Proceedings of the ASME 2023 43rd International Conference on Ocean, Offshore and Arctic Engineering
OMAE2024
June 9-14, 2024, Singapore EXPO, Singapore

OMAE2024-126768

DEVELOPMENT OF A HIGH FIDELITY MULTI-AGENT MARITIME SIMULATOR

Mohammed Ibrahim M[1,†,*], Vallabh Deogaonkar[1,†,*], Abhilash Somayajula[1],

[1]Marine Autonomous Vessels Laboratory,Indian Institute of Technology (IIT) Madras,Chennai,India

ABSTRACT

The maritime industry has experienced significant growth in autonomous systems in recent years, with substantial research dedicated to developing diverse algorithms for the autonomous navigation of surface vessels. To thoroughly gauge the credibility of such autonomy algorithms often requires evaluation across a diverse range of scenarios. Due to the vast number of potential scenarios that necessitate testing, it is impractical to conduct all these tests in a practical manner. Furthermore, these tests are not one-shot endeavors and need to be evaluated with the changing maneuvering characteristics of the vessel as it evolves over its operational years. This paper introduces the concept of a novel multi-agent, adaptive simulator designed specifically for testing various autonomy algorithms for surface vessels such as path planning and collision avoidance. The key highlight of the simulator lies in its potential to progressively improve the accuracy of the hydrodynamic models of the vessels present in the simulation.

Keywords: Multi-agent simulator, System Identification, ROS, 3js, Ship maneuvering

1. INTRODUCTION

Maritime simulators play a pivotal role in advancing the field of autonomous ship navigation by providing a controlled and realistic environment for testing and training algorithms [1]. These simulators serve as indispensable tools for researchers and engineers, enabling them to assess the performance of various autonomy algorithms without the need for physical vessels. In recent years, the maritime industry has witnessed a surge in the development of such simulators, designed to address various challenges in autonomous navigation. These simulators are required to represent the dynamics of the system accurately to ensures a level of realism that closely mirrors real-world conditions. The fidelity of a simulator directly impacts the accuracy and reliability of algorithm testing, making it an essential consideration.

Maritime simulators need to be adaptable because real-world maritime environments are dynamic and complex. Adaptability is crucial for simulators to accurately represent vessel behaviors. Using real-world data, known as data-driven identification, helps refine simulator models to closely match actual vessel behaviors. This iterative process enhances simulation accuracy, making it more realistic.

Accurate representation of calm water hydrodynamics is also vital for simulators as mentioned in [2]. Calm water conditions act as a baseline for understanding how a vessel behaves without external disturbances. Ensuring the simulator captures these inherent characteristics is crucial for reliable algorithm testing, providing a dependable baseline for evaluating different algorithms' impact on vessel behavior.

In a maritime setting, interactions between multiple vessels are complex and dynamic , necessitating the integration of multi-agent simulation for a comprehensive evaluation of algorithmic performance [3].Cooperative autonomy, where vessels work together, is a key application of these interactions. Simulating how multiple vessels interact helps researchers see how well autonomy algorithms handle the challenges of actual maritime operations. This is crucial for convoy navigation, where vessels travel together while staying safe and following rules. Simulating these cooperative behaviors also helps test and improve communication between vessels, ensuring they can share information and coordinate effectively in dynamic environments

Highlighting the importance and widespread use of ROS (Robot Operating System) in robotics is crucial. ROS serves as a fundamental framework for developing and controlling robotic systems, fostering a collaborative and standardized approach across the robotics community. Its versatility and extensive set of tools make it a go-to choice for various robotic applications.

In the context of maritime simulation, several popular simulators leverage ROS and Gazebo for their robustness and adaptability. Notable examples include VRX (Virtual RobotX)[4] and USV sim (Unmanned Surface Vehicle Simulator)[5]. These simulators capitalize on the capabilities of ROS, enabling seamless integration of autonomy algorithms and facilitating realistic sim-

†Joint first authors
*Corresponding author: na21b046@smail.iitm.ac.in

ulations of maritime scenarios. The utilization of ROS in such simulators underscores its pivotal role in advancing the field, providing a standardized platform for the development and testing of autonomous maritime systems.This integration facilitates seamless transition and deployment of algorithms from simulation environments to physical vessels.

The paper underscores the importance of a robust visualization setup in maritime simulators. Leveraging 3D visualization libraries like 3js is imperative for providing high-quality, client-based visualization. This not only aids researchers and engineers in understanding algorithm behavior but also contributes to the potential use of the simulator in human training scenarios and the simulation of various sensors, including cameras.

This paper presents the approach towards developing a high fidelity, adaptive simulator with multi-agent capabilities, ROS integration, and 3js visualization libraries. Drawing insights from key research contributions, this study delves into the critical elements that collectively shape the simulator for the development and testing of autonomy algorithms.

2. OVERVIEW

This section provides an overview of the methodology described in this study for creating an adaptable hydrodynamic simulator. The methodology is based on the idea of data driven identification of maneuvering model using free running maneuvers. In this study, the data from standard maneuver trials has been considered for identifying the 3DOF maneuvering model of a scaled vessel. Before diving into the details of the adaptive methodology, the details of physics modelling of the simulator are covered in Section 3 and further details of the procedure have been included in Section 4. The vessel considered in this study is the Panamax class single-screwed Kriso Container Ship (KCS) [6]. The choice of the KCS vessel for this study is underpinned by its prominence as a popular research vessel, coupled with the availability of a 1:75.5 scaled model of the vessel to conduct experimental study. The principal particulars of the full scale and model scale vessel are given in Table 1. Since there has been a considerable amount of research done on analysis of this vessel, it becomes possible to verify and improve the accuracy of the proposed methodology using other studies as a baseline.

TABLE 1: KCS principal particulars

Main Particulars	full scale	model scale
Scale	1	1:75.5
Length $L(m)$	230	3.0464
Breadth $B(m)$	32.2	0.4265
Draft $d(m)$	10.8	0.143
Displacement $\nabla(m^3)$	52030	0.1209

The fully fitted vessel used for experiments is shown in Figure 1. This model named "Matsya" is powered by a 48V lead acid battery. The entire system consists of a stepper motor that actuated the rudder, a BLDC motor that is attached to the propeller, onboard controllers, the motor drivers for stepper and BLDC motor and the sensors. The onboard computer on the vessel was a Raspberry Pi 4 Model B with Ubuntu Server 20.04 LTS operating system and ROS (Robot Operating System) installed. The sensors were interfaced with using their respective ROS drivers. A GPS and an IMU sensor were placed onboard the vessel to measure its position, linear acceleration, heading and angular velocity.

