

## Research paper

# A framework for a digital twin for a Kamsarmax bulk carrier's maneuvering to avoid grounding consists of deep learning ship dynamic models and model predictive controller

Ahmed Shehata <sup>a</sup>, Mingyang Zhang <sup>b,c,\*</sup>, Nikolaos Tsoulakos <sup>d</sup>, Pentti Kujala <sup>a</sup>

<sup>a</sup> Estonian Maritime Academy, Tallinn University of Technology, Tallinn, Estonia

<sup>b</sup> MOE Key Laboratory of Marine Intelligent Equipment and System, Shanghai Jiao Tong University, Shanghai, 200240, China

<sup>c</sup> State Key Laboratory of Submarine Geoscience, Shanghai Jiao Tong University, Shanghai, China

<sup>d</sup> Laskaridis Shipping Co. Ltd., Athens, Greece

## ARTICLE INFO

## Keywords:

Ship digital twin  
Ship maneuvering prediction  
Kamsarmax bulk carrier  
Deep learning

## ABSTRACT

The prediction and control of bulk carrier movements in waterways are essential for safe navigation and the development of autonomous surface vessels. This paper introduces a deep learning-based digital twin framework for a Kamsarmax bulk carrier, integrating surrogate models within a model predictive controller (MPC) to determine control inputs for path-following and grounding avoidance. A framework was introduced and ship motions and positions were predicted using autoregressive deep learning models, validated through mean absolute error, probability density comparisons, and Kullback–Leibler divergence (KL). The result from the autoregressive model showed good accuracy with predicting surge and sway speeds, yet it underestimated the proportional change of heading. The MAE from the surge and sway speeds were 0.175 and 1.052 respectively while the heading produced a MAE of 0.751. A trust-region optimization algorithm was applied to minimize trajectory error within the MPC. The results from the MPC showed that the model was able to follow the desired path; however, greater attention is needed to the sampling frequency in which the data is collected as higher sampling frequency will improve the accuracy of the deep learning dynamic models in predicting surge speed and in capturing the proportional change of the desired heading resulting from adjustments to the rudder angle. The reason for that is the motions are nonlinear and the current sampling frequency leads to the loss of important information. Finally, the developed model can be used within a real-time system onboard ship or within a simulation environment.

## Nomenclature and acronyms

FSA	Formal safety assessment
IMO	International maritime organization
MASS	Maritime Autonomous surface vessels
RCM	Risk control measures
DL	Deep learning
Machine learning	ML
AI	Artificial intelligence
Digital twin	DT
$\hat{y}$	A predicted value
$y_{desired}$	A desired value
lon	Longitude
lat	Latitude
Surge speed	$u$

(continued on next column)

## (continued)

Sway speed	$v$
$\tau$	Vector of control inputs
$\tau_{environmental}$	Vector of the environmental condition
$\tau_{wind}$	Vector of the wind x and y components
$\varphi$	Rudder angle
$\omega$	Shaft Torque in RPM
Rate of turn	ROT
Heading	$\beta$
error	$e$
$x$	vector of variables, also called unknowns or parameters
$f(x)$	Objective function
$c_i$	Set of equality or inequality constraints that vector $x$ must satisfy
$E$	Set of indices for equality constraints

(continued on next page)

\* Corresponding author. Shanghai Jiao Tong University, Shanghai, China.

E-mail address: [mingyang.zhang@sjtu.edu.cn](mailto:mingyang.zhang@sjtu.edu.cn) (M. Zhang).

<https://doi.org/10.1016/j.oceaneng.2026.125686>

Received 5 February 2026; Received in revised form 7 April 2026; Accepted 18 April 2026

Available online 28 April 2026

0029-8018/© 2026 Elsevier Ltd. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

(continued)

$J$	Set of indices for inequality constraints
$x_o$	Initial state
$p_k$	Direction to be taken by the optimization algorithm
$x_k$	Set of values at iteration k
$\alpha$	The step that is taken by the algorithm to minimize the function
$m_k$	The function that approximates the objective function at iteration k
$f_k$	The scalar value of the objective function at iteration k
$\nabla f_k$	The gradient at iteration k
$B_k$	The hessian at iteration k
$\nabla_k$	The radius of the trust region at iteration k
GPU	Graphical processing unit
CPU	Central processing unit
Kullback-Leibler divergence	$D_{KL}$
$B$	The number of buckets(discrete)of the distribution (bins)
$P(x), Q(x)$	The probability distribution of data sets 1 and 2
$\mu$	mean of a data set
$\sigma$	The standard deviation of a data set
$H$	Bandwidth of the proposed kernel
KDE	Kernel density estimation
AIS	Automatic identification system
PID controller	Proportional-Integral-Derivative controller
MPC	Model predictive controller
NMAE	The normalized mean absolute error

## 1. Introduction

Safe navigation of maritime transport is recognized as a fundamental pillar of global trade and safety. The maritime industry carries more than 80% of global trade, and accidents at sea can have catastrophic consequences. These range from loss of human life to environmental disasters and severe economic disruption (UNCTAD, 2025). To address these risks systematically, the International Maritime Organization (IMO) developed the Formal Safety Assessment (FSA) (IMO, 2018) after the Piper Alpha disaster in 1988. Initially partial, FSA was later adopted as a mandatory process in IMO rule-making. Its purpose is to identify and control risks in the maritime sector.

The FSA framework consists of five steps:

- 1 Identification of hazards
- 2 Risk analysis
- 3 Risk control options
- 4 Cost-benefit assessment
- 5 Recommendations for decision-making

Hazards are defined as “a potential to threaten human life, health, property or the environment.” Risk is “the combination of the frequency of hazards and the severity of the consequence.” Step one focuses on identifying hazards and associated scenarios, prioritized by risk level. Importantly, FSA guidelines recommend a proactive approach, not limited to hazards that have occurred in the past (Haugene, 1998). A proactive model must not only identify hazards and analyse risks but also provide risk control measures (RCM) and risk control options (RCO). RCM is “a means of controlling a single element of risk,” while RCO is “a combination of risk control measures.” Hazards in navigation may arise from technical failures or accidents. Accidents are defined as “an unintended event involving fatality, injury, ship loss or damage, other property loss or damage, or environmental damage.”

Two main sources of accidents as reported in (Kujala et al., 2009) were grounding and ship-ship collision. Grounding accounts for 41.2% of accidents in the Gulf of Finland between 1990 and 2017 (Mingyang Zhang et al., 2022). Moreover (Montewka et al., 2010), stated that collisions occur when the distance is insufficient to perform anti-collision maneuvers. Therefore, developing an accurate control and decision making system for maneuvering is essential to ensure safe and reliable navigation (Tao et al., 2024).

In the context of Maritime Autonomous Surface Vessels (MASS), no

universal framework has yet been adopted. However, DNV introduced the first non-mandatory framework in 2018 (DNV, 2018). Within this framework, an “autonomous ship” is defined by three aspects: the location of control, the degree of system independence, and the level of human involvement. DNV categorized autonomy into four types, and Bureau Veritas (BV) (BV, 2019) proposed another framework in 2019, defining five degrees of automation (A0–A4). These degrees represent the extent to which tasks are transferred from seafarers to machines across four functions: information acquisition, information analysis, decision-making, and action implementation.

These frameworks laid the foundation for IMO guidelines on autonomous vessels, developed in 2021 by the Maritime Safety Committee (MSC), Legal Committee (LEG), and Facilitation Committee (FAL) (IMO, 2021). The MASS working group defined four degrees of autonomy based on whether seafarers are on board and whether the ship is remotely controlled:

- **Degree 1:** Seafarers operate the ship, though some systems may be fully autonomous.
- **Degree 2:** The ship is remotely controlled, but seafarers remain on board to operate systems and intervene if needed.
- **Degree 3:** Remote control only, with no seafarers on board.
- **Degree 4:** The ship makes critical decisions independently.

Furthermore, MSC 108 (2024) agreed to a Road Map for the development of a MASS Code in which by 1 July 2030, at the latest, adoption of the mandatory Code, for entry into force on 1 Jan 2032. All these frameworks agreed on the importance of developing reliable maneuvering system that can receive the information from the onboard sensors, analyse this information, anticipate the risks and mitigate these risks. Therefore, developing a model that can predict ship-ship collision and grounding and is able to suggest measures to be taken to avoid them is an essential step in risk mitigation and a cornerstone in developing MASS. Also (Wahlström et al., 2015), reported that for a MASS maneuvering system to be reliable it has to satisfy some minimum requirements. The threshold requirements include but are not limited to monitoring voyage, updating the ship route, monitoring the ship health and statuses and making decisions in view of all that.

Early models for hazard identification focused on probabilistic approaches (Fujii et al., 1974). developed one of the first, estimating the probability of grounding in specific locations based on ship course, speed and size. This model used accident records from the Japanese Association for Preventing Sea Casualties. Another modelling technique is the usage of Bayesian Network (Hänninen, 2014). More recent models used big data records from Automatic identification system (AIS) data streams (Mingyang Zhang et al., 2022). identified the possibility of ship grounding by reconstructing the ship trajectory and clustering them using K-means algorithm. While these statistical methods provide valuable insights, they have limitations. They do not offer case-based simulations or real-time risk control measures. Moreover, they often fail to account for ship manoeuvrability under real-world conditions.

Furthermore, the digital twin has emerged as a solution to these limitations. Defined by (Glaessgen and Stargel, 2012) as “The digital twin is an integrated multiphysics, multiscale, probabilistic simulation of an as-built vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its corresponding twin. A predictive digital twin anticipates future states of the ship (Menges and Rasheed, 2024). A prescriptive digital twin goes further, recommending actions based on risk analysis and uncertainty quantification. For maneuvering prediction and risk mitigation, a digital twin must:

- Simulate ship dynamics accurately.
- Predict future states.
- Anticipate risks.
- Recommend timely risk control measures.

In this process, Sensor fusion plays a central role. Onboard sensors collect data, which is processed by the ship's CPU. Then the CPU processes this data into meaningful information. Within control frameworks, the plant represents the ship's system, translating sensor data into meaningful information (Clarke et al., 1987). The plant in the context of ship's maneuvering control is a mathematical model or data driven surrogate model which acts as a representation of the ship's system that receives data from the sensors and translates this data into meaningful information. This meaningful information is then translated by an onboard controller to actions to mitigate the risks due to grounding or collision. Reactive controllers, such as PID, respond to immediate changes. Predictive controllers, such as MPC, anticipate future states and are therefore more proactive (Darby and Nikolaou, 2012). The earlier work in developing MPC for surface ships mostly relying on mathematical models such as the work in (He et al., 2023).

To mitigate risks due to grounding, the sensor fusion system must collect and analyse data about shallow and open water areas. It should also evaluate the probability of grounding, as demonstrated in (Zhang et al., 2021). In addition, the system needs to gather information about obstacle locations and assess the probability of collision. Once this data is available, the ship must plan a route from its current position to the destination. Route planning can be optimized by moving between a sequence of waypoints using algorithms such as A\* or Dijkstra (Zeng and Church, 2009). The final step is applying MPC to determine the optimal rudder angle and engine torque required to follow the planned path (Clarke et al., 1987). Within this MPC framework, a representation of ship dynamics — known as the plant — is developed. The plant takes control inputs and environmental conditions as inputs, then returns information about ship speed and location. Crucially, it can predict multiple points into the future, enabling proactive risk mitigation.

Therefore, the objective of this paper is to introduce a new framework to build a prescriptive digital twin for ship maneuvering that can predict the ship's future states and determine controls measures to mitigate a risk. Therefore, the biggest characteristic of this digital twin is that it is proactive. This new framework uses deep learning models (Chollet, 2017) for ship dynamic modelling to act as a plant and uses optimization algorithm that is based on the trust region method (Yuan, 2015) to find the ship control inputs (rudder angle and engine shaft torque) to minimize the error between the ship's desired location and its current location iteratively. The output of such a model is the control actions per unit time interval that can be taken for the ship to follow a desired path to the target. The resulted developed model is similar to MPC (Morato and Felix, 2024) that is able to predict the ship future states and find the desired control actions for the ship to follow a desired path. The case study that we are going to use in this work is a Kamsarmax bulk carrier and the dynamic model to estimate ship future states is based on the transformer encoder model developed before in the work by (Shehata et al., 2025). To be able to use the mentioned model in this task, the model is extended to be auto regressive in which the output of the current time step is used to estimate the next future state. As a result, the contribution of this paper is as follows:

- Introduce the detailed steps to develop the prescriptive digital twin and the model predictive controller.
- Extend the deep learning models developed in (Shehata et al., 2025) to be auto regressive and validate these models.
- Build a model to use these auto regressive models with trust region optimization algorithm to develop a model predictive controller to mitigate the risk of grounding.
- Validate and test the developed model with test cases that show the model capability to be integrated within a simulation environment or on board in a real time system and shows the computational efficiency of the proposed approach.

Therefore, this paper will be structured as follows:

- In chapter 2, the framework for the prescriptive digital twin will be introduced. Within this chapter a description of the auto regressive model and the optimization framework for the model predictive controller will be introduced
- In chapter 3, the test cases that the paper used to test the models will be introduced.
- In chapter 4, the results of these test cases are introduced.
- Chapter 5 will be dedicated to discussing the work done in this paper and introducing some of the important conclusions from the work.

## 2. The prescriptive digital twin basic framework

A prescriptive digital twin for a Kamsarmax bulk carrier to avoid grounding is a virtual model of the vessel that can detect the risk of grounding or collision, plan a path to avoid such events, simulate scenarios resulting from various control inputs, and determine a sequence of control actions to follow the desired path. It consists of several sub-systems, each serving a specific purpose:

- A dynamic model to predict the ship's position either a mathematical model such as (He et al., 2023) or a deep learning surrogate model such as the work done in (Shehata and Dashtimanesh, 2023) and (Shehata et al., 2025).
- A model to predict the probability of grounding, such as the work done in (Zhang et al., 2021).
- A system responsible for planning of alternative path using methods such as the great circle (Tao and Du, 2024).
- An MPC to find the optimal control commands to follow the desired path (Clarke et al., 1987). In the context of ship maneuvering, the flow chart to build this prescriptive digital twin is introduced and described in Figs. 1 and 2 shows the MPC flowchart.

In the context of ship maneuvering, the flowchart to build this prescriptive digital twin is introduced and described in Fig. 1, while Fig. 2 shows the MPC flowchart.

