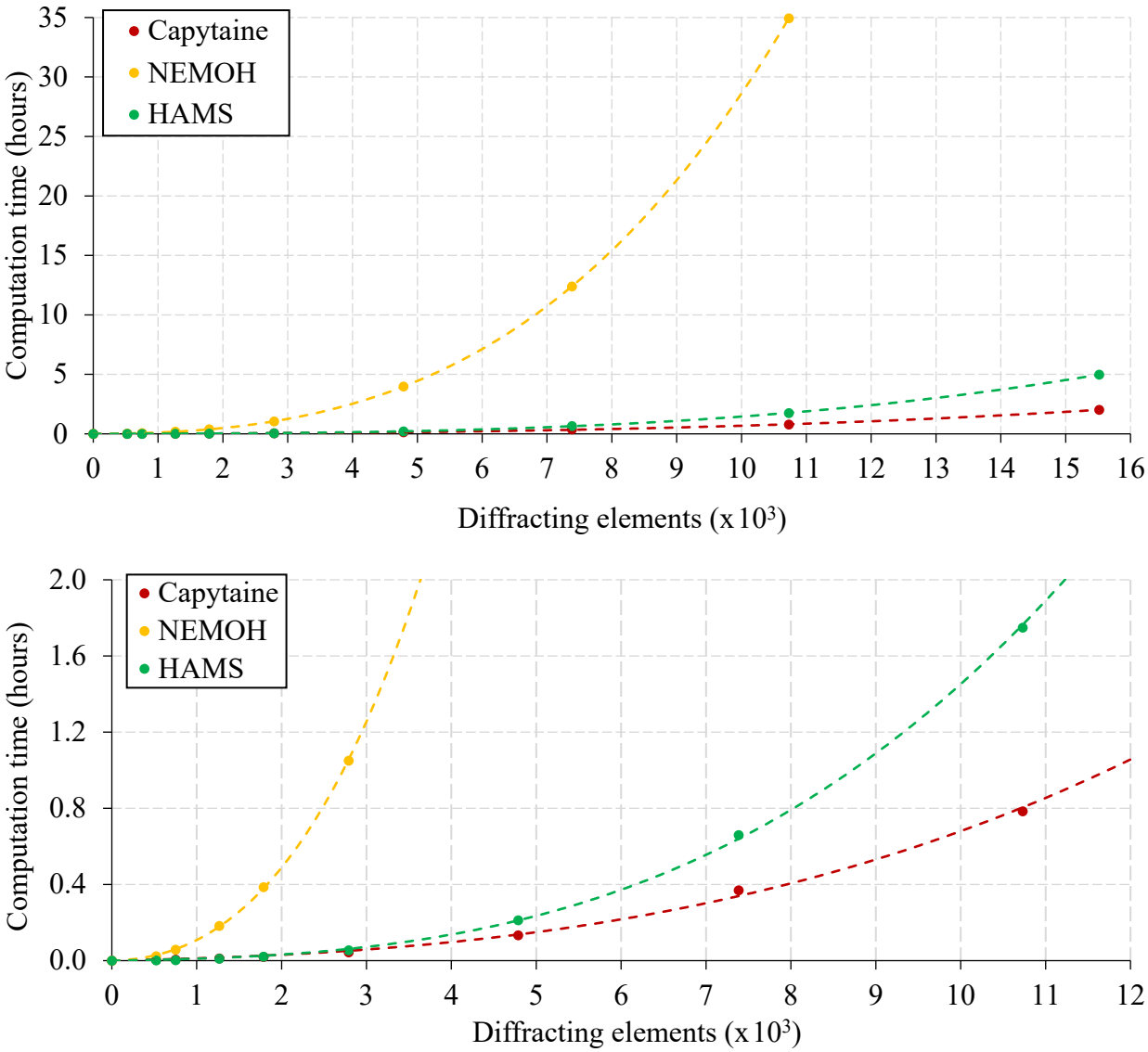


Fig. 2.25 Computation performance of open-source BEM codes

An interesting result is the comparison between Capytaine and HAMS. Despite HAMS being written in the much faster FORTRAN programming language, Capytaine, implemented in Python, proved to be more than 2 times more efficient in all simulated cases. The model with 15516 diffracting elements was resolved in 2.02 hours in Capytaine and 4.98 hours in HAMS. The performance test results are summarized in Table 2.3.

Table 2.3 Results of computation performance

Diffracting elements	Total runtime, hours		
	Capytaine	NEMOH	HAMS
15516	2.020	–	4.981
10730	0.784	34.940	1.749
7383	0.369	12.391	0.659
4786	0.133	3.976	0.212
2790	0.044	1.050	0.054
1787	0.019	0.386	0.021
1266	0.011	0.182	0.010
752	0.005	0.058	0.003
522	0.003	0.023	0.001

Thus, the Capytaine solver offers the best performance and customization capabilities due to its use of open-source code in the Python language. These advantages make it an excellent choice for developing a crane ship modeling method that needs to be both fast, widely accessible, and customizable. It's worth noting that in WEC-Sim, results obtained from other open-source software can be utilized. However, in such cases, the hydrodata structure file, which is used in time-domain computation by WEC-Sim, needs to be manually generated.

2.3.3. Modeling of vessel hydrodynamics in the time domain. Assessment of the computation reliability and performance

Model preparation.

Next, the previously tested approach is applied to the numerical modeling of the dynamics of the crane vessel and suspended load in the time domain using WEC-Sim. This will allow us to assess the computation reliability and performance and identify additional requirements for the numerical model. The computations were performed using a model presented in Fig. 2.26. The conclusions are followed by a comparison of the modeling results with wave basin experiments on the determination of wave-induced motions of crane ship and also discussed in Makarov and Harada [71].

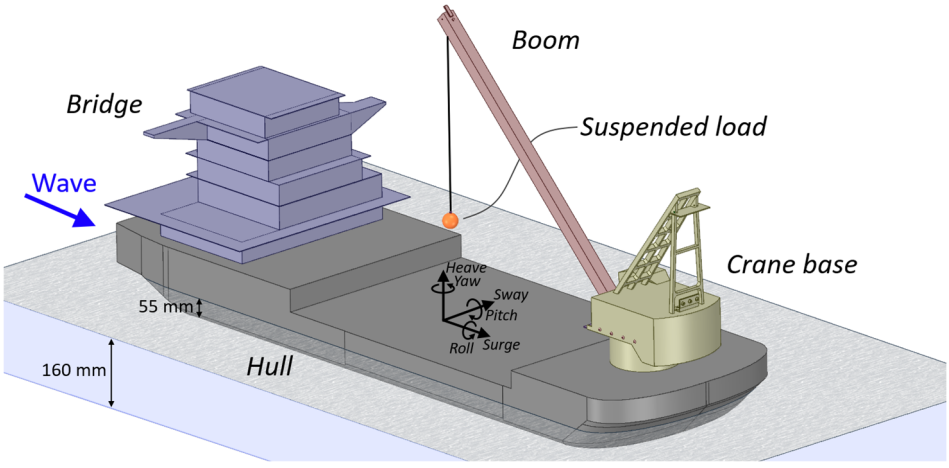


Fig. 2.26 Numerical model of a crane ship with a suspended load [71]

The author understands that when using seakeeping ship models, various approaches exist for computing external hydrodynamic forces on the crane vessel hull, denoted in the right part of Eq. (1.2), which is rewritten below once again:

$$\mathbf{M}\ddot{\mathbf{x}}(t) = \mathbf{F}_{hs}(t) + \mathbf{F}_{exc}(t) + \mathbf{F}_{rad}(t) + \mathbf{F}_{v}(t) + \mathbf{F}_{ext}(t), \tag{2.2}$$

where $\mathbf{F}_{hs}(t)$ is hydrostatic restoring force vector; $\mathbf{F}_{exc}(t)$ is the wave excitation and diffraction force vector; $\mathbf{F}_{rad}(t)$ is the force vector resulting from wave radiation; $\mathbf{F}_{v}(t)$ is the viscous force vector; $\mathbf{F}_{ext}(t)$ is externally applied forces vector (including other terms of wave forces).

And when it comes to reliability, it is essential to understand how the dynamics of the vessel will change when different approaches are applied for force calculations. In addition, the accuracy of the calculations depends on the mesh size, so appropriate studies need to be carried out. In this regard, the results of dynamic analysis of a crane ship and suspended load motions in different formulations of the equation of motion are presented below.

At the first stage the following hydrodynamic coefficients were calculated using Capytaine software – added mass $A(\omega)$, radiation damping $B(\omega)$, wave excitation $F_{exc}(\omega, \theta)$ and hydrostatic restoring coefficients K_{hs}. The mesh of the hull is illustrated in Fig. 2.27 and consists of 15504 diffracting elements. Since the further simulations assumes the action of regular wave with period T equal to 1.4 s and wave heading θ equal to 0 deg, then the hydrodynamic coefficients are of interest only for this case. The values of these hydrodynamic coefficients are summarized in Tables 2.4-2.7. Values are given in the SI system (N, N/m, N/rad, N·m/rad, N/(m/s), etc.).

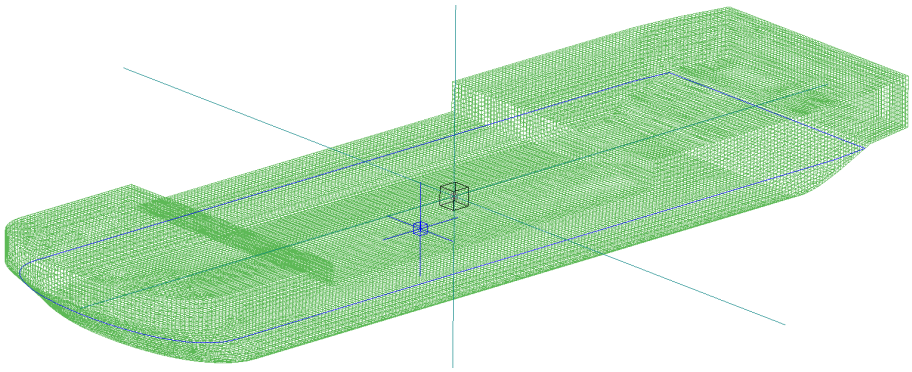


Fig. 2.27 Geometric model (mesh) of the crane ship hull [71]

Table 2.4 Hydrostatic stiffness matrix K_{hs}

DoF	Surge	Sway	Heave	Roll	Pitch	Yaw
Surge	0	0	0	0	0	0
Sway	0	0	0	0	0	0
Heave	0	0	8438.399	-0.019	-469.699	0
Roll	0	0	-0.019	183.784	-0.065	-34.321
Pitch	0	0	-469.699	-0.065	1795.336	-0.001
Yaw	0	0	0	0	0	0

Table 2.5 Added mass matrix $A(\omega)$

DoF	Surge	Sway	Heave	Roll	Pitch	Yaw
Surge	2.590	0	1.994	0	5.391	0
Sway	0	12.111	0	-2.571	0	0.828
Heave	1.988	0	229.763	0	-14.434	0
Roll	0	-2.542	0	2.515	0	-0.233
Pitch	5.382	0	-14.384	0	46.651	0
Yaw	0	0.827	0	-0.232	0	2.504

Table 2.6 Radiation damping matrix $B(\omega)$

DoF	Surge	Sway	Heave	Roll	Pitch	Yaw
Surge	9.474	0	2.282	0	39.325	0
Sway	0	56.568	0	-19.327	0	4.365
Heave	2.758	0	1353.951	0	-75.093	0
Roll	0	-19.132	0	6.582	0	-1.261
Pitch	39.262	0	-75.222	0	189.444	0
Yaw	0	4.357	0	-1.258	0	5.870

Table 2.7 Excitation force coefficients $F_{exc}(\omega, \theta)$

DoF	Magnitude (N/m)	Phase (rad)
Surge	203.247	-2.743
Sway	0.015	-0.965
Heave	920.322	-1.900
Roll	0.052	0.323
Pitch	1368.088	-2.248
Yaw	0.025	2.993

At second step, the Simulink model of crane vessel has been built. In current implementation the nonhydrodynamic bodies were replaced by conventional Simscape rigid body blocks since there is no need to account for the buoyancy of upper parts of the vessel. This also resolved a minor issue in the WEC-Sim software, which was that nonhydrodynamic blocks did not allow specifying values for products of inertia. As the current model involves asymmetric geometry, we reported this issue to the developers as feedback [72], leading to the addition of such an option in the new version of WEC-Sim [73]. Thus, a small contribution was made to the development of the WEC-Sim software.

The crane ship model was assembled from 8 rigid bodies (Fig. 2.28) – hull, extra lead weight, crane, bridge, boom, cable (rigid), hook, and suspended load. The mass properties of rigid bodies are presented in Table 2.8. The origin of the coordinate system is located at the sea level for z coordinate and center of gravity of the vessel's hull for x and y coordinates. The process of numerical model creation involves the import of the geometry of each part (STL format), a description of connections (joints), and a definition of geometric and inertial properties (masses, moments of inertia, centers of gravity position, etc.) as described earlier in Subsection 2.3.1.

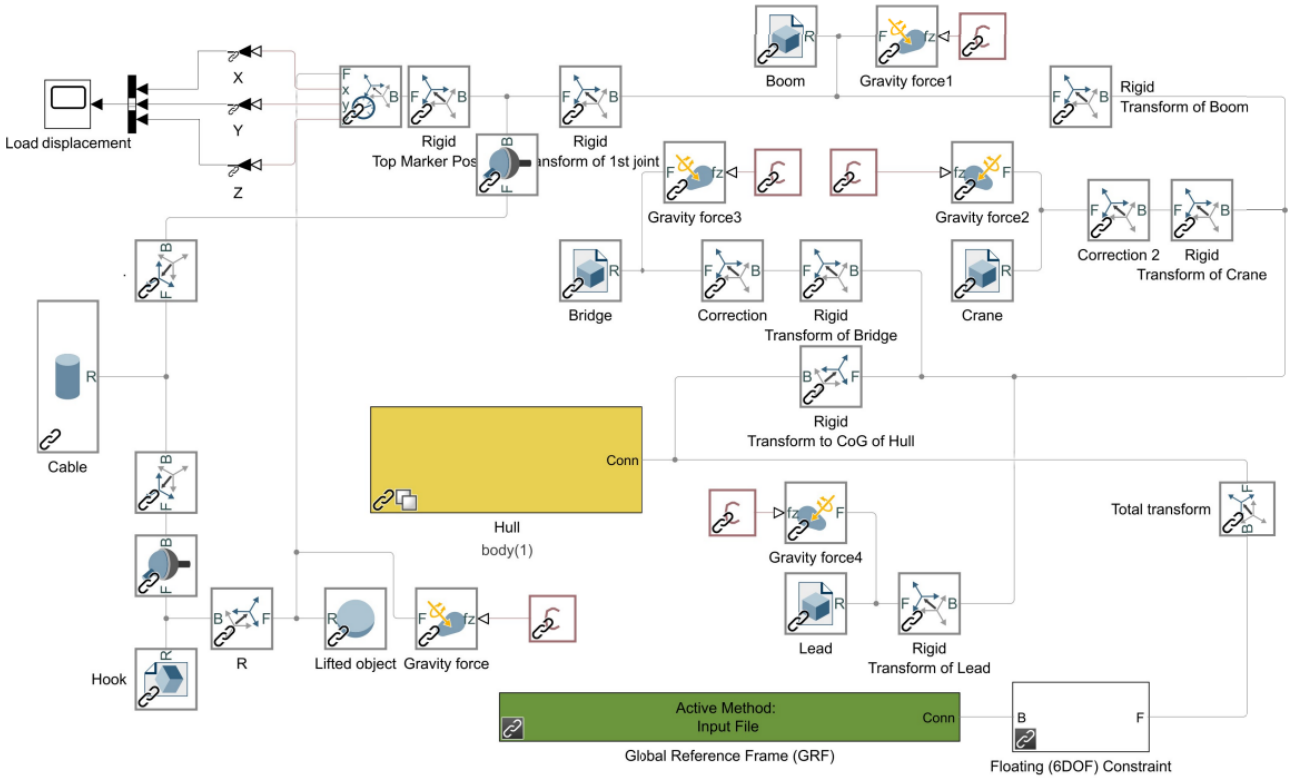


Fig. 2.28 Model assembly in Simulink environment

Table 2.8 Mass and inertia of rigid bodies

Body	Mass (kg)	Principal moments of inertia (kg·m^2)			Center of gravity coordinates (m)		
		I_{p1}	I_{p2}	I_{p3}	x	y	z
Hull	21.4	0.541	5.580	6.052	0	0	0.015
Lead weight	9.76	0.044	0.007	0.043	0.656	0	0.010
Bridge	9.28	0.255	0.175	0.275	-0.590	0.013	0.254
Crane	3.15	0.055	0.058	0.019	0.765	0.003	0.269
Boom	0.56	0.0001	0.033	0.033	0.313	0	0.483
Cable	0.003	6e-05	6e-05	4e-10	0.028	0	0.517
Hook	0	0	0	0	0.028	0	0.289
Suspended load	0.16	2e-5	2e-5	2e-5	0.028	0	0.267

Linear force calculation.

In the simple linear case, all forces are derived based on the coefficients obtained from the boundary element analysis. The restoring force $\mathbf{F}_{hs}(t)$ in this case is calculated as follows:

$$\mathbf{F}_{hs}(t) = \mathbf{K}_{hs}\mathbf{x} \tag{2.3}$$

where $\mathbf{K}_{hs}$ is the hydrostatic stiffness matrix; $\mathbf{x}$ is the column vector of vessel offsets (Heave, Roll, Pitch, Yaw) relative to its datum position.

Consequently, the relative position of the wave crest and the wetted surface itself is not taken into account here. This approach is quite common in the case of small changes in the vessel's position and orientation.

The wave excitation forces $\mathbf{F}_{exc}(t)$ are determined by linear (Airy) regular wave theory for a given wave height and wave frequency, and water depth:

$$\mathbf{F}_{exc}(t) = \Re\left[R_f(t)\frac{h}{2}\boldsymbol{F}_{exc}(\omega,\theta)e^{i\omega t}\right], \tag{2.4}$$

where $\Re$ stands for the real part of a complex number; h is the wave height; $\boldsymbol{F}_{exc}(\omega,\theta)$ is the complex wave-excitation amplitude vector obtained from the frequency domain solution; $R_f(t)$ is the ramp function [46] included in WEC-Sim numerical method to avoid strong transient flows at the earlier time steps of the simulation:

$$R_f(t) = \begin{cases} \frac{1}{2}(1+\cos\left(\pi+\frac{\pi t}{t_r}\right), & \frac{t}{t_r} < 1 \\ 1, & \frac{t}{t_r} \geq 1 \end{cases} \tag{2.5}$$

where t is the simulation time and t_r is the ramp time.

As for radiation forces, their determination is available in WEC-Sim in two formulations. The first formulation involves the direct use of frequency-dependent added mass and radiation damping matrices:

$$\mathbf{F}_{rad}(t) = -\mathbf{A}(\omega)\ddot{\mathbf{x}} - \mathbf{B}(\omega)\dot{\mathbf{x}}, \tag{2.6}$$

where $\mathbf{A}(\omega)$ is a frequency-dependent added mass matrix; $\mathbf{B}(\omega)$ is a frequency-dependent radiation damping matrix; $\dot{\mathbf{x}}$ is the velocity vector of the floating body; and $\ddot{\mathbf{x}}$ is the acceleration vector.

In this case, the complete equation of motion will correspond to equation (1.4), but without considering the viscous damping matrix $\mathbf{C}_v$:

$$\big(\mathbf{M}+\mathbf{A}(\omega)\big)\ddot{\mathbf{x}}(t) + \mathbf{K}_{hs}\mathbf{x}(t) + \mathbf{B}(\omega)\dot{\mathbf{x}}(t) = \Re\left[R_f(t)\frac{h}{2}\mathbf{F}_{exc}(\omega,\theta)e^{i\omega t}\right] + \mathbf{F}_{ext}(t), \tag{2.7}$$

where $\mathbf{A}(\omega)$ and $\mathbf{B}(\omega)$ are the same matrices as in Eq. (2.6); $\mathbf{F}_{exc}(\omega,\theta)$ is the same as in Eq. (2.4); $\mathbf{K}_{hs}$ is the matrix of hydrostatic stiffness coefficients of the floating body K_{hs}.

Thus, having 8 rigid bodies in the multibody system, of which one (the hull) is subject to the hydrodynamic forces, we can assemble the following differential-algebraic equation of motion (similar to Eq. (1.11)):

$$\left[\begin{array}{cccccc|c|c|c} \mathbf{M}_{hull} & & & & & & & & \\ & \mathbf{M}_{lead} & & & & & & & \\ & & \mathbf{M}_{bridge} & & & & & & \\ & & & \mathbf{M}_{crane} & & & \mathbf{C}_{\mathbf{x}}^{\mathrm{T}} & & \\ & & & & \mathbf{M}_{boom} & & & & \\ & & & & & \mathbf{M}_{cable} & & & \\ & & & & & & \mathbf{M}_{hook} & & \\ & & & & & & & \mathbf{M}_{load} & \\ \hline & & & & & & & & \\ & & & \mathbf{C}_{\mathbf{x}} & & & \mathbf{0} & & \\ & & & & & & & & \end{array}\right] \left[\begin{array}{c} \mathbf{x}^{hull} \\ \mathbf{x}^{lead} \\ \mathbf{x}^{bridge} \\ \mathbf{x}^{crane} \\ \mathbf{x}^{boom} \\ \mathbf{x}^{cable} \\ \mathbf{x}^{hook} \\ \mathbf{x}^{load} \\ \hline \\ \boldsymbol{\gamma} \end{array}\right] = \left[\begin{array}{c} \mathbf{F}_e^{hull} \\ \mathbf{F}_e^{lead} \\ \mathbf{F}_e^{bridge} \\ \mathbf{F}_e^{crane} \\ \mathbf{F}_e^{boom} \\ \mathbf{F}_e^{cable} \\ \mathbf{F}_e^{hook} \\ \mathbf{F}_e^{load} \\ \hline \\ \boldsymbol{\lambda} \end{array}\right] \tag{2.8}$$

where mass matrix $\mathbf{M}_{hull}$ and external force matrix $\mathbf{F}_e^{hull}$ for the hull are constructed as follows:

$$\boldsymbol{M}_{hull} = \begin{bmatrix} m + A_{11} & 0 & A_{13} & 0 & A_{15} & 0 \\ 0 & m + A_{22} & 0 & A_{24} & 0 & A_{26} \\ A_{31} & 0 & m + A_{33} & 0 & A_{35} & 0 \\ 0 & A_{42} & 0 & I_{11} + A_{44} & I_{12} & I_{13} + A_{46} \\ A_{51} & 0 & A_{53} & I_{21} & I_{22} + A_{55} & I_{23} \\ 0 & A_{62} & 0 & I_{31} + A_{64} & I_{32} & I_{33} + A_{66} \end{bmatrix} \tag{2.9}$$

$$\mathbf{F}_e^{hull} = \begin{bmatrix} R_f(t)\Re\left(\frac{h}{2}F_{exc,1}e^{i\omega t}\right) \\ R_f(t)\Re\left(\frac{h}{2}F_{exc,2}e^{i\omega t}\right) \\ R_f(t)\Re\left(\frac{h}{2}F_{exc,3}e^{i\omega t}\right) \\ R_f(t)\Re\left(\frac{h}{2}F_{exc,4}e^{i\omega t}\right) \\ R_f(t)\Re\left(\frac{h}{2}F_{exc,5}e^{i\omega t}\right) \\ R_f(t)\Re\left(\frac{h}{2}F_{exc,6}e^{i\omega t}\right) \end{bmatrix} - \tag{2.10}$$

$$- \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & K_{33} & K_{34} & K_{35} & 0 \\ 0 & 0 & K_{43} & K_{44} & K_{45} & K_{46} \\ 0 & 0 & K_{53} & K_{54} & K_{55} & K_{56} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} - \begin{bmatrix} B_{11} & 0 & B_{13} & 0 & B_{15} & 0 \\ 0 & B_{22} & 0 & B_{24} & 0 & B_{26} \\ B_{31} & 0 & B_{33} & 0 & B_{35} & 0 \\ 0 & B_{42} & 0 & B_{44} & 0 & B_{46} \\ B_{51} & 0 & B_{53} & 0 & B_{55} & 0 \\ 0 & B_{62} & 0 & B_{64} & 0 & B_{66} \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix}$$

Another available formulation of radiation forces takes into account fluid memory effect by integrating all the way back to the start of the simulation [74]. This effect is handled by the convolution integral as follows:

$$\mathbf{F}_{rad}(t) = -\mathbf{A}_{\infty}\ddot{\mathbf{x}} - \int_0^t \mathbf{K}_r(t-\tau)\,\dot{\mathbf{x}}(\tau)d\tau, \tag{2.11}$$

$$\mathbf{K}_r(t) = \frac{2}{\pi}\int_0^{\infty} \mathbf{B}(\omega)\cos(\omega t)d\omega, \tag{2.12}$$

where $\mathbf{A}_{\infty}$ is the added mass matrix at infinite frequency; $\mathbf{K}_r(t)$ is the radiation impulse response function; τ is a time lag integration variable.

In current simulation all terms in the added mass matrix at infinite frequency $\mathbf{A}_{\infty}$ determined in Capytaine are equal to 4.489 kg. When using equations (2.11) and (2.12), the equation of motion for the floating body corresponds to Eq. (1.6), but without the added viscous damping matrix $\mathbf{C}_v$:

$$\begin{aligned}(\mathbf{M}+\mathbf{A}_{\infty})\ddot{\mathbf{x}}(t) + \mathbf{K}_{hs}\mathbf{x}(t) + \int_0^t \mathbf{K}_r(t-\tau)\dot{\mathbf{x}}(\tau)\mathrm{d}\tau = \\ = \Re\left[R_f(t)\frac{h}{2}\mathbf{F}_{exc}(\omega,\theta)e^{i\omega t}\right] + \mathbf{F}_{ext}(t),\end{aligned} \tag{2.13}$$

The modeling results in time-domain are shown in Fig. 2.29. For comparison, measurements in the wave basin and simulations with and without considering the fluid memory effect (named "Linear FME") are illustrated. The total runtime was only 3 seconds for a 25-second event with 0.005s timestep size, which can be considered extremely fast. The ramping time t_r of the wave in the numerical experiment was accepted to be equal to four periods (5.6 seconds) to achieve the coincidence with laboratory experiments.

As can be seen from the results, the motions in Heave DOF are very close to experiment measurements. However, the linear calculation does not provide good accuracy for the other degrees of freedom. For example, the wave impact does not cause a gradual forward movement of the ship (Surge).

The drawback of the linear calculation is also apparent in the Sway-Roll-Yaw response. Due to the asymmetry of the upper structures of the model (Fig. 2.30), the center of gravity of the vessel is not located in its centerline (fore-aft line). The initial roll was calibrated at 0.4 degrees. The results indicate that the vessel tends to return to the straight position that was calculated in the BEM solver despite the initial roll, and it experiences an initial momentum. This is even more evident in the

calculation using the impulse response function. Due to the restoring force, a progressive movement along the local *y*-axis (Sway) and a slight yaw occurred.

Accordingly, the linear calculation cannot handle the inclined position of the floating body properly. Therefore, it is not applicable in those models where the neutral position of the floating body can change during the simulation. This is especially relevant for crane vessels since cargo handling operations involve changes in the ship's center of gravity even while the cargo is moving in the air.

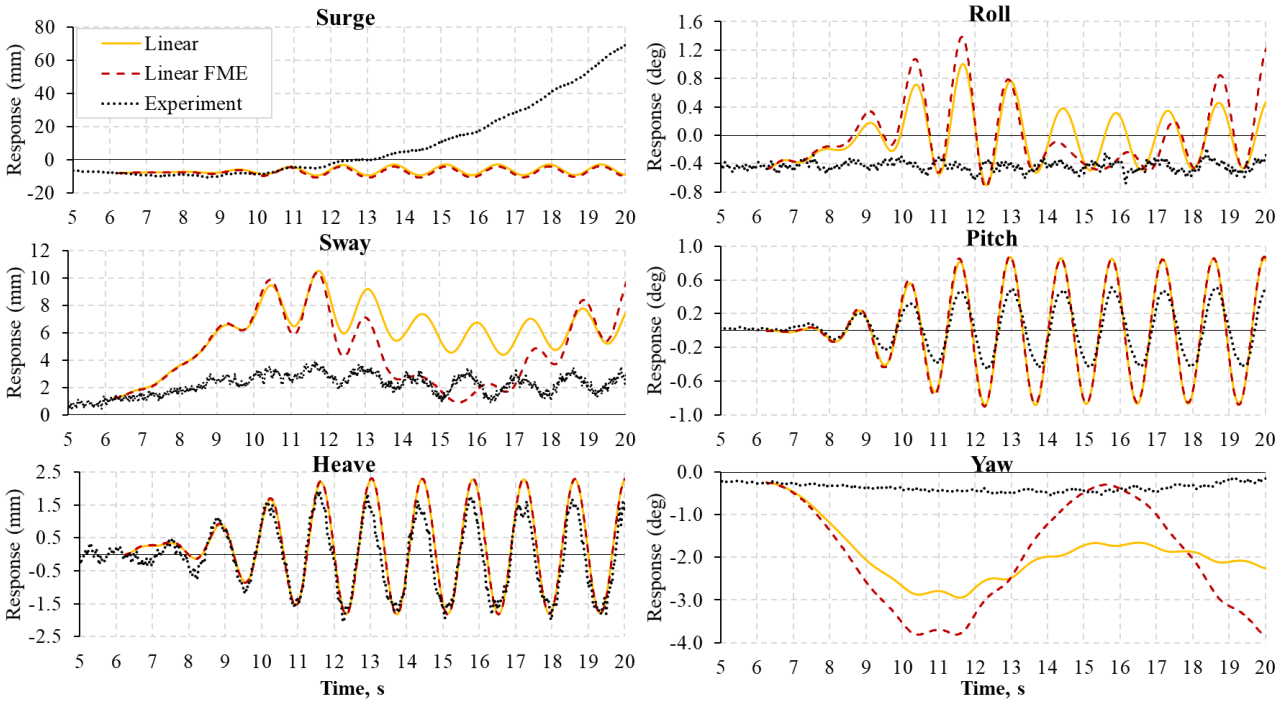


Fig. 2.29 Motions of the ship model in the linear case compared to experiments in the wave basin. Wave height h = 0.02 m, wave period T = 1.4 s [71]

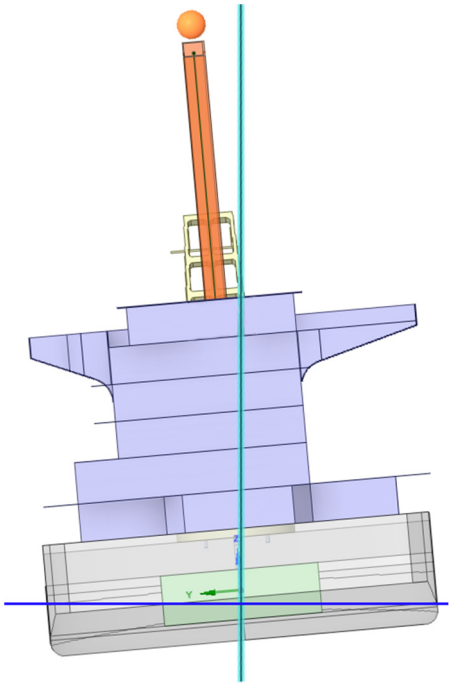


Fig. 2.30 Initial inclined position of crane vessel model (roll is magnified for clarity)

Since the rotational DOFs did not match the experiment well, the displacements of the suspended cargo are also dissimilar, as illustrated in Fig. 2.31. The relative displacements of the cargo along the local *y*-axis differ significantly because of the large rolling of the ship.

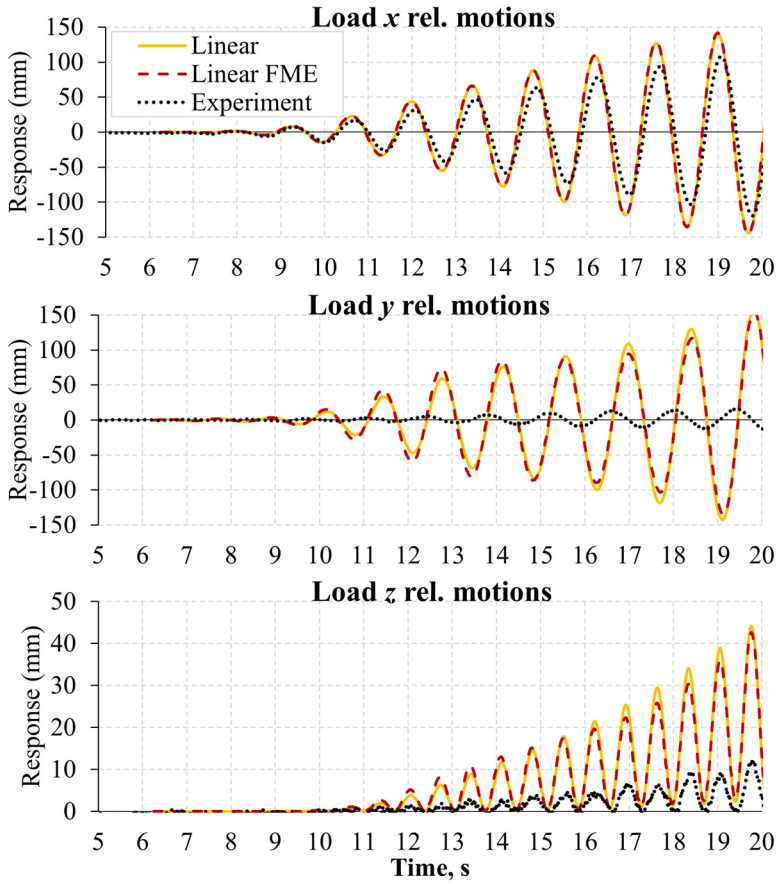


Fig. 2.31 Displacements of the suspended load relative to the upper joint in the linear calculation. Wave height h = 0.02 m, wave period T = 1.4 s [71]

Nonlinear force calculation.

Linear calculations do not provide the necessary accuracy, although they are extremely fast. Nonlinear terms must be taken into account to obtain a more accurate dynamic response of the crane ship and the suspended load. The first such term is nonlinear Froude-Krylov force. An enhanced approach involves calculating the hydrodynamic pressures on panels of hull mesh at each time step [75]. For this purpose, the instantaneous positions of the submerged mesh panels at centroid become the input parameters. The wave pressure in such implementation is expressed as follows:

$$p_{FK,i} = \frac{1}{2}\rho g h \frac{\cosh\big(k(z_i + d)\big)}{\cosh(kd)} \cos\varphi, \tag{2.14}$$

where ρ is water density; g is gravitational acceleration; h is wave height; k is wave number; z_i is vertical coordinate of centroid of the mesh panel i relative to the water surface; d is water depth, and φ is the wave phase angle.

Since, in this calculation formulation, the ship's hull mesh is involved, it also makes sense to compute the restoring forces by determining the hydrostatic pressure of the water column on each mesh element:

$$p_{hs,i} = -\rho g z_i, \tag{2.15}$$

This also solves the problem of determining the equilibrium state with asymmetric geometry of the crane vessel, as mentioned earlier in the full linear calculation. In this way, the variable wetness of the ship hull can be taken into account, which is crucial point for crane ships since CoG of such vessels change during cargo lifting operations.

As the restoring forces are now determined based on the instantaneous position of each grid element, the hydrostatic stiffness matrix $\mathbf{K}_{hs}$ is no longer used. The vectors of Froude-Krylov force $\mathbf{F}_{FK}$ and restoring force $\mathbf{F}_{hs}$ are determined by integrating the pressures over the wetted surface on-the-fly during simulation:

$$\mathbf{F}_{FK} = \iint_S p_{FK} dS \approx \sum_i^N p_{FK,i} S_i \tag{2.16}$$

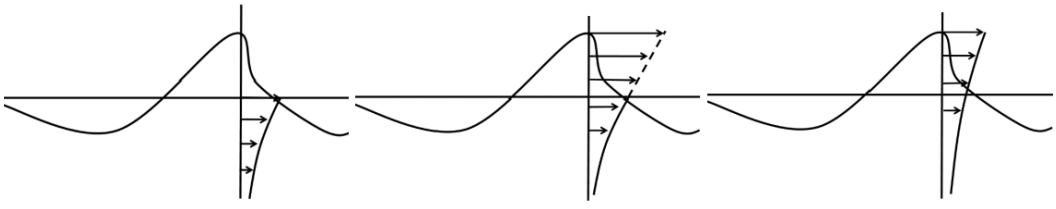
$$\mathbf{F}_{hs} = \iint_S p_{hs} dS \approx \sum_i^N p_{hs,i} S_i \tag{2.17}$$

where S_i is the area of face i; N is the number of faces of STL mesh.

Also, a modified formulation accounts for the instantaneous position of the wave crest in relation to the crane ship waterline using the mathematical method named Wheeler stretching [76]. It corrects the unrealistically large pressure values for wetted panels above mean water level by artificially increasing the water column (Fig. 2.32):

$$z_{\mathrm{i}}^* = d\frac{d + z_i}{d + \eta} - d \tag{2.18}$$

where z_i^* is an effective (corrected) z_i-coordinate for pressure calculation; d is still water depth; η is instantaneous wave elevation.



(a) Linear theory (b)Linear extrapolation (c) Wheeler Stretching

Fig. 2.32 Conceptual sketch of horizontal water particle velocity according to linear extrapolation and Wheeler Stretching for points above mean water level [77]

Since the Froude-Krylov forces are the part of the wave excitation forces, which also include diffraction and scattering forces, a simple addition of nonlinear Froude-Krylov forces to the model will be incorrect. In such a case, a double accounting of these forces will occur, since the linear part is already accounted for in Eq. (2.4). In WEC-Sim the following approach for correct calculation of excitation forces with nonlinear option is implemented:

1. The total nonlinear Froude-Krylov forces $\mathbf{F}_{FK}$ are represented by a combination of the linear $\mathbf{F}_{FK}^{lin}$ and nonlinear $\mathbf{F}_{FK}^{nonlin}$ parts:

$$\mathbf{F}_{FK} = \mathbf{F}_{FK}^{lin} + \mathbf{F}_{FK}^{nonlin} \tag{2.19}$$

2. The linear part of the Froude-Krylov forces is determined only by Eq. (2.14) and Eq. (2.16) for the mean wave surface and does not take into account its elevation ($\eta = 0$).
3. The total Froude-Krylov forces $\mathbf{F}_{FK}$ are determined by the above algorithm for the current wave elevation η.
4. The nonlinear part of the Froude-Krylov forces is determined as $\mathbf{F}_{FK}^{nonlin} = \mathbf{F}_{FK} - \mathbf{F}_{FK}^{lin}$ and summarized with the linear wave excitation forces determined by Eq. (2.4):

$$\mathbf{F}_{exc} = \Re\left[R_f(t)\frac{h}{2}\boldsymbol{F}_{exc}(\omega,\theta)e^{i\omega t}\right] + \mathbf{F}_{FK}^{nonlin} \tag{2.20}$$

In other words, only the nonlinear part of the Froude-Krylov forces is added to the linear wave excitation forces determined through the hydrodynamic coefficients obtained from the BEM potential flow solver. The diffraction and scattering forces are preserved, and there is no double accounting of the Froude-Krylov forces.

Therefore, the term “nonlinear” in the current formulation specifically refers to the expression for determining the nonlinear part of Froude-Krylov force. Nevertheless, these forces still correspond to linear wave theory (Airy waves). Approximating the wave according to linear theory naturally has obvious limitations, expressed in its applicability only in a certain range of wave lengths and heights. Méhauté [78], for example, provides a diagram showing ranges of acceptable values for wave parameters in various wave theories (Fig. 2.33). Unfortunately, in the current implementation of WEC-Sim, there are no formulations other than the linear one. A positive aspect is that crane vessels are not operated in harsh wave conditions, and accordingly, the assumptions of linear theory can be justified. Nevertheless, the development of algorithms for determining wave pressures according to other wave theories is an important task and is a direction for future research.

