

Obstacle avoidance and trajectory optimisation for an autonomous vessel utilising MILP path planning, computer vision based perception and feedback control

Garofano, V.; Hepworth, M.; Shahin, R.; Pang, Y.; Negenborn, R. R.

DOI

[10.1080/20464177.2024.2330172](https://doi.org/10.1080/20464177.2024.2330172)

Publication date

2024

Document Version

Final published version

Published in

Journal of Marine Engineering and Technology

Citation (APA)

Garofano, V., Hepworth, M., Shahin, R., Pang, Y., & Negenborn, R. R. (2024). Obstacle avoidance and trajectory optimisation for an autonomous vessel utilising MILP path planning, computer vision based perception and feedback control. *Journal of Marine Engineering and Technology*, 23(3), 209-223. <https://doi.org/10.1080/20464177.2024.2330172>

Important note

To cite this publication, please use the final published version (if applicable). Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.



Obstacle avoidance and trajectory optimisation for an autonomous vessel utilising MILP path planning, computer vision based perception and feedback control

V. Garofano, M. Hepworth, R. Shahin, Y. Pang & R. R. Negenborn

To cite this article: V. Garofano, M. Hepworth, R. Shahin, Y. Pang & R. R. Negenborn (06 Apr 2024): Obstacle avoidance and trajectory optimisation for an autonomous vessel utilising MILP path planning, computer vision based perception and feedback control, Journal of Marine Engineering & Technology, DOI: [10.1080/20464177.2024.2330172](https://doi.org/10.1080/20464177.2024.2330172)

To link to this article: <https://doi.org/10.1080/20464177.2024.2330172>



© 2024 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.



Published online: 06 Apr 2024.



[Submit your article to this journal](#)



Article views: 82



[View related articles](#)



[View Crossmark data](#)

Obstacle avoidance and trajectory optimisation for an autonomous vessel utilising MILP path planning, computer vision based perception and feedback control

V. Garofano, M. Hepworth, R. Shahin, Y. Pang and R. R. Negenborn

Maritime and Transport Technology Department, Delft University of Technology, Delft, The Netherlands

ABSTRACT

In this study, we investigated autonomous vessel obstacle avoidance using advanced techniques within the Guidance, Navigation, and Control (GNC) framework. We propose a Mixed Integer Linear Programming (MILP) based Guidance system for robust path planning avoiding static and dynamic obstacles. For Navigation, we suggest a multi-modal neural network for perception, demonstrating the identification of obstacle type, position, and orientation using imaging sensors. Additionally, the paper compares an error-based PID control strategy and a Model Predictive Control (MPC) scheme as well. This evaluation aids in better evaluating their performance and determining their applicability within the GNC scheme. We detail the implementation of these systems, present simulation results, and offer a performance evaluation using an experimental dataset. Our findings, analysed through qualitative discussion and quantitative performance indicators, contribute to advancements in autonomous navigation and the control strategies to achieve it.

ARTICLE HISTORY

Received 23 June 2023
Accepted 9 March 2024

KEYWORDS

GNC scheme; mixed linear-integer programming; computer vision; artificial intelligence; feedback control; model-based design

1. Introduction

Maritime goods transportation has significantly expanded over the past two decades, with its market share doubling to encompass 90% of global commodity transport (Sánchez-Beaskoetxea et al. 2021). Supported by advancements in network infrastructure and vessel efficiency, this modality's growth is expected to continue, rising from \$537.07 billion in 2022 to \$650.21 billion in 2026—an estimated annual growth rate of approximately 5% (Sánchez-Beaskoetxea et al. 2021). However, accompanying this growth is an increased demand for technological improvements. According to statistics from 1999 to 2019, human error contributed to 50% of pollution incidents, 65% of personal injuries, 80% of property damages, and 90% of collisions (Corić et al. 2021). Overconfidence, recklessness, fatigue, and inadequate communication are often cited as leading factors. Autonomous navigation offers potential solutions to these issues, potentially enhancing overall safety and efficiency in maritime goods transportation.



To enable autonomous navigation, an autonomous vessel typically integrates three distinct modules: the Guidance, Navigation, and Control (GNC) system, as illustrated in Figure 1. This paper explores the potential of employing advanced techniques within the GNC subsystems to augment the current standard practices and state-of-the-art in autonomous navigation, specifically in the areas of obstacle vision recognition and avoidance.

The Guidance System in an autonomous vessel is tasked with global and local path planning. The application of a Mixed Integer Linear Programming (MILP) algorithm, for instance, enables the path-planning optimiser to update the global path to circumvent local obstacles. This technique, which requires the formulation of a cost function and a set of constraints to minimise path deviation amidst static and dynamic obstacles, has proven effective for global trajectory planning across different fields (Schouwenaars et al. 2001; Coleman 2024). This paper further explores its use in the maritime domain. Alternative approaches, based on deep learning

algorithms were taken into consideration as well (Parkes et al. 2018; Sivaraj et al. 2023). However in the context of autonomous ship navigation, particularly for obstacle avoidance, MILP offers distinct advantages over deep learning approaches. MILP provides deterministic outcomes crucial for safety and regulatory compliance in maritime operations. It excels in efficiently handling multiple constraints and objectives, ensuring optimal pathfinding adaptable at various navigational scenarios. Unlike deep learning, MILP requires less computational power for real-time decision-making and is not heavily reliant on large datasets for training, making it more adaptable to unpredictable maritime environments. Additionally, MILP's robustness against uncertainty, along with its capacity to incorporate safety regulations and its potential of scalability, makes it a more suitable and reliable choice for the specific application in this studies.

Within a Navigation System, perception plays a vital role in autonomous sailing. Obstacles must be detected, and their position and orientation determined (Kim et al. 2018). Though point cloud clustering is a common perception approach used by autonomous vehicles (Khalid et al. 2013; Kato et al. 2015; Wang et al. 2019; Choy et al. 2015), this research proposes a novel multi-modal neural network architecture to provide high-level situational awareness for collision avoidance purposes. This approach utilises a CNN for object detection from colour images and processes depth data to provide 3D data. The data is then fused to offer a cropped birds-eye view input to a secondary neural network, which is trained specifically to extract the orientation of an obstacle—in this case, a vessel.

Control strategy design forms the third pillar in the GNC framework. In the context of autonomous navigation, its primary role is to measure the internal state of a vessel system (i.e. position, velocity, orientation) and minimise the error between the measurement and the reference provided by the guidance module. Despite the prevalence of Proportional, Integral, and Derivative (PID) controllers in industrial automation (accounting for 95% of closed-loop control algorithms) due to their simple implementation and robustness

CONTACT V. Garofano  v.garofano@tudelft.nl  Maritime and Transport Technology Department, Delft University of Technology, Mekelweg 2, 2628 CD, Delft, The Netherlands

© 2024 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.

This is an Open Access article distributed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives License (<http://creativecommons.org/licenses/by-nc-nd/4.0/>), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited, and is not altered, transformed, or built upon in any way. The terms on which this article has been published allow the posting of the Accepted Manuscript in a repository by the author(s) or with their consent.

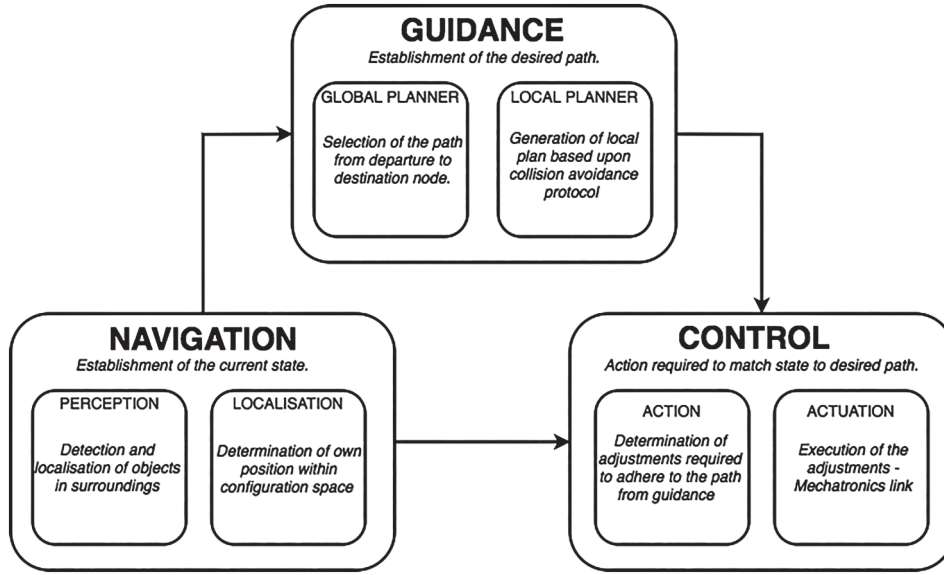


Figure 1. Guidance navigation and control system framework (Hepworth et al. 2021).

against external disturbances (Dey and Sen 2020), this paper investigates the Model Predictive Control an optimisation-based control technique—as an alternative, used to regulate a process while fulfilling a set of constraints (Haseltalab et al. 2020).

While considering the potential of deep learning-based control methods (Parkes et al. 2022; Ye et al. 2023; Deraj et al. 2023), this study has opted for these classical control strategies, recognising their established efficacy in managing the dynamic and uncertain conditions characteristic of an autonomous vessel navigation domain. Their robust frameworks enable handling of variable conditions, a challenge where deep learning approaches often require extensive, context-specific training data to achieve similar levels of reliability. Additionally, the reliance of deep learning on large and diverse datasets for training is a significant limitation, especially in fields where such datasets are scarce or challenging to compile. The computational intensity and time requirements for training deep learning models further enhance this issue, rendering them less practical for the specific applications investigated in this paper.

The rest of this paper is organised as follows: Section 2 introduces the proposed GNC scheme, with each sub-system discussed in its own sub-section. Section 3 presents the results and performance evaluations of the individual sub-systems, gleaned from both simulated and experimental testing. Finally, Section 4 summarises the paper's findings and offers conclusions and recommendations for future work. This research paper is an expanded and enhanced version of our previous work, which was presented at the International Ship Control Systems Symposium (iSCSS) in Delft, The Netherlands, in November 2022 (Garofano et al. 2022).

2. Methodology

2.1. Guidance strategy

The general formulation of a Mixed Integer Linear Programming problem can be formulated as (Vielma 2015; Richards et al. 2002):

$$\begin{aligned}
 \min \quad & Z = \mathbf{c}^T \mathbf{x} \\
 \text{s.t.}, \quad & \\
 & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\
 & \mathbf{x} = (x_1, \dots, x_n, i_1, \dots, i_k, w_1, \dots, w_j)^T \\
 & x_n, w_j \in \mathbb{R}, \quad i_k \in \mathbb{Z}
 \end{aligned} \tag{1}$$

Here, \mathbf{x} denotes the vector containing the variables to be optimised and \mathbf{c}^T corresponds to a coefficient vector. The optimisation vector is comprised of continuous real variables x_n and integer variables i_k and a number of slack variables are introduced w_j . The slack variables are introduced to transform hard constraints into soft constraints that are penalised for in the cost function. This proves useful in the context of path planning as will become evident. The linear matrix inequality $\mathbf{A}\mathbf{x} \leq \mathbf{b}$ corresponds to the constraints on the optimisation variables.