FIGURE 1: Free running model equipped with sensors

3. PHYSICS MODELLING

The fidelity of a maritime simulator depends on its ability to faithfully replicate the real movement and behavior of vessels in water. The study of ship maneuvering aims to achieve this through the integration of hydrodynamics models, which are governed by Newton's laws of motion expressed as a set of differential equations. These equations encompass both linear and nonlinear terms, collectively forming the maneuvering equations of motion. The vessel specific hydrodynamic parameters in these equations are termed as hydrodynamic coefficients and these are instrumental in achieving the required level of accuracy of simulation. Various maneuvering models, such as the MMG model (Maneuvering Modelling Group) [7] and the Abkowitz model [8], employ different sets of parameters and coefficients to accurately simulate vessel dynamics.

3.1 Kinematics

Two frames of reference play a pivotal role in most maneuvering models: the Global Coordinate System (GCS) and the Body-Fixed Coordinate System (BCS). The kinematics expressions relate the physical quantities related to the vessel states in these two frames.

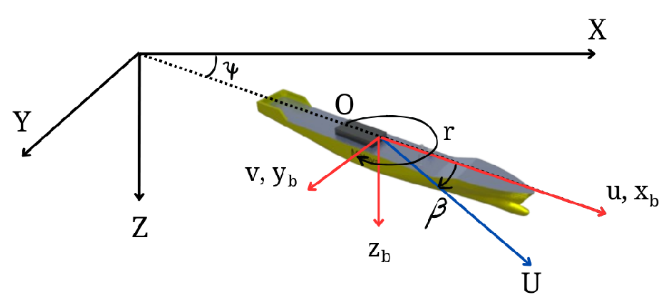


FIGURE 2: Representation of ship kinematic variables

3.1.1 Global Coordinate System (GCS). The GCS presents a global reference for the simulator and the position and orientation of the vessel is expressed with respect to this frame. In this study, the popular NED convention has been followed for this frame. In Figure 2 this frame is denoted with X-Y-Z axes.

3.1.2 Body-Fixed Coordinate System (BCS). Concurrently, the BCS is a body-fixed reference frame and all the variables used for expressing the dynamics, such as the the surge (u) , sway (v) , and yaw (r) velocities of the vessel are expressed in this frame. Its origin is positioned at the intersection of midship, centerline, and waterplane of the vessel. The x-axis (x_b) aligns with the bow, and the y-axis (y_b) points towards the starboard and z-axis (z_b) points vertically down. The heading of the vessel (ψ) is determined by the angle between the x-axes of the BCS and GCS as shown in Figure 2.

The kinematics expressions relating the variables in these frames are as shown in (1) where $R(\psi)$ is the rotation matrix that transforms from BCS to GCS:

$$\begin{aligned} \dot{\boldsymbol{\eta}} &= [R(\psi)]\boldsymbol{\nu} \\ \boldsymbol{\nu} &= \begin{bmatrix} u & v & r \end{bmatrix}^T \\ \boldsymbol{\eta} &= \begin{bmatrix} x_G & y_G & \psi \end{bmatrix}^T \end{aligned} \tag{1}$$

3.2 MMG model of KCS

Most prior studies have used MMG model for KCS vessel to simulate its dynamics and since it was derived by conducting an extensive set of experiments [9], this study uses the MMG model as the baseline to improve the data driven identification methodology. The Maneuvering Modeling Group (MMG) model was proposed in the Japanese Towing Tank Conference (JTTC). The model follows the Prime II non-dimensionalization system [10]. In this system, forces and moments are non-dimensionalized by $(1/2)\rho L d U^2$ and $(1/2)\rho L^2 d U^2$ respectively where L is the principal length, U is the design speed and d is the draft of the vessel. The mass and mass moment of inertia terms are non-dimensionalized by $(1/2)\rho L^2 d$ and $(1/2)\rho L^4 d$ respectively. Velocities are non-dimensionalized by U and lengths are non-dimensionalized by L.

In matrix form the MMG mathematical model [9] is given by the following equations of motion.

$$\begin{bmatrix} m' + m'_x & 0 & 0 \\ 0 & m' + m'_y & m' x'_G \\ 0 & m' x'_G & I'_{zz} + J'_{zz} \end{bmatrix} \begin{bmatrix} \dot{u}' \\ \dot{v}' \\ \dot{r}' \end{bmatrix} = \begin{bmatrix} X' \\ Y' \\ N' \end{bmatrix} \tag{2}$$

Here $[\dot{u}', \dot{v}', \dot{r}']$ denote the non-dimensional accelerations in the surge, sway and yaw directions respectively. m' and I'_{zz} are the non-dimensional mass and mass moment of inertia respectively. m'_x , m'_y are the non-dimensional surge and sway added masses respectively. J'_{zz} is the non-dimensional added mass moment of inertia in the yaw mode. The numerical values used for these parameters are given in Table 2

The non dimensional external forces and moments acting on the vessel in surge, sway and yaw modes respectively are given by X', Y' and N'. These forces are decomposed into various components as described below :

TABLE 2: KCS Added mass coefficients

Non dimensional parameter	Value
Surge added mass (m'_x)	0.006269
Sway added mass (m'_y)	0.155164
Yaw added mass moment (J'_{zz})	0.009268
Yaw mass moment of inertia (I'_{zz})	0.011432
Mass of the vessel (m')	0.182280

$$\begin{aligned} X' &= X'_H + X'_R + X'_C + X'_P \\ Y' &= Y'_H + Y'_R + Y'_C \\ N' &= N'_H + N'_R + N'_C \end{aligned} \tag{3}$$

The subscripts in the above equations are denoting the components as follows:

1. H : Hydrodynamic forces and moments on the hull
2. R : Rudder forces and moments
3. C : Coriolis forces and moments arising due to non inertial midship reference frame
4. P : Propeller forces.

Since the thrust provided by the propeller is only along the surge direction, the sway and yaw components of propeller forces are taken to be zero. The details of the implementation of the above hydrodynamic forces are as given in [9] and a short description about the modelling approach for each of the components are given further.

