Automatic Docking for Underactuated Ships Based on Multi-Objective Nonlinear Model Predictive Control

Li, Shijie; Liu, Jialun; Negenborn, Rudy R.; Wu, Qing

DOI
10.1109/ACCESS.2020.2984812

Publication date
2020

Document Version
Final published version

Published in
IEEE Access

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 11, 2020, accepted March 29, 2020, date of publication April 1, 2020, date of current version April 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2984812

Automatic Docking for Underactuated Ships Based on Multi-Objective Nonlinear Model Predictive Control

SHIJIE LI[1], JIALUN LIU[2,3], (Member, IEEE), RUDY R. NEGENBORN[3,4], AND QING WU[1]

[1]School of Logistics Engineering, Wuhan University of Technology, Wuhan 430063, China
[2]National Engineering Research Center for Water Transport Safety, Wuhan 430063, China
[3]Intelligent Transportation Systems Research Center, Wuhan University of Technology, Wuhan 430063, China
[4]Department of Maritime and Transport Technology, Delft University of Technology, 2628 CD Delft, The Netherlands

Corresponding author: Jialun Liu (jialunliu@whut.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1601505, in part by the National Natural Science Foundation of China under Grant 51709217, in part by the Research on Intelligent Ship Testing and Verification under Grant 2018473, in part by the Natural Science Foundation of Hubei Province under Grant 2018CFB640, in part by the State Key Laboratory of Ocean Engineering (Shanghai Jiao Tong University) under Grant 1707, and in part by the Researchlab Autonomous Shipping at Delft University of Technology.

ABSTRACT Autonomous shipping refers to the ability of a ship to independently control its own actions while transporting cargo from one port to another, which places higher requirements on ship motion control methods. When a ship enters a port, it is important to ensure that the ship sails from the fairway area to the assigned position at the berth with a desirable speed and that it finally stops at the desired position. Ship docking is known as one of the most challenging tasks due to the non-linearity of low-speed ship movements and the high requirements on collision avoidance with the quayside. This paper proposes a nonlinear model predictive control (NMPC) -based approach for underactuated ships, providing optimal ship rudder angles and propeller revolution rate to automate the ship docking process. At each sampling instant, a finite horizon optimal control problem is formulated based on a nonlinear ship maneuverability model. A lexicographic multi-objective optimization strategy is proposed in the design of the NMPC controller, saving the efforts on control parameters tuning. Simulation experiments are carried out to evaluate the effectiveness of the proposed approach.

INDEX TERMS Automatic docking, motion control, nonlinear model predictive control.

I. INTRODUCTION

With the trends towards autonomous shipping, advanced ship motion control techniques are being developed to ensure that ships can independently control their own actions, especially in complicated situations. When a ship enters a port, it is important to ensure that the ship sails from the fairway area to the assigned position at the berth with a desirable speed and that it finally stops at the desired position. In practice, the ship docking procedure performed by a skilled shipmaster usually consists of three steps [1]: firstly, when the ship has sufficient acceleration to alter its course with the rudder, the course of the ship is changed to the desired berth-approach direction. Secondly, the ship decelerates, and the ship's maneuverability is also reduced due to the low speed, which makes it difficult to maneuver the ship with rudder; thirdly, the main engine is stopped at the appropriate time, and the ship moves to the berth with the remaining speed. In short, a ship docking procedure includes course changing, speed deceleration, and engine stopping.

Ship docking is known as one of the most challenging tasks [1]–[4], inappropriate ship maneuvers may lead to collision accidents with the quayside, which causes losses of lives and property. To ensure maritime safety, as well as to contribute to the development of smart ships and autonomous shipping, automatic ship docking techniques are required. When a ship is moving at a low speed, the ship maneuverability also reduces, which makes it difficult to steer the ship in a flexible way. In addition, most marine ships are under-actuated systems, only the yaw and surge movements

The associate editor coordinating the review of this manuscript and approving it for publication was Bohui Wang.

can be actuated directly, which means that there are more variables to be controlled than the number of control actuators. Unlike fully-actuated ships, the control feedback of under-actuated ships is difficult to linearize, which makes it an open and interesting problem in control. While numerous under-actuated systems have been controlled, there are few general principles. Moreover, the under-actuated nature of a ship is coupled with the nonlinear characteristics of the hydrodynamics associated with ship motion, this further increases the difficulty of designing a suitable controller for ship docking control.

To dock a ship to a berth safely, ship states including position, heading angle, and velocity need to be controlled appropriately by the rudder and the propeller. The key for a successful docking controller is how to regulate the rudder and the propeller properly. To automate this challenging process, several control methods have been proposed in recent years.

Artificial Neural Networks (ANN) have been one of the most commonly used methods because of their learning ability and mimicking actions of the human brain when performing stages of ship docking. In [1], an ANN controller is designed based on a head-up ship coordinate system to design the ANN controller, which includes the relative bearing and distance from the ship to the berth. This makes the ANN controller able to adapt to different ports without having to retain the ANN structure. An ANN auto-docking control method is proposed in [5], considering unknown ship dynamics and external disturbances, which is based on deep-rooted information and performed using the additional control method, dynamic surface control, and minimum learning parameter techniques.

Another ANN approach is proposed in [6], in which the network is trained by consistent teaching data to replicate human brain decisions during ship docking, considering wind disturbances. The docking maneuver process is divided into three types, including course changing, step deceleration and stopping, in order to prepare teaching data. Then feed-forward multi-layered ANN controllers are proposed and trained with the Lavenberg-Marquart algorithm in back propagation technique for different rudder angles and propeller revolution rates. While the results are effective, it still takes time to train ANN models.

The authors in [2] proposed a quasi real-time method to solve minimum-time approaching control for automatic docking. In [3], the automatic ship docking problem is formulated as an off-line minimum-time optimal control problem. To deal with the computational difficulty caused by the non-linearity of the problem, a covariance matrix adaption evolution strategy is used to optimize the real-valued variables.

Most research takes the surge forces and yaw moments as the control inputs, as they can directly affect ship movements. This also makes it easier to handle the non-linearity of ship dynamics. In practice, a shipmaster uses the propeller revolution rate and rudder angle as the actual control inputs. Considering compliance with practicality, this paper adopts the propeller revolution rate and rudder angles as the main control variables. In addition, most marine ships are underactuated, and equipped with propellers and rudders for surge and sway movements, without any actuators to directly control the sway movements. Therefore, this paper focus on underactuated ships.

This paper proposes to solve the auto-docking problem of underactuated ships in light of model predictive control theory. Model predictive control (MPC) is an optimization-based control strategy that could provide a flexible framework to deal with complicated and nonlinear system dynamics and system constraints. It has been extensively studied and applied to numerous industrial problems, a literature review can be found in [7]. It has also received attention for the trajectory tracking control problem of ships [8]–[15]. Firstly, it is a systematic way to deal with complicated problems with multiple inputs and outputs. Secondly, it could explicitly consider constraints on state and control variables. Moreover, its optimization-based nature could guarantee optimality or sub-optimality of generated control moves.

As a first step, the automatic docking problem is formulated as an optimal control problem. Due to the non-linearity of the ship dynamics in docking, a multi-objective nonlinear model predictive control scheme is proposed to generate optimal control inputs for the ship.

Nonlinear model predictive control (NMPC) is the extension of classic MPC with nonlinear models, constraints and objective functions. In an NMPC problem, the objective function usually consists of a weighted sum of different terms, but the selection of appropriate weight assigned to each term could be a time-consuming task. Reducing one weight on one objective and increasing another does not necessarily lead to a proportional response in the results.

Multi-objective nonlinear model predictive control (MO-NMPC) could deal with the optimization problem of multiple conflicting performance criteria over a receding horizon for constrained nonlinear systems [16]. The optimization in MO-NMPC is based on multi-objective optimization algorithms including the weighted sum [17], the goal attainment [18], utopia tracking [16], [19], and the lexicographic method [20], [21], etc.

Among these methods, the lexicographic method could explicitly consider the priorities of different optimization objectives. This is in accordance with ship docking practice, in which the shipmaster first adjusts its course and then positions. In other words, the shipmaster prefers course angle alteration in the beginning and changes its operation priority into reaching the berth location precisely. Therefore, this paper uses lexicographic MO-NMPC to design a ship docking controller, as it can explicitly take into account the priorities of different objectives.

In the final stage of the docking operation, it is important to avoid collisions against the berths or nearby ships docked in neighboring berths. This paper deals with this situation

by introducing reference points and planning of safe docking paths.

The main contributions of this paper include:

1) Most research simplified the ship model and uses surge force and yaw moment as the control variables, while in practice most ships use RPS and rudder angle as the control inputs. This paper formulates a nonlinear ship model that uses RPS and rudder angles as main control inputs, which is in compliance with ship docking practice.
2) To save the efforts on tuning the weights assigned to each term in the objective function, a lexicographic multi-objective optimization strategy is adopted in the design of the MO-NMPC controller for underactuated ships.

This paper is organized as follows: Section 2 introduces nonlinear ship dynamics and formulates the ship docking problem as an optimal control problem. Section 3 introduces the MO-NMPC controller design and solution steps of the auto-docking approach. Simulation results are presented in Section 4. Conclusions and future work are given in Section 5.

II. PRELIMINARIES AND PROBLEM FORMULATION

An NMPC problem is usually formulated as a discrete-time optimal control problem over a finite horizon constrained by nonlinear model equations. Therefore, this section first introduces the preliminaries of ship dynamics and then gives the formulation of the ship docking control as an optimal control problem.

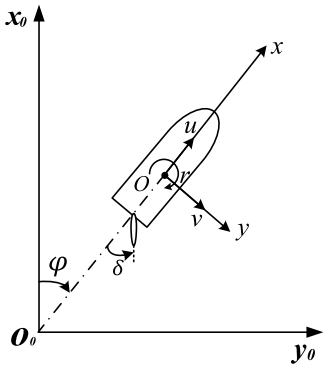


FIGURE 1. Ship coordinate system.

A. NONLINEAR SHIP DYNAMICS

Figure 1 shows the ship coordinate system used in this paper: the space-fixed coordinate system $O_0 - x_0\ y_0\ z_0$ and the moving ship-fixed coordinate system $o-xyz$, where o is taken on the midship of the ship, with x, y and z axes that point towards the ship's bow, towards the starboard and vertically downwards, respectively. Heading angle ψ is defined as the angle between the x_0 and x axes, variable δ represents the rudder angle and r represents the yaw rate. Variables u and v denote the surge and sway velocity in x and y directions, respectively.