In the context of path planning, the continuous variables that are to be optimised are defined assuming an a-priori planned path for the vessel denoted by the vector $\eta_{ref}(t)$ and the optimisation solution $\eta_{sol}(t)$. One may define the state tracking error of the solution by relationship in Equation (2). For maneuvering purposes, a vessel's pose is uniquely defined by its 2-dimensional position in NED-coordinates and its heading angle as in Equation (3). For the purpose of path deviation minimisation, the heading angle is neglected and the error state is limited to the reduced 2-dimensional error vector defined by Equation (4).

$$\mathbf{x}_n(t) = \eta_{sol}(t) - \eta_{ref}(t) \tag{2}$$

$$\eta(t) = (Y \ X \ \psi)^T \tag{3}$$

$$\mathbf{x}_n(t) = \begin{pmatrix} |Y_{sol}(t) - Y_{ref}(t)| \\ |X_{sol}(t) - X_{ref}(t)| \end{pmatrix} \tag{4}$$

This l_2 norm may be transformed to a l_1 norm suitable for linear optimisation by introducing the augmented error system in Equation (5). This error system can be transformed into a system of Linear Matrix Inequalities (LMI) that are to be optimised. For the purpose of path deviation minimisation, these are transformed into the LMIs in Equation (6) as the first set of constraints used in the optimisation problem. Where $w_{\tilde{Y}}(t)$ and $w_{\tilde{X}}(t)$ represent the slack variables that are introduced at every time step.

$$\begin{bmatrix} \tilde{Y}(t) \\ -\tilde{Y}(t) \\ \tilde{X}(t) \\ -\tilde{X}(t) \end{bmatrix} = \begin{bmatrix} (Y_{sol}(t) - Y_{ref}(t)) \\ -(Y_{sol}(t) - Y_{ref}(t)) \\ (X_{sol}(t) - X_{ref}(t)) \\ -(X_{sol}(t) - X_{ref}(t)) \end{bmatrix} \tag{5}$$

$$\begin{bmatrix} 1 & 0 & -1 & 0 \\ -1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & -1 & 0 & -1 \end{bmatrix} \begin{bmatrix} \tilde{Y}(t) \\ \tilde{X}(t) \\ w_{\tilde{Y}}(t) \\ w_{\tilde{X}}(t) \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (6)$$

Hence, the hard constraints enforcing the path deviation are transformed into soft constraints by introducing the slack variables. These slack variables are thus included in the optimisation vector (4). By setting the values in the coefficient vector as stated in (1) to appropriate values, one can minimise for the value of the slack variables and thus for path deviation. For this purpose, these coefficient values are set to positive, real values. Moreover, the vector elements in \mathbf{c}^T corresponding to the errors themselves $\tilde{Y}(t)$ and $\tilde{X}(t)$ are set to zero.

Having established the framework for path deviation minimisation, additional constraints can be introduced. For the purpose of enforcing a maximum speed constraint, LMI in Equation (7) is formulated for the linearised maximum speed constraints. The maximum speeds in both positive and negative surge and sway directions are assumed to be known in this application. Furthermore Δt is a designed parameter that defines the time interval between solutions of the optimisation algorithm. The value of this time interval subsequently determines the update frequency of the algorithm and must be chosen accordingly.

$$\begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} Y_{sol}(t + \Delta t) - Y_{sol}(t) \\ X_{sol}(t + \Delta t) - X_{sol}(t) \end{bmatrix} \leq \begin{bmatrix} (V_{Y+}^{\max} * \Delta t) - (Y_{ref}(t + \Delta t) - Y_{ref}(t)) \\ -(V_{Y-}^{\max} * \Delta t) + (Y_{ref}(t + \Delta t) - Y_{ref}(t)) \\ (V_{X+}^{\max} * \Delta t) - (X_{ref}(t + \Delta t) - X_{ref}(t)) \\ -(V_{X-}^{\max} * \Delta t) + (X_{ref}(t + \Delta t) - X_{ref}(t)) \end{bmatrix} \quad (7)$$

Central to the goal of this guidance system is the incorporation of an obstacle avoidance strategy which is introduced in the form of a MILP problem. Once the positions of static and dynamic obstacles are known, the avoidance constraints can be formulated. The space to be avoided is formulated here such that a rectangular box is constructed around obstacles, with $i_k \in [0, 1]$ and $X_{obs}^{\min}(t)$ and $Y_{obs}^{\min}(t)$ denoting the minimum coordinates of the obstacle. For safety purposes, obstacles are to be avoided by some safety distance, this is taken into account in the strategy proposed by adding a constant M , in metres, that correspond to a safe margin, i.e. the area around the obstacle that is identified as a no-navigation zone. The obstacle boundaries can then be extended to act as a safety factor as required. The constraint in Equation (8) is added to guarantee that at least one of the constraints is satisfied and a feasible path in any of the given directions can be generated at all times. This results in the LMI defined in Equation (9).

$$i_1 + i_2 + i_3 + i_4 \leq 3 \quad (8)$$

$$\begin{bmatrix} 1 & 0 & -M & 0 & 0 & 0 \\ -1 & 0 & 0 & -M & 0 & 0 \\ 0 & 1 & 0 & 0 & -M & 0 \\ 0 & -1 & 0 & 0 & 0 & -M \\ 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} Y_{sol}(t) \\ X_{sol}(t) \\ i_1 \\ i_2 \\ i_3 \\ i_4 \end{bmatrix} \leq \begin{bmatrix} \tilde{Y}_{obs}^{\min}(t) - Y_{ref}(t) \\ -\tilde{Y}_{obs}^{\max}(t) + Y_{ref}(t) \\ \tilde{X}_{obs}^{\min}(t) - X_{ref}(t) \\ -\tilde{X}_{obs}^{\max}(t) + X_{ref}(t) \\ 3 \end{bmatrix} \quad (9)$$

The linear programming framework also requires an equilibrium solution in the form of a Linear Matrix Equality (LME). In the use case presented here, the equilibrium solution that can be used is the current state of the vessel, i.e. $\mathbf{x}(t=0) = \mathbf{x}_0 = (Y_0 \ X_0)^T$. This results in the Linear Matrix Equality in Equation (10) with η_Y and η_X corresponding to the current vessel position.

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{Y}_0 \\ \tilde{X}_0 \end{bmatrix} = \begin{bmatrix} \eta_Y - Y_{ref} \\ \eta_X - X_{ref} \end{bmatrix} \quad (10)$$

These separate matrices can now be concatenated into one single LMI and formulated as a single MILP problem as in Equation (11). The main objective of the MILP approach proposed is to minimise the path deviation and avoiding obstacles within a finite time horizon. The final product yields the system of LMIs in Equation (12), with $k\Delta t$ representing the time horizon. Using an MILP solver, a solution for the values of $[\mathbf{x}(t), \dots, \mathbf{x}(t + k\Delta t)]$ can be calculated. These are then used to generate a new set of way points that serve as input for the control system.

$$A_{total}(t)\mathbf{x}(t) \leq \mathbf{b}_{total}(t) \quad (11)$$

$$\begin{bmatrix} A_{total}(t) \\ A_{total}(t + \Delta t) \\ \vdots \\ A_{total}(t + k\Delta t) \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{x}(t + \Delta t) \\ \vdots \\ \mathbf{x}(t + k\Delta t) \end{bmatrix} \leq \begin{bmatrix} \mathbf{b}_{total}(t) \\ \mathbf{b}_{total}(t + \Delta t) \\ \vdots \\ \mathbf{b}_{total}(t + k\Delta t) \end{bmatrix} \quad (12)$$

2.2. Navigation system

Situational awareness is achieved through the implementation of a vision system specifically configured to generate data pertinent to the path planning technique, being capable of providing a constant update of environmental conditions in the presence of both static and dynamic obstacles. The perception approach presented hereon is developed to complement the capacity of the MILP based guidance system to make use of a constantly updating environment, whilst further providing a reliable estimation of orientation for both static and dynamic obstacles alike.

The approach to perception proposed in this paper comes in the form of a multi-modal neural network whose architecture is presented in Figure 4. A single stereovision device is deployed in this system, namely the Intel® RealSense™ D435i which provides both colour images through the RGB feed and stereo-depth frames via a depth feed. These two feeds act as the input to the perception system where they are subsequently processed individually with the results then being fused. Both feeds are configured to a resolution of 848x480 and 30 frames per second. It is worth noting that other sensors capable of providing depth data could also be used within this network structure.

The colour frames supplied by the RGB feed are utilised for obstacle detection, achieved through the application of a Conventional Neural Network trained to detect obstacles within the experimental environment, referred to in this work as the 'VesselNet'. The VesselNet is built upon the Faster R-CNN framework, with an Inception 101 feature extractor, with proposals limited to 50 to allow for an adequate trade-off between speed and accuracy. For further details on this algorithm and the training choices, the reader is directed to the work of (Huang et al. 2016). Whilst this specific model, therefore the details of the algorithm shall not be any further divulged in this work. The key outputs of this network are the bounding box and classification of each obstacle detected in the frame. The deep learning methodology, particularly concerning the training, testing, and validation splits, as well as the variety and generalisation of the training set, takes inspiration from the approach used in (Hepworth



Figure 2. 'Tito Neri' model scale tugboat.



Figure 3. 'Grey Seabax' model scale offshore vessel.

et al. 2021). In this model, similar to the one presented in the other study, a dataset of 1500 annotated colour frames was used for training. This dataset includes a diverse range of model-scale tugboats and offshore service vessels, as depicted in Figures 2 and 3. The dataset was divided into training, validation, and test segments to ensure a robust and general model, in line with best practices established in the field.

In parallel to the obstacle detection procedure, the depth feed is processed through the application of filters to improve the usability of the data. A breakdown of the filter process can be seen in Figure 4, where the steps are encompassed by the dashed blue box. As the depth frames generated by the stereovision setup are not inherently aligned with the images from colour sensor, this must be achieved pragmatically. The frames are aligned by translating the depth pixels to a synthetic viewport which matches the field of view of the colour sensor. This allows for depth to be directly extracted from a ROI identified in a colour frame. This aligned depth frame then requires some further processing, starting with a decimation filter. Decimating the depth is a sub-sampling technique to smooth the data, improving later point-cloud visualisation. Spatial and temporal filters are applied thereafter to improve edge sharpness and remove noise. Finally any dead-pixels that remain in the depth frame after

decimation are accounted for by a hole-filling operation. This operation replaces each instance of a dead pixel by the largest depth value from its surrounding neighbours, assigning it an estimated value.

