

# Trajectory Optimization for Autonomous Berthing of a Twin-Propeller Twin-Rudder Ship

Changyu Lee<sup>1</sup> and Jinwhan Kim<sup>2</sup>

<sup>1</sup>Graduate Student, Department of Mechanical Engineering, KAIST, Daejeon, Korea

<sup>2</sup>Professor, Department of Mechanical Engineering, KAIST, Daejeon, Korea

**KEYWORDS:** Two-point boundary value problem, Trajectory planning, Autonomous berthing, Twin-propeller twin-rudder ship, Model predictive control

**ABSTRACT:** *Autonomous berthing is a crucial technology for autonomous ships, requiring optimal trajectory planning to prevent collisions and minimize time and control efforts. This paper presents a two-phase, two-point boundary value problem (TPBVP) strategy for creating an optimal berthing trajectory for a twin-propeller, twin-rudder ship with autonomous berthing capabilities. The process is divided into two phases: the approach and the terminal. Tunnel thruster use is limited during the approach but fully employed during the terminal phase. This strategy permits concurrent optimization of the total trajectory duration, individual phase trajectories, and phase transition time. The efficacy of the proposed method is validated through two simulations. The first explores a scenario with phase transition, and the second generates a trajectory relying solely on the approach phase. The results affirm our algorithm's effectiveness in deciding transition necessity, identifying optimal transition timing, and optimizing the trajectory accordingly. The proposed two-phase TPBVP approach holds significant implications for advancements in autonomous ship navigation, enhancing safety and efficiency in berthing operations.*

## 1. Introduction

Recently, as interest in autonomous ships grows, autonomous berthing has garnered attention as a crucial technology for enabling fully autonomous ship navigation. Conventionally, larger ships depend on tugboat assistance for berthing instead of conducting the process independently (Quan et al., 2019). However, in pursuit of the ultimate goal of achieving full autonomy in ship navigation, automation of the berthing process is essential.

Effective berthing requires plotting a path that moves the ship from its current position to the intended berthing point. This procedure calls for trajectory planning over simple path planning, to ensure smoother movement, increased efficiency, and adept handling of constraints. Previous research proposed berthing algorithms for ships equipped with azimuth thrusters using optimal control (Martinsen et al., 2019, 2022), as well as path optimization methods for single-rudder, single-propeller ships with two side thrusters while considering real port spatial constraints (Miyachi et al., 2022). To tackle the universal challenge of finding a globally optimal solution in optimization-based methods, graph search (Ødven et al., 2022) and warm-started semi-

online trajectory planners (Rachman et al., 2022) have been utilized. Most recently, experimental verification of the miiliampere ship's docking algorithm - an urban ferry with two azimuth thrusters - was conducted using MPC-based trajectory optimization and trajectory tracking controllers (Bitar et al., 2020; Bitar et al., 2021; Martinsen et al., 2020).

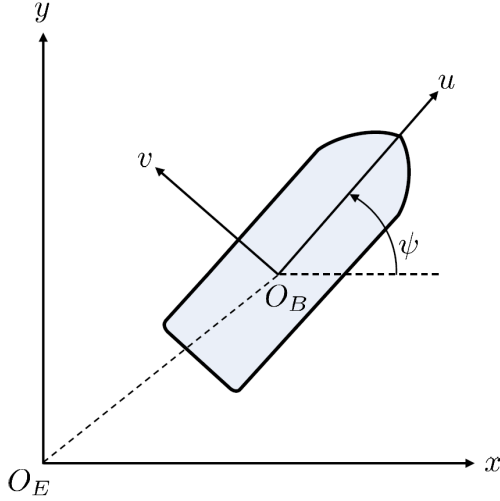
In this study, we propose a berthing trajectory generation algorithm for twin-propeller, twin-rudder ships with self-berthing capabilities. It is challenging to anticipate effective force when a pair of propellers and rudders are rotated in reverse (Lindegard and Fossen, 2003; Skjetne et al., 2004). Additionally, tunnel thrusters only show effectiveness at low speeds (Fossen, 2011). Addressing these motion characteristics, we adopt a two-phase, two-point boundary value problem approach, segmenting the process into an approaching phase—where the ship maneuvers close to the target berthing position—and a terminal phase, continuing until the ship aligns with its final position. In the approaching phase, the use of the tunnel thruster is limited, whereas in the terminal phase, it is actively deployed. We conducted numerical simulations to validate the proposed trajectory generation algorithm's efficacy, highlighting its successful application in

Received 16 May 2023, revised 12 June 2023, accepted 14 June 2023

Corresponding author Jinwhan Kim: +82-42-350-1519, [jinwhan@kaist.ac.kr](mailto:jinwhan@kaist.ac.kr)

© 2023, The Korean Society of Ocean Engineers

This is an open access article distributed under the terms of the creative commons attribution non-commercial license (<http://creativecommons.org/licenses/by-nc/4.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.



