Obstacle Avoidance and Trajectory Optimization for an Autonomous Vessel Utilizing MILP Path Planning, Computer Vision based Perception and Feedback Control

Garofano, V.; Hepworth, M.; Shahin, R.

DOI
10.24868/10713

Publication date
2022

Document Version
Final published version

Published in
Proceedings of the International Ship Control Systems Symposium

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

Ir. V Garofano[a*†], *Ir. M Hepworth*[a*], *R Shahin,*[a*]

[a]*Delft University of Technology, The Netherlands;*

[†]Corresponding author. Email: v.garofano@tudelft.nl

[*]The three authors contribute equally to this paper

Synopsis

The framework of an autonomous vessel is typically composed of three distinct and independent blocks known as the Guidance, Navigation and Control (GNC) system. This paper presents a combination of advanced complementary techniques in the different GNC subsystems to improve upon the current common practices/state of the art in obstacle avoidance. The novel Guidance system is based on Mixed Integer Linear Programming (MILP). This optimisation technique allows quick, robust path planning with the possibility for a variety of constraints. The feasibility of this method will be investigated with the goal of providing optimal path planning in the presence of static and dynamic obstacles during autonomous sailing operations. Within the Navigation System, a multi-modal neural network architecture is proposed for the perception branch to provide high-level situational awareness for collision avoidance purposes. The computer vision approach allows for the vessel type, position and orientation all to be extracted for encounters with both dynamic and static obstacles using only imaging sensors. Two Control methods are studied in the paper, an error-based PID control strategy as well as an MPC control scheme. These two techniques will be compared to evaluate the performance and reviewing the suitability for use within the specific GNC scheme and the generic application environment. This paper details the specific implementation of each system within the overall framework, presents simulation results of the path planner and control strategy, with a performance evaluation of the navigation system using an experimental dataset. The results obtained are analysed through qualitative discussion as well as quantitative performance indicators and key conclusions are consequently drawn.

Keywords: GNC scheme; Mixed Linear-Integer Programming; Computer Vision; Artificial Intelligence; Feedback Control; Model-based Design

1 Introduction

In the last twenty years the market share of maritime goods transportation has doubled, with 90% of commodities now being transported around the world using ships. This trend is expected to continue with growth being forecast from $537.07 billion in 2022 to $650.21 billion in 2026 at an annual growth rate of about 5% (Sánchez-Beaskoetxea et al., 2021). The rapid growth to now has been supported by improved network, infrastructure and vessel efficiency and as the growth of this modality continues, so does its reliance on further technological advancements. Statistics reports from 1999-2019 show that human error was responsible for 50% of pollution, 65% of personal injuries, 80% of property damages and 90% of collisions (Corić et al., 2021). Overconfidence, recklessness in responding to commercial pressures, fatigue and lack of adequate communication are found as the leading causes. Autonomous navigation could theoretically improve overall safety during navigation and goods transportation over water can be performed more efficiently and sustainably.

In order to achieve autonomous navigation, an autonomous vessel is typically composed of three distinct modules known as the Guidance, Navigation and Control (GNC) system as shown in Fig. 1. This paper investigates the potential of using a combination of advanced techniques in the different GNC subsystems to improve upon the current standard practices/state of the art in autonomous navigation while performing obstacle vision recognition and avoidance.

A Guidance System of an autonomous vessel is required to conduct both global and local path planning tasks, with some strategies allowing for both tasks to be fulfilled by a collective method. The application of a Mixed Integer Linear Programming (MILP) algorithm for example provides a path-planning optimizer, allowing the global path to be updated to avoid local obstacles. This framework requires the formulation of a cost function with a set of constraints aiming to minimize path deviation in the presence of static and dynamic obstacles. This has been

Authors' Biographies

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

Ir. Matt 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.

Ramzi Shahin is an MSc student at Delft University of Technology and formerly a Research Assistant at the ResearchLab Autonomous Shipping (RAS).

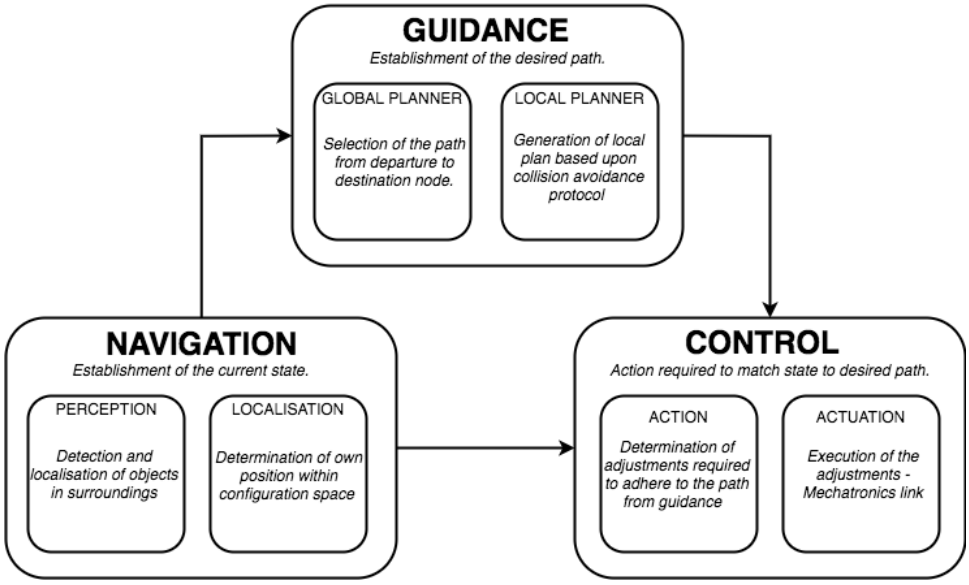

Figure 1: Guidance Navigation and Control System Framework (Hepworth et al., 2021)