Fig. 2 shows the main flowchart or pseudo code for the developed MPC within the prescriptive digital twin framework introduced in Fig. 1. The model starts by using the current ship states as initial condition in this time step as shown in Fig. 2. Then, the model calculates the next state  $i+1$  using a set of random control variables  $\tau$ . The model that is used in estimating the next time step's variables is the deep learning transformer model developed in our previous work (Shehata et al., 2025). Throughout the whole process, the objective is to Minimize the error between the desired position and the resulted position from the current set of control variables. The error between the predicted state and the desired state will be estimated from equation (1).

$$e = \sum_{i=0}^{i=n} ((\widehat{lon}_i - lon_{desired})^2 + (\widehat{lat}_i - lat_{desired})^2) * w_i + rudder_{penalty} \quad (1)$$

where,  $\widehat{lon}_i$  and  $\widehat{lat}_i$  are the predicted longitude and latitude at any point  $i$ .

$lon_{desired}$  and  $lat_{desired}$  are the desired longitude and latitude at any point  $i$ .

$w_i$  a weight for the error at any time step and it's a number between 0 and 1.

$rudder_{penalty}$  is a penalty that is given to the rudder to avoid extreme turns and changing the rudder angle rapidly.

For the DT to not take any extreme actions to correct the ship path, it is predicting  $n$  number of points in advance and estimating the error based on this  $n$  number of points as shown in Fig. 1 and equation (1). This process is demonstrated in Fig. 3. Furthermore, the rudder penalty introduced in equation (1) increases non-proportionally with the increase in the rudder angle. If the change in rudder angle is 0 then this term will be zero and it increases unproportionally and in accordance

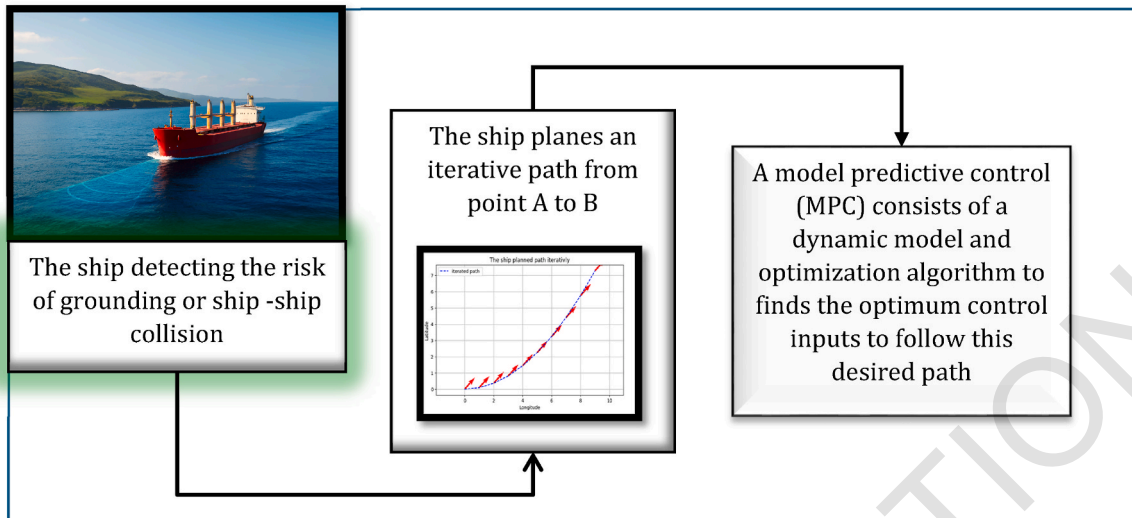


Fig. 1. The prescriptive digital twin framework to avoid grounding or ship to ship collision.

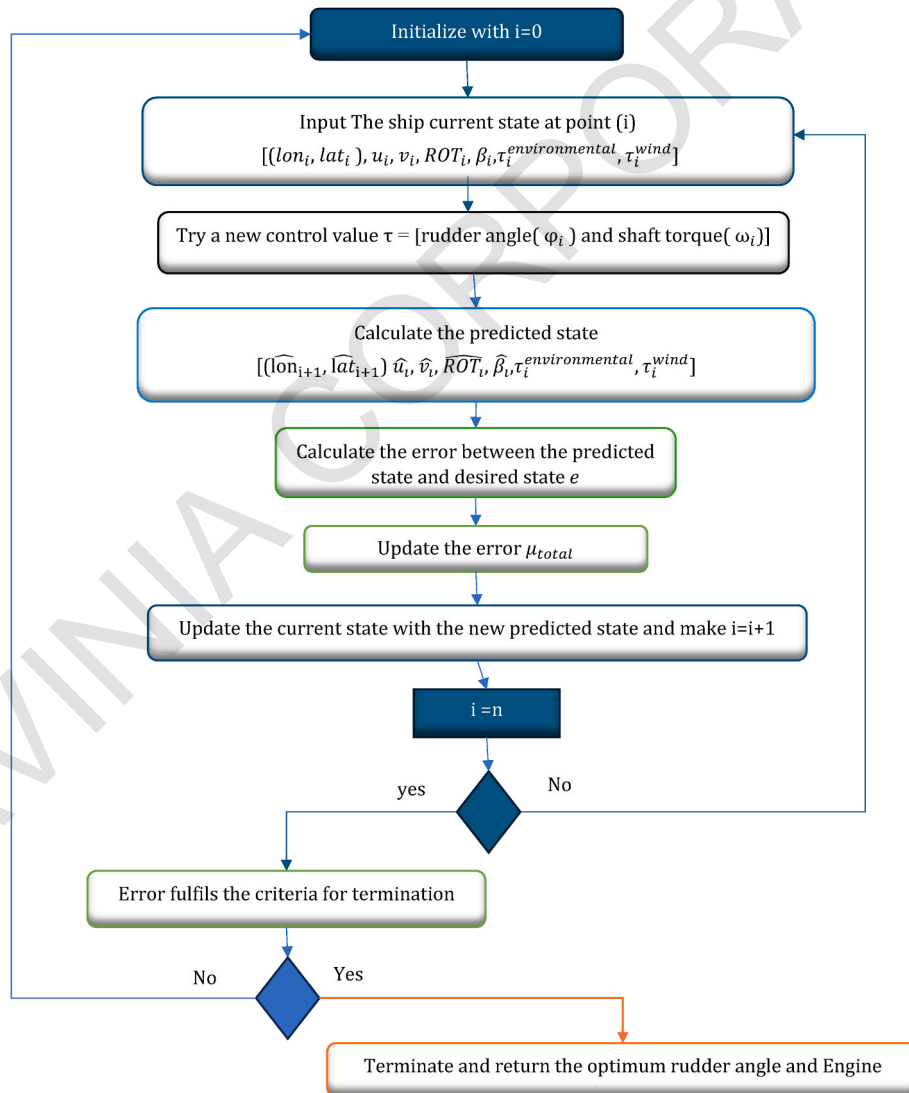


Fig. 2. Flowchart describes the process for developing the MPC.

with the error. This results in preventing instant heavy turning to minimize the next point's error, unless it is necessary, which might lead

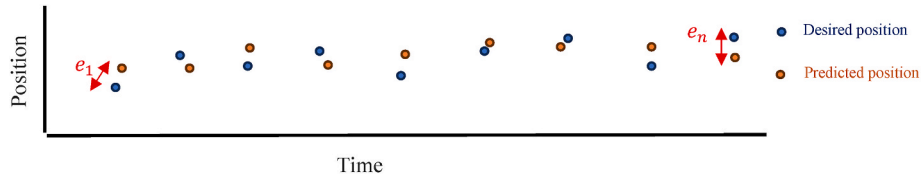


Fig. 3. The desired and predicted position and the error throughout the process.

to some sort of zigzag maneuvering.

Then the optimization algorithm is exploring the search space with the objective function is to minimize the error described in equation (1) There are several optimization algorithms that can be used to achieve this task. In this work, a trust-region algorithm was employed. The trust-region algorithm, like other optimization methods, seeks to determine the set of  $x$  values that minimize or maximize a function  $f(x)$ . Its suitability for the present problem lies in its ability to redefine the search direction, thereby reducing reliance on the proposed initial condition. Trust-region methods simultaneously optimize both the direction and the magnitude of the step within a specified neighborhood around the current iterate (Nocedal and Wright, 2006a)

$$\min_{x \in \mathbb{R}^n} f(x) \text{ subject to } \begin{cases} c_i(x) = 0 & i \in \mathcal{E} \\ c_i(x) \geq 0 & i \in \mathcal{I} \end{cases} \quad (2)$$

Here  $\mathcal{E}$  and  $\mathcal{I}$  are sets of indices for equality and inequality constraints.

And  $c_i(x)$  are set of equality or inequality constraints that vector  $x$  must satisfy.

In the developed work and as mentioned earlier, The function  $f(x)$  introduced in equation (2) refers to the error introduced in equation (1). This error is calculated using the ship's dynamic surrogate models. The function takes as input the environmental conditions along with the ship's surge, sway, heading, rate of turn, longitude, and latitude. At each time step, it iteratively computes the ship's new states and then evaluates the error between the predicted and desired states. Thus, the function effectively implements the flowchart described in Fig. 1. One advantage of using deep learning models is that the gradient is obtained during model development, allowing it to be directly incorporated into the optimization framework to minimize the function.

### 2.1. The auto regressive model

In this work, the ship dynamics is calculated using a data driven surrogate model developed in (Shehata et al., 2025). In this model a time series surrogate model was developed to estimate a bulk carrier surge, sway, rate of turn, heading, longitude and latitude. The model structure was based on the transformer encoder developed in the work attention is all you need (Vaswani et al., 2017). It estimates the maneuvering characteristics for the bulk carrier and is able to estimate its longitude and latitude. The longitude was measured as a reference [km] east or west of the prime meridian while latitude was measured as reference [km] north or south the equator This model can estimate bulk carrier's maneuvering and relays only on one data point in the past which will act

as the initial conditions and can estimate one time step ahead of time. For this model to be effective in such a task it must be extended to be auto regressive. Auto regressive models use the model prediction as a basis to predict (n) future states as shown in Fig. 4.

Fig. 4 describes how the developed auto regressive model works. In the auto regressive model, at any point in time the current known state of the ship is considered as  $state|_{t=0}$ . Then this known state is used with environmental conditions and ship controls to predict what the state would be at  $t = 1$  then this process is repeated (n) number of times with n is the number of predictions in the future.

In our previous work we developed models for surge, sway, rate of turn, latitude and longitude and it was reported that the accuracy of the rate of turn (ROT) model was lower than the other motions. Yet, and as we will see further in the results, the latitude is so dependable on the ship's heading. To improve the model and use it within the control, this paper proposes using an approximation of the bulk carrier's heading as in equation 3

$$\beta_{t+1} = \beta_t + ROT \quad (3)$$

where,  $\beta_t$  is the heading at time t in degree

ROT is in degree per minute.

This model works in our case due to the fact that our model was trained with data that was collected every 1 min so heading of t+1 is 1 min from the heading of t and ROT is the rate of turn in degree per minute so adding this rate of turn to the old heading can find the ship's new heading. Furthermore, and as shown in Fig. 5, the majority of the trip the heading was not changing or a negligible change and by examining the distribution of the data, we can see that normalized rudder angle value between  $[-2,2]$  was needed to keep the ship in its course without any changing in the heading. Therefore, in our model we are setting the rate of change for normalized rudder angle between  $[-2,2]$  to zero. Therefore, the ship will only change its heading in case of experiencing  $|Normalised\ rudder\ angle| > 2$  and  $ROT|_{Rudder| < 2} = 0$ .

### 2.2. The optimization framework

In a prescriptive digital twin and as described in Fig. 3, the model needs to find the optimum rudder angle and engine torque at every time step. Optimization as defined in (Nocedal and Wright, 2006b) is the process of minimizing or maximizing an objective function  $f(x)$  subjects to constraints  $c_i$  on its variables vector  $x$ . Optimization algorithms are iterative process. They begin with an initial guess of the variable  $x$  and generate a sequence of improved estimates (called "iterates") until they

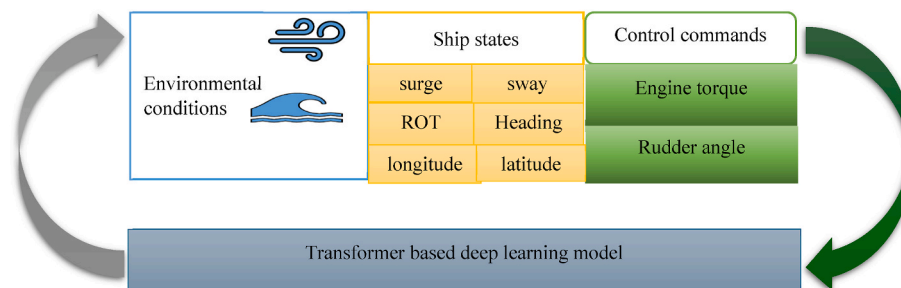


Fig. 4. The process to develop an auto regressive model for n ship states.

