The Journal of Navigation (2022), **75**:6 1389–1409
doi:10.1017/S0373463322000522

RESEARCH ARTICLE

Data-driven model predictive control of underactuated ships with unknown dynamics in confined waterways

Shijie Li,[1] Chengqi Xu,[1] and Jialun Liu[2,3]*

[1] School of Transportation and Logistics Engineering, Wuhan University of Technology, Wuhan 430063, P. R. China
[2] Intelligent Transportation Systems Research Center, Wuhan University of Technology, Wuhan 430063, P. R. China
[3] National Engineering Research Center for Water Transport Safety, Wuhan 430063, P. R. China.
*Corresponding author. E-mail: jialunliu@whut.edu.cn

Received: 15 December 2021; **Accepted:** 9 September 2022; **First published online:** 5 October 2022

Keywords: underactuated ships; path-following; iterative learning; data-driven control

Abstract
Inland waterway transportation is one of the most important means to transport cargo in rivers and canals. To facilitate autonomous navigation for ships in inland waterways, this paper proposes a data-driven approach for predictions and control of underactuated ships with unknown dynamics, which integrates model predictive control (MPC) with an iterative learning control (ILC) scheme. In each iteration, kernel-based linear regressors are used to identify the relations between the evolution of ship states and control inputs based on the stored data from previous iterations and the collected data during operation, so as to build the system prediction model. The data are dynamically used to fix the prediction model over iterations, as well as to improve the controller performance until it converges. The proposed approach does not require prior knowledge regarding the hydrodynamic coefficients and ship parameters, but learns from the data instead. In addition, it exploits the advantages of MPC in handling constraints with minimised overall cost. Simulation results show that the controller could start from a nominal, linear data-driven ship model and then learn to reduce the path-following errors based on the data obtained over iterations.

1. Introduction

With the trends towards autonomous shipping, advanced ship motion control techniques are being developed to ensure that ships can independently control their actions, especially in complicated situations. When a ship is navigated in a confined waterway with limited width, it needs to maintain a sufficient distance from the banks to ensure safety. This brings in the problem of path-following, in which the control design is to calculate the control inputs to drive the ship to a reference path. Most cargo ships are underactuated with propellers and rudders for surge and yaw motions and without other actuators for the sway motion, which are coupled with the nonlinear ship hydrodynamic characteristics. Various control methods have been proposed for solving the path-following problem of ships, such as sliding-mode control (Zhang et al., 2022), adaptive control (Culverhouse et al., 2015), observer-based control (Liu et al., 2019), model predictive control (Liang et al., 2021), etc. These methods require an ideal and accurate mathematical model of the ship. Inland ships face speed loss and decline of manoeuvrability when sailing in a confined waterway, according to experienced seafarers, which increases the risk of collision with the bank (Du et al., 2020). To investigate the ship–waterway interactions, a numerical model has been developed by Du et al. (2017) to predict ship manoeuvring in a confined waterway using a nonlinear model with optimisation techniques to accurately identify the hydrodynamic coefficients involved. Finding a prior ship model for such effects is challenging as the ship would be affected by uncertainties and disturbances induced by the environment.

Over the last decade, several data-driven control methods have been proposed, which may alleviate the efforts spent in identifying and modelling all disturbances that a model-based controller may need to take into account. Weng and Wang (2020) developed a data-driven robust back-stepping control approach for tracking of unmanned ships with uncertainties and unknown parametric dynamics, in which the requirement of model information, including system order and inertia matrix, could be completely removed. A data-driven performance-prescribed reinforcement learning control scheme was investigated by Wang et al. (2021) to deal with the trajectory tracking problem considering control optimality and prescribed tracking accuracy simultaneously. Gao et al. (2022) proposed a data-driven model-free resilient speed control method using available input and output data only with pulse-width-modulation inputs. Wang et al. (2022) proposed an antenna mutation beetle swarm predictive reinforcement learning algorithm to address the path-following problem of underactuated ships using the input and output data retrieved from experiments.

A growing number of data-driven methods have been proposed and integrated with Model Predictive Control (MPC), which uses system identification techniques and stored data to build the prediction model, and the data collected during operation can be dynamically incorporated in MPC controller design and be used to fix the system prediction model (Hewing et al., 2020). MPC enables optimal control inputs while guaranteeing constraints satisfaction, and data-driven modelling enhances performances and adapts to system changes. MPC requires a system model that can capture the characteristics of its dynamics while not being too complicated to be incorporated in an online optimisation framework (Kabzan et al., 2019). For a system that involves complex high-order nonlinear terms, applying MPC directly on a nonlinear system model may be computationally expensive, whereas an over-simplified system may result in reduced performance and control accuracy. In this paper, a system identification method is proposed to build a linear, data-driven ship model based on the data gathered over time, which enables accurate approximations of the true ship dynamics.

Iterative Learning Control (ILC) is an effective strategy in handling repeated control processes, due to its structural simplicity and effective learning ability (Jin, 2016). By taking advantage of the repetitive nature in the learning process, ILC algorithms can improve the control performance gradually with the increase of iterations. By combining the iterative learning control scheme and MPC, in each iteration, the controller could learn from stored data in previous iterations and improve its closed-loop tracking performance and modelling accuracy.

This paper proposes a data-driven approach for predictions and control of underactuated ships with unknown dynamics in confined waterways via integrating MPC with an ILC scheme. The stored datasets form the basis of the learning procedure, which consist of ship states and control sequences that could successfully steer the ship to complete path-following tasks while satisfying all required constraints, upon which regression techniques are used to approximate the unknown ship dynamics as a linear state-space model. This alleviates the need for accurate ship dynamics and parameter-specific observers, while maintaining the advantages of MPC including predictive behaviour and constraints satisfaction. Based on the identified linear prediction model, a learning-based MPC controller is designed.

The main contribution of this paper is threefold.

- A learning-based MPC strategy is proposed, which makes use of the stored data from previous iterations and the collected data during operation to improve the performance of the controller with an ILC scheme.
- A system identification strategy is proposed to build a linear state-space prediction model that approximates unknown ship dynamics, which uses kernel-based linear regressors to minimise the error between the predicted evolution of states and true states.
- A confined waterway is divided into a series of straight-line and curved lines, a line-of-sight guidance law is used as the guidance principle for straight-line segments, and a curvilinear reference frame is introduced, as the guidance principles for curved segments.

The rest of the paper is organised as follows. Section 2 gives the nonlinear ship dynamics. Section 3 introduces the guidance principles. Section 4 describes the proposed learning scheme and controller

https://doi.org/10.1017/S0373463322000522 Published online by Cambridge University Press

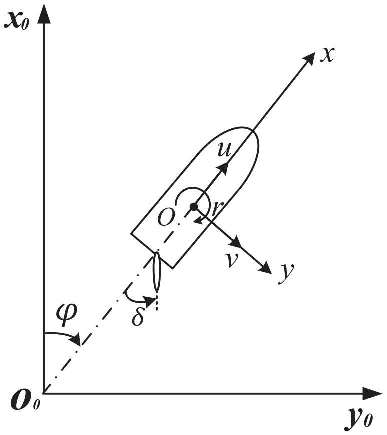

Figure 1. *Ship coordinate system.*

design steps. Simulation results are presented in Section 5. Conclusions and future work are given in Section 6.

2. Nonlinear ship dynamics

A 3-DOF (degree of freedom) model is used to represent the ship dynamics on the surge, sway and yaw axes with the MMG (Manoeuvring Modelling Group) form, in which the hydrodynamic forces and moments on the ship are divided into hull, rudder and propeller, expressed in the following form (SNAME, 1950):

$$\begin{cases} (m+m_x)\dot{u} - (m+m_y)vr - x_G m r^2 = X_H + X_P + X_R \\ (m+m_y)\dot{v} + (m+m_x)ur + x_G m\dot{r} = Y_H + Y_P + Y_R \\ (I_z + x_G^2 m + J_z)\dot{r} + x_G m(\dot{v} + ur) = N_H + N_P + N_R \end{cases} \tag{2.1}$$

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 moments of inertia and added moment of inertia around the z-axis.

Figure 1 shows the ship coordinate system used in this paper: the earth-fixed coordinate system O_0—$x_0y_0z_0$ and the ship-fixed coordinate system o—xyz, where o locates on the midship of the ship, with x, y and z axes that point towards the bow, towards the starboard and vertically downwards, respectively. Variables u and v are ship longitudinal and lateral speeds. Course angle ψ is defined as the angle between the x_0 and x axes, δ represents the rudder angle and r represents the yaw rate. The evolution of ship states is usually expressed in the following way (Fossen, 2011):

$$\begin{cases} \dot{x} = u\cos\psi - v\sin\psi \\ \dot{y} = u\sin\psi + v\cos\psi \\ \dot{\psi} = r \\ \dot{u} = \dfrac{m_v}{m_u}vr - \dfrac{f_u(\boldsymbol{v})}{m_u} + \dfrac{T_u(\cdot)}{m_u}|n|n + d_{wu} \\ \dot{v} = -\dfrac{m_u}{m_v}ur - \dfrac{f_v(\boldsymbol{v})}{m_v} + d_{wv} \\ \dot{r} = \dfrac{(m_u - m_v)}{m_r}uv - \dfrac{f_r(\boldsymbol{v})}{m_r} + \dfrac{F_r(\cdot)}{m_r}\delta + d_{wr} \end{cases} \tag{2.2}$$

https://doi.org/10.1017/S0373463322000522 Published online by Cambridge University Press

Parameters d_{wu}, d_{wv} and d_{wr} represent the disturbances on the x, y and z axes. 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 rate n and rudder angle δ, respectively. It is also noted that there is no direct control force on its lateral movement on the y axis, which makes the ship an underactuated system. In practice, a ship rarely changes its propeller revolution rate when it is sailing. Therefore, this paper assumes that control input n and force $T_u(\cdot)$ are constants and considers the rudder angle δ as the main control input, as it is the case in actual ship manoeuvring operations. The yaw moment F_r is a nonlinear term, and together with the high-order fluid dynamics items $f_u(\boldsymbol{v})$, $f_v(\boldsymbol{v})$ and $f_r(\boldsymbol{v})$, add up to the nonlinear characteristics of ship manoeuvring.