In order to generate a 2D birds-eye-view (BEV) plot as an input for the second neural network, the obstacle detection data and the depth data need to first be fused and processed. Figure 4 shows a breakdown of the fusion procedure in the dashed red box. The obstacle bounds supplied by the VesselNet are utilised to extract the region of interest (ROI) from the processed depth data creating a cropped depth matrix. A point-cloud is then generated by projecting each pixel to a point within 3D space with the known device intrinsics. Median clustering is applied to the raw point-cloud to remove any outliers. This is specifically achieved through the exclusion of points outside a depth radius of half the vessel beam from the median point. This vessel beam is estimated from a look up table based upon the vessel classification from the VesselNet. Finally, the fusion stage generates a BEV plot of the processed point-cloud data (Beltran et al. 2018). This BEV plot is created in the form of a 3-channel image, encoding the vertical measurement that was lost when reducing to two dimensions as a colour-map.

The second Convolutional Neural Network, 'BearingNet' is tasked with extracting an estimate of yaw angle from the generated BEV plot. This network is built upon the RetinaNet framework (Lin et al. 2017, 2020) with the ResNet50 feature extractor, which on top of conducting traditional 2D object detection, is also endowed with the capacity to infer object orientation. This capacity is achieved by generating anchor boxes at various angles whilst still applying the usual factors of scale and aspect ratio. This addition not only adds computational load to the object detector, but also alters the method for calculating intersection over union (IoU). The problem of IoU calculation is solved by applying a sequential cutting method which recursively assesses the intersection of edges between two boxes. The former issue of computational load is rectified through the parallel processing using the CUDA integration available within the NVIDIA object detection toolkit with which this network was trained. As with the first object detection neural network, this 'BearingNet' provides bounding box coordinates and a classification as well as the new angle attribute. The bounding box angle (θ) produced is measured anti-clockwise from the x-axis of the BEV plot and constrained between $-\pi$ and π . At this stage of research, the BearingNet is trained on a dataset of 750 BEV plot images specific to the offshore model-scale vessel only.

The final procedure in the perception approach is to post-process the output of the BearingNet and the point-cloud from the fusion stage. This is processing involves a simple translation from the BEV plot pixel frame to the real-world coordinate frame and transforming the angle to align with that of the autonomous vessel. The final outputs of the multi-modal network are the coordinates of the obstacle bounds, the bearing of the obstacle and the vessel classification. All three attributes of particular use to a guidance system conducting collision avoidance.

2.3. Control design

In this section, we outline the Control Design methodology for our Guidance, Navigation, and Control (GNC) scheme. We leverage a Model-Based design approach, exploiting analytical solutions for the control problem while examining the dynamics of a ship system. Our focus will be on two distinct control methods.

The first method stems from classical control theory, wherein we'll scrutinise the fundamental characteristics of a vessel's mathematical model and its interaction with the derived control law. The second method is rooted in the field of optimal control, specifically

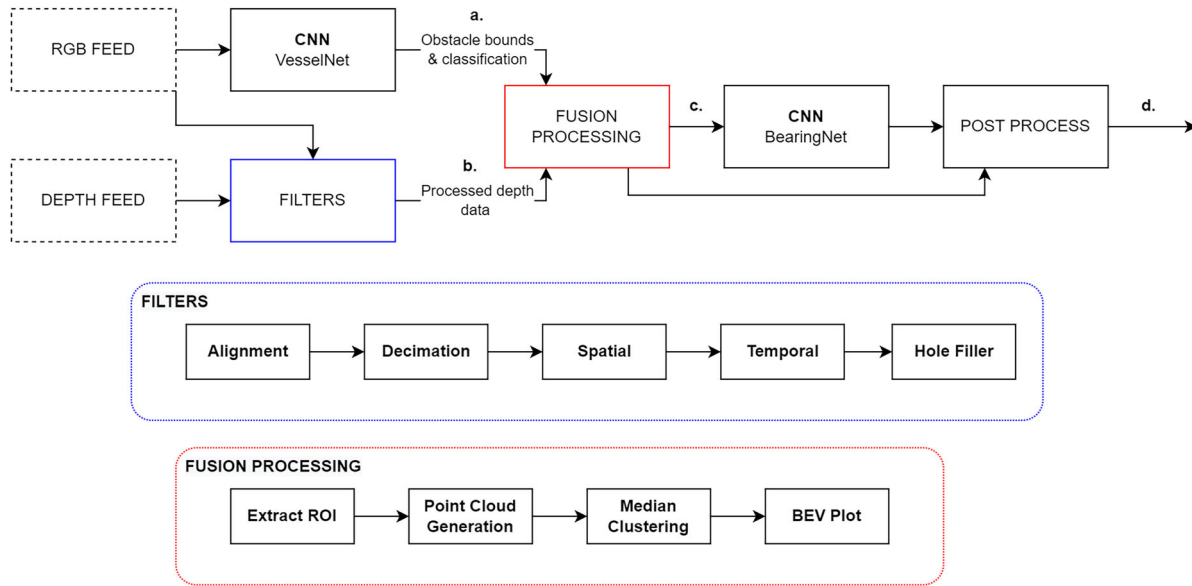


Figure 4. Multimodal neural network architecture for perception tasks.

Model Predictive Control (MPC). Here, the presence of a cost function and constraints allows the designer to better balance between performance and control effort.

The dynamics of a vessel navigating on water, subjected to environmental disturbances like wind, waves, and currents, involves six degrees of freedom. The equations of motion can be obtained using either the Newton-Euler or the Lagrange equations.

A vessel hull of constant mass m and centre of gravity (x_g, y_g, z_g) can be described by the following set of coupled differential equations (Fossen 2011):

$$\begin{cases} m[\dot{u} - vr + wq - x_g(q^2 + r^2) + y_g(pq - \dot{r}) + z_g(pr + \dot{q})] = X \\ m[\dot{v} - wp + ur - y_g(r^2 + p^2) + z_g(qr - \dot{p}) + x_g(qp + \dot{r})] = Y \\ m[\dot{w} - uq + vp - z_g(p^2 + q^2) + x_g(rp - \dot{q}) + y_g(rq + \dot{p})] = Z \\ I_x \dot{p} + (I_z - I_y)qr - (\dot{r} + pq)I_{xz} + (r^2 - q^2)I_{yz} + (pr - \dot{q})I_{xy} + m[y_g(\dot{w} - uq + vp) - z_g(\dot{v} - wp + ur)] = K \\ I_y \dot{q} + (I_x - I_z)rp - (\dot{p} + qr)I_{xy} + (p^2 - r^2)I_{zx} + (qp - \dot{r})I_{yz} + m[z_g(\dot{u} - vr + wq) - x_g(\dot{w} - uq + vp)] = M \\ I_z \dot{r} + (I_y - I_x)pq - (\dot{q} + rp)I_{yz} + (q^2 - p^2)I_{xy} + (rq - \dot{p})I_{zx} + m[x_g(\dot{v} - wp + ur) - y_g(\dot{u} - vr + wq)] = N \end{cases} \quad (13)$$

In these equations, X, Y, Z, K, M, N denote the external forces and moments, u, v, r, p, q, w represent the linear and angular body velocities, and m, I_{**} symbolise the mass and inertia tensor of the vessel. This model is one of the most comprehensive to describe the time-domain dynamic evolution of a marine craft.

However, a vectorial representation of the equations of motion could be adopted to better exploit the physical properties of the model. This representation offers distinct advantages, such as revealing specific mathematical system properties and allowing for a reduction in the number of coefficients required for control by rearranging the equation to yield semi-definite positive and skew-symmetric matrices (Fossen 2011; Sciavicco et al. 1998).

The compact vessel model can then be expressed as:

$$M\dot{v} + C(v)v + Dv = \tau. \quad (14)$$

In this equation, v represents the generalised velocities vector $v = [u, v, r, w, p, q, r]'$, $\tau = [X, Y, Z, K, M, N]$ stands for the generalised force vector acting on the craft, and $M, C,$ and D are respectively the

rigid-body inertia matrix, the Coriolis and centripetal forces matrix, and the Damping matrix.

An alternative to the 6DoF vector formalism, shown in Equation (14), is to apply manoeuvring theory for a three degrees of freedom (3DoF) representation of a marine vessel's motion. This 3DoF model focuses on the horizontal motion of a ship in surge, sway, and yaw, so the state vector becomes $v = [u, v, r]'$. Therefore, the motion associated with heave, roll, and pitch is ignored, i.e. $w = p = q = 0$. This simplified model garners our attention in this paper, since our primary control design goal is enabling a vessel to autonomously navigate a predetermined path on an xy Cartesian plane.

The section that follows delves into a more comprehensive analysis of ship dynamic manoeuvring based on the 3DoF model. We aim to demonstrate how a system of equation of different mathematical models, each with its unique mathematical properties, can be used to formulate an appropriate control law. We can categorise the different sailing scenarios of a surface vessel as follows:

- (1) **Unmooring:** This is the initial stage where a ship begins its departure. Here, surge, sway, and yaw velocities are minimal and approximately equal: $u = v = r \approx 0$.
- (2) **Low-velocity manoeuvrings:** This stage precedes actual navigation towards the destination, typically taking place in a harbour environment. The ship executes slow maneuvers to orient towards its destination: $u = v = r < \varepsilon$.
- (3) **Sailing at cruise speed:** At this stage, the vessel is aligned towards the destination and navigates at cruise speed. The surge velocity dominates, and the heading change is relatively small compared to it: $u = u_0; v \approx 0; r < \varepsilon$.

To accurately represent these different navigation scenarios mathematically, we propose the following model:

$$M\dot{v} + C(v)v + Dv = \tau \quad u \approx v \approx r \quad (15)$$

$$\begin{cases} u = \sqrt{u^2 + v^2} \\ T\dot{\psi} + \dot{\psi} = K\delta \end{cases} \quad u \gg v \gg r \quad (16)$$

Equation (16), also known as the Nomoto Model (Nomoto 1967), represents a yaw subsystem derived from the manoeuvring

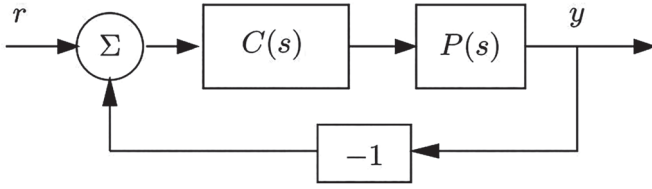


Figure 5. Feedback loop generalised block-scheme.

model (14) by selecting the yaw rate r as the output. This model illustrates the relationship between a vessel's heading and the propeller angle when the vessel sails at a constant surge speed u_0 . Given that it's a linear single-input-single-output (SISO) model, it lends itself to a frequency response analytical PID implementation (Aström and Murray 2004).

