Marie Dubromel

Academic Year 2023 – 2024
Research Unit: CROSSING IRL CNRS 2010
Team(s): Robotics
Supervisor(s): Benoit Clement

ROBOTICS SPECIALTY

COLREGs Simulator

Marie Dubromel

CROSSING
French Australian Laboratory for Humans-Automonous Agents Teaming

Academic Year 2023 – 2024

Abstract

The rising popularity of autonomous vessels necessitates the development of robust collision avoidance systems to ensure safe navigation in unpredictable maritime environments. This internship report explains the design and implementation of two simulators, each aimed at addressing the challenges associated with collision avoidance while adhering to International Regulations for Preventing Collisions at Sea (COLREGs). The first simulator, Simulator A, employs artificial potential fields to identify potential collision scenarios, calculate appropriate vector fields, with an adapted controllers. It offers a comprehensive approach to collision avoidance, incorporating real-time monitoring systems to ensure adherence to COLREGs during simulated scenarios. By allowing users to select from a range of agent types representing different categories of unmanned surface vessels (USVs), it facilitates analysis by experienced sailors. In contrast, Simulator B utilises real Automatic Identification System (AIS) data from past USV or vessels trips, enabling the seamless integration of fictive and real-world USV scenarios. By leveraging machine learning techniques, particularly deep reinforcement learning (DRL), these simulators pave the way for adaptive and model-independent collision avoidance systems in various environmental conditions. In conclusion, the development of these simulators help to progress towards building collision avoidance systems that comply with COLREGs for manned, unmanned, and hybrid vessels. By fostering collaboration between private companies and research laboratories, these advancements hold the promise of unlocking new possibilities for safe and efficient maritime exploration. Moving forward, continued refinement and integration of AI-driven enhancements will be crucial in ensuring the safety and efficacy of autonomous and hybrid vessel operations, ushering in a new era of maritime innovation and exploration.

Keywords: Robotics, USV, avoidance collision system, artificial potentials fields, guidance, controller

Contents

1 Introduction 4

2 Search for the existing 4

2.1 The Fossen Simulator 4

2.2 The UTSeaSim Simulator 5

3 Conception of an adapted controller 5

3.1 The rules of the sea according to the COLREGs 5

3.2 Artificial potential field controller 6

4 Avoiding collision system 11

4.1 Overall functioning 11

4.2 Cross-path situation 12

5 The Simulator A 12

5.1 The different types of agents 13

5.2 The structure 13

5.3 The user manual : the initialisation of a simulation 14

6 The Simulator B : AIS data 15

6.1 The AIS data extraction 15

6.2 AIS data processing 16

7 Further updates 17

7.1 The merger of Simulator A and Simulator B 17

7.2 Interpolation on the AIS data 17

7.3 Integration of Maritime Chart Sources 17

7.4 AI 18

7.5 Uncertainties 18

8 Conclusion 19

9 Acknowledgements 19

References 20

A Appendice 21

1 Introduction

Whether for commercial, tourist or military purposes, the use of autonomous vessels is increasingly popular, as they don't require any crew onboard, removing human from risks, and enable far more dangerous and advanced operations in increasingly hostile environments. To fulfil their missions, those vessels require a efficient and trustworthy control and guidance system enabled by recent development in sensor technology. But as operating in harsh, unpredictable waters and around maned and unmanned vessels make accidents more likely, the use of Unmanned Surface Vessels (USV), Autonomous Surface Vessels (ASV), Unmanned Undersea Vehicles (UUV), and Maritime Autonomous Surface Ships (MASS) also require to have a strong avoiding collision system.

The IRL CROSSING is an International (French Australian) Research Laboratory for Humans-Autonomous Agent Teaming irl. They are more specifically building projects upon four different thrusts : New Models to Understand and Anticipate Human Behaviour, New Algorithms for Energy-Efficient and Human-Based AI, New Paradigms for Autonomous Agents/Human Interaction and Understanding and New Solutions for the Management of Hybrid Teams. Therefore, the development of innovative approaches to problems around humans-autonomous agents teaming such as USV are subjects of particular interest to the laboratory. They are currently developing several thesis subjects to do some research about Hybrid Navigation Acceptability and Safety (H-NAS). Indeed, to ensure safe interactions between vessels, the autonomous vessel needs to adhere to the International Regulations for Preventing Collisions at Sea (COLREGs). These regulations were initially designed in the 19th century and have been shaped by human interpretation. Consequently, the rules are presented in unclear language, assuming they would be understood and executed by skilled human sailors rather than an autonomous systems. Furthermore, while implementing the COLREGs in a setting with only unmanned vessels is relatively straightforward, but introducing interactions between unmanned and manned vessels, it adds complexity due to the often unpredictable behaviour of human navigators. Consequently, conventional model-based methods are deemed too intricate to account for the wide array of potential encounters, environmental conditions, and human actions.