shown to be an effective method for global trajectory planning in a number of different application fields (Coleman; Tom Schouwenaars 2001) and will be further exploited in this paper to investigate its use in the maritime domain.

Perception within a Navigation System plays a critical role in autonomous sailing, with obstacles needing to be detected and their position and orientation determined. The localisation branch is not considered within this research focus. Point cloud clustering is one prevalent perception approach utilised by autonomous vehicles for object detection (Kato et al., 2015), including within the maritime field (Wang et al., 2020; Khalid et al., 2013). However monocular imaging techniques offer advantages over point-cloud clustering, with Convolutional Neural Networks (CNNs) gaining popularity for providing classification of object types as well as the detected object bounds (Shao et al., 2018; Prasad et al., 2017). Although 3D bounds can be extracted with these techniques, orientation is not directly inferred. Viewpoint estimators have come someway to resolving this challenge for CNNs. Whilst these have shown significant promise in the automotive field (Choy et al., 2015), ideally CAD models for all obstacles that wish to be detected should be available for training purposes. Viewpoint estimation without CAD models using birds-eye-view (BEV) representation from a LiDAR point cloud presents an alternative (Beltrán et al., 2018), however in a maritime setting surface-water generates noise, limiting its appeal.

This work instead proposes a multi-modal neural network architecture to provide high-level situational awareness for collision avoidance purposes. Inspiration is taken from existing approaches, whereby a CNN is utilised for object detection from colour images and depth data is processed to provide 3D data. The novelty in the proposed approach comes through the fusion of this data to provide a cropped birds-eye view input to a secondary neural network which is trained specifically to extract the orientation of an obstacle, in this application a vessel.

The third pillar in the GNC framework is the design of a control strategy. Its main function in an autonomous navigation context 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. Due to its relatively simple implementation and robustness against external disturbance, 95% of closed-loop control algorithms used in industrial automation are based on a Proportional, Integral and Derivative (PID) controller (Chanchal Dey, 2020). However, typically its application is purely theoretically based and a final design is mainly achieved by trial and error tuning or empirical heuristics. In this paper, an analytical controller with analog properties of proportional, derivative and integral action will be investigated based on the frequency response relation of a closed-loop control system. The other control strategy investigated in this paper is the Model Predictive Control, an optimization-based control technique, used to regulate a process while fulfilling a set of constraints (Haseltalab et al., 2020).

This paper is organised as follows: in Section (2) the proposed GNC scheme is introduced, with each subsystem of the scheme being encompassed within its own sub-section. In Section (3), the results gained from the individual sub-systems are presented, with performance evaluations from a combination of simulated and experimental testing. Finally, in Section(4) the paper and its findings are summarised in the conclusions and recommendations for future work.

Figure 2: 'Tito Neri' Model Scale Tugboat

Figure 3: 'Grey Seabax' Model Scale Offshore Vessel

2 Methodology

2.1 Guidance Strategy

The general formulation of a Mixed Integer Linear Programming problem can be formulated as (Vielma, 2015):

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

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 $A\mathbf{x} \leq \mathbf{b}$ 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 Eq, 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 Eq. 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 Eq. 4.

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

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

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

This l_2 norm may be transformed to a l_1 norm suitable for linear optimisation by introducing the augmented error system in Eq. 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 Eq. 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.

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

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

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 Eq. 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.

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

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 $i_k \in [0,1]$, M corresponding to a weight 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 margin. This obstacle boundary can be extended to act as a safety factor as required. The constraint in Eq. 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 Eq. 9.

$$i_1 + i_2 + i_3 + i_4 \leq 3 \tag{8}$$

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

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. $\mathbf{x}(t=0) = \mathbf{x}_0 = (\mathbf{Y}_0\, \mathbf{X}_0)^T$ This results in the Linear Matrix Equality in Eq. 10 with η_Y and η_X corresponding to the current vessel position.

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

These separate matrices can now be concatenated into one single LMI and formulated as a single MILP problem as in Eq. 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 Eq. 12, with $k\Delta t$ representing the time horizon. Using an MILP solver, a solution for the values of

$[\mathbf{x}(t), .., \mathbf{x}(t+k\Delta t)]$ can be calculated. These are then used to generate a new set of way points that serve as input for the control system.

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

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

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.