Figure 5 portrays a standard negative feedback control architecture. The transfer function of the controller is represented by C , while P denotes the vessel dynamic model. The open-loop transfer function is expressed as $L = C \times P$, and the closed-loop correlation between the reference input r and the output y is $W = \frac{L}{1+L}$. The approach in this paper aims to define C such that:

$$W = \frac{L}{1+L} = \frac{K_{dc}\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (17)$$

where the system's settling time T_s , rise time T_r and control damping ξ can be controlled using the following formula (Doyle et al. 1990):

$$T_s \approx \frac{4.6\xi}{\omega_n}; \quad T_r \approx \frac{1.8}{\omega_n}; \quad \xi = \frac{-\ln(M_p)}{\sqrt{\pi^2 + \ln^2(M_p)}} \quad (18)$$

The primary objective of the control design is to achieve:

- Zero steady-state error between r and y , implying that the ship successfully reaches the destination coordinates as specified by the Guidance strategy.
- Zero overshoot M_p when following the heading reference.
- Settling time T_s and Rise time T_r within the limits of what's feasible, considering the propeller power capabilities. In the particular case of this study, the control law is designed to satisfy: $T_s < 3$ sec; $T_r < 4$ sec, achievable by the propulsion of the Tito Neri model scale vessel.

We also explore Model Predictive Control (MPC) in this paper. Here, MPC is grounded on the iterative optimisation of a mathematical model of a vessel over a finite time horizon. The present state of the plant is sampled at time t , and a cost-minimising control strategy is computed using a numerical minimisation technique. This is assessed over a fixed future time horizon, $[t, t + T]$. The predictive nature of MPC enables the control algorithm to find an analytical solution, introducing a beneficial feature to the overall Control architecture: the ability to define a trade-off between plant output and the required control effort.

The model in Equation (14) needs to be modified, and the overall MPC problem can be formulated as follows:

$$\begin{cases} \dot{v} = \left(\frac{C(v)}{M} + \frac{D}{M} \right) v + B\tau \\ y = Cv \end{cases} \quad (19)$$

$$\begin{aligned} J = \min & \quad (R_s - Y)^T (R_s - Y) + \Delta\tau^T \bar{R} \Delta\tau \\ \text{s.t.} & \quad \tau_{\min} \leq \tau \leq \tau_{\max} \\ & \quad \dot{\tau}_{\min} \leq \dot{\tau} \leq \dot{\tau}_{\max} \end{aligned} \quad (20)$$

In the above equations, R_s is the data vector comprising the system set-point values, i.e. the state values the ship system aims to achieve. \bar{R} is a diagonal matrix represented as $\bar{R} = r_w I_{N_c \times N_c}$ ($r_w \geq 0$), where r_w serves as a tuning parameter for the desired closed-loop performance. Furthermore, Y is the vector containing the predicted output variables over the prediction horizon, and $\Delta\tau$ is the vector containing the future control trajectory as computed by the MPC solver algorithm.

$$Y = [y(k_i + 1 | k_i), y(k_i + 2 | k_i), \dots, y(k_i + N_p | k_i)]^T \quad (21)$$

$$\Delta\tau = [\Delta\tau(k_i), \Delta\tau(k_i + 1), \dots, \Delta\tau(k_i + N_c - 1)] \quad (22)$$

The optimisation problem in Equation (20), which minimises the error between the vessel's current state and the reference provided by the Guidance strategy, will be solved using commercially available software. The structure of the algorithm is outlined in the pseudo-code Algorithm 1.

While the solution is derived using commercial software, the pseudo code presented is used for illustrating the approach in the paper. It specifically highlights when certain operations should be carried out in the algorithm's sequence. Crucially, it details the specific types of inputs required by the MPC control strategy at different stages.

3. Results

3.1. Guidance strategy

The goal of the guidance strategy is to provide a reference route to an autonomous surface vessel in order to reach the destination coordinates avoiding all the obstacles in the environment. The starting point is at location (7,4) and destination at (22,175) in a Cartesian x, y plane. Without loss of generality, all the coordinates are expressed in decimetres in this section, due to the size of the model scale vessel. Three obstacles are present in the environment and the guidance strategy calculates the path between the initial and the final way-points as illustrated in Figure 6. The safety distance factor M to formulate constraints (9) is chosen as two times the ship length ($2L_{ship}$) and it is represented as a black area around the obstacles.

As shown in the plot, the algorithm provides a pathway whilst minimising the deviation from the obstacle free areas. In this test the obstacles are visible within the prediction horizon between the initial and final position, demonstrating how the MILP algorithm approaches a case of static obstacle avoidance. This is an example of a global trajectory planner, where the optimisation is done over the entire path.

The computational time of the global path planner was measured and equal to $T_{computation} \approx 12.3$ s.

3.2. Navigation system

The performance of the perception system is evaluated through feeding a validation dataset into the multi-modal network and reviewing the resultant output, as well as subjecting some focus on individual tasks to identify sub-task weaknesses. The validation dataset is comprised of 100 data points gathered within the indoor tank environment, with each data point providing a colour frame and a depth frame to be inputted to the network for inference.

Algorithm 1 MPC Controller

```

1: function SENDMPCCONTROLINPUT
2:   Tito Neri Model  $\leftarrow$  Define Tito Neri model parameters (e.g. ...)
3:    $N_p \leftarrow$  Define prediction Horizon (e.g. ...)
4:    $N_c \leftarrow$  Define Control Horizon (e.g. ...)
5:    $J \leftarrow$  Define Cost Function based on Tito Neri Model (e.g. ...)
6:   Define control input constraints (e.g. constraint type, constraint limits)
7:   Mainloop:
8:   if  $i > N_p$  then return Prediction Horizon achieved
9:   if  $i > N_c$  then
10:     Solve optimisation problem using (e.g. optimisation algorithm)
11:      $u \leftarrow$  Calculate optimal control input using (e.g. control algorithm)
12:    $x \leftarrow$  Update vessel model with the predicted states
13:    $y \leftarrow$  Update and store the plant output
14:   Send Control input to the ship
15:   goto Mainloop.

```

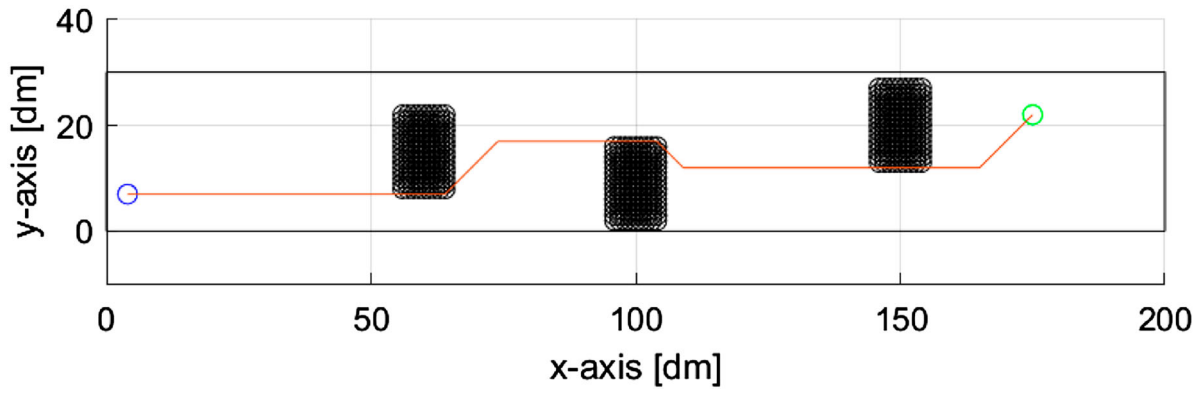


Figure 6. Global MILP trajectory planning.

The sub-figures presented in Figure 7 present an example of the visual results from the major stages throughout the multi-modal network. Each sub-figure is referenced to in the labels within Figure 4. An output from inference on the VesselNet CNN can be seen in Figure 7(a) where the inputted colour frame has been annotated with the detected object bounds and classification. The processed depth data can be visualised in Figure 7(b) as a colour-mapped image, with the pixels aligned to the colour frame and the colour-map range encoding the depth range. The fusion of the first stages creates a birds-eye-view plot such as that presented in Figure 7(c). The resultant obstacle bounds and angle from BearingNet CNN are then used to create a scaled occupancy map, as presented in Figure 7(d).

The VesselNet CNN demonstrated strong performance in detecting the vessel and distinguishing its class. Quantitative evaluation using the recall performance indicator defined in Equation (23) yields a result of 0.98, meaning that the VesselNet CNN correctly detected and classified the vessels 98% of the time within the validation dataset. As the bounding box from this network is only used to attain a ROI for ongoing tasks, the bounding box precision is not of critical importance and is therefore not quantitatively evaluated in this work however it is observed to be more than satisfactory to fulfil its role. Furthermore in a collision avoidance application, vigilance leans in favour of heightened recall at the sacrifice of precision.

The extraction and processing of depth data to create the point-cloud BEV plot also proved itself to be robust. The hardware limitation of the stereovision device does however become clear beyond approximately eight metres with the resultant point cloud being of low resolution, producing a sparse BEV plot. The multimodal

architecture inherently leads to overall system performance being dictated by the weakest link in the chain, which in this case proved to be the BearingNet. With a recall of 0.59 at an angular resolution of $\frac{\pi}{8}$, the BearingNet does not maintain the performance level of its predecessors yet does still demonstrate promise in this proof-of-concept application.

$$\text{Recall} = \frac{\text{correct detections}}{\text{all ground truths}} = \mathbf{98\%} \quad (23)$$

This method allows to determine both the position and the heading of obstacles relative to the controlled vessel frame of reference. For the guidance algorithm, once known the position of the controlled vessel, it's straightforward to calculate the global position of any obstacles nearby using basic geometry and a rotation matrix.

3.3. Control Design

Results are obtained using one of the model scale vessel available at ResearchLab Autonomous Shipping (RAS) of the Delft University of Technology, a Tugboat vessel with a scale factor 1:30 called Tito Neri (Figure 8). A Tito Neri has three propeller, two stern 360° azimuth thrusters and a tunnel bow thruster. It is a fully actuated ship system with respect to the Cartesian plane since, based on its thrust allocation, it is possible to provide thrust in all of the 3DoF. The ship is equipped with the hardware and sensors summarised in Table 1.