3.2.1 Hull forces and moments. The non dimensional forces and moments acting on the hull are expressed as a Taylor series in u', β and r'. Here u' and r' denote the non dimensional surge velocity and yaw rate respectively. β denotes the drift angle, defined as the angle between the total velocity vector of ship and its longitudinal direction, given by $\beta = \tan^{-1}(-v'/u')$ where v' is the non dimensional sway velocity. The non dimensional hull forces and moments are given by (4) and the coefficients values are considered as per [9].

$$\begin{aligned} X'_H &= X'_0 u'^2 + X'_{\beta\beta}\beta^2 + (X'_{\beta r} - m'_y)\beta r' \\ &\quad + X'_{rr} r'^2 + X'_{\beta\beta\beta\beta}\beta^4 \end{aligned} \tag{4a}$$

$$\begin{aligned} Y'_H &= Y'_\beta \beta + (Y'_r - m'_x) r' + Y'_{\beta\beta\beta}\beta^3 \\ &\quad + Y'_{\beta\beta r}\beta^2 r + Y'_{\beta r r}\beta r^2 + Y'_{rrr} r^3 \end{aligned} \tag{4b}$$

$$\begin{aligned} N'_H &= N'_\beta \beta + N'_r r' + N'_{\beta\beta\beta}\beta^3 \\ &\quad + N'_{\beta\beta r}\beta^2 r + N'_{\beta r r}\beta r^2 + N'_{rrr} r^3 \end{aligned} \tag{4c}$$

The hull hydrodynamic forces are primarily influenced by the hull form geometry and the coefficients in (4) vary for different hull forms. Traditionally, discerning these coefficient values requires various captive model tests conducted in specialised facilities such as towing tank and Planar motion mechanism (PMM) facility. Since these facilities are capital intensive, these are not readily available. The data driven modelling strategy using free running maneuvers explained in Section 2 is a potential cost effective alternative.

3.2.2 Propeller force. The non dimensional propeller thrust force along the surge direction is given by

$$X'_p = (1 - t_p)\rho K_T D_P^4 n_P^2 / (0.5\rho L d U^2) \tag{5}$$

where D_p is the propeller diameter, n_P is the propeller revolution rate, K_T is the propeller thrust coefficient and t_P is the thrust deduction factor. The additional resistance introduced by the propeller rotating behind the hull as opposed to the bare hull without propeller behind it is accounted by introducing the thrust deduction factor t_P. The thrust coefficient K_T is obtained by fitting a quadratic curve to the propeller open water test data and for the KCS vessel it is given by (6)

$$K_T = a_0 + a_1 J + a_2 J^2 \tag{6}$$

Here J represents the advance coefficient and is given by (7)

$$J = \frac{u(1-w)}{nD_p} \tag{7}$$

where w represents the effective wake fraction. The effective wake fraction w accounts for the reduction in inflow fluid velocity to the propeller when operating in the wake of the hull. The values of the above mentioned propeller parameters for the KCS hull form are given in Table 3

TABLE 3: Propeller parameters

Parameter	Non-dimensional value
t_P	0.207
D_p	0.03435
n_P	35.86
w	0.355
a_0	0.5228
a_1	-0.4390
a_2	-0.0609

3.2.3 Rudder forces and moments. The non dimensional forces and moments due to rudder are expressed as follows :

$$X'_R = -(1 - t_R)F_N \sin\delta \tag{8a}$$
$$Y'_R = -(1 + a_H)F_N \cos\delta \tag{8b}$$
$$N'_R = -(x_R + a_H x_H)F_N \cos\delta \tag{8c}$$

where δ is the instantaneous rudder angle, t_R is the steering resistance deduction factor, x_R is the non dimensional position of rudder along the longitudinal direction of ship measured from midship and x_H is the non dimensional position of the point where additional lateral force induced by steering is acting. F_N is the non dimensional rudder normal force and the detailed expression along with the necessary parameters are taken as per [9].

It can be noted from Section 3.2.2 and Section 3.2.3 that modelling of propeller and rudder forces also involves estimation of various parameters that capture the interaction of these components with the hull and amongst each other. Data driven modelling approach using free running experiments can aid in effectively capturing the dynamics with presence of these interactions. This is possible because, instead of evaluating the parameters for propeller and rudder separately, this free running data is collected with vessel in its regular operation.

3.2.4 Coriolis forces and moments. The non dimensional Coriolis forces and moments about the midship frame of reference are given by (9)

$$X'_C = m'v'r' + m'x'_G r'^2 \tag{9a}$$
$$Y'_C = -m'u'r' \tag{9b}$$
$$N'_C = -m'x'_G u'r' \tag{9c}$$

3.2.5 Rudder angle variation. In this study, to account for rudder rate constraints to match practical scenarios the actual rudder angle δ is governed by a first order equation as shown in (10). Here δ_c denotes the commanded rudder angle. The actual rudder angle δ evolves slowly for a step change in commanded rudder angle δ_c. A lower value of T_R leads to a faster response time.

$$T_R\dot{\delta} + \delta = \delta_c \tag{10}$$

In addition, the rudder rate $\dot{\delta}$ is also saturated at $\dot{\delta}_{max}$. So the rudder-rate is given by:

$$\dot{\delta} = \begin{cases} \frac{\delta_c - \delta}{T_R} & \text{if } \left|\frac{\delta_c - \delta}{T_R}\right| \leq \dot{\delta}_{max} \\ \dot{\delta}max & \text{if } \frac{\delta_c - \delta}{T_R} > \dot{\delta}max \\ -\dot{\delta}max & \text{if } \frac{\delta_c - \delta}{T_R} < -\dot{\delta}max \end{cases} \tag{11}$$

In this study, the non-dimensional rudder time-constant T_R is taken as 1 and $\dot{\delta}_{max}$ is chosen as 5° per second for the full scale ship.

The force components described so far, only consider calm water hydrodynamics. But testing of realistic scenarios for an autonomous surface vessel also involves modelling of external disturbances such as wind and waves.

3.3 Wind force effect

The wind external disturbance effects have been modelled as described in [11]. The forces and moments from wind are added to the right hand side of (3). The non dimensional wind velocity is denoted as V_w and the direction of wind is given by angle β_w. The angle β_w is measured with respect to the X-axis of GCS frame. The components of non dimensional wind velocity with respect to BCS are then given as follows:

$$\begin{aligned} u_w &= V_w cos(\beta_w - \psi) \\ v_w &= V_w sin(\beta_w - \psi) \end{aligned} \tag{12}$$

These are used to obtain the non dimensional relative wind velocity components along the BCS as $u_{rw} = u - u_w$ and $v_{rw} = v - v_w$. The magnitude of non dimensional relative wind velocity U_{wr} and relative wind angle γ_w with respect to the vessel are given as follows:

$$\begin{aligned} U_{wr} &= \sqrt{u_{rw}^2 + v_{rw}^2} \\ \gamma_w &= \arctan 2(-v_{rw}, -u_{rw}) \end{aligned} \tag{13}$$