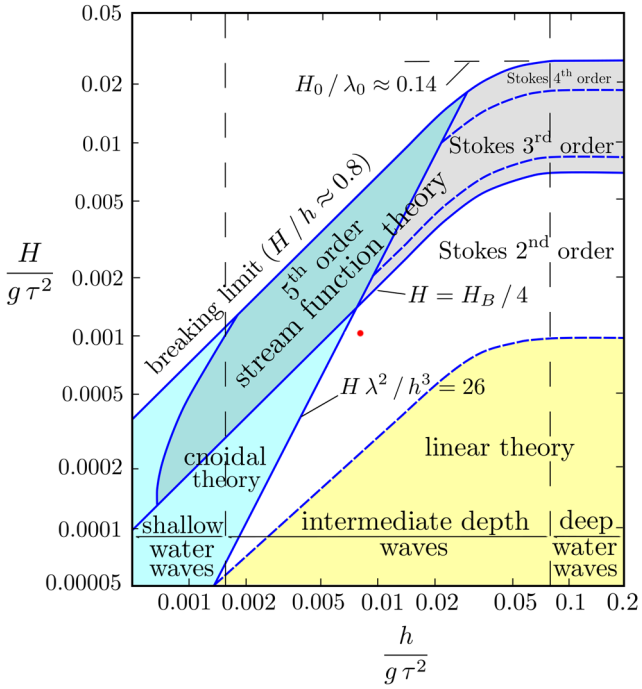


Fig. 2.33 Limits of validity for various wave theories [78]. The red dot indicated the experiment conditions in this thesis. Here h is water depth (m), τ is wave period (sec), H is wave height (m)

The nonlinear calculation resulted in ship motions more similar to the experiments, as illustrated in Fig. 2.34 (green line). The hull mesh of 78400 elements was used for simulation. In this case, the total runtime of simulation has increased to 46 seconds for the 25-second event. Since the buoyancy force is now calculated based on the mesh, the vessel keeps its deviated position from the horizontal (Roll). Moreover, the amplitude of the Roll and Heave motions converges with reasonable accuracy. This resulted in horizontal Sway motions without initial impulse, and their pattern is more similar to experiments.

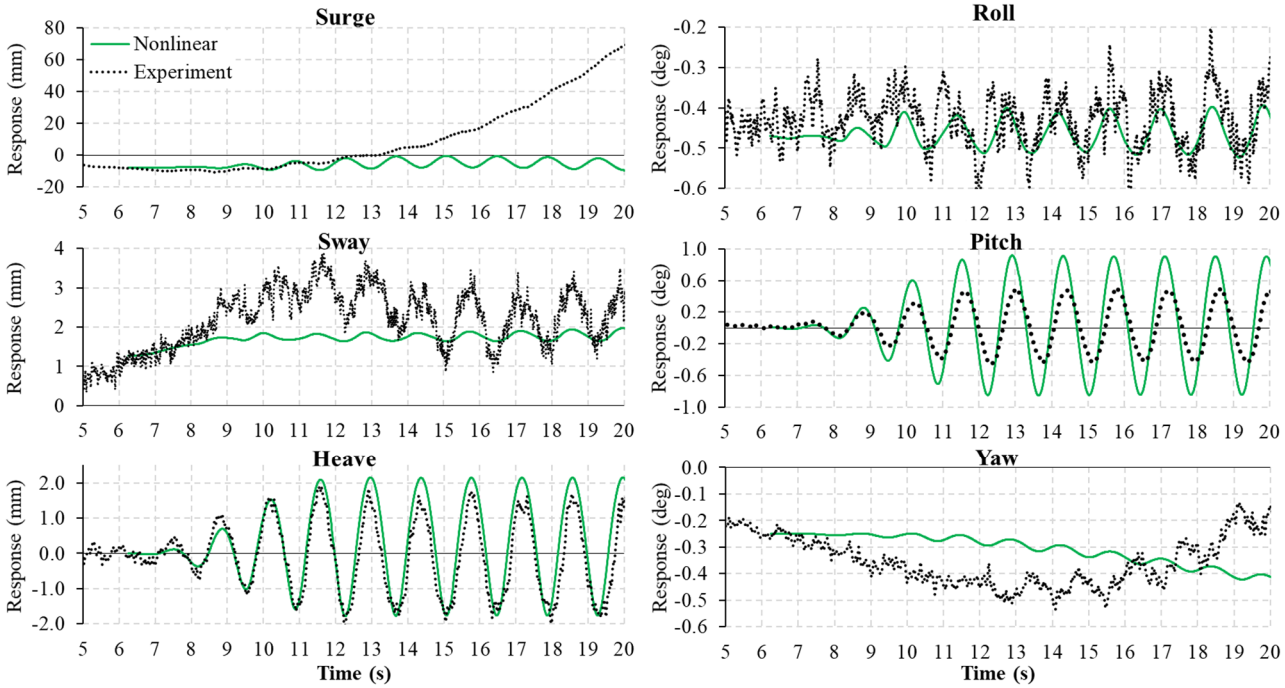


Fig. 2.34 Motions of the ship model in the nonlinear case compared to experiments in the wave basin. Wave height h = 0.02 m, wave period T = 1.4 s [71]

However, there is still a problem of the absence of vessel forward motion (Surge), which is a locked problem of inconsistent rotational Pitch motions. Obviously, the phase and frequency of vessel motions coincide with the experiment, but the damping calculated in the BEM solver is not enough to simulate the real movements of the vessel. Wave radiation does not contribute much to damping because the wave frequency is relatively small. This issue is crucial for crane ships because the natural frequencies of the cargo are not high. Consequently, the resonance effect will occur at low wave frequencies, where wave radiation is not so important. To correctly handle the dynamics of suspended load, other sources of damping are required, which dominate at low wave frequencies.

The displacements of the suspended load for the case of nonlinear calculation are presented in Fig. 2.35. Without additional Pitch damping (with radiation damping only), the load oscillation amplitude in the x-direction was too large, resulted in a complete divergence of vertical displacements (z-axis) since the cable length is finite. Consequently, in order to achieve the coincidence of the suspended load displacements with the experiment results, it is necessary to achieve the coincidence of the Pitch displacements of the crane vessel.

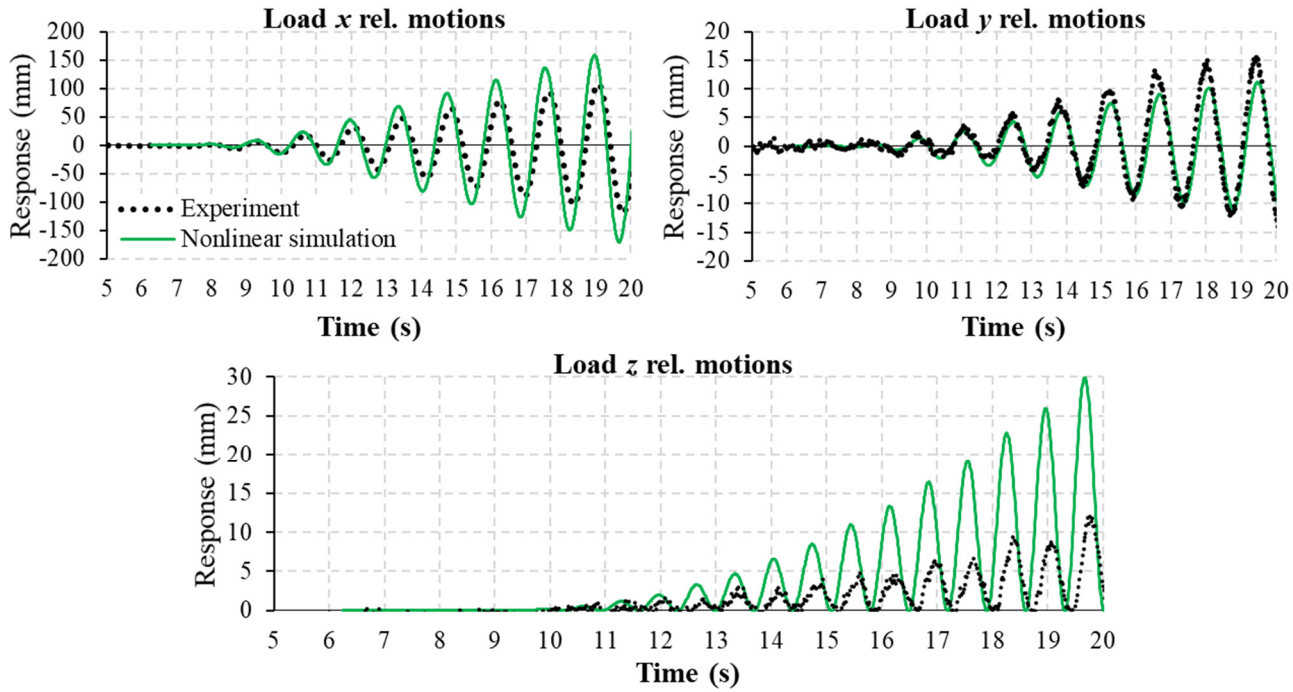


Fig. 2.35 Displacements of the suspended load relative to the upper joint in the nonlinear case compared to experiments in the wave basin. Wave height h = 0.02 m, wave period T = 1.4 s [71]

Additional damping.

As seen from the preceding results, the damping due to wave radiation is insufficient to obtain the correct motions of the vessel and suspended load. Therefore, it is necessary to consider additional damping effects that may arise from the interaction between the vessel and the wave. In a detailed consideration, it is possible to separate the damping loads caused by different physical phenomena, such as hydrodynamic drag and skin friction on the vessel hull, yaw rate drag, wave drift damping, and maneuvering load [79-81].

We will further consider only the hydrodynamic and yaw rate drag on the vessel hull, which are treated as viscous forces $\mathbf{F}_v(t)$ in Eq. (2.2). These forces can be calculated using the following expression:

$$\mathbf{F}_v(t) = -\mathbf{C}_v\dot{\mathbf{x}} - \mathbf{C}_D\dot{\mathbf{x}} \circ |\dot{\mathbf{x}}|, \tag{2.21}$$

where sign $\circ$ denotes for Hadamard product of matrices used to keep the sign (direction) of terms; $\mathbf{C}_v$ is the matrix of linear coefficients of viscous damping; $\mathbf{C}_D$ is the matrix of quadratic coefficients:

$$\mathbf{C}_D = \frac{C_d \rho A_d}{2}, \tag{2.22}$$

where C_d is the quadratic drag coefficient; A_d is the characteristic area for drag calculation; ρ is water density.

The introduction of additional damping sources is a standard practice in seakeeping simulations. Kapsenberg et al. [82] presented the results of a comparison study of different numerical simulation programs owned by class societies and research organizations. The hydrodynamics in the programs was determined on the basis of linear potential flow theory. The authors performed additional roll decay tests on a wooden model of the ship at a scale of 1: 37.89 to determine the roll damping and derived the corresponding damping coefficients for the 3rd order damping model. As a result of taking into account the additional damping, the dynamic response of the ship is considerably close to the experiments.

The equation (2.22) is well-known in CFD applications and helps to include the effects of viscosity to simulation. The input data are 6x6 matrices, taking into account the cross-coupling of degrees of freedom. In such a way the drawback of potential flow theory (non-viscous flow) can be resolved. It is also worth mentioning that the quadratic drag coefficient C_d characterizes the resistance associated with the viscosity of the fluid and is usually divided into two components – pressure resistance C_p, and frictional resistance C_f. The pressure resistance C_p is also referred to as form resistance because it is associated with the formation of the boundary layer on the surface and normal forces in contact with the fluid. The frictional resistance C_f, in turn, is related to the shear forces arising from contact with viscous fluid. Thus, the total viscous resistance is commonly characterized through these two coefficients [83]:

$$C_d = C_p + C_f \tag{2.23}$$

However, the determination of resistance coefficients is a comprehensive task, as they depend on geometry, scale, and relative velocity between the floating body and the flow around it.

The well-known characteristic dimensionless parameters of the flow are Reynolds number (Re) and Froude number (Fr), which affects the magnitude of the hydrodynamic friction coefficient. Reynolds number represents the ratio between inertial and viscous forces and Froude number represents the ratio of the flow inertia to the external field. High Reynolds numbers, are typical in ship hydrodynamics [84]. Reynolds and Froude numbers in general case can be determined as follows:

$$Re = \frac{\rho V L}{\mu} \tag{2.24}$$

$$Fr = \frac{V}{\sqrt{gL}} \tag{2.25}$$

where V is the flow velocity; L is the characteristic length; ρ and μ are the density and viscosity of the fluid respectively; g is the acceleration due to gravity.

In marine hydrodynamic applications, the aforementioned eqauations have specific expressions [84, 85]:

$$Re = \frac{\rho V_s L_{pp}}{\mu} \tag{2.26}$$

$$Fr = \frac{V_s}{\sqrt{gL_{pp}}} \tag{2.27}$$

where V_s is the ship speed and L_{pp} is the length between the between the forward and aft perpendiculars (geometric property of vessel hull).

Some values of C_p and C_f for simple geometric shapes can be found in literature and regulatory documents. Certain equations, known in the recommendations of ITTC [84], allow for the calculation of the friction coefficient C_f for surge motion:

for laminar flow conditions

$$C_f = \frac{1.328}{\sqrt{Re}} \tag{2.28}$$

for transitional flow conditions

$$C_f = \frac{0.455}{(\ln Re)^{2.58}} - \frac{1700}{Re} \tag{2.29}$$

for turbulent flow on a smooth surface

$$C_f = \frac{0.455}{(\ln Re)^{2.58}} \tag{2.30}$$

for fully turbulent flow on a rough surface with roughness height ε

$$C_f = \frac{1}{\left[1.89 - 1.62 \ln\left(\frac{\varepsilon}{L}\right)\right]^{2.5}} \tag{2.31}$$

ITTC formulation taking in account surface curvature and the pressure gradient:

$$C_f = \frac{0.075}{[\log_{10}(Re) - 2]^2} \tag{2.32}$$

The pressure coefficient C_p can be calculated through C_f and the form factor$k$:

$$C_p = kC_f \tag{2.33}$$

Nevertheless, the aforementioned methods are only an approximate solutions within certain scales and flow conditions, derived from various hull forms. Therefore, the most comprehensive solution to determining the resistance of a specific vessel is to conduct experiments. Alongside laboratory experiments, numerical simulation based on the Navier-Stokes equations, mentioned earlier in Section 1.2, is currently successfully applied. In contrast to potential flow theory, which was used in this study to determine hydrodynamic coefficients in Capytaine, the Navier-Stokes theory offers partial differential equations, taking into account turbulence, viscosity, and possible fluid compressibility.

For instance, Niklas and Pruszko [86] emphasized that determining ship resistance through experiments is a challenging task both at the experimental stage and at the step of extrapolating the results from the model scale to the full scale. They compared the results of CFD simulations with the ship's sea trial measurements and achieved a convergence of about 10%. Also, they concluded that the towing tank experiments and full-scale CFD simulations could provide similar accuracy.

In vector form, the Navier-Stokes equations are given by:

$$\frac{\partial \boldsymbol{\upsilon}}{\partial t} = -(\boldsymbol{\upsilon} \cdot \nabla)\boldsymbol{\upsilon} + \nu \Delta \boldsymbol{\upsilon} - \frac{1}{\rho}\nabla p + \mathbf{f} \tag{2.34}$$

where $\boldsymbol{\upsilon}$ is the velocity vector; t is time; ν is the kinematic viscosity of the fluid; ρ is the density; p is the pressure; $\mathbf{f}$ is the vector of mass forces; ∇ and Δ are the nabla and Laplace operators, respectively:

$$\nabla = \frac{\partial}{\partial x}\mathbf{i} + \frac{\partial}{\partial y}\mathbf{j} + \frac{\partial}{\partial z}\mathbf{k} \tag{2.35}$$

$$\Delta = \nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \tag{2.36}$$

For crane ship applications, both water and air are assumed to be incompressible, allowing uncoupling momentum and continuity equations from the energy equation [84]. Conservation of mass or so-called continuity equation is expressed as follows:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{\upsilon}) = 0 \tag{2.37}$$

However, direct utilization of the Navier-Stokes equations without any assumptions for modeling turbulent flow requires such fine grid discretization that the computational speed becomes extremely low. This approach, known as Direct Numerical Simulation (DNS), is impractical in many situations due to its computational cost. Consequently, another approach gained popularity, which adds some additional terms in the momentum equations to average turbulent effects in the flow, allowing the use of a coarser grid. In this case, the so-called Reynolds Averaged Navier-Stokes (RANS) equations are used, along with numerical analysis by RANS modeling. This method was specifically employed in this dissertation to determine quadratic drag coefficients C_D. During the CFD simulations, recommendations from two documents, which regulate the principles of constructing and tuning the numerical model, were followed:

- ITTC – Recommended Procedures and Guidelines (2014) 7.5-03-02-03 Practical Guidelines for Ship CFD Applications [84].
- ITTC – Recommended Procedures and Guidelines (2014) 7.5-03-02-04 Practical Guidelines for Ship CFD Applications [85].

The modeling was performed using the commercial solver **Ansys Fluent** with the application of the finite volume method (FVM), which is a common numerical method for solving partial differential equations in CFD applications. The geometric model with discretization for determining the surge drag coefficient is shown in Fig. 2.36. As in any numerical application, mesh discretization should be optimal. A finer mesh is required where flow features of interest exist, while a coarse mesh can be used in regions where less accurate solutions are acceptable. In our case, it is important to capture all nonlinear phenomena near the ship hull and at the water-air boundary. The degree of mesh discretization in the near-wall region is determined by the wall treatment method applied.

A two-equation eddy-viscosity k-ω SST turbulence model (Menter model [88]) with a wall function was used. The SST model combines both the standard k-ε model and the k-ω model, which is developed for resolving the near-wall region. The combination of models allows obtaining well-matched results for both the near-wall region and the flow region away from the walls.

Folllowing the abovementioned ITTC guidelines the following mesh settings were applied:

- octahedral elements were used;
- near the outer boundaries of the model, the maximum cell length was set to 0.25 m with a growth rate of 1.2;

- near the hull of the crane ship, the local element size was reduced to 0.008 m;
- A zone of local mesh refinement was designated near the model with an average size of 0.035 m.

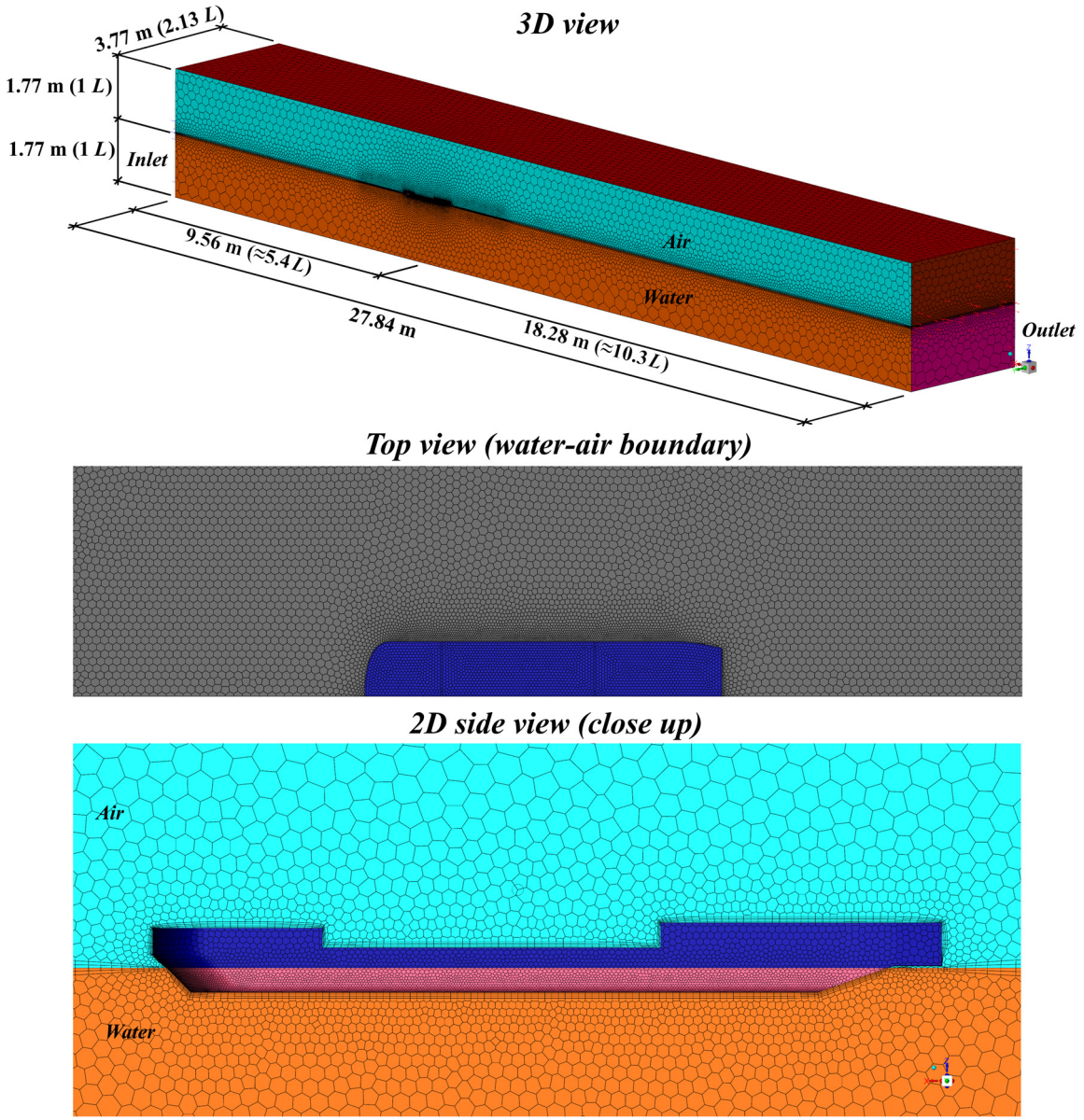


Fig. 2.36 Mesh of computational domain

The element size near the ship hull was determined by the recommended distance from the first point to the wall, denoted as y [84]:

$$y = \frac{y^+ L_{pp}}{Re\sqrt{\frac{C_f}{2}}} \tag{2.38}$$

where Re is the Reynolds number; L_{pp} is the length between perpendiculars of the hull equal to 1.63 m for current model; C_f is the friction resistance coefficient; y^+ is non-dimensional wall distance of the first point from the wall.

When using the wall function, the recommended value of y^+ is $30 < y^+ < 100$ [84, 85]. For the purposes of method development in this research, the exact determination of the drag coefficient is not the primary goal. Numerous examples in the literature demonstrate the possibility of achieving an accurate solution with the correct model settings. Therefore, for illustrative purposes, a y^+ value of 99 was chosen.

Three simulations were conducted in total to determine the drag coefficient C_D, which is part of Eq. (2.21) (not C_d in Eq. (2.22)) at flow velocities v_x of 0.05, 0.1, 0.2 and 0.3 m/s. Using Eq. (2.26) for Reynolds number and Eq. (2.32) for the friction resistance coefficient , the distance from the first point to the wall y can be calculated for a flow velocity (ship speed) of $v_x = 0.2$ m/s (not the most strict conditions):

$$Re = \frac{1000 \cdot 0.2 \cdot 1.63}{0.001} = 326000$$

$$C_f = \frac{0.075}{[\log_{10}(326000) - 2]^2} = 0.006$$

$$y = \frac{99 \cdot 1.63}{326000\sqrt{\frac{0.0055}{2}}} = 0.009\,m$$

In this regard, the wall element size was set to 0.008 m. The determination of the free surface was performed using the Volume-of-Fluid (VOF) method [89], , where the volume fraction of fluid C in a computational grid cell serves as the marker function. The evolution of the marker function is described by the convective transport equation

$$\frac{dC}{dt} = \frac{\partial C}{\partial t} + u_i \frac{\partial C}{\partial x_i} = 0 \tag{2.8}$$

The mass conservation and momentum equations for the "water-air" system in the Volume-of-Fluid method are solved as for a single medium according to Eq. (2.38), but with variable properties expressed through the volume fraction of fluid C:

$$\rho = C\rho_1 + (1 - C)\rho_2 \tag{2.39}$$

$$\mu = C\mu_1 + (1 - C)\mu_2 \tag{2.40}$$

where ρ_1 and ρ_2 denote the density of the first and second mediums, andμ_1 and μ_2 represent the dynamic viscosity, respectively.

As a result of steady-state solution for all three computational cases, the resistance force of the hull was determined, as illustrated in Fig. 2.37. A steady-state analysis involves iteratively solving

the model until convergence was achieved. The velocity field for the model with $v_x = 0.1$ m/s, depicting the flow around the hull with water and air, is shown in Fig. 2.38.

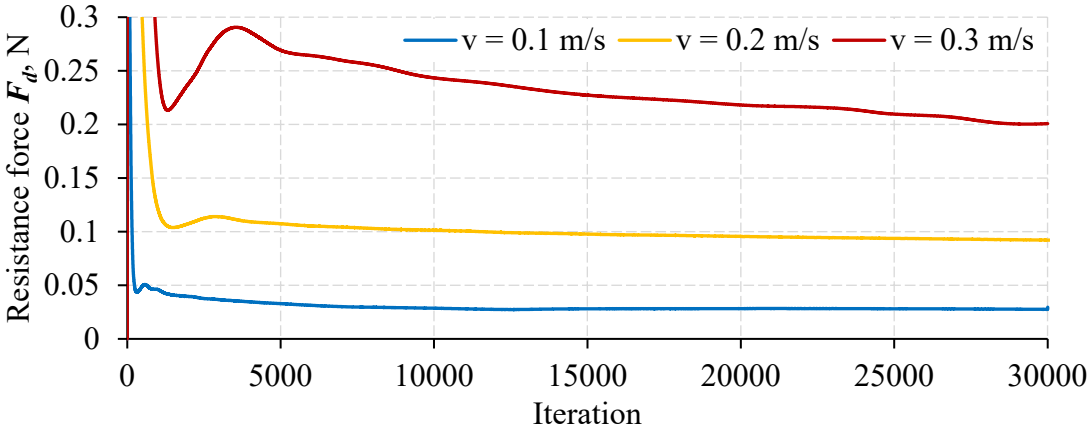


Fig. 2.37 Resistance force curves for surge flow CFD modeling

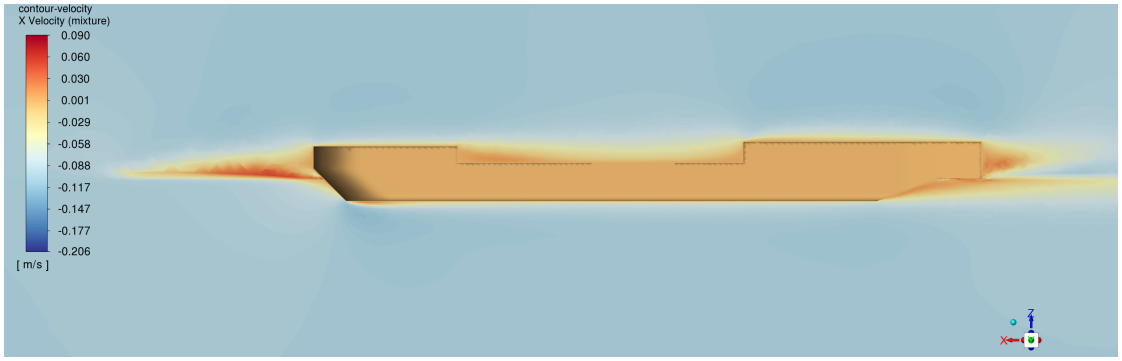


Fig. 2.38 Flow velocity field for x direction

The value of the C_D coefficient for the surge-surge component can be determined as:

$$C_D = \frac{F_d}{v^2} \tag{2.41}$$

where F_d is the resistance force at the last iteration; v is the inlet flow velocity for the particular case.

The resulting value of the C_D coefficient was determined as the slope of the linear graph of the function $F_d = f(v^2)$, illustrated in Fig. 2.34. Thus, $C_D = 2.24$.

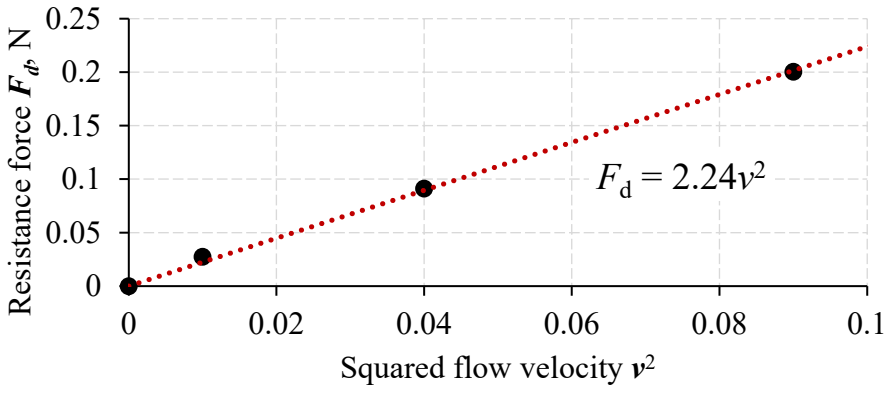


Fig. 2.39 Resistance force vs Squared flow velocity

Also, using Eq. (2.22), the quadratic drag coefficient C_d can be determined as:

$$C_d = \frac{2C_D}{\rho A_d} \tag{2.42}$$

Assuming a density ρ equal to 1000 kg/m^3 and reference area A_d equal to 1.01 m^2 (taken as wet surface area):

$$C_d = \frac{2 \cdot 2.24}{1000 \cdot 1.01} = 0.004$$

It is challenging to assess the correctness of the obtained value. However, it is worth noting that it is of the same order of magnitude as the value of 0.006 calculated earlier for the C_f coefficient using the analytical formula. Additionally, the obtained value may significantly differ from the actual value for a real model due to the roughness of the hull. In CFD simulations, the surface of the hull was assumed to be perfectly smooth, unlike the model tested in the wave basin. Nevertheless, CFD simulations using the RANS approach serve as a valuable tool to complement the crane ship model with damping coefficients, thus mitigating the drawbacks of potential flow theory, which forms the basis for determining hydrodynamic forces. With this combination of CFD methods, it is possible to maintain the capability for rapid modeling of various sea conditions (wave directions and periods) and achieve high computational accuracy.

The same CFD approach was applied to determine the Pitch-Pitch C_D coefficient, which is also significantly affects the suspended load motions because of the zero-wave heading angle. The resulting value of C_D is 6500.

The changes in the dynamics of the ship and suspended load due to the additional damping are also shown in Fig. 2.40.

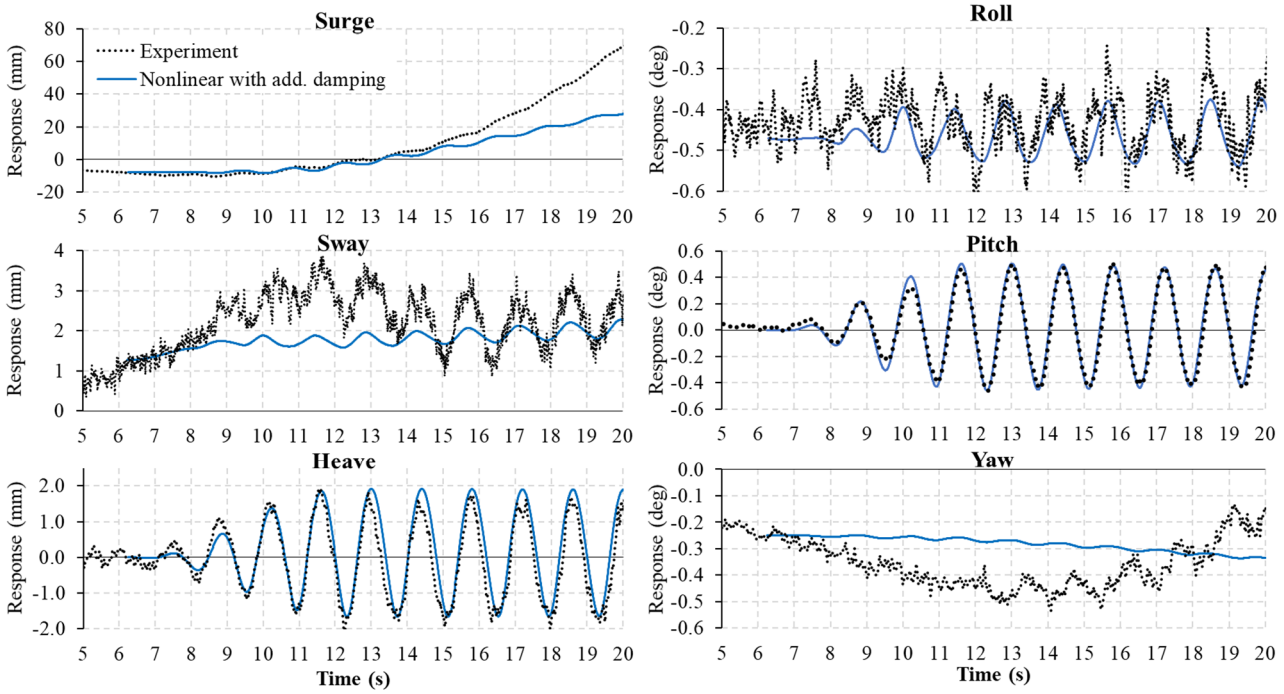


Fig. 2.40 Motions of the ship model in the nonlinear case with additional viscous damping compared to experiments in the wave basin. Wave height h = 0.02 m, wave period T = 1.4 s [71]

The introduction of additional Pitch damping contributed to the complete coincidence of the simulation and experiment for this degree of freedom. In addition, small improvements are observed for Sway, Heave, and Roll motions. It was finally possible to achieve a coincidence of the Surge motion of the numerical model with the experiment for up to 15 seconds. However, further simulations diverge, possibly due to the absence of second-order forces or another nonlinear terms. The coincidence of the Pitch motions of the crane ship resulted in a complete match of the payload displacements in the x-axis direction, as shown in Fig. 2.41. The displacements in the y and z axes also converged well. The maximum errors between experiment and simulation with respect to the maximum amplitude: 4.3% for Heave, 6.9% for Pitch, and 5% for suspended load x displacements.

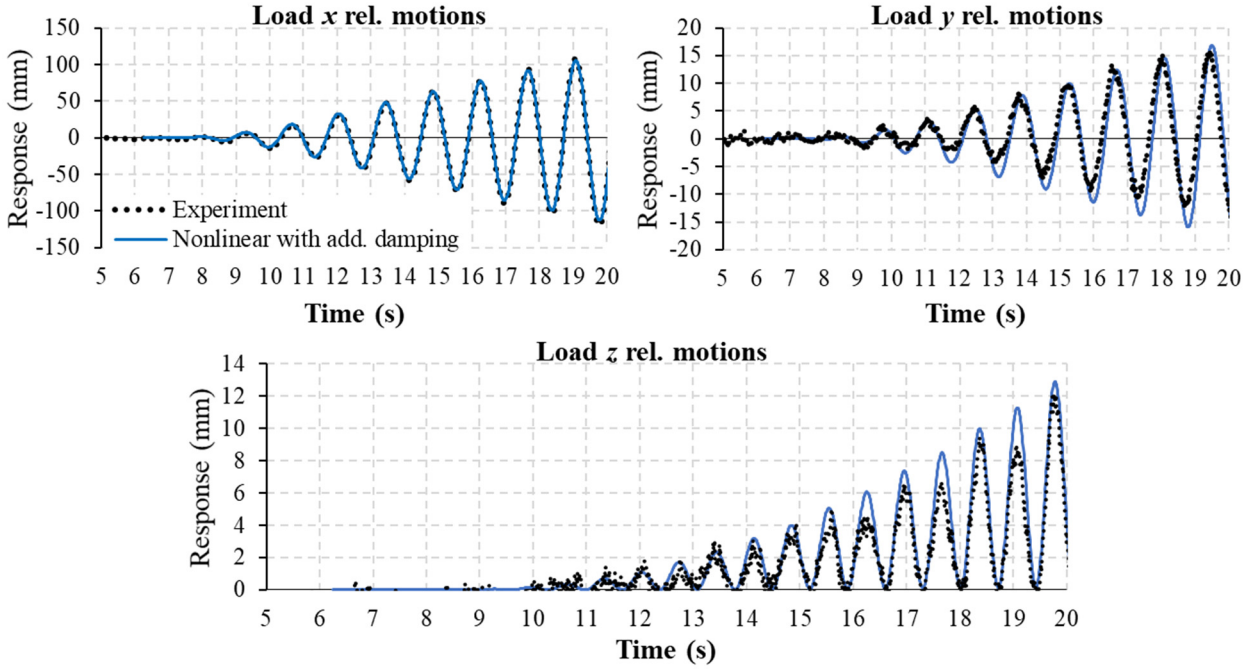


Fig. 2.41 Displacements of the suspended load relative to the upper joint in the nonlinear case with additional viscous damping compared to experiments in the wave basin. Wave height h = 0.02 m, wave period T = 1.4 s [71]

Mean drift force.

Mean drift force is an additional wave force acting on the floating body. In contrast to the wave excitation force $\mathbf{F}_{exc}(t)$, the drift force is a second-order force and is defined as

$$\mathbf{F}_{md}(t) = \left(\frac{h}{2}\right)^2 \mathbf{F}_{md}(\omega, \theta) \tag{2.43}$$

where h is the wave height and $\mathbf{F}_{md}(\omega, \theta)$ is the vector of mean drift force coefficients.

Coefficients $\mathbf{F}_{md}(\omega, \theta)$ can be also determined in Capytaine during solving the boundary problem. Nonlinear drift forces can contribute significantly because they include low-frequency components that might excite the slow drift motion of the vessel. Their contribution will be most noticeable in irregular waves affecting from different directions. In present case, the direction of the

wave did not change. Therefore, the drift forces' contribution to the vessel's dynamics was significant only for Surge motion (Fig. 2.42) and did not affect the suspended load.

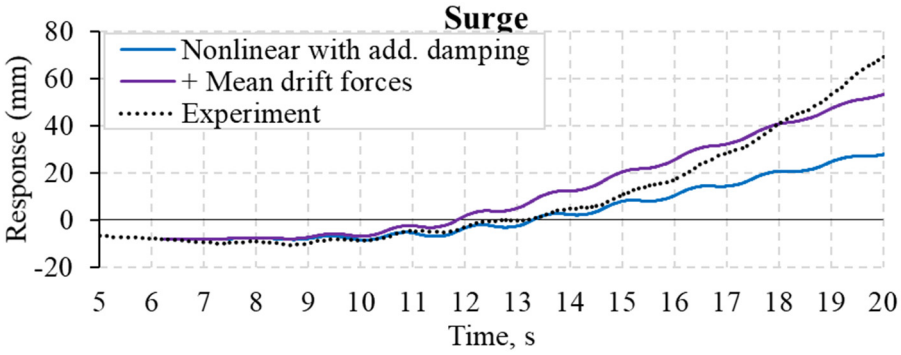


Fig. 2.42 Contribution of the mean drift force to the Surge displacements of the crane vessel [71]

Mesh dependency.

The apparent drawback of the nonlinear approach is that the accuracy of force determination depends on the mesh size, which in turn affects the calculation speed. Thus, it becomes necessary to conduct a mesh influence test to find the optimal mesh size that would ensure both computational accuracy and maintain calculation speed above real-time, which is one of the objectives in this thesis. For this purpose, five simulations were performed with the same conditions but with different numbers of ship hull panels. Simulation results with the following average panel sizes were analyzed: 8, 10, 12, 15, and 20 mm. It was found that the vessel's response in the Surge, Heave, and Pitch DOFs was almost unchanged. The difference in displacement values was less than 1%.

However, substantial changes were observed for the degrees of freedom Sway, Roll, and Yaw, as can be seen in Fig. 2.43. A known Roll-Sway-Yaw lock is observed, which is not coupled to Heave-Pitch because the submerged part of the ship is a symmetric port-starboard. The changes in the results were irregular, without a clear trend of increasing or decreasing accuracy with changes in the mesh size. This indicates an imbalance of forces on the port and starboard sides of the vessel due to the possible asymmetry of the mesh, which causes uncontrolled yawing. As a consequence, Sway motions and rolling also appear.