The first step is to identify the Nomoto Model parameters in (16). This is achieved with experimental measurement campaigns using the High Precision Positioning Feedback Camera system in the Towing tank lab facility of the Maritime and Transport Technology

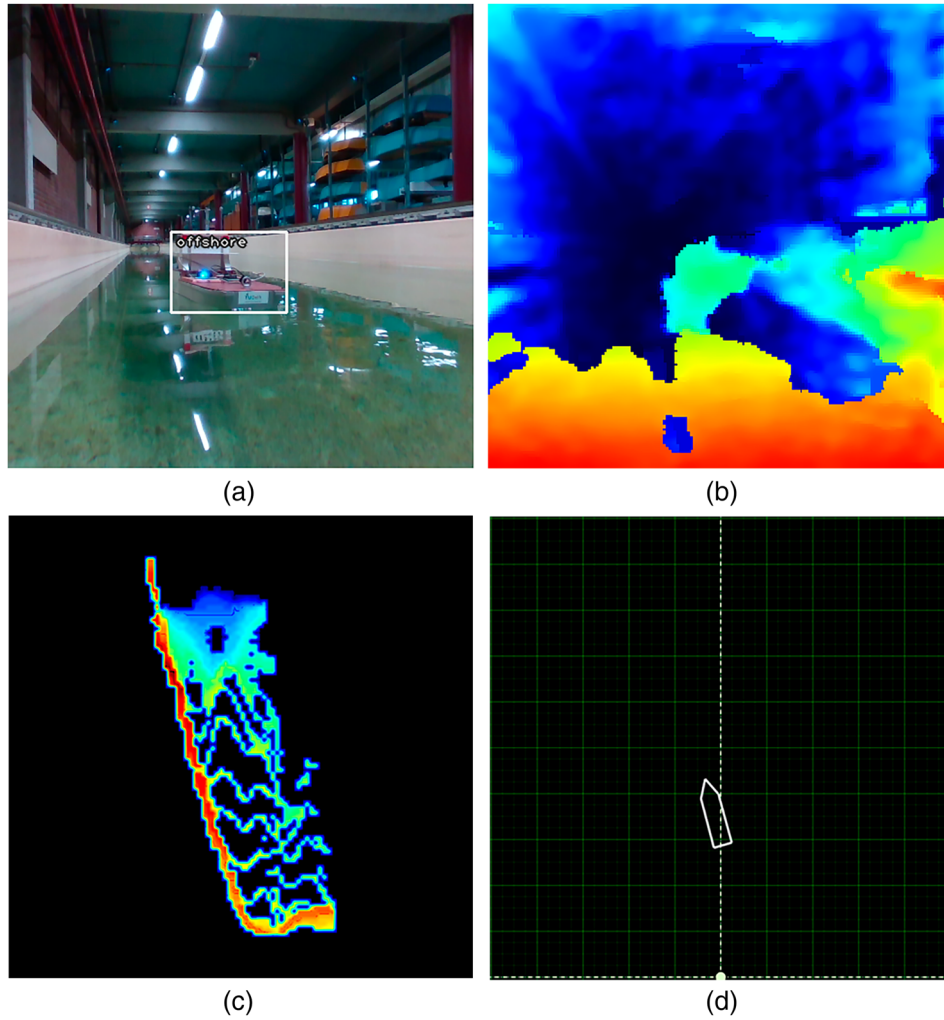


Figure 7. Perception system results. (a) Output from VesselNet neural network, (b) Colour-map visualisation of processed depth data, (c) Birds-eye-view plot of fused data and (d) Planar occupancy map – Grid size: 1m, coordinate origin marked by white dot..

Table 1. Tiro Neri's hardware ad sensor.

Hardware and sensor list	
Positioning measurement sensor	IR reflective passive markers
Inertial Measurement Unit	Bosch BNO055
Motor Encoder	Absolute rotary IR encoder
Propeller Control Board	Arduino – ATmega328P
Position Control board	Raspberry PI, Nvidia Jetson TX2, Intel NUC

Department at Delft University of Technology. The tests were performed reaching steady surge velocity on the Tiro Neri model scale ship and injecting at time t_0 a deviation of 45° angle on the stern propellers. The experiments were repeated 10 times and the separate results were compared and averaged in order to reduce the amount of uncertainties during the test phase .

Figure 9 show the evolution over time of the heading rotational velocity r measured by the camera system and the simulated output of the mathematical Nomoto model response were extracted using nonlinear least-squares fit method.

The parameters where then calculated from the plots: $K = 3.55$, $T = 1.67$ leading to the formulation of the Nomoto model

$$P = \frac{\psi}{\delta}(s) = \frac{3.55}{s(1 + 1.67s)} \quad (24)$$

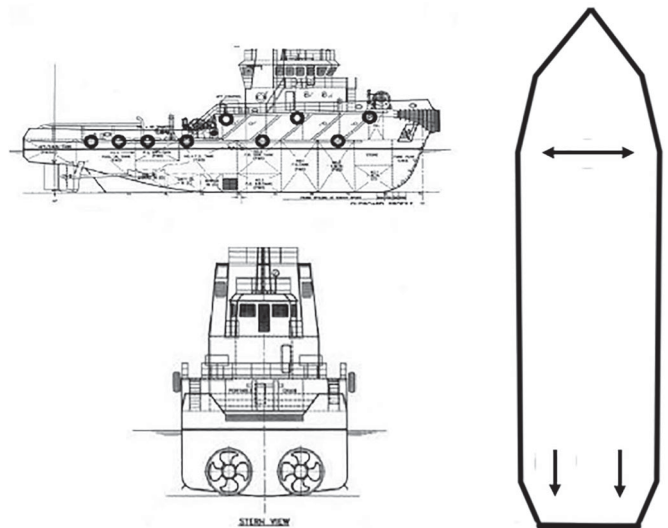


Figure 8. Tiro Neri's thrust allocation.

Based on the transfer function obtained and recalling (17), the control law C is calculated as

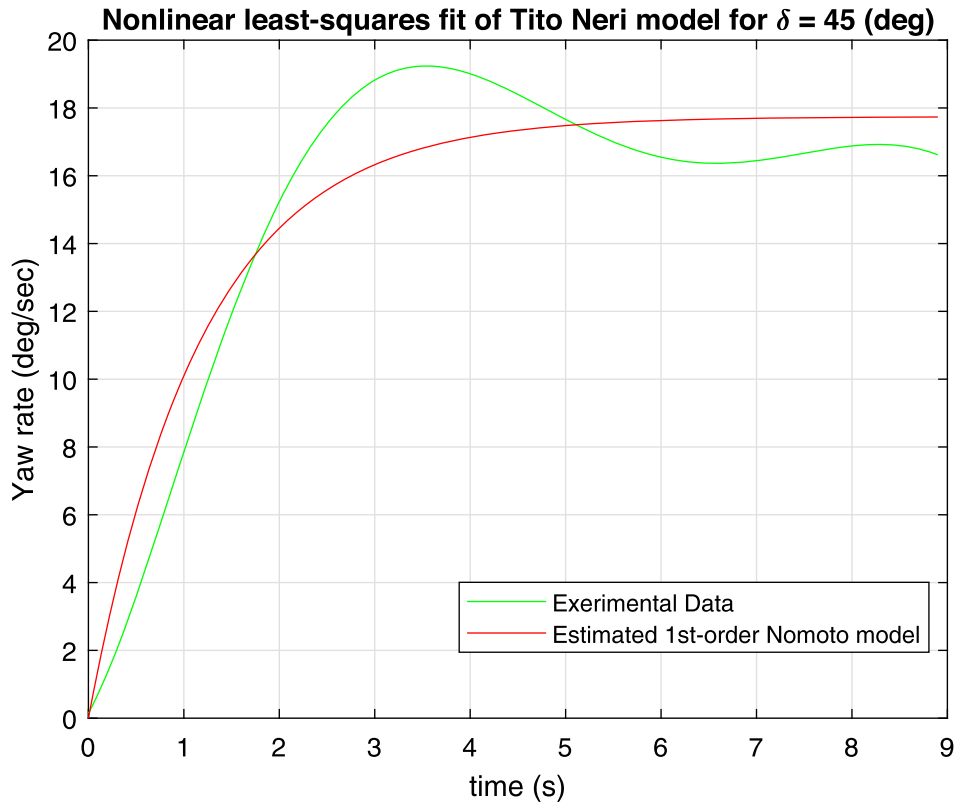


Figure 9. Estimated nomoto model using nonlinear least-square fit.

$$C = \frac{K_c}{s + a} \quad (25)$$

As depicted in Figure 10, the controlled Tito Neri vessel follows the reference path stipulated by the Guidance algorithm. The dynamics remain stable as the ship avoids environmental obstacles, despite noticeable path deviations, particularly around curves. Such transient deviations can be attributed to the control law's focus on heading dynamics, while not considering the physical correlations between sway and yaw motion. The model predictive control (MPC) design formulation, a comprehensive model-based control strategy, accounts for multi-input-multi-output dynamics by incorporating a full 3DoF ship model. Recalling the expression (14), in the following the expanded matrices:

$$M_{RB} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & mx_g \\ 0 & mx_g & I_z \end{bmatrix}, \quad M_A = \begin{bmatrix} -X_{\dot{u}} & 0 & 0 \\ 0 & -Y_{\dot{v}} & -Y_{\dot{r}} \\ 0 & -Y_{\dot{r}} & -N_{\dot{r}} \end{bmatrix}, \quad (26)$$

$$C_{RB} = \begin{bmatrix} 0 & 0 & -m(x_g r + v) \\ 0 & 0 & mu \\ m(x_g r + v) & -mu & 0 \end{bmatrix}, \quad (27)$$

$$C_{RB} = \begin{bmatrix} 0 & 0 & Y_{\dot{v}} + (Y_{\dot{r}} + N_{\dot{v}})r \\ 0 & 0 & -X_{\dot{u}} \\ -Y_{\dot{v}}v - (Y_{\dot{r}} + N_{\dot{v}})r & -X_{\dot{u}}u & 0 \end{bmatrix}, \quad (28)$$

$$D = \begin{bmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & -Y_r \\ 0 & -N_v & -N_r \end{bmatrix}, \quad (29)$$

The values of the ship model parameter are summarised in Table 2.

As illustrated in Figure 11, this consideration of the correlated dynamics between lateral and angular motion enables the control

Table 2. Tito Neri 3DoF model parameters.

m	16.9
x_g	0
I_z	0.51
$X_{\dot{u}}$	-1.2
$Y_{\dot{v}}$	-49.2
$Y_{\dot{r}}$	0
$N_{\dot{v}}$	0
$N_{\dot{r}}$	-1.8
X_u	1.98
Y_v	7.8
Y_r	0
N_v	0
N_r	3.87

algorithm to guide the ship along the reference route with zero path deviation. However, it is worth noting that the control effort demanded by the MPC controller is considerable, especially for the actuators. Figure 12 elucidates how the control law requires a force that continually changes at a frequency of approximately 0.2 Hz, resulting in a substantial load on the actuators. This is primarily to compensate for the sway drift during navigation, which, however, might induce aggressive actuator dynamics, potentially leading to mechanical failure. Key performance indicators (KPIs) of the control strategies analysed in this study are concisely summarised in Tables 3 and 4. Figures 13–15 document the results of the PID-based control approach. Figure 14 specifically displays the ship's heading control performance, demonstrating no overshoot and a rise time of 2.1 seconds. This performance, along with a steady-state settling time of 3 seconds, reveals the adaptability of the system across diverse avoidance maneuvers. Figure 15 presents the control inputs directed towards the ship's azimuth propeller system, demonstrating how the frequency response-based control law can generate feasible control

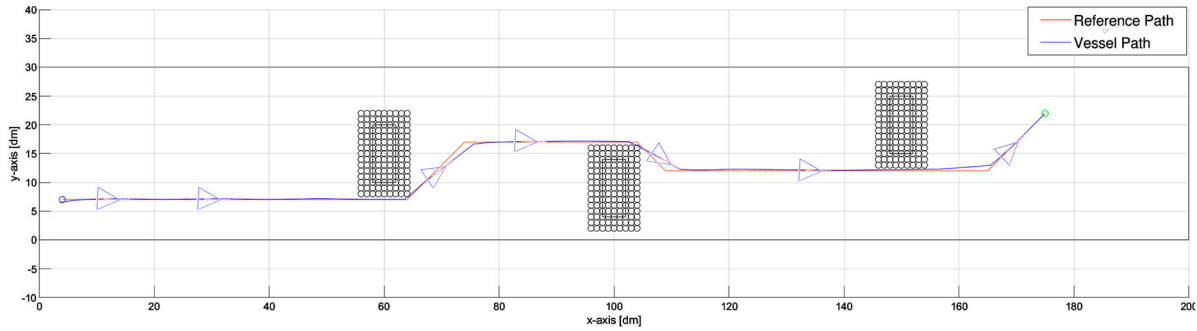


Figure 10. xy Cartesian plot using PID controller.

