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

Changyu Lee and Jinwhan Kim

1Graduate Student, Department of Mechanical Engineering, KAIST, Daejeon, Korea
2Professor, 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 (Miyauchi et al., 2022). To tackle the universal challenge of finding a globally optimal solution in optimization-based methods, graph search (Odven et al., 2022) and warm-started semi-online trajectory planners (Rachman et al., 2022) have been utilized. Most recently, experimental verification of the milliampere 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 (Lindegaard 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
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:

\[
\dot{\eta} = J(\psi)\nu,
\]

\[
J(\psi) = \begin{bmatrix}
\cos \psi & -\sin \psi & 0 \\
\sin \psi & \cos \psi & 0 \\
0 & 0 & 1
\end{bmatrix}
\] (1)

where \( \eta = [x, y, \psi]^{T} \) and \( \nu = [u, v, r]^{T} \) are the position and velocity vectors, respectively, and \( J(\psi) \) 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:

\[
M\ddot{\nu} + C(\nu)\nu + D(\nu)\nu = \tau
\] (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:

\[
M = M_{BB} + M_{A}
\]

\[
M_{BB} = \begin{bmatrix}
m & 0 & 0 \\
0 & m & m_{zz} \\
0 & m_{xy} & I_{e}
\end{bmatrix}
\]

\[
M_{A} = \begin{bmatrix}
-X_{r} & 0 & 0 \\
0 & -Y_{r} & -Y_{c} \\
0 & -N_{r} & -N_{c}
\end{bmatrix}
\] (3)

\[
C(\nu) = C_{BB}(\nu) + C_{A}(\nu),
\]

\[
C_{BB}(\nu) = \begin{bmatrix}
0 & 0 & -m(x_{e}r + v) \\
m(x_{y}r + v) & -mu & 0 \\
0 & 0 & 0
\end{bmatrix}
\]

\[
C_{A}(\nu) = \begin{bmatrix}
0 & 0 & Y_{c}r + Y_{r}l \\
0 & 0 & -X_{a}u \\
-Y_{c}l - Y_{r}r & X_{a}l & 0
\end{bmatrix}
\]

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

\[
D(\nu) = D_{l}(\nu) + D_{n}(\nu),
\]

\[
D_{l}(\nu) = \begin{bmatrix}
X_{r} & 0 & 0 \\
0 & Y_{r} & Y_{c} \\
0 & N_{r} & N_{c}
\end{bmatrix}
\] (4)

\[
D_{n}(\nu) = \begin{bmatrix}
0 & Y_{r}l + Y_{c}l & 0 \\
0 & N_{r}l + N_{c}l & 0 \\
0 & 0 & X_{a}l + N_{c}l
\end{bmatrix}
\]

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 \( \delta_{p} \) and \( \delta_{s} \), respectively. Given these, the control force can be formulated with the
control input $[ T_1, T_2, \delta_1, \delta_2, T_1, T_2]^T$, as detailed below:

$$\tau_i = g(u) = \begin{bmatrix} T_1 + T_2 - D_1 - D_2 \\ L_1 + L_2 + T_1 \\ (D_1 - T_1 - D_2 + T_1, \delta_i) - (L_1 + L_2, \delta_i) + T_1 \end{bmatrix}$$

(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 $l_1$, $l_2$, and $l_3$ 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 $J$, as follows:

$$T_i = K_T J \rho n_i D_i$$

(6)

where $\rho$ denotes the density of water and $D_i$ stands for the propeller's diameter. The tunnel thruster is also modeled using the rotational speed $n_t$, the propeller’s diameter $D_t$, and the thruster coefficient $K_T$.