In Equation (2.2), parameters m_u, m_v and m_r are calculated as follows:

$$\begin{cases} m_u = m - X_{\dot{u}} = m + m_{xx} \\ m_v = m - Y_{\dot{v}} = m + m_{yy} \\ m_r = I_{zz} - N_{\dot{r}} = I_{zz} + J_{zz} \end{cases} \tag{2.3}$$

Variables $f_u(\boldsymbol{v})$, $f_v(\boldsymbol{v})$ and $f_r(\boldsymbol{v})$ represent the high-order fluid dynamics items, which are defined as

$$\begin{cases} f_u(\boldsymbol{v}) = -(X_{|u|u}|u|u + X_{vr}vr + X_{vv}v^2 + X_{rr}r^2) \\ \qquad\quad = -\frac{1}{2}\rho L_{pp}TV^2(-R_0' + X_{vv}'v'^2 + X_{vr}'v'r' + X_{rr}'r'^2 + X_{vvvv}'v'^4) \\ f_v(\boldsymbol{v}) = -(Y_v v + Y_r r + Y_{|v|v}|v|v + Y_{|r|r}|r|r + Y_{wr}v^2 r + Y_{vrr}vr^2) \\ \qquad\quad = -\frac{1}{2}\rho L_{pp}TV^2(Y_v'v' + Y_r'r' + Y_{vvv}'v'^3 + Y_{vvr}'v'^2r' + Y_{vrr}'v'r'^2 + Y_{rrr}'r'^3) \\ f_r(\boldsymbol{v}) = -(N_v v + N_r r + N_{|v|v}|v|v + N_{|r|r|}|r|r + N_{wr}v^2 r + N_{vrv}v^2) \\ \qquad\quad = -\frac{1}{2}\rho L_{pp}^2TV^2(N_v'v' + N_r'r' + N_{vvv}'v'^3 + N_{vvr}'v'^2r' + N_{vrr}'v'r'^2 + N_{rrr}'r'^3) \end{cases} \tag{2.4}$$

in which

$$\begin{cases} u' = u/V \\ v' = v/V \\ r' = rL_{PP}/V \\ V = \sqrt{u^2 + v^2} \end{cases} \tag{2.5}$$

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

$$T_u(\cdot) = (1 - t_P)\rho D_P^4 K_T \tag{2.6}$$

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

$$K_T = k_2 J_P^2 + k_1 J_P + k_0 \tag{2.7}$$

in which J_P can be obtained as

$$J_P = \frac{u(1 - w_P)}{nD_P} \tag{2.8}$$

In Equation (2.8), w_P is the wake factor at the propeller position in manoeuvring. It is commonly estimated based on the wake factor at the propeller position in straight moving w_{P_0} and the geometrical inflow angle to the propeller in manoeuvring β_P, defined as

$$\beta_P = \beta - x_P' r' \tag{2.9}$$

where $\beta = \arctan(-v/u)$, $x_P' = x_P/L_{pp} = -0{\cdot}48$ and x_P is the longitudinal portion of the propeller.

Parameter w_P, introduced as

$$\frac{(1-w_P)}{(1-w_{P_0})} = 1 + \{1 - \exp(-C_1|\beta_P|)\}(C_2 - 1) \tag{2.10}$$

where w_{P_0} is the wake factor at the propeller position in straight moving, and 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

$$F_r(\cdot) = (x_R + a_H x_H)\left[-\frac{6{\cdot}13\lambda}{\lambda + 2{\cdot}25}\frac{A_R}{L_{PP}^2}(u_R^2 + v_R^2)\cos\delta\right] \tag{2.11}$$

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

$$u_R = u\varepsilon(1-w_P)\sqrt{\eta\left\{1 + \kappa\left[\sqrt{\left(1 + \frac{8K_T}{\pi J^2}\right)} - 1\right]\right\}^2 + (1-\eta)} \tag{2.12}$$

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

$$v_R = V\gamma_R(\beta - \ell'_R r') \tag{2.13}$$

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 the rudder position. We refer the readers to Yasukawa and Yoshimura (2014) for more details on the nonlinear ship manoeuvrability model.

As can be seen, the evolution of states u, v and r in the nonlinear ship model in Equation (2.2) are dependent on the hydrodynamic coefficients and ship parameters. These coefficients and parameters are usually identified via computational fluid dynamics analysis and tank tests, which takes time and much effort. Meanwhile, the manoeuvrability of a ship changes when it is sailing with different speed and rudder angle alterations, which makes the identification procedure a complicated task. Therefore, this paper proposes to learn the ship dynamics from data in a linear state-space form, which will be further explained in Section 4.

3. The guidance principles for straight-line and curved waterways

A confined waterway usually consists of straight-line and curved segments. This paper divides a waterway into a set of segments S. When the ship is sailing within a straight-line segment, a line-of-sight guidance principle is used generate reference course angle. For curved segments, a curvilinear reference frame is introduced.

3.1. Line-of-sight guidance law

Line-of-sight (LOS) is a conventional guidance principle, and its main idea is that if a ship is able to keep its course angle aligned with the so-called LOS angle, then the convergence to the desired position is also achieved. The LOS scheme was first applied to surface ships by Fossen et al. (2003). Researchers found it is useful when considering the control of underactuated ships, since it renders possible an

https://doi.org/10.1017/S0373463322000522 Published online by Cambridge University Press

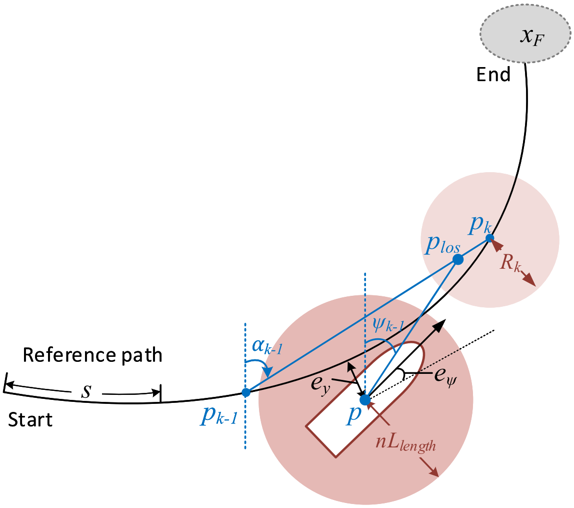

Figure 2. *Illustration of the guidance principles.*

approach of reducing the desired reference from x_d, y_d, ψ_d to only one reference ψ_d. In this way, the path-following task $\psi \rightarrow \psi_d$ is achieved using only the one control input δ.

The desired reference path is composed of a set of way-points. As shown in Figure 2, if the ship's current position is $p = [x, y]$, the LOS position p_{los} is located between the previous p_{k-1} and current p_k way-points. Let the ship's current horizontal position p be the centre of a circle with the radius of n times the ship length L_{length}. This circle then intersects the current straight-line segment at two points where p_{los} is selected as the point closest to the next way-point.

To calculate $p_{\text{los}} = [x_{\text{los}}, y_{\text{los}}]$, the following two equations need to be solved:

$$(y_{\text{los}} - y)^2 + (x_{\text{los}} - x)^2 = (nL_{\text{length}})^2 \tag{3.1}$$

$$\frac{y_{\text{los}} - y_{k-1}}{x_{\text{los}} - x_{k-1}} = \frac{y - y_{k-1}}{x - x_{k-1}} + \tan(\alpha_{k-1}) \tag{3.2}$$

Selecting way-points in the way-point table relies on a switching algorithm. A criteria for selecting the next way-point, located at $p_{k+1} = [x_{k+1}, y_{k+1}]^\top$, is for the ship to be within the range of acceptance of the current p_k, as shown in Figure 2. If at time t, the ship's current position satisfies

$$(x_k - x(t))^2 + (y_k - y(t))^2 \leq R_k^2 \tag{3.3}$$

then the next way-point will be selected from the way-point set.

With the LOS position p_{los}, the LOS angle can be computed:

$$\psi_{\text{los}} = \arctan 2\left(\frac{y_{\text{los}} - y}{x_{\text{los}} - x}\right) \tag{3.4}$$

in which the four quadrant inverse tangent function $\arctan 2(y, x)$ is used to ensure that $\psi_{\text{los}} \in [-\pi, +\pi]$.

The ship state vector $\boldsymbol{x} = [u, v, r, x, y, \psi]^\top \in \mathcal{R}^6$. Then ship kinematics can be discretised as follows:

$$\begin{cases} \psi_{k+1}(\boldsymbol{x}) = \psi_k + r_k \Delta t \\ x_{k+1}(\boldsymbol{x}) = x_k + (u_k \cos\psi_k - v_k \sin\psi_k)\Delta t \\ y_{k+1}(\boldsymbol{x}) = y_k + (u_k \sin\psi_k + v_k \cos\psi_k)\Delta t \end{cases} \tag{3.5}$$

https://doi.org/10.1017/S0373463322000522 Published online by Cambridge University Press

where Δt refers to the discretisation time. We can rewrite it with a linear state-space form:

$$\begin{bmatrix}\psi_{k+1}\\ x_{k+1}\\ y_{k+1}\end{bmatrix} = \begin{bmatrix}(\nabla_{\boldsymbol{x}}\psi_{k+1}(\boldsymbol{x}))^\top\\ (\nabla_{\boldsymbol{x}}x_{k+1}(\boldsymbol{x}))^\top\\ (\nabla_{\boldsymbol{x}}y_{k+1}(\boldsymbol{x}))^\top\end{bmatrix}\begin{bmatrix}\psi_k\\ x_k\\ y_k\end{bmatrix} + \begin{bmatrix}0 & 0\\ 0 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}n_k\\ \delta_k\end{bmatrix} + \begin{bmatrix}\psi_{k+1}(\boldsymbol{x}) - (\nabla_{\boldsymbol{x}}\psi_{k+1}(\boldsymbol{x}))^\top\psi_k\\ x_{k+1}(\boldsymbol{x}) - (\nabla_{\boldsymbol{x}}x_{k+1}(\boldsymbol{x}))^\top x_k\\ y_{k+1}(\boldsymbol{x}) - (\nabla_{\boldsymbol{x}}y_{k+1}(\boldsymbol{x}))^\top y_k\end{bmatrix}. \tag{3.6}$$

3.2. *Curvilinear reference frame*

Curvature $C(s)$ is introduced to describe the curved features of the waterway segment $s \in S$, which is a known parameter that can be drawn from the map information of a waterway. As shown in Figure 2, the control task is to ensure $e_y \longrightarrow 0, e_\psi \longrightarrow 0$ so that the ship can follow the reference path. The evolution of ship states e_ψ, s, e_y are expressed as follows:

$$\begin{cases}\dot{e}_\psi = r - \dfrac{u\cos(e_\psi) - v\sin(e_\psi)}{1 - C(s)e_y}C(s)\\ \dot{s} = \dfrac{u\cos(e_\psi) - v\sin(e_\psi)}{1 - C(s)e_y}\\ \dot{e}_y = u\sin(e_\psi) + v\cos(e_\psi)\end{cases} \tag{3.7}$$