Modern machine learning (ML) techniques, particularly deep reinforcement learning (DRL), offer an opportunity for a versatile and adaptive model-independent guidance and obstacle avoidance system. This system could abstract numerous potential interactions and scenarios from past observations, integrating an understanding of the COLREGs into decision-making processes using concepts from game theory. The expected product of this internship is a simulator implemented in Python, with a collision avoidance system on every USV, that adheres to the COLREGs. With this simulator, several research could then use it to train AI using historical AIS-based simulations of real-world scenarios.

2 Search for the existing

Before developing the expected simulator, the realisation of a state of the art is essential in order to identify possible solutions for similar problem around USV in a simualtor. Unfortunately, only few works have been shared, and all the others seem private, done by private companies. These are the two main works that could be used for the project.

2.1 The Fossen Simulator

The Fossen Simulator Fossen [2021] is designed to simulate the behaviour of different types of vehicles in a 3D simulation environment. It takes into account different simulation parameters such as gravity, friction, air resistance, and vehicle dynamics to simulate the movement and behaviour of vehicles in real-time. Users can adjust simulation parameters to represent different types of vehicles and environments. Each vehicle is modelled as an object in Python and the vehicle class has methods for guidance, navigation and control. The main program main.py is used to define vehicle objects for real-time simulation.

To conclude with this simulator, it has a good dynamic modelisation, with the possibility to choose to simulate different types of vehicles, but it does not take into account the concept of collision with other boats, and because of the accuracy of the simulation, the complexity of the program is a bit too high to be run a thousand times in an AI training context. However, the state equations chosen to implement a maritime environment such as waves and currents could be useful to upgrade the USV simulator of the project.

2.2 The UTSeaSim Simulator

The UTSeaSim simulator Fossen [2013] is a multi-agent simulation environment for underwater robotics research. It allows users to simulate underwater vehicles and their interactions with the environment, as well as communication between vehicles and with a surface station. The simulation environment includes several modules, such as a physics engine, a sensor module, a communication module, and a behaviour module. The physics engine simulates the dynamics and kinematics of the underwater vehicles, while the sensor module simulates various sensors such as sonar and vision sensors. The communication module simulates acoustic and radio communication between vehicles and with the surface station. The behaviour module is responsible for controlling the behaviour of the vehicles in the simulation. The simulator also includes a graphical user interface (GUI) for visualising the simulation, controlling the simulation parameters, and monitoring the behaviour of the vehicles (Crase et al. [2013]. Users can interact with the GUI to create and modify scenarios, as well as to run simulations and analyse the results. Overall, the UTSeaSim simulator uses an RRT algorithm to avoid obstacles. So from this simulator, the idea of a GUI window with the choice in the command-line flag, and the possibility to choose the speed of the simulation could be really useful for the new simulator and its users. But this simulator does not take into account any rules from the COLREGs. Moreover their solution to avoid collisions may not work if the obstacles are close, as the robot will follow approximately the RRT path with this method.

3 Conception of an adapted controller

The main objective of this simulator is to both replay a past scenario based on collected AIS data, and add multiple USV in the scene without causing any disturbance. But it must accomplishes that while also taking into account the rules of the COLREGs to avoid any kind of collision. To do so, all the USV need to have an adapted controller to become autonomous while respecting the rules of the COLREGs and physcically move as an USV.