Table 3. KPIs control strategies.

KPI	MPC	PID
Steady-State position error	0	0
Heading Overshoot	0%	0%
Max Path Deviation	0.02 m	0.23 m
Obstacles avoided	✓	✓

Table 4. KPIs GNC scheme adopted in the experimental setup.

KPIs	
Steady-State ship's heading error	0
Controlled Heading Rise time	2.1 sec
Controlled Heading Settling time	3 sec
Controlled propulsion – Rise time	1.6 sec
Controlled propulsion – Settling time	3.4 sec
Controlled propulsion – error	$\pm 5\%$

inputs for the motors driving the azimuth rotation mechanics. Figures 16 and 17 elaborate on the finer aspects of the low-level control on the ship's propulsion system. The propellers maintain a constant speed on both the port and starboard sides throughout the navigation. With a rise time of 1.6 seconds and a settling time of 3.4 seconds, the performance of the propeller speed control reemphasizes the compatibility of the control law with the ship's hardware. In conclusion, the PID-based approach coupled with the frequency response-based control law produces an efficient navigation control system. Its robustness, exhibited in both heading control and actuator-level operations, suggests potential suitability for real-world maritime navigation applications, thereby aiding the enhancement of a vessel's autonomous capabilities.

4. Conclusions and further research

A complete GNC framework has been presented in this paper, enhancing the importance of how these three modules are dependent upon one another to achieve autonomous navigation for a surface vessel.

In this work the sub-systems have only been evaluated independently however the authors intend to evaluate the complete GNC framework under experimental collision avoidance scenarios in future work. The MILP guidance algorithm as presented serves as a proof on concept for a global trajectory planning algorithm. It generates an alternative, minimally deviated, evasive path for the vessel whilst respecting kinematic constraints. In the context of real ship dynamics, especially for large vessels, 12 seconds planning time may be adequate given the slower responses time and longer distances required for maneuvers. However, it is important to emphasise that the primary focus was on establishing a proof of concept, and the

presented complete GNC scheme was chosen to balance computational efficiency with realistic and feasible model-scale operational scenarios. Faster hardware can theoretically minimise the computational load even further, for application where fast moving objects are in the environment as well. Comparatively, existing maritime navigation systems, which often rely on a combination of radar and manual inputs, operate on a similar timescale for trajectory planning.

Additionally, the guidance approach focuses on immediate spatial positioning rather than a predictive, kinematic based, modelling using the obstacle vessels headings. Incorporating future obstacle position prediction could, however, represent an interesting direction for future work, where the dynamics of moving obstacles could be integrated into the guidance system for more complex navigational scenario taking into account their heading and the possibility to estimate the velocity of the surrounding objects as well.

The perception approach investigated in this work demonstrates the promise of a multi-modal network architecture in achieving situational awareness for obstacle avoidance activities. In the research the camera sensor was primarily chosen for its accessibility and compatibility with the model scale used, to provide a proof of concept. The D435i, with its active IR projections, is not merely a stereo camera and is less suited for large-scale sea applications due to its range and sensitivity to environmental conditions. In future work alternative for true development can be considered, for example a mix of specialised marine-grade LiDAR system with more sophisticated camera technology. Nonetheless, the underlying principles of the Computer Vision algorithm demonstrated here would largely remain applicable. The Faster R-CNN based VesselNet proves itself to be a suitable candidate for detecting vessels and distinguishing their classification from colour image frames. Depth-data from a stereo-vision sensor can provide sufficient point-cloud resolution to detect both vessel orientation and distance, albeit within the range capacity of the device's hardware. Although the overall performance of the BearingNet limited the performance of the network, this sub-optimal performance could be improved in future work through additional training data and fine tuning of the network model. Even with a limited overall system performance, the slow-moving nature of vessels and the capacity for multiple frames to be processed each second could still permit application to autonomous vessels. Additionally, it is also noteworthy to mention that in our approach, setting missing depth values to the maximum of their surroundings represents an optimistic scenario. For applications in real-world environments, a more robust method for filling these gaps could be explored. This could include statistical techniques or machine learning-based interpolation strategies, which potentially offer a more precise estimation of missing data points in a sparse depth field. Moreover, considering the notably high recall rate of 98% achieved by VesselNet,

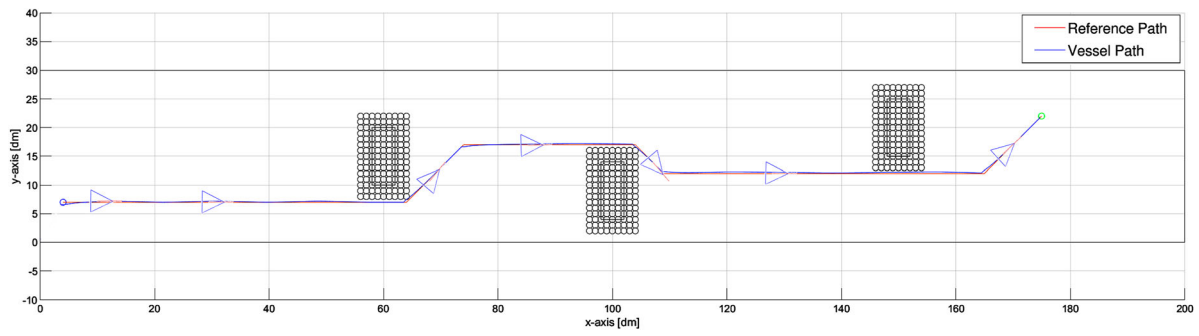


Figure 11. xy Cartesian plot using MPC controller.

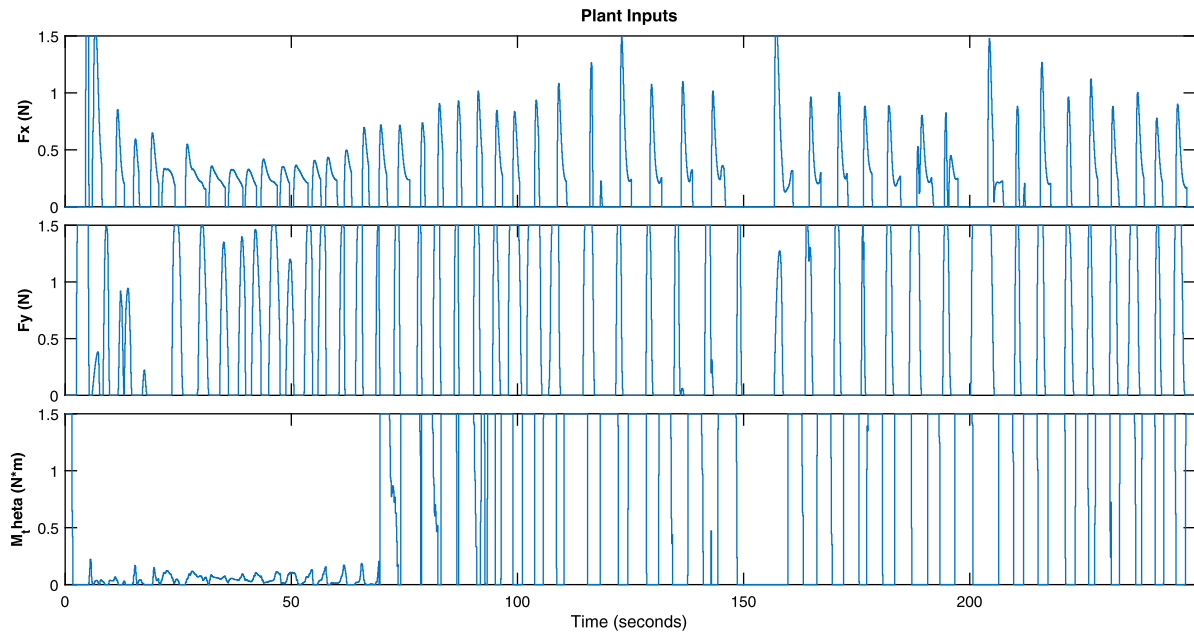


Figure 12. MPC – control forces requested.

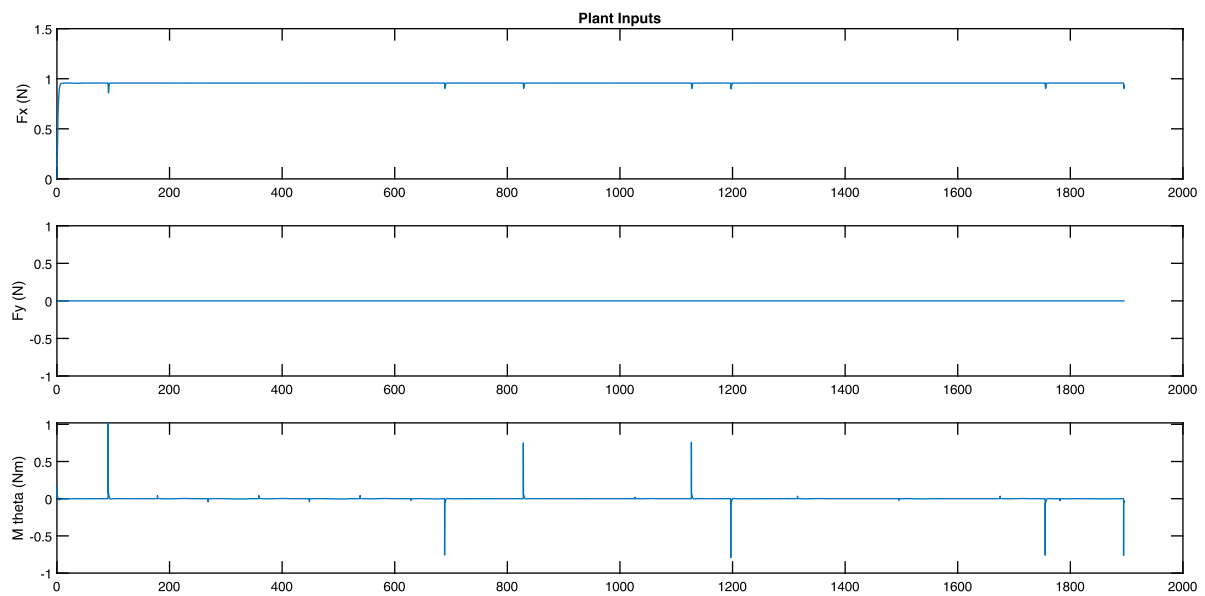


Figure 13. PID – control forces requested.