Therefore, the ship state vector $\boldsymbol{x} = [u, v, r, e_\psi, e_y, s]^\top \in \mathcal{R}^6$. Then, the ship kinematic model in Equation (3.7) is discretised as follows:

$$\begin{cases}e_{\psi_{k+1}}(\boldsymbol{x}) = e_{\psi_k} + \left(r_k - \dfrac{u_k\cos(e_{\psi_k}) - v_k\sin(e_{\psi_k})}{1 - C(s_k)e_{y_k}}C(s_k)\right)\Delta t\\ s_{k+1}(\boldsymbol{x}) = s_k + \left(\dfrac{u_k\cos(e_{\psi_k}) - v_k\sin(e_{\psi_k})}{1 - C(s_k)e_{y_k}}\right)\Delta t\\ e_{y_{k+1}}(\boldsymbol{x}) = e_{y_k} + \left(u_k\sin(e_{\psi_k}) + v_k\cos(e_{\psi_k})\right)\Delta t\end{cases} \tag{3.8}$$

Similar to Equation (3.6), it can be reformulate as follows:

$$\begin{bmatrix}e_{\psi_{k+1}}\\ s_{k+1}\\ e_{y_{k+1}}\end{bmatrix} = \begin{bmatrix}(\nabla_{\boldsymbol{x}}e_{\psi_{k+1}}(\boldsymbol{x}))^\top\\ (\nabla_{\boldsymbol{x}}s_{k+1}(\boldsymbol{x}))^\top\\ (\nabla_{\boldsymbol{x}}e_{y_{k+1}}(\boldsymbol{x}))^\top\end{bmatrix}\begin{bmatrix}e_{\psi_k}\\ s_k\\ e_{y_k}\end{bmatrix} + \begin{bmatrix}0 & 0\\ 0 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}n_k\\ \delta_k\end{bmatrix} + \begin{bmatrix}e_{\psi_{k+1}}(\boldsymbol{x}) - (\nabla_{\boldsymbol{x}}e_{\psi_{k+1}}(\boldsymbol{x}))^\top e_{\psi_k}\\ s_{k+1}(\boldsymbol{x}) - (\nabla_{\boldsymbol{x}}s_{k+1}(\boldsymbol{x}))^\top s_k\\ e_{y_{k+1}}(\boldsymbol{x}) - (\nabla_{\boldsymbol{x}}e_{y_{k+1}}(\boldsymbol{x}))^\top e_{y_k}\end{bmatrix}. \tag{3.9}$$

In this paper, 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.

According to Section 2, the dynamic equations of ship motion variables u, v and r in Equation (2.2) include many nonlinear high-order terms. To deal with the nonlinearity, this paper proposes to learn a linear model around states (u, v, r), in which regression vectors $\boldsymbol{\Gamma}^l \in \mathcal{R}^5$ $(l = \{u, v, r\})$ are introduced. Based on the values of the regression vectors, the prediction model can be reformulated as follows:

$$\begin{bmatrix}u_{k+1}\\ v_{k+1}\\ r_{k+1}\end{bmatrix} = \begin{bmatrix}\boldsymbol{\Gamma}^u_{1:3}(\boldsymbol{x})\\ \boldsymbol{\Gamma}^v_{1:3}(\boldsymbol{x})\\ \boldsymbol{\Gamma}^r_{1:3}(\boldsymbol{x})\end{bmatrix}\begin{bmatrix}u_k\\ v_k\\ r_k\end{bmatrix} + \begin{bmatrix}\boldsymbol{\Gamma}^u_4(\boldsymbol{x}) & 0\\ 0 & \boldsymbol{\Gamma}^v_4(\boldsymbol{x})\\ 0 & \boldsymbol{\Gamma}^r_4(\boldsymbol{x})\end{bmatrix}\begin{bmatrix}n_k\\ \delta_k\end{bmatrix} + \begin{bmatrix}\boldsymbol{\Gamma}^u_5(\boldsymbol{x})\\ \boldsymbol{\Gamma}^v_5(\boldsymbol{x})\\ \boldsymbol{\Gamma}^r_5(\boldsymbol{x})\end{bmatrix}, \tag{3.10}$$

where $\mathbf{\Gamma}_i^l(\boldsymbol{x})$ denotes the ith element of vector $\mathbf{\Gamma}^l(\boldsymbol{x})$. Linear regression methods are used to identify $\mathbf{\Gamma}$, which will be introduced in Section 4.

Combining Equation (3.6) or (3.9) with Equation (3.10), the 6-states ship prediction model can be restructured in the form of $\boldsymbol{x}_{k+1} = A\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k + C$. As can be seen, the evolution of states (ψ, x, y) or (e_ψ, s, e_y) are derived from ship kinematic characteristics and independent of ship parameters. The values of $\mathbf{\Gamma}$ are determined via regression methods based on the data collected from the ship. In other words, the formulation in Equation (3.10) does not require pre-knowledge on ship parameters and hydrodynamic coefficients. This makes it possible to construct a prediction model for unknown ship dynamics.

4. Iterative learning scheme and controller design

Figure 3 illustrates the iterative learning scheme and controller design steps of the proposed method. First, a model-free control law is applied to the ship and steer the ship from the start to finish point of a confined waterway for several loops while the ship states and the control inputs are stored. Each loop is referred to as one iteration. In this paper, we choose a proportional-integral-derivative (PID) controller as the nominal controller. In other words, each iteration finishes a successful ship track around the waterway. Based on the stored dataset X_{PID}, a linear time-varying (LTV) prediction model is constructed using a system identification strategy. Then a rolling horizon optimisation problem is formulated with the LTV model, and an MPC controller is designed. Similarly, another dataset is collected after applying the MPC controller to the ship for several loops (iterations), referred as X_{MPC}. Datasets X_{PID} and X_{MPC} are then employed to build the initial prediction model of the proposed Learning-based MPC (LMPC) controller. Together with the data obtained from LMPC running during iterations, the prediction model is improved with an iterative learning scheme. In each iteration, the data collected from previous iterations are used to construct a sampled safe set, terminal constraint set, and cost function, which are exploited in the LMPC controller design. In LMPC, the control problem of path-following is formulated as a repetitive task, and the controller uses the data from previous iterations to enhance the controller performance.

4.1. *Kernel-based linear regression of prediction model*

To facilitate a data-driven modelling of unknown ship dynamics, this paper estimates the evolution of u, v and r via linear regression instead of identifying the ship parameters first and then linearising them. It learns a linear affine time-varying ship model around each point so as to construct the prediction model in the MPC controller.

For the unknown ship dynamics, a PID controller is designed and applied to the ship, and steering the ship from the start point $\boldsymbol{x}_S$ to the finish point of the waterway for M_1 loops:

$$\delta = K_p\boldsymbol{e} + K_d\dot{\boldsymbol{e}} + K_i\int \boldsymbol{e}\,\mathrm{d}t \tag{4.1}$$

where δ represents the rudder angle and $\boldsymbol{e} = \{e_y, e_\psi\}$. After running M_1 loops (iterations), the ship states and control inputs are stored in dataset X_{PID} with vectors $\boldsymbol{x}^i = [\boldsymbol{x}_0^i, \ldots, \boldsymbol{x}_{T^i}^i]$ and $\boldsymbol{u}^i = [\boldsymbol{u}_0^i, \ldots, \boldsymbol{u}_{T^i}^i]$, in which T^i denotes the time when the ship reaches the finish point, and $\boldsymbol{x}_{T^i} \in \mathcal{X}_F$. Here, $\mathcal{X}_F$ is a set of points beyond the finish point, and $\mathcal{X}_F = \{\boldsymbol{x} \in \mathcal{R}^6 : [0\,0\,0\,0\,1\,0]\cdot\boldsymbol{x} = s \geq L\}$, in which L is the length of the waterway. Here, $s \geq L$ means that the ship has passed the finish point of the waterway.

Based on dataset X_{PID}, a set time of indices $I_i(\boldsymbol{x})$ of P nearest neighbours of point $\boldsymbol{x}$ at iteration i are identified and defined as

$$I_i(\boldsymbol{x}) = [k_1^{i*}, \ldots, k_P^{i*}] = \arg\min_{k_1,\ldots,k_P} \sum_{m=1}^{P} \|\boldsymbol{x} - \boldsymbol{x}_{k_m}^i\|_Q^2 \tag{4.2}$$

https://doi.org/10.1017/S0373463322000522 Published online by Cambridge University Press

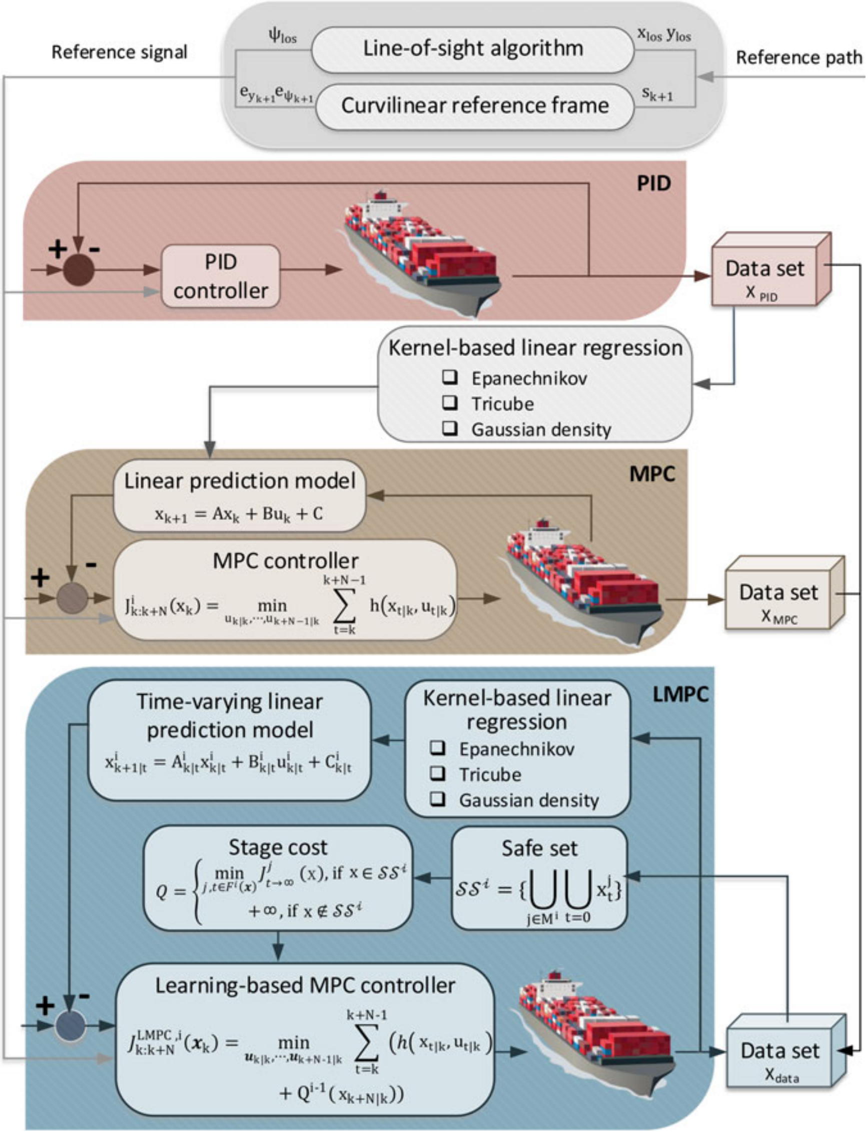
Reference signal
ψ_{los}
Line-of-sight algorithm
x_{los} y_{los}
Reference path
$e_{y_{k+1}}e_{\psi_{k+1}}$
Curvilinear reference frame
s_{k+1}
PID
+
-
PID controller
Data set X_{PID}
Kernel-based linear regression
❑ Epanechnikov
❑ Tricube
❑ Gaussian density
MPC
Linear prediction model
$x_{k+1} = Ax_k + Bu_k + C$
MPC controller
$J^i_{k:k+N}(x_k) = \min_{u_{k|k},\cdots,u_{k+N-1|k}} \sum_{t=k}^{k+N-1} h(x_{t|k}, u_{t|k})$
Data set X_{MPC}
LMPC
Time-varying linear prediction model
$x^i_{k+1|t} = A^i_{k|t}x^i_{k|t} + B^i_{k|t}u^i_{k|t} + C^i_{k|t}$
Kernel-based linear regression
❑ Epanechnikov
❑ Tricube
❑ Gaussian density
Stage cost
$Q = \begin{cases} \min_{j,t\in F^i(x)} J^j_{t\to\infty}(x), \text{if } x \in \mathcal{SS}^i \\ +\infty, \text{if } x \notin \mathcal{SS}^i \end{cases}$
Safe set
$\mathcal{SS}^i = \{\bigcup_{j\in M^i}\bigcup_{t=0} x^j_t\}$
Learning-based MPC controller
$J^{LMPC,i}_{k:k+N}(x_k) = \min_{u_{k|k},\cdots,u_{k+N-1|k}} \sum_{t=k}^{k+N-1} (h(x_{t|k}, u_{t|k}) + Q^{i-1}(x_{k+N|k}))$
Data set X_{data}