**Fig. 1** The coordinate system of a ship

autonomous berthing scenarios.

The structure of this paper is as follows: Section 2 introduces the necessary preliminaries, providing the groundwork for understanding the subsequent methodologies. The formulation of our proposed trajectory optimization scheme is presented in detail in Section 3. In Section 4, we provide a comprehensive review of the simulation results, demonstrating the efficacy of our approach. Finally, we draw conclusions from our findings in Section 5, encapsulating the key points of the study and their implications for autonomous ship berthing.

## 2. Preliminary

### 2.1 Ship Dynamic Model

The equations of motion of a ship are composed of both kinematic and kinetic equations, represented in the earth-fixed and body-fixed coordinate systems, as illustrated in Fig. 1. The kinematic equations, which deal with the geometric properties of motion, are defined within the context of the earth-fixed frame. Conversely, the kinetic equations are modeled within the body-fixed coordinate system. The relationship between these two coordinate systems is articulated through the ship's heading  $\psi$  as follows:

$$\dot{\eta} = \mathcal{J}(\psi)\nu,$$

$$\mathcal{J}(\psi) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

where  $\eta = [x, y, \psi]^\top$  and  $\nu = [u, v, r]^\top$  are the position and velocity vectors, respectively, and  $\mathcal{J}(\psi)$  is the rotation matrix that transforms the velocities from the body-fixed coordinate to the earth-fixed coordinate.

The kinetic equations represent the relationship between the forces acting on the vehicle and its motion within the body-fixed frame.

These equations can be articulated using Newton's second law, as follows:

$$M\dot{\nu} + C(\nu)\nu + D(\nu)\nu = \tau \quad (2)$$

In this equation,  $M$  represents the inertia matrix,  $C(\nu)$  stands for the Coriolis and centripetal matrix,  $D(\nu)$  signifies the damping matrix, and  $\tau$  is the vector of control force. The matrices  $M$  and  $C(\nu)$  can be depicted as the sum of the rigid body and added mass components, as outlined below:

$$M = M_{RB} + M_A$$

$$M_{RB} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & mx_g \\ 0 & mx_g & I_z \end{bmatrix}$$

$$M_A = \begin{bmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & -Y_r \\ 0 & -N_v & -N_r \end{bmatrix} \quad (3)$$

$$C(\nu) = C_{RB}(\nu) + C_A(\nu),$$

$$C_{RB}(\nu) = \begin{bmatrix} 0 & 0 & -m(x_g r + v) \\ 0 & 0 & mu \\ m(x_g r + v) & -mu & 0 \end{bmatrix}$$

$$C_A(\nu) = \begin{bmatrix} 0 & 0 & Y_v \nu + Y_r r \\ 0 & 0 & -X_u u \\ -Y_v \nu - Y_r r & X_u u & 0 \end{bmatrix}$$

Additionally,  $D(\nu)$  can be depicted by a combination of the nonlinear and linear damping matrices, as follows:

$$D(\nu) = D_l(\nu) + D_n(\nu),$$

$$D_l(\nu) = \begin{bmatrix} X_u & 0 & 0 \\ 0 & Y_v & Y_r \\ 0 & N_v & N_r \end{bmatrix} \quad (4)$$

$$D_n(\nu) = \begin{bmatrix} X_{|u|}|u| & 0 & 0 \\ 0 & Y_{|v|}|v| + Y_{|r|}|r| & Y_{|v|}|v| + Y_{|r|}|r| \\ 0 & N_{|v|}|v| + N_{|r|}|r| & N_{|v|}|v| + N_{|r|}|r| \end{bmatrix}$$

### 2.2 Actuator Model

In this study, as illustrated in Fig. 2, we develop an algorithm for a ship equipped with twin propellers and twin rudders at the stern, in addition to a bow tunnel thruster. The forces generated are as follows:  $T_1$  represents the force produced by the port stern propeller,  $T_2$  corresponds to the force originating from the stern starboard propeller, while  $T_3$  designates the bow tunnel propeller forces. Additionally, the rudder angles at the port and starboard sides are denoted by  $\delta_1$  and  $\delta_2$ , respectively. Given these, the control force can be formulated with the