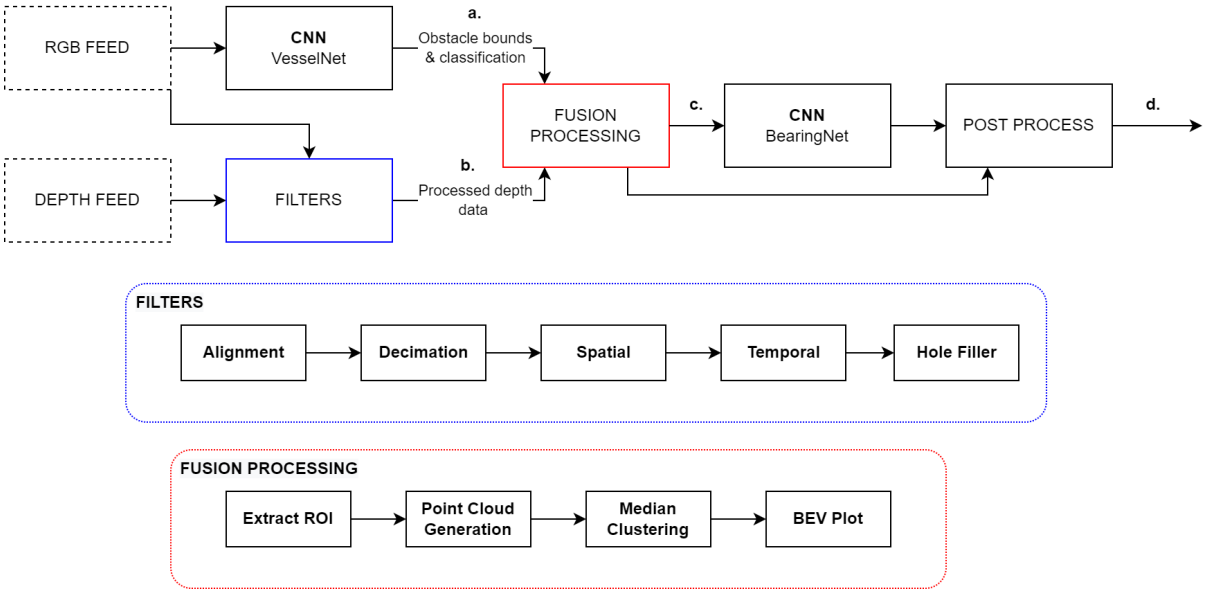

Figure 4: Multimodal Neural Network Architecture for Perception Tasks

The approach to perception proposed in this paper comes in the form of a multi-modal neural network whose architecture is presented in Fig. 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 VessselNet 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 as is well documented in (Huang et al., 2016). The network is trained on a dataset of 1500 annotated colour frames to be capable of detecting two classifications of vessel, specifically the model-scale tugboat and offshore service vessel shown in Figures 2 and 3 respectively.

In parallel to the obstacle detection procedure, the depth feed is processed through to improve the usability of the data. Firstly, the depth frame is aligned with the colour frames to ensure that each pixel of the two frames match. Following which, a decimation filter is applied as a sub-sampling technique to smooth the data, improving later point-cloud visualisation. Spatial and temporal filters are further applied to improve edge sharpness and

remove noise. Finally dead-pixels that remain in the depth feed after decimation are accounted for by a hole-filling operation, by replacing these instances by the largest depth value from its surrounding neighbours.

The obstacle bounds supplied by the VesselNet are utilised to extract the region of interest from the processed depth data creating a cropped depth matrix. A point-cloud is then generated by mapping 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 2D birds-eye-view plot (BEV) of the processed point-cloud data. 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 with the ResNet50 feature extractor, which on top of conducting traditional 2D object detection, is also endowed with the capacity to infer object orientation. The bounding box angle (θ) 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, the overall Control Design methodology is presented for the GNC scheme. This paper utilises a Model-Based design approach to exploit an analytical solution to the control problem while analyzing the dynamics of a ship system. Two different control methods will be analyzed and compared hereon. The first method is based on the classical control theory analyzing the main properties of a vessel's mathematical model and its interaction with the control law derived. Whilst the other method stems from the optimal control field, a so-called Model Predictive Control, where a cost function and constraints allow the designer to better set the trade-off between performances and control effort.

Six degrees of freedom are involved in the motions that take place when a vessel navigates on the water and is subjected to environmental disturbances like wind, waves, and currents. It is possible to obtain the equations of motion by utilizing either the Newton–Euler or the Lagrange equations. A vessel hull with constant mass m and center of gravity (x_g, y_g, z_g) can indeed be described by the following coupled differential equations (Fossen, 2011):

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

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

However, to better exploit the physical properties of the model, a vectorial representation of the equation of motion can be adopted. This mathematical representation is advantageous for exploiting better mathematical system properties that allow the designer to reduce the number of coefficients needed for control, rearranging the equation to obtain semi-definite positive and skew-symmetric matrices (Fossen, 2011; Sciavicco L., 1998). The compact vessel model can then be written as:

$$M\dot{\nu} + C(\nu)\nu + D\nu = \tau. \tag{14}$$

ν the vector representing the generalized velocities $\nu = [u, v, r, w, p, q, r]'$, $\tau = [X, Y, Z, K, M, N]$ the vector of the generalized force acting on the craft, M, C and D are respectively the rigid-body inertia matrix, the Coriolis and centripetal forces matrix and the Damping matrix.