As for the calculation speed, it increased linearly from 46 seconds for 25-second event with 8mm panels size (74400 panels in total) to 8 seconds with 20mm panels size (13012 panels in total). In this way, the calculation speed exceeding the real-time was achieved with 12 mm panels. The computations were performed on a PC with an Intel Core i9-12900 CPU.

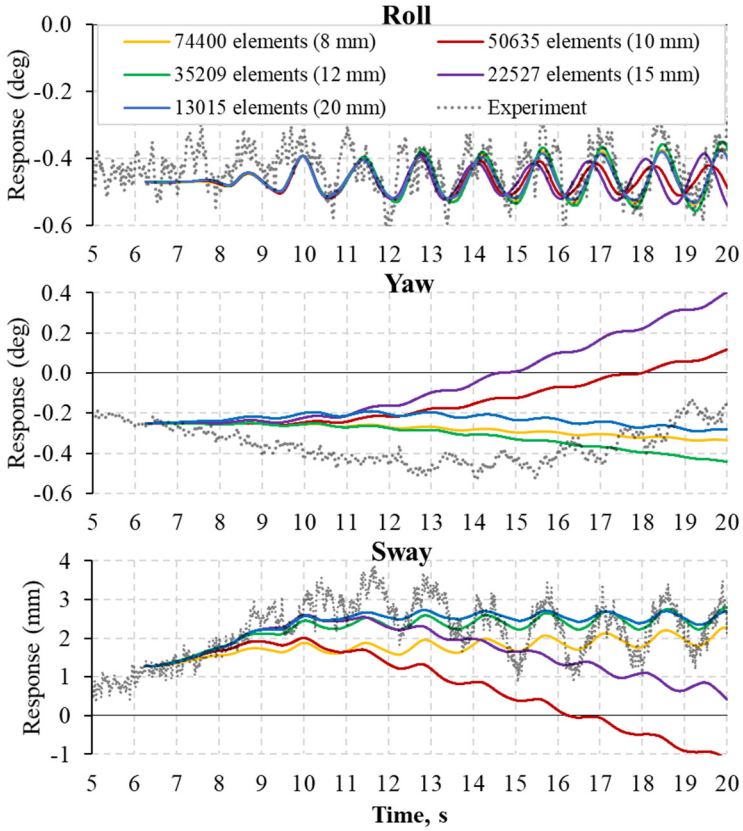


Fig. 2.43 Roll, Yaw, and Sway motions for models with different mesh sizes [71]

2.4. Section conclusions

In this section an approach for modeling the time-domain response of a crane vessel was described and tested. The development of the method is aimed at achieving high calculation speed, acceptable accuracy, and maintaining scalability and accessibility. The tested method involves preliminary frequency-domain analysis using the potential flow Boundary Element Method (BEM) solver Capytaine, followed by time-domain analysis in the MATLAB/Simulink environment using the WEC-Sim software. A small contribution was also made to the development of the WEC-Sim software by pointing out on the problem of determining the diagonal terms of the inertia matrix [72, 73].

In summary, the considered method of modeling crane ship dynamics showed promising results. It allows the dynamics of the crane ship and suspended cargo to be modeled with acceptable accuracy. However, the basic linear calculation of hydrodynamic forces is not sufficient. The results indicated that determining restoring forces based on a linear calculation is unacceptable for asymmetric models and where the COG position changes during the simulation. Moving to a nonlinear wave and buoyancy force calculation based on the tracking the position of submerged mesh elements greatly improves the results. Moreover, obtaining an appropriate solution is possible with a non-fine mesh, which keeps the computation speed high. In our case, real-time computing speed was achieved by using 12 mm mesh. The introduction of additional damping, which can be determined

by CFD modeling, also appeared to be necessary. This method is already actively used in engineering practice, but it is necessary to take care with setting up the parameters of numerical models. Simulations using Reynolds-Averaged Navier-Stokes (RANS) equations serve as a valuable tool to complement the model of the crane vessel with damping coefficients, thereby reducing the limitations of potential flow theory, which forms the basis for determining hydrodynamic forces. With this combination of Computational Fluid Dynamics (CFD) methods, it is possible to maintain the capability for rapid modeling of various sea states (wave directions and periods) while achieving high computational accuracy.

Considering all the described enhancements to the model, the resulting equation of motion for the crane vessel as a floating body can be expressed as follows:

$$\begin{aligned}(\mathbf{M}+\mathbf{A}_{\infty})\ddot{\mathbf{x}}(t)+\rho g\iint_{S} z(t)dS+\mathbf{C}_{v}\dot{\mathbf{x}}(t)+\mathbf{C}_{\boldsymbol{D}}\dot{\mathbf{x}}\circ|\dot{\mathbf{x}}|+\int_{0}^{t}\mathbf{K}_{\boldsymbol{r}}(t-\tau)\dot{\mathbf{x}}(\tau)\mathrm{d}\tau=\\ =\Re\left[R_{f}(t)\frac{h}{2}\mathbf{F}_{exc}(\omega,\theta)e^{i\omega t}\right]+\mathbf{F}_{FK}^{nonlin}+\left(\frac{h}{2}\right)^{2}\mathbf{F}_{md}(\omega,\theta)+\mathbf{F}_{ext}(t)\end{aligned} \tag{2.44}$$

Although other wave directions have not been tested in the experiment, we believe that in more complex wave conditions, the non-diagonal components of the damping matrix would play a significant role. Cross-coupled Roll-Sway-Yaw motions were also observed.

As for the yaw of the ship, on the one hand, it was not possible to achieve convergence with the experiment. This may be due to inaccurate determination of the positions of the centers of gravity since they were determined using CAD tools and may differ slightly from the physical model. On the other hand, the absolute value of Yaw in the experiment did not exceed 0.5 deg, which is not too significant.

Among the reasons for the mismatches, we can single out the fact that in the wave basin, the vessel was affected by a not perfectly regular wave in contrast to the numerical experiment. Also, all wave effects are calculated based on Airy wave theory. Therefore, the improving the software by adding the ability to simulate Stokes waves should be considered in future. Also, the influence of the mooring system used in laboratory experiments should be checked carefully. We assumed that its contribution can be neglected in the presented case, but experiments with more complex wave conditions can clarify this.

Nevertheless, the motions of the suspended load agree quite accurately with the experiments in the wave basin, which demonstrates the possibility of using this method for implementation in real practice. For instance, the approach can be used in combination with integrated systems on the ship to correct the ship's motion in real-time and create a digital twin.

The flowchart illustrating the developed methodology for modeling the dynamic response of the crane vessel and suspended load is shown in Fig. 2.44, where $\boldsymbol{x}$, $\dot{\boldsymbol{x}}$ and $\ddot{\boldsymbol{x}}$ are the displacement, velocity, and acceleration vectors, respectively. The subscripts S and L refer to the *ship* and the suspended *load* respectively.

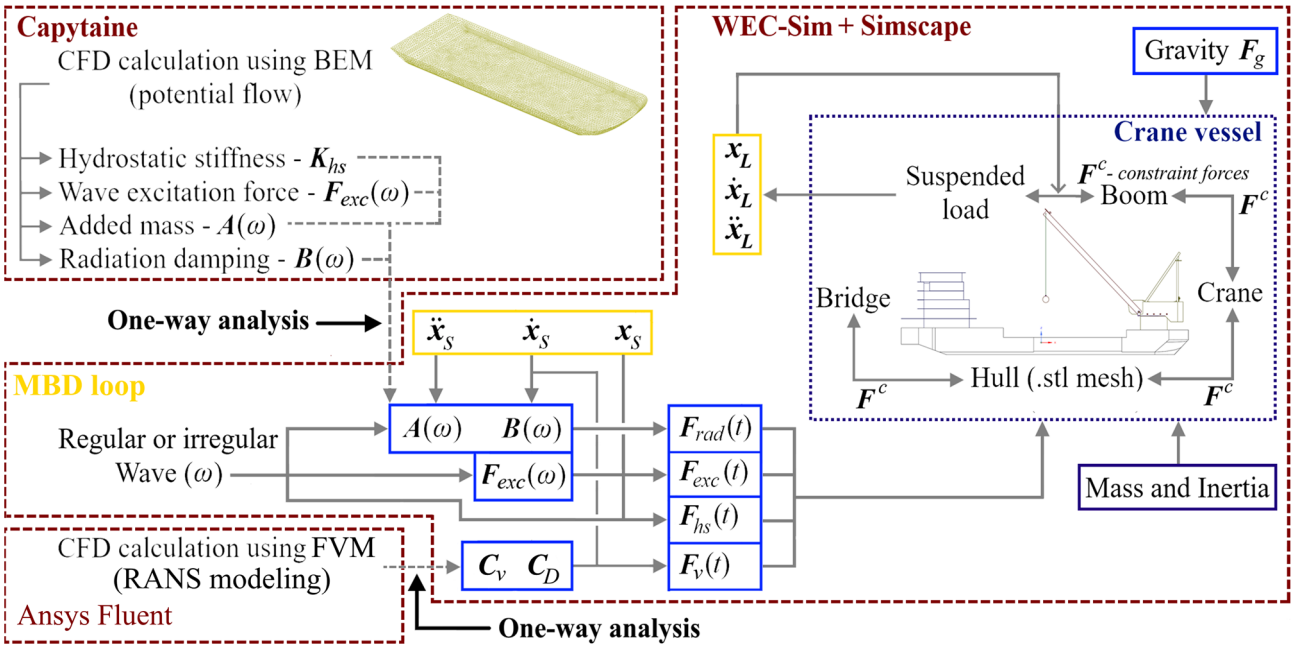


Fig. 2.44 Schematic diagram for calculating the dynamic response of a crane ship and suspended load [90]

3. Interactive modeling of crane vessel

3.1. Interactive control implementation

The method of modeling the dynamics of a crane ship described in Section 2 represents the initial significant step in this study. Through this method, it becomes possible to quickly and conveniently model a crane ship. However, to account for real operating scenarios, a methodology for controlling the crane's components to carry out loading and unloading operations needs to be developed. In Subsection 1.3, a review of other authors' research was conducted, where they implemented the modeling of loading operations. Primarily, the reviewed works implemented scenario-based control, which involves predefining crane and cargo movements before initiating the calculation. Obviously, it is not applicable in real-time applications, such as digital twins, interactive simulations (VR and AR), and other modern smart solutions and developments, including unmanned ships and artificial intelligence-driven (AI-driven) systems.

In connection with this, the next step in this research is the implementation of interactive control, allowing real-time control of the model through input signals from the user or another automated system. The conceptual difference between scenario-based control and interactive control is presented in Fig. 3.1. Potentially, such an approach has a broader range of applications in engineering practice.

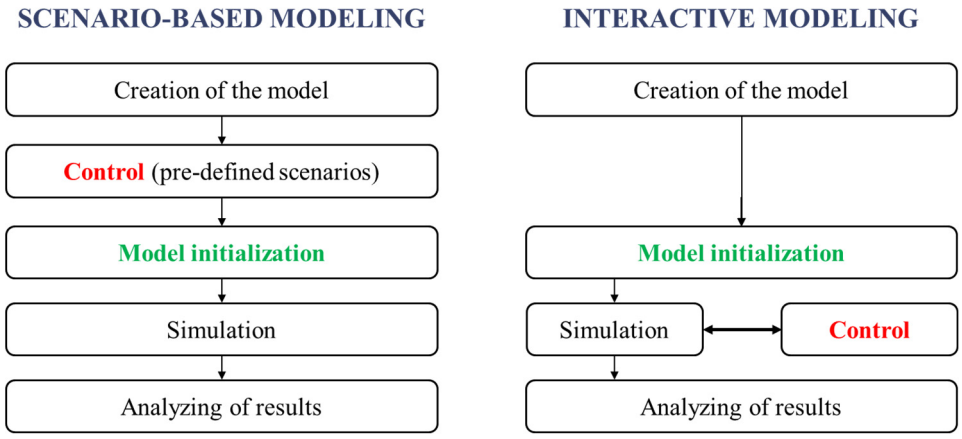


Fig. 3.1 Modeling algorithms with of scenario-based control and interactive control

An evident feature of interactive control is that it cannot be implemented without real-time computations, enabling the model to respond instantly to user input. Moreover, the control itself should not significantly slow down the calculation to prevent real-time synchronization issues with the virtual environment. Thus, this chapter aims to implement the capability of interactive control during the cargo transfer stage and extend it to the lifting and lowering stages.

To implement interactive control, the previously developed numerical model of the crane ship with a suspended load requires modification. The model in the MATLAB/Simulink environment was enhanced by replacing the rigid connection of the crane base to the hull and the crane boom with

revolute joints, releasing one degree of freedom (Fig. 3.2). Also, to be able to lift and lower the load, it is necessary to replace the rigid cable with a constant length by a cable with a variable length. For this purpose, we used the library of tools “Belts and Cables” included in Simscape Multibody tools. They allow to model belt-cable circuits consisting of spools, pulleys and cables.

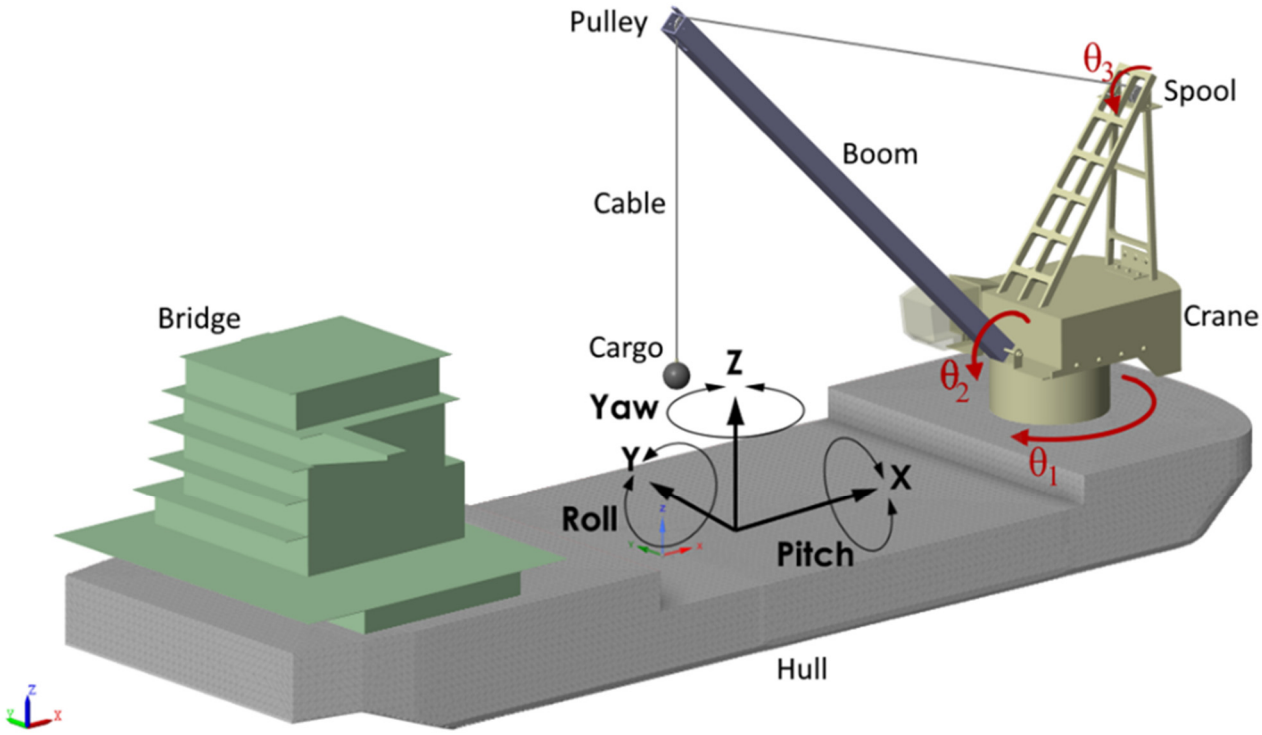


Fig. 3.2 Crane ship as a multibody system with controllable parts [93]

On the top part of the crane base, a cable spool was positioned, connected to the crane itself through a revolute joint. Additionally, the spherical joint in the previous model was replaced with a cable pulley, which is also connected to the crane boom via a revolute joint. The cable is attached to three blocks in the model – the spool, pulley, and rigid hook of the suspended load. The length of the cable changes as the revolute joint connecting the spool with the crane base rotates. This way, the entire hoist system was integrated. The mass of the pulley and spool was assumed to be 0.001 kg to exclude their influence on the dynamics of the floating crane. In this setup, the cable is considered a massless, inextensible, and always taut cord, which is an additional assumption. The block diagram of the hoisting system in the Simulink environment is presented in Fig. 3.3.

On whole, the multibody system in current implementation consists of 9 rigid bodies which motion in total is defined by 54 generalized coordinates. Equation of motion includes 44 algebraic constraints, so the system has 10 degrees of freedom. Inertia properties of the model are presented in Table 3.1. The numerical model of crane ship as a block diagram in the Simulink environment is shown in Fig. 3.4.

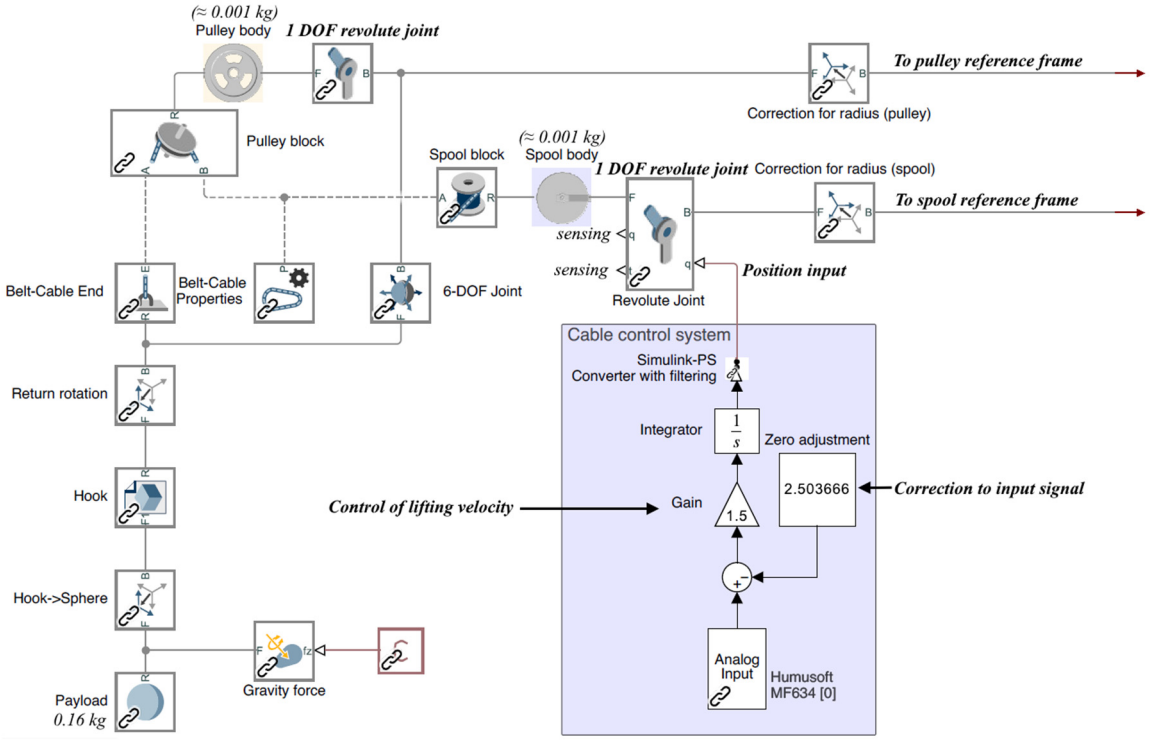


Fig. 3.3 The block diagram of hoisting system in Simulink environment

Table 3.1 Inertia properties of multibody system with controllable parts

Body	Mass (kg)	Principal moments of inertia (kg·m^2)			Center of gravity coordinates (m)		
		I_{p1}	I_{p2}	I_{p3}	x	y	z
Hull	21.4	0.541	5.580	6.052	0	0	0.015
Lead weight	9.76	0.044	0.007	0.043	0.656	0	0.010
Bridge	9.28	0.255	0.175	0.275	-0.590	0.013	0.254
Crane	3.15	0.055	0.058	0.019	0.765	0.003	0.269
Boom	0.56	0.0001	0.033	0.033	0.313	0	0.483
Spool	0.001	8.424e-08	1.444e-07	1.678e-07	0.890	0	0.504
Pulley	0.001	5.140e-08	1.015e-07	5.143e-08	0.038	0	0.767
Hook	0	0	0	0	0.028	0	0.291
Suspended load	0.16	2.310e-05	2.310e-05	2.310e-05	0.028	0	0.267

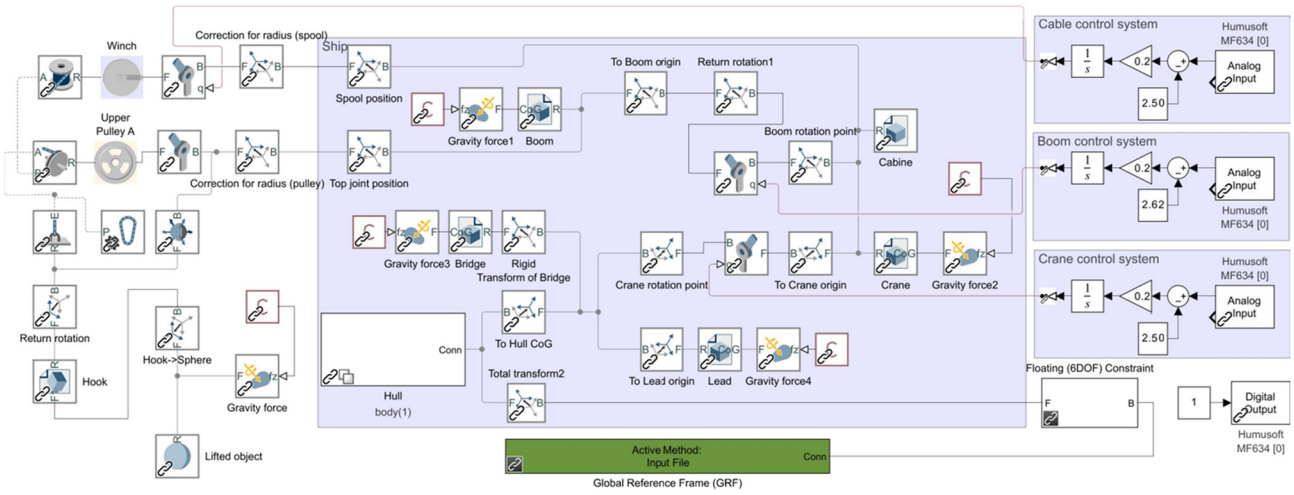


Fig. 3.4 Block diagram of the numerical model in Simulink environment [90]

By actuating all revolute joints in the model, it becomes possible to interactively control it. The actuation input in current implementation is the position provided as analog signals form hardware connected to PC. As for the hardware part the HUMUSOFT MF634 I/O card was installed in PCI Express slot of PC for data acquisition. The model was controlled by the user via 2-DOF joysticks connected to the screw terminal, which transferred a signal to the I/O card as presented in Fig. 3.5.

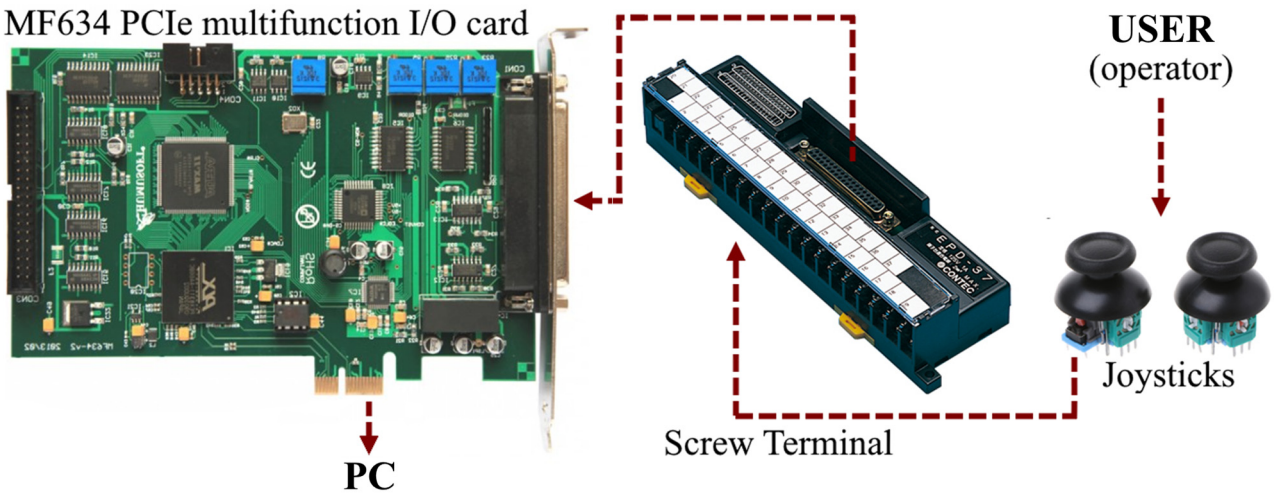


Fig. 3.5 Hardware used in interactive modeling

Connection of the terminal block to the I/O board is done using a cable with 37 pin D-sub connector plugs (Fig. 3.6). Joysticks are connected to the terminal block by simply clamping the wires on the lines VRx, VRy (analog signals for *x* and *y* axes of the joystick), as well as GND (ground) and +5V (power supply).

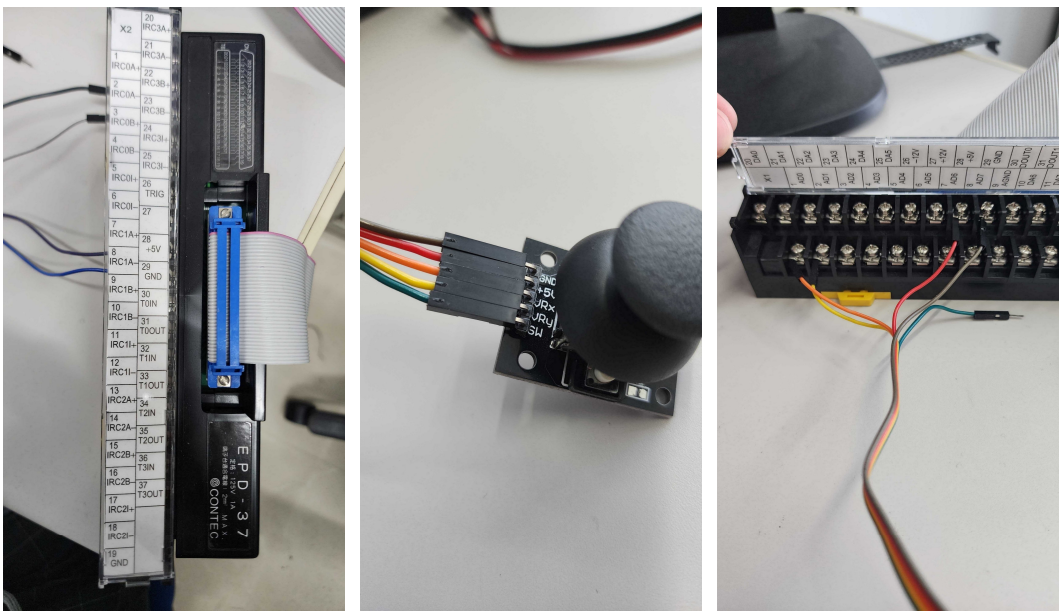


Fig. 3.6 Connection of joysticks to screw terminal

Software part of control system is implemented as Simulink blocks named “Analog input” as illustrated in Fig. 3.3 and Fig. 3.4. These blocks are part of the Desktop real-time Toolbox in MATLAB [91]. After reading the input signal from the I/O card, it is corrected by adding a numerical

value to it (2.50 or 2.62 in Fig. 3.4) and multiplying the result by 0.2. These values were manually adjusted. Thus, in the neutral position of the joysticks, the signal magnitude is 0, and the speed of crane parts' movement is acceptable. Since the input signal has a constant value when the joystick position is unchanged, it requires processing with an integrator to provide an increasing value of the desired rotation angle of the revolute joints. The final value applied to the revolute joint position (rotation angle) θ_a is determined by the following expression:

$$\theta_a = k_1 \left(k_0 + \int S dt \right) \tag{3.1}$$

where k_1 is velocity control constant equal to 0.2; k_0 is zero-correction constant equal to 2.50 or 2.62 depending on the joystick; S is the signal provided by I/O board to Simulink; t is a time.

The final signal undergoes second-order filtering with a time constant equal to 0.01 s to ensure smooth movement. The process of interactive control of the numerical model of the floating crane is illustrated in Fig. 3.7.

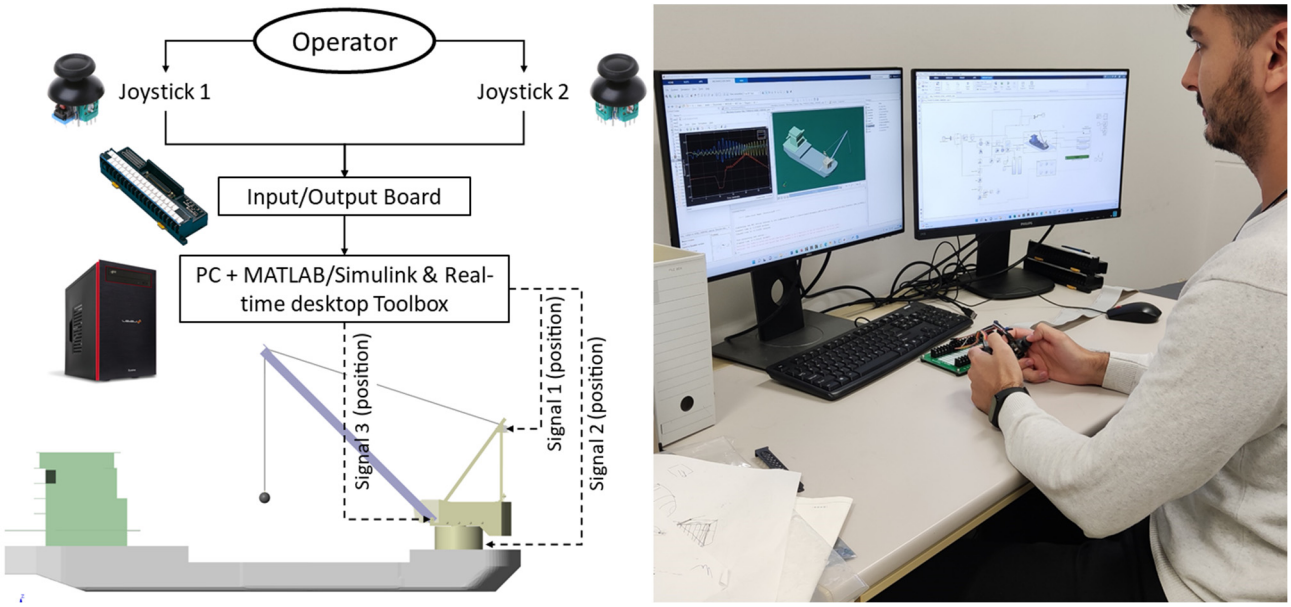


Fig. 3.7 Interactive control of the model by the operator during the simulation [93]

The first attempts of simulation with described configuration were not successful. The reason was that simulations with the ode4 fixed-time solver (fourth-order Runge-Kutta) did not achieve calculation convergence at the selected time step of the same constant size (0.01 s). If the position of the crane elements changed too quickly, the computing slowed down a lot and increased again when the position remained static. Therefore, the solver was changed to ode45 type. It is a variable time step solver, which uses a pair of explicit Runge-Kutta formulas of orders 4 and 5 (Dormand-Prince pair) for numerical integration. After this modification the convergence problem was resolved.

In the simulated case, the change of position of the ship's parts occurred sequentially. The response of the vessel in the simulated event is shown in Fig. 3.8. From a neutral position at 1.3 s

(time moment 1), the ship's crane began to rotate and was eventually rotated by 95.9 degrees (time moment 2). Due to the change in position of the centers of gravity of the crane, boom, and suspended load, an additional roll of the vessel occurred. The model reached a new neutral position, as determined by a non-linear calculation of the restoring forces. Then, at simulation time 9 s, the crane boom was moved, which also resulted in inverse rolling (time moment 3). At the time moment 4, the crane's boom changed its position again. The displacement of the boom center of gravity from the vessel longitudinal axis resulted in a progressive yawing of the vessel. Therefore, nonlinear computation of wave excitation forces described in subsection 2.3.3 is important, even if there is no manual control of the vessel's rotation (e.g., control of motor).

Another interesting result is the timestep size variation during the computation process. Fig. 3.9 illustrates the distribution of the timestep size during the simulation. It can be seen that a significant part of the computation was performed with a timestep size from 5.7×10^{-4} s to 1.6×10^{-3} s. Hence, the solver sometimes automatically reduced the timestep in order to resolve the complexity of the problem.

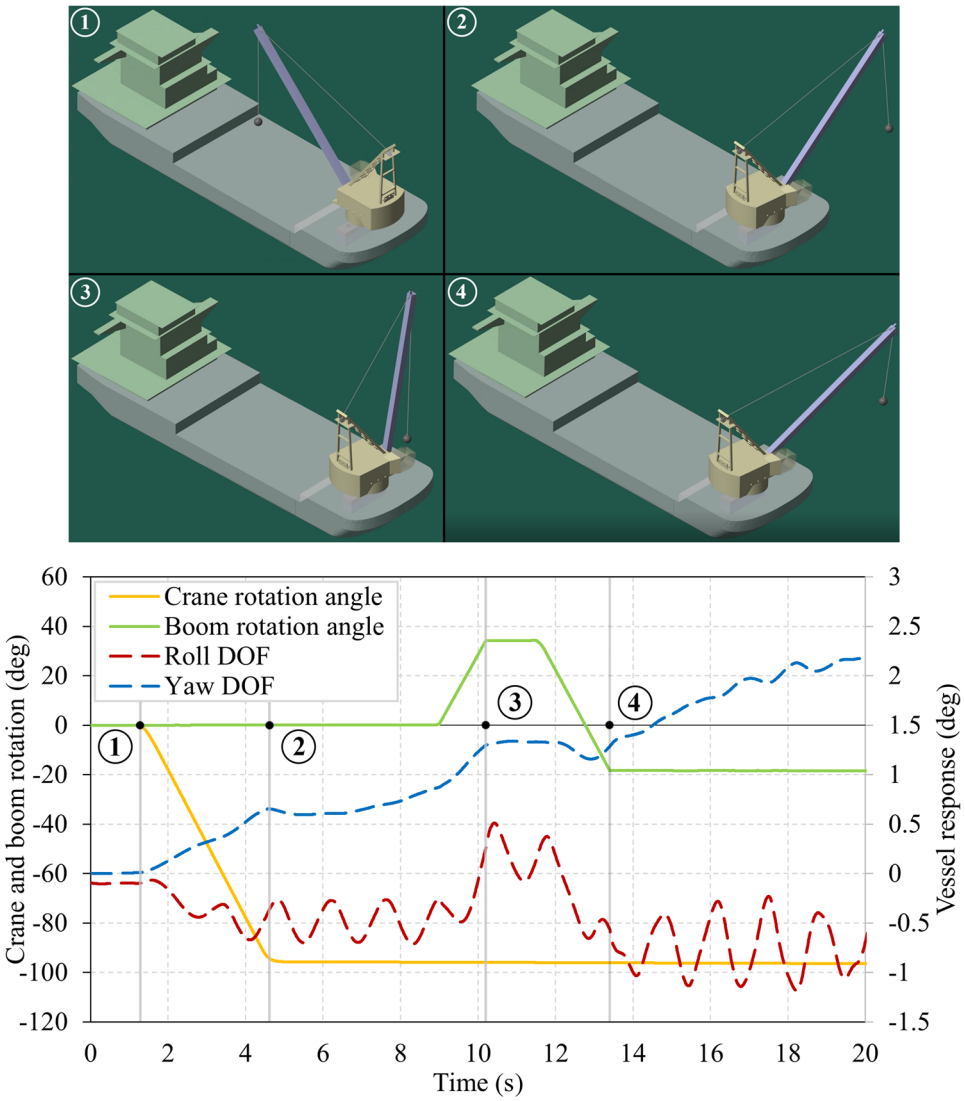


Fig. 3.8 The dynamic response of the crane ship model during the interactive control process [90]

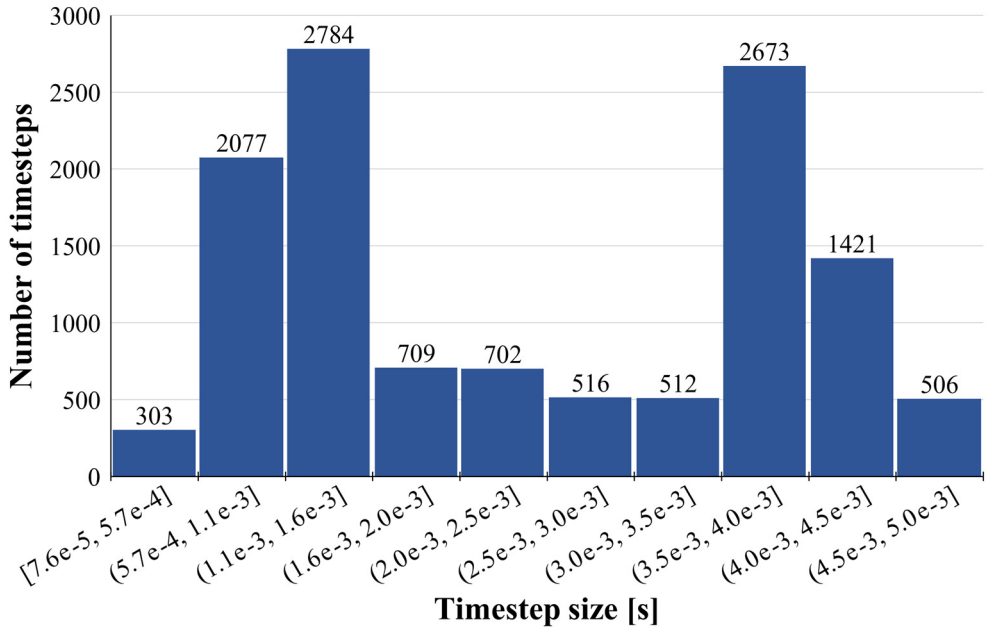


Fig. 3.9 Distribution of the timestep size during the simulation [90]

The total number of calculation steps of the 30 s event was 12203, which is significantly more than 6000 steps when using the ode4 solver with a fixed timestep in the previous implementation without interactive control. During the simulation, some missed ticks occurred due to the complexity of the model. This can easily be fixed, for instance, by using a simpler ship hull mesh. The total elapsed time in the case of the interactive crane control turned out to be 24.9 sec, which leads to a ratio of event time/elapsed time equal to 1.2 when using a conventional PC with Intel Core i9-12900K CPU (16 cores).

Hence, the presented numerical model in MATLAB Simulink allows the interactive simulation of a lifting operation with controlled crane and boom orientation and cable length preserving real-time computing and having sufficient calculation accuracy of the floating vessel dynamics.

3.2. Interactive control during cargo lift-off and installation

In the conducted test of interactive modeling of the crane ship, only the stage of cargo transfer in the air was considered. While the results were promising, the current configuration of the numerical model is unsuitable for modeling other stages of maritime loading operations. This limitation arises from the fact that the cable in the Simscape library is always taut, which can be considered as a constant-distance constraint. Thus, it cannot reproduce the loosened state that occurs when the load is lowered onto the surface. Implementing other stages, such as cargo lift-off and installation, requires an enhanced approach where the cable can change its state from taut to loosened, and the cargo can interact with the surface (Fig. 3.10). Accordingly, the previously described approach to interactive modeling of the crane ship has been supplemented and improved.