The non dimensional wind force W_x , W_y and non dimensional wind yaw moment W_ψ are then calculated as follows:

$$
\begin{aligned}
W_x &= C_{wx}(\gamma_w)\frac{\rho_a}{\rho}A_x U_{wr}^2 \\
W_y &= C_{wy}(\gamma_w)\frac{\rho_a}{\rho}A_y U_{wr}^2 \\
W_\psi &= C_{w\psi}(\gamma_w)\frac{\rho_a}{\rho}A_y L_{OA} U_{wr}^2
\end{aligned}
\tag{14}
$$

where A_x and A_y represent the non-dimensional lateral and longitudinal projected areas of the hull above water in the yz and xz planes in the BCS respectively. The non-dimensional factor for the projected areas is taken as Ld_{em}. The non-dimensional wind coefficients C_{wx}, C_{wy} and $C_{w\psi}$are assumed to be functions of the relative wind direction γ_w. ρ_a is the density of air and ρ is the density of water. L_{OA} represents the non-dimensional overall length of the vessel, non-dimensionalized with respect to the length between perpendiculars L. The values of wind parameters considered in this study are shown in Table 4

TABLE 4: Wind force parameters

Parameter	Value
A_x	0.1064
A_y	0.7601
C_{wx}	$cos(\gamma_w)$
C_{wy}	$sin(\gamma_w)$
$C_{w\psi}$	$0.5sin(\gamma_w)$
$\rho_a(kg/m^3)$	1.225
$\rho(kg/m^3)$	1025

3.4 Wave force effect

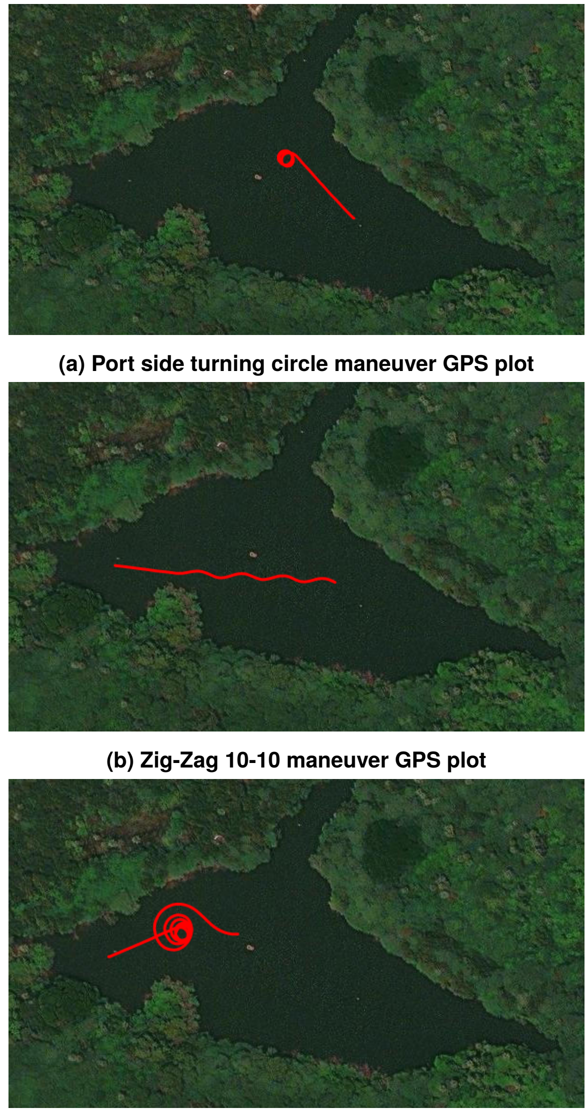
To capture the response of the vessel in real seas, wave force effects need to be considered. This involves modelling of second order wave drift forces [12] and is planned as a future work in this study.

4. DATA DRIVEN IDENTIFICATION

In this study, to capture the hydrodynamics of the model scale vessel three types of standard ship maneuvers were performed.

- Turning circle maneuver
- Zig-Zag maneuver
- Spiral maneuver

These standard maneuvers were all performed in the IIT Madras Institute lake. In total, 3 turning circle maneuvers, 12 zig-zag maneuvers and 10 spiral maneuvers were attempted. The propeller rpm was set to a constant value to achieve the design speed of the vessel for all the tests. The plots of GPS data from 3 sample runs from these experiments are shown in Figure 3a, Figure 3b and Figure 3c. During these experiments, the surge velocity, sway velocity and yaw rate data was recorded along with the rudder angle commands. Since none of the sensors measured the surge and sway velocities directly, they were estimated using a kinematics based Kalman filter implementation [13]. All the measurements were collected at a frequency of 10 Hz. A sample plot of the time series data collected for the spiral maneuvering experiment is shown in Figure 4. The collected data was non-dimensionalized using the model scale design speed and length of the vessel.



(a) Port side turning circle maneuver GPS plot

(b) Zig-Zag 10-10 maneuver GPS plot

(c) Spiral maneuver GPS plot

FIGURE 3: GPS Plots for Different Maneuvers

For system identification, the non-linear Abkowitz model as described in [14] was rewritten for surge, sway and yaw modes into a linear form with respect to the hydrodynamic coefficients as shown in (15).

$$A(k)\cdot X_c = (m' - X'_{\dot u})\dot u'(k) + X'_C(k) \tag{15a}$$

$$B(k)\cdot Y_c = (m' - Y'_{\dot v})\dot v'(k) + (m'x'_G - Y'_{\dot r})\dot r'(k) + Y'_C(k) \tag{15b}$$

$$C(k)\cdot N_c = (m'x'_G - N'_{\dot v})\dot v'(k) + (I'_{zz} - N'_{\dot r})\dot r'(k) + N'_C(k) \tag{15c}$$

Here $\dot u'(k)$, $\dot v'(k)$ and $\dot r'(k)$ denote the non dimensional accelerations, computed at sample times k using 7 point forward,

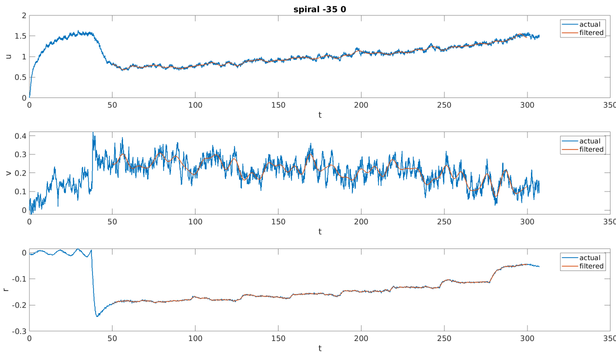


FIGURE 4: Spiral -35 to 0 maneuver data

central and backward differences. $[A, B, C]$ denote the parameter vectors and $[X_c, Y_c, N_c]$ are the coefficient vectors and are given by (16) and (17) respectively.