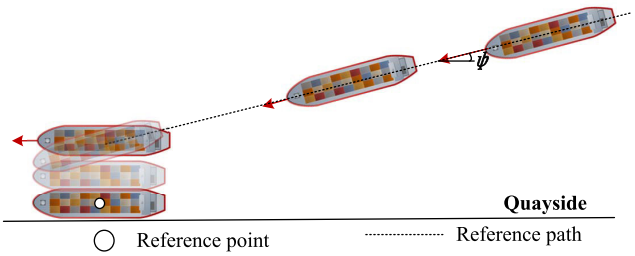


FIGURE 2. Ship docking process.

For ship maneuvering model, a 3-DOF (degree of freedom) model is used to represent the ship dynamics on the surge, sway and yaw axes. An MMG (Maneuvering Modeling Group) model is applied to represent the ship movements, in which the hydrodynamic forces and moments on the ship are divided into hull, rudder, and propeller, expressed in the following form:

$$\begin{cases} (m+m_x)\dot{u} - (m+m_y)vr - x_G m r^2 = X_H + X_P + X_R \\ (m+m_y)\dot{v} + (m+m_x)ur + x_G m\dot{r} = Y_H + Y_P + Y_R \\ (I_z + x_G^2 m + J_z)\dot{r} + x_G m(\dot{v} + ur) = N_H + N_P + N_R \end{cases} \tag{1}$$

where, subscripts H, P, R represent the hull, the propeller, and the rudder; m, m_x and m_y are ship mass, added mass in x-direction, and added mass in y-direction; I_z and J_z are moment of inertia and added moment of inertia around the z-axis, u and v are ship longitudinal and lateral speed, r is ship yaw rate around midship, and the dot notation of u, v and r represents the derivative of each parameter.

Then, the system state of (1) can be transformed into the following form [22]:

$$\begin{cases} \dot{x} = u\cos\psi - v\sin\psi \\ \dot{y} = u\sin\psi + v\cos\psi \\ \dot{\psi} = r \\ \dot{u} = \dfrac{m_v}{m_u}vr - \dfrac{f_u(\boldsymbol{v})}{m_u} + \dfrac{T_u(\cdot)}{m_u}|n|n + d_{wu} \\ \dot{v} = -\dfrac{m_u}{m_v}ur - \dfrac{f_v(\boldsymbol{v})}{m_v} + d_{wv} \\ \dot{r} = \dfrac{(m_u - m_v)}{m_r}uv - \dfrac{f_r(\boldsymbol{v})}{m_r} + \dfrac{F_r(\cdot)}{m_r}\delta + d_{wr}, \end{cases} \tag{2}$$

where

$$\begin{cases} m_u = m - X_{\dot{u}} = m + m_{xx} \\ m_v = m - Y_{\dot{v}} = m + m_{yy} \\ m_r = I_{zz} - N_{\dot{r}} = I_{zz} + J_{zz}. \end{cases} \tag{3}$$

Parameters d_{wu}, d_{wv} and d_{wr} represent the disturbances on the x, y and z axes. Variables $f_u(\boldsymbol{v}), f_v(\boldsymbol{v})$ and $f_r(\boldsymbol{v})$ represent the high-order fluid dynamics items:

$$\begin{aligned} f_u(\boldsymbol{v}) &= -(X_{|u|u}|u|u + X_{vr}vr + X_{vv}v^2 + X_{rr}r^2) \\ &= -\frac{1}{2}\rho L_{pp}TV^2(-R'_0 + X'_{vv}v'^2 + X'_{vr}v'r' + X'_{rr}r'^2 \\ &\quad + X'_{vvvv}v'^4) \end{aligned} \tag{4}$$

$$
\begin{aligned}
f_v(\boldsymbol{v}) &= -(Y_v v + Y_r r + Y_{|v|v}|v|v + Y_{|r|r}|r|r + Y_{wr}v^2 r \\
&\quad + Y_{vrr} v r^2) \\
&= -\frac{1}{2}\rho L_{pp} T V^2 (Y'_v v' + Y'_r r' + Y'_{vvv} v'^3 + Y'_{vvr} v'^2 r' \\
&\quad + Y'_{vrr} v' r'^2 + Y'_{rrr} r'^3) \qquad (5) \\
f_r(\boldsymbol{v}) &= -(N_v v + N_r r + N_{|v|v}|v|v + N_{|r|r}|r|r \\
&\quad + N_{wr} v^2 r + N_{vrv} v r^2) \\
&= -\frac{1}{2}\rho L_{pp}^2 T V^2 (N'_v v' + N'_r r' + N'_{vvv} v'^3 \\
&\quad + N'_{vvr} v'^2 r' + N'_{vrr} v' r'^2 + N'_{rrr} r'^3), \qquad (6)
\end{aligned}
$$

in which

$$
\begin{cases}
u' = u/V \\
v' = v/V \\
r' = r L_{PP}/V \\
V = \sqrt{u^2 + v^2}.
\end{cases} \qquad (7)
$$

The main control forces are the surge force $T_u(\cdot)$ and the yaw moment $F_r(\cdot)$, which are generated with different propeller revolution n (per second) and rudder angle δ, respectively.

The surge force $T_u(\cdot)$ is determined by propeller revolution n, propeller diameter D_P and the propeller thrust coefficient K_T:

$$
T_u(\cdot) = (1 - t_P)\rho D_P^4 K_T, \qquad (8)
$$

where K_T is commonly expressed by second order polynomials of the propeller advance ratio J_P as:

$$
K_T = k_2 J_P^2 + k_1 J_P + k_0, \qquad (9)
$$

in which J_P can be obtained as:

$$
J_P = \frac{u(1 - w_P)}{n D_P} \qquad (10)
$$

In equation (10), J_p is the propeller advance ratio, which is an indicator of the propeller performance in certain flow speed. $u(1 - w_P)$ represents the flow speed at the propeller. w_P is the wake factor at propeller position in maneuvering, which is commonly estimated based on the wake factor at propeller position in straight moving w_{P_0} and the geometrical inflow angle to the propeller in maneuvering β_P, defined as:

$$
\beta_P = \beta - x'_P r', \qquad (11)
$$

where $\beta = \arctan(-\frac{v}{u})$, $x'_P = x_P/L_{pp} = -0.48$ and x_P is longitudinal portion of propeller.

This paper applies the methods of w_P introduced by [23] as:

$$
\frac{(1 - w_P)}{(1 - w_{P_0})} = 1 + \{1 - exp(-C_1|\beta_P|)\}(C_2 - 1), \qquad (12)
$$

where w_{P_0} is the wake factor at propeller position in straight moving, C_1 and C_2 are experimental constants. Furthermore, C_1 and C_2 are different in motions for port and starboard owing to an asymmetric wake factor with respect to the propeller rotational effect.

The yaw moment $F_r(\cdot)$ is defined as:

$$
F_r(\cdot) = (x_R + a_H x_H)\left[-\frac{6.13\lambda}{\lambda + 2.25}\frac{A_R}{L_{PP}^2}(u_R^2 + v_R^2)\cos\delta\right], \qquad (13)
$$

where $a_H = a'_H = L_{PP}$, and $x_H = x'_H = L_{PP}$.

Considering the effect of the propeller on the increment of the rudder inflow velocity, the longitudinal velocity of the inflow to the rudder u_R is expressed as:

$$
u_R = u\varepsilon(1 - w_P)\sqrt{\eta\left\{1 + \kappa\left[\sqrt{\left(1 + \frac{8K_T}{\pi J^2}\right)} - 1\right]\right\}^2 + (1 - \eta)} \qquad (14)
$$

where $\varepsilon = (1 - w_R)/(1 - w_P)$, w_R is the wake factor at the rudder position in manoeuvring, κ is an experimental constant for expressing u_R, and η is the ratio of the propeller diameter to the rudder span. The lateral inflow velocity to the rudder v_R is written as:

$$
v_R = V\gamma_R(\beta - \ell'_R r'), \qquad (15)
$$

where γ_R is the flow straightening factor and different for port and starboard motions, $\ell'_R = \ell_R/L_{pp}$ is the effective longitudinal coordinate of rudder position. We refer the readers to [23] for more details on the nonlinear ship maneuverability model.

B. OPTIMAL CONTROL PROBLEM

For automatic ship docking, the control objective is to steer the ship state to follow the reference state $\boldsymbol{x_r} = [x_r, y_r, \psi_r, u_r, v_r, r_r]^\top$, in which $u_r = 0, v_r = 0, r_r = 0$ so that the ship is stopped, (x_r, y_r) are the designed docking position, ψ_r equals $0°$, $\pm 180°$ or other values, depending on the layout of quayside of port.

The state equation of ship dynamics can be represented as:

$$
\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}),
$$

where $\boldsymbol{x} = [x, y, \psi, u, v, r]^\top \in \mathcal{R}^6$ is the ship state vector, and $\boldsymbol{u}$ is the control input vector. Most research takes surge force T_u and yaw moment F_r as the main control variables, as they can lead to surge and yaw movements. In this paper, we consider the fact that shipmasters use rudder angle and propeller revolution rate docking operation in practice as the main control signal. Therefore, the control input is chosen as $\boldsymbol{u} = [n, \delta]^\top \in \mathcal{R}^2$, in which n refers to the propeller revolution rate and δ refers to the rudder angle.

The nonlinear ship dynamics are discretized with the Euler method, and then a system of discrete-time state-space equations with state $\boldsymbol{x}_k$, control input $\boldsymbol{u}_k$ is obtained. At each sampling instant k, the ship control objective is to follow the reference states, and it is formulated as an optimal control problem that takes the following form:

$$
\min_{\boldsymbol{x}(k), \boldsymbol{u}(k)} J(\boldsymbol{x}, \boldsymbol{u}) = \sum_{k=0}^{N-1} L(\boldsymbol{x}(k), \boldsymbol{u}(k)) + E(\boldsymbol{x}_{N|k}) \qquad (16)
$$

$$\text{subject to: } \boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{u}_k) \tag{17}$$
$$\boldsymbol{x}_{k=0} = \boldsymbol{x}_0 \tag{18}$$
$$\boldsymbol{x}_k \in X, \quad \boldsymbol{u}_k \in U, \ k \in I \tag{19}$$

where, $f(\cdot,\cdot)$ refers to the right-hand side of discretized equation (2), $X \subset \mathcal{R}^6$, $U \subset \mathcal{R}^2$. The objective function J consists of stage cost and terminal cost, in which the stage cost function is represented as $L = \|\boldsymbol{x}(t) - \boldsymbol{x}_r(t)\|_Q + \|\boldsymbol{u}(t) - \boldsymbol{u}_r(t)\|_R$, and the terminal cost function is formulated as $||\boldsymbol{x}(N) - \boldsymbol{x}_r(N)||_P$. Here, Q, R, P are positive semidefinite weight matrices. N refers to the prediction horizon. Objective function J aims to minimize the deviations of the system inputs and states from the reference states.