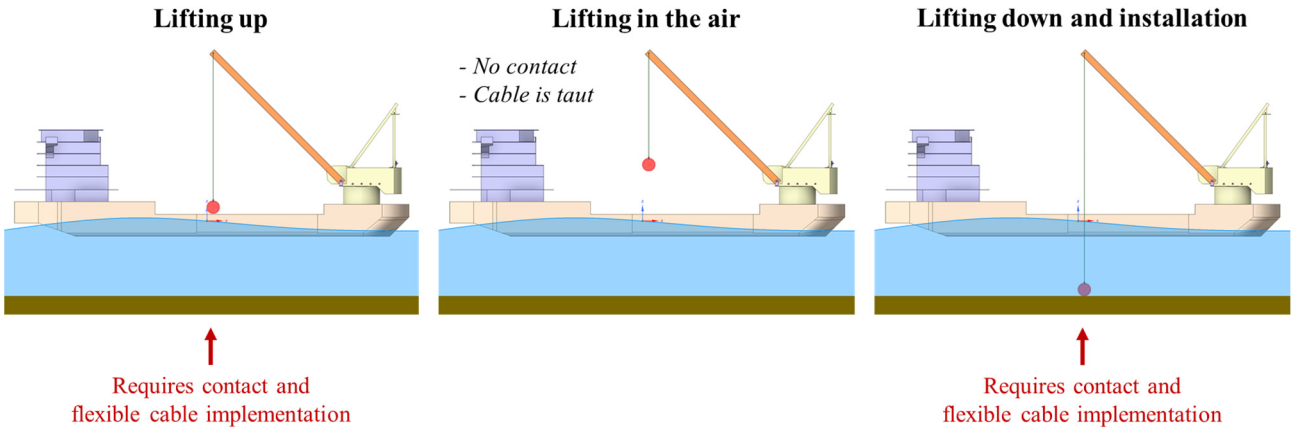


Fig. 3.10 Stages of marine crane ship operations

The first improvement was the introduction of the ability to interact between rigid bodies in the numerical model. The collision of bodies is handled via a penalty-based contact [92]. For this purpose, a Spatial Contact Force block is used, which links the geometry of the contact bodies to each other (Fig. 3.11) and simulates the interaction force when contact is detected.

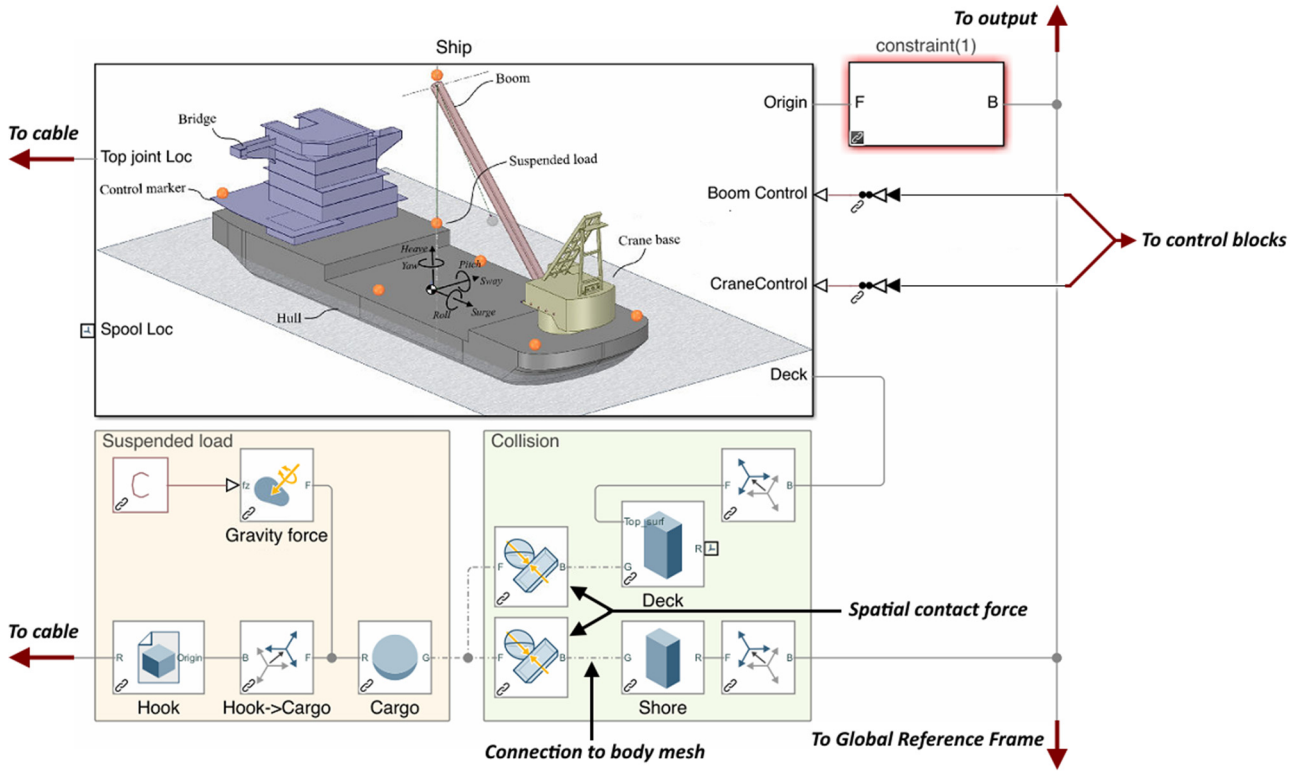


Fig. 3.11 Spatial Contact Force blocks connecting suspended load to ship deck and shore

This contact model allows the bodies to penetrate into each other and the force proportional to the penetration depth and speed is applied to resist. The advantage of this technique is the high computation speed and simple parameter setting. Then, the normal contact force f_n is computed similarly as the force in an elastic-damped element depending on the penetration value and the frictional force f_f is treated according to Coulomb friction model (Fig. 3.12):

$$f_n = kd + c\dot{d} \tag{3.2}$$

$$|f_f| = \mu |f_n| \tag{3.3}$$

where k is the normal-force stiffness, taken as 10 000 N/m; c is the normal-force damping, taken as 10 N/(m/s); d is the penetration depth between two contacting geometries equal to 0.1 mm; μ is the friction coefficient assumed equal to 0.5 as for concrete-steel interaction.

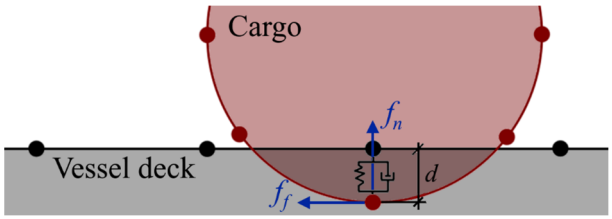


Fig. 3.12 Penalty-based contact [93]

To highlight the problem of using a rigid cable, a simulation was performed in which a load was lowered onto the deck of a ship (Fig. 3.13). At the moment of contact of the spherical load with the surface, instead of transitioning to a relaxed state, the cable instead began to transmit force to the deck through the contact surface, which resulted in unphysical behavior of the model.

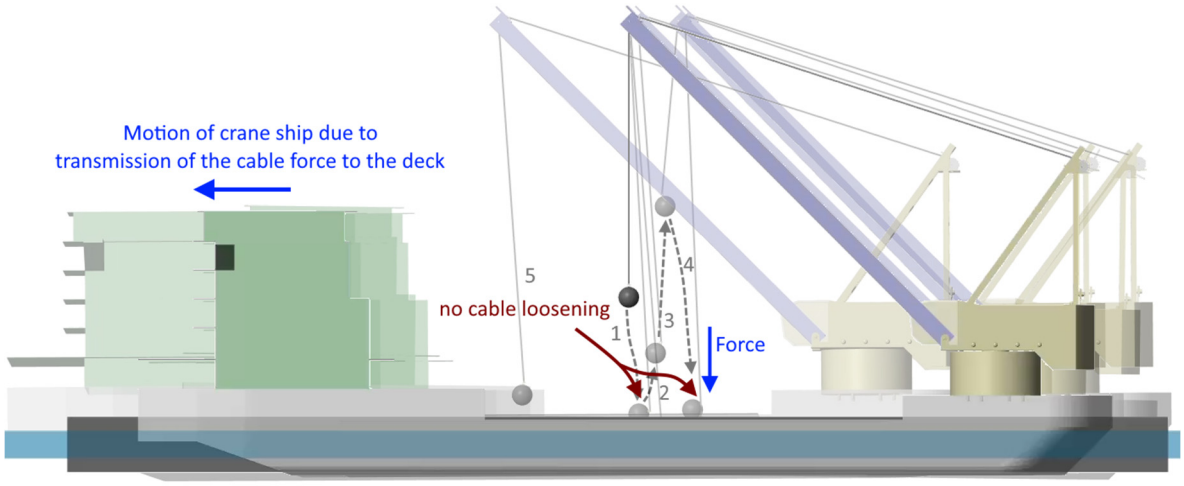


Fig. 3.13 Unphysical behavior of crane ship model with rigid cable during contact event

Therefore, the next improvement is the introduction of flexible cable, which is a part of the hoisting system of a crane. Flexible cables can be modeled using different approaches (Fig. 3.14). For example, the *lumped-parameter method*, in which a cable is modeled as a multibody system consisting of many rigid bodies and joints, can be distinguished. Another way is to use the *finite element method* to model a cable as a flexible deformable body discretized into elements. These methods either greatly slow down the calculations (*lumped-parameter method*) or make it difficult to change the neutral length of the cable (*finite element modeling*). There are of course other techniques available in the literature. However, in order to achieve the objectives of this research another approach based on implicit modeling of the cable by constraint equations allowing for interactive length control was developed.

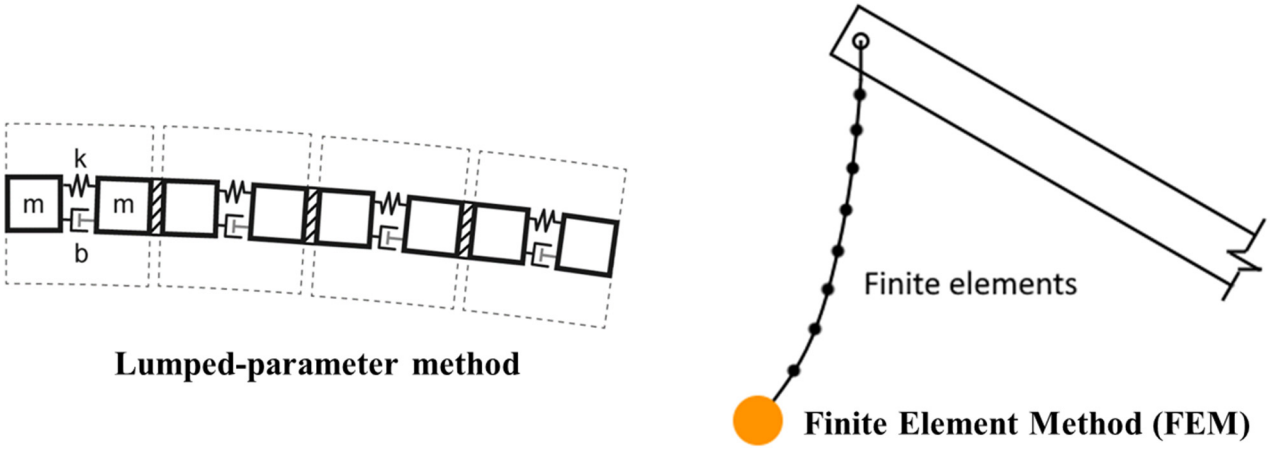


Fig. 3.14 Cable modeling using Lumped-parameter method [94] and Finite element method

The approach used involves modification of the connection between the cable and the crane boom. In the modified approach, the interaction between the boom and the suspended load is realized through a telescopic joint with 4 degrees of freedom. This ensures that the load frame (follower) rotates with respect to the base in all directions and the cable length becomes a controlled variable – a translational motion. Lifting and lowering of the load is performed along the local *z*-axis of the joint, which is always directed towards the suspended load. In addition, custom block has been developed to introduce the constraint equations linking the load to the joint, thereby modeling the cable. The physics handled by the block was introduced using the Simscape programming language. This language is a built-in language of MATLAB/Simulink and allows to implement user-defined physics and create custom custom blocks. A generalized scheme of the interaction of the numerical model elements in the new approach in contrast to Fig. 3.7 (left) is shown in Fig. 3.15.

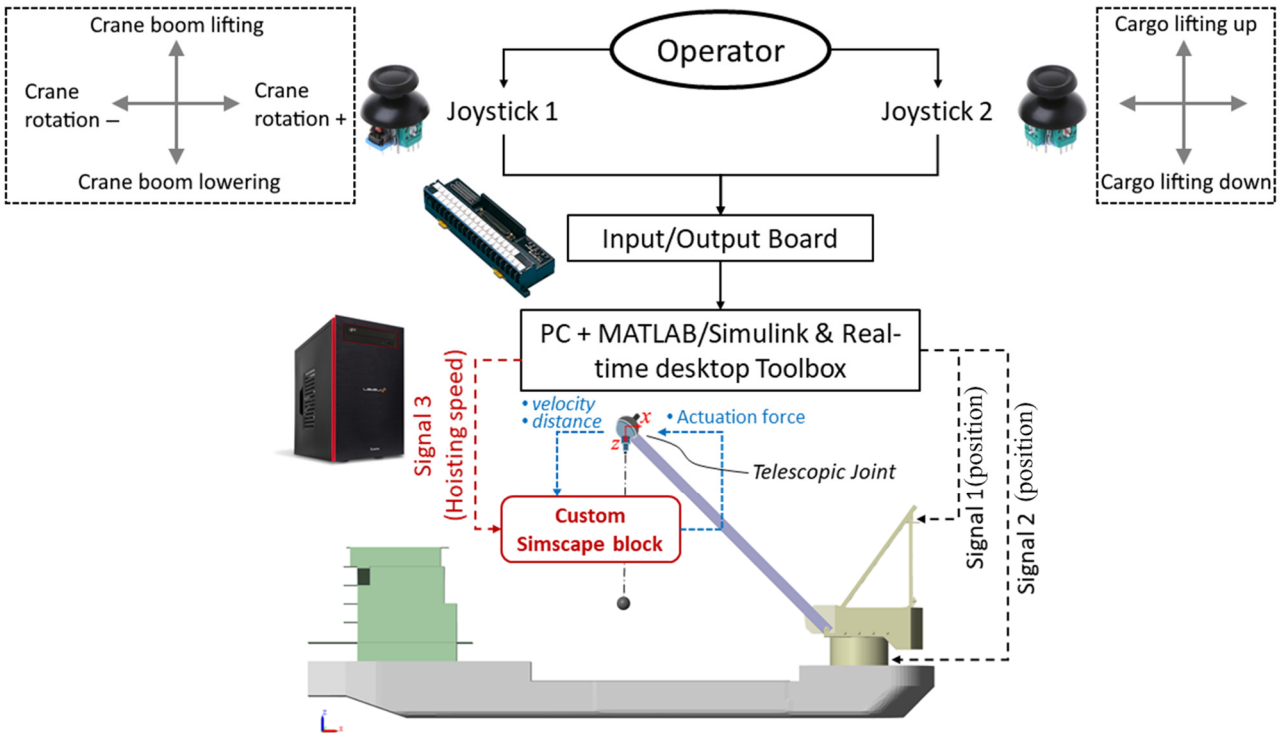


Fig. 3.15 Generalized scheme of interactive modeling with the developed custom block [93]

The block works as follows. During model initialization, the equilibrium state is set by specifying the current cable length, weight, cable stiffness and damping, and initial cable elongation. Then, within the simulation cycle, the input parameters to the block are:

- distance between base and follower frame of telescopic joint D;
- instantaneous cargo velocity provided by joint sensor v;
- hoisting speed provided by input from manipulators (control signal) v_{ctrl}.

The internal algorithm of the custom block, which is computed every timestep (Fig. 3.16):

1. At first, three parameters are calculated – cable length increment $\Delta L_{(t)}$, total cable length at the current time step $L_{(t)}$ and instantaneous free cable length $x_{(t)}$:

$$\Delta L_{(t)} = \int v_{ctrl}\, dt = v_{ctrl_{(t)}} \Delta t, \tag{3.4}$$

$$L_{(t)} = L_{(t-\Delta t)} + \Delta L_{(t)}, \tag{3.5}$$

$$x_{(t)} = L_{(t)} - D_{(t)}, \tag{3.6}$$

where $L_{(t-\Delta t)}$ is the cable length at previous time step; Δt is the timestep size; $\Delta L_{(t)}$ is the cable length increment at the current time step; $D_{(t)}$ is the instantaneous distance between the base frame and the follower frame of the telescopic joint.

2. The cable state is assessed. If the free cable length is greater than or equal to 0, then the stiffness and damping of the cable becomes 0 and the cable is loosened:

$$\begin{cases} k = 1000\dfrac{N}{m};\ c = 10\dfrac{N \cdot s}{m}, & x < 0 - \text{cable is taut} \\ k = 0; c = 0, & x \geq 0 - \text{cable is loosened} \end{cases} \tag{3.7}$$

where k is the cable stiffness and c is the cable damping.

3. The tension force F in the cable is determined using the elastic-damped model:

$$F_{(t)} = kx_{(t)} + cv_{(t)} \tag{3.8}$$

The output parameter of the block is the force, which is then imported as the actuation force in the telescopic joint along the *z*-axis (which is always directed toward the suspended load). In this way, the interaction of the joint and the custom block is modeled. The joint transmits the current distance and velocity of the cargo to determine the forces by the custom block that constraints the cargo movement. In this way the prescribed distance constraint is not common one, since distance is now only used to estimate the tether condition and to assign stiffness and damping coefficients "on-the-fly".

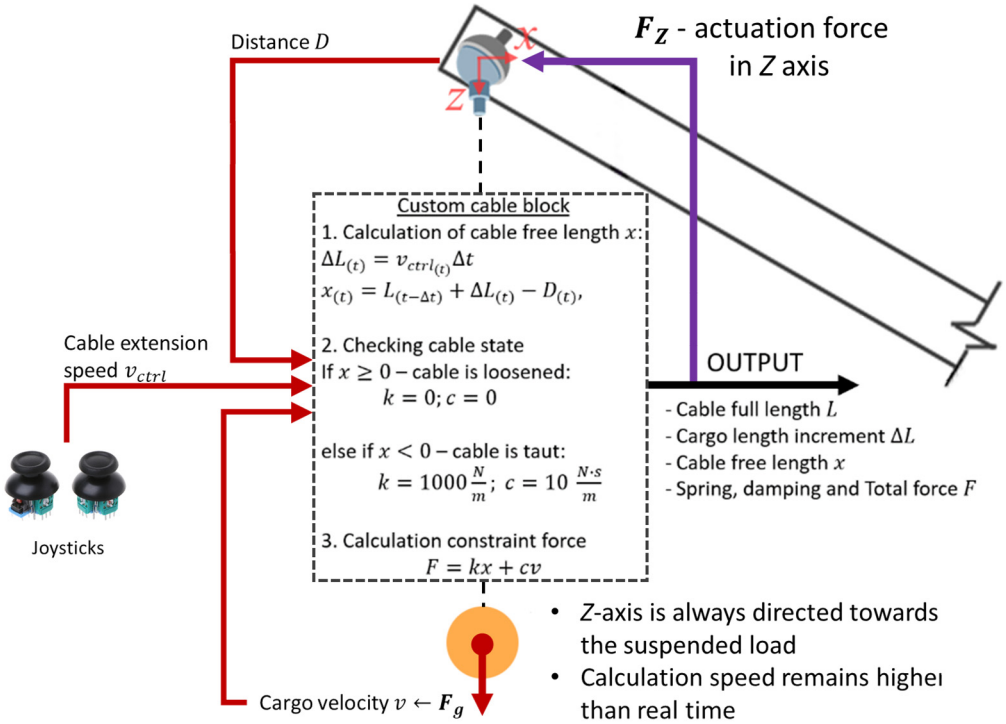


Fig. 3.16 The concept of a custom block developed in the Simscape language [93]

The code of developed custom block is presented below and also in Appendix B. The "`inputs`" section is used to define input signals – velocity and distance from the spherical joint and input signal from joysticks (control of hoisting speed):

```
inputs
    v_inp = {0, 'm/s'};            % Input velocity:right
    position_inp = { 0, 'm' };     % Input position:right
    v_ctrl = {0, 'm/s'};           % Controled velocity:right
end
```

Similarly, the "`outputs`" section defines the signal output ports of the custom block (labeled in Fig. 3.16):

```
outputs
    force_tot_out = { 0, 'N' }; % Total force:left
    force_spr_out = { 0, 'N' }; % Spring force:left
    force_dmp_out = { 0, 'N' }; % Damping force:left

    x_out = { 0, 'm' }; % Cable free length:left
    delta_l_out = { 0, 'm' }; % Cable increment:left
    l_out = { 0, 'm' }; % Cable full length:left
end
```

In the "`parameters`" section, physical and mechanical parameters of the cable that can be controlled by the user (stiffness, damping, initial length, load mass and gravity) are configured:

```
parameters
    k = { 10000, 'N/m' };      % Cable axial stiffness
    D = { 10, 'N*s/m' };       % Damping coefficient
```

```
    length = { 0.5, 'm' };    % Initial cable length
    m = { 0.16, 'kg' };       % Suspended mass [kg]
    g = { 9.8067, 'm/(s^2)'}  % Gravity acceleration
end
```

Next, the "variables" section defines internal parameters that will be used by Simscape in the calculation process and are not available for manual editing by the user:

```
variables (Access = private)
    x = {value = { 0, 'm' },priority = priority.high}; % Cable elongation
    kk = {value = k,priority = priority.high};        % Variable stiffness
    DD = {value = D,priority = priority.high};        % Variable damping
    delta_l = { 0, 'm' }; % Length increment
    tensioned = 0;        % Tension flag
end
```

The "intermediates" section includes the governing equations for calculating the instantaneous cable length, the elastic and damping forces, and the total tension force:

```
intermediates
    force_spr = -kk*x;                  % Spring force equation
    force_dmp = -DD*v_inp;              % Damping force equation
    force_tot = force_spr + force_dmp;  % Total force equation
    l = length + delta_l;               % Instantaneous cable length
end
```

The "function" section is optional in this case, but serves as an additional check on the data entered by the user to ensure that negative stiffness, damping and length values are not used:

```
function setup
    if k <= 0
        error('Cable stiffness must be greater than 0');
    end
    if D < 0
        error('Cable damping cannot be negative');
    end
    if length < 0
        error('Initial cable length must be greater than 0');
    end
end
```

Finally, the " equations" sections contain equations to calculate the free length of the cable, estimate its condition (tensioned or loosened) and, depending on the situation, modify the stiffness and damping values:

```
equations (Initial = true)
    force_spr_out == m * g;                  % Initial force due to gravity
    x == (position_inp + l + force_spr_out/k) % Initial cable
end

equations (Initial = false) % regular equations
```

```
    x == (position_inp + l) % Cable free length

    if x > 0
        kk == 0;
        DD == 0;
        tensioned == 0;
        delta_l.der == v_ctrl
    else
        kk == k;
        DD == D;
        tensioned == 1;
        delta_l.der == v_ctrl
    end
    force_tot_out == force_tot;   % Total force
    force_spr_out == force_spr;   % Spring force
    force_dmp_out == force_dmp;   % Damping force
    x_out == x;                   % Free length output
    delta_l_out == delta_l;
    l_out == l;
end
```

The block diagram of numerical model is presented in Fig. 3.17. The custom block is located at blue area. It receives the input of position and velocity from telescopic joint and control velocity from the user via the same set of blocks (analog input, sum, gain, integrator and filtering) as described in previous subsection. It has 6 output signals but only one (total force) is threated at actuating force for telescoping joint. In addition, the contact modeling is presented in green area. A rigid massless body representing the crane ship deck was added to the model. It is rigidly connected to the crane ship hull and serves as simple geometry, which provides the convex mesh for contact search. Of course, it is possible to use the mesh of ship hull itself but in that case the computation will slow down significantly since STL file of the hull contains a large number of panels.

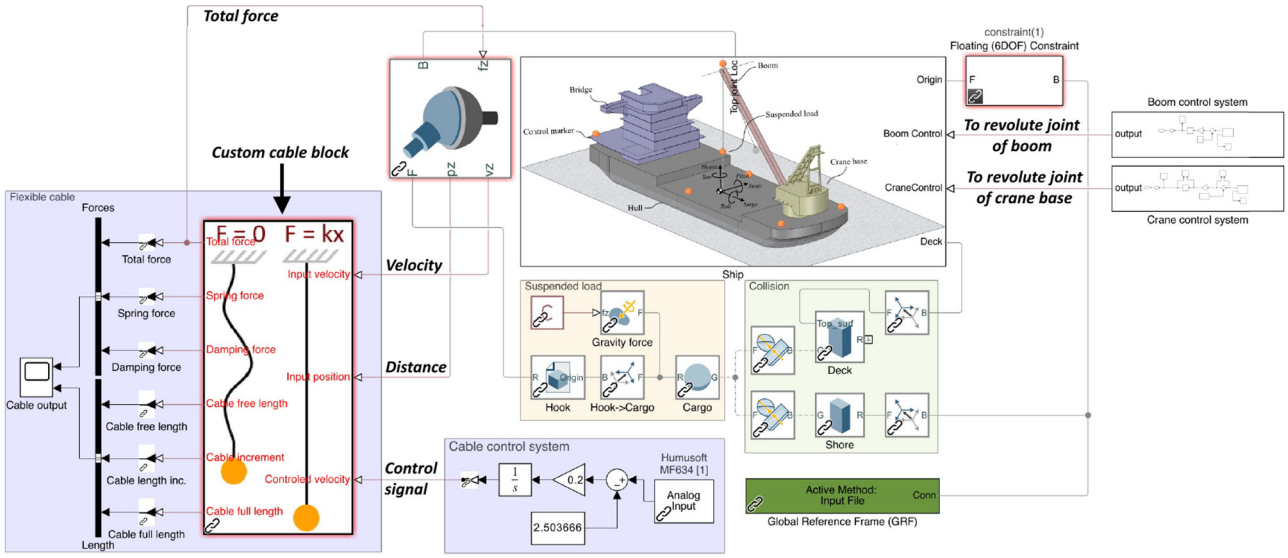


Fig. 3.17 The block diagram of numerical model in Simulink environment with custom cable block

The numerical simulation was performed with the same conditions as in Section 2. The height of the regular wave in the model is 0.02 m, the wave period is 1.4 s, and the wave heading is 0 degrees

(stern-to-bow direction). In the process of simulation, the cargo was lowered on the ship's deck and then lifted up several times. The time step of the calculation was not more than 0.01 s.

Fig. 3.18 shows how the length of the cable changed over time. The first moment of contact between the cargo and the deck of the ship can be observed at 1.8 s of modeled event, where the cargo itself (black line) stops its gradual lowering. At further increase of the cable length (red line) its free length becomes apparent and cable switches to the loosened state. Then at 3.3 seconds, the cable length starts to decrease and at about 5 seconds the cargo lifts off. The cargo movement and cable extension are smooth and show a good quality picture.

The force metrics of the "vessel-cable-cargo" system are shown in Fig. 3.19. At the beginning of the simulation the cargo is in a suspended state and the cable tension force is equal to 1.57 N, which correctly corresponds to the cargo weight of 0.16 kg. At the moment of contact, contact force of 2.5 N is sharply arising due to the cargo hitting the deck. At the same time, the tension force of the cable becomes 0 because it switches to a loosened state. Further contact force is equal to 1.57 N, which indicates that the balance of forces in the model is calculated correctly.

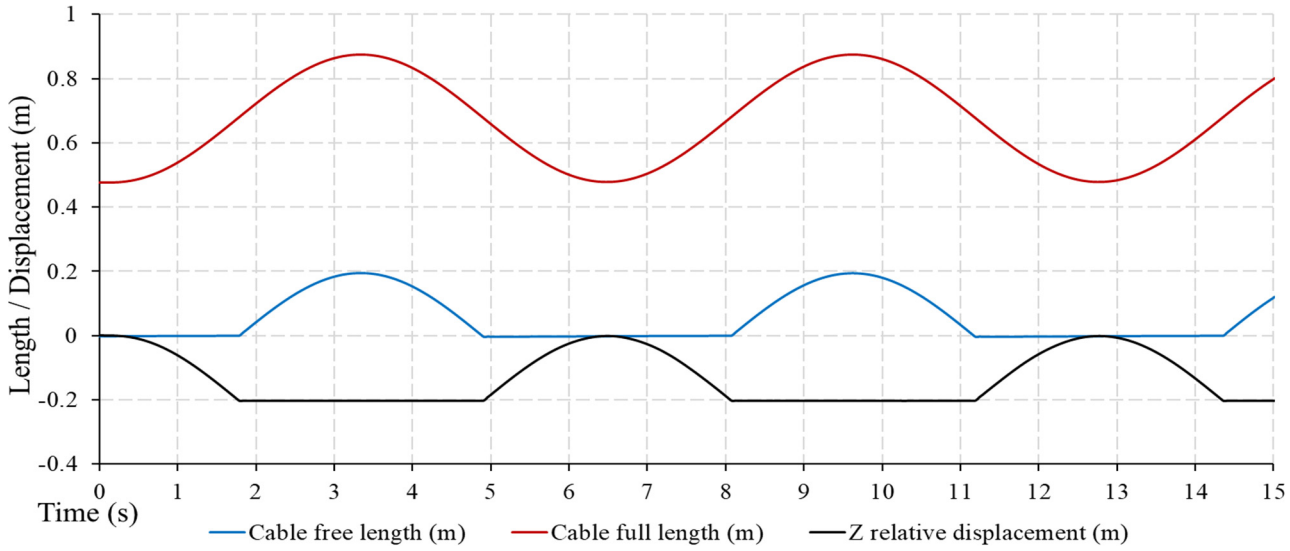


Fig. 3.18 Cable and cargo output

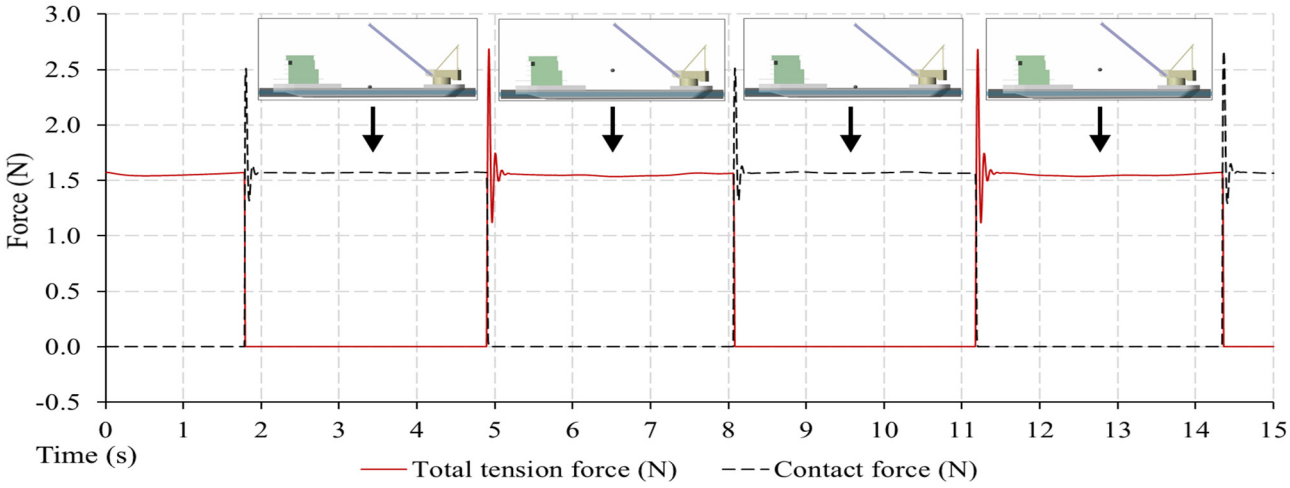


Fig. 3.19 Tension and contact forces

It is worth to consider particularly the moment of cargo detachment from the deck. After the transition of the cable to the taut state in the next time steps the tension force increases sharply over the value of 2.5 N, which does not quite correspond to the correct picture. We associate this with the mechanics of the mutual behavior of the cable length control mechanism and the contact model. The point is that the spring-like contact behavior does not allow the instantaneous transition to a zero force value and still keeps the load "inside" the penetration depth for several time steps as presented in Fig. 3.20. At the same time, the cable is controlled by velocity control, not force control. As a consequence, for several time iterations, an excessive distance of the load from the joint, over-tensioning of the cable and an increase of the tension force occur. However, after 0.2 s the tension force returns to its correct value.

This effect is repeated every time the load is detached from the deck. There are several ways to overcome this problem. The simplest is to introduce a PID controller to correct the control signal from the joysticks. If there is too much cable tension, the hoisting speed will be adjusted to compensate for this effect. Another way is to adjust the algorithms of the custom block itself. For example, it is possible to introduce a state-based cable control where the force control is performed when the cable is taut, and the velocity control is performed only when the cable is loosened. Also, it is worth to mention that additional simulation cases showed normal contact behavior when the load is sliding on the deck.

The motion sequence of the suspended load during simulation is illustrated in Fig. 3.21. The computational efficiency of the cable modeling approach with constraint equations implemented as custom block can be observed in Fig. 3.22. It is clear from the figure that the application of the custom block did not slow down the model as a whole. When the cargo was suspended or placed on the deck, model convergence was achieved by applying a time step of 0.01 s. Decreasing the time step down to 0.002-0.004 s occurred for a short time only in the moments of cargo detachment and lowering. Thus, the used approach is effective and allows to keep calculations at a speed higher than real-time.

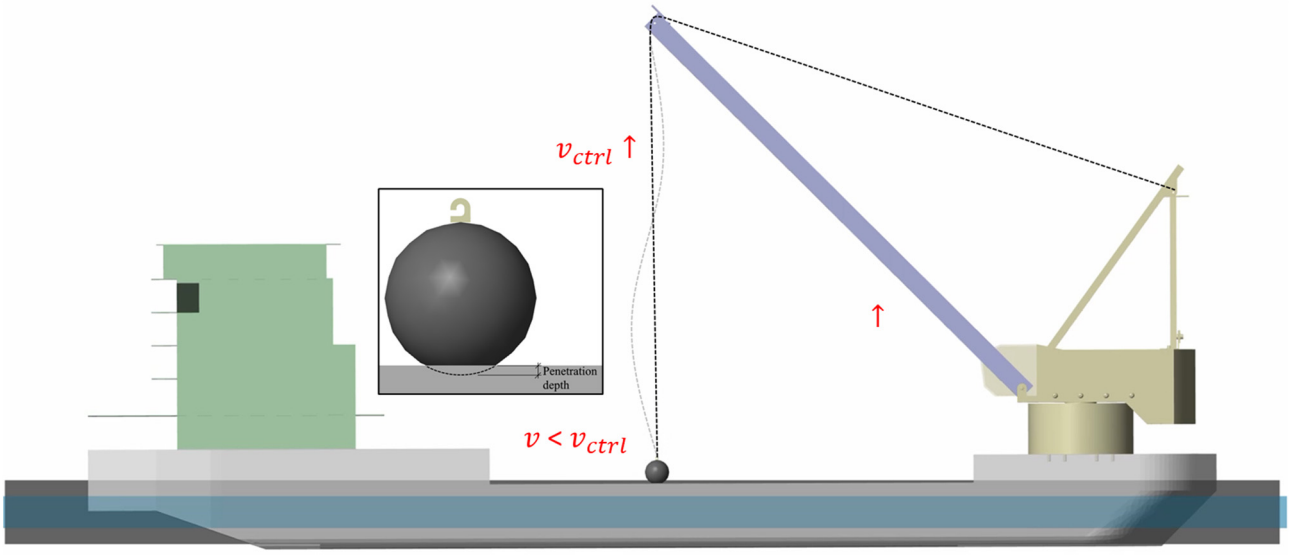


Fig. 3.20 Overtightening of the cable at lift-off moment

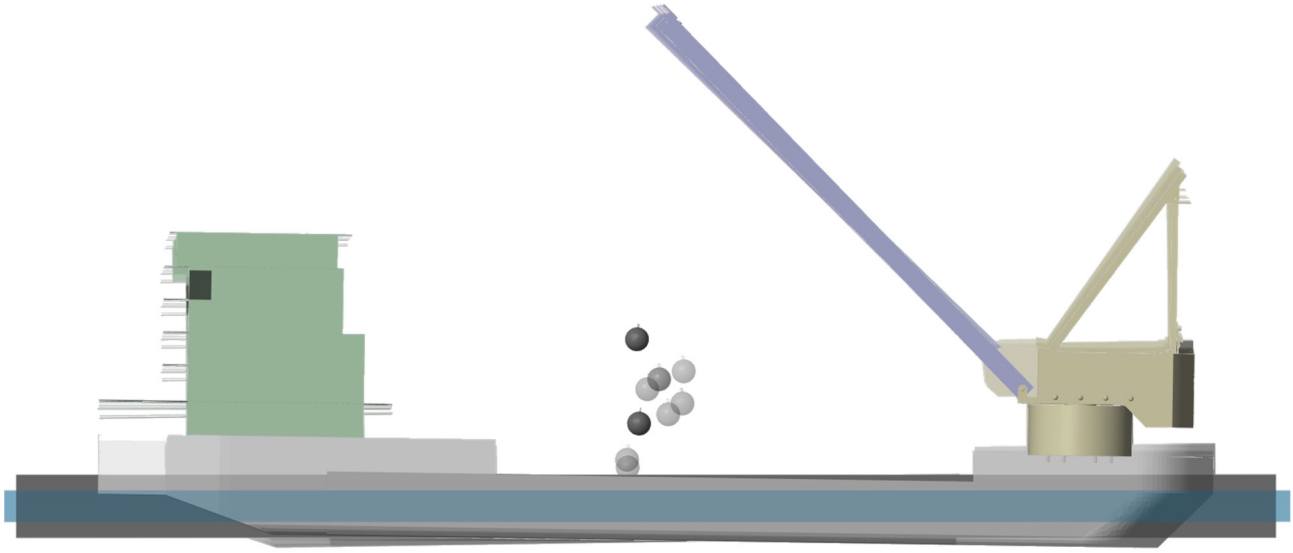


Fig. 3.21 Motion sequence of suspended load and crane ship during simulation with custom block

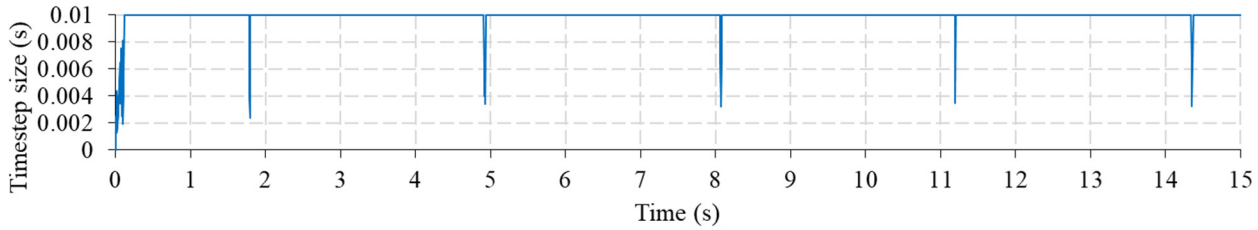


Fig. 3.22 Timestep size during simulation

Since the flexible cable was introduced, it becomes possible to check the effect of the flexible cable on the dynamics of the whole crane ship system. The simulation was performed with cable stiffness of 1000 N/m and a damping factor of 10 N/(m/s), which is 0.4 of critical damping, and *showed no difference* from the modeling case with rigid cable described in Section 2.3.3. Indeed, the experimental and simulation conditions suggest the effect of resonance – the wave frequency coincides with the pendulum's natural frequency (0.71 Hz). Under such conditions, the cable will always be taut. The contribution of the flexible cable will be noticeable if the mass of the suspended load is significantly increased, since with a rigid cable there will be a transfer of inertial forces to the ship itself.