$$A(k) = [1, \Delta u'(k), \Delta u'^2, v'^2, r'^2, \delta^2, v'\delta, r'\delta \\ \Delta u'^3, v'^2\Delta u' r'^2\Delta u', \delta'^2\Delta u', \Delta u' v' r' \\ \Delta u' v' \delta', \Delta u' r' \delta, v' r' \delta]_{1\times16} \tag{16a}$$

$$B(k) = [1, \Delta u', v', r', \delta, \Delta u'^2, \Delta u' v', \Delta u' r', \Delta u' \delta \\ v'^3, r'^3, \delta^3, \Delta u'^2 v', \Delta u'^2 r', \Delta u'^2 \delta \\ v'^2 r', v'^2 \delta, r'^2 v', r'^2 \delta \delta'^2 v', \delta'^2 r' + v' r' \delta]_{1\times22} \tag{16b}$$

$$C(k) = [1, \Delta u', v', r', \delta, \Delta u'^2, \Delta u' v', \Delta u' r', \Delta u' \delta \\ v'^3, r'^3, \delta^3, \Delta u'^2 v', \Delta u'^2 r', \Delta u'^2 \delta \\ v'^2 r', v'^2 \delta, r'^2 v', r'^2 \delta \delta'^2 v', \delta'^2 r' + v' r' \delta]_{1\times22} \tag{16c}$$

$$X_c = [X_0, X'_u, X'_{uu}, X'_{vv}, X'_{rr}, X'_{\delta\delta}, X'_{v\delta}, X'_{r\delta}, X'_{uuu} \\ X'_{vvu}, X'_{rru}, X'_{\delta\delta u}, X'_{uvr}, X'_{uv\delta}, X'_{ur\delta}, X'_{vr\delta}]^T_{16\times1} \tag{17a}$$

$$Y_c = [Y_0, Y'_u, Y'_v v', Y'_r, Y'_\delta, Y'_{uu}, Y'_{uv}, Y'_{ur}, Y'_{u\delta}, Y'_{vvv} \\ Y'_{rrr}, Y'_{\delta\delta\delta}, Y'_{uuv}, Y'_{uur}, Y'_{uu\delta}, Y'_{vvr}, Y'_{vv\delta}, Y'_{rrv} \\ Y'_{rr\delta}, Y'_{\delta\delta v}, Y'_{\delta\delta r}, Y'_{vr\delta}]^T_{22\times1} \tag{17b}$$

$$N_c = [N_0, N'_u, N'_v v', N'_r, N'_\delta, N'_{uu}, N'_{uv}, N'_{ur}, N'_{u\delta}, N'_{vvv} \\ N'_{rrr}, N'_{\delta\delta\delta}, N'_{uuv}, N'_{uur}, N'_{uu\delta}, N'_{vvr}, N'_{vv\delta}, N'_{rrv} \\ N'_{rr\delta}, N'_{\delta\delta v}, N'_{\delta\delta r}, N'_{vr\delta}]^T_{22\times1} \tag{17c}$$

These hydrodynamic derivatives in (17) were obtained by solving (15) using least squares regression. Using the regressed coefficients, (15) was simulated and compared against the MMG model simulation described in Section 3.2 for 2 standard maneuvers which were not present in the collected data from experiments. The plots of the comparison are shown in Figure 5 and Figure 6. In the plots, the MMG simulation trajectories are labelled as Original and the trajectories obtained using the predicted coefficients from least squares regression are labeled as LS.This difference is primarily attributed to the fact that the collected data from experiments had considerable effect of wind disturbances.

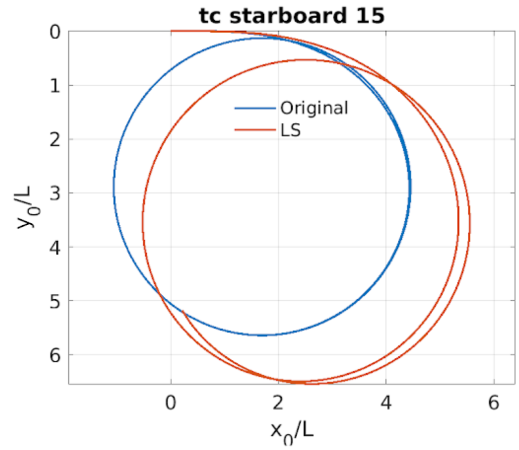


FIGURE 5: Comparison of starboard side turning circle (δ = 15°) MMG simulation : Original, Trajectory from identification : LS

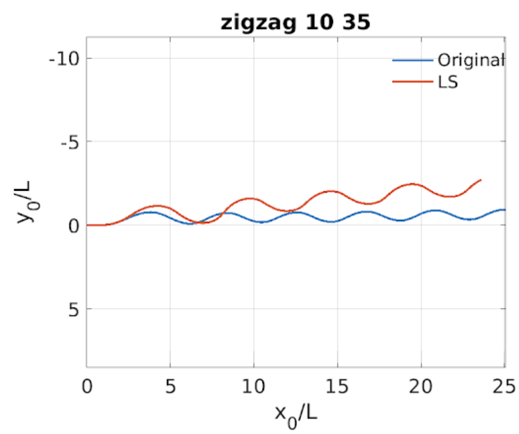


FIGURE 6: Comparison of 10-35 zig-zag maneuver trajectory MMG simulation : Original, Trajectory from identification : LS