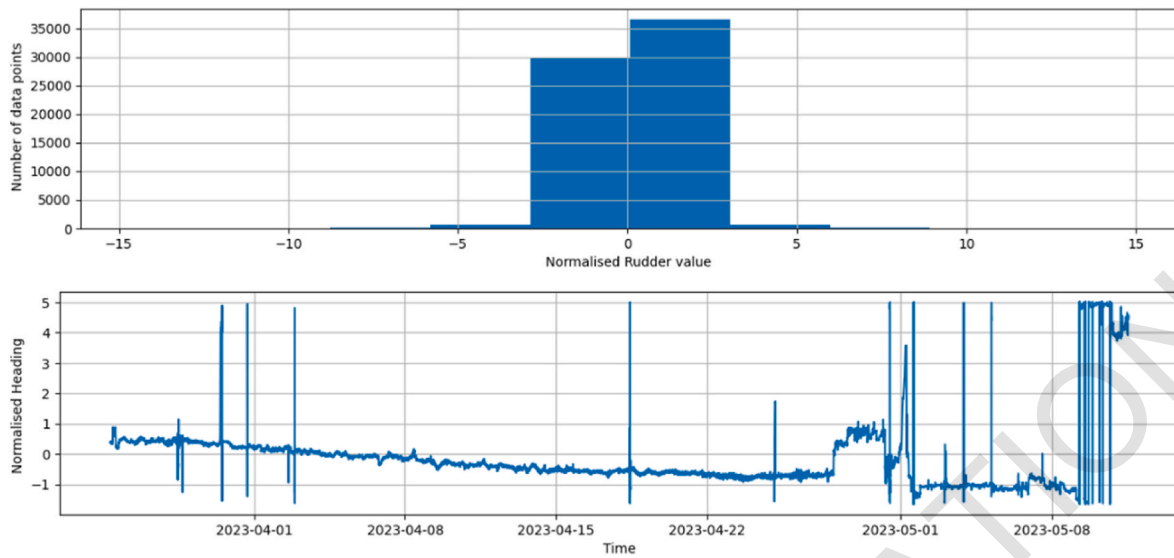


Fig. 5. Histogram shows the distribution of rudder values in our data set.

terminate, hopefully at a solution. Some algorithms accumulate information gathered at previous iterations, while others use only local information obtained at the current point. Most of the algorithms make use of the 1st derivative (called “the gradient”) and some of them make use of the 2nd derivative (called “the hessian”). There are two main search strategies to solve a robust optimization problem, line search method and trust region method. In the line search methods, the algorithm chooses a direction  $p_k$  (called “the search direction”). The algorithm uses this direction to move from set of values at iteration  $k$  ( $x_k$ ) to the next iterate  $x_{k+1}$  subject to satisfy the condition in equation (4). The algorithm also uses trial and error to decide the step that it needs to take to satisfy the condition in equation (4)

$$\min_{\alpha > 0} f(x_k + \alpha p_k) \quad (4)$$

Where  $\alpha$  is the step length

For every iteration and to build a good approximation, the line search algorithm generates a limited number of trial step lengths until it finds one that loosely approximates the minimum of equation (4) and consider that the next iterate  $x_{k+1}$ . The disadvantage of this method is that it relies heavily on the set of initial guesses  $x_0$ . Initial guess is mainly estimated using past knowledge or experience for instance in this work we are assuming our initial guess is the control inputs of the previous data point therefore we are assuming, as initial guess, that the control will not need to change its magnitude.

In this work we are proposing using trust region method (Nocedal and Wright, 2006a). In trust region method, the algorithm is minimizing a model function  $m_k$  in which the constructed model acts as an approximation to the behavior of the objective functions as in equation (5).

$$\min_p m_k(x_k + p) \text{ where } (x_k + p) \text{ lies inside the trust region} \quad (5)$$

Usually, the trust region is a circle which can be defined by the radius  $\Delta > 0$  and the model  $m_k$  is a quadratic function defined as in equation (6).

$$m_k(x_k + p) = f_k + p^T \nabla f_k + \frac{1}{2} p^T B_k p \quad (6)$$

where,  $f_k$ ,  $\nabla f_k$ , and  $B_k$  are a scalar, vector, and matrix, respectively.  $f_k$  and  $\nabla f_k$  and  $B_k$  are the function value, the gradient and the hessian of the function using set of values  $x$  at iteration  $k$ .

The trust-region strategy differs from line search, which stays with a

single search direction. While a trust region starts by computing a radius  $\Delta$  then reduce this radius if the algorithm cannot find an optimum solution. That setup makes it more robust and more efficient and allows the algorithm to explore more area around the current solution which leads to more guarantee to find a local minimum point. Furthermore, as in equation (6),  $m_k(x_k + p)$  relies on the gradient and hessian of the objective function and due to our objective function was driven from a deep learning model, therefore it is possible to compute its gradient with using the TensorFlow gradient tape function (Chollet, 2017) which makes it easier and faster for the algorithm to find the local minimum of the function.

### 3. The test and validation of the developed models

The developed models in (Shehata et al., 2025) was developed using data collected for Kamsarmax 81,000 TDW Bulk carriers shown in Table 1.

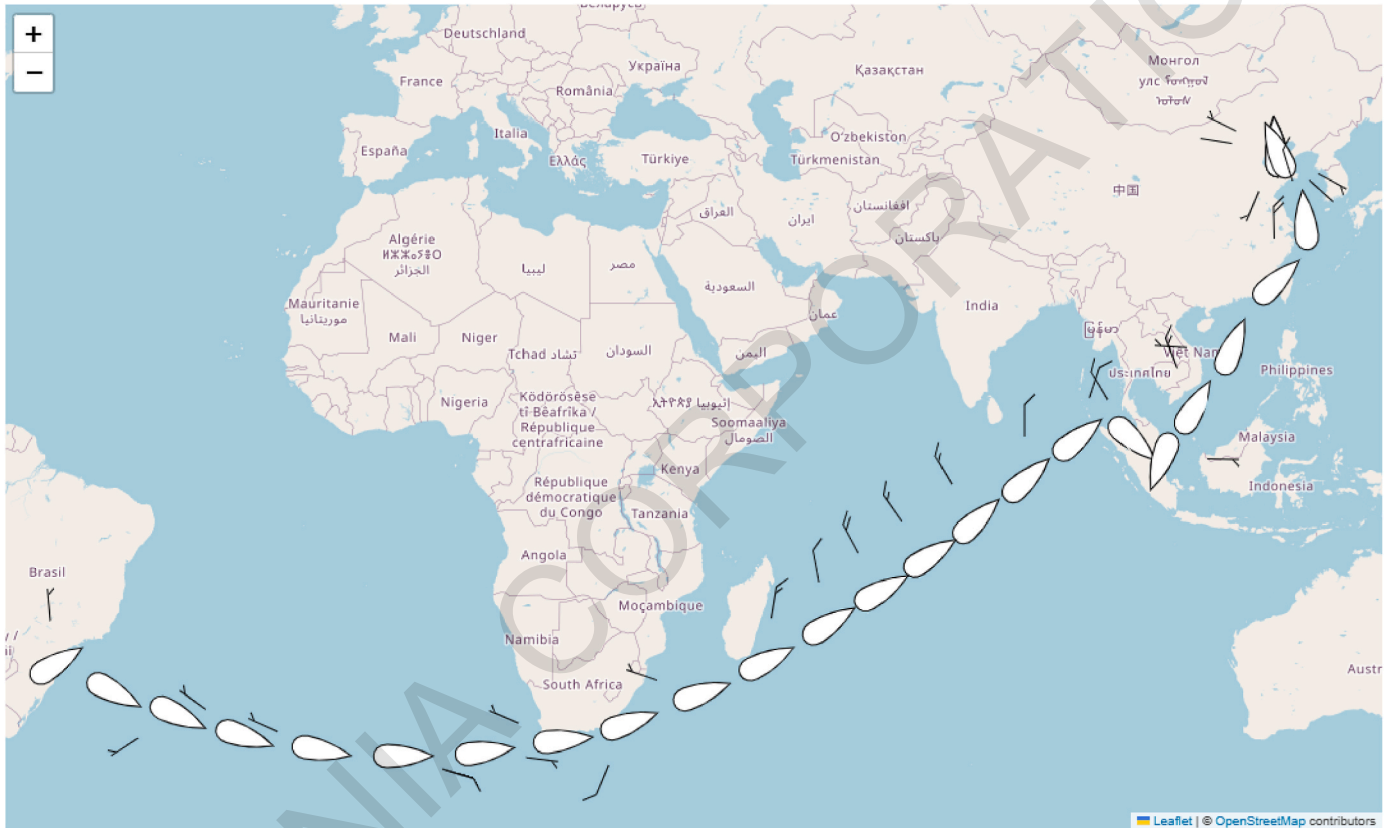
In order to test the auto regressive model performance. we will regenerate the data points. The regeneration will be for some of the data that was used in training and validating the models. The bulk carrier sailed between Brazil and China. It was collected in the period between 11th of March 2023 to 19th of May 2023. The collected data considers the environmental sea waves and current as well as the wind information (Zhang et al., 2023) as shown in Fig. 6.

#### 3.1. The test case for auto regressive model

To evaluate the model's performance, we selected the case illustrated in Fig. 7. This case was chosen because the bulk carrier changes direction multiple times, thereby providing diverse scenarios to test the model's robustness and sensitivity to variations in motion-driving variables. In the first segment, the vessel moved northeast, with both longitude and latitude increasing. In the subsequent segment, latitude remained constant while longitude increased, corresponding to eastward motion. Later, latitude decreased, indicating southward movement. Finally, the trajectory returned to simultaneous increases in longitude and latitude. As noted, the objective of developing these autoregressive models is their application within predictive control for bulk carrier navigation. Accordingly, we examine surge, sway, heading, longitude, and latitude. The recorded rudder angle and shaft torque are used to assess whether the autoregressive model can reproduce the observed data. In the next step, we will evaluate the model's performance within predictive control.

**Table 1**  
Ship characteristics for the bulk carrier.

Length overall LOA(m)	229
Length B.p $L_{Bp}$ (m)	225
Service speed in (knots)	14.3
Depth (m)	20.05
Breadth (m)	32.26
Design draught(m)	12.2



**Fig. 6.** The trip in which the data was collected.

In order to validate the auto regressive model and evaluate its performance, we will generate the ship surge and sway speeds, rate of turn, heading, longitude and latitude using our auto regressive model and use visual inspection of the results and quantitative evaluation (Duarte et al., 2019). We will investigate its performance using the mean absolute error between the ground truth and the predicted data. The mean absolute error (MAE) is introduced in equation (7). And due to mean absolute error is a scale-dependent metric, we will use the normalized mean absolute error (NMAE) introduced in equation (8) to show the percentage of the deviation between the MAE and the data bounds.

$$MAE = \frac{\sum_{i=1}^n y_i - \hat{y}_i}{n} \tag{7}$$

$$NMAE = \left( \frac{MAE}{y_{max} - y_{min}} \right) * 100 \tag{8}$$

where,  $y_i$  is the ground truth of the target variable at  $i$ th position and  $\hat{y}_i$  is the prediction at the same position. Furthermore, to investigate how much deviation between the approximated model and the ground truth data, we will use Kullback-Leibler (KL) divergence (Kullback and Leibler, 1951) shown in equation (8).

$$D_{KL}(P(x)||Q(x)) = \sum_{i=0}^b P(x_i) \cdot \ln \left( \frac{P(x_i)}{Q(x_i)} \right) \tag{8}$$

where,  $P(x)$  and  $Q(x)$  are the two probability distributions for the two data sets and  $b$  is the number of bins.

$D_{KL}$  measure the difference in both value and direction between one normal distribution and another as an interpretation of the divergence of the predicted from the ground truth. It can show how good a surrogate model is a representation of the actual data. The lower this KL divergence, the better the model. Therefore, a KL divergence of zero indicates a perfect model. However, using KL divergence is not that much straight forward as it relies on the number of bins ( $b$ ). In this work, we are using

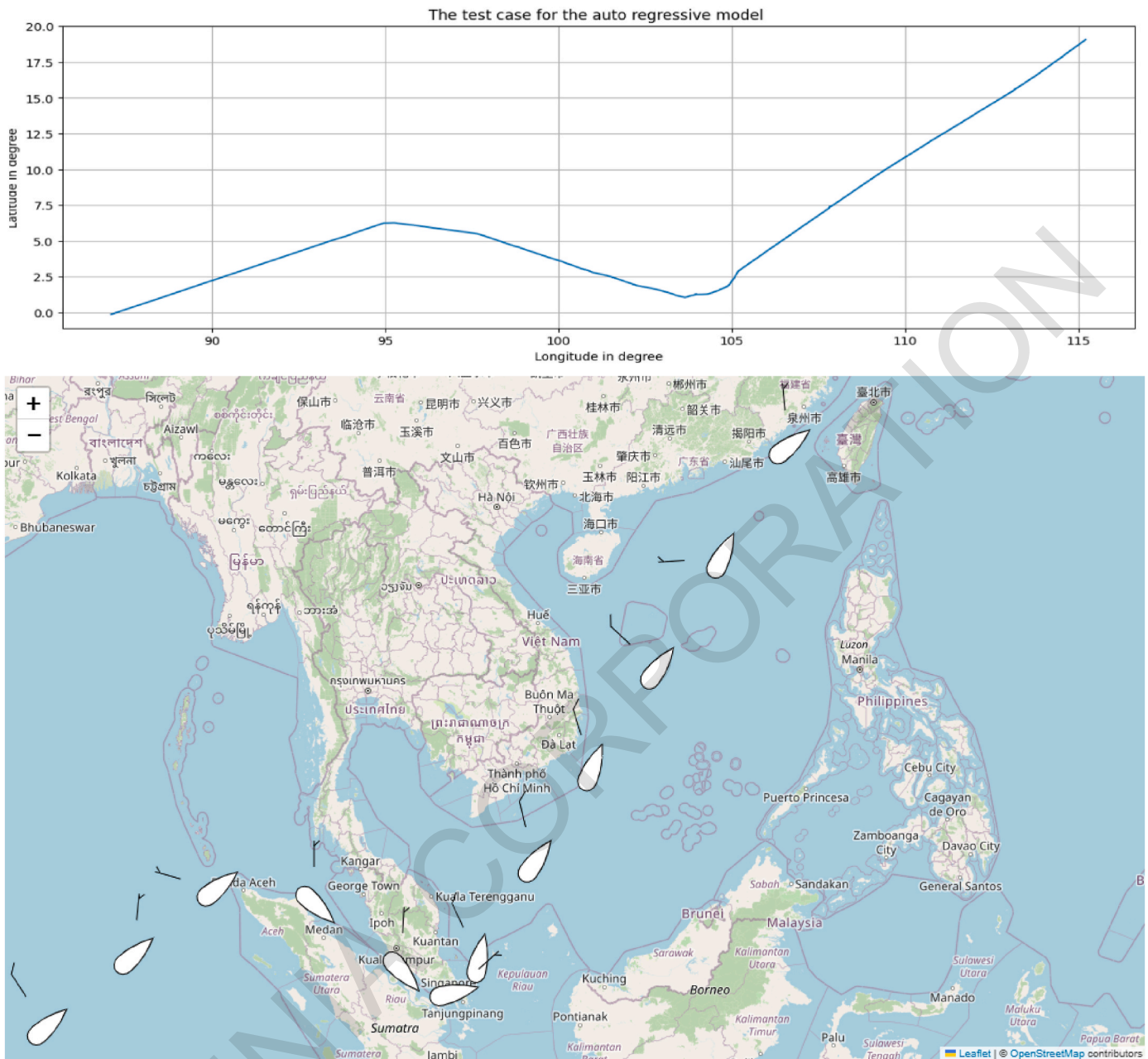


Fig. 7. The test case for the auto regressive model.

kernel density estimation (KDE) (Parzen, 1962) as a method to estimate/smooth the probability density function of our two data sets (“Density Estimation, 2017) as shown in equation (9).

$$\hat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right) \quad (9)$$

$\hat{f}_h(x)$ : is smoothed probability density function with kernel K and bandwidth h.