3.3. A case example of modeling the tetrapod carrying operation

This subsection presents an example of using the developed approach to model the process of cargo transfer from a crane ship to the “shore” and back to the ship deck. The simulation was performed in 5 steps (Fig. 3.23) accompanied by interactive control of the crane ship elements using joysticks:

1. Turning the crane base towards the shore.
2. Lowering the crane boom until the load comes into contact with the surface.
3. Raising the crane boom and holding the load in the air.

4. Turning the crane base back towards the ship's deck.
5. Lowering the load onto the deck until it comes into contact with the surface.

For this task, the contact algorithm described earlier was set up for both the deck of the crane ship and the shore. The simulation was performed in real time without computational delay at the moment of cargo contact with the surface. The force in the cable and the contact force during the event is shown in Fig. 3.24. The values of the forces at the contact and in the cable correspond to each other. There is also a slight noise, especially in the cable force at the time interval from 11.5 to 18 s due to the significant swaying of the suspended load during steps 3-5. As in the previous subsection, the disadvantage of the simple cable model and velocity control is observed at the moment the payload is placed on the surface and at the moment it is lifted up. It is evident in the form of a significant force spikes that quickly disappear after a few time steps. Motion sequence of suspended load and crane ship during simulation steps is presented in Fig. 3.25 and Fig. 3.26.

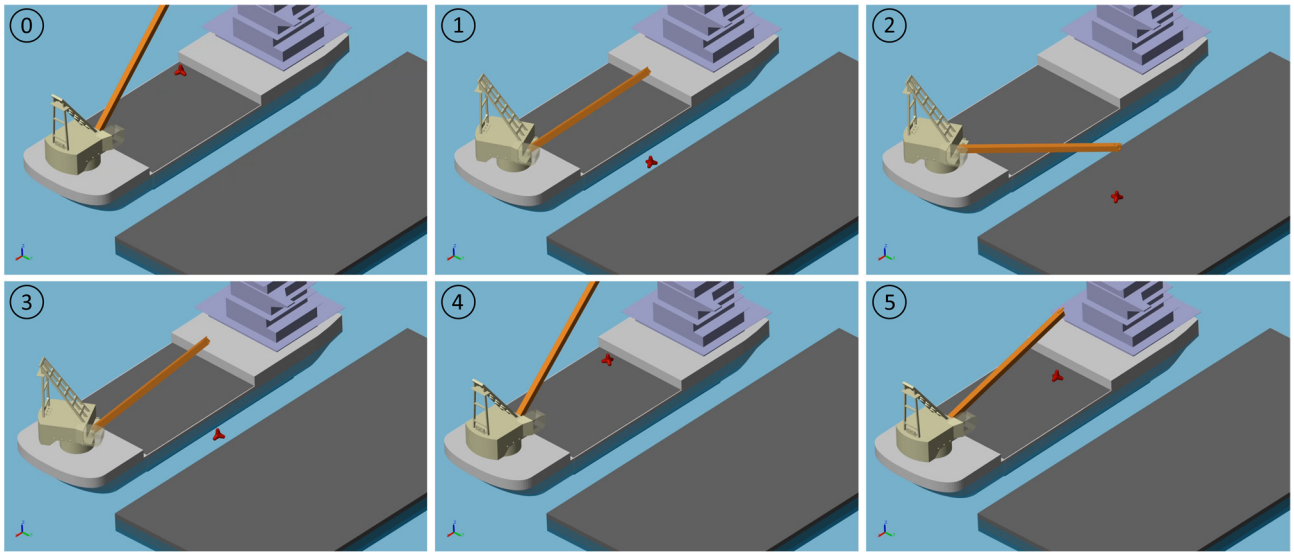


Fig. 3.23 Cargo carrying steps during interactive modeling example case

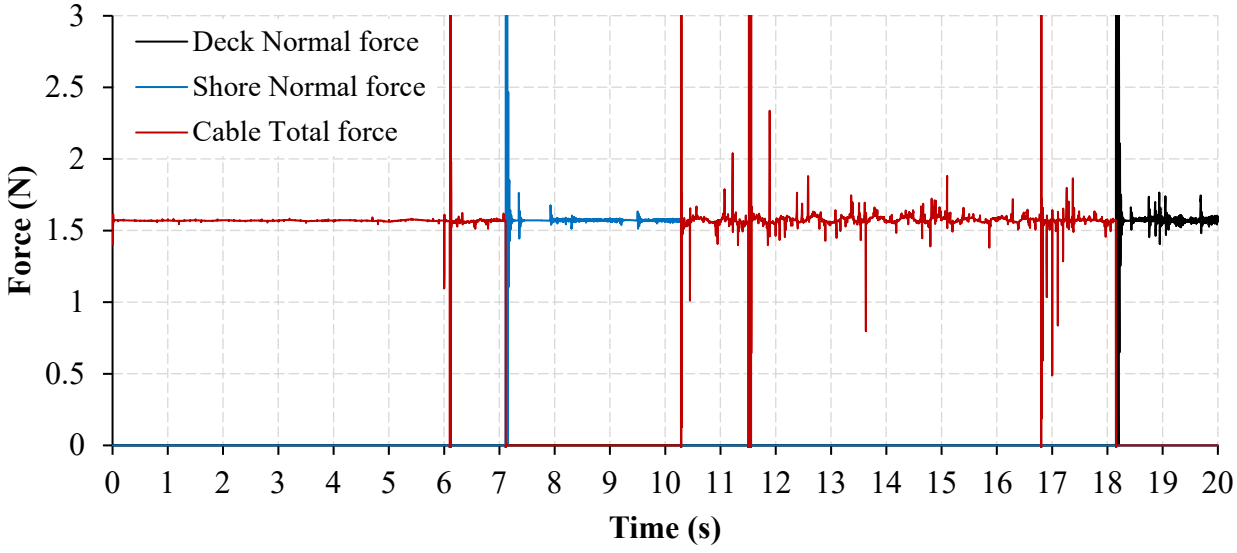


Fig. 3.24 The force in cable and normal contact forces during simulated example case

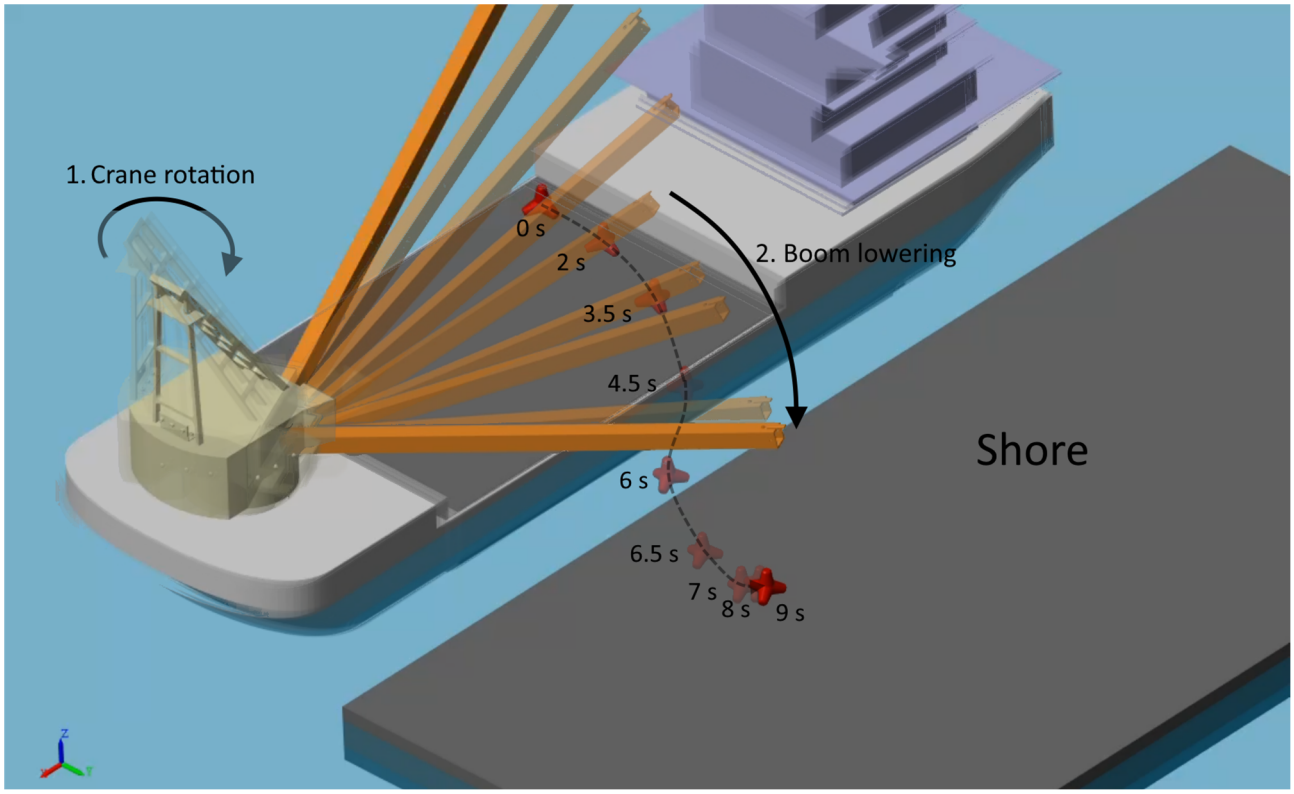


Fig. 3.25 Motion sequence of suspended load and crane ship during simulation (steps 0-2)

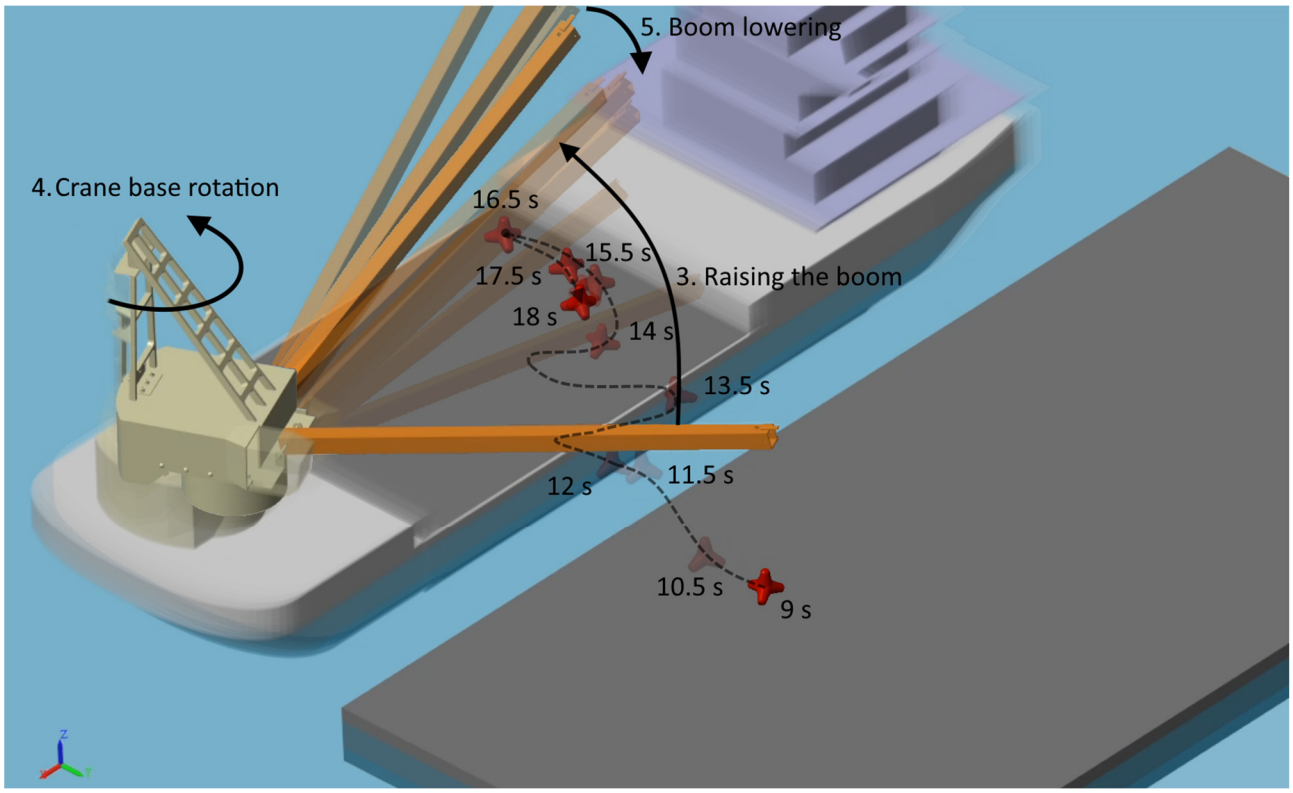


Fig. 3.26 Motion sequence of suspended load and crane ship during simulation (steps 3-5)

3.4. Section conclusions

In this section, the method and model configuration for interactive modeling of crane ship offshore operations was developed. The interactive control is implemented using MATLAB/Simulink

environment with Desktop real-time Toolbox and hardware connected to PC via installed I/O card and 2-DOF manipulators (joysticks). The first attempts assumed the use of rigid cable for lifting the suspended load in the air and showed promising results in case of handling the crane ship hydrodynamics and real-time computing. The father modification implies the use of a custom-designed block that senses the control signal from the hardware, calculates the constraint forces and uses them as actuation force of the joint coupling the crane boom and the suspended load. To simulate the “cargo-ship” contact, a penalty-based contact was used.

The results show that this approach is quite effective for modeling cargo motions. An increase in performance is required only for the processing of cargo-surface contact moments. This is an important advantage especially when the approach is used to simulate hoisting systems with multiple cables.

Nevertheless, the developed method has several drawbacks in the current implementation. Thus, the excessive tension of the cable at the moment of cargo detachment from the surface due to the delay of spring-like motion by the contact model was noticed. It is also obvious that the inertia of the cable in the given method is not taken into account. However, all this can be resolved by additional improvements in the algorithm of the custom block, which is one more task for future works.

4. AI-driven anti-sway control and positioning of suspended load

4.1. Preliminary considerations about the problem at hand

Since offshore operations using crane ships are characterized by complex nonlinear dynamics and increased amplitudes of oscillations of suspended cargoes, such a process is subject to increased safety requirements. Therefore, this section raises the question “How to control the swing (sway) of suspended load?”. The necessity to ensure the safety of cargo handling operations led to the active development of anti-swing (sway) systems for floating cranes. One example of such a system is the dynamic positioning system, which corrects the vessel's motion in space by automatically controlling the vessel’s thrusters.

However, such systems have low performance in controlling the sway of the cargo itself, as controlling all 6 degrees of freedom of the hull is difficult to implement. Therefore, research engineers are developing control systems that minimize the displacements of individual mechanisms and parts of the floating crane relative to the global reference frame. The control solutions related to conventional control theory (such as PID controllers in Fig. 4.1) considered in Section 1 showed good results but have significant disadvantages – the difficulty of adjustment of proportional factors and controller parameters for different states of the system and poor performance in the presence of nonlinearities. Therefore, the problem of minimizing load oscillations requires an improved approach that would circumvent these limitations.

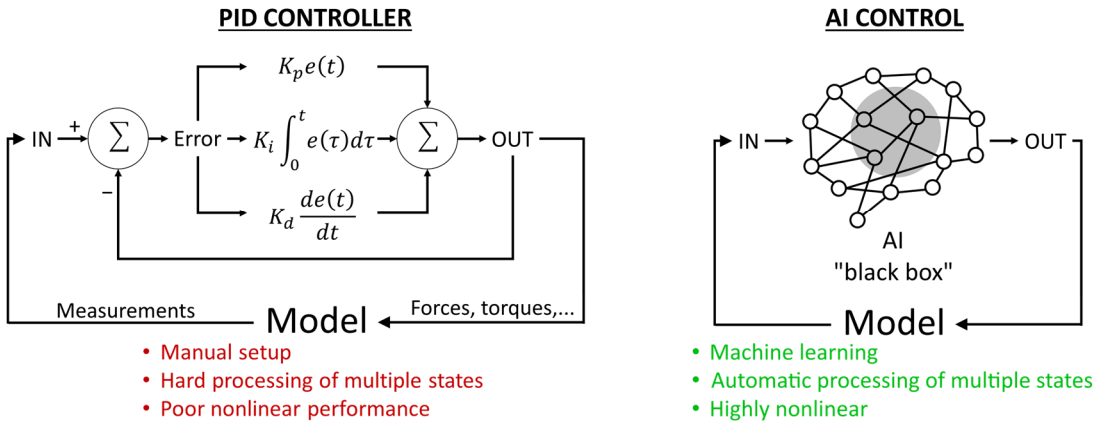


Fig. 4.1 Proportional-integral-derivative controller vs AI control

Interactive modeling method introduced in this thesis earlier opens up the possibility of applying a new modern method of controlling physical systems – AI-driven control. AI here means “artificial intelligence” and implies that the entire control process will be carried out by a complex intelligent system specially trained for this task. This method is on the edge of modern engineering practice and is investigated in this thesis in relation to the problem of minimizing the sway of the suspended load. Moreover, with the use of AI it is possible not only to minimize the swing of suspended load but also implement the automatic positioning of the cargo to the desired location. This adds even more advantages over conventional approaches.

The application of AI and neural networks for sway suspension of suspended load on crane ships is quite a novel problem. However, the foregoing studies demonstrated promising results. Ramli et al. [95] suggested the artificial neural network (ANN) input shaping method for minimizing the payload swing of an overhead crane with payload hoisting and payload mass variations. Qian et al. [96] proposed an adaptive neural network (NN)-based anti-swing tracking control strategy to predict motions of suspended cargo during ship-to-ship operations. The authors performed several experiments with developed hardware model of ship-mounted crane and showed that this strategy can handle complicated unknown wave-induced disturbances. Kim et al. [97] designed a neural network-based estimator that applied the sliding-mode anti-sway control law to solve the problem of transporting an underwater object by industrial cranes.

From the very essence of the machine learning process, there are two basic requirements for a numerical model: high computation speed and acceptable modeling accuracy. The first requirement (computation speed) is necessary to effectively model hundreds and thousands of training episodes, and the second is to ensure reliability when transferring AI trained on the model to a real crane ship. The method of numerical modeling of crane ship dynamics developed in this thesis fulfills these requirements.

Current research is focused on implementation of AI-driven positioning and anti-swing control of suspended load configured by reinforcement learning (RL). This training approach consists in iterative computation of a nonlinear ship-cargo system with the same or different initial conditions at each iteration (episode). During training the artificial intelligence automatically learns gradually how to compensate for cargo swing at different states of the system. The result is a trained control system, which is characterized by nonlinear behavior and can show high efficiency at different states of the numerical model. After training, such a system will not require manual adjustment, as all the necessary dependencies will already be established and calibrated during the training process itself. It can be called an AI “black box”, which has its own logic of actions that are taken depending on the current dynamics of the “vessel-cargo” system.

The implementation of reinforcement learning-based artificial intelligence is also conducted using MATLAB tools, specifically the Reinforcement Learning Toolbox. This approach allows us to maintain the principle of constructing the entire numerical model within a single software, minimizing losses in computation speed. To achieve this, the RL Agent block is used (Fig. 4.2).

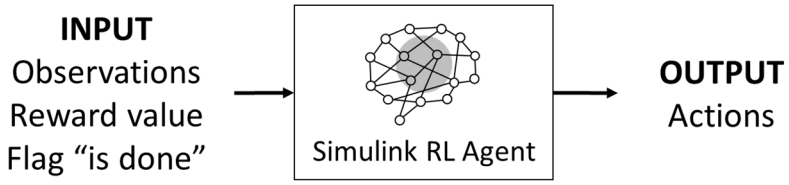


Fig. 4.2 Simulink RL Agent block

The input data for the RL Agent consists of an observation vector, reward value, and a “is done” flag, which terminates the training of the current episode if a pre-defined criterion is met. The output data is a vector of actions (signals) that actuate the necessary mechanisms in the model. In other words, the RL Agent reads observations from the environment (the states of the model) at each computation step and takes actions based on these observations using the learnable component "policy", as depicted in Fig. 4.3.

The policy is a mapping from the current environment observation to a probability distribution of the actions to be taken [98]. Thus, we can say that the policy is a kind of "logic" for artificial intelligence, based on which it performs actions, and through training (using a learning algorithm), we adjust this logic.

Next, the results of applying reinforcement learning method to train an AI acting as a control system are summarized.

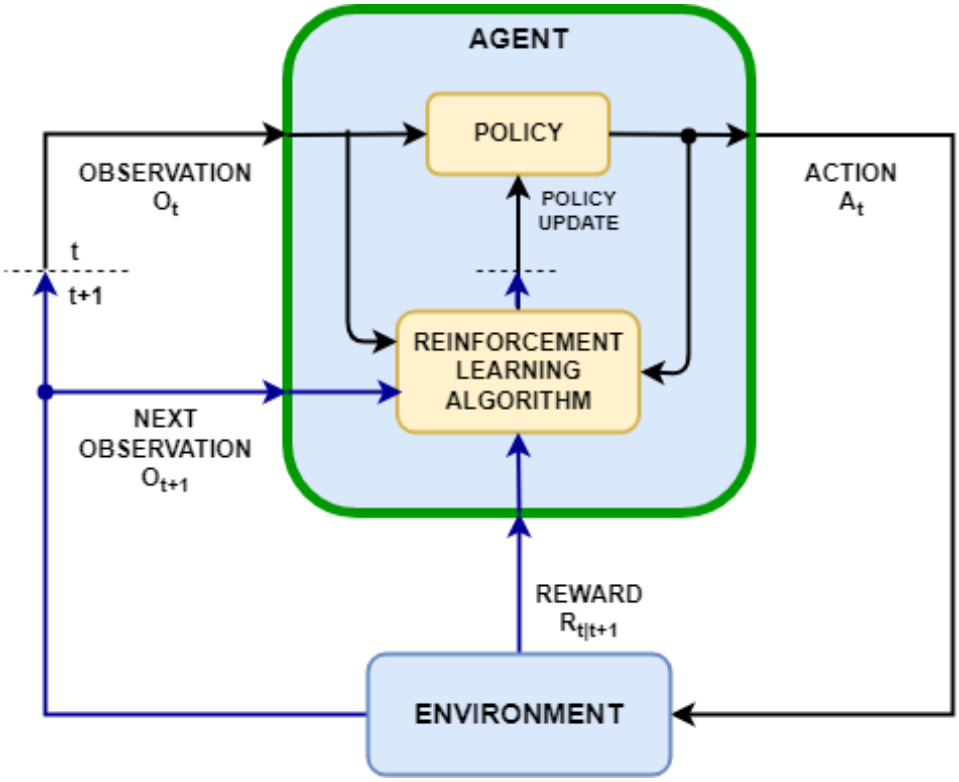


Fig. 4.3 Coupling of RL Agent and Environment (model) [98]

4.2. Testing AI-control on the example of a pendulum with a moving base

Before implementing the AI control system into the numerical model of the crane ship, a simplified task was addressed. In the simplest case, the crane ship and the suspended load can be represented as a two-mass system, as shown in Fig. 4.4. In other words, the suspended load behaves as a pendulum on a moving base. Therefore, the first step in developing an AI-driven system for minimizing sway and positioning was to practice the approach on a simplified model.

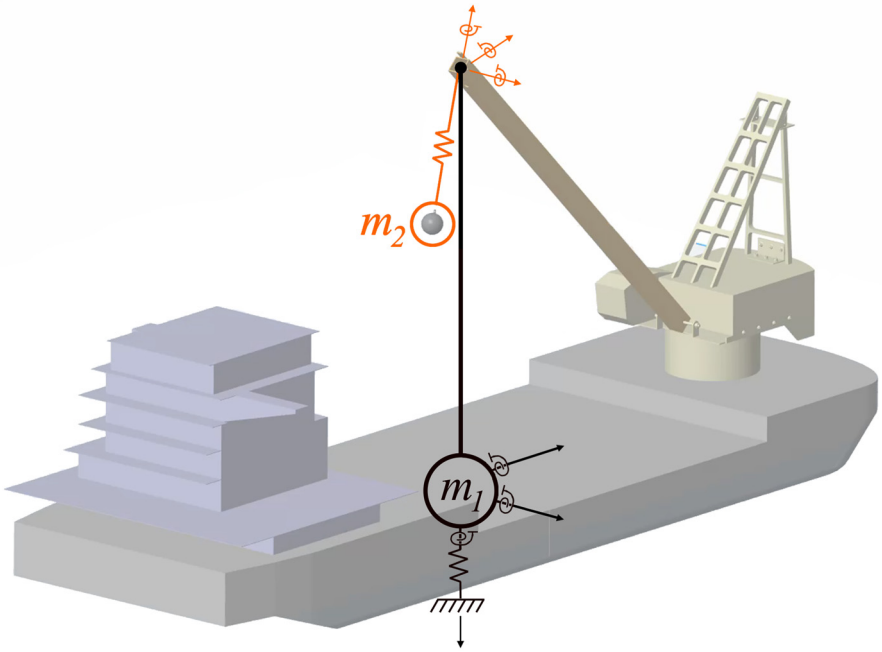


Fig. 4.4 Crane ship and suspended load as two-mass system

A numerical model of the pendulum, shown in Fig. 4.5 , was created to test the training methodology and adjust the model parameters. The model included two bases and a suspended load. Base 1 is connected to the global frame using a rectangular joint and has 2 degrees of freedom in the x and y-axis direction. Base 2 with a mass of 1 kg was connected to base 1 with dimensions of 0.35x0.35 m via a prismatic joint, thereby allowing base 2 to move in the global x-axis direction. A suspended load was attached to base 2 using a custom cable block developed earlier.

The fundamental concept here was to train the RL Agent to apply a force to Base 2 to minimize the distance (x_L) and angle (θ) from the suspended load to world frame vertical line, while the sway of the payload is induced by the motion of Base 1. As Base 1 moved randomly along the x and y axes through the actuation (position input) of the prismatic joint connecting it to the global frame, a disturbing force initiated the motion of Base 2. To transmit force to Base 2, a damping coefficient of 20 N/(m/s) was assigned to the prismatic joint.

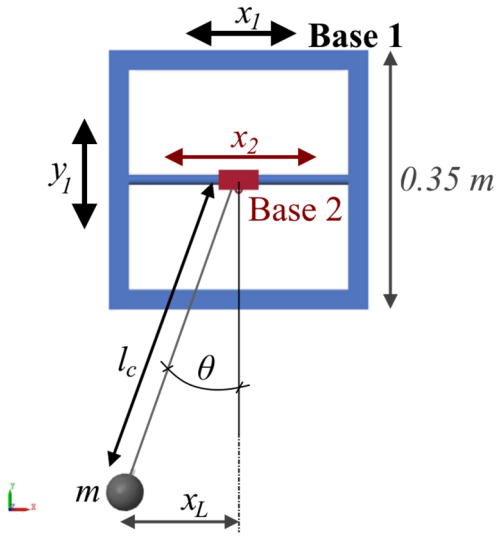


Fig. 4.5 Model of the pendulum with the moving base

To implement the AI control system in the model, the following steps need to be taken:

- select observations (model parameters) to be tracked during the computation of the model.
- determine the actions RL Agent will take to accomplish the defined task.
- develop a reward model, based on which the training will occur, and perform reward shaping.
- define a criterion at which further modeling of an episode is not feasible ("is done" flag) to terminate unsuccessful episodes earlier.
- choose a learning algorithm and design the structure of deep neural networks as approximators to train the policy of the RL Agent.

Observations are instantaneous model parameters that can be measured during the modeling process (such as displacements, velocities, forces, etc.). The selection of observed parameters (system states) during the learning process should be based on an assessment of whether a particular parameter will influence the RL Agent's policy. Thus, 10 observed parameters were selected, as presented in Table 4.1.

Table 4.1 Observations used in pendulum test

Number	Observations	Symbol
1	Cable length	l_c
2	Cable force	F_c
3	Displacements of base 2 relative to base 1 frame	x_2
4	Velocity of base 2 relative to base 1 frame	$\dot{x}_2$
5	Cosine of the load inclination angle	$\cos\theta$
6	Sine of the load inclination angle	$\sin\theta$
7	Angular velocity of the suspended load	$\dot{\theta}$
8	Velocity of base 1 along the *x*-axis	$\dot{x}_1$
9	Velocity of base 2 along the *y*-axis	$\dot{y}_1$
10	Horizontal distance from the load to the vertical	x_L

It is evident that training should be conducted with various initial states of the system so that artificial intelligence can effectively perform the task under different conditions. Accordingly, the following random parameters were used in each training episode:

- cable length l_c – from 0.1 to 0.5 m;
- the mass of the load m – from 0.1 to 2 kg;
- the initial inclination angle θ – from -20 to 20 deg.

As mentioned earlier, a single signal transmitted to the prismatic joint of Base 2, serving as the actuating force, is adopted as the action of the RL Agent. The training did not impose any limitation on the maximum magnitude of the force.

Policy training is performed by two parameterized function approximators – actors and critics – that are part of the machine learning algorithm. The policy parameters are approximated based on the reward value (a measure of success), which is computed according to the user-defined function at

each sampling step. The goal during policy training is to maximize the reward. Based on whether the RL Agent's actions have increased or decreased the reward, the policy is updated. Thus, gradually with each new episode, the RL Agent learns and understands what actions to take depending on the instantaneous state of the model to maximize the reward. The following reward model for positioning the cargo close to vertical was designed by a trial-and-error process:

$$r = \begin{cases} 30, & if\ \dfrac{k_1}{x_L + 10^{-9}} + k_2 > 30 \\ \dfrac{k_1}{x_L + 10^{-9}} + k_2, & if\ |x_L| \le 0.02\ m \\ k_3 x_L^2 + 0.33 k_2, & if\ 0.02\ m < |x_L| \le 0.15\ m \\ C + k_3 x_L^2 + 0.33 k_2, & if\ |x_L| > 0.15\ m \end{cases} \quad (4.1)$$

where x_L is horizontal distance from the load to the vertical; k_1; k_2 and k_3 are constant proportional factors; C is penalty, represents a penalty applied if the distance x_L exceeds 0.15 m.

This model assumes that the reward will be higher when the suspended load is closer to the central vertical line relative to the global coordinate system. Of course, it would be good practice to include in the model a reward for the deviation angle as well, not only to minimize the distance but also the oscillations themselves. However, in this simple case, it is not necessary, as the presence of swaying leads to periodic deviations of the load from the vertical, reducing the reward. In other words, minimizing swaying is already somewhat implied in this model.

The coefficient values were tuned through several iterations, and ultimately, the following values showed good results: $k_1 = 0.05$, $k_2 = 25$, $k_3 = -5000$ and $C = -250$. Thus, the reward is a function only of the distance $r = f(x_L)$ and can be represented graphically (Fig. 4.6) and below as a continuous function:

$$r = \begin{cases} 30, & if\ \dfrac{0.05}{x_L + 10^{-9}} + 25 > 30 \\ \dfrac{0.05}{x_L + 10^{-9}} + 25, & if\ |x_L| \le 0.02\ m \\ -5000 x_L^2 + 8.25, & if\ 0.02\ m < |x_L| \le 0.15\ m \\ -250 - 5000 x_L^2 + 8.25, & if\ |x_L| > 0.15\ m \end{cases} \quad (4.2)$$

As the termination criterion for an episode, the condition $x_L \ge 0.15$ m was adopted, indicating a situation where Base 2 deviated too much from the central vertical line in the model. Thus, in the step of the calculation within an episode in which this condition was met, a penalty equal to -250 was applied, then the calculation was interrupted and a new episode was started.

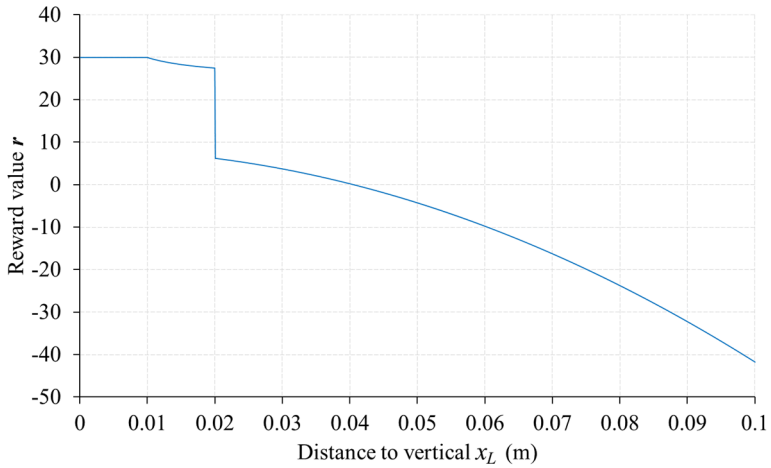


Fig. 4.6 Reward function used in pendulum test

There are numerous different reinforcement learning algorithms available in MATLAB such as DQN, DDPG, SAC, TD3, PPO, TRPO, etc. Some information about the mathematical background of these algorithms can be found in [98]. Firstly, the DDPG agent was tested for task at hand. However, it was not possible to achieve the task at different configurations of hyperparameters. In this regard, it was decided to change the algorithm of the RL agent. Then, the twin-delayed deep deterministic (TD3) policy gradient reinforcement learning agent was used for policy tuning. The twin-delayed deep deterministic (TD3) policy gradient algorithm is an actor-critic, model-free, online, off-policy, continuous action-space reinforcement learning method which attempts to learn the policy that maximizes the expected discounted cumulative long-term reward [99]. TD3 has two critics at once, which avoids overtraining the model to local optima (an advantage over DQN and DDPG).

The tuning of hyperparameters for the RL Agent algorithm was carried out using the built-in MATLAB tool, Reinforcement Learning Designer (Fig. 4.7). This tool allows for the configuration of sample time, learn rate for each approximator, various regularization options, exploration models, and other hyperparameters. It's worth noting that the machine learning process is highly sensitive to both model reward and hyperparameters. Therefore, implementing artificial intelligence based on reinforcement learning demands significant user experience. The parameter tuning process is inherently iterative, involving the adjustment of values through trials and errors to identify suitable configurations.

The approximators (critics and actor) are deep neural networks, which are shown in Fig. 4.8. The critics are a single-output Q-value function (value as a function of observation and action) approximators and includes input layers of observations and actions, fully connected (FC) and ReLU layers with 128 or 256 units each and an output layer. The actor is represented as a deterministic actor with a continuous action space, which returns an action value that is a parametrized deterministic function of the observation [100]. The actor includes an input layer of observations, fully connected (FC) and ReLU layers with 128 units each and one output layer for action value.

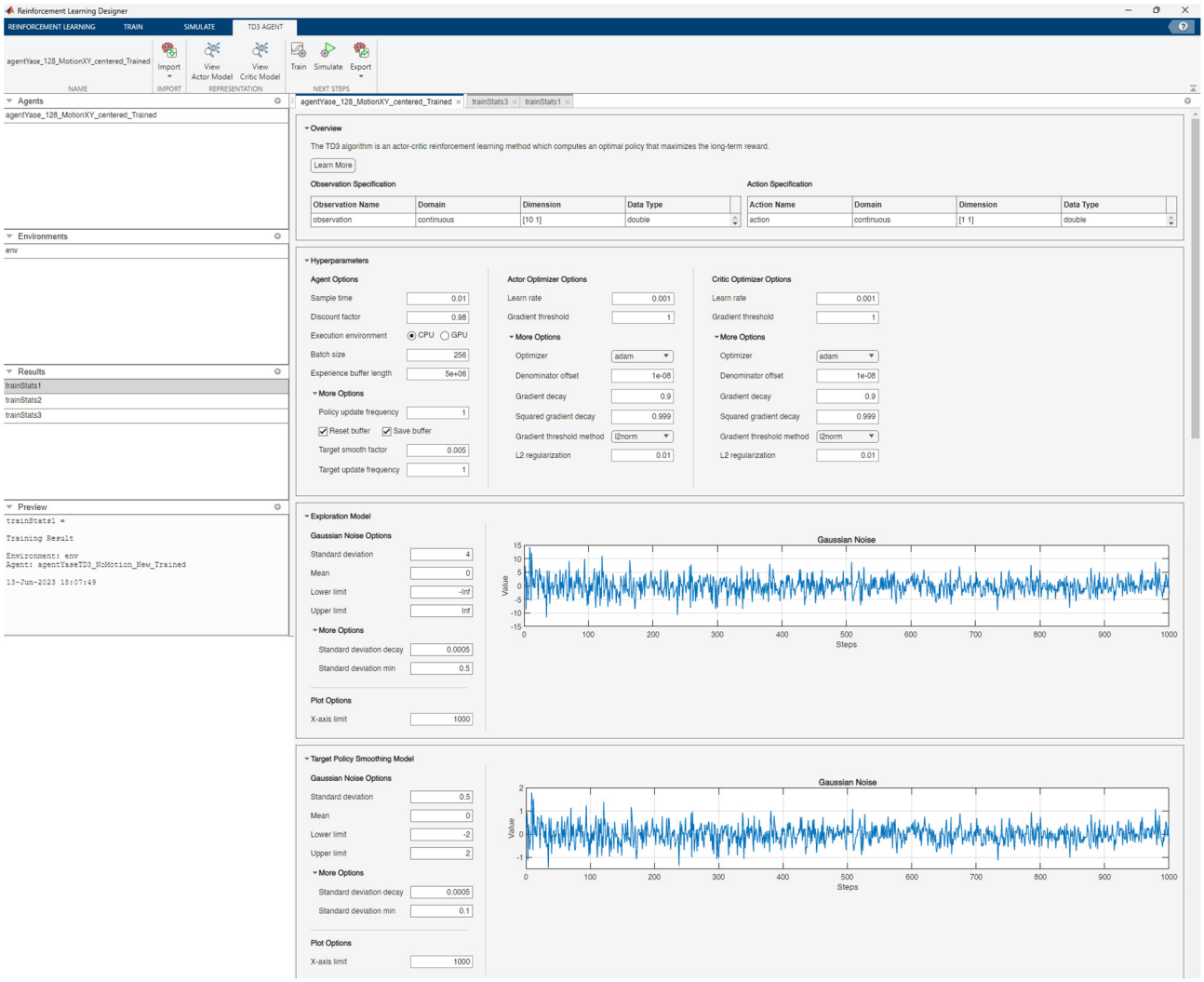


Fig. 4.7 Reinforcement learning designer GUI in MATLAB. Hyperparameters of RL Agent used in training are illustrated

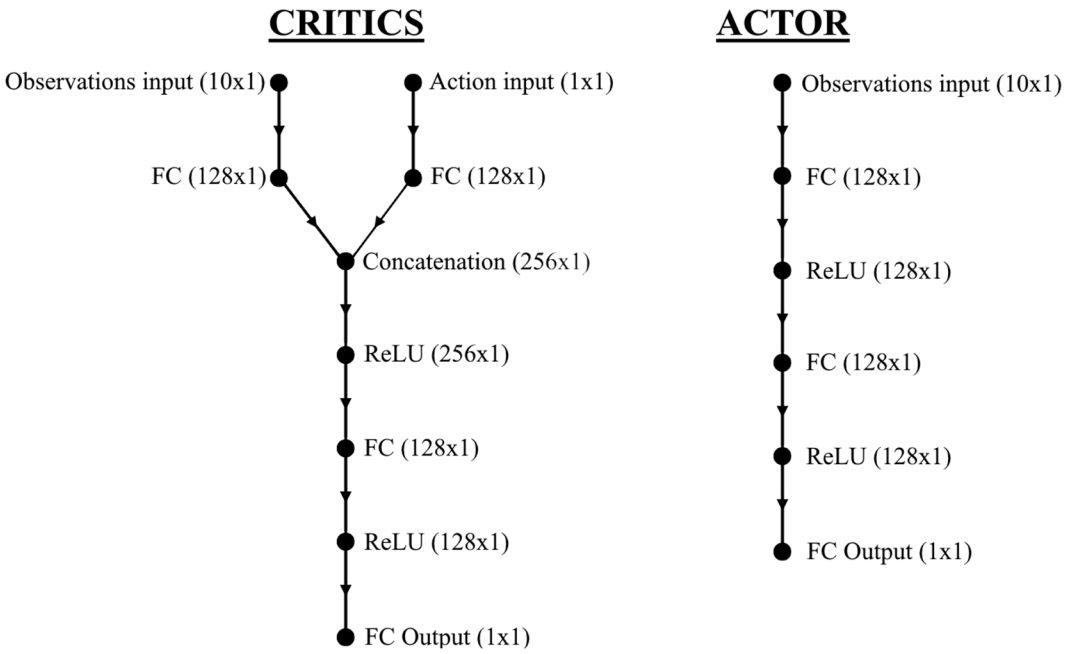


Fig. 4.8 The structure of deep neural networks of critics and actor in pendulum test. The numbers in parentheses indicate the number of units in each layer. "FC" stands for Fully Connected layer, and “ReLU” represents a Rectified Linear Unit layer

The results of training the RL Agent with the described parameters are presented below. The simulated duration of one episode was set to 10 seconds. Fig. 4.9 shows the reward obtained by the RL agent in each training episode, as well as the averaged value over 30 episodes. The training took 158 episodes and automatically terminated when the agent could maintain the suspended load in the target position until the end of the episode for 30 consecutive episodes. From this illustration, it is evident that the training of the RL agent was not smooth. On the contrary, during the first 128 episodes, the RL agent's policy led to extremely incorrect actions, causing Base 2 to move rapidly beyond permissible limits. As a result, these episodes were terminated prematurely. Then, starting from episode 129, the RL agent suddenly arrived at the correct policy and maintained it for the subsequent 29 episodes. In our opinion, such a result is not satisfactory enough. Ideally, training should progress smoothly with a gradually increasing average reward. The most likely reason is an insufficiently well-chosen reward function. However, for testing purposes, this was sufficient, and additional shaping of the reward model and tuning of the hyperparameters was not performed.