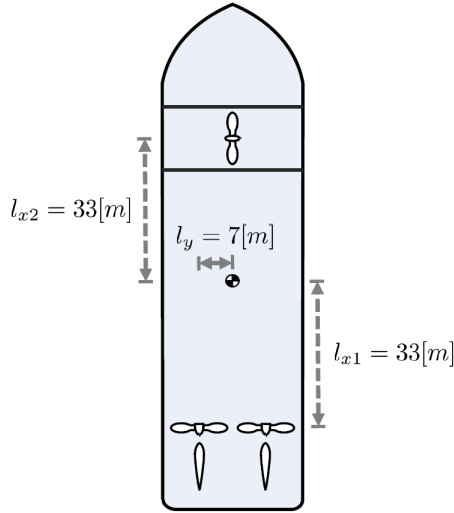


Fig. 2 The control configuration of a ship

control input  $[T_1, T_2, \delta_1, \delta_2, T_3]^T$ , as detailed below:

$$\tau_c = g(u) = \begin{bmatrix} T_1 + T_2 - D_1 - D_2 \\ L_1 + L_2 + T_3 \\ (D_1 - T_1 - D_2 + T_2)l_y - (L_1 + L_2)l_{x1} + T_3l_{x2} \end{bmatrix} \quad (5)$$

In this formulation,  $D_1$  and  $D_2$  represent the drag forces, while  $L_1$  and  $L_2$  represent the lift forces for the port and starboard rudder, respectively. The terms  $l_{x1}$ ,  $l_{x2}$  and  $l_y$  refer to the distances from the center of gravity to each propeller. The force generated by the propeller can be defined by the rotational speed  $n_i$ , and the thruster coefficient  $K_T$  with advance ratio  $J_i$ , as follows:

$$T_i = K_T(J_i)\rho n_i |n_i D^4, \quad i = 1, 2 \quad (6)$$

where  $\rho$  denotes the density of water and  $D$  stands for the propeller's diameter. The tunnel thruster is also modeled using the rotational speed  $n_3$ , the propeller's diameter  $D_3$ , and the thruster coefficient  $K_{T3}$ . However, it is important to note that the tunnel thruster can only generate sufficient force when the ship moves at low speeds. Consequently, we employ the tunnel thruster at lower speeds and model it with a constant thruster coefficient, independent of the advance ratio, as follows:

$$T_3 = K_{T3}\rho n_3 |n_3 D_3^4 \quad (7)$$

The force exerted by a ship's rudder at low speed is notably influenced by the direction of the propeller's rotation. When the propeller rotates forward, ample lift is generated through propeller force and rudder angle interplay. However, generating effective lift is challenging when the propeller rotates in reverse. Consequently, the lift and drag forces produced by a pair of propellers and rudders can be formulated as follows, according to (Lindgaard, 2003):

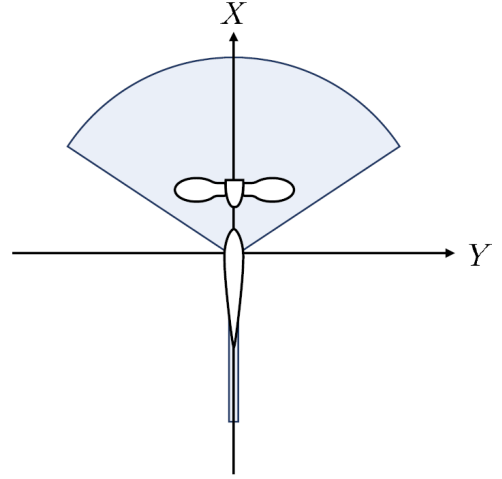


Fig. 3 The feasible force domain of single-propeller, single-rudder

$$L_i = \begin{cases} T_i k_{L\delta} \delta_i, & T_i \geq 0 \\ 0, & T_i < 0 \end{cases}, \quad i = 1, 2,$$

$$D_i = \begin{cases} T_i (k_{D\delta_1} |\delta_i| + k_{D\delta_2} \delta_i^2), & T_i \geq 0 \\ 0, & T_i < 0 \end{cases}, \quad i = 1, 2. \quad (8)$$

Fig. 3 visually illustrates the force generated by a pair of propellers and rudders. Given that the ship's propellers and rudders can operate independently, the ship is viewed as a fully-actuated system capable of controlling motion across three degrees of freedom. Nevertheless, when considering the propulsion properties of the propellers and rudders, as demonstrated in Eq. (8), the system exhibits specific characteristics of being partially under-actuated.

### 3. Berthing Trajectory Optimization

Given the initial and target berthing states, a trajectory optimization algorithm can be devised to minimize both the berthing duration and control effort, while taking into account the motion and control characteristics represented by Eqs. (1)-(8). However, the nonlinear nature and complexity of the optimization problem present a challenge in assuring optimal solutions. To tackle this issue, we propose an optimization approach that separates the trajectory into two distinct phases, thereby effectively reducing the complexity of the problem.