After formulating the optimal control problem, the objective of a NMPC problem is to find the optimal control sequence $\boldsymbol{U} = \{\boldsymbol{u}_0, \cdots, \boldsymbol{u}_{N-1}\}$ with prediction horizon N at each sample time k, such that the resulting state sequence $\boldsymbol{X} = \{\boldsymbol{x}_0, \cdots, \boldsymbol{u}_N\}$ and the control sequence $\boldsymbol{U}$ minimize the objective function $J(\boldsymbol{x}, \boldsymbol{u})$ without violating Constraints (17)-(19).

At each sampling instant k, the NMPC algorithm uses the discretized nonlinear ship model (2) and the measured states to predict the future system states within a pre-defined time. After solving the optimal problem, a nonlinear state feedback $\boldsymbol{u}$ is obtained.

III. CONTROLLER DESIGN BASED ON NONLINEAR MODEL PREDICTIVE CONTROL

This section introduces the controller design steps. Firstly, the reference path from the ship's current position to the docking position needs to be determined. Secondly, the automatic ship docking problem is formulated as a lexicographic multi-objective NMPC problem, so as to relieve the burden of controller parameter tuning. Thirdly, the solution steps of the proposed NMPC control scheme are introduced. Finally, the proof of stability of the MO-NMPC strategy is given.

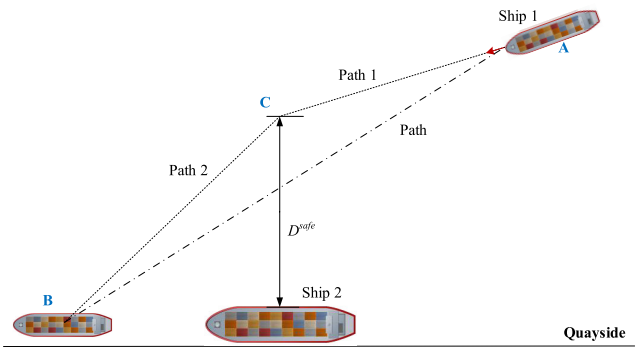


FIGURE 3. Ship docking path planning.

A. REFERENCE PATH GENERATION

Figure 3 gives an example of a reference docking path for Ship 1 when it starts docking from location A to berth B. If the nearby berths are empty, the reference path would be the direct line from point A to the point B. The reference path is generated as a series of points. Assuming that point A's coordinate is (x_A, y_A) and point B's coordinate is (x_B, y_B), and that takes time T for the ship from A to B in its regular speed, then a series of points are generated. When Ship 1 starts docking procedure from point A at time $t = 0$. The reference point (x_t, y_t) at time t, is calculated as follows:

$$x_t = x_B + (x_A - x_B) * \frac{T - t}{T},$$
$$y_t = y_B + (y_A - y_B) * \frac{T - t}{T}.$$

When the $t \geq T$, the reference point become (x_B, y_B) afterwards.

Under certain circumstances, it could happen that the neighborhood berth has been occupied by another ship. As shown in Figure 3, Ship 2 has been docked at the right side of berth B. Therefore, Ship 1 needs to keep enough distance with the Ship 2 and avoid collisions in its docking operation. The originally planned path, which is the direct line from A to B, is no longer feasible.

Because of this, reference point C is introduced, which divides the path into two parts. The coordinate of point C is determined by the midship of Ship 2 and the required safe distance D^{safe}. The control task from points A to C is to ensure that the distance between itself and Ship 2 should always be larger than the required safe distance before it passes point C. After Ship 1 passes point C, the collision avoidance task has been finished and the control task is to dock and stabilize the ship at berth B. The reference points from points A to C, and from points C to B are generated in a similar way as the reference points from points A to B.

B. LEXICOGRAPHIC MULTI-OBJECTIVE NMPC

As described earlier, the objective function of an NMPC problem typically is the weighted sum of different terms including the ship's position deviations, course deviations and velocity deviations from desired values. The selection of different weights generates different solutions. Selecting an appropriate set of weights could be a complicated task, which is usually done by trial and error. Therefore, considering the applicability of the NMPC controller, it is important to find an efficient way to tune the weights in the objective function.

Multi-objective optimization, which aims at finding optimal solutions for multiple and conflicting objectives, has been adopted in tuning MPC control parameters [16], [21], [24]. This paper adopts the multi-objective model predictive control scheme proposed in [21], in which the NMPC problem is reformulated as a lexicographic optimization problem. The optimal solutions are obtained by solving a series of optimization problems.

The discretized nonlinear ship dynamics are $\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{u}_k)$, where $\boldsymbol{x}_k$ and $\boldsymbol{u}_k$ are the state and control vectors at sampling time interval k. Consider a finite sequence of future control actions at time k, $\boldsymbol{u}_{k,N} = \{\boldsymbol{u}_{0|k}, \boldsymbol{u}_{1|k}, \cdots, \boldsymbol{u}_{N-1|k}\}$, in which N represents the prediction horizon. The overall

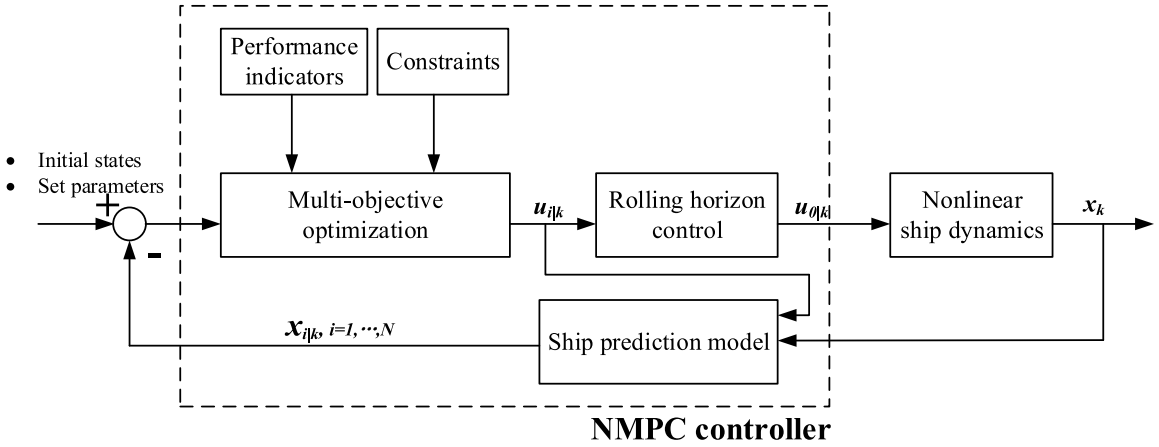


FIGURE 4. The scheme of the proposed auto-docking method.

objective function (16) is divided into several sub-functions, and their importance of different objective functions are ranked with an order from J_1 to J_M, in which J_1 is the most important and J_M is the least important. Therefore, the prioritized multi-objective optimization problem is formulated as

$$\min_{\boldsymbol{u}_{k,N}} J(\boldsymbol{u}, \boldsymbol{x})$$

where, $J(\boldsymbol{u}, \boldsymbol{x}) = [J_1(\boldsymbol{u}, \boldsymbol{x}), \cdots J_M(\boldsymbol{u}, \boldsymbol{x})]^T$ is the objective function vector that maps the constrained control sequence $\boldsymbol{u}$ and current state $\boldsymbol{x}$ to a set of N objective functions.

At time k, given the ship state $\boldsymbol{x}_k$ and control action sequence $\boldsymbol{u}_{k,N}$, the future ship states at time $k+t$ is denoted as $\boldsymbol{x}_{t|k}$. Therefore, $\boldsymbol{x}_{t+1|k} = f(\boldsymbol{x}_{t|k}, \boldsymbol{u}_{t|k})$ with $\boldsymbol{x}_{0|k} = \boldsymbol{x}_k$.

Solving a lexicographic multi-objective optimization problem involves finding:

$$\boldsymbol{u}^*_{k,N} = \arg\min_{\boldsymbol{u}_{k,N}} \{J_M(\boldsymbol{u}, \boldsymbol{x}) | J_j(\boldsymbol{u}_{k,N}, \boldsymbol{x}_{k,N}) \leq J^*_j(\boldsymbol{x}_k) + \sigma_j, \quad \forall j \in \{1, 2, \cdots, N-1\}\}$$

where $J^*_j(\boldsymbol{x}_k)$ represents the optimal value function of the j-th optimization problem, $\sigma_j \geq 0$ are small tolerance to be determined. Initially, the first or the most important objective function J_1 is minimized subject to original Constraints (17)-(19). When the problem is feasible and an optimal solution is obtained, it is also the solution to the whole optimization problem. Then the second most important objective function J_2 is minimized by adding a new constraint which ensures that the value of J_1 when solving J_2 could preserve its optimal value, i.e., it should not be worse than the value of the solution of the in the previous optimization problem. If this subproblem is feasible and has a unique solution, it is the solution to the original problem. Otherwise, the process continues. The whole process repeats at each time interval.

After minimizing the last objective function J_M, solution $\boldsymbol{u}^*_{k,N}$ is obtained, which is the lexicographic optimal solution to the whole problem. The MO-NMPC control law for the ship at time k will take the first value in the set $\boldsymbol{u}^*_{k,N} = \{\boldsymbol{u}^*_{0|k}, \boldsymbol{u}^*_{1|k}, \cdots, \boldsymbol{u}^*_{N-1|k}\}$, therefore $\boldsymbol{u}^{\text{nmpc}}_k = \boldsymbol{u}^*_{0|k}$.