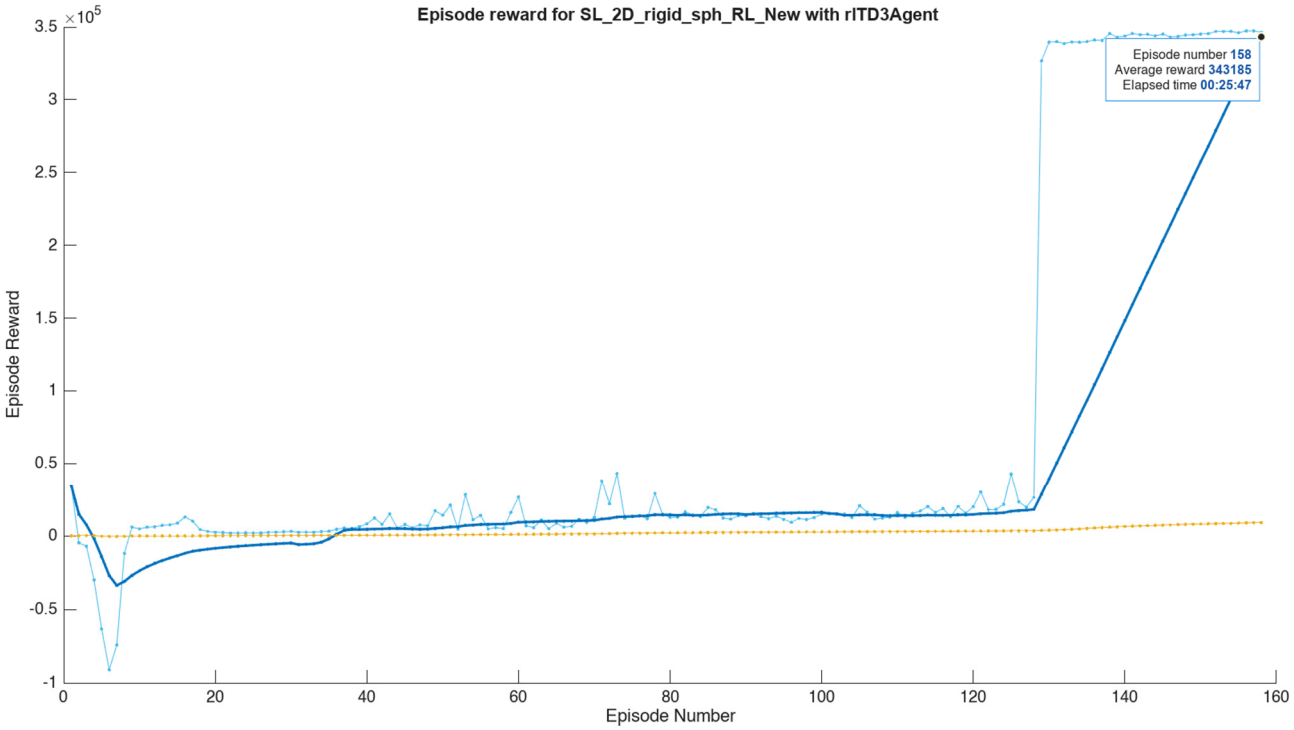


Fig. 4.9 Reward value during training of RL Agent

Upon completing the training, a test computation of the numerical model was performed with the following initial parameters:

- cable length l_c equal to 0.3 m;
- the mass of the load m equal to 1 kg;
- the initial inclination angle θ equal to 20 deg.

The initial state of the model corresponds to Fig. 4.5. The performance of the AI-controlled position of the suspended load with the trained agent is presented in Fig. 4.10. The results indicate that the RL agent took about 2 seconds to bring the suspended load close to the target position and

minimize the swaying (angle). Subsequently, the suspended load remained relatively close to the neutral position despite the movement of Base 1. Although the result could have been even better with more detailed tuning and training of the RL agent. Motion sequence of base 2 and suspended load is illustrated in Fig. 4.11.

With other values of initial parameters lying within the range used for training, the RL agent exhibited a similar pattern. The most challenging case for control is the episode with the maximum cable length. Thus, a trained RL agent has been obtained, acting as a kind of “black box”, where the logic of actions is calibrated for different system states. The test computation of the pendulum with a moving base showed promising results, which can be further applied to the numerical model of the crane ship developed earlier.

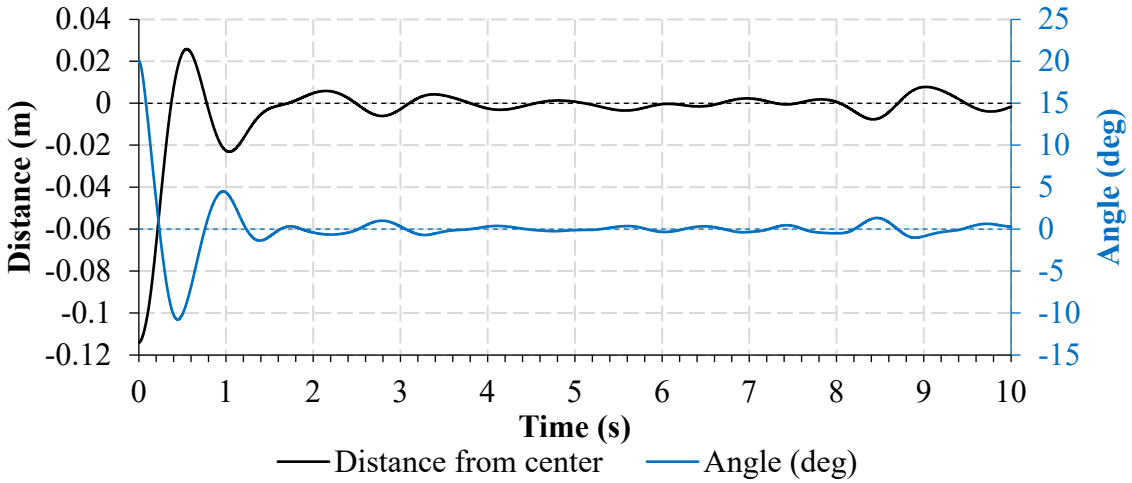


Fig. 4.10 Performance of AI-control by RL Agent on minimizing angle and distance to vertical

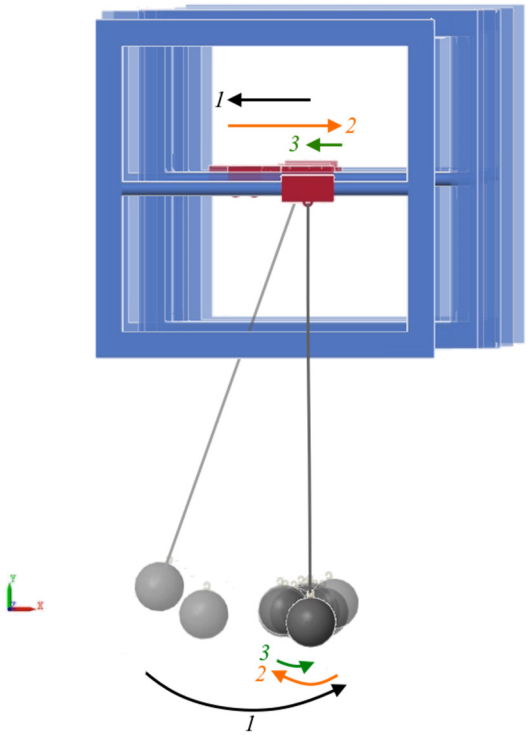


Fig. 4.11 Motion sequence of base 2 and suspended load

4.3. Implementation of AI-control in a crane ship model

The next step in applying AI control for solving the given problem is to integrate it into the numerical model of the crane ship. The RL Agent's task in this case differs from the previous one. Instead of positioning the load at a single point, the decision was made to train the artificial intelligence to automatically move the suspended load to random positions within a certain area, depicted in green on Fig. 4.12. As can be seen from the figure, for simplification purposes, the agent's training zone lies in a single plane. However, theoretically, with this approach, the control system's applicability could be ensured in the entire the crane's operational zone.

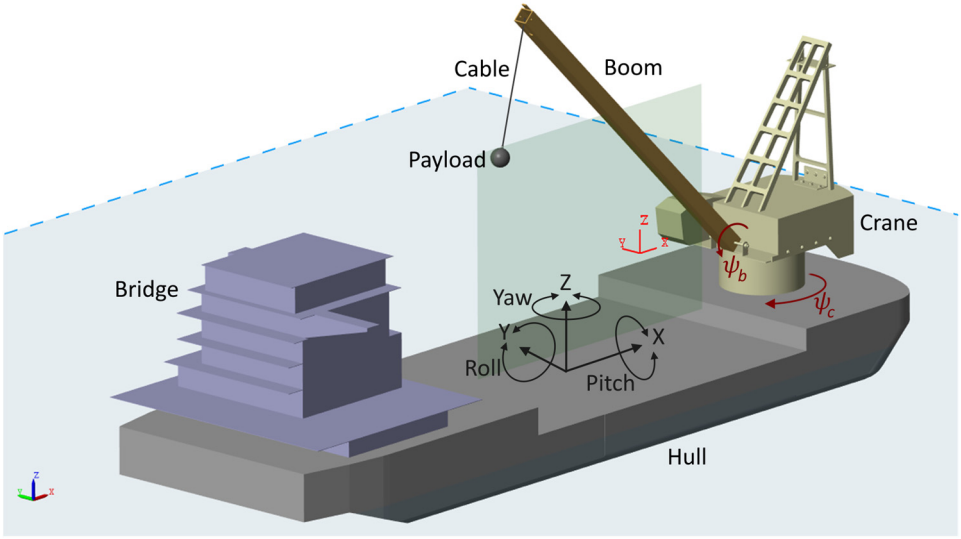


Fig. 4.12 The location of the target points initialization zone

In this problem statement, modeling was conventionally performed in a 2D formulation, although a 3D model was used (sway, roll, and yaw degrees of freedom were locked). To position the load, the RL agent controls the crane's boom joint and the lifting/lowering velocity by inputting signal into a custom block developed earlier. Thus, when waves act on the ship (and theoretically wind and currents), the trained RL Agent will be able to move the suspended load in the horizontal direction by changing the boom angle, and in the vertical direction by changing the cable length. Parameters and components of the simulated system are presented in Fig. 4.13. The dimensions of the green target point initialization zone are 0.635x0.45 m.

During the initialization of the model, a new problem appeared in the implementation of RL agent training when using WEC-Sim libraries together. The problem was the impossibility of simultaneous launching of the dynamic model itself and the machine learning block, since WEC-Sim algorithms assume background launching of dynamics with closing of the Simulink environment. This problem was pointed out to the developer, and as a result, changes were made to the program code [101]. During the period of code correction and revision by the WEC-Sim developers, reinforcement learning was performed on the model with its own blocks for calculating hydrodynamic forces.

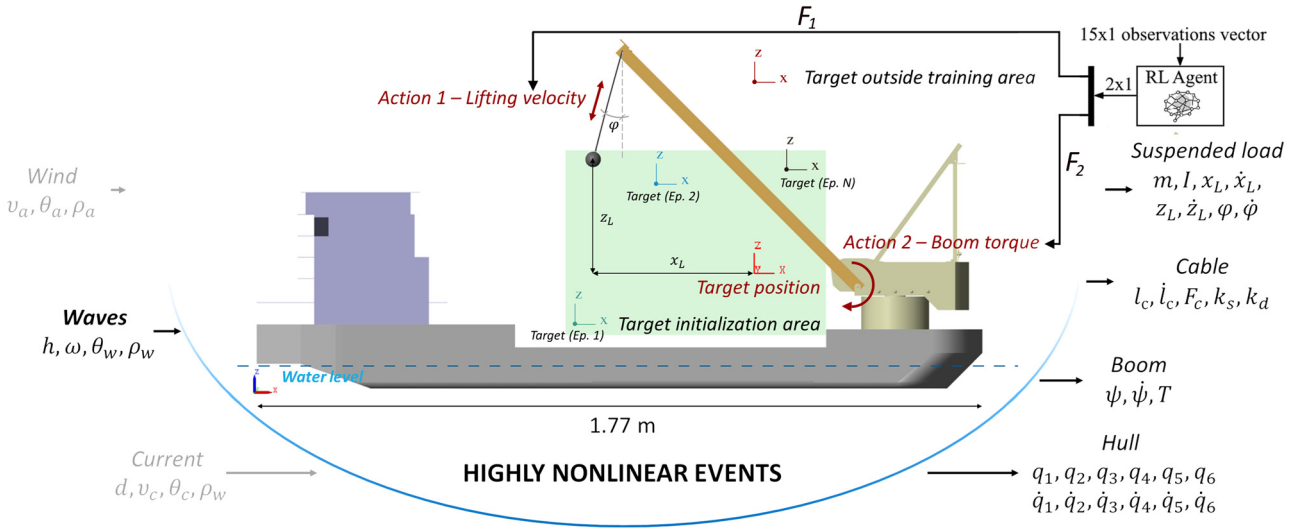


Fig. 4.13 The parameters and components of the simulated system

In this model, the 15 parameters listed in Table 4.2 were used as observations. These observations were selected based on the consideration that they could, to some extent, influence the movements of the load and, consequently, the decision-making logic of the RL agent. For example, movements of the crane ship in the surge and heave degrees of freedom were not particularly crucial and were not included in the observation vector since the distances from the load to the target (x_L and z_L) already determine the load's position in space. At the same time, pitch displacements were added to help the RL agent anticipate the system dynamics at different angles of deviation of the load from the vertical and inclinations of the crane's boom.

Table 4.2 Observations used in training of RL Agent for crane ship

Number	Observations	Symbol
1	Cable length	l_c
2	Cable force	F_c
3	Angular displacements of the crane boom	ψ
4	Angular velocity of the crane boom	$\dot{\psi}$
5	Cosine of the load inclination angle	$\cos\theta_L$
6	Sine of the load inclination angle	$\sin\theta_L$
7	Angular velocity of the cargo	$\dot{\theta}_L$
8	Hull Surge velocity	$\dot{q}_1$
9	Hull Heave velocity	$\dot{q}_2$
10	Hull Pitch displacements	q_3
11	Hull Pitch velocity	$\dot{q}_3$
12	Horizontal distance from cargo to target point	x_L
13	Vertical distance from cargo to target point	z_L
14	Action 1 (lifting velocity) on the previous step	F_1
15	Action 2 (torque) on the previous step	F_2

In accordance with the defined task, the reward function was constructed as follows:

$$r = a \cdot \exp\left(b\sqrt{x_L^2 + z_L^2}\right) + c \cdot B + d \cdot \theta_L^2 + e \cdot C \tag{4.3}$$

where r is a reward value at the current computation step; θ_L is the angular displacement of the load relative to the vertical (rad); x_L and z_L are the horizontal and vertical distance from the load to the target point, respectively (m); B is a prize flag equal to 1 if the distance from the load to the target point is less than 0.02 m; C is a penalty flag equal to 1 if the distance from the load to the target point is more than 0.6 m; a, b, c and a are constant proportional factors.

The term $\sqrt{x_L^2 + z_L^2}$ represents the direct distance from suspended load to target position. Thus, the closer the suspended load is to the target position and the smaller the angular displacements, the greater the reward. If the suspended load exceeds the allowed limits (0.6 m), a penalty is applied, the episode is prematurely terminated ("is done"), and the next training episode begins. A favorable result was obtained with $a = 2$, $b = -10$, $c = 1$, $d = -4$, $e = -50$ and reward model can be described as state-based model:

$$r = \begin{cases} 1 + 2 \cdot \exp\left(-10\sqrt{x_L^2 + z_L^2}\right) - 4\theta_L^2, & if\ \sqrt{x_L^2 + z_L^2} < 0.02\ m \\ 2 \cdot \exp\left(-10\sqrt{x_L^2 + z_L^2}\right) - 4\theta_L^2, & if\ 0.02\ m \le \sqrt{x_L^2 + z_L^2} < 0.6\ m \\ -50 + 2 \cdot \exp\left(-10\sqrt{x_L^2 + z_L^2}\right) - 4\theta_L^2, & if\ \sqrt{x_L^2 + z_L^2} \ge 0.6\ m\ or\ l_c < 0.02\ m \end{cases} \tag{4.4}$$

The graphical representation of reward as a function of distance $\sqrt{x_L^2 + z_L^2}$ and angular displacements θ is shown in Fig. 4.14.

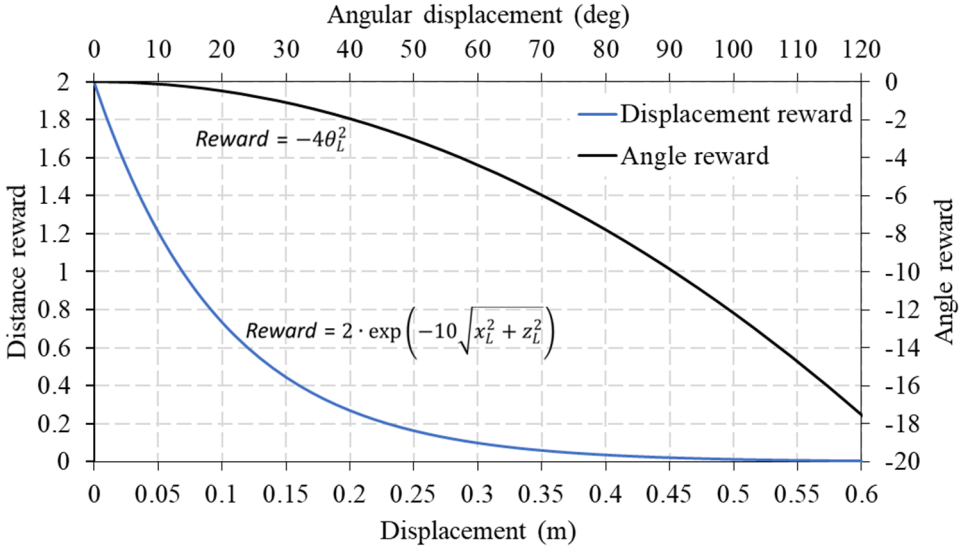


Fig. 4.14 The graphical representation of reward as a function of distance and angular displacements

The model developed in Simulink environment is illustrated in Fig. 4.15 . At the top level of the model there is the RL-agent. It is connected to the entire environment using the “action”, “is done”, “reward" and “observation” ports. The environment is arranged at the next level and includes the crane ship submodel itself, the reward model and the penalty model. Observations are defined in the model as output signals combined by Mux block. The reward and penalty models are designed using math blocks of Simulink library (Fig. 4.16). These blocks process the observation signals and calculate the final reward value at the current calculation step according to Eq. (4.4).

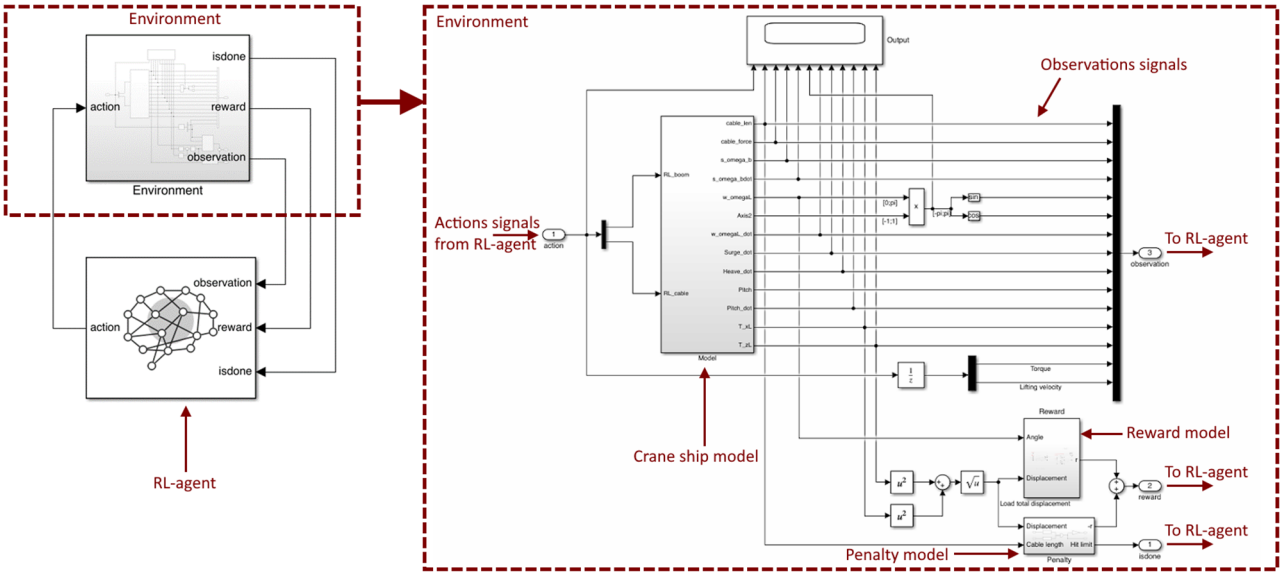


Fig. 4.15 The numerical model used for reinforcement learning of RL-agent in MATLAB/Simulink

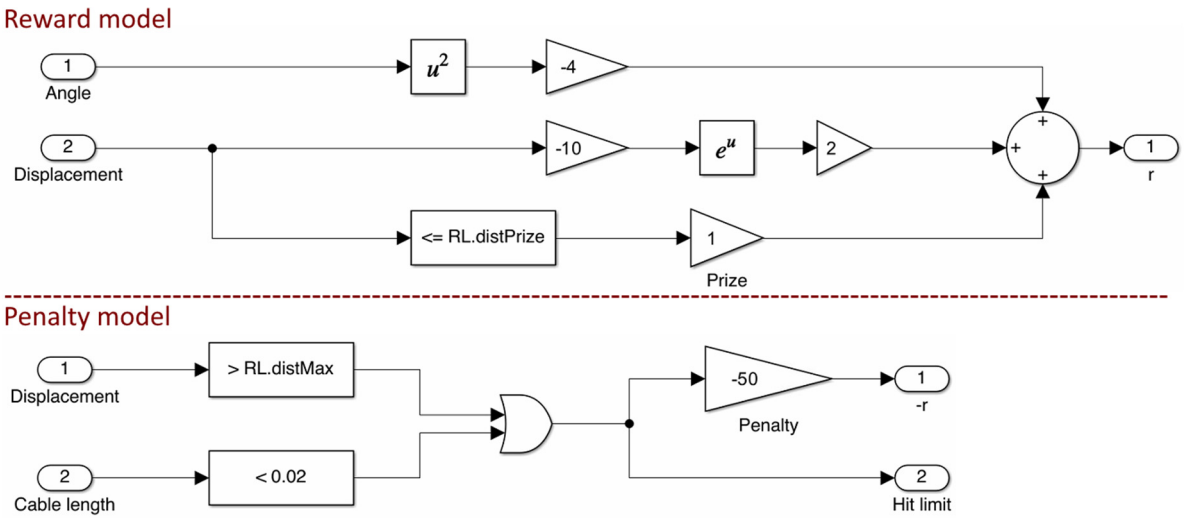


Fig. 4.16 The reward and penalty models

In order to ensure the reliability of AI anti-sway control and positioning of the load during training at each training episode, random values of initial parameters were used:

- cable length l_c – from 0.1 to 0.5 m;
- the mass of the load m – from 0.1 to 2 kg;
- the initial inclination angle φ – from -20 to 20 deg;
- the target position – within the green area.

The values of these parameters within a given range were generated using the Reset function (Appendix C), which was run at the beginning of each training episode. This ensured non-repeating initial states of the model, which will eventually lead to reliable behavior of the RL Agent within the specified ranges.

The first attempts using the same algorithm (TD3) with the same approximator structure did not yield successful results. Several attempts were made with different structures (varying the number of layers and units) and different hyperparameter values, but they also did not lead to the desired outcome. Consequently, another training algorithm, Soft Actor-Critic (SAC) [102, 103], was tested for training the RL Agent. It is a model-free, online, off-policy, actor-critic reinforcement learning method, which computes an optimal policy that maximizes both the long-term expected reward and the entropy of the policy (a measure of policy uncertainty) [103]. Also, in current implementation, the RL Agent also has two critics.

The approximators (critics and actor) are deep neural networks, which are shown in Fig. 4.17. The critic is a single-output Q-value function approximator and includes input layers of observations and actions, fully connected and ReLU layers with 128 units each and an output layer. The actor is represented as a stochastic actor with a continuous action space that generates a random action value within given bounds sampled from a parametrized Gaussian probability distribution. The actor includes an input layer of observations, fully connected and ReLU layers with 128 units each and two output layers for mean and standard deviation.

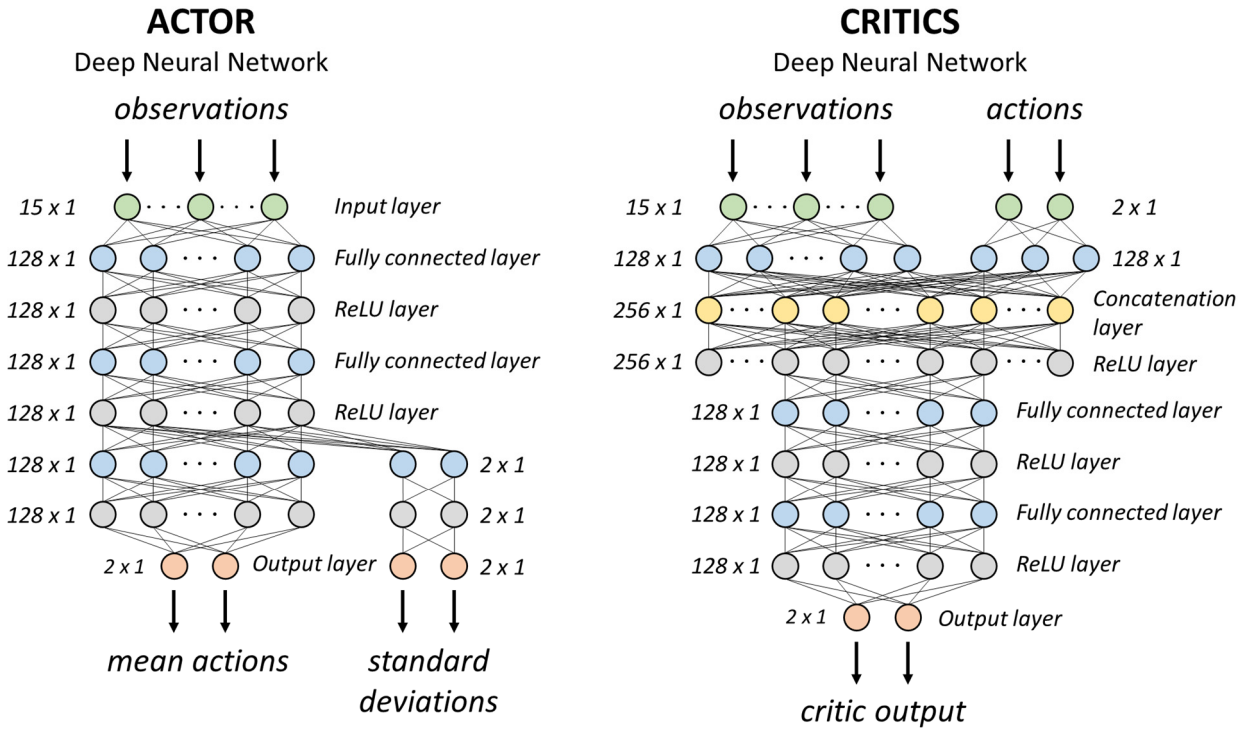


Fig. 4.17 Structure of actor and critics

In the current task, the magnitude of actions taken by the RL agent was limited. The maximum velocity of lifting the load was set to 0.15 m/s, and the maximum torque at the crane joint was set to

15 N·m. Accordingly, the taken actions have different absolute signal magnitudes. Due to this, it became impossible to use Reinforcement learning designer since the current version lacks the ability to configure different values for multiple actions. Moreover, the adopted approximator structure also cannot be configured using this standard tool. Therefore, the complete configuration of the RL agent, including the approximator structure, hyperparameter values, and initialization of initial parameters, was implemented using MATLAB programming language code. The used scripts are presented in Appendix C of this dissertation.

Fig. 4.18 illustrates the reward value earned by the RL Agent at each training episode. The training was stopped after 1000 episodes. From the figure, a smooth increase in reward can be seen after little accumulated experience (about 100 episodes), indicating well-chosen values of hyperparameters. Between 500-600 episodes, a local decrease in the average reward value is observed. The exact cause is difficult to determine, as it could be due to the random selection of unsuccessful episodes in the experience buffer, or due to the approximation of the policy into a local optimum or other reason. However, after the 600th episode, the average reward value was maintained above 2250. It is therefore very important in training not to interrupt it early in order to handle such issues.

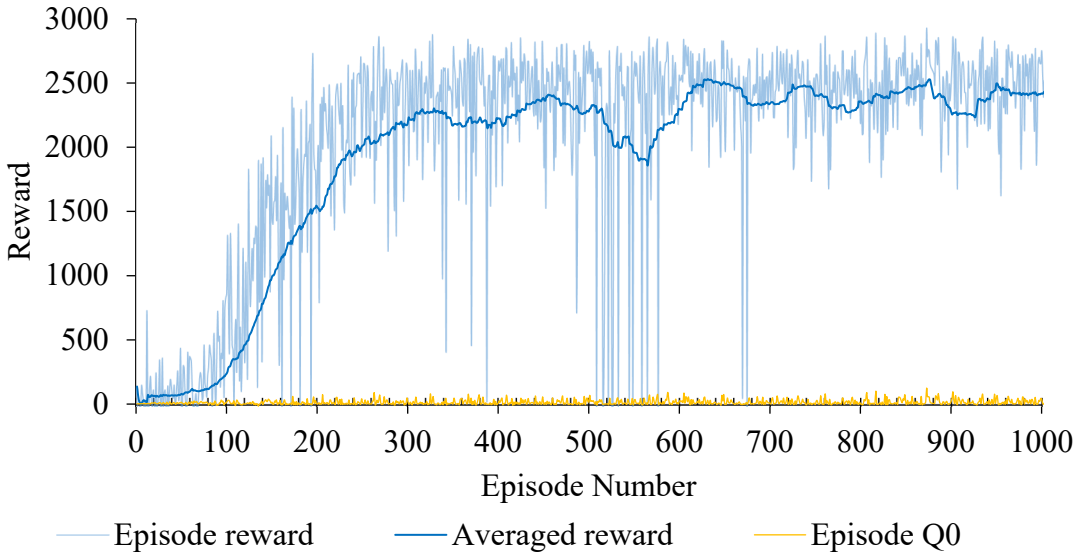


Fig. 4.18 Reward earned by the agent during the training process

Fig. 4.19 illustrates the simulation results of a 10-second episode using the trained RL Agent for three cases. The initial state of the model was similar to Fig. 4.13. The case “No AI control” represents the simulation with the free swinging of the load and without any action from the RL Agent.

The “Outside the training area” case represents the result when the target point is outside the green area. Here we can see a good performance on the task of minimizing the distance to the target. However, the swing of the load (angle from vertical) is not well controlled. Since this location was not captured by the RL Agent's training process, the agent took actions with too large value of the mean deviations of the actuation torque.

The “Within the training area” case represents the result when the target point is inside the green area. Both the angle of deviation of the load from the vertical and distance to the target point smoothly decreases and becomes close to 0 within 3.2 s of the episode. Such performance of the RL Agent is preserved for other episodes where the target point is initialized inside the green training area.

The weak point of the SAC Agent is a slight noise that leads to shaking of the suspended load (Fig. 4.20) because of standard deviation part in the signal output from agent. The quality of actuation can be improved by more accurate tuning of hyperparameters. In addition, further training of the RL Agent may be useful for policy optimization in terms of standard deviation values but is not considered in this thesis.

The sequence of motion of the suspended load and crane boom during described simulated event is shown in Fig. 4.21.

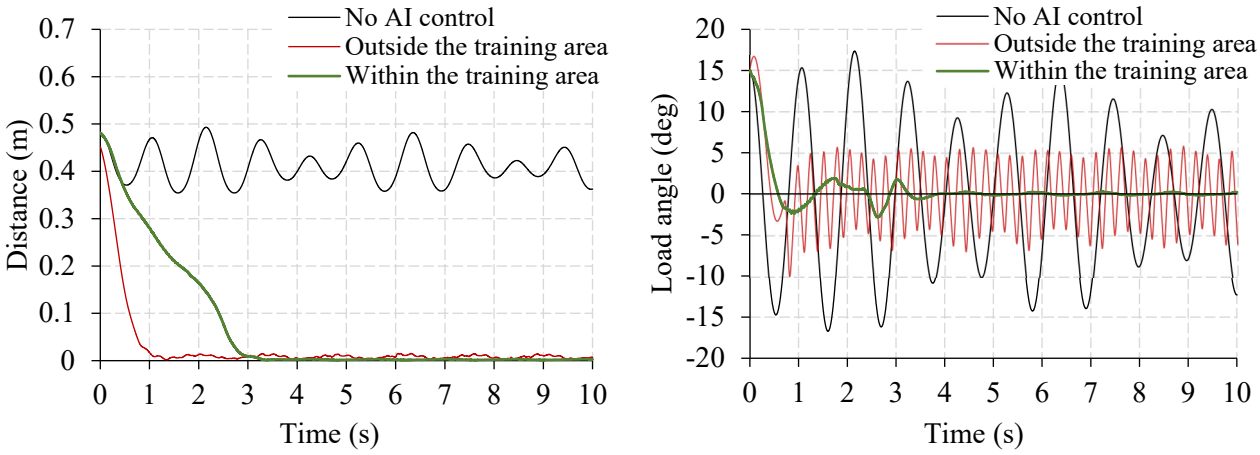


Fig. 4.19 Distance to target point and Angle from vertical

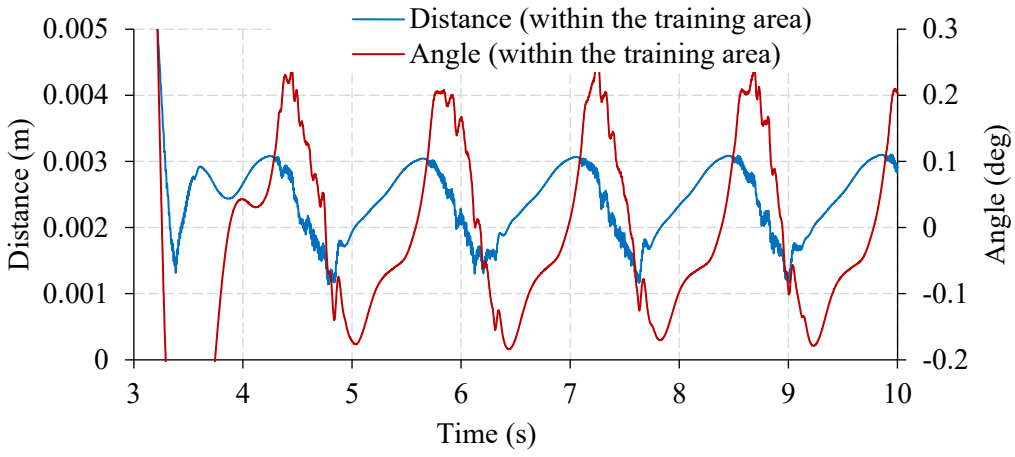


Fig. 4.20 Noisy response of suspended load

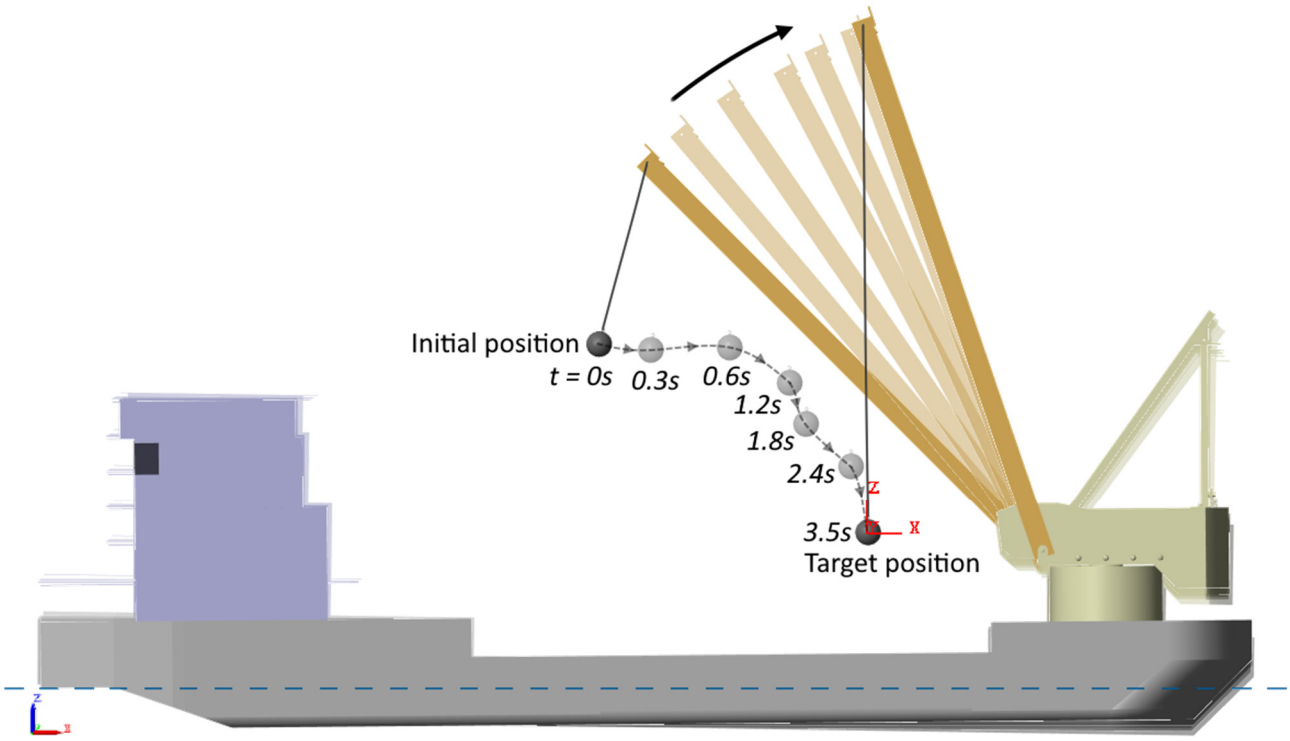


Fig. 4.21 Motion sequence of suspended load and crane boom during simulated event

In order to show the capabilities of AI-positioning of the suspended load, several test simulations were performed. A crane ship model with AI-control of the crane boom and cable was updated with interactive control blocks in the same way as described in Section 3 of this thesis. The blocks were similarly linked to the hardware – joysticks and I/O board. However, in the current case, it is assumed that the operator will not control the crane itself, but the position of the target point. In other words, the user can move the virtual point in space, and the trained RL-agent should automatically follow the target position. The concept of the implemented approach is presented in Fig. 4.22.