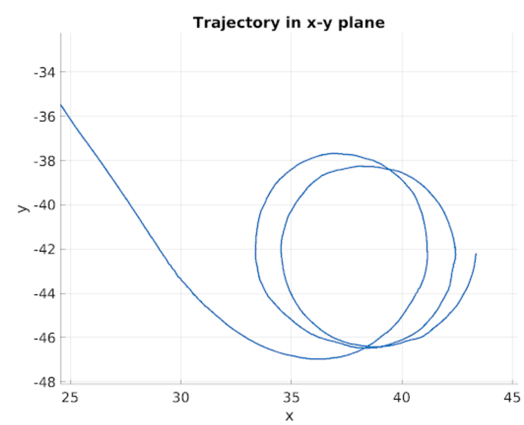


FIGURE 7: Turning circle maneuver experiment showing drifting trajectory

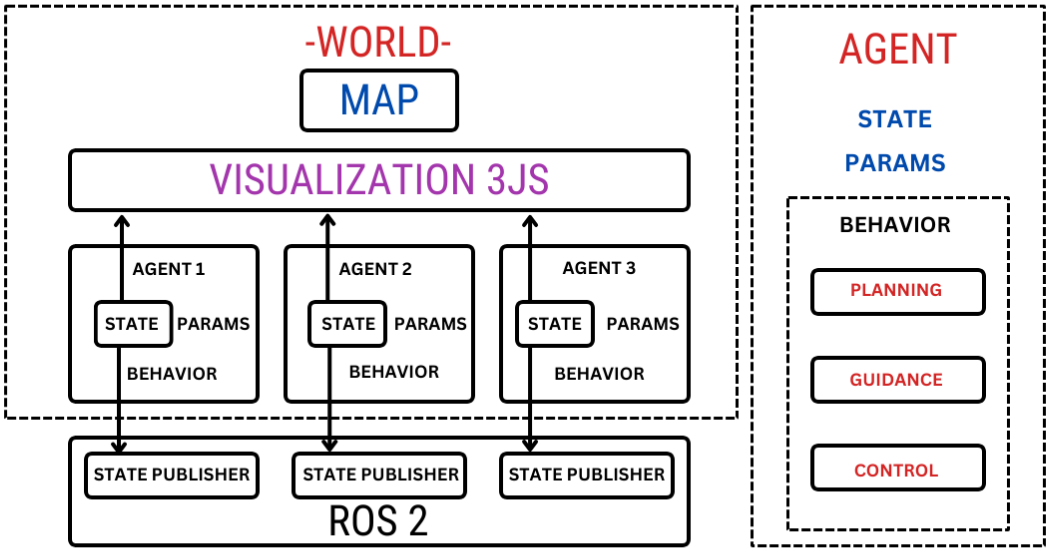


FIGURE 8: Simulator architecture

This resulted in drifting trajectories of the standard maneuvers in the dataset used for identification. A sample turning circle trajectory showing this drift is shown in Figure 7. In future, more experiments are planned and further investigation will be carried out to separate the effects of wind in dynamics and improve the identified model. For the rest of the paper, MMG model has been used as described in Section 3.2 for simulating the vessel dynamics.

5. ARCHITECTURE

This section describes the various classes implemented in the simulator which act as the backbone of its architecture. Figure 8 shows a diagrammatic representation of the overall architecture of the simulator including the main classes and their interlinks amongst each other. The `WORLD` class consists of the parameter `MAP` and the method `VISUALIZATION`. The `MAP` parameter is essentially an array that represents the discretization of the region where the autonomy algorithm is being tested. The elements of the array make a distinction between the regions that are free and the regions that have static obstacles using a cost map. The `VISUALIZATION` module displays the various static and dynamic elements of the `WORLD` and is described in further detail in Section 6. Another important class is the `AGENT` class which is used to simulate all the dynamic agents in the `WORLD`. Multiple instances of the `AGENT` class can be created to simulate multiple vessels. Each instance of the `AGENT` class has a variable to store its current `STATE`, the list of `OBSTACLES` in the simulation and a dictionary `PARAMS` to describe the various parameters of the vessel such as its length, design speed, dynamic model function, etc. Along with these, at the start of the simulation when the agent is initialised, it is assigned a list of three `BEHAVIORS`, one each for `PLANNING`, `GUIDANCE` and `CONTROL`. Each class of `BEHAVIORS` has different functions to implement different algorithms. As an example, `GUIDANCE` class has a function for implementing different types of reactive guidance laws as described in [15]. More detailed examples of the implementation are provided in Section 9.

6. VISUALIZATION

The simulator's visualization component relies on the versatile Three.js library. Three.js, a JavaScript library tailored for 3D graphics in web browsers, simplifies the intricacies of WebGL programming through its high-level abstraction. This empowers developers to construct interactive 3D visualizations seamlessly.Notably, Three.js's compatibility with JavaScript, a widely-supported programming language, makes it an ideal choice for web-based applications. This compatibility ensures smooth integration with other web technologies, enhancing the accessibility and user-friendliness of our simulation platform.

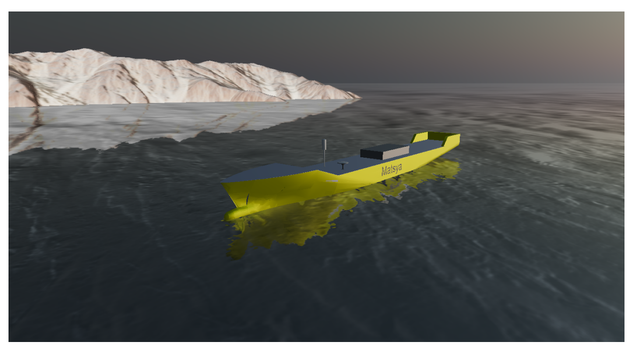


FIGURE 9: Visualisation using Three.js library

An inherent strength of Three.js is its open-source nature and browser-based architecture, aligning perfectly with the simulator's goal of becoming an open-source tool for researchers. Hosting simulations is simplified as Three.js operates within web browsers, eliminating the need for additional software installations.

This accessibility promotes a collaborative environment,facilitating the easy sharing and dissemination of researchers' work. Furthermore, the extensive and active Three.js community enhances the user experience by providing abundant resources and support for customization. Researchers can tailor visualizations to their specific requirements, leveraging the flexibility offered by Three.js.

In the simulator, the connection between the physics engine responsible for hydrodynamic modeling and Three.js is facilitated through WebSockets as shown in Figure 10. WebSockets are a communication protocol that offers several merits, making them particularly well-suited for scenarios requiring real-time and bidirectional data exchange.One of the key merits of WebSockets lies in their low latency. By maintaining a persistent connection between the client and the server, WebSockets eliminate the need for repeated connection establishment, resulting in faster data transfer. Another noteworthy advantage of WebSockets is their bidirectional communication capability. In the context of our simulator, this means that both the hydrodynamic modeling engine and the Three.js visualization can independently send and receive data. This bidirectional communication is critical for ensuring a real-time and synchronized link between the hydrodynamic model's outputs and the visual representation in Three.js.

7. ROS

The integration of the Robot Operating System ROS into the simulator significantly enhances its primary purpose: the testing of multi-agent autonomy algorithms. This enhancement is particularly crucial as these algorithms heavily rely on sensor data from each agent to assess their performance effectively. ROS provides a comprehensive framework for simulating a variety of sensors, including GPS, IMU, LIDAR, and cameras, which is essential for emulating real-world scenarios.

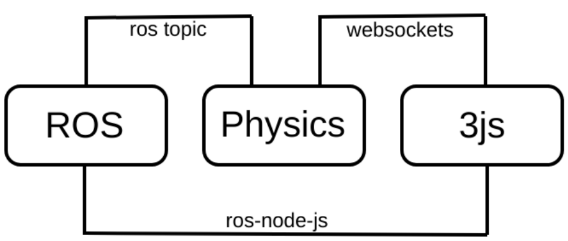


FIGURE 10: ROS integration

In the simulator, the incorporation of ROS has commenced with the initiation of publishing standard odometry topics from the physics modeling through ROS2. These topics are made accessible by the visualization component, allowing it to retrieve data from ROS topics as shown in Figure 10 and simulate the interaction of real-life Autonomous Surface Vehicles (ASVs).This initial implementation not only enables the visualization component to mirror the real-life ASVs' behavior by taking data from ROS topics but also establishes a foundation for the integration of more advanced sensors. Future developments aim to include sensors like lidar and cameras as studied in [16] into the simulator, further expanding its capacity to emulate complex real-world scenarios.