An alternative to a 6DoF vector formalism 14 is to use manoeuvring theory to describe the motions of a marine vessel in 3 DoF. Since the motion components describe the horizontal motion of a ship in surge, sway and yaw, the state vector is chosen to be $\nu = [u, v, r]'$ and this implies that the motion related to heave, roll and pitch are neglected, i.e. $w = p = q = 0$. The last model obtained will be further investigated in this paper since the main requirement of the control design is to allow a vessel to autonomously navigate in a prescribed path in a *xy* Cartesian plane.
Based on the 3DoF model described, further analysis of a ship dynamic manoeuvring will be addressed in this section to better demonstrate how a combination of different mathematical models, with different mathematical properties, can be used to design an appropriate control law. It is possible to summarize the different sailing scenarios of a surface vessel like the ones in the following:
1) Unmooring. The first stage when a ship starts its departure. In this scenario surge, sway and yaw velocities are all very small and of the same magnitude: $u = v = r \approx 0$;
2) Low velocities Manoeuvrings. This is the stage antecedent to the actual navigation towards the destination. It usually occurs in a harbor environment where the ship performs slow maneuvers, preparing itself to point at the destination: $u = v = r < \varepsilon$
3) Sailing at cruise speed. This is the stage where the vessel is pointing at the destination, navigating at cruise speed where surge velocity is the dominant dynamic and heading change is relatively small compared to it. $u = u_0; v \approx 0; r < \varepsilon$.
In order to have a complete mathematical representation of the different vessel navigation scenarios, the following model is proposed in this paper:

$$M\dot{\nu} + C(\nu)\nu + D\nu = \tau \qquad u \approx v \approx r \tag{15}$$

$$\begin{cases} u = \sqrt{u^2 + v^2} \\ T\ddot{\psi} + \dot{\psi} = K\delta \end{cases} \qquad u \gg v \gg r \tag{16}$$

Where 16 is the so-called Nomoto Model (Nomoto, 1967): a yaw subsystem derived from the manoeuvring model 14 by choosing the yaw rate *r* as the output. This model represents the correlation between a vessel's heading and the propellers' angle when a vessel is sailing at constant surge speed u_0. It is a linear single input single output (SISO) model; therefore a frequency response analytical PID can be adopted (Aström and Murray, 2004). Figure 5

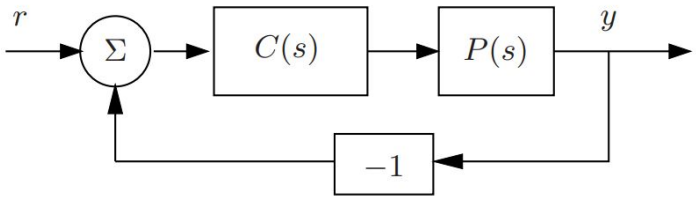

Figure 5: Feedback Loop generalized block-scheme

shows a block scheme of a standard negative feedback control architecture. *C* is the transfer function of the Controller, while *P* represents the vessel dynamic model. The open-loop transfer function is defined as $L = C \times P$ while the closed-loop relation between the reference input *r* and the output *y* is $W = \frac{L}{1+L}$. The strategy presented in this paper is to define *C* such as:

$$W = \frac{L}{1+L} = \frac{K_{dc}\omega_n^2}{s^2 + 2\xi\,\omega_n s + \omega_n^2} \tag{17}$$

where:

$$T_s \approx \frac{4.6\xi}{\omega_n} \tag{18}$$

$$T_r \approx \frac{1.8}{\omega_n} \tag{19}$$

$$\xi = \frac{-ln(M_p)}{\sqrt{\pi^2 + ln^2(M_p)}} \tag{20}$$

The control design main requirement is to achieve:

- zero steady state error between r and y, i.e. the ship is able to reach the destination coordinates given by the Guidance strategy;
- zero overshoot M_p in following heading reference;
- Avoid the obstacles present in the environment respecting the safety distance requirements settled by the Guidance law.

The other control strategy investigated in this paper is the Model Predictive Control (MPC). In this paper, MPC is based on the iterative optimization of a mathematical vessel model over a finite time horizon. The current plant state is sampled at time t and a cost-minimizing control strategy is calculated using a numerical minimization technique. This is assessed for a fixed time horizon in the future $[t, t+T]$. The prediction characteristic of MPC allows the control algorithm to find an analytical solution adding an interesting property to the overall Control architecture: defining a trade-off between plant output and control effort needed. The model 14 needs to be adapted and the overall MPC problem can be written in the following way:

$$\begin{cases} \dot{\nu} = \left(\frac{C(\nu)}{M} + \frac{D}{M}\right)\nu + B\tau \\ y = C\nu \end{cases} \tag{21}$$

$$\begin{aligned} J &= \min(R_s - Y)^T (R_s - Y) + \Delta\tau^T \bar{R} \Delta\tau \\ \text{s.t.} \quad & \tau_{min} \leq \tau \leq \tau_{max} \\ & \dot{\tau}_{min} \leq \dot{\tau} \leq \dot{\tau}_{max} \end{aligned} \tag{22}$$