#### 3.1 Two-phase Approach

The berthing process can be strategically segmented into two phases. The first, referred to as the "approaching phase," guides the ship towards a region proximate to the final berthing position. The second, termed the "terminal phase," involves the precise maneuvering of the ship to reach the final position. During the approaching phase, the ship retains sufficient speed, thus hindering the bow tunnel propeller from generating effective force. As a result, in this phase, only the stern propellers and rudders are employed as control inputs, as described below:

$$u_a = [T_1, T_2, \delta_1, \delta_2, 0]^\top \quad (9)$$

Conversely, the terminal phase occurs at relatively lower speeds, thus allowing the use of the tunnel thruster. In this phase, the ship's movements and rotations are slower, diminishing the effects of the Coriolis and nonlinear damping matrices. This reduction permits a simplified model approximation similar to those used in dynamic positioning (Skjetne et al., 2004). Accordingly, the dynamic model and control input vectors are formulated as follows:

$$M\dot{\nu} + D_1(\nu)\nu = \tau_c, \quad (10)$$

$$u_t = [T_1, T_2, \delta_1, \delta_2, T_3]^\top$$

### 3.2 Two-point Boundary Value Problem

Employing models for each phase allows for the formulation of the two-point boundary value problem with the state vector  $x = [x, y, \psi, u, v, r]^\top$ , as detailed below:

$$\min_{x(\cdot), u(\cdot)} J_T + \int_0^{T_{f1}} J_a(t) dt + J_a^{T_{f1}} + \int_{T_{f1}}^{T_{f2}} J_t(t) dt \quad (11)$$

Here  $T_{f1}$  and  $(T_{f2} - T_{f1})$  represent the time durations for each phase, respectively. Additionally,  $J_T$  denotes the time penalty cost function, which is defined as follows:

$$J_T(t) = \rho_t (T_{f1} + T_{f2}) \quad (12)$$

Here  $\rho_t$  is the weighting scalar.  $J_a(t)$ ,  $J_a^{T_{f1}}$  represent the cost functions for the approaching, which are defined as follows:

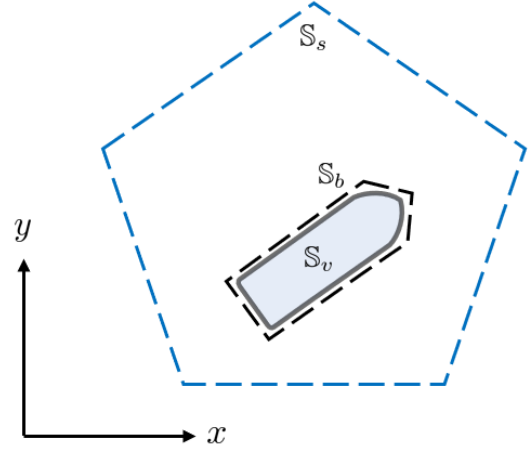
$$J_a(t) = \nu(t)^\top Q_a \nu(t) + u_a(t)^\top P_a u_a(t) + \dot{u}_a(t)^\top R_a \dot{u}_a(t)$$

$$J_a^{T_{f1}} = (\eta(T_{f1}) - \eta_f)^\top Q_a^{T_{f1}} (\eta(T_{f1}) - \eta_f) \quad (13)$$

In these equations, the matrices  $Q_a$ ,  $P_a$ , and  $R_a$  denote the weight matrices that penalize the velocity, control input, and rate of change of control input, respectively.  $Q_a^{T_{f1}}$  represents the weight matrix that penalizes the terminal position of the approaching phase, while  $\eta_f$  is the final position vector. Additionally,  $J_t(t)$  denotes the cost function for the terminal phase, given as follows:

$$J_t(t) = \nu(t)^\top Q_t \nu(t) + u_t(t)^\top P_t u_t(t) + \dot{u}_t(t)^\top R_t \dot{u}_t(t) \quad (14)$$

In this context, the matrices  $Q_t$ ,  $P_t$ , and  $R_t$  represent the weight matrices that penalize the velocity, control input, and rate of change of control input, respectively. The optimization problem formulated above also encompasses several constraints, which are as follows:



**Fig. 4** The illustration of collision avoidance constraints. (Ship  $S_v$  with a safety boundary  $S_b$ , as well as state constraints  $S_s$ )

$$\dot{x}(t) = f_a(x(t), u_a(t)), \forall t \in [0, T_{f1}]$$

$$\dot{x}(t) = f_t(x(t), u_t(t)), \forall t \in [T_{f1}, T_{f2}]$$

$$x(0) = x_0$$

$$x(T_{f2}) = x_f \quad (15)$$

$$|u(t)| \leq \overline{u(t)}, \forall t \in [0, T_{f2}]$$

$$S_b \subseteq S_s$$

Here  $f_a$  and  $f_t$  represent the dynamic models of the approaching and terminal phases, respectively, while  $x_0$  and  $x_f$  denote the initial and target berthing states.  $\overline{(\cdot)}$  denotes the maximum value. The final constraints represent the collision avoidance condition. Here,  $S_b$  and  $S_s$  indicate the safety boundary of the ship and state constraints, as shown in Fig. 4.

To address the nonlinear optimization problem formulated in this manner, we utilized the Interior Point Optimizer (IPOPT) (Wächter and Biegler, 2006), implemented in the CasADI framework (Andersson et al., 2019), within the MATLAB environment.

## 4. Simulation Results

Simulations were executed within the MATLAB environment to evaluate the effectiveness of the proposed trajectory optimization algorithm. We incorporated the port geometry of Jangsaengpo Port in Ulsan, Korea, and devised two scenarios featuring distinct initial and final positions. The state constraints were formulated convexly, considering the berthing position, the port layout, and the ship's navigable area. It is worth noting that the specific configuration of these constraints can be altered based on the available space in the intended target port. The parameters of the supply ship's model used for simulation are derived from (Skjetne et al., 2004), with the

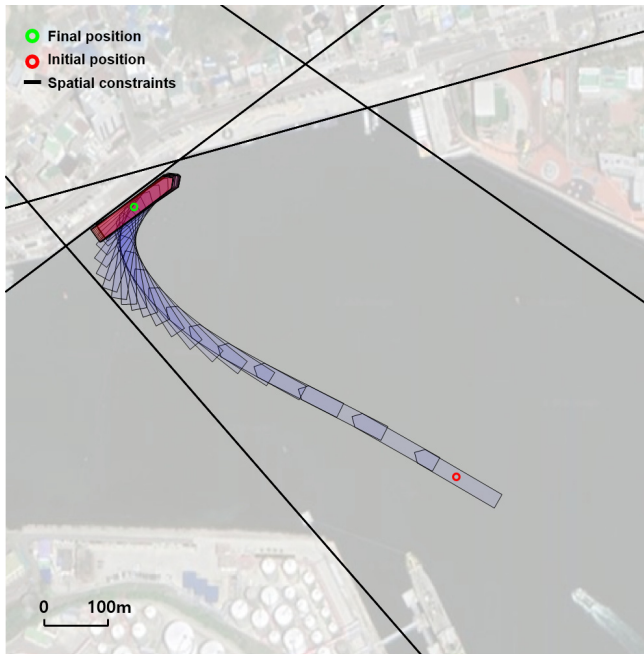
remaining parameters detailed in Table 1. The parameters specific to our proposed algorithm can be found in Table 2. The results of the first simulation are visually represented in Figs. 5-7.

**Table 1** The model parameters of the ship

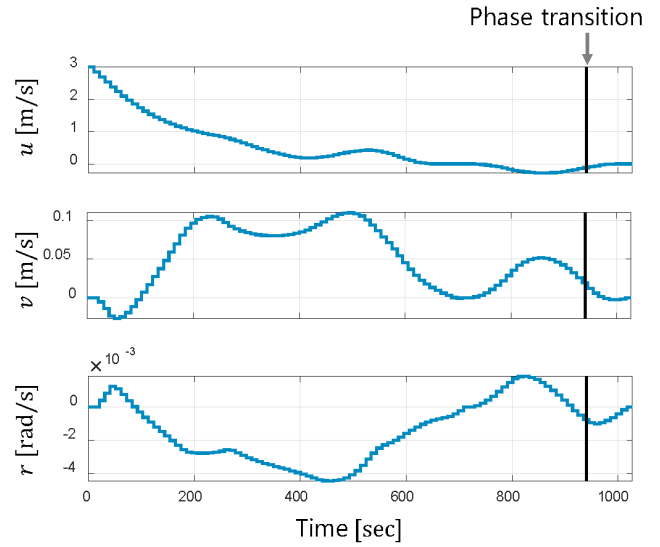
Parameter	Value
$k_{D\delta_1}$	0.093
$k_{D\delta_2}$	0.727
$k_{L\delta}$	0.920
$l_{x_1}$ (m)	-33
$l_{x_2}$ (m)	33
$l_y$ (m)	7