Figure 3. *Iterative learning scheme of data-driven MPC.*

in which $k_m \in \{0, \ldots, T^i\}$, $m \in \{1, \ldots, P\}$, and T^i refers to the ship sailing time spent in iteration i. Additionally, Q is a scaling matrix of different variables. In other words, the set I_i include the associated indices of the neighbours of each state point $\boldsymbol{x}$. Based on these data with index I, three kernel functions are introduced as the linear regressors, including Epanechnikov, Tri-cube and Gaussian density kernel functions:

$$K^{\text{Epanechnikov}}(z) = \begin{cases} \frac{3}{4}(1 - z^2) & \forall |z| < 1 \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

$$K^{\text{Tri-cube}}(z) = \begin{cases} (1 - |z|^3)^3 & \forall |z| < 1 \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

$$K^{\text{Gaussian}}(z) = \begin{cases} e^{-(1/2)z^2} & \forall |z| < 1 \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

https://doi.org/10.1017/S0373463322000522 Published online by Cambridge University Press

where $z = \|\boldsymbol{x} - \boldsymbol{x}^i_{k_m}\|^2_Q / h$, and h is a hyperparameter that represents the bandwidth.

To find $\boldsymbol{\Gamma}^u(\boldsymbol{x}), \boldsymbol{\Gamma}^v(\boldsymbol{x}), \boldsymbol{\Gamma}^r(\boldsymbol{x}) \in \mathcal{R}^5$ in Equation (3.10) for u, v and r, the following three optimisation functions are formulated:

$$\begin{cases} J_u = \min \displaystyle\sum_{k,i \in I(x)} K\left(\frac{\|\boldsymbol{x} - \boldsymbol{x}^i_{k_m}\|^2_Q}{h}\right) \boldsymbol{y}^{i,u}(\boldsymbol{\Gamma}^u) \\ J_v = \min \displaystyle\sum_{k,i \in I(x)} K\left(\frac{\|\boldsymbol{x} - \boldsymbol{x}^i_{k_m}\|^2_Q}{h}\right) \boldsymbol{y}^{i,v}(\boldsymbol{\Gamma}^v) \\ J_r = \min \displaystyle\sum_{k,i \in I(x)} K\left(\frac{\|\boldsymbol{x} - \boldsymbol{x}^i_{k_m}\|^2_Q}{h}\right) \boldsymbol{y}^{i,r}(\boldsymbol{\Gamma}^r) \end{cases} \tag{4.6}$$

in which

$$\begin{cases} \boldsymbol{y}^{i,u}(\boldsymbol{\Gamma}^u) = \|u^i_{k+1} - \boldsymbol{\Gamma}^u(\boldsymbol{x}) \cdot [u^i_k, v^i_k, r^i_k, n^i_k, 1]^\top\| \\ \boldsymbol{y}^{i,v}(\boldsymbol{\Gamma}^v) = \|v^i_{k+1} - \boldsymbol{\Gamma}^v(\boldsymbol{x}) \cdot [u^i_k, v^i_k, r^i_k, \delta^i_k, 1]^\top\| \\ \boldsymbol{y}^{i,r}(\boldsymbol{\Gamma}^r) = \|r^i_{k+1} - \boldsymbol{\Gamma}^r(\boldsymbol{x}) \cdot [u^i_k, v^i_k, r^i_k, \delta^i_k, 1]^\top\| \end{cases} \tag{4.7}$$

Problems J_u, J_v and J_r form quadratic programming problems, which can be easily solved via an optimisation solver. Here, $\boldsymbol{x}^i_k$ represents the stored ship states data in iteration i at time k. Solutions to problems J_u, J_v and J_r are $\boldsymbol{\Gamma}^u(\boldsymbol{x}) = \arg_{\boldsymbol{\Gamma}} \min J_u, \boldsymbol{\Gamma}^v(\boldsymbol{x}) = \arg_{\boldsymbol{\Gamma}} \min J_v$ and $\boldsymbol{\Gamma}^r(\boldsymbol{x}) = \arg_{\boldsymbol{\Gamma}} \min J_r$, respectively.

Based on the identified model in Equation (3.10), a linear time-varying prediction model can be constructed at time t of iteration i as follows:

$$\boldsymbol{x}^i_{k+1|t} = A^i_{k|t} \boldsymbol{x}^i_{k|t} + B^i_{k|t} \boldsymbol{u}^i_{k|t} + C^i_{k|t}, k \in \{t, \ldots, t+N-1\} \tag{4.8}$$

in which it is noted that $\boldsymbol{x}^i_{k|t} = [u^i_{k|t}, v^i_{k|t}, r^i_{k|t}, \psi^i_{k|t}, y^i_{k|t}, x^i_{k|t}]$ if it is a straight-line waterway segment and that $\boldsymbol{x}^i_{k|t} = [u^i_{k|t}, v^i_{k|t}, r^i_{k|t}, e^i_{\psi_{k|t}}, e^i_{y_{k|t}}, s^i_{k|t}]$ if it is a curved waterway segment. The matrices $A^i_{k|t}$, $B^i_{k|t}$ and $C^i_{k|t}$ are calculated as follows:

$$\begin{gathered} A^i_{k|t} = \begin{bmatrix} \boldsymbol{\Gamma}^u_{1:3}(\bar{\boldsymbol{x}}^i_{k|t}) & 0 & 0 & 0 \\ \boldsymbol{\Gamma}^v_{1:3}(\bar{\boldsymbol{x}}^i_{k|t}) & 0 & 0 & 0 \\ \boldsymbol{\Gamma}^r_{1:3}(\bar{\boldsymbol{x}}^i_{k|t}) & 0 & 0 & 0 \\ \multicolumn{4}{c}{(\nabla_{\boldsymbol{x}} e_{\psi_{k-1}} / \psi_{k-1}(\boldsymbol{x})|_{\bar{\boldsymbol{x}}^i_{k|t}})^\top} \\ \multicolumn{4}{c}{(\nabla_{\boldsymbol{x}} e_{y_{k-1}} / y_{k-1}(\boldsymbol{x})|_{\bar{\boldsymbol{x}}^i_{k|t}})^\top} \\ \multicolumn{4}{c}{(\nabla_{\boldsymbol{x}} s_{k-1} / x_{k-1}(\boldsymbol{x})|_{\bar{\boldsymbol{x}}^i_{k|t}})^\top} \end{bmatrix} \quad B^i_{k|t} = \begin{bmatrix} \boldsymbol{\Gamma}^u_4(\bar{\boldsymbol{x}}^i_{k|t}) & 0 \\ 0 & \boldsymbol{\Gamma}^v_4(\bar{\boldsymbol{x}}^i_{k|t}) \\ 0 & \boldsymbol{\Gamma}^r_4(\bar{\boldsymbol{x}}^i_{k|t}) \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \\ C^i_{k|t} = \begin{bmatrix} \boldsymbol{\Gamma}^u_5(\bar{\boldsymbol{x}}^i_{k|t}) \\ \boldsymbol{\Gamma}^v_5(\bar{\boldsymbol{x}}^i_{k|t}) \\ \boldsymbol{\Gamma}^r_5(\bar{\boldsymbol{x}}^i_{k|t}) \\ e_{\psi_{k-1}} / \psi_{k-1}(\bar{\boldsymbol{x}}^i_{k|t}) - (\nabla_{\boldsymbol{x}} e_{\psi_{k-1}} / \psi_{k-1}(\boldsymbol{x})|_{\bar{\boldsymbol{x}}^i_{k|t}})^\top \bar{\boldsymbol{x}}^i_{k|t} \\ e_{y_{k-1}} / y_{k-1}(\bar{\boldsymbol{x}}^i_{k|t}) - (\nabla_{\boldsymbol{x}} e_{y_{k-1}} / y_{k-1}(\boldsymbol{x})|_{\bar{\boldsymbol{x}}^i_{k|t}})^\top \bar{\boldsymbol{x}}^i_{k|t} \\ s_{k-1} / x_{k-1}(\bar{\boldsymbol{x}}^i_{k|t}) - (\nabla_{\boldsymbol{x}} s_{k-1} / x_{k-1}(\boldsymbol{x})|_{\bar{\boldsymbol{x}}^i_{k|t}})^\top \bar{\boldsymbol{x}}^i_{k|t} \end{bmatrix} \end{gathered} \tag{4.9}$$

where

$$\bar{\boldsymbol{x}}^i_{k|t} = \begin{cases} \bar{\boldsymbol{x}}^{i,*}_{k|t-1}, & k \in \{t, \ldots, t+N-1\} \\ z^i_t, & k = t+N \end{cases} \tag{4.10}$$

https://doi.org/10.1017/S0373463322000522 Published online by Cambridge University Press

Here, $\bar{\boldsymbol{x}}^i_{k|t} \in \bar{\boldsymbol{x}}^i_t = \{\bar{x}^i_{k|t}, \ldots, \bar{x}^i_{k+N|t}\}$ refers to a set of candidate solutions that are defined using the optimal solution from the previous time step $t-1$, and z^i_t represents one of the candidate terminal state of the planned ship trajectory at time t.