Looking forward, the inclusion of ROS holds promising potential. One notable application is the seamless transition from simulation to real Autonomous Surface Vehicles (ASVs), considering ROS's widespread adoption in the robotics community. This transition would facilitate a more comprehensive evaluation of algorithms in real-world scenarios following their validation in the simulated environment. Furthermore, ROS can be employed for real-time simulation of ASVs during experiments. This capability not only enhances the flexibility of experimentation but also provides a means to evaluate algorithms under dynamic and evolving conditions. As ROS continues to evolve, its integration into the simulator aligns with the cutting-edge developments in autonomous systems and robotics, making a substantial contribution to the advancement of multi-agent autonomy research.

8. MULTIAGENT

The simulator's capacity to handle multi-agents, as mentioned in the Section 5, is a foundational feature. Each agent is endowed with its own path planning, control, and guidance behaviors. The integration of Three.js and WebSockets further strengthens the simulator's multi-agent capability, providing a dynamic and interactive environment for agents to operate.

Looking ahead, the simulator is going to expand its capabilities by incorporating different types of vessels, each characterized by its unique identified dynamics. This evolution will enable a more diverse and realistic representation of maritime scenarios, facilitating comprehensive testing and validation of multi-agent autonomy algorithms.The simulator's ability to handle multiple agents is showcased in the upcoming section Section 9.2, where a detailed test case is presented. This case study highlights the simulator's effectiveness in coordinating multiple agents. This demonstration emphasizes the simulator's versatility and its potential to contribute to advancements in multi-agent autonomy research.

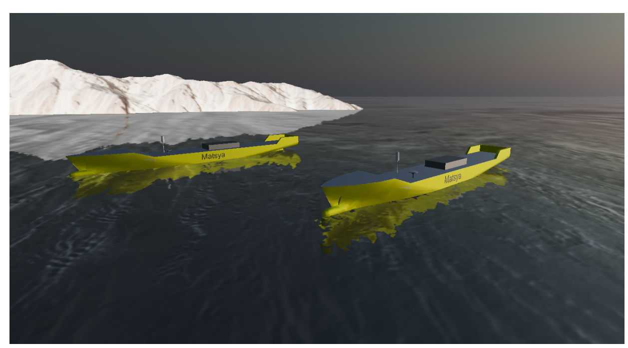


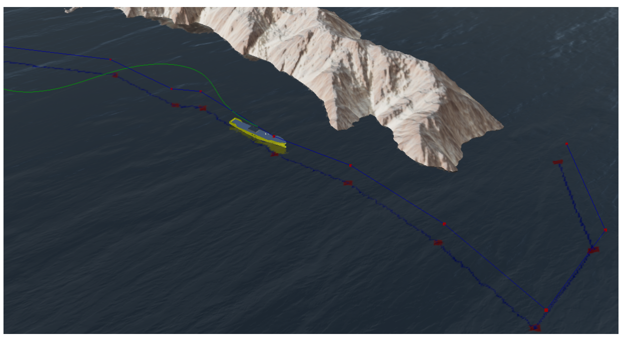
FIGURE 11: Multi-agent simulation

9. CASE STUDIES PERFORMED USING THE SIMULATOR

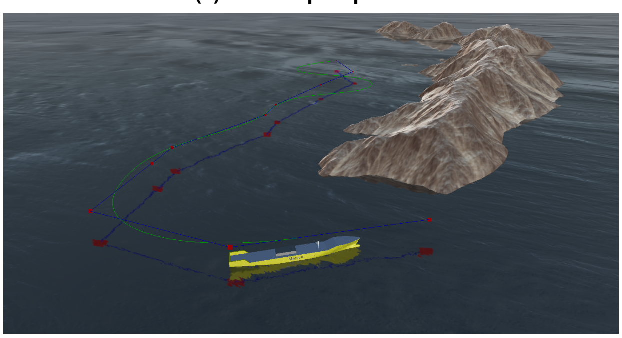
This section shows the demonstration of two test scenarios with different autonomy algorithms tested in the simulator.

9.1 Path following test

In the path following test case of the simulator, a single KCS AGENT was simulated with the PRM (Probabilistic Roadmap) algorithm as the PLANNING behavior, Integral Line Of Sight (ILOS) algorithm [17] as the GUIDANCE behavior and a PD control algorithm [18] as the CONTROL behavior and the path tracked by the vessel is shown in Figure 12a and Figure 12b . Specifically, PRM path planning algorithm was implemented for a single KCS vessel within the simulator's environment. A map featuring static obstacle is provided in the form of a cost map, and the vessel is tasked with planning a trajectory to navigate through this map. The algorithm randomly sampled 100 points in the free space of the map and a maximum of 10 nearest node connections were allowed for each sample to generate a graph. The optimal path for traversal through this graph was generated using A* search algorithm. This test case serves as an illustrative example of how the simulator can effectively model and evaluate path planning strategies for autonomous vessels. PRM sim is the screen recording of the simulator while performing PRM path planning test case.



(a) Distant perspective



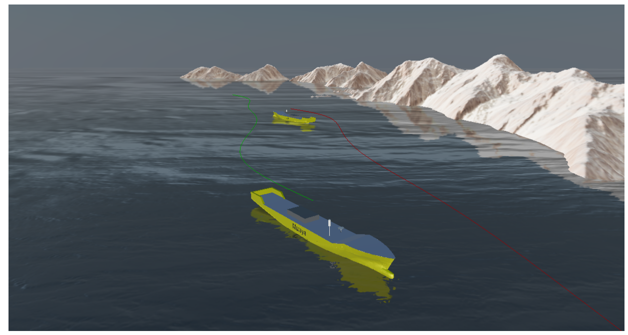
(b) Close-up perspective

FIGURE 12: PRM Path Planning Test

9.2 Multiagent simulation

In the multi-agent test case, the simulator is employed to simulate collision avoidance scenario by creating two instances of KCS AGENT. For both the agents the GUIDANCE behavior was set to modified sink-vortex [15] with PD control algorithm [18] as the CONTROL behavior. Each agent was tasked with reaching the starting point of other agent, navigating through the shared space while avoiding collision. Each vessel has the other agent as a part of its OBSTACLES list. The simulation results are shown in

(a) Distant perspective



(b) Close-up perspective

FIGURE 13: APF Collision Avoidance Test

Figure 13a and Figure 13b where the vessel tracing the green path is the own ship and the vessel tracing the red path is a dynamic obstacle for it. This scenario involves the utilization of Artificial Potential Fields (APF) reactive guidance law for obstacle avoidance, focusing on two vessels within a mapped environment. This test case demonstrates the simulator's capability to model and assess multi-agent collision avoidance strategies, specifically showcasing the application of APF in a dynamic environment with interacting vessels. APF sim is the screen recording of the simulator while performing APF collision avoidance test case.