C. ALGORITHMIC STEPS

The scheme of the proposed NMPC-based auto-docking method is given in Figure 4. Firstly, the ship prediction model is formulated and Euler discretized based on the nonlinear ship dynamics introduced in Section II-A. For the multi-objective NMPC formulation, according to the importance of the auto-docking operation in practice, three objective functions are considered:

$$J_1(\boldsymbol{u}_k, \boldsymbol{x}_k) = ||\boldsymbol{x}(N) - \boldsymbol{x}_r(N)||_{P_1} + \sum_{t=0}^{N-1}(||\boldsymbol{x}(t|k) - \boldsymbol{x}_r(k)||_{Q_1} + ||\boldsymbol{u}(t|k) - \boldsymbol{u}_r(k)||_{R_1}) \tag{20}$$

$$J_2(\boldsymbol{u}_k, \boldsymbol{x}_k) = ||\boldsymbol{x}(N) - \boldsymbol{x}_r(N)||_{P_2} + \sum_{t=0}^{N-1}(||\boldsymbol{x}(t|k) - \boldsymbol{x}_r(k)||_{Q_2} + ||\boldsymbol{u}(t|k) - \boldsymbol{u}_r(k)||_{R_2}) \tag{21}$$

$$J_3(\boldsymbol{u}_k, \boldsymbol{x}_k) = ||\boldsymbol{x}(N) - \boldsymbol{x}_r(N)||_{P_3} + \sum_{t=0}^{N-1}(||\boldsymbol{x}(t|k) - \boldsymbol{x}_r(k)||_{Q_3} + ||\boldsymbol{u}(t|k) - \boldsymbol{u}_r(k)||_{R_3}) \tag{22}$$

For the docking problem, it is important for the ship to stop accurately and stabilize at the desired berth. Therefore, the ship firstly needs to follow the reference states, when it arrives at close quarters of the quayside, it needs to stop and stabilizes at the desired position. This implies that variables x, y are more important in the beginning, and will be given higher priorities in the objective function. When it is close to the quayside, variables u, v, r will be given higher priorities in the objective function.

To get initial solutions, in objective function J_1, all the weights for state variables $\boldsymbol{x} = [x, y, \psi, u, v, r]^\top$ are equal, therefore matrix $Q_1 = diag[1, 1, 1, 1, 1, 1]$. Then variables (x, y) and (u, v) are given higher priorities in objective function J_2, and matrix $Q_2 = diag[0, 0, 1, 0, 0, 1]$. In objective function J_3, states (ψ, u, v, r) are given higher priorities, and matrix $Q_3 = diag[0, 0, 0, 1, 1, 0]$. In all the objective functions, $R_1 = R_2 = R_3 = diag[0.1, 0.1]$, and that $P_1 = P_2 = P_3 = diag[1, 1, 1, 1, 1, 1]$.

Algorithm 1 The Algorithmic Steps of Ship Docking Control Method

1. Set sampling time k, initialization at time $k = 0$, prediction horizon N, objective functions $J_i(\boldsymbol{u}_k, \boldsymbol{x}_k)$ and their priorities.
2. Measure the state $\boldsymbol{x}_k$ at time interval k.
3. Based on ship's current coordinate (x, y), plan the reference trajectories (x_r, y_r) to the docking location (x_R, y_R).
4. Compute the optimal control sequence $\boldsymbol{u}_k^*$:
   (1) Solve the 1st-layer problem (20) subject to Constraints (17)-(19), obtain one series of optimal control actions,
   $\boldsymbol{u}_{k,1}^* = \arg\min J_1(\boldsymbol{u}_k, \boldsymbol{x}_k)$, and the optimal objective value J_1^*;
   (2) Solve the 2nd-layer problem (21) subject to Constraints (17)-(19), obtain one series of optimal control actions,
   $\boldsymbol{u}_{k,2}^* = \arg\min\{J_2(\boldsymbol{u}_k, \boldsymbol{x}_k) | J_1(\boldsymbol{u}_k, \boldsymbol{x}_k) \le J_1^* + \sigma\}$, and the optimal objective value J_2^*;
   (3) Solve the 3rd-layer problem (22) subject to Constraints (17)-(19), obtain one series of optimal control actions,
   $\boldsymbol{u}_{k,3}^* = \arg\min\{J_3(\boldsymbol{u}_k, \boldsymbol{x}_k) | J_1(\boldsymbol{u}_k, \boldsymbol{x}_k) \le J_1^* + \sigma, J_2(\boldsymbol{u}_k, \boldsymbol{x}_k) \le J_2^* + \sigma\}$, and the optimal control sequence $\boldsymbol{u}_k^* = \boldsymbol{u}_{k,3}^*$.
5. Apply the first element $\boldsymbol{u}^{\text{nmpc}} = \boldsymbol{u}_{0|k,3}^*$ of the lexicographic optimal sequences to ship model (2), and update ship's states.
6. Wait for the next sample and set the time index $k = k+1$, then go to 2.

The control law $\boldsymbol{u}$ is obtained by solving on-line, at each sampling instant k, a finite horizon open-loop optimal control problem and using the actual state of the ship as the initial state. The optimization generates an optimal control sequence and the first element $\boldsymbol{u}_{0|k}$ of this sequence is applied to the ship. Based on the nonlinear ship dynamics, ship states are updated as $\boldsymbol{x}_{i|k}$. The algorithmic steps are given in Algorithm 1. It is noted that when the ship starts docking procedure, only optimization problem (20) needs to be solved. When the ship is close to the docking points, problems (20), (21) and (22) need to be solved sequentially afterward.

D. NONLINEAR OPTIMIZATION

To solve the nonlinear optimization problems (16) with objective functions J_1,J_2 and J_3 sequentially at each sampling interval k, an interior point method [25] is used. This section uses the solution process of objective function J_2 as an example to explain. When the first optimization problem J_1 is solved, the value of J_1^* is obtained. For optimization problem J_2, it has the following form:

$$\min_{\boldsymbol{u}_{k,N}} J_2(\boldsymbol{u}_k, \boldsymbol{x}_k) \tag{23}$$

$$s.t.\ h(\boldsymbol{u}_k, \boldsymbol{x}_k) = \boldsymbol{x}_{k+1} - f(\boldsymbol{x}_k, \boldsymbol{u}_k) = 0 \tag{24}$$

$$g(\boldsymbol{u}_k, \boldsymbol{x}_k) = J_1(\boldsymbol{u}_k, \boldsymbol{x}_k) - (J_1^* + \sigma) \le 0 \tag{25}$$

Given the current ship states value $\boldsymbol{x}_k$, the optimization problem (23) will be solved to get optimal control input $\boldsymbol{u}_k$ and predicted ship states. This paper adopts an interior method to replace the nonlinear program by a sequence of barrier sub-problems of the form:

$$\min_{z} \xi_\mu(z) = J_2(\boldsymbol{u}) - \mu \sum_{i=1}^{m} \ln \boldsymbol{s}_i \tag{26}$$

here, $\boldsymbol{u}_k$ is replaced with $\boldsymbol{u}$, and $\boldsymbol{s} > 0$ is a vector of slack variables, $z = (\boldsymbol{u}, \boldsymbol{s})$ and $\mu > 0$ is the barrier parameter. The Lagrangian function associated with problem is defined as:

$$\mathcal{L}(\boldsymbol{z}, \boldsymbol{\lambda}; \mu) = \xi_\mu(z) + \lambda_h^T h(\boldsymbol{u}) + \lambda_g^T (g(\boldsymbol{u}) + \boldsymbol{s}) \tag{27}$$

where λ_h and λ_g are Lagrange multipliers and $\boldsymbol{\lambda} = (\lambda_h, \lambda_g)$. The first-order optimality condition for the barrier problem (26) can be written as:

$$\begin{bmatrix} \nabla J_2(\boldsymbol{u}_k) + A_h(\boldsymbol{u})^T \lambda_h + A_g(\boldsymbol{u})^T \lambda_g \\ S\Lambda_g e - \mu e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tag{28}$$

together with (24) and (25) and restrictions that s and λ_g be non-negative. Here, S and Λ_g represent diagonal matrices whose diagonal entries are given by vector s and λ_g, respectively. e is the vector of all ones. Matrices A_h and A_g are the Jacobian matrices of h and g. Then Newton's method is applied to (28), (24) and (25). Based on the value of $(\boldsymbol{z}, \boldsymbol{\lambda})$ in current iteration, a primal-dual system is formulated:

$$\begin{bmatrix} W(\boldsymbol{z}, \boldsymbol{\lambda}; \mu) & A(\boldsymbol{u})^T \\ A(\boldsymbol{u}) & 0 \end{bmatrix} \begin{bmatrix} d_z \\ d_\lambda \end{bmatrix} = - \begin{bmatrix} \nabla_z \mathcal{L}(\boldsymbol{z}, \boldsymbol{\lambda}; \mu) \\ c(z) \end{bmatrix}, \tag{29}$$

where,

$$A(\boldsymbol{u}) = \begin{bmatrix} A_h(\boldsymbol{u}) & 0 \\ A_g(\boldsymbol{u}) & I \end{bmatrix}, \tag{30}$$

$$W(\boldsymbol{z}, \boldsymbol{\lambda}; \mu) = \nabla_{zz} \mathcal{L}(\boldsymbol{z}, \boldsymbol{\lambda}; \mu) = \begin{bmatrix} \nabla_{uu}^2 \mathcal{L}(\boldsymbol{z}, \boldsymbol{\lambda}; \mu) & 0 \\ 0 & S^{-1}\Lambda_g \end{bmatrix}. \tag{31}$$

Then the new iterate is given by

$$z^+ = z + \alpha_z d_z, \boldsymbol{\lambda}^+ = \boldsymbol{\lambda} + \alpha_\lambda d_\lambda. \tag{32}$$

The step-lengths α_z and α_λ are computed in two stages. Firstly, we compute:

$$\alpha_z^{\max} = \max\{\alpha \in (0, 1] : \boldsymbol{s} + \alpha d_s \ge (1 - \tau)\boldsymbol{s}\} \tag{33}$$

$$\alpha_\lambda^{\max} = \max\{\alpha \in (0, 1] : \lambda_g + \alpha d_g \ge (1 - \tau)\lambda_g\}, \tag{34}$$

in which $\tau \in (0, 1)$. Then a backtracking line search is performed, which computes step-lengths:

$$\alpha_z \in (0, \alpha_z^{\max}], \quad \alpha_\lambda \in (0, \alpha_\lambda^{\max}], \tag{35}$$

that could provide a sufficient decrease in a merit function. For more details regarding the formulation of merit function and setting barrier parameter μ, we refer the readers to [25].

Here, we list the termination criteria of the nonlinear optimization algorithm. The algorithm terminates if an iterate $(\boldsymbol{z}, \boldsymbol{\lambda})$ satisfies

$$\|\nabla J_2 + A_h^T \lambda_h + A_g^T \lambda_g\|_\infty \leq \max\{1, \|\nabla J_2\|_\infty\}\epsilon^{\text{opt}} \quad (36)$$

$$\|S\lambda_g\|_\infty \leq max\{1, \|\nabla J_2\|_\infty\}\epsilon^{\text{opt}} \quad (37)$$

$$\|h, \max\{0, g\}\|_\infty \leq \max\{1, \|h, \max\{0, g\}\|_\infty\}\epsilon^{\text{feas}} \quad (38)$$

where ϵ^{opt} and ϵ^{feas} are termination tolerances to be determined.