Then an MPC controller can be designed using Equation (4.8) as the prediction model. The control objective is to steer the ship from initial state $\boldsymbol{x}_S$ to the terminal set $\mathcal{X}_F$. In each sampling interval k, the MPC controller solves an infinite time horizon optimal control problem:

$$J^*_{0\to\infty}(\boldsymbol{x}_S) = \min_{\boldsymbol{u}_0, \boldsymbol{u}_1, \ldots} \sum_{k=0}^{\infty} h(\boldsymbol{x}_k, \boldsymbol{u}_k) \tag{4.11}$$

$$\text{such that} \quad \boldsymbol{x}_{k+1} = A\boldsymbol{x}_k + B\boldsymbol{u}_k \quad \forall k \geq 0 \tag{4.12}$$

$$\boldsymbol{x}_0 = \boldsymbol{x}_S \tag{4.13}$$

$$\boldsymbol{x}_k \in \mathcal{X} = \{\boldsymbol{x} \in \mathcal{R}^6 : F_x \leq b_x\} \quad \forall k \geq 0 \tag{4.14}$$

$$\boldsymbol{u}_k \in \mathcal{U} = \{\boldsymbol{u} \in \mathcal{R}^2 : F_u \leq b_u\} \quad \forall k \geq 0 \tag{4.15}$$

where Equation (4.12) represents the linearised ship prediction model, Equation (4.13) defines the initial ship states, Equations (4.14) and (4.15) represent input and state constraints. In Equation (4.11), it is assumed that $h(\cdot,\cdot) = \|\boldsymbol{x}_F - \boldsymbol{x}_k\|_Q + \|\boldsymbol{u}_k\|_R$, which refers to the stage cost. It is continuous and satisfies the following conditions in all iterations:

$$h(\boldsymbol{x}_F, 0) = 0, \quad h(\boldsymbol{x}_k, \boldsymbol{u}_k) \succ 0 \quad \forall \boldsymbol{x}_k \in \mathcal{R}^6 \setminus \{\boldsymbol{x}_F\} \quad \forall \boldsymbol{u}_k \in \mathcal{R}^2 \setminus \{0\} \tag{4.16}$$

After solving Equation (4.11), an optimal control sequence $\boldsymbol{U} = \{\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{N-1}\}$ with prediction horizon N at each sample time k, such that the resulting state sequence $\boldsymbol{X} = \{\boldsymbol{x}_0, \ldots, \boldsymbol{u}_N\}$ and the control sequence $\boldsymbol{U}$ are obtained without violating Constraints (4.12)–(4.15). This controller is then applied to the ship to run for M_2 loops to create another dataset X_{MPC}. Datasets X_{PID} and X_{MPC} form the basis for LMPC controller design.

4.2. *Learning-based MPC controller design*

In each iteration i, the controller selects N_{ss} points from the previous $i - N_{\text{s}}$ iteration to construct a sampled safe set. Here, N_{ss} and N_{s} are the control parameters to be determined. A sampled safe set $\mathcal{SS}^i$ at iteration i consists of all successful trajectories performed in the previous $i-1$ number of iterations, which is defined as

$$\mathcal{SS}^i = \left\{ \bigcup_{j\in M^i} \bigcup_{t=0}^{\infty} \boldsymbol{x}^j_t \right\}, \quad M^i \in \left\{ k \in [1, i] : \lim_{t\to\infty} \boldsymbol{x}^k_t = \boldsymbol{x}_F \right\} \tag{4.17}$$

Figure 4 gives the illustration of safe set $\mathcal{SS}^i$ in iteration i, which is the collection of all ship states at iteration j for $j \in M^i$, where M^i refers to the set of indexes k associated with successful iteration k for $k \leq i$. It is also noted that $M^j \subseteq M^i$ and $\mathcal{SS}^j \subseteq \mathcal{SS}^i$, $\forall j \leq i$. For every point in $\mathcal{SS}^i$, there exists a feasible control strategy which satisfies the state constraints and steers the state towards terminal state $\boldsymbol{x}_F$.

In addition, terminal cost and constraints are updated in each time step based on the planned ship trajectory in the previous time steps, so as to reduce computational burden. The LMPC solves a finite

https://doi.org/10.1017/S0373463322000522 Published online by Cambridge University Press

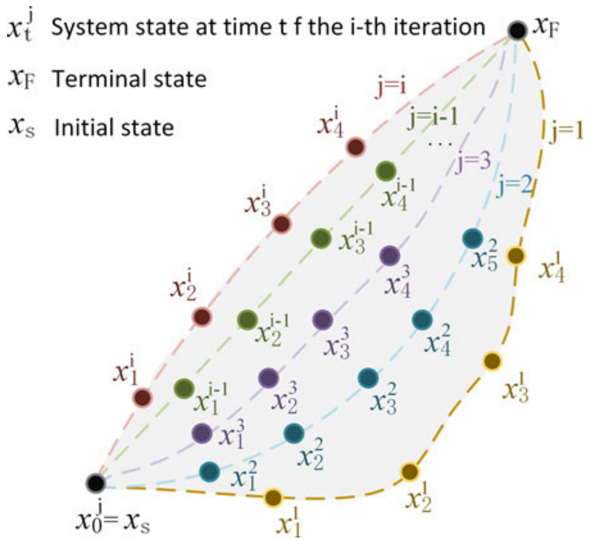

Figure 4. *Illustration of safe set* $\mathcal{SS}^i$.

time constrained optimal control problem at time k of iteration i:

$$J_{k:k+N}^{\mathrm{LMPC},i}(\boldsymbol{x}_k^i) = \min_{\boldsymbol{u}_{k|k},\ldots,\boldsymbol{u}_{k+N-1|k}} \left(\sum_{t=k}^{k+N-1} h(\boldsymbol{x}_{t|k}, \boldsymbol{u}_{t|k}) + Q^{i-1}(\boldsymbol{x}_{k+N|k}) \right) \tag{4.18}$$

$$\text{such that} \quad \boldsymbol{x}_{t+1|k} = A\boldsymbol{x}_{t|k} + B\boldsymbol{u}_{t|k} \quad \forall t \in [k, \ldots, k+N-1] \tag{4.19}$$

$$\boldsymbol{x}_{k|k} = \boldsymbol{x}_k^i \tag{4.20}$$

$$\boldsymbol{x}_{t|k} \in \mathcal{X}, \boldsymbol{u}_{t|k} \in \mathcal{U} \quad \forall t \in [k, \ldots, k+N-1] \tag{4.21}$$

$$\boldsymbol{x}_{k+N|k} \in \mathcal{SS}^{i-1} \tag{4.22}$$

where Equation (4.19) represents the linearised ship dynamics and Equation (4.20) defines the initial condition. The state and input constraints are defined via Equation (4.21). Equation (4.22) ensures that the terminal state reaches one of the points in the sampled safe set $\mathcal{SS}^{i-1}$ of the previous iteration $i-1$. Stage cost $h(\cdot,\cdot)$ is used to quantify the controller performance, which is the same as the stage cost in MPC formulation.

Function $Q^i(\cdot)$ is defined over $\mathcal{SS}^i$, which represents the learned minimum cost from previous iterations:

$$Q^i(\boldsymbol{x}) = \begin{cases} \min\limits_{j,t \in F^i(\boldsymbol{x})} J_{t\to\infty}^j(\boldsymbol{x}) \sum\limits_{k=t^*}^{\infty} h(\boldsymbol{x}_k^{j*}, \boldsymbol{u}_k^{j*}) & \text{if } \boldsymbol{x} \in \mathcal{SS}^i \\ +\infty & \text{if } \boldsymbol{x} \notin \mathcal{SS}^i \end{cases} \tag{4.23}$$

where

$$F^i(\boldsymbol{x}) = \{(j,t) : j \in [0,i], \boldsymbol{x} = \boldsymbol{x}_t^j, \boldsymbol{x}_t^j \in \mathcal{SS}^i\} \tag{4.24}$$

For every point $\boldsymbol{x} \in \mathcal{SS}^i$, the value of Q^i is determined over index pairs (j^*, t^*) in the N_{ss} points, which is the minimum cost along the ship trajectories in $\mathcal{SS}^i$, in which

$$J_{t^*\to\infty}^{j*}(\boldsymbol{x}) = \sum_{k=t^*}^{\infty} h(\boldsymbol{x}_k^{j*}, \boldsymbol{u}_k^{j*}) \tag{4.25}$$

https://doi.org/10.1017/S0373463322000522 Published online by Cambridge University Press

After Equation (4.18) at time k of iteration i is solved, solutions are obtained, including $\boldsymbol{x}^{i*}_{k:k+N|k}$ and $\boldsymbol{u}^{i*}_{k:k+N|k}$:

$$\begin{cases} \boldsymbol{x}^{i*}_{k:k+N|k} = [\boldsymbol{x}^{i*}_{k|k}, \cdot, \boldsymbol{x}^{i*}_{k+N|k}] \\ \boldsymbol{u}^{i*}_{k:k+N|k} = [\boldsymbol{u}^{i*}_{k|k}, \cdot, \boldsymbol{u}^{i*}_{k+N-1|k}] \end{cases} \tag{4.26}$$

Then the first element of $\boldsymbol{u}^{i*}_{k:k+N|k}$ is applied to the ship. The finite-time optimal control problem in Equation (4.18) is solved with an primal-dual interior point method based on the Nesterov–Todd scaling at time $k+1$, based on updated state $\boldsymbol{x}_{k+1|k+1} = \boldsymbol{x}^{i}_{k+1}$. For more details regarding the solution algorithm, we refer readers to Andersen et al. (2003) and Sturm (2002). Algorithm 1 concludes the algorithmic steps.