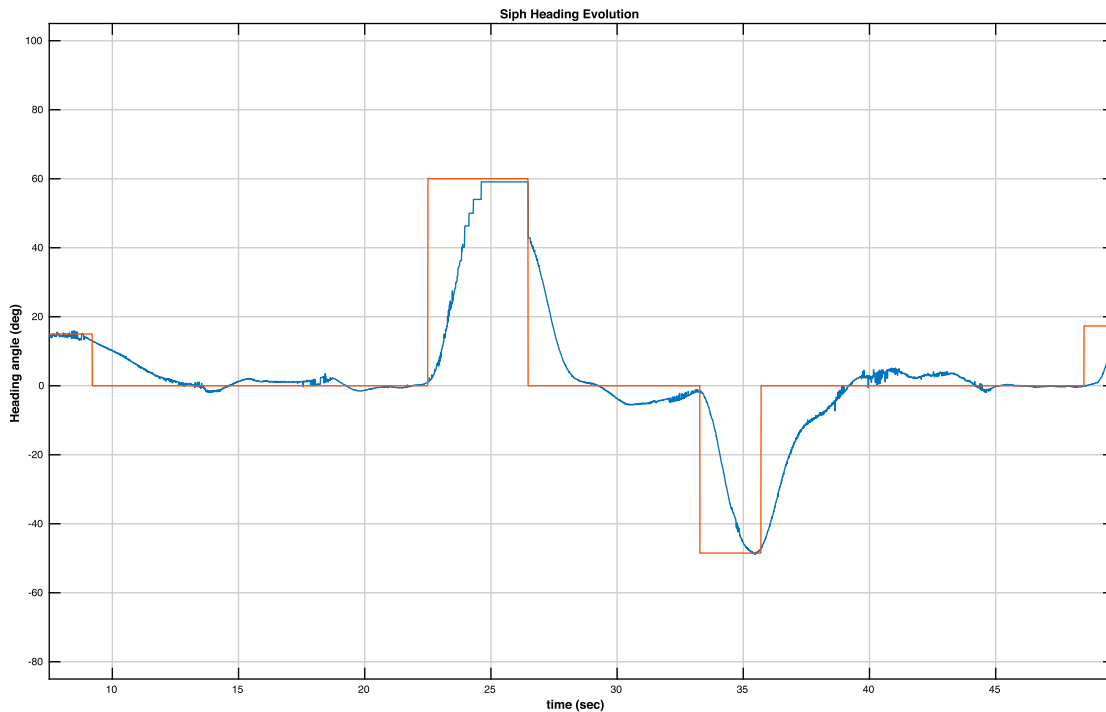


Figure 14. Ship's heading evolution.

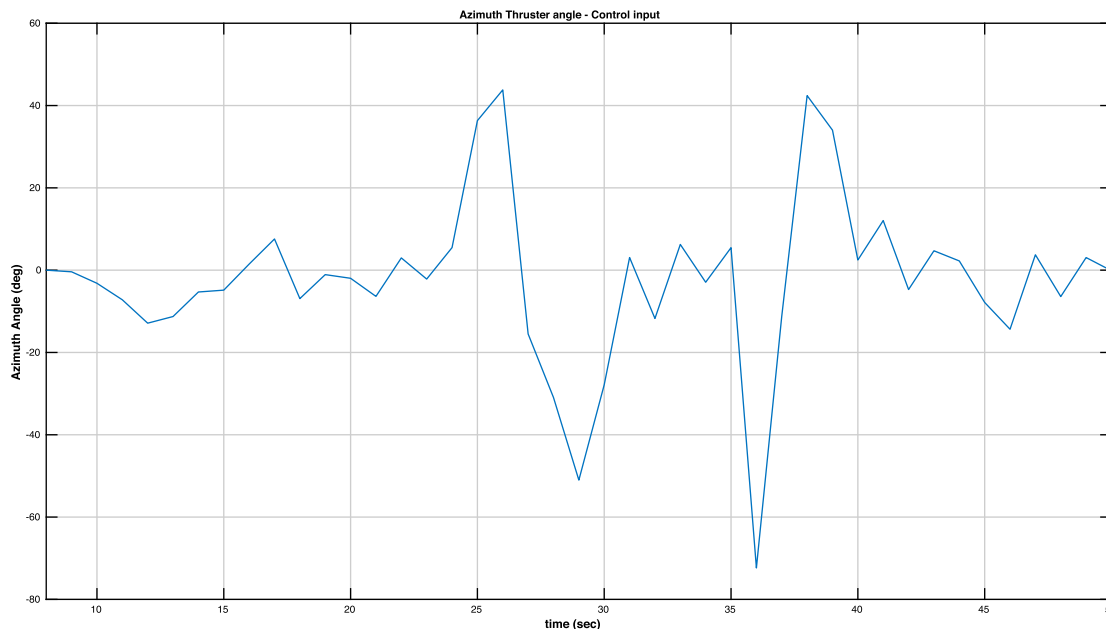


Figure 15. Azimuth propeller control inputs.

primarily attributed to the training on model-scale ships similar to those used in our test scenarios, it becomes important to address the potential limitations of our current dataset. The application of VesselNet in real-world, densely trafficked maritime environments may reveal performance degradation if the training dataset lacks diversity and fails to integrate a comprehensive array of ship types and conditions comparable with those encountered in actual navigation settings. This underlines the necessity of expanding the dataset used in this work to include a broader spectrum of real-world maritime images, enhancing the neural network capability to accurately detect and classify vessels under a wide range of scenarios. Furthermore, the impact of potentially noisier detected object positions,

as well as the occurrence of false negatives and positives, demands in-depth analysis because of their substantial influence on the path planning executed by the Guidance system. While the inherent control stability of ship system in a path following context may mitigate the risks associated with path inconsistencies, the increased control effort demanded from actuators in response to detection inaccuracies could potentially compromise vessel integrity. Therefore, the pursuit of further research to enhance VesselNet by training with real ship navigation data becomes essential. This effort is crucial for guaranteeing the robustness and dependability of our proposed guidance and control strategy among the diverse and challenging dynamics of marine environments.

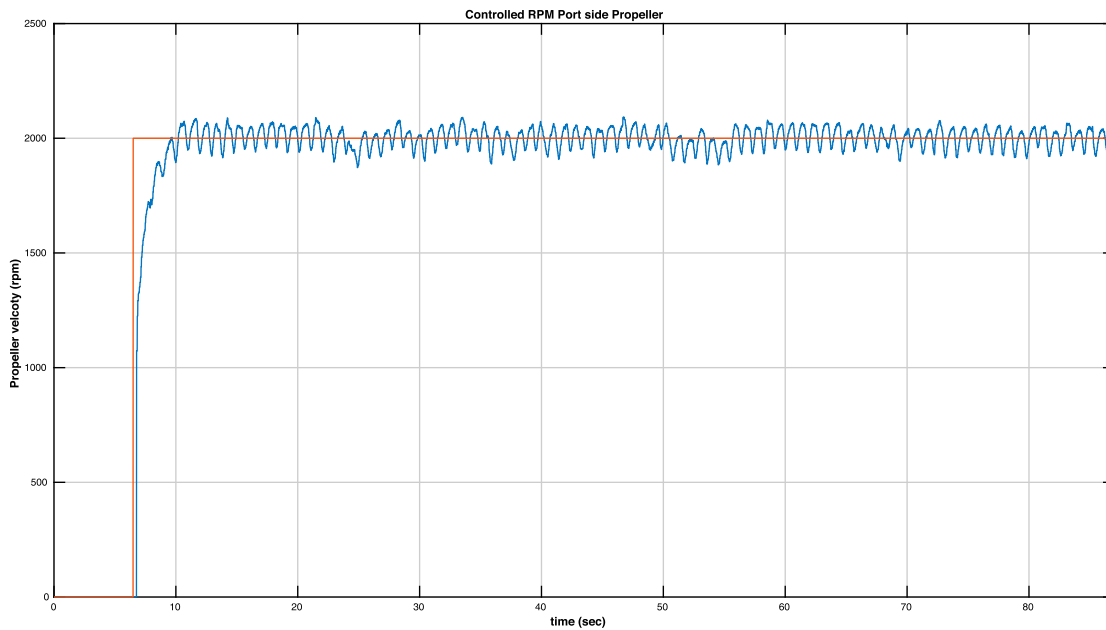


Figure 16. Controlled rpms, portside propeller.

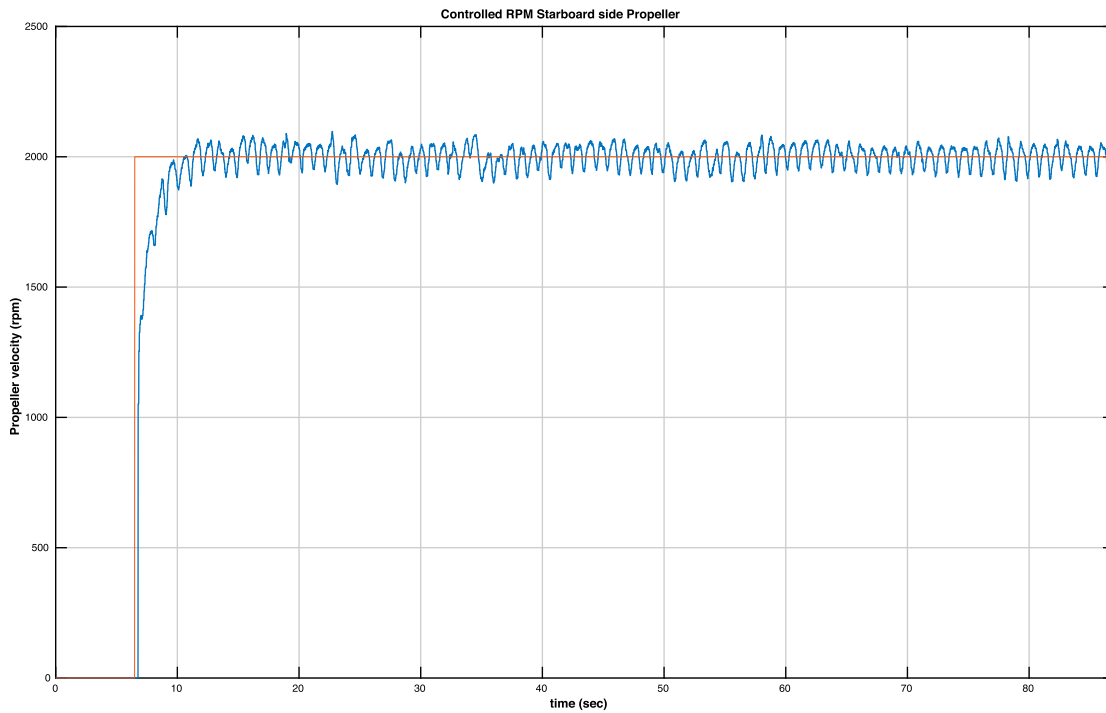


Figure 17. Controlled rpms, starboard propeller.