3.1 The rules of the sea according to the COLREGs

To ensure the safe navigation of autonomous vessels during encounters with other vessels, adherence to the International Regulations for Preventing Collisions at Sea (COLREGs) is essential COLREGs [1972], Cockcroft and Lameijer [2012]. Here is a list of the relevant COLREGs rules that have been implemented in the simulator

- Rule 8 - *Actions to avoid collision*: if there is sufficient sea-room, alteration of course alone may be most effective. Reduce speed, stop or reverse only if necessary.
- Rule 13 - *Overtaking*: Any vessel overtaking any other shall keep out of the way of the vessel being overtaken. It can overtaken on both sides and the boat which is being overtaken has the priority.
- Rule 14 - *Head-on*: Each head-on vessel shall alter her course to starboard so that each shall pass on the port side of the other. This rule can also be called the *'Red to Red'* rule.
- Rule 15 - *Crossing*: The vessel which has the other on her own starboard side shall keep out of the way.
- Rule 16 - *Actions by give-way vessel*: Take early and substantial action to keep well clear.

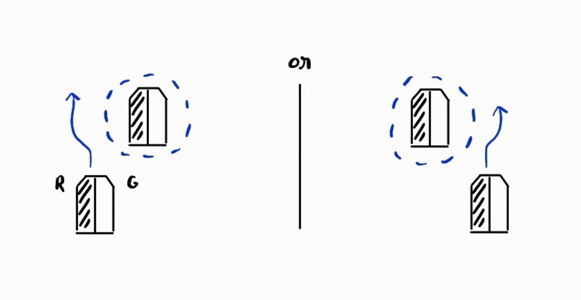


Figure 1 – Illustration of Rule 13 - *Overtaking* - the dotted circle represent who has the priority and G and R mean *Green* for port side and *Red* for starboard

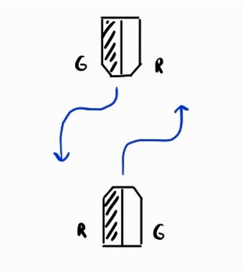


Figure 2 – Illustration of Rule 14 - *Head-on*

3.2 Artificial potential field controller

3.2.1 Command for a potential field controller

To simulate correctly the USV of the simulator, the following state equation has been chosen:

$$\begin{cases} \dot{x} = v \cdot \cos(\theta) \\ \dot{y} = v \cdot \sin(\theta) \\ \dot{v} = u_1 \\ \dot{\theta} = u_2 \end{cases} \tag{1}$$

Then, the considered USV will be called $\mathbf{p}$, with $\mathbf{p} = \begin{bmatrix} x \\ y \end{bmatrix}$, and $\hat{\mathbf{p}}$ the targeted position. And considered obstacle $\mathbf{q}$, the its targeted position $\hat{\mathbf{p}}$. In this study, the mobile robots, or the USV are navigating through a crowded environment, which includes both mobile and stationary obstacles such as other boats (manned and unmanned) or other environmental obstacles such as rocks. An interesting approach to find an adapted controller could be the artificial potential field method Jaulin [2023]. In this method, the USV is conceptualised as an electric particle capable of being either attracted or repelled by other objects based on their "electric charge." This approach is reactive, meaning that the USV's path is not predetermined but rather dynamically influenced by the surrounding environment. Therefore, this approach fits quite well to the context and environment of the study.

In physics, we have the following relation :

$$\mathbf{f} = -grad(V(\mathbf{p})) = \mathbf{w}(\mathbf{p}, t) = -\left(\frac{\partial V}{\partial \mathbf{p}}(\mathbf{p})\right)^T \tag{2}$$

where $\mathbf{p}$ is the position of point particle in space, $\mathbf{V}$ is the potential and $\mathbf{f}$ the force applied on the particle. The potential fields serve to articulate the intended behaviour for the USV, with obstacles represented by potentials creating a repulsive force on the robot, while the goal creates an attractive force. Here is an expression of $\mathbf{V}$ which include the speed of the USV, represented in the first term, the attraction field in the second term, and the repulsion, in the last term :

$$V(\mathbf{p}) = -\mathbf{v}^T \cdot \mathbf{p} + ||\mathbf{p} - \hat{\mathbf{p}}||^2 + \frac{1}{||\mathbf{p} - \hat{\mathbf{q}}||} \tag{3}$$

where $\mathbf{v}$ is the speed of the USV.