Algorithm 1 The algorithmic steps of the proposed LMPC.

```
Require: Curvature $C(s)$ of each segment $s \in S$ of the waterway, reference path information and control parameters determination
 1: // Generate dataset $X_{\text{PID}}$
 2: Apply PID control law to the ship for $M_1$ iterations, store the data, calculate the cost of ship trajectories according to the stage cost function $h$ in Equation (4.14)
 3: // Linear state-space model construction
 4: Use linear regression techniques to approximate ship states evolution, formulate linear time-varying prediction model in Equation (4.8) based on data stored in 2
 5: // Generate dataset $X_{\text{MPC}}$
 6: Formulate optimisation problem in Equation (4.11) and apply MPC to the ship model for $M_2$ iterations, repeat similar procedure as in 2
 7: // Starting LMPC procedure
 8: While $i \leq M_3$ do
 9:   if ship has not reached finish point then
10:     Initialisation of ship states at the beginning of iteration $i$
11:     Measure the state $\boldsymbol{x}_k$ at time interval $k$, use data in previous $i - N_{\text{s}}$ iterations to build prediction model in Equation (4.19)
12:     Construct safe set $\mathcal{SS}^i$ based on previous $i - N_{\text{s}}$ ship trajectories, select $N_{\text{ss}}$ points
13:     Formulate function $Q^i$ and optimal control problem in Equation (4.18) in each time interval $k$ of iteration $i$
14:     Compute the optimal control sequence $\boldsymbol{u}^{i*}_{k:k+N} = \arg\min J^{\text{LMPC,i}}(\boldsymbol{u}_k, \boldsymbol{x}_k)$ with primal-dual interior point method
15:     Apply the first element of optimal sequences to ship, and update ship's state, set time $k = k + 1$, then go to 10
16:   else $i = i + 1$, go to 9
17: Terminate
```

4.3. *Asymptotic stability*

To guarantee the asymptotic stability of the proposed LMPC, it is desirable to use infinite prediction and control horizons. While it is not feasible to get solutions for an infinite horizon nonlinear optimisation problem, stability of LMPC can still be guaranteed by choosing suitable safe sets and setting initial conditions. This has been studied by Rosolia and Borrelli (2018), and the required stability conditions are summarised as follows.

1. There exists a controller that keeps the ship in the waterway when the ship has passed the finish point of the waterway. Assume that $\mathcal{X}_F$ is a control invariant, $\forall \boldsymbol{x}_k \in \mathcal{X}_F$, $\exists \boldsymbol{u}_k \in \mathcal{U} : \boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k \in \mathcal{X}_F$.

https://doi.org/10.1017/S0373463322000522 Published online by Cambridge University Press

2. Let $\mathcal{SS}^i$ be the sampled safe set at iteration i, $\mathcal{SS}^0$ is non-empty, and $\boldsymbol{x}_0 \in \mathcal{SS}^i$ is feasible and convergent to control invariant set $\mathcal{X}_F$.
3. At $t = 0$, $J^{\text{LMPC,i}}_{0\rightarrow N}(\boldsymbol{x}^i_0) \leq J^{\text{LMPC,i}}_{0\rightarrow N}(\boldsymbol{x}^{i-1}_0)$ holds, $\forall i \geq 1$.

If Conditions (1)–(3) hold, then the system in a closed-loop with the controller obtained by Algorithm 1 converges to a steady-state trajectory when the number of iterations goes to infinity.

5. Simulation results

To evaluate the effectiveness of the proposed method, a KVLCC2 tanker model with a length of 7 m, and a width of 1·16 m is taken as the target ship, its parameters are given in Appendix A1. The hydrodynamic parameters can be found in Yasukawa and Yoshimura (2015). We use a modular type ship manoeuvring model which was proposed in our earlier work in Liu et al. (2016) as the simulation model. This model was validated by comparing simulated and tested results (Lee et al., 2007; Yasukawa and Yoshimura, 2015) to reflect the actual characteristics of ship motion.

Experiments are performed on an Intel Core i9-10900K CPU with 16 GB RAM running Windows 10 with Python 3.7.6. CVXOPT 1·2 is used as an optimisation solver, in which a quadratic cone program solver is used. The initial ship state is set as $x_0 = 0$, $y_0 = 0$, $\psi_0 = 0$, $u_0 = 1$ m/s, $v_0 = 0$ m/s and $r_0 = 0$ rad/s. A small input rate cost is added to take into account the changing rate of the rudder angle alterations of a ship. As a ship engine usually runs at a fixed speed in practice, the value of the control input n, which represents the propeller revolution rate, is set as $n = 10{\cdot}34$. The rudder angle input ranges from −35° to +35°.

In addition, unknown disturbances are also considered, in which the following stochastic dynamics are employed:

$$\begin{cases} d_{wu} = 0{\cdot}1\sin(0{\cdot}03t + 0{\cdot}2\pi) + 0{\cdot}1u^2vr^3 \\ d_{wv} = 0{\cdot}1\sin(0{\cdot}02t + 0{\cdot}1\pi) + 0{\cdot}1u^3vr^2 \\ d_{wr} = 0{\cdot}01\sin(0{\cdot}01t + 0{\cdot}1\pi) + 0{\cdot}01u^2vr \end{cases} \tag{5.1}$$

To generate dataset X_{PID}, a PID controller is used to manoeuvre the ship to a reference path for $M_1 = 2$ loops, in which $K_p = [2, 0{\cdot}6]$, $K_i = [\text{random}(0{\cdot}1, 0{\cdot}2), \text{random}(0, 0{\cdot}1)]$, $K_d = [0{\cdot}6, 0{\cdot}9]$. It is noted that a random perturbation has been introduced in the control action deployed on the ship, so as to cover a larger region of the state space. Based on X_{PID}, a linear state space prediction model is formulated and used to construct an MPC controller. The parameters of the MPC controller are $Q = \text{diag}(1, 0, 0, 100, 0, 10)$, $R = \text{diag}(0, 10)$. Then, dataset X_{MPC} is generated via applying the MPC controller to the ship to perform another $M_2 = 2$ loops along the reference path.

The safe sets and the Q-functions in the proposed LMPC controller are retrieved from datasets X_{PID} and X_{MPC}. The LMPC controller is then applied to the ship for $M_3 = 30$ loops. In each loop, the previous $N_{\text{s}} = 3$ trajectories are chosen, in which $N_{\text{ss}} = 60$ points are chosen from each trajectory. In the LMPC controller, $Q = \text{diag}(1, 0, 0, 100, 0, 10)$, $R = \text{diag}(0, 10)$, $Q_{\text{input rate}} = [0, 15{\cdot}8°]$, $Q_{\text{terminal cost}} = \text{diag}(100, 1, 1, 1, 10, 1)$. In both MPC and LMPC controllers, the prediction horizon $N = 15$. In each iteration loop, the LMPC problem in Equation (4.18) is solved in each sampling interval with a length of $t = 1s$, and the ship data are stored to update the controller for the next iteration.

5.1. Open-loop prediction performance

To evaluate the prediction performance, the changes of ship states over time are predicted with the identified linear state-space model in Equation (4.19) with a time horizon of 20 s, starting from each point on its simulated trajectory with 1-second interval. Figure 5 shows the Root Mean Square Error (RMSE) between the predicted states and true states of the ship over prediction horizon in

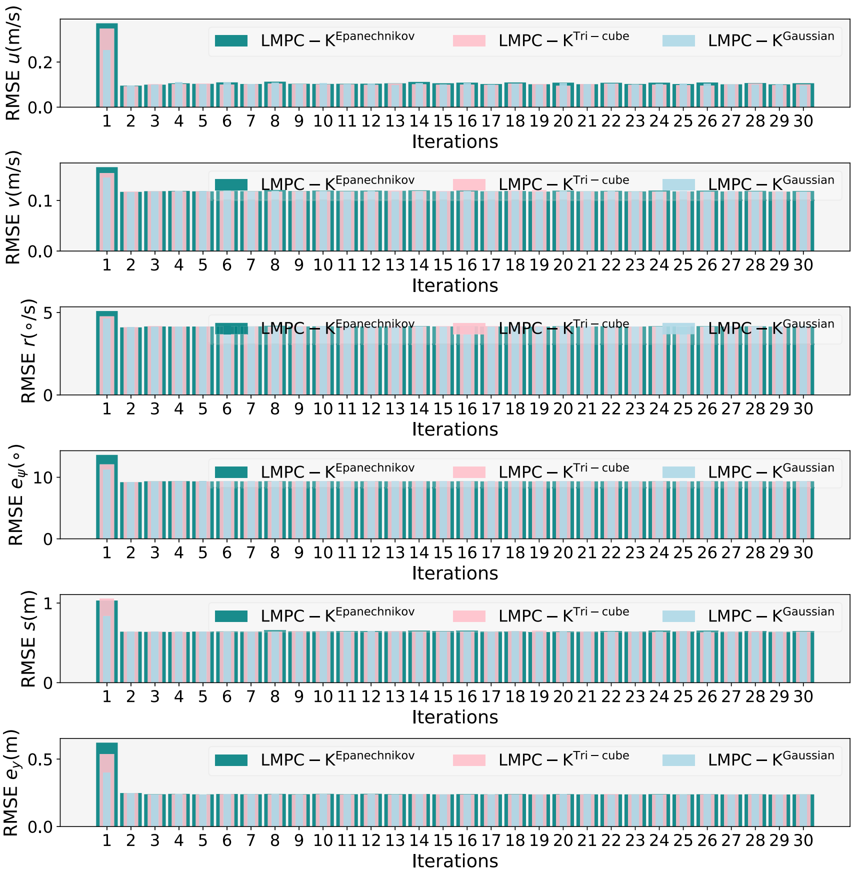

Figure 5. *Open-loop prediction performance of ship states.*

Table 1. *Comparison of Root Mean Square Errors (RMSEs) over iterations.*

	RMSE u (m/s)			RMSE v (m/s)			RMSE r (°/s)		
	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg
LMPC-$K^{\text{Epanechinikov}}$	0·372	0·096	0·114	0·165	0·116	0·120	5·09	4·099	4·190
LMPC-$K^{\text{Tri-cube}}$	0·348	0·092	0·109	0·153	0·116	0·119	4·76	4·120	4·173
LMPC-K^{Gaussian}	0·253	0·096	0·107	0·145	0·116	0·118	4·63	4·107	4·167
	RMSE e_ψ (°)			RMSE s (m)			RMSE e_y (m)		
	Max	Min	Avg	Max	Min	Avg	Max	Min	Avg
LMPC-$K^{\text{Epanechinikov}}$	13·634	9·207	9·486	1·032	0·637	0·658	0·622	0·238	0·253
LMPC-$K^{\text{Tri-cube}}$	12·096	9·227	9·422	1·054	0·633	0·655	0·538	0·235	0·247
LMPC-K^{Gaussian}	11·226	9·202	9·396	0·838	0·633	0·648	0·399	0·236	0·244

all iterations, with the prediction models constructed with different kernel functions labelled as LMPC-$K^{\text{Epanechinikov}}$, LMPC-$K^{\text{Tri-cube}}$ and LMPC-K^{Gaussian}, respectively. Table 1 presents the maximum, minimum and average RMSEs of different prediction models. As can be seen, the RMSEs of

https://doi.org/10.1017/S0373463322000522 Published online by Cambridge University Press