To estimate bandwidth, there are several methods to accomplish an optimum bandwidth as reported in (Turlach, 1999) however for the purpose and scope of this work, we don't need to get the bandwidth optimally as the purpose is to show if both motions are following the same probability distribution and to estimate the deviation between the prediction and the ground truth data. Therefore, we will use trial and error and visual inspection. and, in this work, we are using gaussian kernel shown in equation (10).

$$K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}} \quad (10)$$

### 3.2. The test case for the predictive controller

In order to evaluate the performance of the predicted controller. We are going to use part of this trip as a test case by allowing the optimization algorithm to use the developed dynamic models to obtain the optimum sequence of rudder angles and shaft torques which will make the bulk carrier follow this desired path. The test case is shown in Fig. 8. This test case is part of the trip introduced in Fig. 7. Furthermore, we will compare the resulting control inputs from our model predictive controller to the actual ones that were taken by the ship to perform this maneuver.

In addition, we investigated the suitability of integrating such a controller within a real-time system. The paper reports the duration

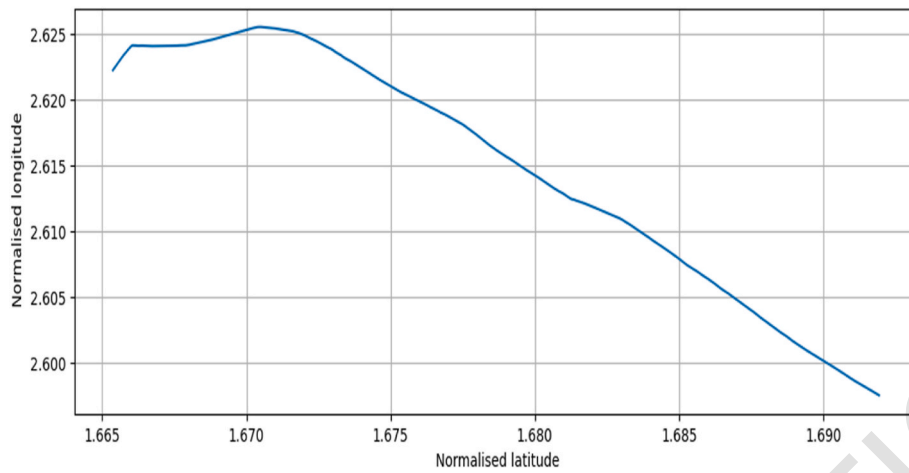


Fig. 8. The desired path for the model predictive controller.

required for the MPC to compute the optimal values, which reflects the computational performance on the hardware specified in. The simulations were implemented in Python 3.11 using Keras 3.9.0, TensorFlow 2.18.0, and SciPy 1.15.2. All models were executed on a CPU without GPU acceleration.

#### 4. Results

##### 4.1. The result of the auto regressive model

In order to validate the results from the models. We will rely on visual confirmation, the MAE and NMAE between the generated data and the actual data, the probability distribution of both quantities and the



Fig. 9. The predicted surge and sway speeds vs the recorded data from the trip in the 1st and the 2nd figure and the shaft torque that was used to produce this data is given in the bottom figure.

$D_{KL}$  between these two distributions. Surge and sway speed's models both were capable of providing good approximation to the original data. In Fig. 9, the predicted surge and sway speeds in orange follow the same behavior as the ground truth in blue as well as the shaft torque in the last subplot. Furthermore, both the predicted and ground truth follows the same probability distribution as shown in Fig. 10. The KL divergence for surge speed was 1.064, slightly higher than desirable but still sufficiently small to indicate a valid representation. For sway speed, KL divergence was 0.1385, reflecting excellent agreement. MAE values were 0.175 for surge speed and 1.052 for sway speed. And because MAE is a scale dependent quantity these numbers doesn't show the whole story. NMAE, which shows the percentage of deviation from the variable's bounds, were 5.6277 % and 4.899 %. These results suggest that surge speed predictions were less accurate in magnitude, while sway speed predictions better captured the overall trend. It also indicates that both surge and sway MAE didn't deviate from their respective bounds by more than 6 %.

Heading prediction remained more challenging. As reported previously, transformer encoder models struggled with heading estimation. In this work, heading was derived from the predicted rate of turn using Equation (3). Fig. 11 shows that the model followed rudder commands but underestimated heading changes. This limitation is attributed to the 1-min sampling interval and the assumption of constant rudder angle within each interval. Higher-frequency data collection (e.g., 10-s intervals) would improve accuracy. Yet, MAE of 0.751 and NMAE of only 11.24 % shows that the prediction overall accuracy can be considered good enough and it only deviate by 11 % from the bounds. Furthermore and as shown in Fig. 11, the model followed rudder commands but underestimated changes in heading. Although the predicted trajectory (yellow) responded to rudder inputs, the magnitude of heading variation was smaller than observed in the ground truth. The primary challenge in achieving higher accuracy lies in the limited number of training data points and the sampling interval (dt). In this study, data were collected at 1-min intervals, with the assumption that rudder angle remained constant throughout each interval. This assumption reduced the model's ability to capture trends in the data. Improving heading prediction requires higher-frequency sampling and training data that include complex manoeuvres, such as rapid directional changes. The dataset used (Fig. 6) contained few such events, limiting the model's ability to

generalize. Moreover, extreme rudder angles were rare, with most values between  $-10^\circ$  and  $+10^\circ$ . Despite these limitations, the model successfully followed rudder commands and provided accurate proportional estimates of heading changes due to the change in rudder angle. Fig. 12 shows that the probability distributions of predicted and ground truth heading values were closely aligned, with most data points falling within the same range. This level of accuracy is sufficient to provide reliable rudder command indications for integration into predictive control, enabling the controller to determine suitable rudder angles for achieving desired longitude and latitude trajectories.

The objective of this work is to develop a prescriptive digital twin capable of predicting bulk carrier motion and determining optimal control inputs to follow a desired path. Accordingly, models for longitude and latitude prediction, which are strongly correlated with ship dynamics, are essential components.

For longitude and latitude, KL divergence and probability distribution analysis are not meaningful, as these variables are uniformly distributed. Instead, evaluation relies on direct comparison of predicted and ground truth trajectories. Figs. 13 and 14 present the predictions for longitude and latitude, respectively.

Fig. 13 shows that longitude increased consistently throughout the voyage. The dataset contained only ascending longitude values, and the model reproduced this behaviour with high accuracy, yielding a near-zero MAE of 0.048 and NMAE of 6.858356. By contrast, latitude exhibited both ascending and descending trends, reflecting its dependence on heading prediction (Fig. 14).

In the mid-voyage segment, the heading model underestimated directional changes, which led to deviations in latitude prediction and increased MAE of 0.253 and NMAE of 8.35249. This limitation stems from the assumption that rudder angle remained constant for each 1-min interval. Higher-frequency sampling would improve the model's ability to capture heading changes and, consequently, latitude dynamics.

Validation against control inputs further confirmed model fidelity. Between April 30 and May 1, torque decreased significantly. As expected, surge speed dropped to zero, sway speed stabilized at zero, heading remained unchanged, and longitude and latitude ceased to vary. These behaviours, consistent across Figs. 9–14 demonstrate that the models accurately captured vessel dynamics under varying control inputs.

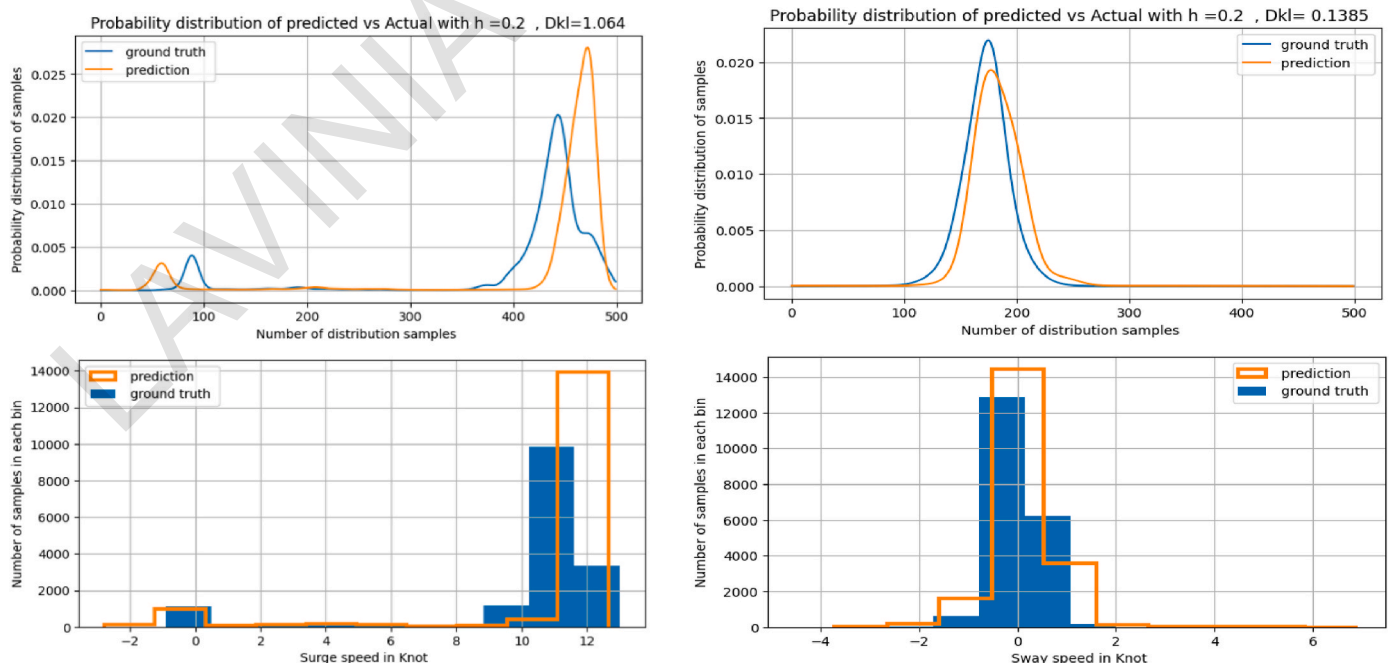


Fig. 10. The probability distribution of ground truth  $P(x)$  and the predicted surge and sway speeds  $Q(x)$  and the second figure shows histogram of the distributions.

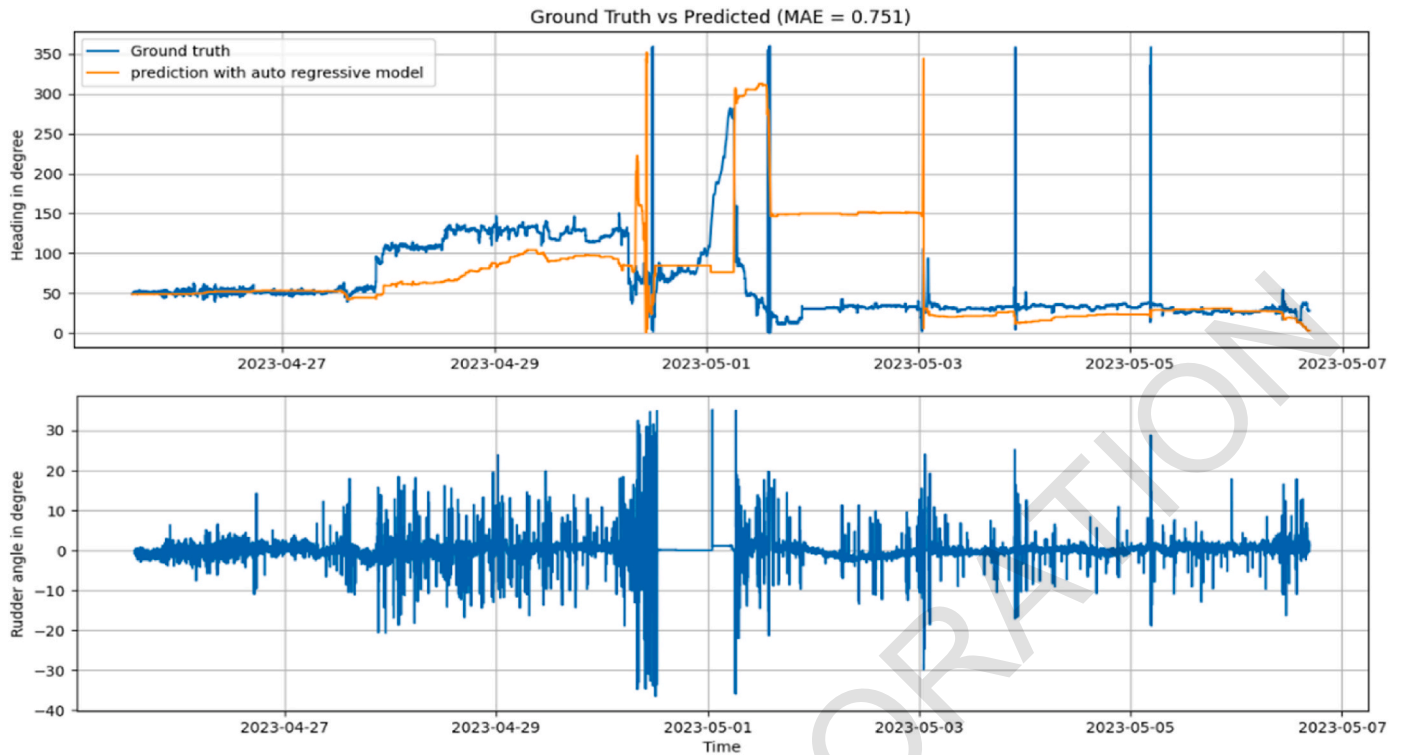


Fig. 11. The prediction of the Heading of the ship with the auto regressive model in yellow and the ground truth in blue. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