**Table 2** Parameters of the trajectory optimization problem

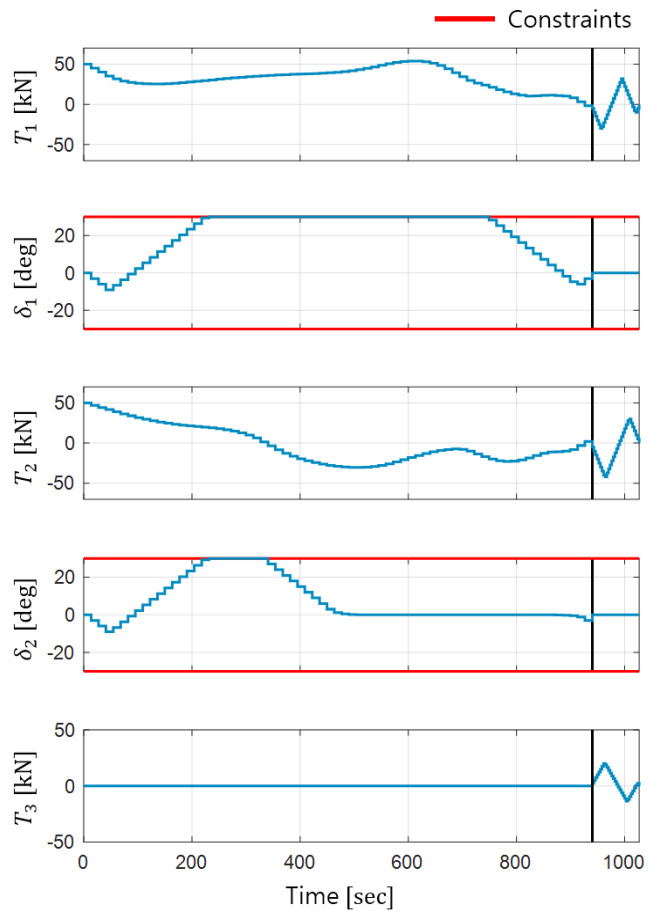
Parameter	Value
$\rho_t$	30
$Q_a$	diag ([0.0, 0.0, 1.0E6])
$P_a$	diag ([0.1, 0.1, 1.0, 1.0, 0.1])
$R_a$	diag ([1.0, 1.0, 1.0, 1.0, 1.0])
$Q_a^{T,\rho}$	diag ([1.0, 1.0, 5.0E4])
$Q_t$	diag ([1.0E2, 0.0, 0.0])
$P_t$	diag ([0.0, 0.0, 1.0, 1.0, 0.0])
$R_t$	diag ([0.01, 0.01, 1.0, 1.0, 0.01])
$\bar{T}_1, \bar{T}_2$ (kN)	200
$\bar{R}_1, \bar{R}_2$ (deg)	30
$\bar{T}_3$ (kN)	100



**Fig. 5** The result of the first scenario



**Fig. 6** Time trajectories of velocities (first scenario)



**Fig. 7** Time trajectories of control inputs (first scenario)

Fig. 5 illustrates the time trajectory for a port side berthing scenario. In this figure, the blue ship trajectory corresponds to the approaching phase, while the red ship trajectory represents the terminal phase. Fig. 6 illustrates the variation in velocities over the course of the berthing process, which lasts approximately 1026 seconds, with the approaching phase accounting for 940 seconds of this duration. This

highlights a distinct phase switch. Fig. 7 shows the alterations in control input over time. These results corroborate that the proposed algorithm effectively considers the characteristic of not utilizing a rudder when the propeller is in reverse rotation.

In the second scenario, the objective is to berth a ship on the starboard side. The results of this simulation are depicted in Figs. 8-10. Fig. 8 displays the time trajectories for each phase. Figs. 9 and 10, on the other hand, portray the velocities and control input fluctuations over time, respectively. Contrary to the first scenario, the second scenario does not require a phase transition. This result arises from the optimization process determining that a sole emphasis on the approaching phase, without a phase transition, is more efficient for this scenario.