Where R_s is the data vector that contains the system set-point, e.g. the values of the states the ship system need to achieve, $\bar{R}$ is a diagonal matrix in the form $\bar{R} = r_w I_{N_c \times N_c}$ $(r_w \geq 0)$ where r_w is used as a tuning parameter for the desired closed-loop performance. Moreover, Y is the vector containing the predicted output variables in the prediction horizon and $\Delta\tau$ is the vector containing the future control trajectory calculated by the MPC solver algorithm.

$$Y = [y(k_i+1|k_i), y(k_i+2|k_i), \cdots, y(k_i+N_p|k_i)]^T \tag{23}$$

$$\Delta\tau = [\Delta\tau(k_i), \Delta\tau(k_i+1), \cdots, \Delta\tau(k_i+Nc-1)] \tag{24}$$

The optimization problem 22 that will minimize the error between the actual state of the vessel and the reference given by the Guidance strategy will be solved using commercial available software. The pseudo-code 1 describes the algorithm structure.

Algorithm 1 MPC Controller

```
function SENDMPCCONTROLINPUT
    Tito Neri Model ← Define Tito Neri model parameters
    N_p ← Define prediction Horizon
    N_c ← Define Control Horizon
    J ← Define Cost Function based on Tito Neri Model
    Define control input constraints (11) (12)
    Mainloop:
    if i > N_p then return Prediction Horizon achieved
    if i > N_c then
        Solve optimization problem
        u ← Calculate optimal control input
    x ← Update vessel model with the predicted states
    y ← Update and store the plant output
    Send Control input to the ship
    goto Mainloop.
```

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 decimeters 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 Fig. 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.

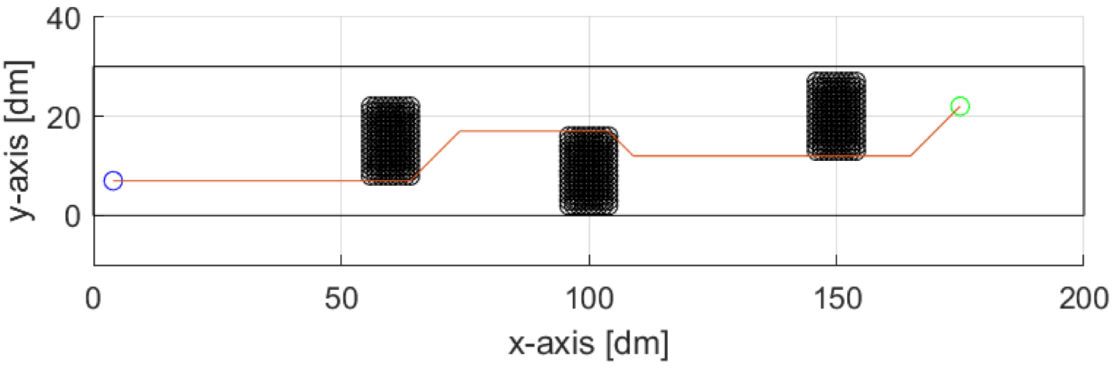

Figure 6: Global MILP Trajectory planning

As shown in the plot, the algorithm provides a pathway whilst minimizing 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 optimization 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.

The sub-figures presented in Fig. 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 Fig. 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 Eq. 25 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