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:

$$T_t = K_T J \rho n_t D_t$$

(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 (Lindegaard, 2003):
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:

\[ \dot{u}_c = [T_1, T_2, \delta_1, \delta_2, 0]^\top \quad (9) \]

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 \( \mathbf{x} = \left[ x, y, \psi, u, v, r \right]^\top \), as detailed below:

\[
\begin{aligned}
\min_{\mathbf{x} \in \mathbb{R}, \mathbf{u} \in \mathbb{R}} & \quad J_T + \int_0^{T_1} J_{\mathbf{x}}(t) dt + J_{\mathbf{u}} + \int_{T_1}^{T_2} J_{\mathbf{u}}(t) dt \\
\text{subject to} & \quad \mathbf{x}(0) = \mathbf{x}_0 \\
& \quad \mathbf{x}(T_2) = \mathbf{x}_f \\
& \quad \mathbf{u}(t) \leq \mathbf{u}_f(t), \quad \forall t \in [0, T_2] \\
& \quad \mathbf{s}_k \subseteq \mathbf{s}_s \\
\end{aligned}
\quad (11)
\]

Here, \( J_T \) and \( J_{\mathbf{x}} - J_{\mathbf{u}} \) represent the time durations for each phase, respectively. Additionally, \( J_T \) denotes the time penalty cost function, which is defined as follows:

\[ J_T = \rho_1(T_{T_1} + T_{T_2}) \quad (12) \]

In these equations, the matrices \( Q_v, P_v, \) and \( R_v \) denote the weight matrices that penalize the velocity, control input, and rate of change of control input, respectively. \( Q_{\mathbf{x}}(t) \) represents the weight matrix that penalizes the terminal position of the approaching phase, while \( \mathbf{u}_p \) is the final position vector. Additionally, \( J_{\mathbf{x}}(t) \) denotes the cost function for the terminal phase, given as follows:

\[ J_{\mathbf{x}}(t) = \mathbf{v}(t)^\top Q_v \mathbf{v}(t) + \mathbf{u}_p(t)^\top P_v \mathbf{u}_p(t) + \dot{\mathbf{u}}_p(t)^\top R_v \dot{\mathbf{u}}_p(t) \quad (13) \]

In this context, the matrices \( Q_v, P_v, \) and \( R_v \) represent the weight matrices that penalize the velocity, control input, and rate of change of control input, respectively. The optimization problem formulated above encompasses several constraints, which are as follows:
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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_{h1}$</td>
<td>0.093</td>
</tr>
<tr>
<td>$k_{h2}$</td>
<td>0.727</td>
</tr>
<tr>
<td>$k_{d}$</td>
<td>0.920</td>
</tr>
<tr>
<td>$l_x$ (m)</td>
<td>-33</td>
</tr>
<tr>
<td>$l_y$ (m)</td>
<td>33</td>
</tr>
<tr>
<td>$l_z$ (m)</td>
<td>7</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rho_i$</td>
<td>30</td>
</tr>
<tr>
<td>$Q_i$</td>
<td>diag ([0.0, 0.0, 1.0E6])</td>
</tr>
<tr>
<td>$P_s$</td>
<td>diag ([0.0, 0.0, 1.0, 1.0, 0.1])</td>
</tr>
<tr>
<td>$R_t$</td>
<td>diag ([1.0, 1.0, 1.0, 1.0, 1.0])</td>
</tr>
<tr>
<td>$Q_{i,n}$</td>
<td>diag ([1.0, 1.0, 5.0E4])</td>
</tr>
<tr>
<td>$Q_l$</td>
<td>diag ([1.0E2, 0.0, 0.0])</td>
</tr>
<tr>
<td>$P_l$</td>
<td>diag ([0.0, 0.0, 1.0, 1.0, 0.0])</td>
</tr>
<tr>
<td>$R_l$</td>
<td>diag ([0.01, 0.01, 1.0, 1.0, 0.01])</td>
</tr>
<tr>
<td>$T_1, T_2$ (kN)</td>
<td>200</td>
</tr>
<tr>
<td>$R_1, R_2$ (deg)</td>
<td>30</td>
</tr>
<tr>
<td>$T_3$ (kN)</td>
<td>100</td>
</tr>
</tbody>
</table>

Fig. 5 The result of the first scenario

**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.

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


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


Author ORCIDs

<table>
<thead>
<tr>
<th>Author name</th>
<th>ORCID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lee, Changyu</td>
<td>0000-0003-1964-9101</td>
</tr>
<tr>
<td>Kim, Jinwhan</td>
<td>0000-0001-6886-2449</td>
</tr>
</tbody>
</table>