E. PROOF OF STABILITY

To guarantee the asymptotic stability by using the control law $\boldsymbol{u}_k^{\text{nmpc}}$, it is desirable to use infinite prediction and control horizons. While it is not feasible to get solutions for an infinite horizon nonlinear optimization problem, stability of the lexicographic MO-NMPC problem can still be guaranteed by choosing suitable objective functions and region of attraction. This has been studied in [24]: considering M prioritized objectives of system (17), which are represented by $J_i(\boldsymbol{u}_{k,N}, \boldsymbol{x}_k) = E_i(\boldsymbol{x}_{N|k}) + \sum_{t=0}^{N} L_i(\boldsymbol{x}_{t|k}, \boldsymbol{u}_{t|k}), i \in M$, where L_i represents stage costs, and E_i represent terminal costs, and that both stage and terminal costs are continuous on their arguments. The required stability conditions are summarized as follows:

1) For the first layer problem J_1, the stage cost L_1 and terminal cost are positive-definite functions with respect to their arguments;
2) There exist an invariant set $\Omega \subseteq X$ of system (17), containing the origin in its interior, and a local control law $\boldsymbol{u} = U_1^{\text{local}}(\boldsymbol{x})$ such that
$$U_1^{\text{local}} \in U, f(\boldsymbol{x}, U_1^{\text{local}}) \in \Omega,$$
$$E_1(f(\boldsymbol{x}, U_1^{\text{local}})) - E_1(\boldsymbol{x}) + L_1(\boldsymbol{x}, U_1^{\text{local}}) \leq 0$$
for any $\boldsymbol{x} \in \Omega$.
3) The first layer sub-problem J_1 is feasible in the set of admissible states $X^{\text{nmpc}}(N) = \{x \in X | \exists \boldsymbol{u}_{k,N} \in U^N, s.t.(x, \boldsymbol{u}_{k,N}) \in Z(N)\}$ at time $k = 0$, where U^N is the product of N sets U, and $Z(N)$ is defined as an admissible set of $(\boldsymbol{x}, \boldsymbol{u}_{k,N})$ pairs:
$$Z(N) = \{(\boldsymbol{x}, \boldsymbol{u}_{k,N}) | \boldsymbol{x}_{t+1|k} = f(\boldsymbol{x}_{t|k}, \boldsymbol{u}_{t|k}), \boldsymbol{x}_{0|k} = \boldsymbol{x}, \boldsymbol{x}_{t|k} \in X, \boldsymbol{u}_{t|k} \in U, \boldsymbol{x}_{N|k} \in \Omega\}.$$

If Conditions 1)-3) hold, then the system in closed-loop with the controller obtained by Algorithm 1 is asymptotically stable with the region of attraction $X^{\text{nmpc}}(N)$.

For our problem, the stage cost $L_i = ||\boldsymbol{x}(t|k) - \boldsymbol{x}_r(k)||_{Q_i} + ||\boldsymbol{u}(t|k) - \boldsymbol{u}_r(k)||_{R_i}$, terminal cost $E_i(\boldsymbol{x}_{N|k}) = ||\boldsymbol{x}(N) - \boldsymbol{x}_r(N)||_{P_i}$. As indicated in earlier sections, our problem formulation satisfies these conditions.

IV. SIMULATION RESULTS

To evaluate the effectiveness of the proposed NMPC controller, the KVLCC2 tanker is used, using the ship parameters given in [26]. The hydrodynamic parameters are listed in Table 4. Our experiments are performed on an Intel Core i7-7500 CPU with 8GB RAM running Windows 10 and are implemented in MATLAB R2019a. The NMPC controller is implemented with MATLAB Model Predictive Control Toolbox. MATLAB-fmincon function was used as the nonlinear optimization solver. To speed up the optimization process, parameters ϵ^{opt} and ϵ^{feas} are set as 0.001. The maximum iterations are set at 1200. The prediction horizon length is selected to be $N = 30s$ and the sampling interval is $k = 1s$, the control horizon is $T_c = 1s$. which leads to a discrete horizon of 30 samples.

Two types of scenarios are considered, one with a ship at the nearby berth and one without. The docking target states in both scenarios are $x = 0, y = 0$ and $\psi = -180°$. The minimum and maximum limits for propeller revolution rate and rudder angle are considered as:$-10.34 \leq n \leq 10.34$ and $-35° \leq \delta \leq +35°$.

A. SCENARIO 1: WITHOUT COLLISION AVOIDANCE

In docking practice, a ship usually starts with an approach angle smaller than 30°. An approach angle refers to the angle between the ship's course and the quayside when it starts docking. To verify the applicability of the proposed control scheme, four cases with different approach angles are considered. It is noted that in these cases, the ship's approach angles are 10°, 20°, 30° and 40°, the ship's course angle is −170°, −160°, −150° and −140°, respectively.

- Case 1: $x = 70, y = 12.35, \psi = -170°, u = 1.2, v = 0, r = 0$;
- Case 2: $x = 70, y = 25.48, \psi = -160°, u = 1.2, v = 0, r = 0$;
- Case 3: $x = 70, y = 40.42, \psi = -150°, u = 1.2, v = 0, r = 0$;
- Case 4: $x = 70, y = 58.74, \psi = -140°, u = 1.2, v = 0, r = 0$.

Figure 5 shows the ship docking trajectories in Cases 1-4. The ship in all cases finally stabilizes at (0, 0) with the desired heading angle −180°.

Figure 6 shows the plots of the surge velocity u, sway velocity v and yaw rate r over time. The surge speed u stabilizes gradually at $0m/s$ at around 200 seconds, and so does the sway velocity v and yaw rate r.

Figure 7 shows the curves of control inputs including propeller revolution rate n and rudder angle δ. It can be seen from Figure 7a that the propeller revolution rate in Case 1 stabilizes at 0 within around 500 seconds, and that it stabilizes at 0 within around 550 seconds, 650 seconds and 700 seconds in Cases 2, 3 and 4, respectively. This is consistent with docking operation practice, the larger the approach angle is, the more difficult for the shipmaster to maneuver the ship at the desired position. Figure 7b shows the same pattern, the rudder angle in Case 1 stabilizes at 0 the first and then Cases 2, 3 and 4. Figure 7 also shows that the ship adjusts its heading angles first, and then uses a propeller revolution rate to decelerate

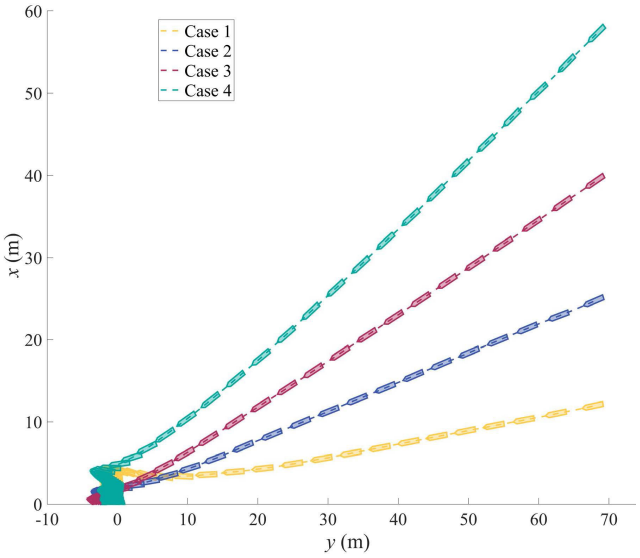


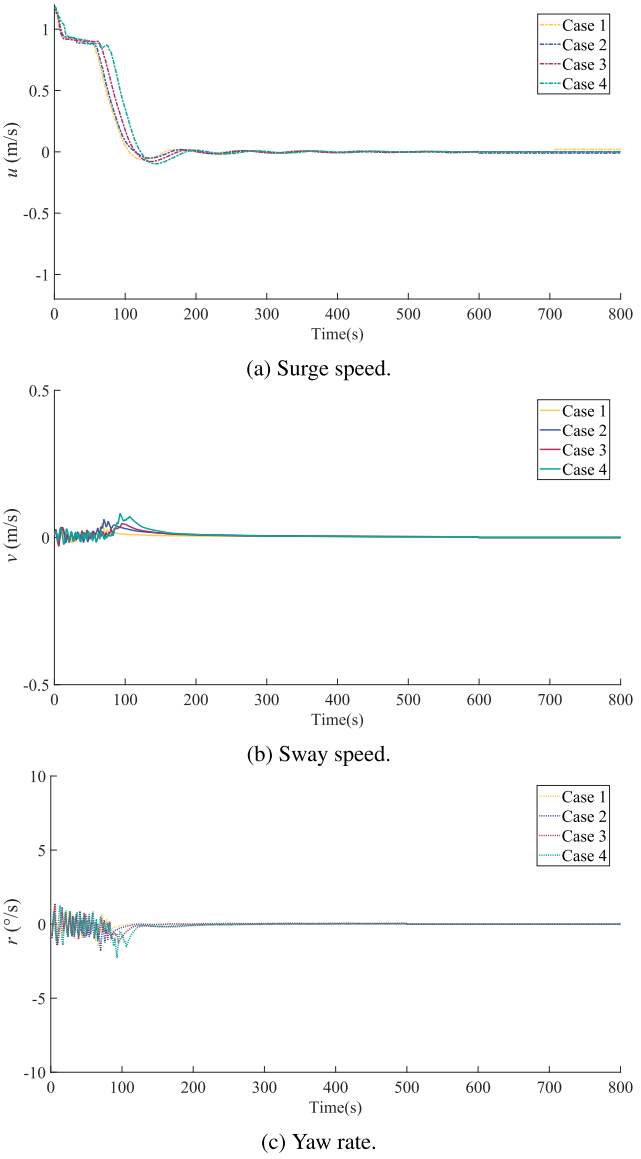
FIGURE 5. Simulated ship docking trajectories in Cases 1-4.

and stop at the desired location. Therefore, the rudder angles stabilize at an earlier time than the propeller revolution rate.

TABLE 1. Computation time in each iteration in Cases 1-4 (in seconds).

	Max	Min	Avg
Case 1	3.9754	0.0424	0.2476
Case 2	3.2966	0.0719	0.3766
Case 3	5.0450	0.0717	0.2975
Case 4	4.7916	0.0423	0.2543

Table 1 presents the maximum, minimum and average computation times in each iteration in Cases 1-4. While in some rate situations, the computation time could be around 3 or 5 seconds, in most situations it is far less than the 1.0 seconds sampling interval, which is acceptable for practical implementations.



(a) Surge speed.