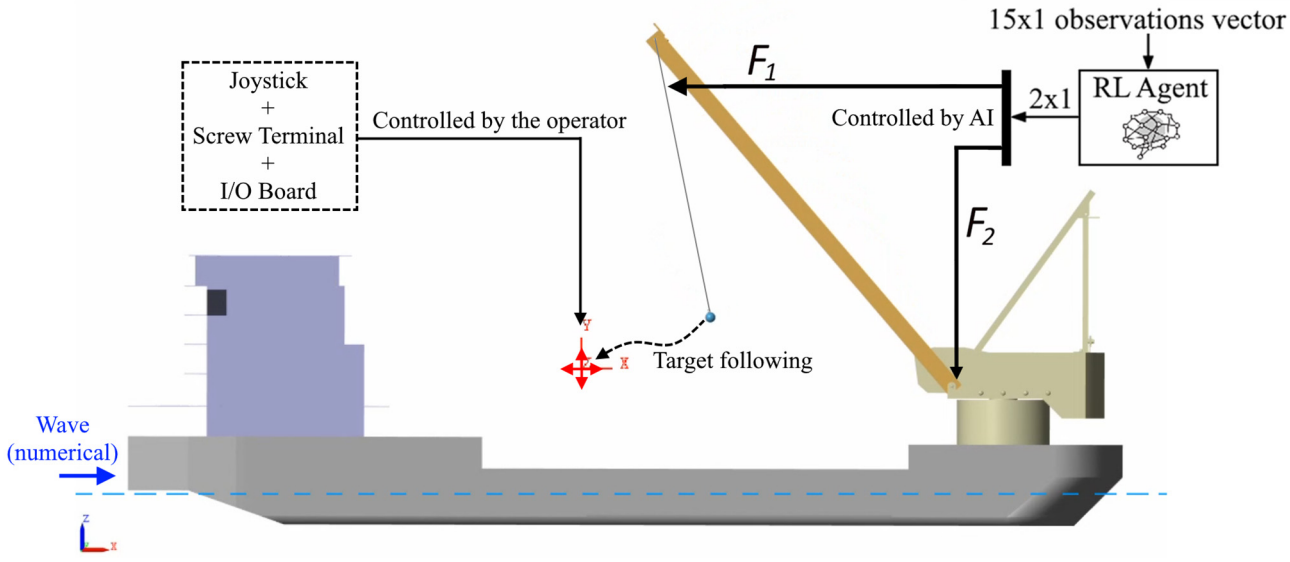


Fig. 4.22 Real-time AI cargo positioning concept

To test the approach, 4 computational cases differing in mass and shape of the load were simulated. The mass of the suspended load was chosen within the range of values used previously in training the RL agent - 0.16 kg, 0.7 kg, 1.5 kg and 2 kg. The model time of one case is 30 sec. The sequence of cargo movements for all four computational cases is presented in Fig. 4.23-4.26.

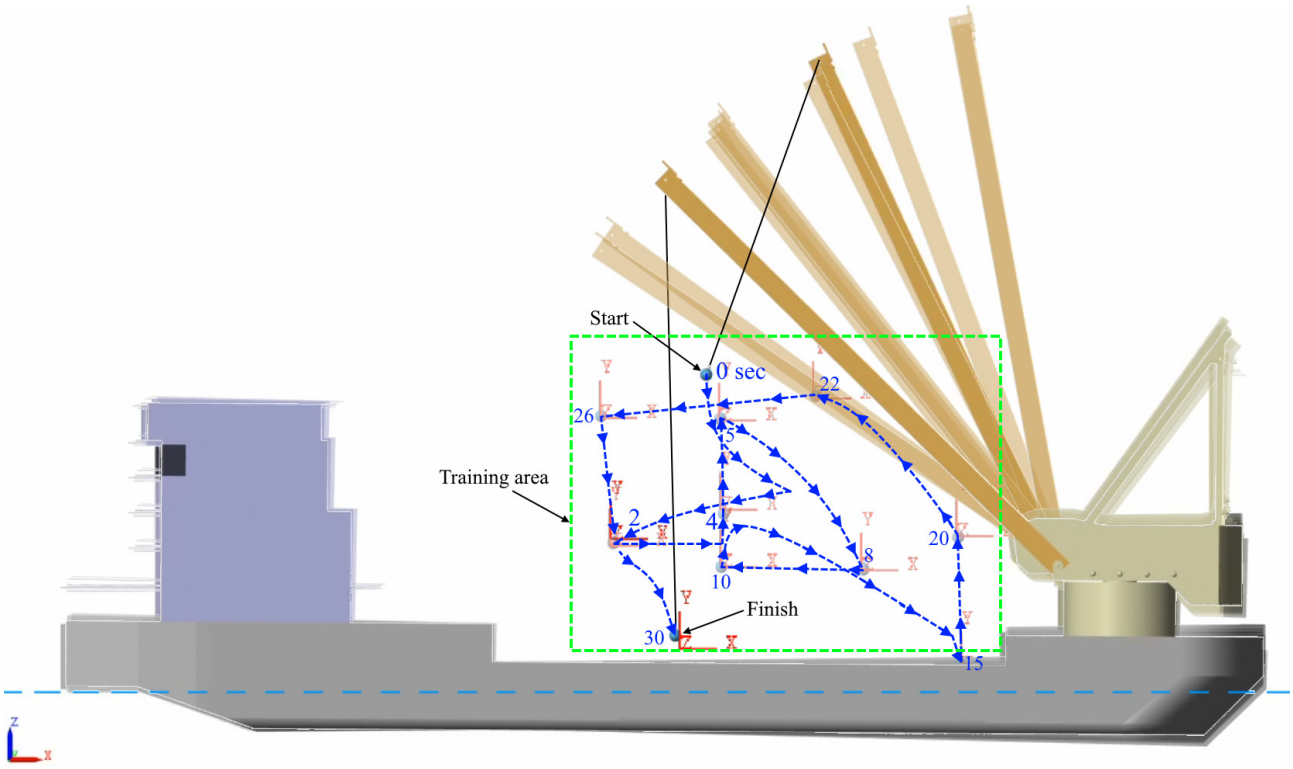


Fig. 4.23 Motion sequence in simulated event with AI-positioning of a 0.16kg load [104]

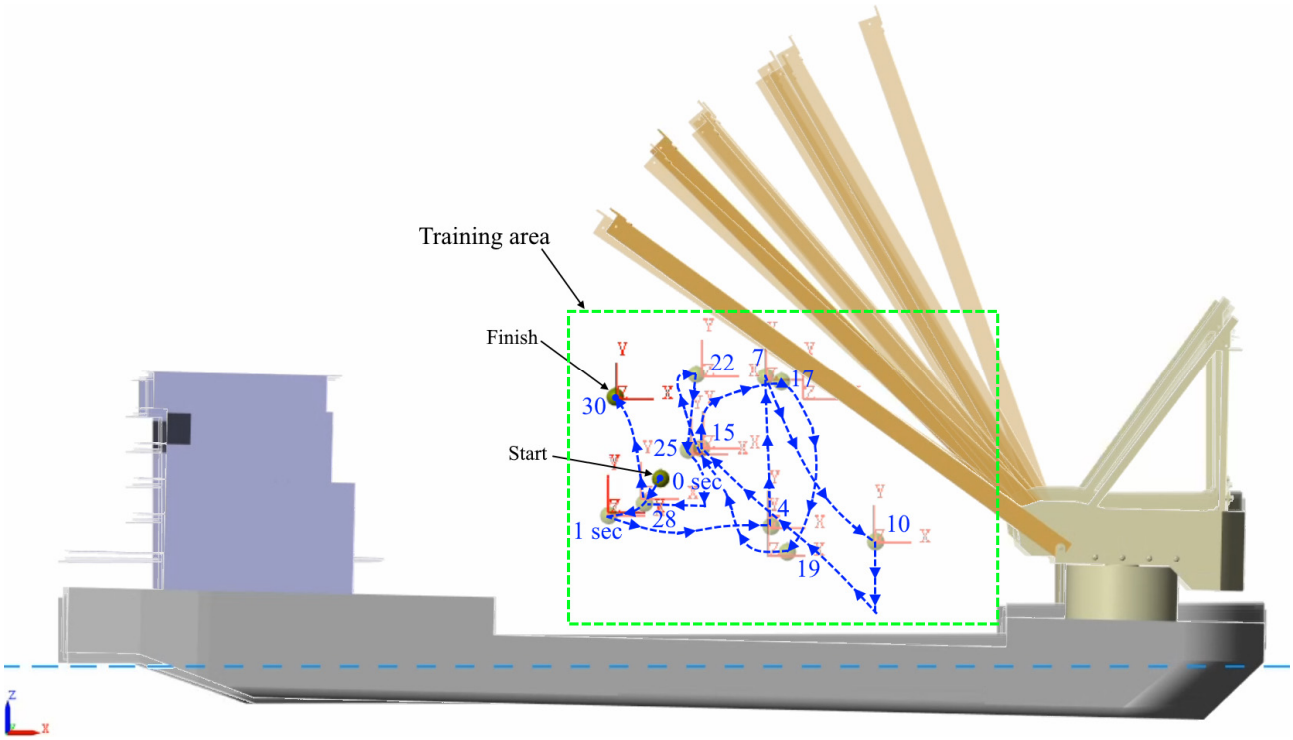


Fig. 4.24 Motion sequence in simulated event with AI-positioning of a 0.7kg load [105]

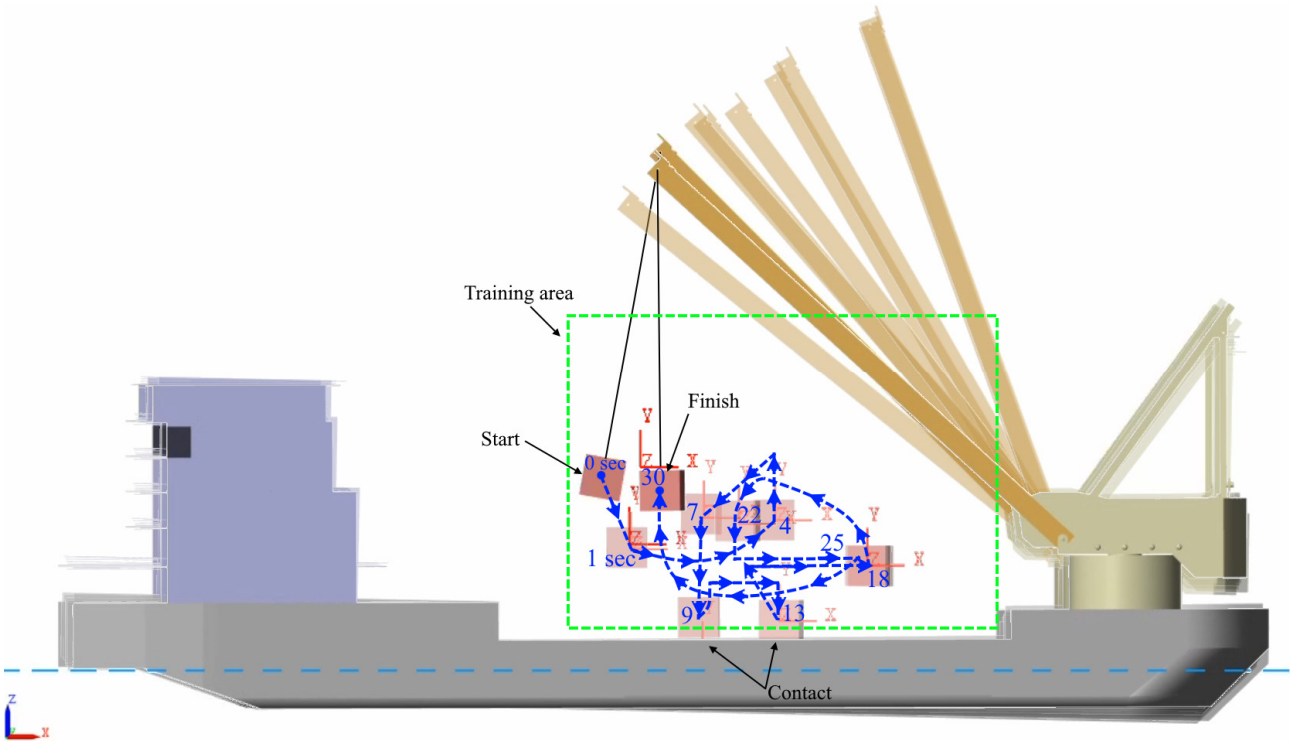


Fig. 4.25 Motion sequence in simulated event with AI-positioning of a 1.5kg load [106]

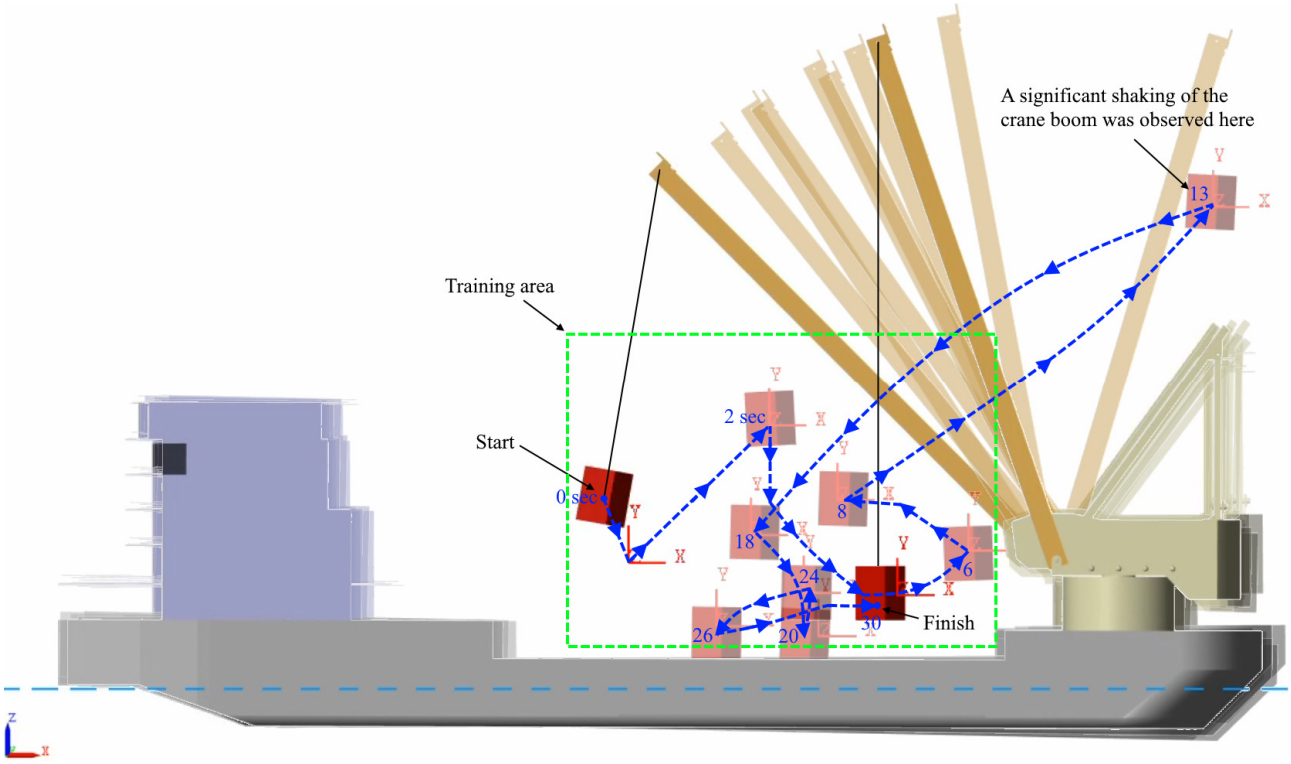


Fig. 4.26 Motion sequence in simulated event with AI-positioning of a 2kg load [107]

Analyzing the qualitative picture of AI-control, the following conclusions can be drawn. In general, AI-control is performed with good quality. RL-agent understands its main task (following the point), and takes necessary actions to minimize the distance from the cargo to the target. Low mass payloads are the easiest to control. As the mass of the payload increases, the control signal from the RL agent becomes more unstable. This effect may be due to the need to apply maximum torque and lift speed to “catch up” to the target point, while low mass payloads require little effort. Videos

of these simulations are available at references [104-107], where it is possible to clearly see the load following the target.

Weaknesses of the trained artificial intelligence were also clearly manifested. First, the control quality decreases significantly when the load approaches the boundaries of the training area, which is represented by the irregularity of the control signal and, consequently, by the shaking of the suspended load. The reason for this is most likely an insufficient amount of training episodes in which the target position was located near the boundaries of the zone. Accordingly, the RL agent was not sufficiently trained under these conditions and the agent's policy does not yet provide a qualitative determination of the required action magnitude. Second, a significant increase in signal irregularity is observed when the cargo comes into contact with the ship deck and the target position is located below the deck. Since learning under these conditions was not performed in principle, the RL agent "does not understand" the presence of the obstacle (deck) and keeps following its policy of increasing the length of the cable to continue lowering the cargo. This problem can be solved, for example, by introducing conditions or "states" into the model. If the force in the cable is 0 (another indicator variable could be used), the control signal of the RL-agent could be multiplied by 0, thus limiting the control when the cable is in a loosened state.

Nevertheless, as a result, we obtained a trained RL Agent, which is characterized by nonlinear behavior and can show high efficiency under different states of the crane ship system and target point positions. RL Agent automatically takes the most appropriate actions based on the optimized policy. Furthermore, theoretically, there is no need to train the RL agent at different wave frequencies because its actions are based on the dynamic response of the ship. In other words, it will automatically respond to changes in the frequency and amplitude of ship motions depending on its instantaneous velocity. The changing dynamic response of the ship is a result of the wave action, but the wave itself does not alter the fundamental observation principles of the model's state.

This approach is scalable to a full three-dimensional problem. For this purpose, it is sufficient to add another action to the RL Agent that controls the actuation torque of the crane base and update the observations and the reward model to take into account the third dimension.

4.4. Section conclusions

In this section, the application of artificial intelligence for the automatic control of crane ship components with the aim of positioning the load at a specified point and minimizing load sway was discussed. Initially, the method was tested on a simple pendulum model with a moving base and then applied to the crane ship model itself. The results showed that although the initial tuning of learning parameters (reward model and hyperparameters) is a challenging task, the effectiveness and the

ability of AI control to automatically adapt to various system states potentially make it a promising solution for crane ship applications.

The developed numerical model was tested on the task of load positioning, in which the position of the target point was interactively controlled by the operator from joysticks, and the trained AI automatically followed the moving target point. Overall, the quality of the AI control was acceptable. Obvious drawbacks are the presence of noise in the control signal produced by the RL agent, as well as the need for careful consideration of possible model states in which AI control is inappropriate (e.g., when lowering cargo onto the ship’s deck).

In our opinion, with a detailed approach and careful tuning of artificial intelligence, it has the potential to fundamentally change the way crane operators work. For example, instead of manually controlling various crane components with multiple manipulators, an operator may only need a few levers to move a virtual target (in virtual or augmented reality), and artificial intelligence would automatically follow this target, selecting safe and optimal actions for the crane's actuators.

This is just one example of the application of the approach developed in this dissertation. Other practical areas that could benefit from this approach include interactive simulators, digital twins, unmanned ships, and other developments that require intelligent control systems.

Conclusion

In this dissertation, research was conducted aimed at developing a numerical method for modeling the coupled dynamics of a floating crane and a suspended load. The developed method involves constructing a multi-body system of the crane ship in the MATLAB/Simulink environment using blocks from the Simscape library and MATLAB-based software WEC-Sim. In this process, hydrodynamic forces are computed within the MBD-loop solver based on coefficients calculated using the Python-based linear potential flow solver Capytaine. The choice of software for developing the numerical model was justified by the need for scalability and accessibility. When comparing open-source potential flow solvers, Capytaine proved to be more efficient than NEMOH and HAMS. It also offers the greatest customization convenience, making it the optimal choice for determining the hydrodynamic coefficients of the submerged hull of the ship. To eliminate the assumption of potential flow theory regarding inviscid fluid, linear viscous and quadratic drag coefficients are additionally introduced, determined through CFD modeling based on RANS equations.

The hydrodynamics of the crane ship, modeled based on potential flow theory, underwent validation through a comparison of simulation results with experimental data. Laboratory experiments involved tracking and recording the wave-induced motions of a 1:50 scale model of the crane ship. By introducing several enhancements to the conventional linear equation of motion, we achieved excellent convergence with the experimental data. Consequently, the developed method demonstrates the necessary accuracy for addressing the engineering challenge of modeling the dynamic response of the crane ship.

The scalability of the software enables the development and computation of the dynamics of the entire numerical model within a unified solution (kernel), thereby ensuring calculation speeds exceeding real-time. High-performance computing, in turn, has opened opportunities for implementing interactive control of the numerical model in real-time during the calculation process. Interactive control was realized by linking the numerical model with hardware through built-in MATLAB tools. Collision detection for rigid bodies was incorporated into the numerical model, allowing the simulation of stages involving the lift-off and installation of the load. Furthermore, to facilitate control of the load lifting and lowering, a custom block was developed in the Simscape programming language. This block is designed to consider changes in the cable's state and read input signals from the hardware controlled by operator of the crane ship.

As a result of implementing the interactive control, a method for positioning and compensation of swaying of the suspended load was developed based on artificial intelligence, utilizing built-in MATLAB tools. Artificial intelligence training was conducted through reinforcement learning by simulating a variety of cases with randomly varying initial conditions. For this purpose, a structure of deep neural network policy approximators was developed using the MATLAB programming language. The trained AI demonstrated high performance in minimizing

swaying under nonlinear response of the crane ship. Furthermore, the AI can not only minimize swaying but also automatically position the load at a user-defined point.

Of course, the proposed model has some limitations. Among them, the following can be highlighted:

The theory of potential flow gives large errors, where high wave nonlinearity can occur and significant vortex effects can appear. Accordingly, the dynamics of the vessel and, as a consequence, the dynamics of the cargo cannot be accurately predicted in this case even with introduction of additional viscous damping. Nevertheless, the area of application of the proposed solution is the dynamic analysis of the vessel during marine operations, where the sea conditions are not so severe and can be successfully described using the proposed approach. At the same time, the possibility to consider multiple sea states is an indisputable advantage and one of the necessary.

The method validation was conducted using experimental data of events with regular waves. It is evident that in real conditions, the sea is characterized by a complex superposition of waves of varying frequencies, resulting in irregular waves. Therefore, for a more comprehensive confirmation of the method's reliability, experiments should be conducted in a wave basin with waves of different frequencies and directions. This represents a significant challenge for future research directions.

The splash zone crossing and submerging stages are not considered in this research. Today there are some solutions, mentioned in the review, which take into account the buoyancy force as a function of the submerged volume. However, this stage requires a more detailed analysis, which one of the difficult and relevant problems in the application of the potential flow theory. The problem is that the submerged body is exposed not only to the buoyancy force but also to other natural forces (wind over water, sea waves, and currents). This fact raises the task of analyzing the load motions to a new level since it requires the real-time calculation of all its hydrodynamic coefficients in different drafts.

The custom block modeling the flexibility of the cable takes into account only linear flexibility and does not consider its inertia. In certain cases, these features can be significant. Consequently, the algorithm of the custom block can be improved for a more detailed modeling of the hoisting system. Additionally, among interesting directions of research, the author may highlight cable vibration analysis, modeling of nonlinear elasticity, and accounting for Karman vortex due to wind effects.

The simple linear penalty-based contact model showed some issues when simulating the moment of lifting the load from the deck due to spring-like behavior. Consequently, this part can be improved. Among possible solutions, one can consider replacing the simple linear elastic contact model with a nonlinear contact algorithm or implementing a force-control of cable with zero-crossing event handling.

Finally, **the AI control methodology, while demonstrating good performance on a macro scale, still has some issues related to the shaking of the output signal.** These problems arise due to the lack of smoothness of the policy. Consequently, further research is necessary to make this control method more reliable and applicable for enhancing the safety of maritime operations. As an option, conducting detailed reward shaping or developing a more complex mechanism involving multiple RL Agents responsible for adjusting the output signal could be considered. Additionally, a potentially effective solution might involve combining AI with standard linear control theory, which could assist the AI in controlling movements in the near-neutral position.

Thus, this research area is full of relevant problems and needs further development of new solutions. The numerical modeling method developed in this thesis has the potential to find widespread application in predicting the motion of crane ships at sea and decreasing the sway of suspended loads to ensure safe and highly accurate offshore crane operations.

References

1. Newman J. N. Marine Hydrodynamics, 40th Anniversary Edition / Foreword by John Grue, MIT Press, 2018, p. 448.

2. Cummins W. E. The impulse response function and ship motions, *Schiffstechnik*, 9, 1962, 101-109. URL: http://hdl.handle.net/1721.3/49049.

3. Faltinsen O. M. Sea loads on ships and Offshore structures. Cambridge University Press, 1998, p. 328.

4. Shabana A. A. Dynamics of Multibody Systems. Cambridge: Cambridge University Press, 2020. https://doi.org/10.1017/9781108757553.

5. Ellermann K., Kreuzer E., Markiewicz M. Nonlinear Dynamics of Floating Cranes. Nonlinear Dynamics, 27, 2002, 107-183. https://doi.org/10.1023/A:1014256405213.

6. Cha J.-H., Roh M.-I., Lee K.-Y. Integrated simulation framework for the process planning of ships and offshore structures. Robotics and Computer-Integrated Manufacturing, 26(5), 2010, 430-453. https://doi.org/10.1016/j.rcim.2010.01.001.

7. Open Dynamics Engine. URL: http://www.ode.org/ (Accessed 03.12.2023).

8. OpenSceneGraph. URL: http://www.openscenegraph.com/ (Accessed 03.12.2023).

9. Cha J.-H., Roh M.-I., Lee K.-Y. Dynamic response simulation of a heavy cargo suspended by a floating crane based on multibody system dynamics. Ocean Engineering, 37(14-15), 2010, 1273-1291. https://doi.org/10.1016/j.oceaneng.2010.06.008.

10. Ku N., Ha S. Dynamic response analysis of heavy load lifting operation in shipyard using multi-cranes. Ocean Engineering, 83, 2014, 63-75. https://doi.org/10.1016/j.oceaneng.2014.03.026.

11. Ha S., Ku N.-K., Roh M.-I., Hwang H.-J. Multibody system dynamics simulator for process simulation of ships and offshore plants in shipyards. Advances in Engineering Software, 85, 2015, 12-25. https://doi.org/10.1016/j.advengsoft.2015.02.008

12. Hong J.-W., Roh M.-I., Ham S.-H., Ha S. 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), 2016, 15. https://doi.org/10.6119/JMST-016-0323-1.

13. Zullah M. A., Lee Y.-H. Review of Fluid Structure Interaction Methods Application to Floating Wave Energy Converter. International Journal of Fluid Machinery and Systems, 11(1), 2018, 63-76. https://doi.org/10.5293/IJFMS.2018.11.1.063.

14. Wang Y., Tang Z., Wang L., Wang X. Linear and quadratic damping coefficients of a single module of a very large floating structure over variable bathymetry: Physical and numerical free-decay experiments. Journal of Ocean Engineering and Science, 7(6), 2022, 607-618. https://doi.org/10.1016/j.joes.2021.10.011.

15. Michailides C. VD-PQ; A Velocity-Dependent Viscous Damping Model for Wave-Structure Interaction Analysis. Journal of Marine Science and Engineering, 9(2), 2021, 175. https://doi.org/10.3390/jmse9020175.

16. Newman J. N. Panel Methods in Marine Hydrodynamics. 11th Australasian Fluid Mechanics Conference, December 14-19, 1992. Keynote paper K-2, 123-129.

17. Yang X.-R., Gan Q.-M., Wang Y.-H., Zhou Y.-B. Dynamic reliability analysis of the crane ship lifting load system. Journal of Marine Science and Technology, 25(5), 2017, 552-562. https://doi.org/10.6119/JMST-017-0621-1.

18. Peña B., McDougall A. An Investigation Into the Limitations of the Panel Method and the Gap Effect for a Fixed and a Floating Structure Subject to Waves. Proceedings of the ASME 2016 35th International Conference on Ocean, Offshore and Arctic Engineering. Volume 7: Ocean Engineering. Busan, South Korea. June 19-24, 2016. V007T06A048. ASME. https://doi.org/10.1115/OMAE2016-54121.

19. Shigunov V., el Moctar O., Papanikolaou A., Potthoff R., Liu S. International benchmark study on numerical simulation methods for the prediction of manoeuvrability of ships in waves. Ocean Engineering, 165, 2018, 365–385. https://doi.org/10.1016/j.oceaneng.2018.07.031.

20. Liu S., Sprenger F., Papanikolaou A., Dafermos G., Zaraphonitis G. Experimental and numerical studies on linear and nonlinear seakeeping phenomena of the DTC ship in regular waves. Ship Technology Research, 68(1), 2021, 41–61. https://doi.org/10.1080/09377255.2020.1857007.

21. Eskilsson C., Pashami S., Holst A., Palm J. A Hybrid linear potential flow - machine learning model for enhanced prediction of WEC performance. Proceedings of the EWTEC, 15, 2023. https://doi.org/10.36688/ewtec-2023-321.

22. Clauss G., Riekert T. Operational limitations of offshore crane vessels. Proceedings of the Offshore Technology Conference (OTC), Houston, TX, OTC 6217, Vol. I, OTC, 1990, 161–170. https://doi.org/10.4043/6217-MS.

23. Jeong D.-H., Roh M.-I., Ham S.-H. Lifting simulation of an offshore supply vessel considering various operating conditions. Advances in Mechanical Engineering, 8(6), 2016, 1–13. https://doi.org/10.1177/1687814016654633.

24. Viswanathan S., Satheesh Babu P.K., Holden C., Egeland O. A Co-simulation Methodology for Offshore Load Handling. Proceedings of the International Conference on Advances in Naval and Ocean Engineering (iCANOE ’21), 2021.

25. Landsverk R., Zhou J., Hovland G., Zhang H. Modeling of Offshore Crane and Marine Craft in Wave Motion. 15th IEEE Conference on Industrial Electronics and Applications (ICIEA), Kristiansand, Norway, 2020, 1463-1470. https://doi.org/10.1109/ICIEA48937.2020.9248110.

26. Fossen T. I., Perez T. Marine Systems Simulator (MSS), 2004. URL: https://github.com/cybergalactic/MSS.

27. Vorhölter H., Hatecke H., Feder D.-F. Design study of floating crane vessels for lifting operations in the offshore wind industry. Proceedings of International marine design conference, The University of Tokyo, Tokyo, 11-14 May 2015, 2015.

28. Li X., Roh M.-I., Ham S.-H. A collaborative simulation in shipbuilding and the offshore installation based on the integration of the dynamic analysis, virtual reality, and control devices. International Journal of Naval Architecture and Ocean Engineering, 11(2), 2019, 699–722. https://doi.org/10.1016/j.ijnaoe.2019.02.010.

29. Chu Y., Sanfilippo F., Æsøy V., Zhang H. An effective heave compensation and anti-sway control approach for offshore hydraulic crane operations. Proceedings of 2014 IEEE International Conference on Mechatronics and Automation, August 3-6, Tianjin, China, 1282–1287. https://doi.org/10.1109/icma.2014.6885884.

30. Chu Y., Æsøy V., Ehlers S., Zhang H. Integrated multi-domain system modelling and simulation for offshore crane operations. Ship Technology Research (Schiffstechnik), 62(1), 2015, 36-46. https://doi.org/10.1179/0937725515Z.0000000004.

31. Ye J., Reppa V., Negenborn R. R. Backstepping Control of Heavy Lift Operations with Crane Vessels. IFAC-PapersOnLine, 53(2), 2020, 14704–14709. https://doi.org/10.1016/j.ifacol.2020.12.1836.

32. Sun N., Fang Y., Chen H. A New Antiswing Control Method for Underactuated Cranes With Unmodeled Uncertainties: Theoretical Design and Hardware Experiments. IEEE Transactions on Industrial Electronics, 62(1), 453-465, 2015. https://doi.org/10.1109/TIE.2014.2327569.

33. Wang J., Liu K., Wang S., Chen H., Sun Y., Niu A., Li H. Dynamic analysis and experiment of underactuated double-pendulum anti-swing device for ship-mounted jib cranes. Polish Maritime Research, 29(4), 2022, 145-154. https://doi.org/10.2478/pomr-2022-0052.

34. Chen H., Wang X., Benbouzid M., Charpentier J.-F., Aït-Ahmed N., Han J. Improved Fractional-Order PID Controller of a PMSM-Based Wave Compensation System for Offshore Ship Cranes. Journal of Marine Science and Engineering, 10(9), 2022, 1238. https://doi.org/10.3390/jmse10091238.

35. Sun M, Wang S, Han G, An L, Chen H, Sun Y. Modeling and Dynamic Analysis of a Triple-Tagline Anti-Swing System for Marine Cranes in an Offshore Environment. Journal of Marine Science and Engineering, 10(8), 2022, 1146. https://doi.org/10.3390/jmse10081146.

36. Fossen T. I. Handbook of Marine Craft Hydrodynamics and Motion Control / Thor Fossen. John Wiley & Sons, Ltd, 2011, p. 582.

37. Makarov O., Harada T. Effective approach to modeling the hydrodynamic response of a multibody system with suspended load in the time domain. Proceedings of Jc-IFToMM International Symposium, 5, 2022, 95–102. https://doi.org/10.57272/jciftomm.5.0_95.

38. Box G. E. P., Draper N. R. Empirical Model-Building and Response Surfaces. John Wiley & Sons, New York, 1987, p. 688.

39. Wave Basin with Multidirectional Random Wave Generator | TOYO CONSTRUCTION CO., LTD. URL: https://www.toyo-const.co.jp/en/technology/2593.html (Accessed 22.11.2023).

40. Ogden D., Ruehl K., Yu Y.-H., Keester A., Forbush D., Leon J., Tom N. Review of WEC-Sim development and applications. International Marine Energy Journal, 5(3), 2022, 293–303. https://doi.org/10.36688/imej.5.293-303.

41. Ruehl K., Ogden D., Yu Y. H., Keester A., Tom N., Forbush D., Leon J. WEC-Sim (Version v4.4), 2021. https://doi.org/10.5281/zenodo.5608563.

42. Ansys Mechanical. Structural FEA Analysis Software. URL: https://www.ansys.com/products/structures/ansys-mechanical#tab1-2 (Accessed 06.05.2022).

43. TDYN SEAFEM Seakeeping - Consultancy for design in engineering. URL: https://www.compassis.com/en/software/seakeeping-tdyn-seafem/ (Accessed 22.11.2023).

44. OrcaFlex – dynamic analysis software for offshore marine systems. URL: https://www.orcina.com/orcaflex/ (Accessed 22.11.2023).

45. Thomsen J. B., Ferri F., Kofoed J. P. Screening of Available Tools for Dynamic Mooring Analysis of Wave Energy Converters. Energies, 10(7), 2017, 853. https://doi.org/10.3390/en10070853.

46. WEC-Sim (Wave Energy Converter SIMulator) — WEC-Sim documentation. URL: https://WEC-Sim.github.io/WEC-Sim/master/index.html (Accessed 22.11.2023)

47. Wamit, Inc. - The State of the Art in Wave Interaction Analysis. URL: https://www.wamit.com/index.htm (Accessed 22.11.2023).

48. Babarit A., Delhommeau G. Theoretical and numerical aspects of the open source BEM solver NEMOH. Proceedings of the 11th European Wave and Tidal Energy Conference (EWTEC2015), Nantes, France, 2015. https://hal.science/hal-01198800.

49. Ancellin M., Dias F. Capytaine: a Python-based linear potential flow solver. Journal of Open Source Software, 4(36), 2019, 1341. https://doi.org/10.21105/joss.01341.

50. ParaView - Open-source, multi-platform data analysis and visualization application. URL: https://www.paraview.org/ (Accessed 22.11.2023).

51. Kurnia R., Ducrozet G. Computation of Second-order Wave Loads on Floating Offshore Wind Turbine Platforms In Bi-chromatic Bi-directional Waves Using Open-Source Potential Flow Solver NEMOH. 18èmes Journées de l'Hydrodynamique, 2022. https://doi.org/10.5281/zenodo.7418379

52. Kurnia R., Ducrozet G., Gilloteaux J.-C. Second Order Dfference- and Sum-Frequency Wave Loads in the Open-Source Potential Flow Solver NEMOH. In International Conference on

Offshore Mechanics and Arctic Engineering, volume 5A: Ocean Engineering, 2022. V05AT06A019. https://doi.org/10.1115/OMAE2022-79163.

53. Xie C., Choi Y., Rongère F., Clément A. H., Delhommeau G., Babarit A. Comparison of existing methods for the calculation of the infinite water depth free-surface Green function for the wave-structure interaction problem. Applied Ocean Research, 81, 2018, 150–163. https://doi.org/10.1016/j.apor.2018.10.007.

54. Frayssé V., Giraud L., Gratton S., Langou J. Algorithm 842: A set of GMRES routines for real and complex arithmetics on high performance computers. ACM Transactions on Mathematical Software, 31(2), 2005, 228–238. https://doi.org/10.1145/1067967.1067970.

55. Delhommeau G. Les problèmes de diffraction-radiation et de résistance des vagues : étude théorique et résolution numérique par la méthode des singularités. PhD Thesis, École Nationale Supérieure de Mécanique de Nantes, 1987.

56. DelhommeaubG. Amélioration des codes de calcul de diffraction-raditation au premier ordre. 2émes Journées de l'Hydrodynamique, 69-88, 1989.

57. Description — Nemoh latest documentation. URL: https://lheea.gitlab.io/Nemoh/description.html (Accessed 22.11.2023).

58. GitHub - BEMRosetta/BEMRosetta: Hydrodynamic coefficients viewer and converter for Boundary Element Method solver formats. URL: https://github.com/BEMRosetta/BEMRosetta (Accessed 04.12.2023).

59. Penalba M., Kelly T., Ringwood J. V. Using NEMOH for modelling wave energy converters: a comparative study with WAMIT. Proceedings of 12th European Wave and Tidal Energy Conference, 28th Aug-1st Sep, 2017.

60. Kurnia R., Ducrozet G. NEMOH: Open-source boundary element solver for computation of first- and second-order hydrodynamics loads in frequency domain. Computer Physics Communications, 292, 2023, 108885. https://doi.org/10.1016/j.cpc.2023.108885.

61. Parisella G., Gourlay T. Comparison of open-source code Nemoh with Wamit for cargo ship motions in shallow water. Report number: 2016-23 CMST, Curtin University, 2016, p. 39.

62. Anderson E. Application of the open source code Nemoh for modelling of added mass and damping in ship motion simulations. Master Thesis. Royal Institute of Technology, p. 36.

63. Sheng W., Tapoglou E., Ma X., Taylor C. J., Dorrell R. M., Parsons D. R., Aggidis G. Hydrodynamic studies of floating structures: Comparison of wave-structure interaction modelling. Ocean Engineering, 249, 2022, 110878. https://doi.org/10.1016/j.oceaneng.2022.110878.

64. GitHub - YingyiLiu/HAMS: An open-source computer program for the analysis of wave diffraction and radiation of three-dimensional floating or submerged structures. URL: https://github.com/YingyiLiu/HAMS (Accessed 23.11.2023).

65. Liu Y. Introduction of the Open-Source Boundary Element Method Solver HAMS to the Ocean Renewable Energy Community. Proceedings of the European Wave and Tidal Energy Conference, Plymouth, UK, Sep. 5–9, 2021.