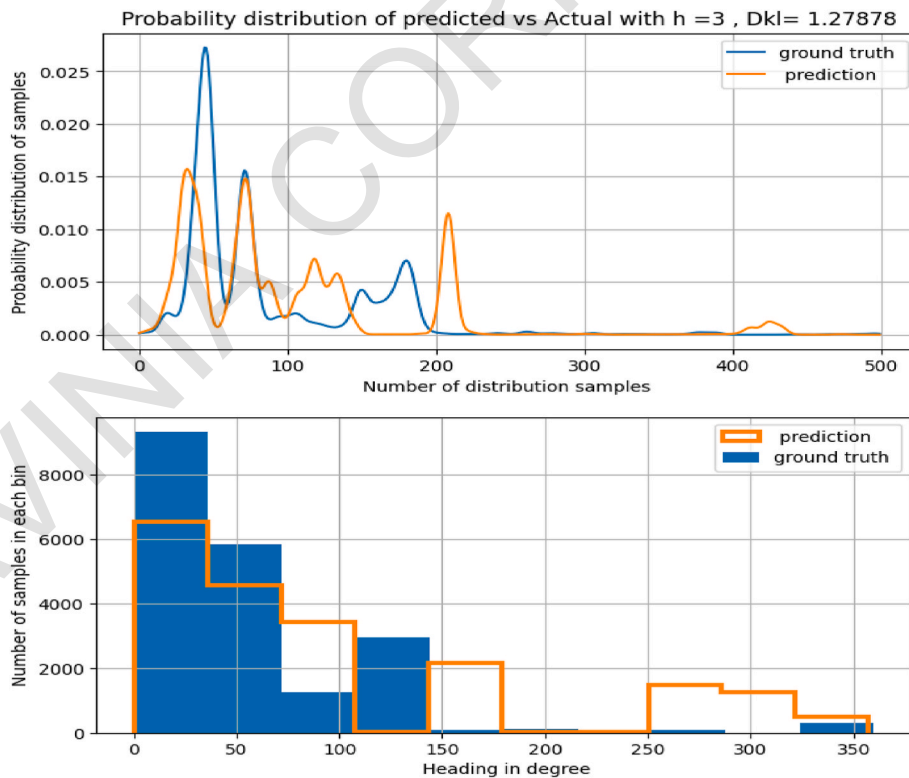


Fig. 12. The probability distribution of the heading and the histogram of the heading.

In conclusion, the developed models successfully predicted ship motions in response to changes in engine torque and rudder angle. Accuracy can be further enhanced by training with higher-frequency data, enabling more precise estimation of heading and latitude.

#### 4.2. The result of the predictive controller

The predictive controller is an optimization problem as mentioned in the second sector. The objective is to minimize the error

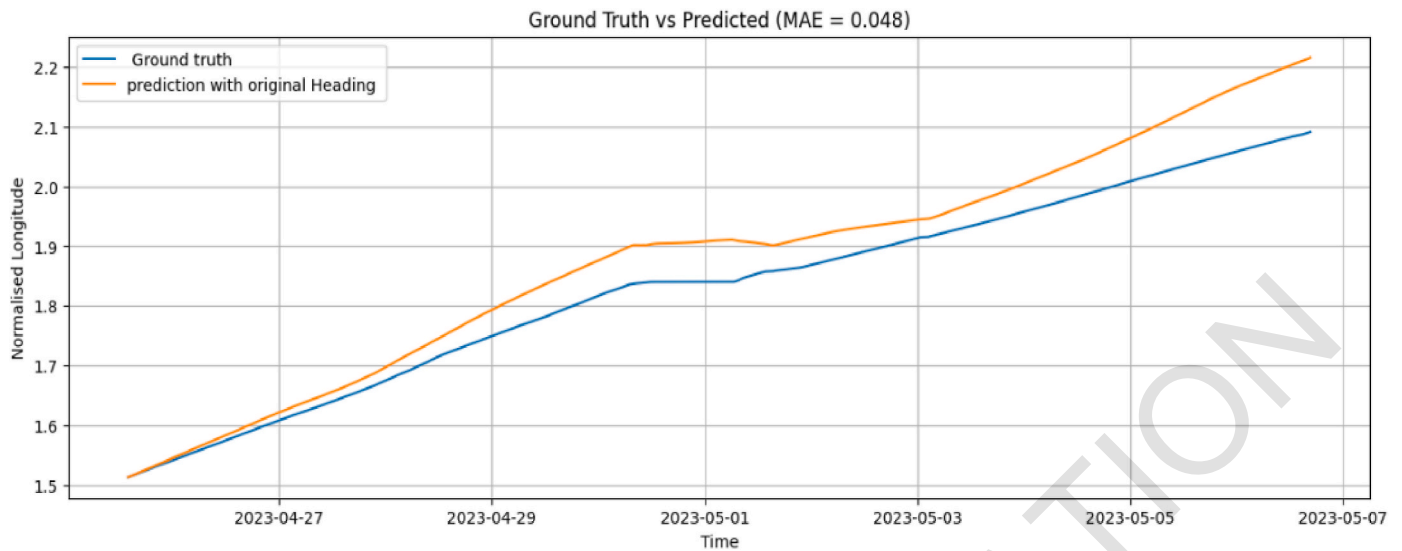


Fig. 13. The predicted longitude in yellow vs the ground truth in blue. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

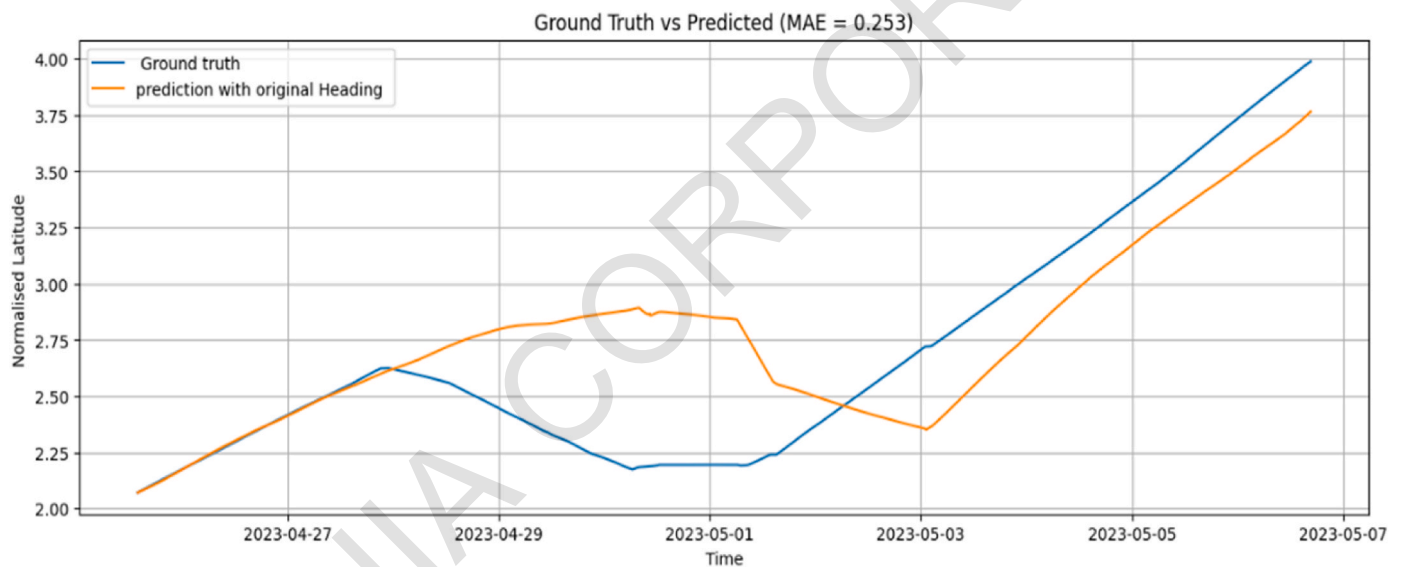


Fig. 14. The latitude prediction from the auto regressive model in yellow and the ground truth in blue. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

introduced in equation (1) subject to constraints such as rudder angle's value is between  $[rudder_{angle_{min}} = -40, rudder_{angle_{max}} = 40]^\circ$  and shaft torque is between  $[shaft_{torque_{min}} = 0, shaft_{torque_{max}} = 80]$  rpm. To validate the performance of the developed model we will use the test case introduced in Fig. 8. Furthermore, one of the objectives of a predictive controller is the ability to look ahead for n number of points. We tested the model with test cases with  $n = 3, 5$  and 10 points in the future. When we used  $n = 10$  points, the performance of the model didn't improve that much, and it was computationally at least 10 times more than when  $n = 3$  or 5. The predicted ship path with  $n = 3$  in yellow and  $n = 5$  in green is shown as well as the desired path in Fig. 15.

Fig. 15 shows that the developed model was capable of following the desired path with very good accuracy. Fig. 16 shows the predicted rudder angles by the models, we can see the rudder is almost following the original rudder that was used in producing the data. For instance, when the model needed to take the first turn, it had to change the rudder angle to almost  $-20^\circ$ , which is  $-8$  in the normalized scale showed in

Fig. 16. The effect of that can be seen in Fig. 17 with a rapid increase in the ship's heading from almost 60 to nearly 100.

Another important observation concerns the controller's anticipatory behaviour. Because the model incorporates a forward-looking mechanism, it occasionally overreacted by correcting the vessel's path earlier and more aggressively than necessary. This tendency highlights a limitation that should be addressed in future work to achieve more optimal and balanced maneuvering. Following the initial correction, the rudder angle was adjusted again to alter the course downward, further illustrating the model's sensitivity to heading changes.

Examining the predicted engine torque (Fig. 18) and surge speed (Fig. 19) reveals that the controller consistently reduces both torque and speed whenever a change in heading occurs. While this demonstrates responsiveness, it cannot be considered optimal behaviour, as the reduction in torque is greater than required. One potential solution would be to introduce a penalty function whenever torque reduction exceeds a defined threshold. However, such an approach must be implemented with caution, since excessive penalization could impair the

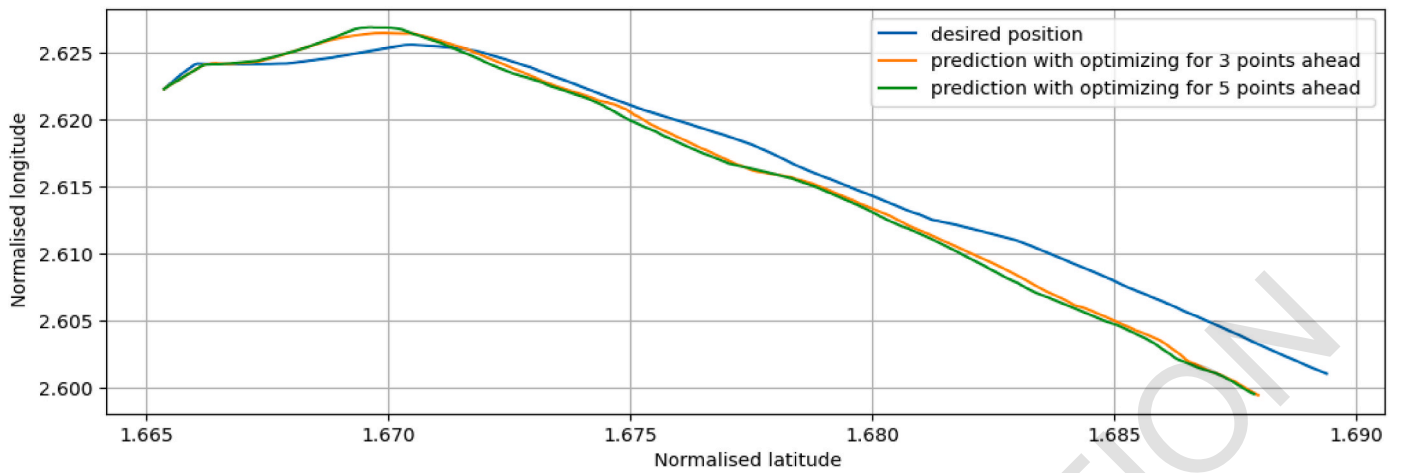


Fig. 15. The desired bath in blue vs the predicted path with  $n = 3$  in yellow and  $n = 5$  in green. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

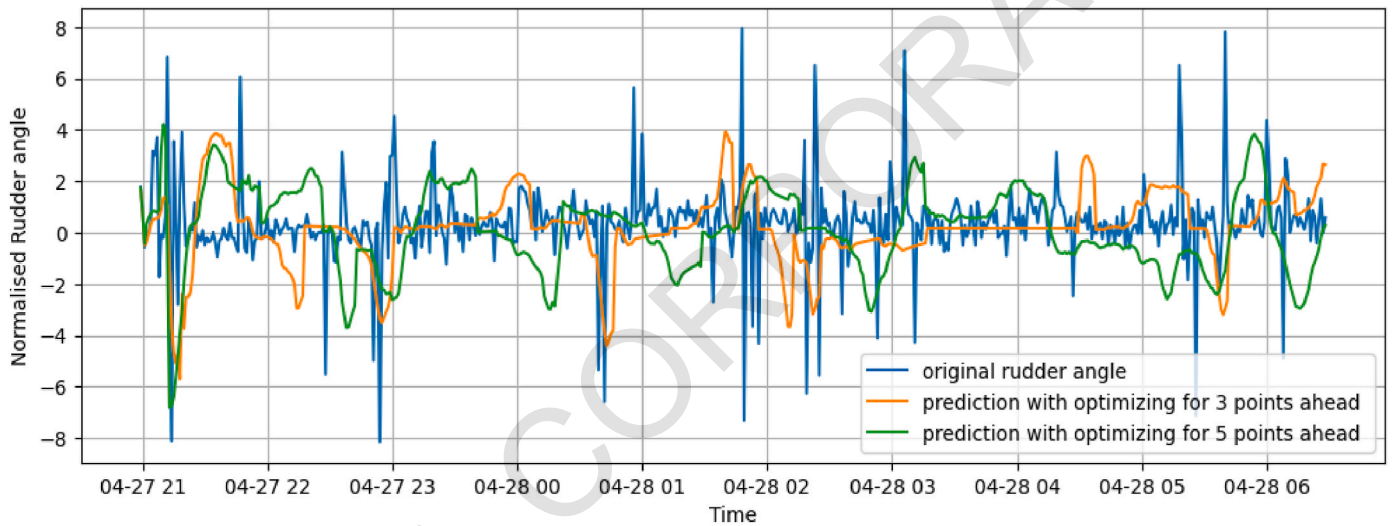


Fig. 16. The comparison of the rudder angles. The original rudder is in blue, the prediction with 3 points in yellow and 5 points in green. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