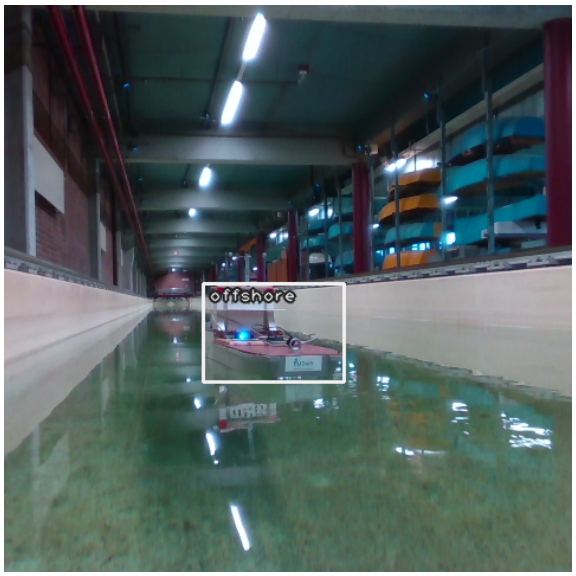

Figure 7 (a): Output from VesselNet Neural Network

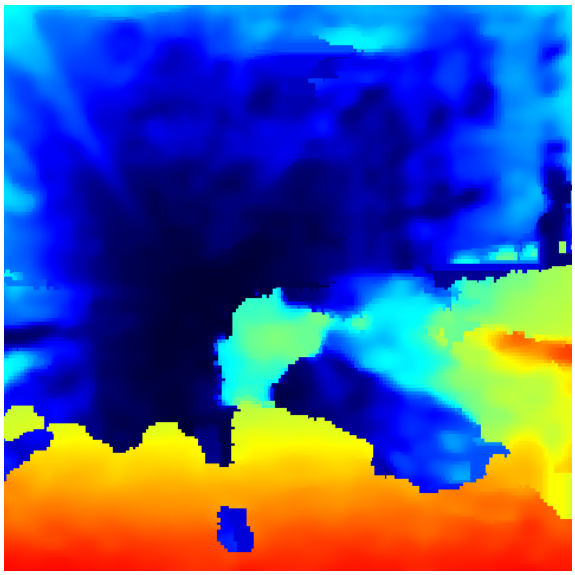

Figure 7 (b): Colour-map Visualisation of Processed Depth Data

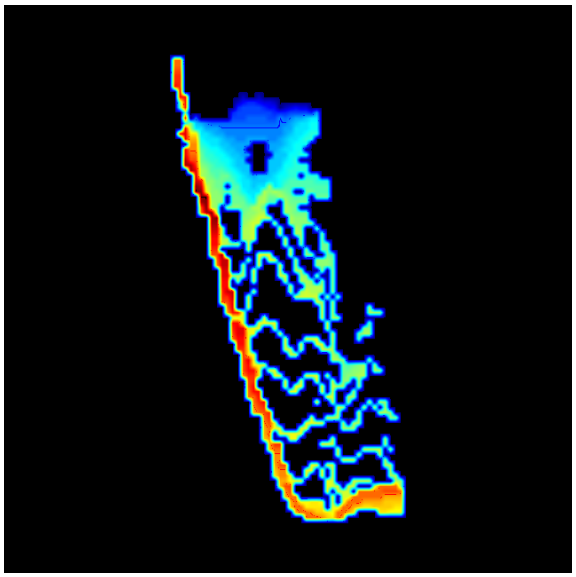

Figure 7 (c): Birds-eye-view Plot of Fused Data

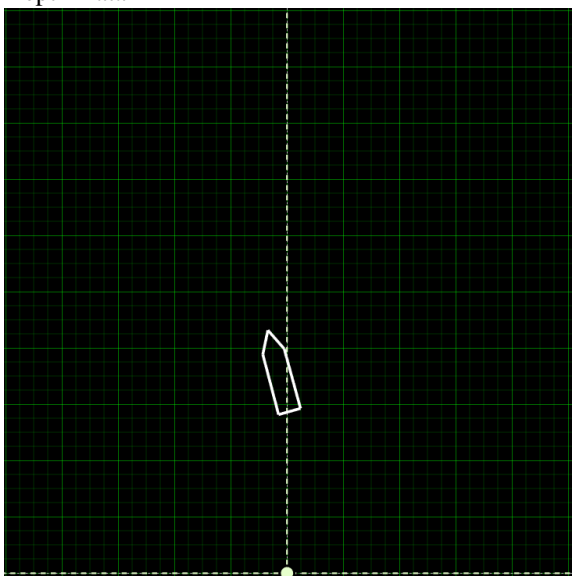

Figure 7 (d): Planar Occupancy Map - *Grid size: 1m, coordinate origin marked by white dot.*

Figure 7: Perception System Results

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.

$$Recall = \frac{correct\ detections}{all\ ground\ truths} \tag{25}$$

3.3 Control Design

Results are obtained using one of the model scale vessel available at RAS, a Tugboat vessel with a scale factor 1:30 called Tito Neri. 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 summarized in table[1].

The first step is to identify the Nomoto Model parameters in (14). 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 Department at Delft University of Technology. The tests were performed reaching steady surge velocity on the Tito Neri model scale ship and injecting at time t_0 a deviation of 45° angle on

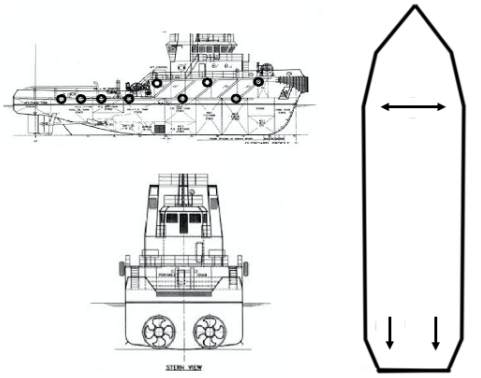

Figure 8: Tito Neri's thrust allocation

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

Table 1: Tiro Neri's Hardware ad Sensor

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.