66. Liu Y. HAMS: A Frequency-Domain Preprocessor for Wave-Structure Interactions — Theory, Development, and Application. Journal of Marine Science and Engineering, 7(3), 2019, 81. https://doi.org/10.3390/jmse7030081.

67. Liang H., Wu H., Noblesse F. Validation of a global approximation for wave diffraction-radiation in deep water. Applied Ocean Research, 74, 2018, 80–86. https://doi.org/10.1016/j.apor.2018.02.025.

68. Liu Y., Yoshida S., Hu C., Sueyoshi M., Sun L., Gao J., Cong P., He G.B. A reliable open-source package for performance evaluation of floating renewable energy systems in coastal and offshore regions. Energy Conversion and Management, 174, 2018, 516–536. https://doi.org/10.1016/j.enconman.2018.08.012.

69. Capytaine: a Python-based distribution of Nemoh – capytaine 2.0 documentation. URL: https://ancell.in/capytaine/latest/ (Accessed 22.11.2023).

70. Ansys SpaceClaim. 3D CAD Modeling Software. URL: https://www.ansys.com/products/3d-design/ansys-spaceclaim (Accessed 22.11.2023).

71. Makarov O., Harada T. Parametric sensitivity of crane ship numerical model with experimental verification in a wave basin. Ocean Engineering, 286(1), 2023, 115554. https://doi.org/10.1016/j.oceaneng.2023.115554.

72. Moments of inertia and Geometry orientation on import. Issue #952. WEC-Sim/WEC-Sim GitHub. URL: https://github.com/WEC-Sim/WEC-Sim/issues/952 (Accessed 01.12.2023).

73. Release v6.0. WEC-Sim/WEC-Sim GitHub. URL: https://github.com/WEC-Sim/WEC-Sim/releases/tag/v6.0 (Accessed 01.12.2023).

74. Chung J.-S., Bernitsas M. M., Hydrodynamic memory effect on stability, bifurcation, and chaos of two-point mooring systems. Journal of Ship Research, 41(1), 1997, 26–44. https://doi.org/10.5957/jsr.1997.41.1.26.

75. Lawson M., Yu Y.-H., Nelessen A., Ruehl K., Michelen C. Implementing Nonlinear Buoyancy and Excitation Forces in the WEC-Sim Wave Energy Converter Modeling Tool. Proceedings of 33rd International Conference on Ocean, Offshore and Arctic Engineering, San Francisco, California, 8-13 June 2014, 2014. https://doi.org/10.1115/OMAE2014-24445.

76. Wheeler J. D. Method for calculating forces produced by irregular waves. Journal of Petroleum Technology, 22(3), 1970, 359–367. https://doi.org/10.2118/2712-PA.

77. Choi H., Lee S. J., Jo H. J., Lee G. N., Jung K. H. Comparison Study of Experiments and Predictions of Wave Kinematics for Rogue Wave. Brodogradnja : An International Journal of Naval

Architecture and Ocean Engineering for Research and Development, 69(1), 2018, 15-38. https://doi.org/10.21278/brod69102.

78. Méhauté B. An Introduction to Hydrodynamics and Water Waves, 1976, p. 323. https://doi.org/10.1007/978-3-642-85567-2.

79. Lamb H. Hydrodynamics. 6th Edition. Dover Publications, 1945, p. 768.

80. Newman J. Wave-drift damping of floating bodies. Journal of Fluid Mechanics, 249, 1993, 241-259. https://doi.org/10.1017/S0022112093001168.

81. Wichers J. E. W. Slowly oscillating mooring forces in single point mooring systems. Proceedings of the 2nd International Conference on Behaviour of Offshore Structures (BOSS'79), London, England, 1, 1979, 357-363, 27.

82. Kapsenberg G. A Comparison of Numerical Simulations and Model Experiments on Parametric Roll in Irregular Seas. Journal of Marine Science and Engineering, 8(7), 2020, 474. https://doi.org/10.3390/jmse8070474.

83. Tober H. Evaluation of drag estimation methods for ship hulls. Degree project in engineering physics. KTH Royal institute of technology. Stockholm, Sweden, 2020.

84. ITTC – Recommended Procedures and Guidelines (2014) 7.5-03-02-03 Practical Guidelines for Ship CFD Applications.

85. ITTC – Recommended Procedures and Guidelines (2014) 7.5-03-02-04 Practical Guidelines for Ship CFD Applications.

86. Niklas K., Pruszko H. Full-scale CFD simulations for the determination of ship resistance as a rational, alternative method to towing tank experiments. Ocean Engineering, 190, 2019, 106435. https://doi.org/10.1016/j.oceaneng.2019.106435.

87. ANSYS, Inc: Ansys Fluent. Fluid Simulation Software, https://www.ansys.com/products/fluids/ansys-fluent (Accessed 30.11.2023).

88. Menter F. R. Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications. AIAA Journal, 32(8), 1994, 1598-1605. https://doi.org/10.2514/3.12149.

89. Hirt C. W., Nichols B.D. Volume of fluid (VOF) method for the dynamics of free boundaries. / C.W. Hirt,B.D. Nichols. Journal of computational physics, 39(1), 1981, 201-225. https://doi.org/10.1016/0021-9991(81)90145-5.

90. Makarov O., Yase H., Harada T. Implementation of interactive control of a crane ship model in MATLAB/Simulink environment. Robomech Journal, 10, 2023, 19. https://doi.org/10.1186/s40648-023-00259-7.

91. Simulink Desktop Real-Time – MATLAB. URL: https://www.mathworks.com/products/simulink-desktop-real-time.html (Accessed 02.12.2023).

92. Model contact between two geometries – MATLAB. URL: https://www.mathworks.com/help/sm/ref/spatialcontactforce.html (Accessed 30.11.2023).

93. Makarov O., Harada T., Yase H. Implementation of the Crane Vessel Interactive Control Using the Constraint-Based Flexible Cable with a Variable Length. In: Okada, M. (eds) Advances in Mechanism and Machine Science. IFToMM WC 2023. Mechanisms and Machine Science. Springer, Cham, 149, 2024. https://doi.org/10.1007/978-3-031-45709-8_61.

94. Hakamata R., Endo M., Sugahara Yu., Takeda Yu. Dynamic modeling and parameter identification of an elastic rod for analyzing fly-fishing. Proceedings of the 5th Jc-IFToMM International Symposium (2022) & The 28th Jc-IFToMM Symposium on Theory of Mechanism and Machines, 16th July 2022, 5, 44-51. https://doi.org/10.57272/jciftomm.5.0_44.

95. Ramli L., Mohamed Z., Jafar H. I. A neural network-based input shaping for swing suppression of an overhead crane under payload hoisting and mass variations. 107, 2018, 484–501. https://doi.org/10.1016/j.ymssp.2018.01.029.

96. Qian Y., Hu D., Chen Y., Fang Y., Hu Y. Adaptive neural network-based tracking control of underactuated offshore ship-to-ship crane systems subject to unknown wave motions disturbances. IEEE Transactions On Systems, Man, and Cybernetics: Systems, 52(6), 2022, 3626-3637. https://doi.org/10.1109/TSMC.2021.3071546.

97. Kim G.-H., Pham P.-T., Ngo Q. H., Nhuyen Q. C. Neural network-based robust anti-sway control of an industrial crane subjected to hoisting dynamics and uncertain hydrodynamic forces. International Journal of Control, Automation and Systems, 19(5), 2021, 1953-1961. https://doi.org/10.1007/s12555-020-0333-9.

98. Reinforcement Learning Agents - MATLAB & Simulink. URL: https://www.mathworks.com/help/reinforcement-learning/ug/create-agents-for-reinforcement-learning.html (Accessed 06.12.2023).

99. Twin-delayed deep deterministic (TD3) policy gradient reinforcement learning agent – MATLAB. URL: https://www.mathworks.com/help/reinforcement-learning/ref/rl.agent.rltd3agent.html (Accessed 06.12.2023).

100. Deterministic actor with a continuous action space for reinforcement learning agents – MATLAB. URL: https://www.mathworks.com/help/reinforcement-learning/ref/rl.function.rlcontinuousdeterministicactor.html (Accessed 06.12.2023).

101. Errors when using WEC-Sim GUI and Run From Simulink features. Issue #1184, WEC-Sim/WEC-Sim GitHub. URL: https://github.com/WEC-Sim/WEC-Sim/issues/1184 (Accessed 29.01.2024).

102. Haarnoja T., Zhou A., Hartikainen K., Tucker G., Ha S., Tan J., Kumar V., Zhu H., Gupta A., Abbeel P., Levine S. Soft Actor-Critic Algorithms and Applications, arXiv preprint, submitted December 13, 2018. URL: https://arxiv.org/abs/1812.05905 (Accessed 06.12.2023).

103. Soft Actor-Critic (SAC) Agents - MATLAB & Simulink. URL: https://www.mathworks.com/help/reinforcement-learning/ug/sac-agents.html (Accessed 06.12.2023).

104. Makarov O., Harada T., Yase H. AI-Positioning (target following) of suspended load 0.16kg by RL-agent in MATLAB [Video], 2024. ResearchGate. https://doi.org/10.13140/RG.2.2.18473.77928.

105. Makarov O., Harada T., Yase H. AI-Positioning (target following) of suspended load 0.7kg by RL-agent in MATLAB [Video], 2024. ResearchGate. https://doi.org/10.13140/RG.2.2.21829.22244.

106. Makarov O., Harada T., Yase H. AI-Positioning (target following) of suspended load 1.5kg by RL-agent in MATLAB [Video], 2024. ResearchGate. https://doi.org/10.13140/RG.2.2.10924.03205.

107. Makarov O., Harada T., Yase H. AI-Positioning (target following) of suspended load 0.16kg by RL-agent in MATLAB [Video], 2024. ResearchGate. https://doi.org/10.13140/RG.2.2.24345.80488.

Appendix A. Capytaine Python script

```
# Importing necessary libraries
import logging
import pathlib
from pathlib import Path

import numpy as np

import meshmagick.hydrostatics as hs
import meshmagick.mesh as mm

import capytaine as cpt
from capytaine.io.xarray import separate_complex_values
from capytaine.io.legacy import write_dataset_as_tecplot_files

from scipy.linalg import block_diag
import time
from py_console import console

time_start = time.time()
console.success('Solution start on', time.ctime(time_start), severe=True)
# Set up logging
logging.basicConfig(level=logging.INFO,
                format="%(levelname)s:\t%(message)s")

# 1. INITIAL DATA
console.info('\n1. INITIAL DATA INFO:\n', showTime=False)
rho_water = 1000.    # Water density in kg/m3
g = 9.8067          # Gravity acceleration
depth = 0.16        # Water depth
draft = 0.055       # Draft of Crane ship
mass  = 21.4        # Manual input of mass
cg = (0, 0, 0.0151675117015839)
omega_range = np.linspace(0.1, 12.5663706, 100) # Frequency vector
wave_directions = np.linspace(0, 2*np.pi, 9)   # Wave headings vector
hydrocalc = 0
'''
   Boolean operator indicates the input method for Inertia (I) and Hydrostatic stiffness (HS) of body.
   If hydrocalc= 0
   If hydrocalc= 1 the I and HS matrices are calculated using Meshmagic library
   If hydrocalc= 2 the manual input of I and HS matrices is required
```

```
'''

geometry_path = Path(pathlib.Path.home(), 'Documents', 'MATLAB', 'WEC-Sim',
                'Projects', 'CS', 'geometry', 'February')# Location of geometry folder
results_path = Path(pathlib.Path.home(), 'Documents', 'MATLAB', 'WEC-Sim',
                'Projects', 'CS',  'hydroData', 'February') # Location of results folder

f_range = omega_range / (2 * np.pi)
T_range = 1 / f_range

print('Gravity:\t', g, ' m/s^2\n',
    'Water density:\t', rho_water, ' kg/m^3\n',
    'Water depth:\t', depth, ' m\n',
    'Crane ship draft:\t', draft, ' m\n',
    'Wave parameters:\n',
    '\tAngular frequencies:\t', omega_range, ' rad/s\n',
    '\tOrdinary frequencies:\t', f_range, ' Hz\n',
    '\tWave periods:\t\t\t', T_range, ' seconds\n',
    '\tWave directions:\t\t', wave_directions, ' radians\n',
    '\t\t\t\t\t\tOR\t', wave_directions * 180 / np.pi, ' degrees\n',
    'Output path: ', str(results_path), sep='')
print('_____\n')


def body_create(path, name, draft, mass, cg, format="nemoh"):

   mesh = cpt.load_mesh(path, file_format=format, name=name)
   body = cpt.FloatingBody(mesh=mesh)  # Read mesh file



   min_z = min(body.mesh.vertices[:, 2])  # Minimum Z coordinate of mesh
   if format=="stl":
      body.translate_z(-min_z)  # Translation of mesh origin from COG to Ship hull's bottom
      body.translate_z(-draft)  # Translation of mesh origin to specified draft
      body.center_of_mass = [0, 0, -draft - min_z]  # Center of Gravity coordinates
   else:
      body.center_of_mass = cg # Center of Gravity coordinates

   body.mass = mass
   body.add_all_rigid_body_dofs()
   body.rotation_center = body.center_of_mass
```

```
    inertia = body.compute_rigid_body_inertia()      # Compute 6 x 6 mass-inertia matrix of a body with 6 rigid dofs
    massMatrix = block_diag(mass, mass, mass, inertia)              # Create Mass block diagram

    print('Body "', body.name, '" Center of Gravity: ', body.center_of_mass, sep='')
    print('Body "', body.name, '" mass: ', mass, sep='')
    print('Body "', body.name, '" inertia:\n', inertia, sep='')

    hydrostatics = body.compute_hydrostatics()
    body.keep_immersed_part()
    return body, hydrostatics, inertia, massMatrix

# 2. BODY PARAMETERS
console.info('2. BODY INFO\n', showTime=False)
tmp = Path(geometry_path, 'CS_8mm_03.02.dat')
ship, hydrostatics, I, M = body_create(tmp, "Crane ship", draft, mass, cg)

# The computation of the RAO requires the values of the inertia matrix and the hydrostatic stiffness matrix.
if hydrocalc == 0:  # Capytaine lib to compute the hydrostatic stiffness matrix
    print('Calculation of Hydrostatics: AUTOMATICALLY (Capytaine)')

    # Save Hydrostatic data to KH.dat and Hydrostatics.dat files

    kHS = hydrostatics['hydrostatic_stiffness']

    tmp = Path(results_path, 'KH.dat')
    np.savetxt(tmp, kHS, fmt='%.13e')  # Write hydrostatic stiffness to KH.dat file (NEMOH format)

    vo = hydrostatics['disp_volume']
    cb = hydrostatics['center_of_buoyancy']
    cg = ship.center_of_mass

    tmp = Path(results_path, 'Hydrostatics.dat')
    f = open(tmp, 'w')
    for j in [0, 1, 2]:
        line = f'XF = {cb[j]:7.5f} - XG = {cg[j]:7.5f} \n'
        f.write(line)
    line = f'Displacement = {vo:E}'
    f.write(line)
    f.close()
```

```
    ship.show()

elif hydrocalc == 1:  # Meshmagick to compute the hydrostatic stiffness matrix
    print('Calculation of Hydrostatics: AUTOMATICALLY (Meshmagic)')

    # Create a hydrostatics body in Meshmagick
    hsd = hs.compute_hydrostatics(mm.Mesh(ship.mesh.vertices, ship.mesh.faces),
                                  cog=cg,
                                  rho_water=rho_water,
                                  grav=g)
    print(hsd)
    m = hsd['disp_mass']  # Read mass of ship from Meshmagic calculation
    I = np.array([[hsd['Ixx'], -1 * hsd['Ixy'], -1 * hsd['Ixz']],  # Read Inertia from Meshmagic calculation
                  [-1 * hsd['Ixy'], hsd['Iyy'], -1 * hsd['Iyz']],
                  [-1 * hsd['Ixz'], -1 * hsd['Iyz'], hsd['Izz']]])

    vo = hsd['disp_volume']  # Read displaced volume from Meshmagic calculation
    cb = hsd['buoyancy_center']  # Read coordinates (array) of Center of Buoyancy from Meshmagic calculation

    M = block_diag(m, m, m, I)  # Create Mass block diagram
    kHS = block_diag(0, 0, hsd['stiffness_matrix'], 0)  # Hydrostatics - create Hydrostatic stiffness block diagram
    ship.mass = ship.add_dofs_labels_to_matrix(M)  # Convert M block diagram to matrix
    ship.hydrostatic_stiffness = ship.add_dofs_labels_to_matrix(kHS)  # Convert kHS block diagram to matrix

    # Save Hydrostatic data to KH.dat and Hydrostatics.dat files
    tmp = Path(results_path, 'KH.dat')
    np.savetxt(tmp, kHS, fmt='%.13e')  # Write hydrostatic stiffness to KH.dat file (NEMOH format)

    tmp = Path(results_path, 'Hydrostatics.dat')
    f = open(tmp, 'w')
    for j in [0, 1, 2]:
        line = f'XF = {cb[j]:7.5f} - XG = {cg[j]:7.5f} \n'
        f.write(line)
    line = f'Displacement = {vo:E}'
    f.write(line)
    f.close()

    print('Body "', ship.name, '" mass: ', m, sep='')
    print('Body "', ship.name, '" inertia:\n', I, sep='')
elif hydrocalc == 2:
    print('Calculation of Hydrostatics: MANUALLY')
```

```
    # Manual definition of Mass, Inertia and Hydrostatic Stiffness
    ship.mass = ship.add_dofs_labels_to_matrix(
        [[0, 0, 0, 0, 0, 0],
         [0, 0, 0, 0, 0, 0],
         [0, 0, 0, 0, 0, 0],
         [0, 0, 0, 0, 0, 0],
         [0, 0, 0, 0, 0, 0],
         [0, 0, 0, 0, 0, 0]]
    )

    ship.hydrostatic_stiffness = ship.add_dofs_labels_to_matrix(
        [[0, 0, 0, 0, 0, 0],
         [0, 0, 0, 0, 0, 0],
         [0, 0, 0, 0, 0, 0],
         [0, 0, 0, 0, 0, 0],
         [0, 0, 0, 0, 0, 0],
         [0, 0, 0, 0, 0, 0]]
    )

    print('Body "', ship.name, '" MASS matrix: ', sep='')
    print(ship.mass, '\n')
else:
    exit()
print('_____\n')

# 3. SOLUTION
problems = [cpt.RadiationProblem(free_surface=0,
                                 sea_bottom=-depth,
                                 rho=rho_water,
                                 body=ship,
                                 g=g,
                                 radiating_dof=dof,
                                 omega=w)
            for dof in ship.dofs for w in omega_range]
problems += [cpt.DiffractionProblem(free_surface=0,
                                    sea_bottom=-depth,
                                    rho=rho_water,
                                    body=ship,
                                    g=g,
                                    omega=w,
                                    wave_direction=heading)
```

```
        for w in omega_range for heading in wave_directions]

if len(problems) == 1:
    print(len(problems), 'problem was defined')
else:
    print(len(problems), ' problems were defined including:\n\t',
        len(ship.dofs) * len(omega_range), ' radiation problems (',
        len(ship.dofs), ' DOFs and ', len(omega_range), ' wave frequencies)\n\t',
        len(wave_directions) * len(omega_range), ' diffraction problems (',
        len(wave_directions), ' wave directions and ', len(omega_range), ' wave frequencies)', sep='')
    problems_sort = sorted(problems)

time.sleep(1)
solver = cpt.BEMSolver(green_function=cpt.XieDelhommeau(),
#                   engine=cpt.BasicMatrixEngine(linear_solver='direct'))  # Direct solver
                  engine=cpt.BasicMatrixEngine(linear_solver='lu_decomposition'))  # LU-decomposition solver
#                   engine=cpt.BasicMatrixEngine(linear_solver='gmres'))  # Iterative solver

results = [solver.solve(problem, keep_details=True) for problem in problems_sort]  # Create array of results

dataset = cpt.assemble_dataset(results,
                        hydrostatics=True,
                        mesh=True,
                        wavelength=True,
                        wavenumber=True
                        )  # Create dataset with defined parameters

console.success('Problems have been solved successfully\n')

# Write output database in Capytaine format to file with conversion of complex numbers
tmp = Path(results_path, 'ship_Capytaine.nc')
separate_complex_values(dataset).to_netcdf(tmp,
                                encoding={'radiating_dof': {'dtype': 'U'},
                                          'influenced_dof': {'dtype': 'U'}})
print('Results were saved to NetCDF format as:', tmp, '\n')

write_dataset_as_tecplot_files(results_path, dataset)
print('Results were saved as Tecplot files "RadiationCoefficients.tec" and "ExcitationForce.tec" to', results_path,
    '\n')

time_end = time.time()
```

```
time_elapsed = time_end - time_start
console.success('Solution end on', time.ctime(time_end), severe=True)
console.info('Elapsed time:', time.strftime("%D:%H:%M:%S", time.gmtime(time_elapsed)), 'or', time_elapsed, 'seconds',
        showTime=False)
```

Appendix B. Simscape code for a custom cable block

```
component Tensile_Cable
% Tensile Cable
% The block represents the cable that transfers tension forces only.

  inputs
      v_inp = {0, 'm/s'};           % Input velocity:right
      position_inp = { 0, 'm' };    % Input position:right
      v_ctrl = {0, 'm/s'};          % Controled velocity:right
  end

  outputs
      force_tot_out = { 0, 'N' }; % Total force:left
      force_spr_out = { 0, 'N' }; % Spring force:left
      force_dmp_out = { 0, 'N' }; % Damping force:left

      x_out = { 0, 'm' }; % Cable free length:left
      delta_l_out = { 0, 'm' }; % Cable increment:left
      l_out = { 0, 'm' }; % Cable full length:left
  end

  annotations
      [g] : ExternalAccess=none;
  end

  parameters
      k = { 10000, 'N/m' };     % Cable axial stiffness
      D = { 10, 'N*s/m' };      % Damping coefficient
      length = { 0.5, 'm' };    % Initial cable length
      m = { 0.16, 'kg' };       % Suspended mass [kg]
      g = { 9.8067, 'm/(s^2)'}  % Gravity acceleration
  end

  variables (Access = private)
      x = {value = { 0, 'm' },priority = priority.high}; % Cable elongation
      kk = {value = k,priority = priority.high};        % Variable stiffness
      DD = {value = D,priority = priority.high};        % Variable damping
      delta_l = { 0, 'm' }; % Length increment
      tensioned = 0;        % Tension flag
  end

  intermediates
      power_dissipated = D*v_inp^2;
      force_spr = -kk*x;                    % Spring force equation
      force_dmp = -DD*v_inp;                % Damping force equation
      force_tot = force_spr + force_dmp;    % Total force equation
      l = length + delta_l;                 % Instantaneous cable length
  end

  function setup
      if k <= 0
          error('Cable stiffness must be greater than 0');
      end
      if D < 0
          error('Cable damping cannot be negative');
      end
      if length < 0
          error('Initial cable length must be greater than 0');
      end
  end

  equations (Initial = true)
      force_spr_out == m * g;                           % Initial force due to gravity
```

```
    x == (position_inp + l + force_spr_out/k)              % Initial cable
  end

  equations (Initial = false) % regular equations
    x == (position_inp + l) % Cable free length

    if x > 0
      kk == 0;
      DD == 0;
      tensioned == 0;
      delta_l.der == v_ctrl
    else
      kk == k;
      DD == D;
      tensioned == 1;
      delta_l.der == v_ctrl
    end

    force_tot_out == force_tot;   % Total force
    force_spr_out == force_spr;   % Spring force
    force_dmp_out == force_dmp;   % Damping force
    x_out == x;                   % Free length output
    delta_l_out == delta_l;
    l_out == l;
  end
end
```

Appendix C. MATLAB code for definition and training of RL Agent

Initial parameters:

```
%% Initial parameters
simu.gravity = 9.8067;
simu.rho = 1000;
g = simu.gravity;                          % Gravity acceleration [m1/s^2]

cable.length = 0.5-0.0045-0.019;                       % Initial cable length [m]
cable.fullLength = cable.length+0.0045+0.038/2;
cable.radius = 0.0004;                      % Cable radius [m]
cable.mat_stiffness = 1.13e11;

sload.mass = 0.16;                    % Mass of the suspened load [kg]
sload.omega = 0;                     % Initial inclination
sload.omegaRad = deg2rad(sload.omega);
sload.maxTarget = [0.3 0.3];         % Maximum random distance from initial position

sload.initpos = [0.279-cable.fullLength*sin(sload.omegaRad) 0.752-
cable.fullLength*cos(sload.omegaRad)]; % Initial pos in Global coordinates
sload.bounds = [-0.11 0.525 0.05 0.50];

randXY = [rand*(sload.bounds(2)-sload.bounds(1))+sload.bounds(1) rand*(sload.bounds(4)-
sload.bounds(3))+sload.bounds(3)];
sload.target = [randXY(1) randXY(2)];

cable.area = pi * cable.radius^2*0.7;             % Cross-section area
cable.k = cable.area*cable.mat_stiffness/cable.length;  % k = A*E/l [N/m]
cable.k = 100000;         % Cable stiffness [N/m]
cable.d = 100;             % Cable damping [N*s/m]
cable.initial_x = sload.mass*g/cable.k; % Initial cable elongation [m]

body.mass = 21.4;
body.volume = 4.39626E-2;
body.centerBuoyancy = [0.07982,-0.02673];
body.centerGravity = [0,0.01517];
```

Reset function:

```
function in = ResetFunctionBase(in)

% Set random cable length [0.1 0.7]m, suspended load mass [0.1 2.0]kg,
% and initial inclination [-20 20] deg
cable.length = rand*0.6+0.1;
sload.mass = rand*1.9+0.1;
sload.omega = rand*40-20;

sload.omegaRad = deg2rad(sload.omega);
cable.fullLength = cable.length+0.0045+0.038/2;
sload.initpos = [0.279-cable.fullLength*sin(sload.omegaRad) 0.752-
cable.fullLength*cos(sload.omegaRad)]; % Initial pos in Global coordinates
sload.maxTarget = [0.3 0.3];


% Set random position for suspended load in range sload.bounds = [-0.11 0.525 0.05
0.50];
randXY = [rand*(0.525+0.110)-0.110 rand*(0.50-0.05)+0.05];
sload.target = [randXY(1) randXY(2)];

%{
randXY = [rand*sload.maxTarget(1)*2-sload.maxTarget(1) rand*sload.maxTarget(2)*2-
sload.maxTarget(2)];
```

```
sload.bounds = [sload.initpos(1)-sload.maxTarget(1) sload.initpos(1)+sload.maxTarget(1)
sload.initpos(2)-sload.maxTarget(2)-0.2 sload.initpos(2)+sload.maxTarget(2)-0.2];
sload.target = [sload.initpos(1)+randXY(1) sload.initpos(2)+randXY(2)-0.2];
%}


% Assign abovementioned values to model parameters
in = setVariable(in,'cable.length',cable.length);
in = setVariable(in,'sload.mass',sload.mass);
in = setVariable(in,'sload.omega',sload.omega);
in = setVariable(in,'sload.target',sload.target);

end
```

SAC agent configuration:

```
%% Model specifications
Init_params;

RL.mdl = "Ship_2D_var5_RL";         % Model name
RL.agentblk = RL.mdl + "/RL Agent";   % Agent block
RL.numObs = 15;                     % Number of observations
RL.numAct = 2;                      % Number of actions
RL.Ts = 0.01;                       % Time step
RL.Tf = 10;                         % Simulation time
rng(0)                            % Fix the random generator seed for reproducibility.
RL.hiddenLayerSize = 128;
RL.VelocityMax = 0.15;
RL.TorqueMax = 15;
RL.distMax = 0.6;
RL.distPrize = 0.02;
%% Create observation & action specifications & env.
RL.obsInfo = rlNumericSpec( ...
    [RL.numObs 1],...
    DataType="double");
RL.obsInfo.Name = "observations";
RL.obsInfo.Description = "Observations from ship";

RL.actInfo = rlNumericSpec(...
    [RL.numAct 1],...
    'LowerLimit',[-RL.TorqueMax; -RL.VelocityMax],...
    'UpperLimit',[RL.TorqueMax; RL.VelocityMax],...
    DataType="double");
RL.actInfo.Name = 'force;speed';

RL.env = rlSimulinkEnv(RL.mdl, RL.agentblk, RL.obsInfo, RL.actInfo);
RL.env.UseFastRestart = 'off';
RL.env.ResetFcn = @ResetFunctionBase;
env = RL.env;

%% Define training options
% https://www.mathworks.com/help/reinforcement-
learning/ref/rl.option.rltrainingoptions.html
RL.maxEpisodes = 1000;
RL.maxSteps = floor(RL.Tf/RL.Ts);
RL.rewardValue = 2.3*RL.maxSteps;
RL.windowLength = 50;

RL.trainOpts = rlTrainingOptions(...
    MaxEpisodes=RL.maxEpisodes, ...
    MaxStepsPerEpisode=RL.maxSteps, ...
    ScoreAveragingWindowLength=RL.windowLength,...
    Verbose=false, ...                                % Option to display training
progress at the command line
```

```
    Plots="training-progress",...                    % "training-progress" or "none"
    StopTrainingCriteria="AverageReward",...
    StopTrainingValue=RL.rewardValue,...
    SaveAgentCriteria="AverageReward",...
    SaveAgentValue=RL.rewardValue*1,...
    SaveAgentDirectory = "savedAgents/New2DRLProblem");

RL.criticOptions = rlOptimizerOptions( ...
    Optimizer="adam", ...
    LearnRate=1e-4,...
    GradientThreshold=1, ...
    L2RegularizationFactor=1e-4);

RL.actorOptions = rlOptimizerOptions( ...
    Optimizer="adam", ...
    LearnRate=1e-4,...
    GradientThreshold=1, ...
    L2RegularizationFactor=1e-4);

RL.agentOptions = rlSACAgentOptions(...
    SampleTime=RL.Ts,...
    DiscountFactor = 0.98,...
    TargetSmoothFactor = 1e-3,... % Smoothing factor for target critic updates,
specified as a positive scalar less than or equal to 1.
    TargetUpdateFrequency= 2,... % Number of steps between target critic updates,
specified as a positive integer.
    ExperienceBufferLength = 1e6,...
    MiniBatchSize = 128, ...
    NumWarmStartSteps = 128, ... % Number of actions to take before updating actor and
critics, specified as a positive integer. By default, the NumWarmStartSteps value is
equal to the MiniBatchSize value.
    NumGradientStepsPerUpdate = 1, ... %Number of gradient steps to take when updating
actor and critics, specified as a positive integer.
    NumStepsToLookAhead = 5, ... % Number of future rewards used to estimate the value
of the policy, specified as a positive integer. Specifically, if NumStepsToLookAhead is
equal to N, the target value of the policy at a given step is calculated adding the
rewards for the following N steps and the discounted estimated value of the state that
caused the N-th reward. This target is also called N-step return.
    ResetExperienceBufferBeforeTraining = false); %Option for clearing the experience
buffer before training, specified as a logical value.

%The fact that the discount factor is 0.99,that is very close to 1, means
% that the agent keeps more into account long term rewards
% (a discount factor closer to 0 would instead place a heavier discount on
% future rewards, therefore encouraging short term ones)

RL.agentOptions.CriticOptimizerOptions(1) = RL.criticOptions;
RL.agentOptions.CriticOptimizerOptions(2) = RL.criticOptions;
RL.agentOptions.ActorOptimizerOptions = RL.actorOptions;

%% Create critic network
% Define path for the state input
RL.statePath = [
    featureInputLayer(prod(RL.obsInfo.Dimension), Name="NetObsInLayer")
    fullyConnectedLayer(RL.hiddenLayerSize,Name="sPathOut")];

% Define path for the action input
RL.actionPath = [
    featureInputLayer(prod(RL.actInfo.Dimension), Name="NetActInLayer")
    fullyConnectedLayer(RL.hiddenLayerSize,Name="aPathOut",BiasLearnRateFactor=0)];

% Define path for the critic output (value)
RL.commonPathCritic = [
    concatenationLayer(1,2,Name="concat")
```

```
    reluLayer
    fullyConnectedLayer(RL.hiddenLayerSize)
    reluLayer
    fullyConnectedLayer(RL.hiddenLayerSize)
    reluLayer
    fullyConnectedLayer(1,Name="CriticOutput")];

% Create layerGraph object and add layers
RL.criticNetwork = layerGraph(RL.statePath);
RL.criticNetwork = addLayers(RL.criticNetwork,RL.actionPath);
RL.criticNetwork = addLayers(RL.criticNetwork,RL.commonPathCritic);

% Connect paths
RL.criticNetwork = connectLayers(RL.criticNetwork,"sPathOut","concat/in1");
RL.criticNetwork = connectLayers(RL.criticNetwork,"aPathOut","concat/in2");

% Convert Critic network to dlnetwork object and create the critic representation using
the specified deep neural network and options

RL.criticNetwork1 = dlnetwork(RL.criticNetwork);
RL.criticNetwork2 = dlnetwork(RL.criticNetwork);
summary(RL.criticNetwork1)
summary(RL.criticNetwork2)
RL.critic1 = rlQValueFunction(RL.criticNetwork1,RL.obsInfo,RL.actInfo, ...
    ObservationInputNames="obsInLyr");
RL.critic2 = rlQValueFunction(RL.criticNetwork2,RL.obsInfo,RL.actInfo, ...
    ObservationInputNames="obsInLyr");

plot(RL.criticNetwork1)

%% Create actor network

RL.commonPathActor = [
    featureInputLayer(prod(RL.obsInfo.Dimension), Name="NetObsInLayer")
    fullyConnectedLayer(RL.hiddenLayerSize)
    reluLayer
    fullyConnectedLayer(RL.hiddenLayerSize)
    reluLayer(Name="comPathOut")
    ];

RL.meanPath = [
    fullyConnectedLayer(RL.hiddenLayerSize,Name="meanFC")
    reluLayer
    fullyConnectedLayer(prod(RL.actInfo.Dimension), Name="meanOut")
    ];

RL.stdPath = [
    fullyConnectedLayer(prod(RL.actInfo.Dimension), Name="stdFC")
    reluLayer
    softplusLayer(Name="stdOut")
    ];

% Connect the layers.
RL.actorNetwork = layerGraph(RL.commonPathActor);
RL.actorNetwork = addLayers(RL.actorNetwork,RL.meanPath);
RL.actorNetwork = addLayers(RL.actorNetwork,RL.stdPath);
RL.actorNetwork = connectLayers(RL.actorNetwork,"comPathOut","meanFC/in");
RL.actorNetwork = connectLayers(RL.actorNetwork,"comPathOut","stdFC/in");

% Convert to dlnetwork and display the number of weights.
RL.actorNetwork = dlnetwork(RL.actorNetwork);
summary(RL.actorNetwork)
plot(RL.actorNetwork)
```

```
% Create the actor in a similar manner to the critic
% https://www.mathworks.com/help/reinforcement-
learning/ref/rl.function.rlcontinuousdeterministicActor.html
RL.actor = rlContinuousGaussianActor(RL.actorNetwork, RL.obsInfo, RL.actInfo, ...
    ObservationInputNames="NetObsInLayer", ...
    ActionMeanOutputNames="meanOut", ...`
    ActionStandardDeviationOutputNames="stdOut");


%% Construct agent
 RL.agentCustom = rlSACAgent(RL.actor,[RL.critic1 RL.critic2],RL.agentOptions);

%% Training
RL.doTraining = false;
%RL.doTraining = true;

%{
evl = rlEvaluator( ...
    NumEpisodes=5, ...
    EvaluationFrequency=50);
%}
if RL.doTraining
    % Train the agent.
    RL.trainingStats = train(RL.agentCustom, RL.env, RL.trainOpts);
end
```

List of research activities

Publications

Articles:

1. Makarov O., Harada T., 2023. Parametric sensitivity of crane ship numerical model with experimental verification in a wave basin. Ocean Engineering, Vol. 286(1), 115554. https://doi.org/10.1016/j.oceaneng.2023.115554 (Elsevier, IF 4.372 (Q1), CiteScore 6.5 (Q1), First author)

2. Makarov O., Yase H., Harada T., 2023. Implementation of interactive control of a crane ship model in MATLAB/Simulink environment. Robomech Journal, 10, 19. https://doi.org/10.1186/s40648-023-00259-7 (Springer, CiteScore 2.9 (Q2), First author)

Proceedings:

1. Makarov O., Harada T., 2022. Effective approach to modeling the hydrodynamic response of a multibody system with suspended load in the time domain. Proceedings of Jc-IFToMM International Symposium, Vol.5, 95–102. https://doi.org/10.57272/jciftomm.5.0_95. (J-STAGE, First author).

2. Makarov O., Harada T., Yase H. Simulation of cargo lifting and landing operations with real-time interactive control of a numerical model of a floating crane. Proceedings of the 9th International Conference on Manufacturing, Machine Design and Tribology. – *Abstract* (First author).

3. Makarov O., Harada T., Yase H., 2024. Implementation of the Crane Vessel Interactive Control Using the Constraint-Based Flexible Cable with a Variable Length. In: Okada, M. (eds) Advances in Mechanism and Machine Science. IFToMM WC 2023. Mechanisms and Machine Science, Vol. 149. Springer, Cham. https://doi.org/10.1007/978-3-031-45709-8_61. (Springer, CiteScore 0.19 (Q4), First author).

4. Makarov O., Harada T., 2024. Reinforcement learning for AI-positioning of suspended load by a crane vessel. Proceedings of the 10th International Conference on Electrical Engineering, Control and Robotics (EECR), Guangzhou, China. – *Accepted for publication* (IEEE Xplore, First author).

Conferences

International:

1. Makarov O., Harada T.: Effective approach to modeling the hydrodynamic response of a multibody system with suspended load in the time domain. The 5th Jc-IFToMM International

Symposium and the 28th Jc-IFToMM Symposium on Theory of Mechanism and Machines (Jc-IFToMM 2022), Kyoto, Japan. 16th July 2022. (Oral presentation, peer review).

2. Makarov O., Harada T., Yase H.: Simulation of cargo lifting and landing operations with real-time interactive control of a numerical model of a floating crane. The 9th International Conference on Manufacturing, Machine Design and Tribology (ICMDT 2023), Jeju, Korea. March 8th–11th 2023. (Oral presentation, peer review);

3. Makarov O., Harada T., Yase H.: Implementation of the Crane Vessel Interactive Control Using the Constraint-Based Flexible Cable with a Variable Length. IFToMM World Congress 2023, Tokyo, Japan. 5th–10th November 2023. (Oral presentation, peer review).

4. Makarov O., Harada T.: Reinforcement learning for AI-positioning of suspended load by a crane vessel. The 10th International Conference on Electrical Engineering, Control and Robotics (EECR), Guangzhou, China. 29th–31st March 2024. (Oral presentation, peer review).

Domestic:

1. Makarov O., Harada T. Real-time numerical modeling of crane ship dynamics and lifting operations. Masters in Graduate School of Science and Engineering (総合理工マスターズ 2023), Osaka, Japan. 28th February 2023. (Oral presentation, no peer review).

2. Makarov O., Yase H., Harada T. Real-Time Interactive Simulation of Offshore Support Vessel with Suspended Load Under Action of Sea Waves (波浪作用下における洋上起重機船クレーン作業のリアルタイムインタラクション). Robotic Symposia 2023 (第 28 回ロボティクスシンポジア), Shirahama, Japan. 15th–16th March 2023. (Oral presentation, peer review).

3. Makarov O., Harada T. Real-Time Numerical Modeling of Crane Ship Dynamics with Experimental Verification in a Wave Basin. The Autumn Conference of The Japan Society of Naval Architects and Ocean Engineers (JASNAOE Autumn meeting), Kobe, Japan. 17th–18th November 2022. (Poster presentation, peer review).

4. Makarov O., Harada T. Application of reinforcement learning for AI anti-sway control and positioning of suspended load on a crane ship. The Autumn Conference of The Japan Society of Naval Architects and Ocean Engineers (JASNAOE Autumn meeting), Nagasaki, Japan. 27th–28th November 2023. (Poster presentation, peer review).