(b) Sway speed.

(c) Yaw rate.

FIGURE 6. Changes of ship states over time in Cases 1-4.

B. SCENARIO 2: COLLISION AVOIDANCE

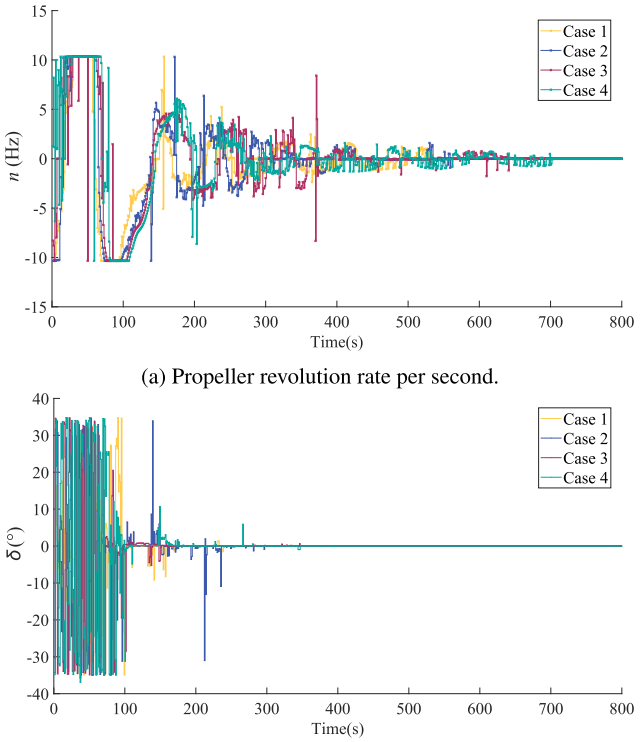
For the second scenario, the initial ship states is set as $x = 70, y = 25.48, \psi = -160°, u = 1.2, v = 0, r = 0$. A neighbor docked ship A with a length of $10m$ and width of $5m$ is considered. Different safe distances $D = 8m, D = 12m$, and $D = 16m$ are considered and labeled as Cases 5, 6 and 7, respectively.

The x coordinate of Ship A's midship is 20, and the y coordinates of its starboard side is 3. Therefore, to ensure the safe distance, the reference points for path planning are set as (20, 11), (20, 15) and (20, 19) accordingly in Cases 5, 6 and 7. The collision avoidance task is to ensure that the ship's y coordinates should be larger than 11, 15 and 19 when it passes $x = 20$ in Cases 5-7.

Figure 8 shows the ship docking trajectories in Cases 5-7. It can be seen that the ship can avoid collisions with the neighboring docked ship and keep enough distances. In all cases, the ship needs to decelerate and accelerate to avoid collisions, and then decelerate again to stabilize at the desired location. In Case 7, the ship needs to make larger adjustments in its heading angle, therefore the increase of its speed is larger than the increase of speeds in Cases 5 and 6. This also means that it takes a longer time and distance for the ship to decelerate in Case 7 than in Cases 5 and 6. Therefore, the ships in Cases 6 and 7 reaches $x = -15$ and $x = -10$ and then moves back to $x = 0$.

Figure 9 shows the curves of surge velocity u, sway velocity v and yaw rate r over time. It can be seen that they stabilize gradually at 0m/s within 300 seconds. As the ship needs to accelerate so as to change its heading angle to avoid collisions with the docked ship A, the surge speed u has increased between 80 seconds to 120 seconds.

Figure 10 shows the curves of control inputs including propeller revolution rate n and rudder angle δ. Similar to



(a) Propeller revolution rate per second.

(b) Rudder angles.

FIGURE 7. Changes of control inputs over time in Cases 1-4.

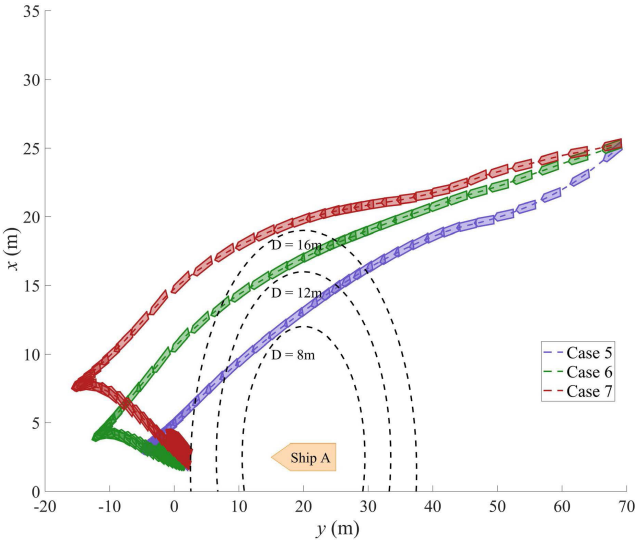


FIGURE 8. Simulated ship docking trajectories in Cases 5-7.

Figure 9a, the propeller revolution rate has also increased from 80 seconds to 120 seconds to accelerate the ship, so as to change its heading angle to avoid collisions. In Figure 10a, the propeller revolution rate stabilizes at around 400 seconds, 600 seconds, and 730 seconds in Cases 5, 6 and 7, respectively. In Figure 10b, the rudder angle stabilizes at around 200 seconds, 250 seconds, and 300 seconds in Cases 5, 6, and 7, respectively. The ship also takes the strategy to firstly adjust its heading angles and then decelerate its speed. Therefore, the rudder angle stabilizes at an earlier time than the propeller revolution rate.

TABLE 2. Computation time in each iteration in Cases 5-6 (in seconds).

	Max	Min	Avg
Case 5	4.7808	0.0380	0.3092
Case 6	4.3441	0.0391	0.3166
Case 7	4.2994	0.0447	0.4071

Table 2 presents the maximum, minimum and average computation times in each iteration in Cases 5-6. Compared with Cases 1-4, the computation times are slightly longer, which means it takes a longer time to find solutions to follow the collision-free path. The average computation time is around 0.3 seconds, which is much smaller than the sampling time.

C. COMPARISON WITH DIRECT NMPC METHOD

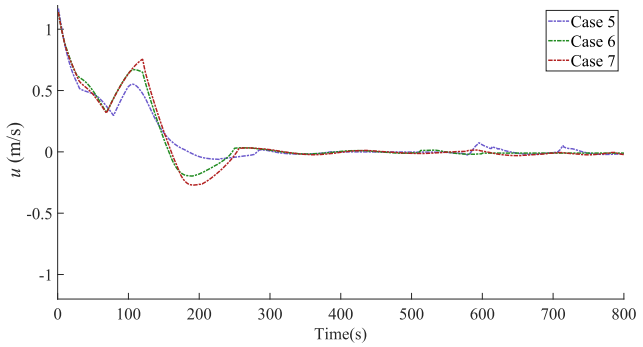
Simulation experiments are also carried out to compare the proposed MO-NMPC approach with a direct NMPC approach. In the direct NMPC, the controller directly solves just one optimization problem J_0 in each iteration instead of three optimization problems in MO-NMPC:

$$J_0(\boldsymbol{u}_k, \boldsymbol{x}_k) = ||\boldsymbol{x}(N) - \boldsymbol{x}_r(N)||_{P_0} + \sum_{t=0}^{N-1}(||\boldsymbol{x}(t|k) - \boldsymbol{x}_r(k)||_{Q_0} + ||\boldsymbol{u}(t|k) - \boldsymbol{u}_r(k)||_{R_0}). \quad (39)$$

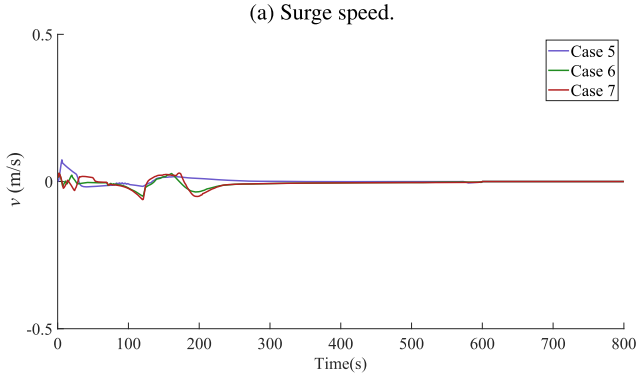
In order to make it consistent with the weights assigned to objective functions J_1, J_2 and J_3, we set $Q_0 = diag[1, 1, 1, 1, 1, 1]$, $R_0 = R_1 = R_2 = R_3 = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$, and that $P_0 = P_1 = P_2 = P_3 = diag[1, 1, 1, 1, 1, 1]$.

Figure 12 shows the comparison of the control performance of MO-NMPC and direct NMPC methods, with respect to ship position variables x, y, ψ Cases 1-4. The results of direct NMPC is labeled as "-NMPC". It can be seen that with MO-NMPC method, the value of variable x converges to 0 with a shorter time than the direct NMPC method in all cases. While the value of variable y in MO-NMPC method reaches 0 with a longer time than the direct NMPC method in all cases, it finally stabilizes at 0. Meanwhile, the value of variable y in direct NMPC method does not converge to 0, and small deviations exist in all cases. Moreover, the ship cannot reach the desired heading angle with the direct NMPC method, and that large deviations still exist at the end of the simulation, while the value of ship heading angle reaches and stabilizes at the desired value with MO-NMPC method.

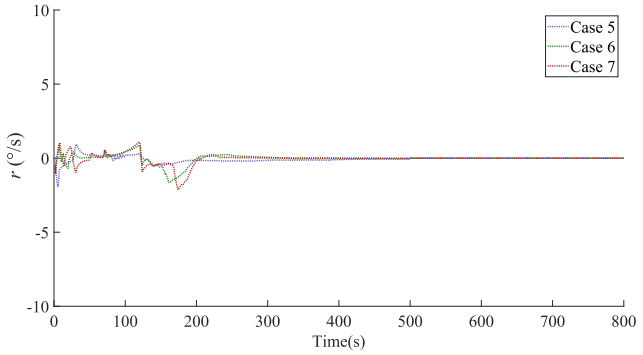
Figure 13 shows the comparison on the control performance with respect to ship states motion variables u, v, r in Cases 1-4. It can be seen that the value of variables u converges to 0 with both methods in all cases and that in Cases 3 and 4, it converges at an earlier time with MO-NMPC than with NMPC. The value of variable v converges to 0 with both methods in all cases. The value of variable r converges to



(a) Surge speed.