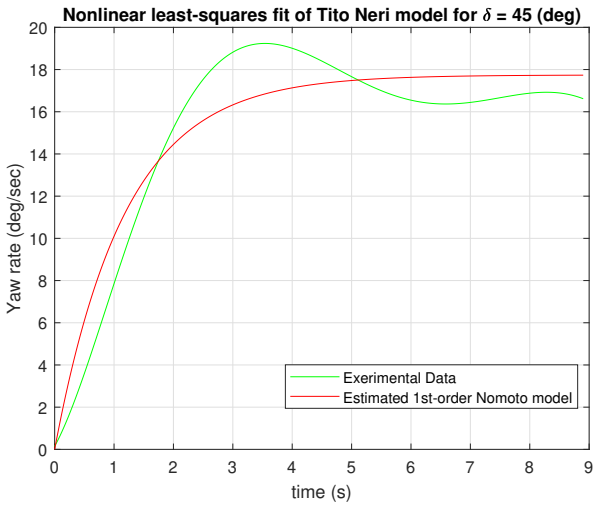

Figure 9: Estimated Nomoto Model

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

$$P = \frac{\psi}{\delta}(s) = \frac{3.55}{s(1+1.67s)} \tag{26}$$

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

$$C = \frac{K_c}{s+a} \tag{27}$$

that fulfill all the control requirements (19),(18),(20) with $K_c = 1.34$ and $a = 2.57$. Figure 13 shows the dynamic evolution of the controlled Tito Neri vessel following the reference path provided by the Guidance algorithm. The ship is able to follow the reference commands with stable dynamics avoiding all the obstacles in the environment, however some path deviation is present in the overall navigation, mainly along the curvature. Actually, the small transitory deviations are to be expected since the control law was designed taking into account only the heading dynamics, neglecting the physical correlations between sway and yaw motion. The MPC design formulation (22), a full model-based control strategy, allows the controller, on the other hand, to better compensate for the Multi Input Multi Output dynamics since it takes into account a full 3DoF ship model (21). As a result, taking into account the correlated dynamic between lateral and angular motion, Fig. 11 shows how the control algorithm allows the ship to follow the reference route with zero path deviation. Nevertheless, it is interesting to analyze how the control effort requested by the MPC controller is quite intensive on the actuator side. Fig. 12 shows how the force requested by the control law continuously change with a frequency of about 0.2 Hz, resulting in an heavy demand to the actuator. This is mainly due to compensate for sway drift during navigation resulting,however, in aggressive actuator dynamics leading to potential mechanical failure. The control key performance indicators (KPI) of the two strategies analyzed in this paper are summarized in Table 2.

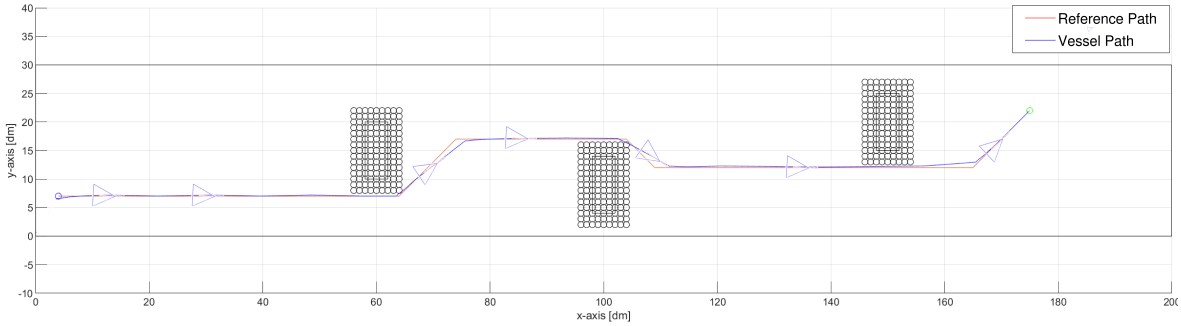

Figure 10: *xy* Cartesian Plot using PID controller

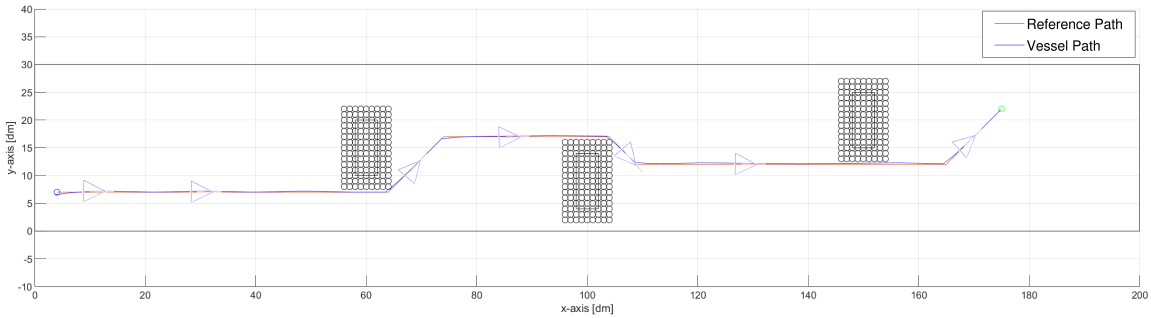

Figure 11: *xy* Cartesian Plot using MPC controller

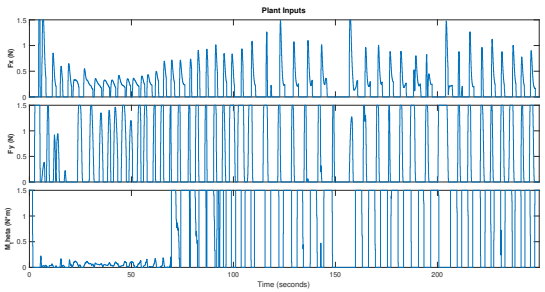

Figure 12: MPC - Control Forces requested

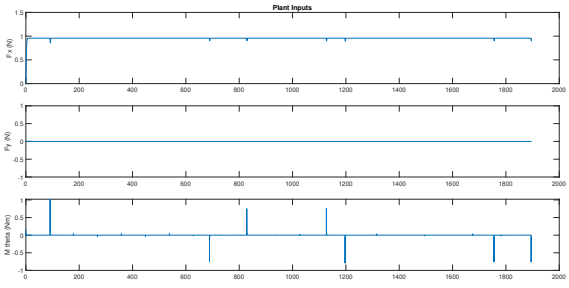

Figure 13: PID - Control Forces requested

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 2: KPIs Control Strategies

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. Due to the computational load of the algorithm, application to real-time path planning may require greater algorithmic efficiency. It is however case dependant whether the computational burden is tolerable or not.