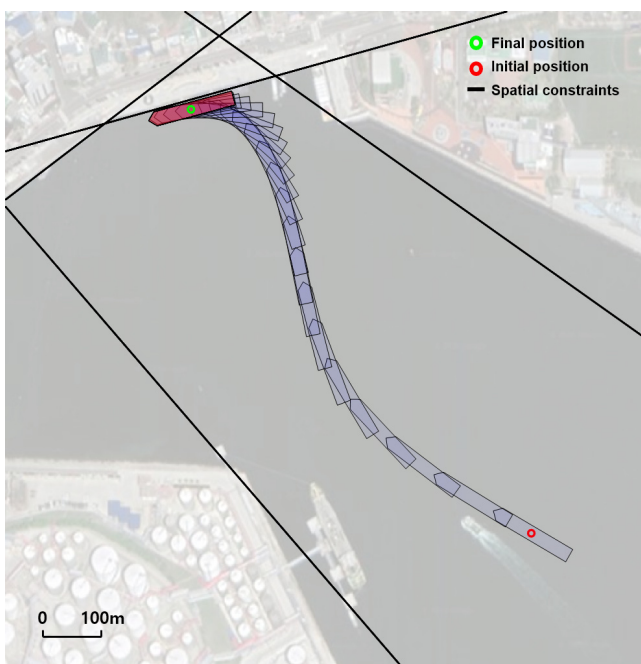


Fig. 8 The result of the second scenario

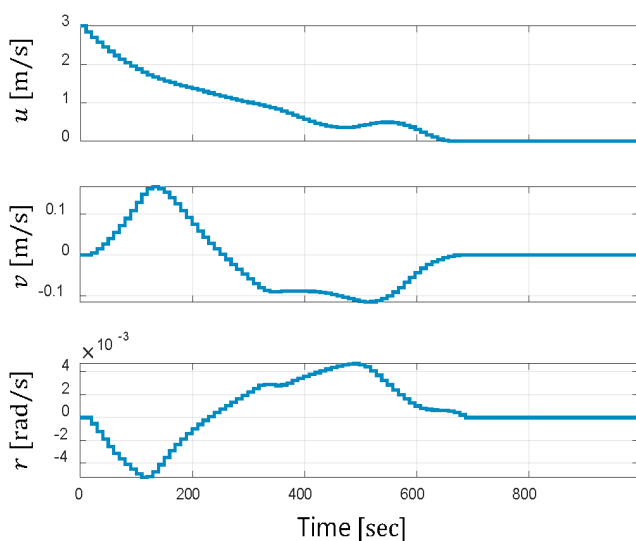


Fig. 9 Time trajectories of velocities (second scenario)

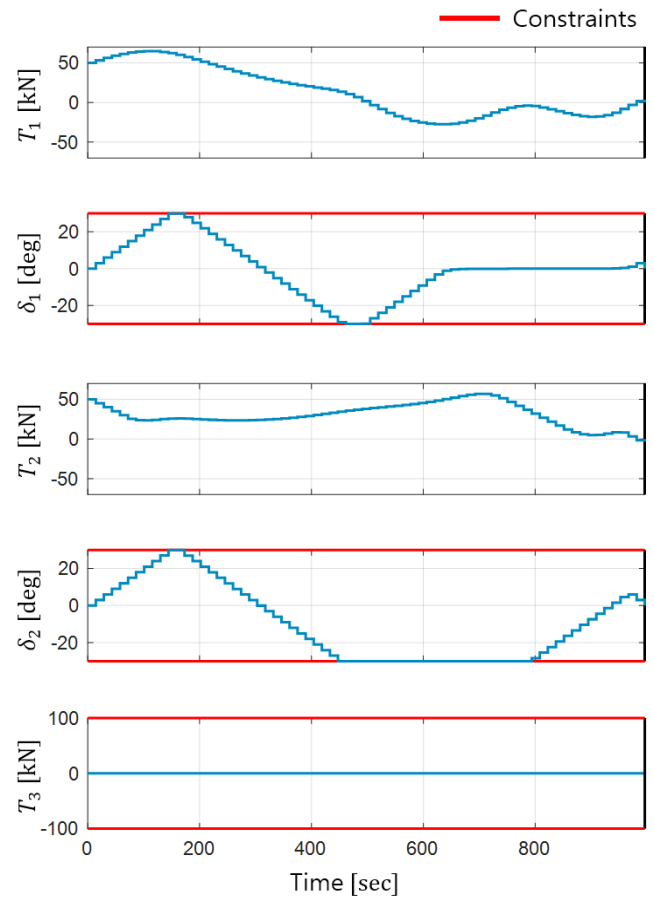


Fig. 10 Time trajectories of control inputs (second scenario)

## 5. Conclusions

In this study, we have proposed a two-phase berthing trajectory optimization algorithm for the autonomous berthing of a twin-propeller, twin-rudder ship. The berthing process was segregated into two distinct phases, taking into account the ship's dynamic characteristics and actuator properties. We formulated models for each phase and generated dynamically feasible trajectories using a two-phase, two-point boundary value problem. To assess the practicality and effectiveness of our proposed algorithm, we created two scenarios grounded in real-world port geometries and conducted comprehensive numerical simulations.