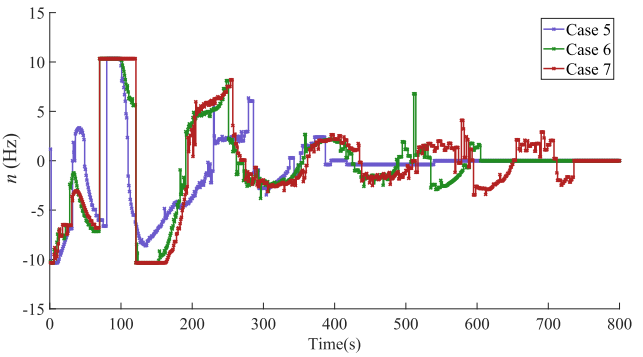


(b) Sway speed.

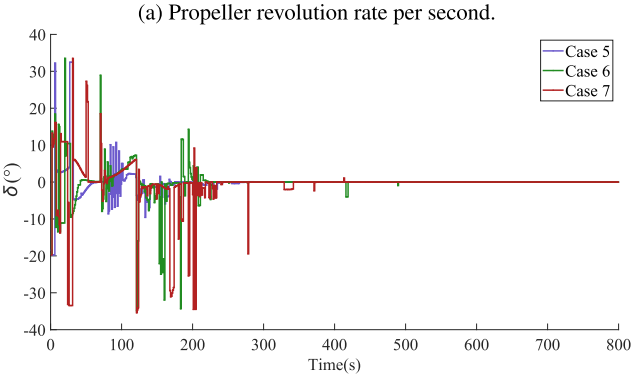


(c) Yaw rate.

FIGURE 9. Changes of ship states over time in Cases 5-7.



(a) Propeller revolution rate per second.



(b) Rudder angle.

FIGURE 10. Changes of control inputs over time in Cases 5-7.

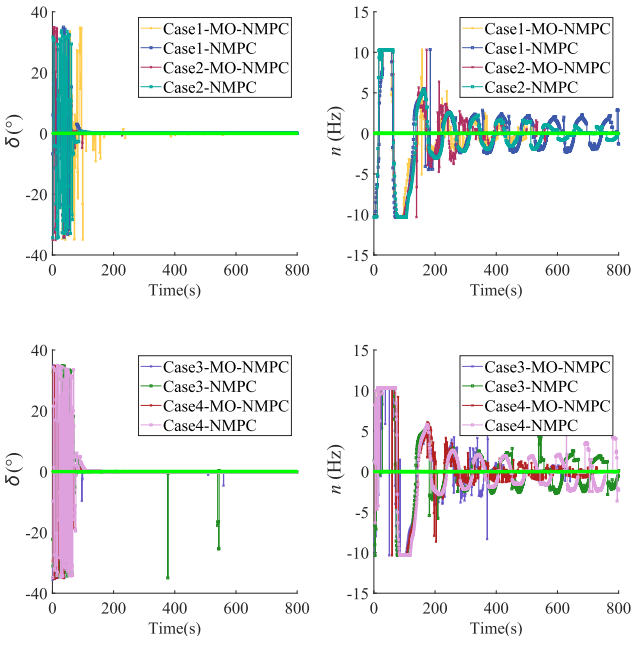


FIGURE 11. Comparison of direct NMPC and proposed MO-NMPC with respect to δ and n.

0 in all cases with MO-NMPC method, while small deviations exist in Cases 2 and 3 with direct NMPC.

Figure 11 shows the comparison of control inputs δ and n in Cases 1-4. It can be seen that the value of the rudder angle δ with both methods converges to 0. For propeller revolution rate n, with the MO-NMPC method, it could stabilize at 0 in all cases within the simulation time. Meanwhile, the value of variable n with direct NMPC method does not converge in all cases.

It can be seen from the simulation results in Cases 1-4 that the proposed MO-NMPC method performs better than the direct NMPC method with respect to ship state variables and control inputs.

Table 3 presents the maximum, minimum and average computation times in each iteration with the direct NMPC method in Cases 1-4. Compared with the computation time of the proposed MO-NMPC method in Table 1, the computation times are slightly shorter. This is because, in the MO-NMPC-based method, the controller needs to solve three optimization problems, while the direct NMPC controller

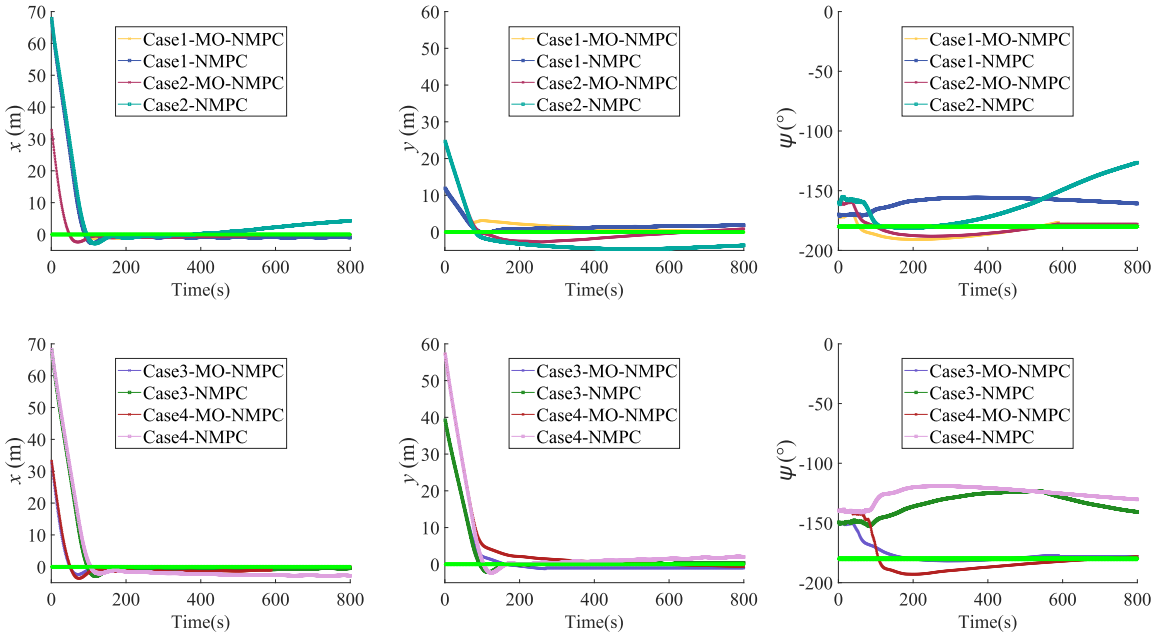


FIGURE 12. Comparison of direct NMPC and proposed MO-NMPC with respect to x, y and ψ.

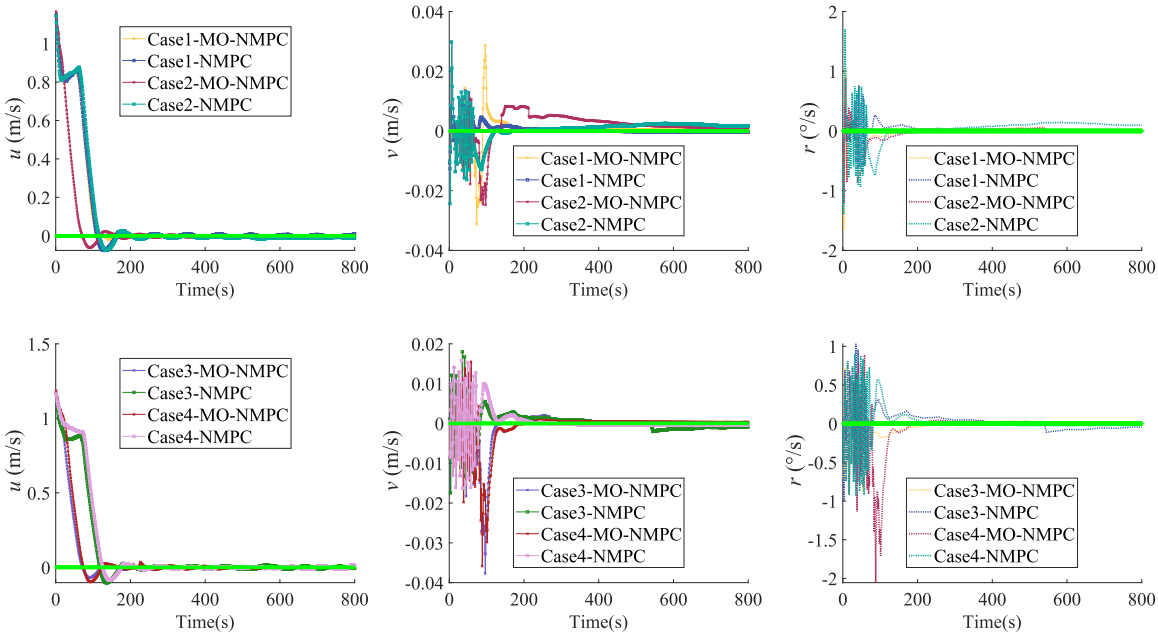


FIGURE 13. Comparison of direct NMPC and proposed MO-NMPC with respect to u, v and r.

TABLE 3. Computation time in each iteration in Cases 1-4 (in seconds).

	Max	Min	Avg
Case 1	4.4810	0.0380	0.1424
Case 2	2.9523	0.0377	0.1279
Case 3	4.0860	0.0376	0.1421
Case 4	3.8706	0.0375	0.1233

only solves one optimization problem in each iteration. In all cases, the average computation times of the direct NMPC method is much shorter than the 1.0 seconds sampling interval.

D. RESULTS DISCUSSIONS

Based on the simulation results, we can concluded that:

1) Rudder angle and propeller revolution rate have been introduced as the main control variables, which makes the control results more in line with the actual ship situation.
2) The proposed MO-NMPC-based automatic docking method could successfully lead the ship to the desired berthing point. It could also avoid collisions when another ship has been docked in its neighboring berth.

TABLE 4. Applied parameters in the simulations of the KVLCC2 tanker. Source: [23].