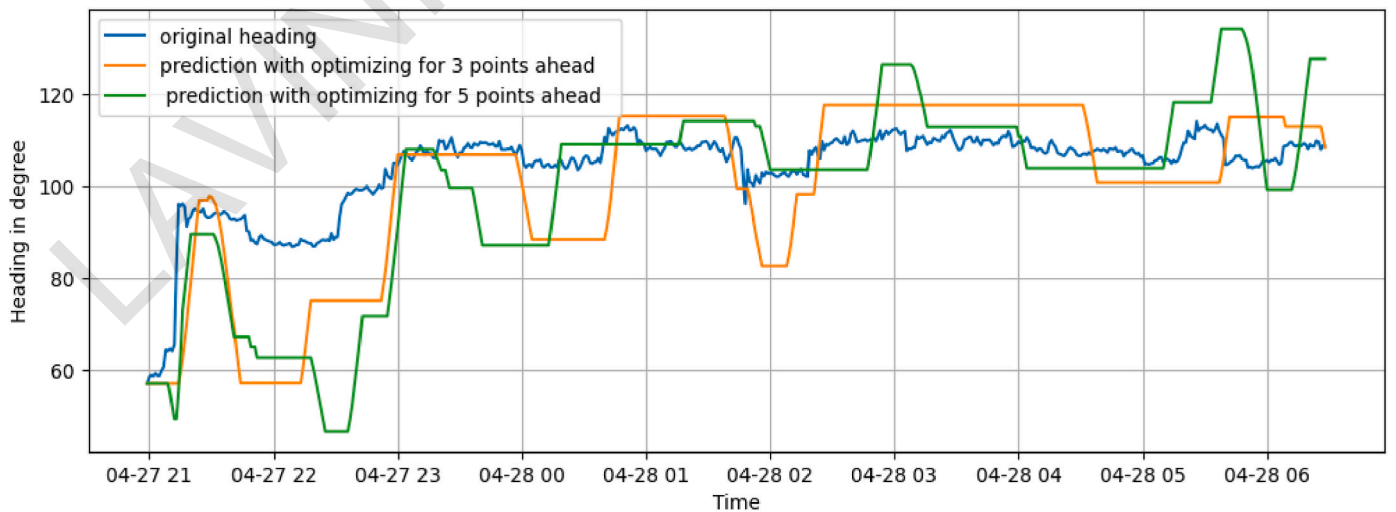


Fig. 17. The original Heading in blue, the produced heading using 3 points in yellow and one using 5 points in green. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

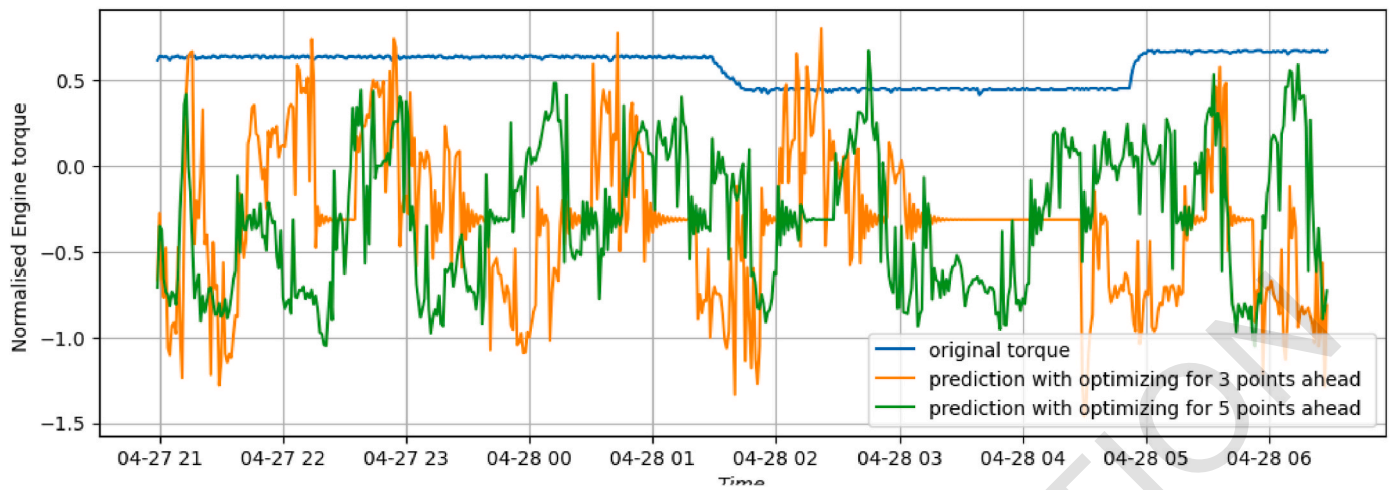


Fig. 18. The Normalized engine torque for optimization with 3 points ahead in yellow, 5 points ahead in green and original torque in blue. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

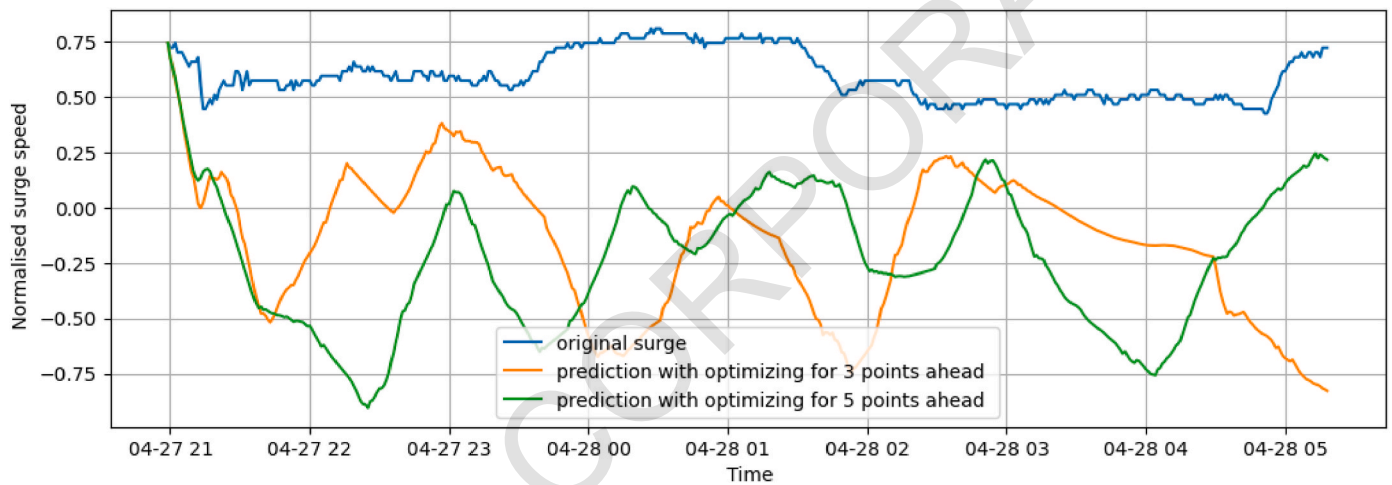


Fig. 19. The predicted surge speed from the optimization model for 3 points in yellow and for 5 points in green and the original surge speed in blue. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

vessel's ability to decelerate or stop effectively when necessary. Finally, the objective of developing such a model is to use it in a Kamsarmax bulk carrier on board to find the optimum control to avoid grounding. Therefore, the computational cost that comes with deploying the model is very important as if it requires a very long period to produce one data point then the model is not efficient enough to be used on board. As we mentioned earlier, we run this model in a CPU which has specifications as in Table 2. The duration of each iteration is shown in Fig. 20.

The computational cost analysis presented in Fig. 20 demonstrates that the developed model is well-suited for real-time deployment. The

majority of iterations required approximately 20 s, with occasional peaks around 40 s and a single outlier reaching 140 s. Given that the model predicts three points in advance—corresponding to a buffer of approximately 3 min—these runtimes are sufficient to ensure timely updates. At each new data point, the model refreshes predictions for points two and three, thereby maintaining continuity without excessive computational overhead. Importantly, these results were obtained using a relatively low-specification CPU. In practice, onboard vessel systems are typically equipped with more powerful CPUs and, in many cases, GPUs, which would further reduce computational costs and enhance efficiency. Consequently, the proposed framework demonstrates strong potential for integration into real-time maritime navigation systems.

**Table 2**  
The characteristics of the computer that was used in this simulation.

Processor	13th Gen Intel(R) Core (TM) i5-1340P, 1900 Mhz, 12 Core(s), 16 Logical Processor(s)
Physical memory	8 GB RAM
Operating system	Windows 11 x64 based system
Virtual memory	24.4 GB
Hard disk	500 GB SSD

### 4.3. The implementation of the developed MPC in grounding avoidance

To validate the model ability to avoid grounding. We are assuming that we have a path for the bulk carrier as shown in white in Fig. 21. And we want to alter this path to avoid the piece of land which it will definitely encounter. The location of the ship is near the location in which the data was collected, thus we can use environmental and sea conditions from the original data.

As we mentioned earlier, the model is trying to minimize an error between a desired state and a current state. Therefore, we need to create

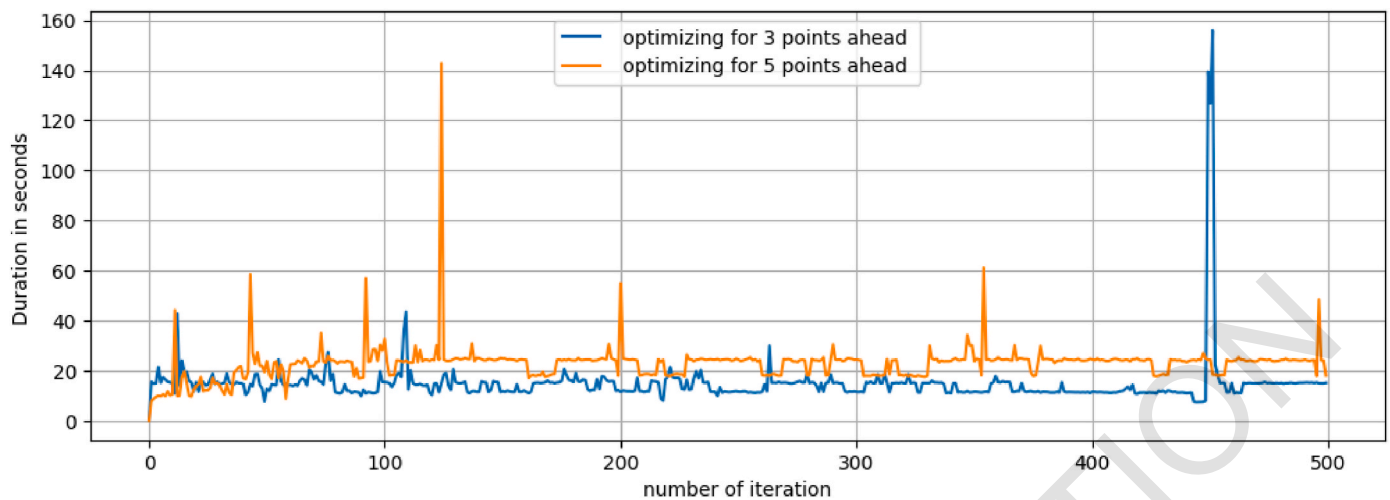


Fig. 20. The duration of running the optimization model with  $n = 3$  and  $n = 5$  points.

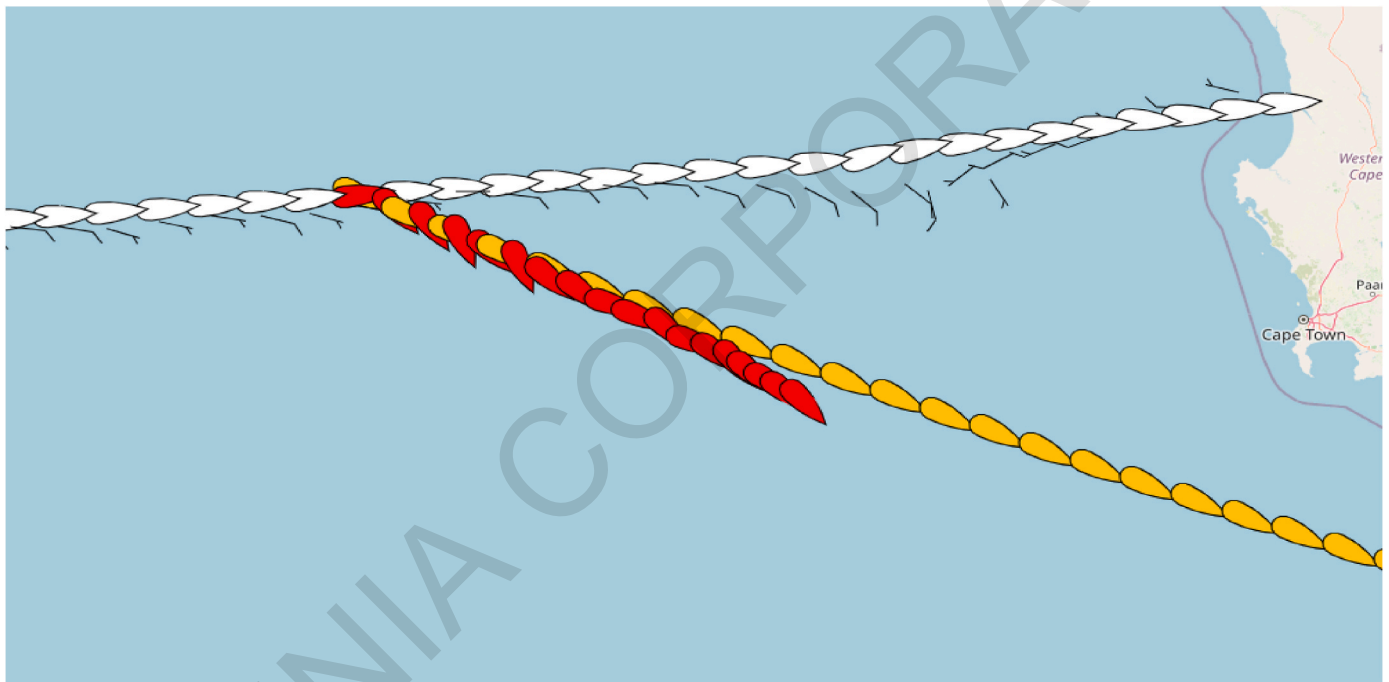


Fig. 21. The original path of the bulk carrier in white, the created desired path using great circle in yellow and the new path from the controller in red. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

and discretize a new path for the model to follow. We did that using the great circle as it is the shortest path between two points on earth (Tao and Du, 2024). Then use the predictive controller to predict the control inputs and motions that will lead for the ship to follow this path. Fig. 22 shows the predicted rudder angle and ship heading to change the ship's course to the new course. It can be seen that at the moment the controller noticed the need to change the course, it changed the rudder rapidly to  $-30^\circ$ , to change the heading from  $80^\circ$  to  $140^\circ$ . Then it tried to keep the ship on its course and didn't change the heading which would lead eventually for the ship to avoid grounding.