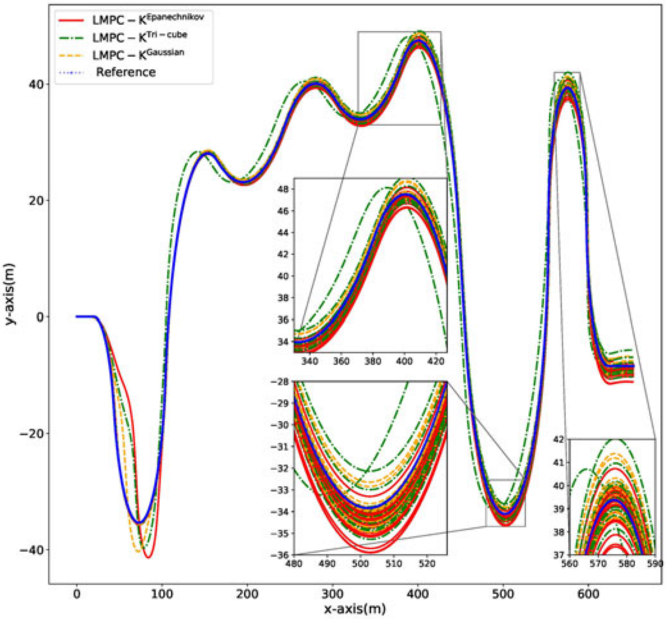

Figure 6. Simulated ship trajectories over iterations.

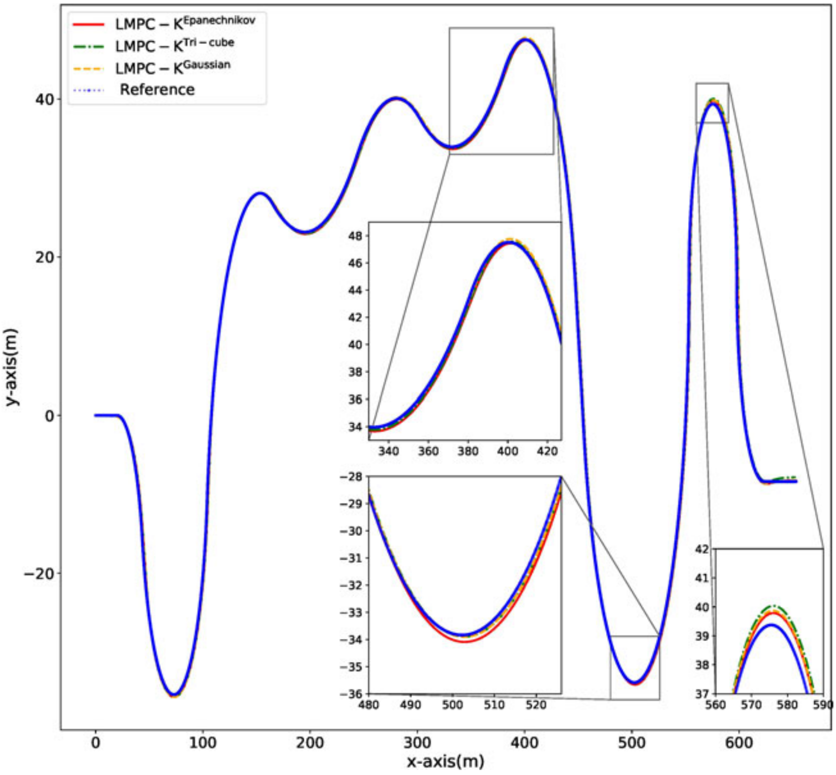

Figure 7. Simulated ship trajectories in the last iteration.

https://doi.org/10.1017/S0373463322000522 Published online by Cambridge University Press

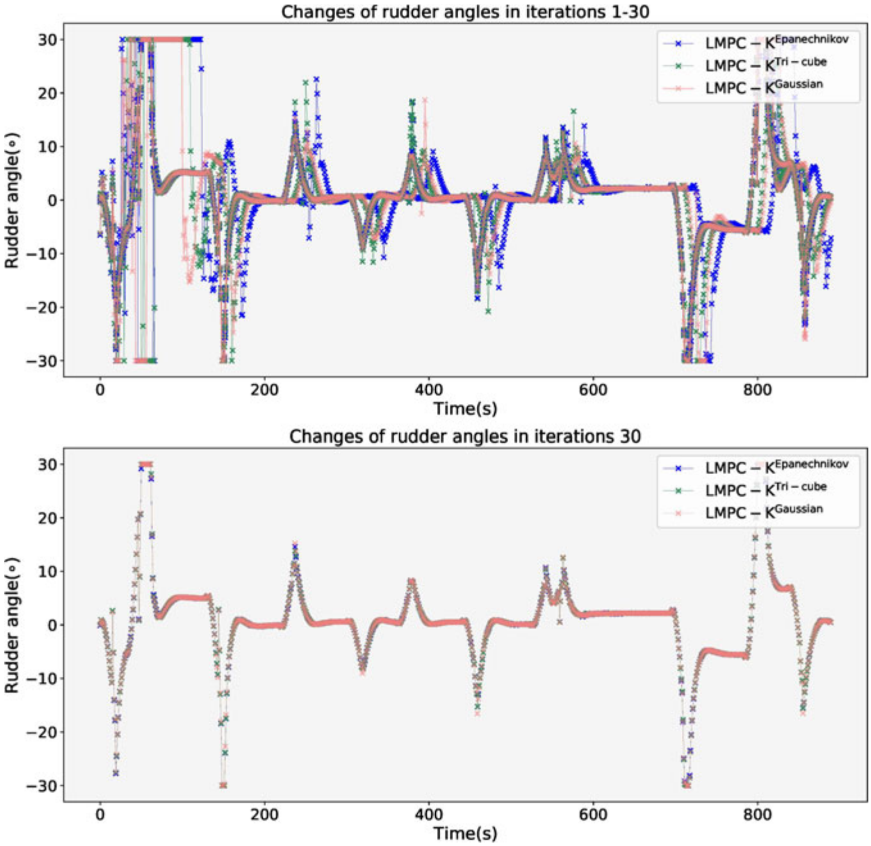

Figure 8. *Simulated control inputs over iterations.*

motion variables u, v and r are much smaller than the ship's nominal speed 1·4 m/s and rated speed 10°/s. The average RMSEs of the position variable e_ψ range from 9·396° to 9·486°, those for variable s range from 0·648 to 0·658 m and those for variable e_y range from 0·244 to 0·253 m. Among these prediction models, LMPC-K^{Gaussian} performs best as it leads to smaller deviations from true states.

5.2. Closed-loop path-following control performance of MPC

To evaluate the optimisation cost and controller performance improvements over iterations, $M_3 = 30$ loops (iterations) have been carried out with different kernel functions. Figure 6 gives the simulated ship trajectories. In the first iteration, the simulated ship trajectories of LMPC-$K^{\text{Epanechinikov}}$ and LMPC-K^{Gaussian} deviate from the reference path in the beginning but then keep up afterwards, while the ship trajectories of LMPC-$K^{\text{Tri-cube}}$ show larger deviations over the whole path. Figure 7 gives the simulated ship trajectories in the last iteration. It can be seen that with the increase of iterations, the simulated ship trajectories converge to better path-following performance. Among these LMPC controllers, LMPC-K^{Gaussian} shows relatively smaller path-following errors over iterations.

Figure 8 illustrates the simulated control inputs over iterations. It can be seen that the control inputs of different LMPC controllers varies in the beginning iterations and then converges to similar values in the last iteration.

https://doi.org/10.1017/S0373463322000522 Published online by Cambridge University Press

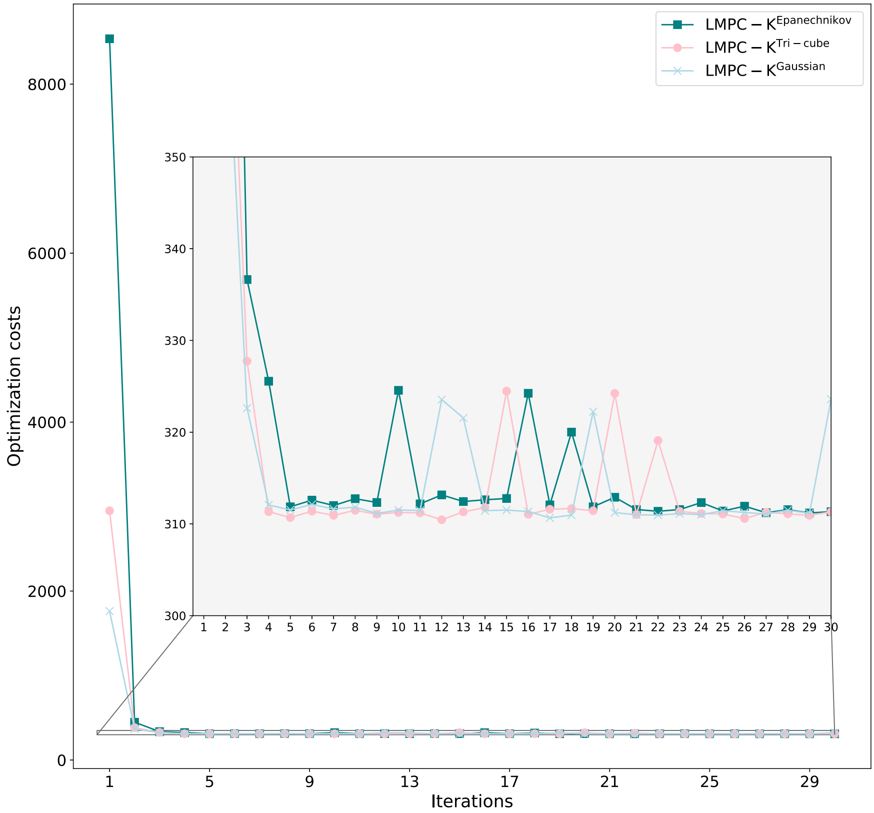

Figure 9. *Changes of optimisation costs over iterations.*

5.3. Computation costs

Figure 9 presents the changes of optimisation costs of ship trajectories over iterations, which is the sum of values of the objective function in Equation (4.18) over all sampling intervals on each trajectory. It can be seen that LMPC-$K^{\text{Epanechinikov}}$ converges at an earlier iteration, while LMPC-$K^{\text{Tri-cube}}$ and LMPC-K^{Gaussian} show larger fluctuations. Figure 10 gives the changes of path-following errors over iterations, in which the course angle error e_{ψ} converges and stabilises at approximately 4·15°, and the error on the y-axis e_y stabilises at approximately 0·18 m.