Then, by injecting **(3)**, in **(2)**, this new expression of **f** will be obtained :

$$\boxed{\mathbf{f} = +\hat{\mathbf{v}} - 2 \cdot (\mathbf{p} - \hat{\mathbf{p}}) + \frac{\mathbf{p} - \hat{\mathbf{q}}}{\|\mathbf{p} - \hat{\mathbf{q}}\|^3}} \tag{4}$$

To complete the controller, the commands of the vector **u** needs to be calculated :

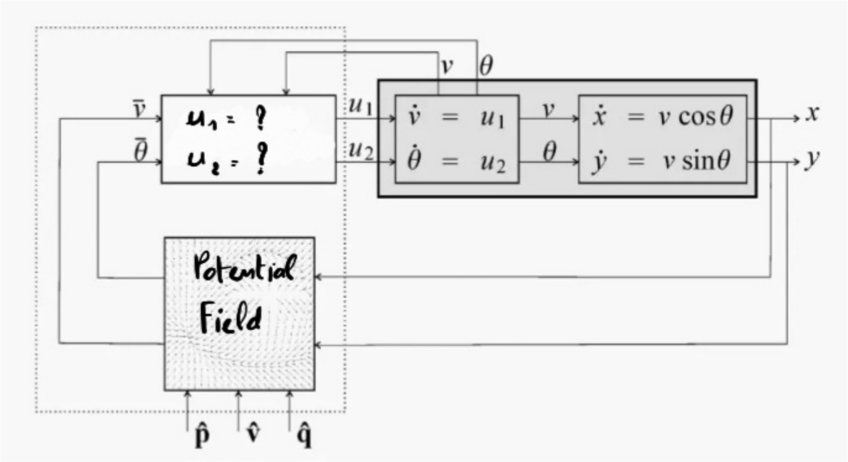


Figure 3 – Illustration of the structure of the chosen controller with a potential field

In the context of the study, $\mathbf{x_3}$ is the current heading of the USV, and ψ is the direction that the USV needs to follow, therefore :

$$\begin{cases} y = x_3 - \operatorname{atan2}(\psi_2, \psi_1) \\ \dot{y} = \dot{x_3} + \frac{\psi_1 \cdot \dot{\psi}_2 - \psi_2 \cdot \dot{\psi}_1}{\psi_1^2 + \psi_2^2} == u + \frac{\psi_2 \cdot \dot{\psi}_1 - \psi_1 \cdot \dot{\psi}_2}{\psi_1^2 + \psi_2^2} \end{cases} \tag{5}$$

After a couple of calculations by using the feedback linearisation method, this final expression (6) of the command u is obtained :

$$\boxed{\begin{cases} u_1 = 0 \\ u_2 = u + \frac{\psi_2 \cdot \dot{\psi}_1 - \psi_1 \cdot \dot{\psi}_2}{\psi_1^2 + \psi_2^2} \end{cases}} \tag{6}$$

Those expressions can now be implemented in the simulator of the study in a function called *control*, knowing that ψ_1 and ψ_2 will take **p** as a parameter. All the intermediate expressions are written in the *sea_object.py* Python file.

3.2.2 Equations of the potential fields

The solution chosen to build an avoiding collision system in this study, is to use **artificial potential field**. To do so, the vector field should have the shape of a circle, where the obstacle will be defined as the centre of this safety circle to avoid any collisions. Three different situations can be identified, and will therefore need an adapted potential field each. The potential field applied to the USV will depend of two parameters : the position of the USV involved in a potential collision, and also their orientation with each other. The evaluation of their orientation will help to determine if it is a situation of overtaking, or Red to Red (rule 13 or 14 according to the COLREG Cockcroft and Lameijer [2012]. Those fields will be applied depending on certain conditions developed in the section 4.

With p being the considered USV and its state vector $\begin{bmatrix} x_p \\ y_p \\ v_p \\ \theta_p \end{bmatrix}$, and q the obstacle USV, a case disjunction can be established :

Let's set the matrix $D=\begin{bmatrix} R & 0 \\ 0 & R \end{bmatrix}$ with **R** the radius of the circle from the vector field.
Then, the expression of a vector field converging counterclockwise to a circle of radius 1 is given by $\phi_0(p) = \begin{bmatrix} -x_p^3 - x_p \cdot y_p^2 + x_p - y_p \\ -y_p^3 - y_p \cdot x_p^2 + x_p - y_p \end{bmatrix}$ and $c = \begin{bmatrix} c_x \\ c_y \end{bmatrix}$ the coordinates of the center of the circle of the potential field. And finally, ϵ is a quantity created so that the last step of the avoiding collision manoeuvre won't start too soon. It will give more time to the USV to safely avoid the collision, in a smoother way, closer to what can be observed in reality