In this test case, the predictive controller demonstrated notable adaptability in handling scenarios outside the original dataset. The model exhibited a tendency to reduce vessel speed, stabilizing around 6 knots—approximately 60% of the operational speed—as shown in Fig. 23. However, fluctuations were observed, with speeds varying between 2–4 knots and 9–10 knots. This behaviour contrasts with real-

world operations, where seafarers typically maintain a more constant speed. The observed variation likely stems from two factors: first, the optimization process inherently seeks an “optimal” operating speed rather than replicating human decision-making; second, minor inaccuracies in longitude and latitude predictions as demonstrated before may have contributed to deviations in surge speed estimation.

A key implication is that higher-frequency data collection would enhance the fidelity of the dynamic models, enabling more stable performance and reducing positional error. Despite these limitations, the controller successfully executed rapid course alterations and demonstrated the capacity to take extreme corrective actions when necessary. Importantly, this scenario was not part of the original dataset but was fully constructed for validation purposes. The model's ability to respond effectively to such novel conditions underscores both its generalization capability and the robustness of the developed dynamic framework in predicting ship states under unforeseen circumstances.

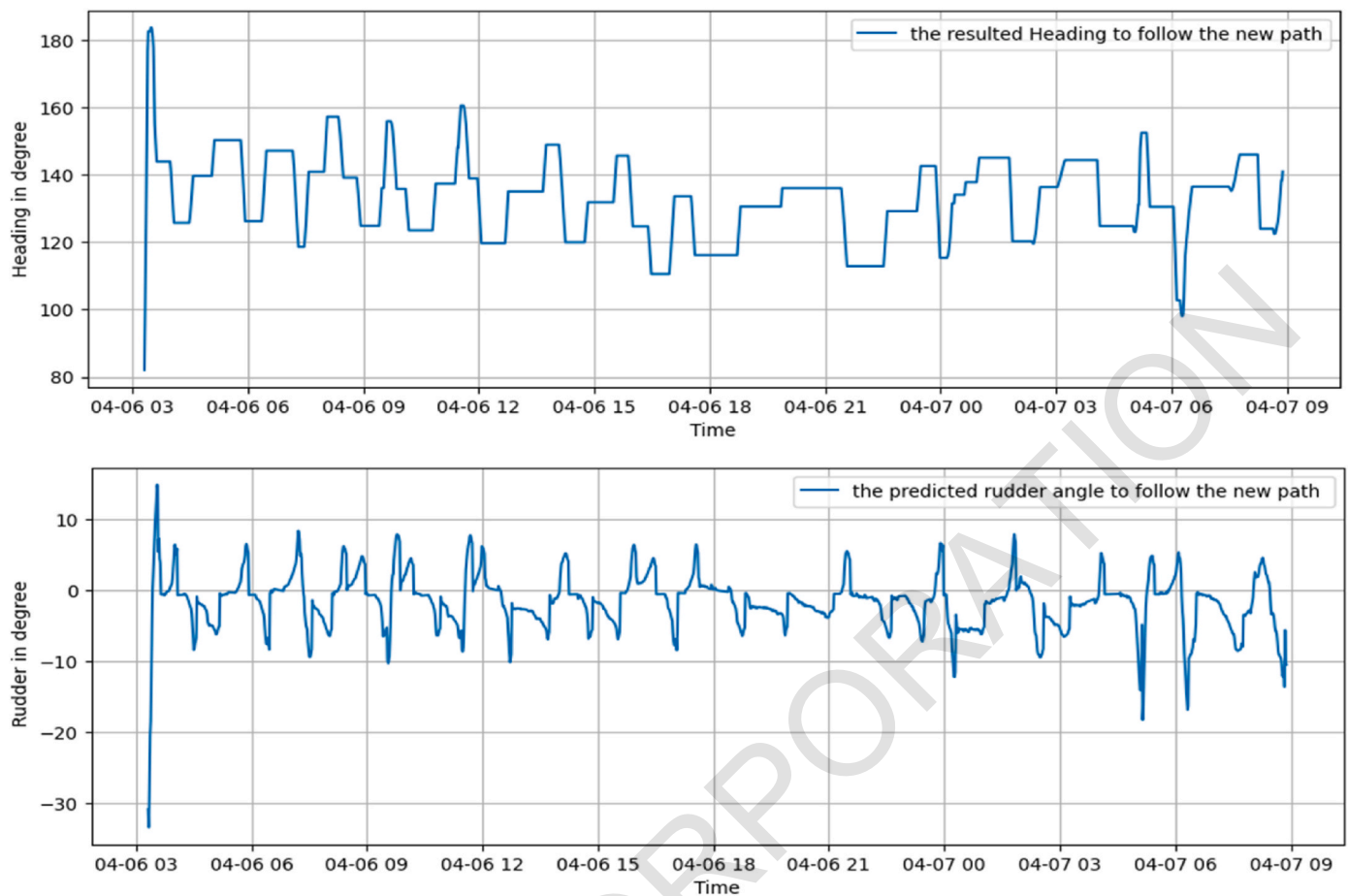


Fig. 22. The rudder angles produced by the controller to generate this new path and the resulting heading.

## 5. Discussion and conclusion

In this paper we developed a prescriptive digital twin able to predict Kamsarmax bulk carriers' maneuvering and control its movement to avoid grounding and ship to ship collision. Within this framework a model predictive controller was developed which consists of two main parts:

- An auto regressive model based on a time series deep learning model developed in the work (Shehata et al., 2025)
- An optimization framework to find the optimal control inputs to control ship's maneuvering.

The model predictive controller is a framework in which the model is making a series of predictions of the future states of the system by varying the control inputs and choose the control inputs that achieve a desired state (Clarke et al., 1987). The accuracy of the model predictive controller (MPC) depends primarily on the tuning of optimization parameters, which influence convergence to an optimal solution. Key parameters include the minimum tolerance for changes in gradients or decision variables prior to termination.

Equally important is the accuracy of the dynamic models employed within the MPC. While state-of-the-art approaches often rely on mathematical models. Such as the work done in (Ming Zhang et al., 2022) which used dynamic models based on the ones developed in ("Maneuvering Theory, 2011; He et al., 2023) developed a model predictive control based on the linear dynamic model developed in (Nomoto et al., 1956). In contrast, the present work adopts a data-driven methodology. Specifically, a time-series deep learning autoregressive model to predict 1 min ahead for surge and sway speeds, rate of turn,

longitude, and latitude. Longitude was measured in kilometers east or west of the prime meridian, while latitude was measured in kilometers north or south of the equator. Moreover, we used the data driven developed model for rate of turn to estimate the heading of the ship as was introduced in equation (3). Consequently, using data driven models has two main advantages:

- High accuracy with computational efficiency compared to numerical solutions of mathematical models.
- The ability to incorporate environmental effects without solving complex differential equations.

Approaching data-driven models requires caution, as they are prone to overfitting and may fail to generalize underlying patterns. To evaluate the proposed dynamic model, a test case was constructed using part of the original dataset, which included extreme maneuvering shown in Fig. 7. The same control inputs from the original dataset were applied to assess how well the model responded to changes in inputs and to quantify deviations between predictions and ground truth. Model accuracy was evaluated using mean absolute error (MAE) and Kullback–Leibler (KL) divergence to compare probability distributions of the developed models against the original data. The metrics shown in Table 3 demonstrated strong agreement:

- MAE values were 0.175 for surge speed, 1.052 for sway, 0.751 for heading, 0.048 for longitude, and 0.253 for latitude.
- NMAE values were 5.6277 for surge speed, 4.899 for sway, 11.24647 for heading, 6.858356 for longitude, and 8.35249 for latitude.
- KL divergence values were 1.064 for surge speed, 0.1385 for sway, and 1.278 for rate of turn

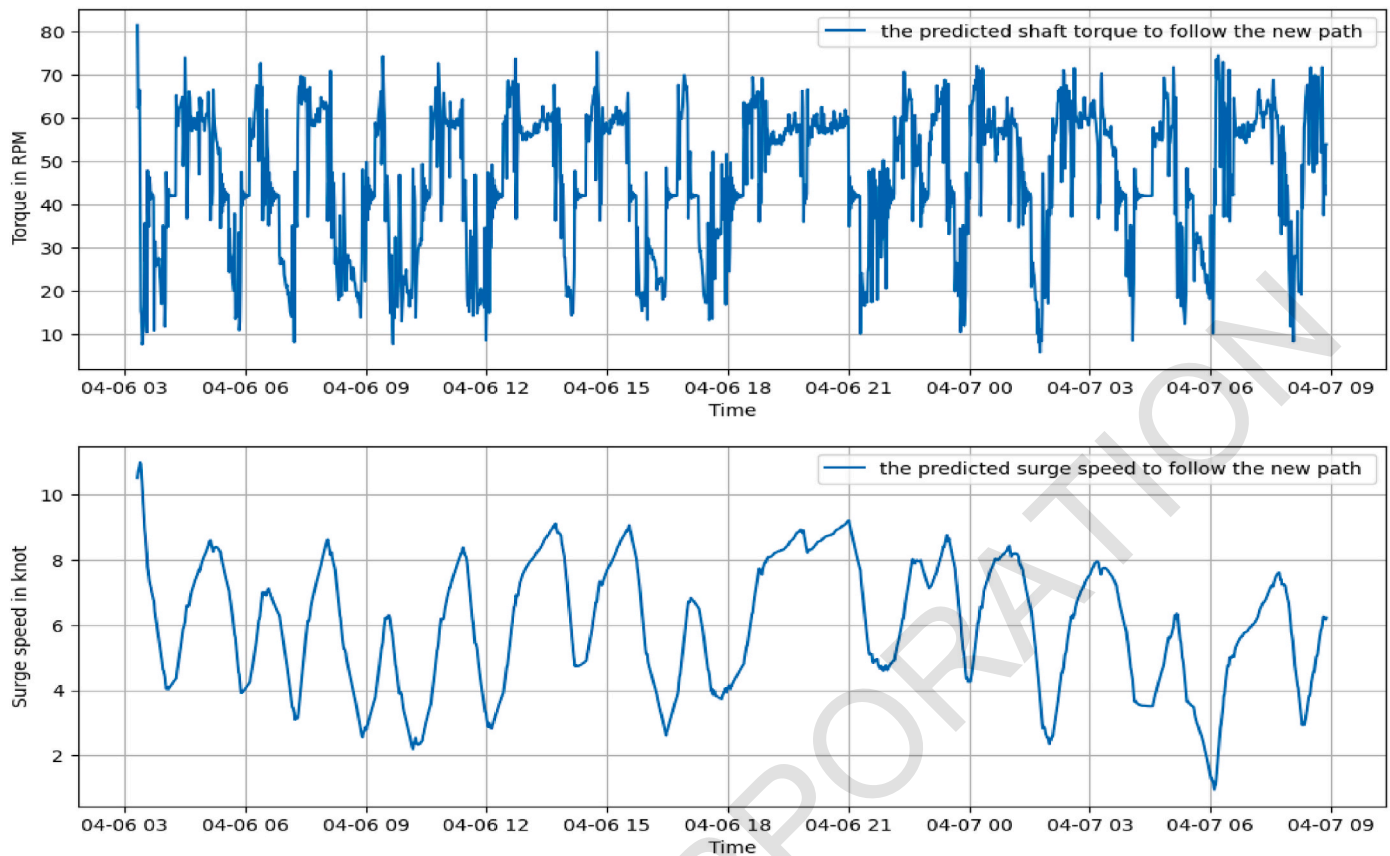


Fig. 23. The produced shaft torque and the resulted surge speed to follow the new path.

Table 3

The MAE, the NMAE between the predicted values and the data ground truth.

Variable	MAE	NMAE %	$D_{kl}$
Surge speed	0.175	5.6277	1.064
Sway speed	1.052	4.899	0.1385
Heading	0.751	11.24647	1.278
Longitude	0.048	6.858356	N/A
Latitude	0.253	8.35249	N/A

While no universal thresholds exist for these metrics, values approaching zero indicate good performance. Visualization further confirmed the model's fidelity. For example, during periods of zero shaft torque, surge and sway speeds dropped to zero, and longitude and latitude remained unchanged Figs. 9–14. Also, NMAE shows that deviation from the bounds never exceeded 11.24 % which is acceptable deviation.

In this paper, a framework for developing a model predictive controller (MPC) was introduced. MPC predicts and utilizes future states over a finite horizon to determine control actions that minimize the error between a reference state and the current state. The prediction horizon may range from a single step to an arbitrarily long horizon. Considering only one step may prevent the model from accounting for future consequences, while a very long horizon can be computationally prohibitive. In this study, horizons of 3 and 5 steps were investigated, with additional simulations at 10 steps. The 10-step horizon proved computationally expensive, while results for 3 and 5 steps showed minimal differences in accuracy. As shown in figures from Figs. 15–20, the computational cost clearly favoured the 3-step horizon. The final trajectory generated by the MPC, compared with the desired reference path, is presented in Fig. 24.