In conclusion, this paper has discussed two model-based control design strategies – a second-order frequency response correlation-based controller and a Model Predictive Control (MPC). Each of these controllers demonstrated the potential to contribute to autonomous navigation when integrated with the previously described Guidance and Navigation systems. Although the MPC offered a higher precision in path following, the second-order frequency response correlation-based controller emerged as the preferable choice for practical implementation. This decision is justified by its robustness, superior computational efficiency, and lesser demand on the actuator side, hence ensuring the preservation of the ship's mechanical integrity. In this study, it was found that the impulses

sent to the actuator by the Model Predictive Control algorithm were primarily due to uncertainties in the maneuvering model parameters Table 2. Essentially, if the ship anticipates that a certain force will move it to a predicted position, but subsequently finds itself in a different position in the next time frame, the MPC algorithm attempts to correct this discrepancy by counteracting the previous effort.

However, this conclusion does not undermine the potential benefits of the MPC and methods to mitigate the illustrated control spikes should be further investigated in future research also to explore ways of reducing the computational demands of more advanced and theoretically superior control strategies like the Non-linear Model

Predictive Control and back-stepping. Such advancements are crucial, especially considering that these control systems must work in tandem with computationally intensive guidance and navigation methods.

Moreover, as our study was primarily based on the Tito Neri model vessel, future research can also investigate the applicability and performance of these control strategies on different vessel types. Furthermore, the exploration of control design methodologies that take into account other dynamics, beyond the heading dynamics considered in this study, could offer a more comprehensive approach to autonomous navigation.

Lastly, further consideration of the potential effects of environmental conditions on the performance of these control systems, and how they can be effectively accounted for in the control design process, is also warranted. While the core of this research primarily focussed on the implementation aspects of a full GNC system, further simulations can play an important role in refining the proposed methods and validating conclusions. Going forward, the plan is to expand simulation scenarios to include diverse maritime environments and dynamic obstacle behaviours, integrating real-world data to bridge the gap between simulation and practical application. This will not only test the robustness and adaptability of our system under varied conditions but also help us understand its scalability and actual efficiency in crowd maritime environments. This will also ensure the adaptability and reliability of autonomous maritime navigation systems in real-world maritime conditions.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

The research presented in this paper was supported by the ResearchLab Autonomous Shipping (RAS) and funded by the European Union and the European Regional Development Fund, as part of the Interreg North Sea Region project AVATAR.

Notes on contributors

Ir. V. Garofano is Research/Education support Engineer and Coordinator of the technical activities in the ResearchLab Autonomous Shipping (RAS) at the Department of Maritime & Transport Technology, Delft University of Technology. His research and engineering work include implementation of control algorithms with focus on autonomous inland vessels navigation, software-hardware integration and design of robust control strategies.

Ir. M Hepworth is Research Engineer at the ResearchLab Autonomous Shipping at Delft University of Technology. He is specialised in the application of computer vision and artificial intelligence to situational awareness tasks in the field of maritime autonomy.

R. Shahin is an MSc student of the Aerospace Faculty at Delft University of Technology and formerly a Research Assistant at the ResearchLab Autonomous Shipping (RAS) working on state of the art Guidance Algorithm.

Dr. ir. Y. Pang is assistant professor in the department of Maritime and Transport Technology, Delft University of Technology. His research mainly focuses on the intelligent control for transport technology and logistics processes including the fields of autonomous shipping applications.

Prof. Dr. R. R. Negenborn is full professor in Multi-Machine Operations & Logistics at Delft University of Technology. He is head of the Section Transport Engineering & Logistics and leads the Researchlab Autonomous Shipping (RAS). His research interests include automatic control and coordination of transport technology, with a focus on smart (autonomous) shipping and smart logistics applications. He has over 300 peer reviewed publications and leads NWO, EU and industry funded research, such as the Horizon 2020 program NOVIMOVE.

References

Aström KJ, Murray RM. 2004. Feedback systems: an introduction for scientists and engineers. Princeton University Press.

- Beltrán J, Guindel C, Moreno FM, Cruzado D, Garcia F. 2018, Nov. Birdnet: a 3D object detection framework from lidar information. 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, Hawaii, USA, p. 3517–3523. IEEE. <https://doi.org/10.48550/arXiv.1805.01195>
- Choy CB, Stark M, Corbett-Davies S, Savarese S. 2015. Enriching object detection with 2d–3d registration and continuous viewpoint estimation. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, p. 2512–2520. IEEE.
- Coleman D. 2024. Obstacle avoidance and path planning using mixed integer programming. Department of Computer Science, University of Colorado.
- Corić M, Mandzuka S, Gudelj A, Lušić Z. 2021 May. Quantitative ship collision frequency estimation models: a review. *J Marine Sci Eng.* 9:533. doi: 10.3390/jmse9050533
- Deraj R, Kumar RS, Alam MS, Somayajula A. 2023. Deep reinforcement learning based controller for ship navigation. *Ocean Eng.* 273:113937. doi: 10.1016/j.oceaneng.2023.113937
- Dey C, Sen SK. 2020. Industrial automation technologies. Boca Raton: CRC Press. doi:10.1201/9780429299346.
- Doyle J, Francis B, Tannenbaum A. 1990. Feedback control theory. London: Macmillan Publishing Group.
- Fossen TI. 2011. Handbook of Marine Craft Hydrodynamics and Motion Control. John Wiley & Sons, Ltd.
- Garofano V, Hepworth M, Shahin R. 2022. Obstacle avoidance and trajectory optimization for an autonomous vessel utilizing milp path planning, computer vision based perception and feedback control. Proceedings of the International Ship Control Systems Symposium. Vol. 16. Conference Proceedings of iSCSS, International Ship Control Systems Symposium, Delft, 8–10 November 2022.
- Haseltalab A, Garofano V, van Pampus M, Negenborn R. 2020. Model predictive trajectory tracking control and thrust allocation for autonomous vessels. *IFAC-PapersOnLine.* 53(2):14532–14538. doi: 10.1016/j.ifacol.2020.12.1457 21st IFAC World Congress 2020; Conference date: 12-07-2020 Through 17-07-2020.
- Hepworth M, Garofano V, Reppa V, Negenborn RR. 2021. Collision avoidance for autonomous inland vessels using stereovision. <https://repository.tudelft.nl/islandora/object/uuid:5e00f536-fa99-4116-8f87-cfb01a525fcb?collection=education>.
- Huang J, Rathod V, Sun C, Zhu M, Korattikara A, Fathi A, Fischer I, Wojna Z, Song Y, Guadarrama S, et al. 2016, Nov. Speed/accuracy trade-offs for modern convolutional object detectors. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, p. 7310–7311. IEEE.
- Kato S, Takeuchi E, Ishiguro Y, Ninomiya Y, Takeda K, Hamada T. 2015. An open approach to autonomous vehicles. *IEEE Micro.* 35(6):60–68. doi: 10.1109/MM.2015.133
- Khalid Z, Mohamed EA, Abdenbi M. 2013. Stereo vision-based road obstacles detection. 2013 8th International Conference on Intelligent Systems: Theories and Applications, SITA 2013. p. 1–6. IEEE and British Council, Rabat, Marocco.
- Kim K, Hong S, Choi B, Kim E. 2018 Jun. Probabilistic ship detection and classification using deep learning. *Appl Sci.* 8(6):936. doi: 10.3390/app8060936
- Lin T-Y, Dollár P, Girshick R, He K, Hariharan B. 2017. Feature pyramid networks for object detection. *IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA.* IEEE.
- Lin T-Y, Dollár P, Girshick R, He K, Hariharan B. 2020. Focal loss for dense object detection. *IEEE Trans Pattern Anal Mach Intell.* 42(2):318–327. doi: 10.1109/TPAMI.34
- Nomoto K. 1967. A simplified analysis on ship motion under manoeuvre and proposed steering. *Osaka University, Japan.* <http://resolver.tudelft.nl/uuid:5c3dd568-013d-4d85-8dfc-2579d84cede8>
- Parkes A, Savasta T, Sobey A, Hudson D. 2022. Power prediction for a vessel without recorded data using data fusion from a fleet of vessels. *Expert Syst Appl.* 187:115971. doi: 10.1016/j.eswa.2021.115971
- Parkes A, Sobey A, Hudson D. 2018. Physics-based shaft power prediction for large merchant ships using neural networks. *Ocean Eng.* 166:92–104. doi: 10.1016/j.oceaneng.2018.07.060
- Richards A, How JP. 2002. Aircraft trajectory planning with collision avoidance using mixed integer linear programming. Proceedings of the 2002 American Control Conference, Anchorage, AK, USA. IEEE
- Sánchez-Beaskoetxea J, Basterretxea-Iribar I, Sotés I, Machado M. 2021 Jan. Human error in marine accidents: is the crew normally to blame? *Maritime Transp Res.* 2:100016. doi: 10.1016/j.martra.2021.100016
- Schouwenaars T, De Moor B, Feron E, How J. 2001. Mixed integer programming for multi-vehicle path planning. 2001 European Conference (ECC), Porto. IEEE.
- Sciavicco L, Siciliano B. 1998. Modeling and control of robot manipulators, Lorenzo Sciavicco and Bruno Siciliano. *Journal of Intelligent and Robotic Systems.* 21(1):99–100. doi: 10.1023/A:1007979428654

- Sivaraj S, Dubey A, Rajendran S. 2023. On the performance of different deep reinforcement learning based controllers for the path-following of a ship. *Ocean Eng.* 286:115607. doi: [10.1016/j.oceaneng.2023.115607](https://doi.org/10.1016/j.oceaneng.2023.115607)
- Vielma JP. 2015. Mixed integer linear programming formulation techniques. *SIAM Rev.* 57(1):3–57. doi: [10.1137/130915303](https://doi.org/10.1137/130915303)
- Wang W, Gheneti B, Mateos LA, Duarte F, Ratti C, Rus D. 2019. Roboat: an autonomous surface vehicle for urban waterways. 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, p. 6340–6347. IEEE.
- Ye J, Li C, Wen W, Zhou R, Reppa V. 2023. Deep learning in maritime autonomous surface ships: current development and challenges. *J Marine Sci Appl.* 22:584–601. doi: [10.1007/s11804-023-00367-1](https://doi.org/10.1007/s11804-023-00367-1)