10. CONCLUSION AND FUTURE WORKS

In conclusion, this study introduces the development of a multi-agent simulator, which has the capability to improve the hydrodynamic models of the vessels being used in it through real-life experiments. The study also provides a comprehensive discussion on the simulator's architecture and visualization components. Real-life experiments were conducted with a model scale vessel to discern its Abkowitz model; however, the results demonstrated inaccuracies attributed to wind effects in the data used for model identification. This highlights the need for further studies to refine the model, specifically accounting for wind conditions.

As part of the future work, the simulator is envisioned to incorporate vessels with distinct hullforms and identified models. Additionally, dynamic models accommodating more degrees of freedom will be integrated, enhancing the simulator's fidelity. Moreover, the simulator's potential application as a bridge simulator will be explored, extending its utility to maritime training and simulations. These future goals underscore the ongoing development and expansion of the simulator to meet evolving research and application needs in the field of maritime autonomy and simulation.

REFERENCES

[1] Vagale, Anete, Osen, Ottar L, Brandsæter, Andreas, Tannum, Marius, Hovden, Christian and Bye, Robin T. "On the use of maritime training simulators with humans in the loop for understanding and evaluating algorithms for autonomous vessels." *Journal of Physics: Conference Series*, Vol. 2311. 1: p. 012026. 2022. IOP Publishing.

[2] Zghyer, Rami. "The Role of Ship Simulators and Maneuvering Models in Maritime Operations." (2023).

[3] Xiao, Zhe, Fu, Xiuju, Zhang, Liye, Zhang, Wanbing, Agarwal, Manu and Goh, Rick Siow Mong. "MarineMAS: A multi-agent framework to aid design, modelling, and evaluation of autonomous shipping systems." *Journal of International Maritime Safety, Environmental Affairs, and Shipping* Vol. 2 No. 2 (2019): pp. 43–57.

[4] Bingham, Brian, Agüero, Carlos, McCarrin, Michael, Klamo, Joseph, Malia, Joshua, Allen, Kevin, Lum, Tyler, Rawson, Marshall and Waqar, Rumman. "Toward maritime robotic simulation in gazebo." *OCEANS 2019 MTS/IEEE SEATTLE*: pp. 1–10. 2019. IEEE.

[5] Paravisi, Marcelo, H. Santos, Davi, Jorge, Vitor, Heck, Guilherme, Gonçalves, Luiz Marcos and Amory, Alexandre. "Unmanned surface vehicle simulator with realistic environmental disturbances." *Sensors* Vol. 19 No. 5 (2019): p. 1068.

[6] Stern, F, Agdraup, K, Kim, SY, Hochbaum, AC, Rhee, KP, Quadvlieg, FHHA, Perdon, P, Hino, Takanori, Broglia, R and Gorski, J. "Experience from SIMMAN 2008—the first workshop on verification and validation of ship maneuvering simulation methods." *Journal of ship research* Vol. 55 No. 02 (2011): pp. 135–147.

[7] Yasukawa, Hironori and Yoshimura, Y. "Introduction of MMG standard method for ship maneuvering predictions." *Journal of marine science and technology* Vol. 20 (2015): pp. 37–52.

[8] Abkowitz, Martin A. "Measurement of hydrodynamic characteristics from ship maneuvering trials by system identification." Technical report no. 1980.

[9] Yoshimura, Yasuo, Masumoto, Yumiko et al. "Hydrodynamic database and manoeuvring prediction method with medium high-speed merchant ships and fishing vessels." *International MARSIM Conference*: pp. 1–9. 2012.

[10] Fossen, Thor I. "Guidance and control of ocean vehicles." *University of Trondheim, Norway, Printed by John Wiley & Sons, Chichester, England, ISBN: 0 471 94113 1, Doctors Thesis* (1999).

[11] Deraj, Rohit, Kumar, RS Sanjeev, Alam, Md Shadab and Somayajula, Abhilash. "Deep reinforcement learning based controller for ship navigation." *Ocean Engineering* Vol. 273 (2023): p. 113937.

[12] Ramachandran, Krishnavelu, Borsutkar, Visharad and Somayajula, Abhilash Sharma. "Comparison Between Experiments and Simulations for a Ship Maneuvering in Waves." *International Conference on Offshore Mechanics and Arctic Engineering*, Vol. 86878: p. V005T06A044. 2023. American Society of Mechanical Engineers.

[13] Welch, Greg, Bishop, Gary et al. "An introduction to the Kalman filter." (1995).

[14] Deogaonkar, Vallabh Vinod, Jadhav, Aditya Kailas, Ramachandran, Krishnavelu and Somayajula, Abhilash Sharma. "Data Driven Identification of Ship Maneuvering Coefficients." *International Conference on Offshore Mechanics and Arctic Engineering*, Vol. 86878: p. V005T06A050. 2023. American Society of Mechanical Engineers.

[15] Jadhav, Aditya Kailas, Pandi, Anantha Raj and Somayajula, Abhilash. "Collision avoidance for autonomous surface vessels using novel artificial potential fields." *Ocean Engineering* Vol. 288 (2023): p. 116011.

[16] Smith, Peter and Dunbabin, Matthew. "High-fidelity autonomous surface vehicle simulator for the maritime RobotX challenge." *IEEE Journal of Oceanic Engineering* Vol. 44 No. 2 (2018): pp. 310–319.

[17] Caharija, Walter, Pettersen, Kristin Y., Bibuli, Marco, Calado, Pedro, Zereik, Enrica, Braga, José, Gravdahl, Jan Tommy, Sørensen, Asgeir J., Milovanović, Milan and Bruzzone, Gabriele. "Integral Line-of-Sight Guidance and Control of Underactuated Marine Vehicles: Theory, Simulations, and Experiments." *IEEE Transactions on Control Systems Technology* Vol. 24 No. 5 (2016): pp. 1623–1642. DOI 10.1109/TCST.2015.2504838.

[18] Mohan, Anand and Sharma Somayajula, Abhilash. "Analyzing Robustness and Accuracy of Different Controllers for Underactuated Ships." *2023 4th International Conference for Emerging Technology (INCET)*: pp. 1–6. 2023. DOI 10.1109/INCET57972.2023.10170098.