## Conflict of Interest

Jinwhan Kim serves as an editorial board member of the Journal of Ocean Engineering and Technology, but he had no role in the decision to publish this article. No potential conflict of interest relevant to this article was reported.

## Funding

This research was supported by a grant from the National R&D Project "Development of an electric-powered car ferry and a

roll-on/roll-off power supply system" funded by the Ministry of Oceans and Fisheries, Korea (Grant No. PMS4420).

## References

- Andersson, J. A. E., Gillis, J., Horn, G., Rawlings, J. B., & Diehl, M. (2019). CasADi: A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, *11*(1), Article 1. <https://doi.org/10.1007/s12532-018-0139-4>
- Bitar, G., Eriksen, B.-O. H., Lekkas, A. M., & Breivik, M. (2021). Three-Phase Automatic Crossing for a Passenger Ferry With Field Trials. *2021 European Control Conference (ECC)*, 2271–2277. <https://doi.org/10.23919/ECC54610.2021.9655139>
- Bitar, G., Martinsen, A. B., Lekkas, A. M., & Breivik, M. (2020). Trajectory Planning and Control for Automatic Docking of ASVs with Full-Scale Experiments\*\*This work is supported by the Research Council of Norway through the project number 269116 as well as through the Centres of Excellence funding scheme with project number 223254. *IFAC-PapersOnLine*, *53*(2), 14488–14494. <https://doi.org/10.1016/j.ifacol.2020.12.1451>
- Fossen, T. I. (2011). *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons.
- Lindgaard, K.-P. (2003). *Acceleration feedback in dynamic positioning*.
- Lindgaard, K.-P., & Fossen, T. I. (2003). Fuel-efficient rudder and propeller control allocation for marine craft: Experiments with a model ship. *IEEE Transactions on Control Systems Technology*, *11*(6), 850–862. <https://doi.org/10.1109/TCST.2003.815613>
- Martinsen, A. B., Bitar, G., Lekkas, A. M., & Gros, S. (2020). Optimization-Based Automatic Docking and Berthing of ASVs Using Exteroceptive Sensors: Theory and Experiments. *IEEE Access*, *8*, 204974–204986. <https://doi.org/10.1109/ACCESS.2020.3037171>
- Martinsen, A. B., Lekkas, A. M., & Gros, S. (2019). Autonomous docking using direct optimal control. *IFAC-PapersOnLine*, *52*(21), Article 21.
- Martinsen, A. B., Lekkas, A. M., & Gros, S. (2022). Optimal Model-Based Trajectory Planning With Static Polygonal Constraints. *IEEE Transactions on Control Systems Technology*, *30*(3), Article 3. <https://doi.org/10.1109/TCST.2021.3094617>
- Miyauchi, Y., Sawada, R., Akimoto, Y., Umeda, N., & Maki, A. (2022). Optimization on planning of trajectory and control of autonomous berthing and unberthing for the realistic port geometry. *Ocean Engineering*, *245*, 110390. <https://doi.org/10.1016/j.oceaneng.2021.110390>
- Ødven, P. K., Martinsen, A. B., & Lekkas, A. M. (2022). Static and dynamic multi-obstacle avoidance and docking of ASVs using computational geometry and numerical optimal control. *IFAC-PapersOnLine*, *55*(31), Article 31. <https://doi.org/10.1016/j.ifacol.2022.10.408>
- Quan, T. D., Suh, J.-H., & Kim, Y.-B. (2019). Leader-Following Control System Design for a Towed Vessel by Tugboat. *Journal of Ocean Engineering and Technology*, *33*(5), 462–469. <https://doi.org/10.26748/KSOE.2019.075>
- Rachman, D. M., Maki, A., Miyauchi, Y., & Umeda, N. (2022). Warm-started semionline trajectory planner for ship's automatic docking (berthing). *Ocean Engineering*, *252*, 111127. <https://doi.org/10.1016/j.oceaneng.2022.111127>
- Skjetne, R., Smogeli, Ø. N., & Fossen, T. I. (2004). A Nonlinear Ship Manoeuvring Model: Identification and adaptive control with experiments for a model ship. *Modeling, Identification and Control: A Norwegian Research Bulletin*, *25*(1), 3–27. <https://doi.org/10.4173/mic.2004.1.1>
- Wächter, A., & Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, *106*(1), 25–57. <https://doi.org/10.1007/s10107-004-0559-y>

## Author ORCIDs

Author name	ORCID
Lee, Changyu	0000-0003-1964-9101
Kim, Jinwhan	0000-0001-6886-2449