Main particulars of the KVLCC2 tanker			
L_{PP}	7.0	m'_{xx}	0.022
B_{WL}	1.1688	m'_{yy}	0.223
∇	3.2724	x_G	0.244
C_b	0.8098	J'_{zz}	0.011
T	0.455		
Parameters for hull forces and moments			
R'_0	0.022	Y'_{vrr}	-0.391
X'_{vv}	-0.040	Y'_{rrr}	0.008
X'_{vr}	0.002	N'_v	-0.137
X'_{rr}	0.011	N'_r	-0.049
X'_{vvvv}	0.771	N'_{vvv}	-0.030
Y'_v	-0.315	N'_{vvr}	-0.294
Y'_r	0.083	N'_{vrr}	0.0550
Y'_{vvr}	0.379	N'_{rrr}	-0.013
Y'_{vvv}	-1.607		
Parameters for propeller forces and moments			
D_P	0.216	k_1	-0.2753
k_0	0.2931	k_2	-0.1385
t_P	0.220	w_{P_0}	0.40
x'_P	-0.48	n	10.4
Parameters for rudder forces and moments			
A_R	0.0539	t_R	0.387
a_H	0.312	ℓ'_R	-0.710
x'_H	-0.464	ε	1.09
C_1	2.0 4	κ	0.50
C_2 $(\beta_P > 0)$	1.6	$\dot{\delta}$	15.8
C_2 $(\beta_P < 0)$	1.1	η	0.6252
γ_R $(\beta_R < 0)$	0.395	Λ	1.827
γ_R $(\beta_R > 0)$	0.640	x'_R	-0.50

3) The proposed MO-NMPC-based docking controller shows better performance than a direct NMPC controller on the convergence of control variables. It also shows better performance in tracking reference states and stabilizing at desired positions.
4) It is possible that a direct NMPC method could provide the same control performance as the MO-NMPC after taking trial-and-error experiments with different weights in the objective function. However, it is a time-consuming process, and our proposed MO-NMPC-based method could save the efforts on tuning these parameters.
5) While the proposed MO-NMPC controller takes slightly longer computation time than the direct NMPC controller in each iteration, it is still much shorter than the sampling time interval in general. This could facilitate the implementation in practice.

V. CONCLUSION AND FUTURE WORK

To ensure safe and autonomous ship docking, this paper proposes a nonlinear model predictive control method. A 3-DOF ship model is used with a propeller revolution per second and rudder angle as the control variables. The automatic ship docking problem is formulated as an NMPC problem, in which the control objective is to make sure that the ship state follows the reference state. A quadratic cost function is adopted, in each sample time, the optimization objective is to minimize the deviations of the ship's current states from desired states. A multi-objective optimization strategy is proposed, which could save the efforts on tuning the weights in the objective function of the formulated NMPC problem. Simulation results are promising and validate the effectiveness of the proposed method.

For future work, this research will be extended in several directions. Firstly, this paper did not consider the external disturbances including wind, wave, and current. External environmental forces might lead to a tracking error if they are not considered in the controller design. Therefore, we will consider the measurements or estimation of these disturbances by introducing a nonlinear disturbance observer, so that the disturbances could be compensated. Secondly, as the hydrodynamic coefficients of the hull, the propeller, and the rudder will be changed in different ship speeds, we will consider this in our future work after the identification of these coefficients in basin tests. Thirdly, a time delay of the control signal and input saturation of rudder and propeller will also be considered in our future work.

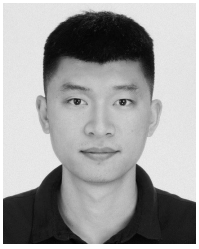
In addition, due to the underactuated characteristic of the ship, the proposed control method does not apply to all initial docking positions. This could be done by designing intelligent path planning algorithms integrated with practitioners' experiences, and carrying out experimental tests in the future.

REFERENCES

[1] N.-K. Im and V.-S. Nguyen, "Artificial neural network controller for automatic ship berthing using head-up coordinate system," *Int. J. Nav. Archit. Ocean Eng.*, vol. 10, no. 3, pp. 235–249, May 2018.
[2] N. Mizuno, Y. Uchida, and T. Okazaki, "Quasi real-time optimal control scheme for automatic berthing," *IFAC-PapersOnLine*, vol. 48, no. 16, pp. 305–312, 2015.
[3] A. Maki, N. Sakamoto, Y. Akimoto, H. Nishikawa, and N. Umeda, "Application of optimal control theory based on the evolution strategy (CMA-ES) to automatic berthing," *J. Mar. Sci. Technol.*, vol. 25, no. 1, pp. 221–233, Mar. 2020.
[4] L. Wang, Q. Wu, J. Liu, S. Li, and R. R. Negenborn, "Ship motion control based on AMBPS-PID algorithm," *IEEE Access*, vol. 7, pp. 183656–183671, 2019.
[5] Z. Qiang, Z. Guibing, H. Xin, and Y. Renming, "Adaptive neural network auto-berthing control of marine ships," *Ocean Eng.*, vol. 177, pp. 40–48, Apr. 2019.
[6] Y. A. Ahmed and K. Hasegawa, "Automatic ship berthing using artificial neural network trained by consistent teaching data using nonlinear programming method," *Eng. Appl. Artif. Intell.*, vol. 26, no. 10, pp. 2287–2304, Nov. 2013.
[7] J. H. Lee, "Model predictive control: Review of the three decades of development," *Int. J. Control, Automat. Syst.*, vol. 9, no. 3, p. 415, 2011.
[8] S.-R. Oh and J. Sun, "Path following of underactuated marine surface vessels using line-of-sight based model predictive control," *Ocean Eng.*, vol. 37, nos. 2–3, pp. 289–295, Feb. 2010.
[9] B. J. Guerreiro, C. Silvestre, R. Cunha, and A. Pascoal, "Trajectory tracking nonlinear model predictive control for autonomous surface craft," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 6, pp. 2160–2175, Nov. 2014.
[10] Z. Yan and J. Wang, "Model predictive control for tracking of underactuated vessels based on recurrent neural networks," *IEEE J. Ocean. Eng.*, vol. 37, no. 4, pp. 717–726, Oct. 2012.
[11] A. Jayasiri, A. Nandan, S. Imtiaz, D. Spencer, S. Islam, and S. Ahmed, "Dynamic positioning of vessels using a UKF-based observer and an NMPC-based controller," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 4, pp. 1778–1785, Oct. 2017.
[12] M. Abdelaal, M. Fränzle, and A. Hahn, "Nonlinear model predictive control for trajectory tracking and collision avoidance of underactuated vessels with disturbances," *Ocean Eng.*, vol. 160, pp. 168–180, Jul. 2018.

[13] J. Zhang, T. Sun, and Z. Liu, "Robust model predictive control for path-following of underactuated surface vessels with roll constraints," *Ocean Eng.*, vol. 143, pp. 125–132, Oct. 2017.
[14] H. Li and W. Yan, "Model predictive stabilization of constrained underactuated autonomous underwater vehicles with guaranteed feasibility and stability," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 3, pp. 1185–1194, Jun. 2017.
[15] H. Li, P. Xie, and W. Yan, "Receding horizon formation tracking control of constrained underactuated autonomous underwater vehicles," *IEEE Trans. Ind. Electron.*, vol. 64, no. 6, pp. 5004–5013, Jun. 2017.
[16] H. Zhao, J. Shen, Y. Li, and J. Bentsman, "Preference adjustable multi-objective NMPC: An unreachable prioritized point tracking method," *ISA Trans.*, vol. 66, pp. 134–142, Jan. 2017.
[17] J. Engwerda, *English LQ Dynamic Optimization and Differential Games*. Hoboken, NJ, USA: Wiley, 2005.
[18] F. Gembicki and Y. Haimes, "Approach to performance and sensitivity multiobjective optimization: The goal attainment method," *IEEE Trans. Autom. Control*, vol. 20, no. 6, pp. 769–771, Dec. 1975.
[19] V. M. Zavala and A. Flores-Tlacuahuac, "Stability of multiobjective predictive control: A utopia-tracking approach," *Automatica*, vol. 48, no. 10, pp. 2627–2632, Oct. 2012.
[20] T. Zheng, G. Wu, G.-H. Liu, and Q. Ling, "Multi-objective nonlinear model predictive control: Lexicographic method," in *Model Predictive Control*. Rijeka, Croatia: InTechOpen, 2010, ch. 7.
[21] D. He, L. Wang, and J. Sun, "On stability of multiobjective NMPC with objective prioritization," *Automatica*, vol. 57, pp. 189–198, Jul. 2015.
[22] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. New York, NY, USA: Wiley, 2011, pp. 48–55.
[23] H. Yasukawa and Y. Yoshimura, "Introduction of MMG standard method for ship maneuvering predictions," *J. Mar. Sci. Technol.*, vol. 20, no. 1, pp. 37–52, Mar. 2015.
[24] M. Vallerio, J. Van Impe, and F. Logist, "Tuning of NMPC controllers via multi-objective optimisation," *Comput. Chem. Eng.*, vol. 61, pp. 38–50, Feb. 2014.
[25] R. A. Waltz, J. L. Morales, J. Nocedal, and D. Orban, "An interior algorithm for nonlinear optimization that combines line search and trust region steps," *Math. Program.*, vol. 107, no. 3, pp. 391–408, Jul. 2006.
[26] *SIMMAN 2008 Committee*. (2008). *MOERI Tanker KVLCC2: Geometry and Test Conditions*. Accessed: Jun. 1, 2018. [Online]. Available: http://www.simman2008.dk/KVLCC/KVLCC2/tanker2.html

SHIJIE LI was born in Jingmen, Hubei, China, in 1988. She received the M.S. degree in control theory and engineering from Harbin Engineering University, Harbin, China, in 2013, and the Ph.D. degree in transport engineering and logistics from the Delft University of Technology, Delft, The Netherlands, in 2016. Her research interests include collaborative optimization of logistics systems, ship motion control, and multiship system operation optimization.



JIALUN LIU (Member, IEEE) was born in Fushun, Liaoning, China, in 1987. He received the M.S. degree in traffic information engineering and control from the Wuhan University of Technology, Wuhan, China, in 2013, and the Ph.D. degree in ship design, production and operation from the Delft University of Technology, Delft, The Netherlands, in 2017. He is currently working at the Intelligent Transportation Systems Research Center, Wuhan University of Technology, as an Associate Professor. His research interests include the motion control of smart ships, (inland) ship maneuverability, and functional testing of smart ships.

RUDY R. NEGENBORN received the Ph.D. degree from the Delft University of Technology, Delft, The Netherlands, in 2007. He is currently a Full Professor of multimachine operations and logistics (full-time, fixed-term), the Head of the Section Transport Engineering and Logistics, and the Director of the studies MSC transport, infrastructure, and logistics. He is the author of four books and more than 100 articles. His research interests include logistics engineering and ship motion control.

QING WU was born in Anhua, Hunan, China, in 1962. She received the M.S. degree in mechanical engineering from the Wuhan University of Technology, Wuhan, China, in 1997. Since 2005, she has been a Professor and a Ph.D. Supervisor with the Department of Logistics Automation, School of Logistics Engineering, Wuhan University of Technology. She holds and participates in more than 30 projects. Her research interests include water traffic safety and information.