The perception approach investigated in this work demonstrates the promise of a multi-modal network architecture in achieving situational awareness for obstacle avoidance activities. 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 stereovision sensor can provide sufficient point-cloud resolution to detect vessel orientation, albeit within the range capacity of the device's hardware. Although the overall performance of the BearingNet limited the performance of the overall 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.

Finally, two model-based control design approach were presented in this paper. Both of them are suitable candidate to provide autonomous navigation in combination with the Guidance and the Navigation system presented. However while MPC resulted as a more accurate path follower controller, an analytical control strategy based on the second-order frequency response correlations is to be preferred for implementation purpose due to its robustness, faster computational time and less demanding effort on the actuator side. Further research would need to better investigate how to decrease the computational time of more advanced and, theoretically, better performing control strategy as Non-linear Model Predictive Control and back-stepping to allow the overall control system to work in combination with guidance and navigation method that are computationally resource consuming.

Acknowledgement

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.

References

Aström, K.J., Murray, R.M., 2004. Feedback Systems: An Introduction for Scientists and Engineers.

Beltrán, J., Guindel, C., Moreno, F.M., Cruzado, D., Garcia, F., de la Escalera, A., 2018. Birdnet: A 3d object detection framework from lidar information, pp. 3517–3523. doi:10.1109/ITSC.2018.8569311.

Chanchal Dey, S.K.S., 2020. Industrial Automation technologies.

Choy, C.B., Stark, M., Corbett-Davies, S., Savarese, S., 2015. Enriching object detection with 2d-3d registration and continuous viewpoint estimation, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2512–2520. doi:10.1109/CVPR.2015.7298866.

Coleman, D., . Obstacle avoidance and path planning using mixed integer programming .

Corić, M., Mandzuka, S., Gudelj, A., Lušić, Z., 2021. Quantitative ship collision frequency estimation models: A review. Journal of Marine Science and Engineering 9, 533. doi:10.3390/jmse9050533.

Fossen, T.I., 2011. Front Matter. John Wiley Sons, Ltd. pp. i–xix. doi:https://doi.org/10.1002/9781119994138.fmatter.

Haseltalab, A., Garofano, V., van Pampus, M., Negenborn, R., 2020. Model predictive trajectory tracking control and thrust allocation for autonomous vessels. IFAC-PapersOnLine 53, 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, R.R., 2021. Collision Avoidance for Autonomous Inland Vessels using Stereovision. URL: 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., Murphy, K., 2016. Speed/accuracy trade-offs for modern convolutional object detectors .

Kato, S., Takeuchi, E., Ishiguro, Y., Ninomiya, Y., Takeda, K., Hamada, T., 2015. An open approach to autonomous vehicles. IEEE Micro 35, 60–68. doi:10.1109/MM.2015.133.

Khalid, Z., Mohamed, E.A., Abdenbi, M., 2013. Stereo vision-based road obstacles detection. 2013 8th International Conference on Intelligent Systems: Theories and Applications, SITA 2013 , 1–6doi:10.1109/SITA.2013.6560817.

Nomoto, K., 1967. A simplified analysis on ship motion under manoeuvre and proposed steering .

Prasad, D.K., Rajan, D., Rachmawati, L., Rajabally, E., Quek, C., 2017. Video processing from electro-optical sensors for object detection and tracking in a maritime environment: A survey. IEEE Transactions on Intelligent Transportation Systems 18, 1993–2016. doi:10.1109/TITS.2016.2634580.

Sciavicco L., S.B., 1998. Modeling and control of robot manipulators, lorenzo sciavicco and bruno siciliano. Journal of Intelligent and Robotic Systems 21.

Shao, Z., Wu, W., Wang, Z., Du, W., Li, C., 2018. Seaships: A large-scale precisely annotated dataset for ship detection. IEEE Transactions on Multimedia 20, 2593–2604. doi:10.1109/TMM.2018.2865686.

Sánchez-Beaskoetxea, J., Basterretxea-Iribar, I., Sotés, I., Machado, M., 2021. Human error in marine accidents: Is the crew normally to blame? Maritime Transport Research 2, 100016. doi:10.1016/j.martra.2021.100016.

Tom Schouwenaars, Bart De Moor, E.F.J.H., 2001. Mixed integer programming for multi-vehicle path planning .

Vielma, J.P., 2015. Mixed integer linear programming formulation techniques. SIAM Review 57, no. 1 (January 2015): 3–57. .

Wang, W., Gheneti, B., Mateos, L.A., Duarte, F., Ratti, C., Rus, D., 2020. Roboat: An autonomous surface vehicle for urban waterways , 6340–6347doi:10.1109/iros40897.2019.8968131.