This case was a good way to check the ability of the developed

controller to change the course of movement. The movement started with the ship heading northeast towards increasing both the longitude and latitude and the ship heading was near  $60^\circ$ . Then the ship changes its course and moves east and for the ship to achieve this behavior, it needs to change its heading to almost  $90^\circ$ . To change the heading the ship turned the rudder to angle to almost  $-9^\circ$ . Then the ship returns back to move northeast, and it changes its heading again to near  $60^\circ$  to be able to move upwards. The final turn is when it moves southeast with heading is almost near  $120^\circ$ . This series of events can be seen in Figs. 24 and 15 while Fig. 17 shows the change in the ship heading as a response to the change in rudder angle in Fig. 16

Furthermore, we wanted to test a case that is not in the data but near the location in which the data was collected, thus we can use environmental and sea conditions from the data. In this case the ship is presumed to hit ground if it keeps its current course. This test case is shown in Fig. 21 and the white course is the presumed course in which the ship will hit the ground. In this figure there are two other courses, a yellow course which describes a reference course created by us/the model using the great circle to create the alternative path. The great circle is the shortest path between two points on a sphere, and our presumed desired destination was  $[20, -36]$  in longitude and latitude. This path will allow the bulk carrier to avoid grounding. The model predictive controller created the path in red in Fig. 21. The resulting rudder angle and engine torque were both introduced in Figs. 22 and 23 in this case, the change in heading was greater than in the previous scenario. The optimum rudder angle reached approximately  $-30^\circ$ , which caused the heading to shift to  $180^\circ$ . The model then corrected the course by applying a rudder angle of  $+10^\circ$ , stabilizing the heading near  $140^\circ$ .

The main conclusions driven from the developed work that we can draw are:

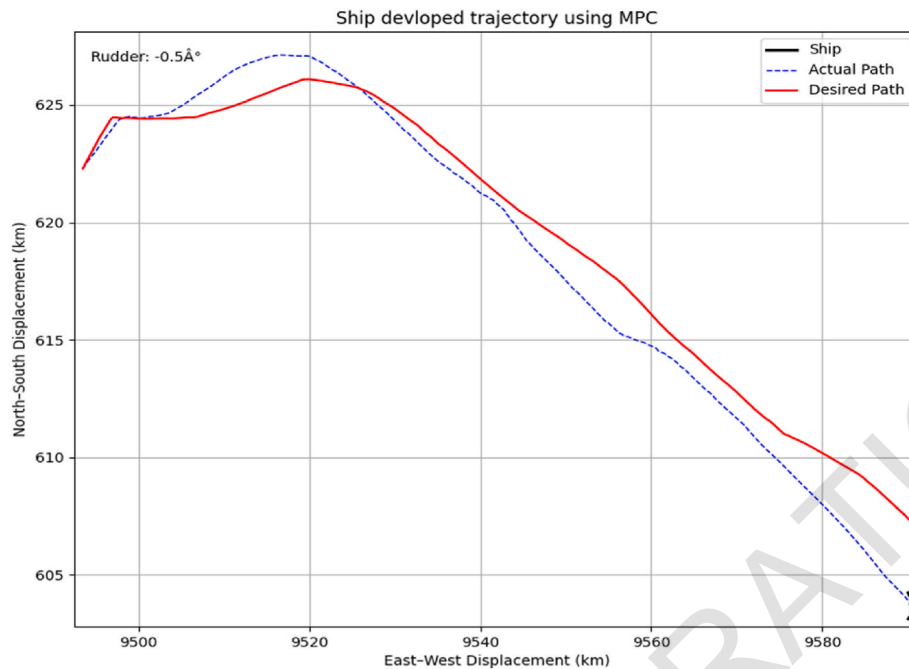


Fig. 24. The ship developed trajectory using MPC in blue dotted line vs the reference path in red. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

- The accuracy of the model predictive controller (MPC) is primarily dependent on the fidelity of the dynamic models. Future work should emphasize accurate heading estimation, including data collection during complex manoeuvres (e.g., zigzag patterns) and at higher sampling frequencies (e.g., every 10 s rather than every minute).
- Deep learning models demonstrate strong capability to produce accurate dynamic representations of ship maneuvering under real environmental conditions.
- The MPC tends to reduce vessel speed, which is a compensatory behaviour. As shown in Fig. 13, the model slightly overestimates longitude, likely due to a slight overestimation of surge speed (Fig. 9). Consequently, the model slows down to maintain alignment with the reference path. This behaviour warrants further investigation, but the primary conclusion is that will be fixed with data collected at higher sampling frequency.
- Data-driven deep learning models within predictive controllers are computationally efficient and capable of accurately describing motion. However, careful attention must be given to the quality and diversity of training data to avoid overfitting.
- When integrated into real-time systems with voyage planning and risk detection, the proposed prescriptive digital twin could detect grounding or collision risks, plan optimal routes, and generate control inputs. It also has potential as a simulation tool for voyage planning.
- The model demonstrated the ability to predict control inputs that is not present in the training data and to optimize paths outside the training set. Nevertheless, experimental validation is required to confirm these responses.

#### CRedit authorship contribution statement

**Ahmed Shehata:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Conceptualization. **Mingyang Zhang:** Writing – review & editing, Investigation, Funding acquisition, Data curation, Conceptualization. **Nikolaos Tsoulakos:** Writing – review & editing, Data curation. **Pentti Kujala:** Writing – review & editing, Supervision.

#### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

#### Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant No.52501441, and National Natural Science Fund for Excellent Young Scientists Fund Program (Overseas) under Grant No.24FAA02373.

#### References

- BV, 2019. NI641 guidelines for autonomous shipping. Marine & Offshore [WWW Document]. URL: <https://marine-offshore.bureauveritas.com/ni641-guidelines-a-utonomous-shiping> (accessed 3.13.26).
- Chollet, F., 2017. *Deep Learning with Python*, first ed. Manning Publications Co., USA.
- Clarke, D.W., Mohtadi, C., Tuffs, P.S., 1987. Generalized predictive control—Part I. The basic algorithm. *Automatica* 23, 137–148. [https://doi.org/10.1016/0005-1098\(87\)90087-2](https://doi.org/10.1016/0005-1098(87)90087-2).
- Darby, M.L., Nikolaou, M., 2012. MPC: current practice and challenges. In: *Control Engineering Practice, Special Section: IFAC Symposium on Advanced Control of Chemical Processes*, 20. ADCHEM 2009, pp. 328–342. <https://doi.org/10.1016/j.conengprac.2011.12.004>.
- Density Estimation, 2017. In: *Kernel Smoothing*. John Wiley & Sons, Ltd, pp. 1–57. <https://doi.org/10.1002/9781118890370.ch1>.
- DNV, 2018. Autonomous and Remotely Operated Ships [WWW Document]. DNV. URL: <https://www.dnv.com/maritime/autonomous-remotely-operated-ships/> (accessed 3.13.26).
- Duarte, F.S.L.G., Rios, R.A., Hruschka, E.R., de Mello, R.F., 2019. Decomposing time series into deterministic and stochastic influences: a survey. *Digit. Signal Process.* 95, 102582. <https://doi.org/10.1016/j.dsp.2019.102582>.
- Fujii, Y., Yamanouchi, H., Mizuki, N., 1974. II—The probability of stranding. *J. Navig.* 27, 239–243. <https://doi.org/10.1017/S0373463300025960>.
- Glaessgen, E.H., Stargel, D.S., 2012. The digital twin paradigm for future NASA and U.S. air force vehicles. In: *Presented at the 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference - Special Session on the Digital Twin*. Honolulu, HI.
- Hänninen, M., 2014. Bayesian networks for maritime traffic accident prevention: benefits and challenges. *Accid. Anal. Prev.* 73, 305–312. <https://doi.org/10.1016/j.aap.2014.09.017>.
- Haugene, B., 1998. *Assessing the Impact of Management and Organizational Factors on the Risk of Tanker Grounding*. Massachusetts Institute of Technology (Thesis).

- He, H., Van Zwijsvoorde, T., Lataire, E., Delefortrie, G., 2023. Model predictive controller for path following ships validated by experimental model tests. *Ocean Eng.* 288, 115971. <https://doi.org/10.1016/j.oceaneng.2023.115971>.
- IMO, 2021. Joint MSC-LEG-FAL Working Group on Maritime Autonomous Surface Ships (MASS) [WWW Document]. URL. <https://www.imo.org/en/mediacentre/meetings/summaries/pages/joint-msc-leg-fal-working-group-on-mass.aspx> (accessed 3.13.26).
- IMO, 2018. MSC-MEPC.2/Circular.12/Rev.2 – Revised Guidelines for Formal Safety Assessment (FSA) for Use in the IMO Rule-Making Process, 9 April 2018).
- Kujala, P., Hänninen, M., Arola, T., Ylitalo, J., 2009. Analysis of the marine traffic safety in the Gulf of Finland. *Reliab. Eng. Syst. Saf.* 94, 1349–1357. <https://doi.org/10.1016/j.res.2009.02.028>.
- Kullback, S., Leibler, R.A., 1951. On information and sufficiency. *Ann. Math. Stat.* 22, 79–86. <https://doi.org/10.1214/aoms/1177729694>.
- Maneuvering Theory, 2011. In: *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons, Ltd, pp. 109–132. <https://doi.org/10.1002/9781119994138.ch6>.
- Menges, D., Rasheed, A., 2024. Digital Twin for Autonomous Surface Vessels: Enabler for Safe Maritime Navigation. <https://doi.org/10.48550/arXiv.2411.03465>.
- Montewka, J., Hinz, T., Kujala, P., Matusiak, J., 2010. Probability modelling of vessel collisions. *Reliab. Eng. Syst. Saf.* 95, 573–589. <https://doi.org/10.1016/j.res.2010.01.009>.
- Morato, M.M., Felix, M.S., 2024. Data science and model predictive control: a survey of recent advances on data-driven MPC algorithms. *J. Process Control* 144, 103327. <https://doi.org/10.1016/j.jprocont.2024.103327>.
- MSC 108, 2024. Maritime safety committee - 108th session. MSC 108), 15–24 May 2024 [WWW Document]. URL. <https://www.imo.org/en/mediacentre/meetingsummaries/pages/msc-108th-session.aspx> (accessed 3.13.26).
- Trust-region methods. In: Nocedal, J., Wright, S.J. (Eds.), 2006a. *Numerical Optimization*. Springer, New York, NY, pp. 66–100. [https://doi.org/10.1007/978-0-387-40065-5\\_4](https://doi.org/10.1007/978-0-387-40065-5_4).
- Introduction. In: Nocedal, J., Wright, S.J. (Eds.), 2006b. *Numerical Optimization*. Springer, New York, NY, pp. 1–9. [https://doi.org/10.1007/978-0-387-40065-5\\_1](https://doi.org/10.1007/978-0-387-40065-5_1).
- Nomoto, K., Taguchi, K., Honda, K., Hirano, S., 1956. ON THE STEERING QUALITIES OF SHIPS (1), 99. *Journal of the Society of Naval Architects of Japan*.
- Parzen, E., 1962. On estimation of a probability density function and mode. *Ann. Math. Stat.* 33, 1065–1076. <https://doi.org/10.1214/aoms/1177704472>.
- Shehata, A., Dashtimanesh, A., 2023. An attempt to predict planing hull motions using machine learning methods. *IOP Conf. Ser. Mater. Sci. Eng.* 1288, 012026. <https://doi.org/10.1088/1757-899X/1288/1/012026>.
- Shehata, A., Zhang, M., Tsoulakos, N., Kujala, P., 2025. A transformer based task execution digital twin of 3-DOF maneuvering of bulk carrier for autonomous maritime systems. *Ocean Eng.* 341, 122797. <https://doi.org/10.1016/j.oceaneng.2025.122797>.
- Tao, J., Liu, Z., Wang, X., Cao, Y., Zhang, M., Loughney, S., Wang, J., Yang, Z., 2024. Hazard identification and risk analysis of maritime autonomous surface ships: a systematic review and future directions. *Ocean Eng.* 307, 118174. <https://doi.org/10.1016/j.oceaneng.2024.118174>.
- Tao, Y., Du, J., 2024. Great circle path planning for USV voyages across oceans with irregular static obstacles. In: 2024 14th Asian Control Conference (ASCC). Presented at the 2024 14th Asian Control Conference (ASCC), pp. 202–207.
- Turlach, B., 1999. *Bandwidth Selection in Kernel Density Estimation: a Review*. Technical Report.
- UNCTAD, 2025. *Shipping Data: UNCTAD Releases New Seaborne Trade Statistics*.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need. <https://doi.org/10.48550/arXiv.1706.03762>.
- Wahlström, M., Hakulinen, J., Karvonen, H., Lindborg, I., 2015. Human factors challenges in unmanned ship operations – insights from other domains. In: *Procedia Manufacturing*, 6th International Conference on Applied Human Factors and Ergonomics (AHFE 2015) and the Affiliated Conferences, 3. AHFE 2015, pp. 1038–1045. <https://doi.org/10.1016/j.promfg.2015.07.167>.
- Yuan, Y., 2015. Recent advances in trust region algorithms. *Math. Program.* 151, 249–281. <https://doi.org/10.1007/s10107-015-0893-2>.
- Zeng, W., Church, R.L., 2009. Finding shortest paths on real road networks: the case for A. <https://doi.org/10.1080/13658810801949850>.
- Zhang, Ming, Hao, S., Wu, D., Chen, M.-L., Yuan, Z.-M., 2022. Time-optimal obstacle avoidance of autonomous ship based on nonlinear model predictive control. *Ocean Eng.* 266, 112591. <https://doi.org/10.1016/j.oceaneng.2022.112591>.
- Zhang, Mingyang, Kujala, P., Hirdaris, S., 2022. A machine learning method for the evaluation of ship grounding risk in real operational conditions. *Reliab. Eng. Syst. Saf.* 226, 108697. <https://doi.org/10.1016/j.res.2022.108697>.
- Zhang, M., Montewka, J., Manderbacka, T., Kujala, P., Hirdaris, S., 2021. A big data analytics method for the evaluation of ship - ship collision risk reflecting hydrometeorological conditions. *Reliab. Eng. Syst. Saf.* 213. <https://doi.org/10.1016/j.res.2021.107674>.
- Zhang, M., Taimuri, G., Zhang, J., Hirdaris, S., 2023. A deep learning method for the prediction of 6-DoF ship motions in real conditions. *Proc. IME M J. Eng. Marit. Environ.* 237, 887–905. <https://doi.org/10.1177/14750902231157852>.