Figure 11 gives the average computation times for solving the optimisation problem in each sampling instant. As can be seen, the computation time ranges from 0·16 to 0·26 s, with an average of 0·20 s. This implies that the proposed LMPC requires far less computation time than the 1-s sampling interval, which could facilitate its implementation in practice.

6. Conclusions and future work

In this paper, a data-driven MPC strategy is proposed for path-following of unknown underactuated ship dynamics in confined waterways. It uses off-line historical data and data collected during operation to create safe sets and terminal costs. A kernel-based linear regression is used for system identification, so as to build a linear time-varying prediction model of ship states evolution. With an ILC scheme, the control approach learns from previous iterations to guarantee the stability of the system and improve controller performance. Simulation results demonstrate that it improves the path-following performance in terms of root mean square tracking error over iterations.

https://doi.org/10.1017/S0373463322000522 Published online by Cambridge University Press

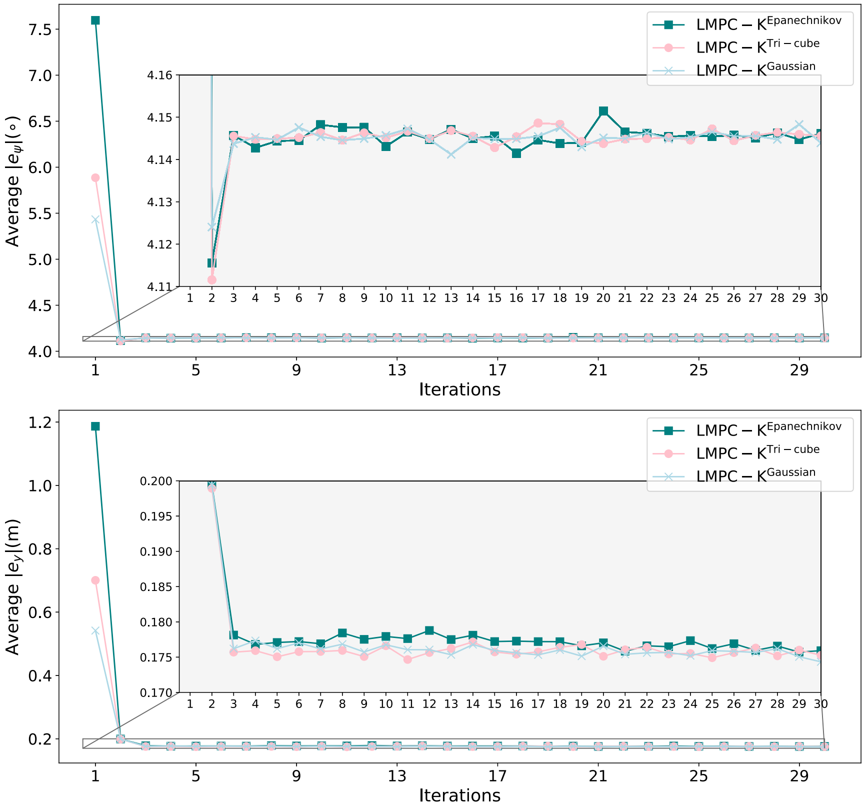

Figure 10. *Changes of average path-following errors over iterations.*

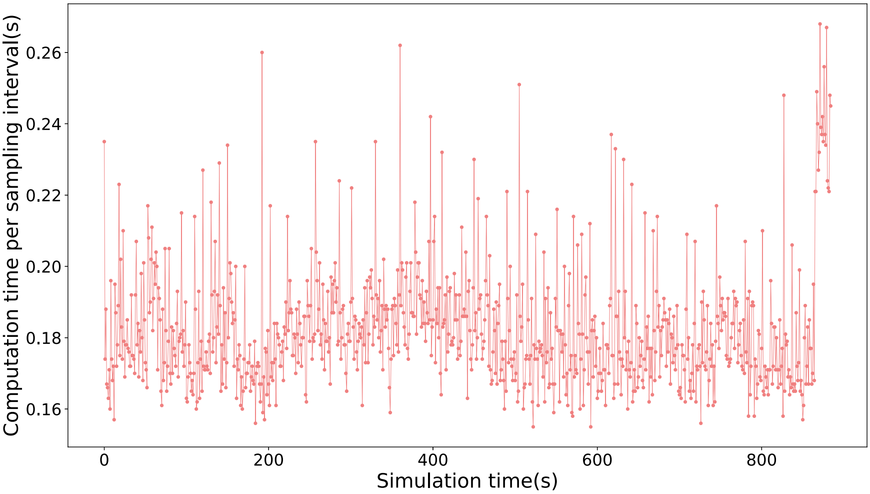

Figure 11. *Average computation times for solving the optimisation problem in each sampling interval over iterations.*

For future work, this research will be extended in several directions. First, experiments in actual waterways will be carried out to further validate its effectiveness. Second, the theoretical properties of the LMPC strategy only applies for deterministic cases. For this, a Gaussian process can be introduced to

https://doi.org/10.1017/S0373463322000522 Published online by Cambridge University Press

model the uncertainties as a function of relevant variables such as the system state and input. Moreover, it is noted that this paper uses PID to generate initial data due to its simplicity and practicality; however, the PID controller could also be replaced with other advance control techniques to generate even better initial data. It would be interesting to investigate how the quality of the off-line data would affect the performance of the LMPC controller.

Acknowledgment. This work was supported by National Natural Science Foundation of China (62003250), Southern Marine Science and Engineering Guangdong Laboratory (Zhuhai) (SML2021SP101).

References

Andersen E. D., Roos C. and Terlaky T. (2003). On implementing a primal-dual interior-point method for conic quadratic optimization. *Mathematical Programming*, **95**(2), 249–277.

Culverhouse P., Yang C., Annamalai A. S. K., Sutton R. and Sharma S. (2015). Robust adaptive control of an uninhabited surface vehicle. *Journal of Intelligent & Robotic Systems: Theory & Application*, **78**(2), 319–338.

Du P., Ouahsine A., Toan K. and Sergent P. (2017). Simulation of ship maneuvering in a confined waterway using a nonlinear model based on optimization techniques. *Ocean Engineering*, **142**, 194–203.

Du P., Ouahsine A., Sergent P. and Hu H. (2020). Resistance and wave characterizations of inland vessels in the fully-confined waterway. *Ocean Engineering*, **210**, 107580.

Fossen T. I. (2011). *Handbook of Marine Craft Hydrodynamics and Motion Control*. New York: Wiley.

Fossen T. I., Breivik M. and Skjetne R. (2003). Line-of-sight path following of underactuated marine craft. *IFAC Proceedings Volumes*, **36**(21), 211–216.

Gao S., Liu L., Wang H. and Wang A. (2022). Data-driven model-free resilient speed control of an autonomous surface vehicle in the presence of actuator anomalies. *ISA Transactions*, **127**, 251–258.

Hewing L., Wabersich K. P., Menner M. and Zeilinger M. N. (2020). Learning-based model predictive control: toward safe learning in control. *Annual Review of Control, Robotics, and Autonomous Systems*, **3**(1), 269–296.

Jin X. (2016). Adaptive iterative learning control for high-order nonlinear multi-agent systems consensus tracking. *Systems & Control Letters*, **89**, 16–23.

Kabzan J., Hewing L., Liniger A. and Zeilinger M. N. (2019). Learning-based model predictive control for autonomous racing. *IEEE Robotics and Automation Letters*, **4**(4), 3363–3370.

Lee S. W., Toxopeus S. L. and Quadvlieg F. (2007). Free sailing manoeuvring tests on KVLCC1 and KVLCC2. Technical report, Maritime Research Institute Netherlands (MARIN), Wageningen, The Netherlands.

Liang H., Li H. and Xu D. (2021). Nonlinear model predictive trajectory tracking control of underactuated marine vehicles: theory and experiment. *IEEE Transactions on Industrial Electronics*, **68**(5), 4238–4248.

Liu J., Quadvlieg F. and Hekkenberg R. (2016). Impacts of the rudder profile on manoeuvring performance of ships. *Ocean Engineering*, **124**, 226–240.

Liu Z., Lu X. and Gao D. (2019). Ship heading control with speed keeping via a nonlinear disturbance observer. *Journal of Navigation*, **72**(4), 1035–1052.

Rosolia U. and Borrelli F. (2018). Learning model predictive control for iterative tasks. a data-driven control framework. *IEEE Transactions on Automatic Control*, **63**(7), 1883–1896.

SNAME (1950). Nomenclature for treating the motion of a submerged body through a fluid. *The Society of Naval Architects and Marine Engineers, Technical and Research Bulletin No. 1-5*, 1–15.

Sturm J. F. (2002). Implementation of interior point methods for mixed semidefinite and second order cone optimization problems. *Optimization Methods & Software*, **17**(6), 1105–1154.

Wang N., Gao Y. and Zhang X. (2021). Data-driven performance-prescribed reinforcement learning control of an unmanned surface vehicle. *IEEE Transactions on Neural Networks and Learning Systems*, **32**(12), 5456–5467.

Wang L., Li S., Liu J. and Wu Q. (2022). Data-driven path-following control of underactuated ships based on antenna mutation beetle swarm predictive reinforcement learning. *Applied Ocean Research*, **124**, 103207.

Weng Y. and Wang N. (2020). Data-driven robust backstepping control of unmanned surface vehicles. *International Journal of Robust and Nonlinear Control*, **30**(9), 3624–3638.

Yasukawa H. and Yoshimura Y. (2014). Introduction of MMG standard method for ship maneuvering predictions. *Journal of Marine Science and Technology*, **20**(1), 37–52.

Yasukawa H. and Yoshimura Y. (2015). Introduction of MMG standard method for ship maneuvering predictions. *Journal of Marine Science and Technology*, **20**(1), 37–52.

Zhang H., Zhang X. and Bu R. (2022). Sliding mode adaptive control for ship path following with sideslip angle observer. *Ocean Engineering*, **251**, 111106.

https://doi.org/10.1017/S0373463322000522 Published online by Cambridge University Press

A. Appendix

Table A.1. *Applied parameters in the simulations of the KVLCC2 tanker.*

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	2·04	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

Source: Yasukawa and Yoshimura (2015) and Liu et al. (2016).

Cite this article: Li S, Xu C, Liu J (2022). Data-driven model predictive control of underactuated ships with unknown dynamics in confined waterways. *The Journal of Navigation* **75**: 6, 1389–1409. https://doi.org/10.1017/S0373463322000522