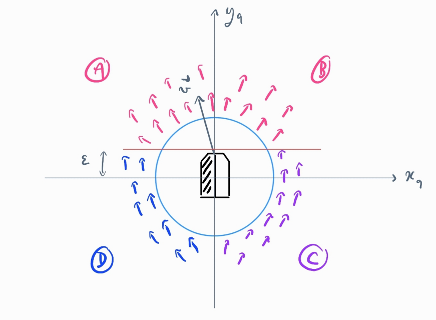


Figure 4 – Detailed breakdown of the various vector fields applied in the avoidance of a collision when USV have similar headings

— If $\mathbf{v_q} \cdot \mathbf{v_p} > 0$ (geometrical scalar product calculated with the function *geo_scalar_prod* defined in the *calcul_tools.py* file): The two USVs are approximately heading towards the same direction, and this is a situation of overtaking (rule 13).

— **Zone D** : $\mathbf{p} \in \begin{bmatrix}]-\infty; x_q \\ ]-\infty; y_q] \end{bmatrix}$
The USV is approaching the position of the obstacle on its left side. The shorter way to avoid the obstacle is to go clockwise, so **e** is set to -1. Therefore, $g(\mathbf{p}) = D \cdot \begin{bmatrix} 1 & 0 \\ 0 & e \end{bmatrix} \cdot \mathbf{p} + \mathbf{c}$.
So the final equation (7) for the clockwise vector field will be

$$\boxed{\phi_{cw}(p) = \mathbf{g} \circ \phi_\mathbf{0}(\mathbf{p}) = (\frac{dg}{dp} \circ g^{-1})(p) \cdot (\phi_\mathbf{0} \circ g^{-1})(p) = \begin{bmatrix} R & 0 \\ 0 & -R \end{bmatrix} \cdot \phi_\mathbf{0}(\begin{bmatrix} R & 0 \\ 0 & -R \end{bmatrix}^{-1} (p - c))} \tag{7}$$

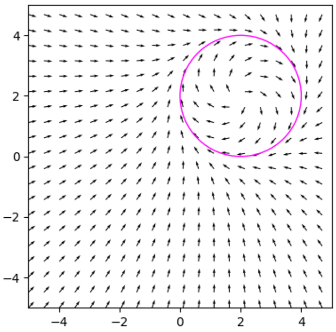


Figure 5 – Illustration of the clockwise artificial vector field applied

- **Zone C** : $\mathbf{p} \in \begin{bmatrix}]x_q; +\infty[\\ ]-\infty; y_q] \end{bmatrix}$
  The USV is approaching the position of the obstacle on its right side. The shorter way to avoid the obstacle is to go counterclockwise, so **e** is set to 1. So the final equation (8) for the clockwise vector field will be:

$$\boxed{\phi_{ccw}(p) = \mathbf{D} \cdot \phi_{\mathbf{0}}(\mathbf{D^{-1}} \cdot (\mathbf{p} - \mathbf{c}))} \tag{8}$$

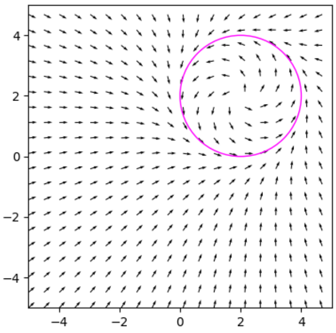


Figure 6 – Illustration of the counterclockwise artificial vector field applied

- **Zone B** : $\mathbf{p} \in \begin{bmatrix}]x_q; +\infty[\\ ]y_q + \epsilon; +\infty[\end{bmatrix}$
  The USV is finishing overtaking the obstacle from the right side. In order to end the avoid collision manoeuvre, a repulsive field will by applied to the considered USV, around the same circle of radius **R** and center **c**. Its point of repulsion will be the center c. So the final equation (10) for the repulsive vector field will be:

$$\boxed{\phi_{rep}(p) = \begin{bmatrix} k \cdot (x_p - c_x) \\ k \cdot (y_p - c_y) \end{bmatrix}} \tag{9}$$

  In the simulation, after several tests, **k** has been fixed at **0.5**.
- **Zone A** : $\mathbf{p} \in \begin{bmatrix}]-\infty; x_q] \\ ]y_q + \epsilon; +\infty[\end{bmatrix}$
  The USV is finishing overtaking the obstacle from the left side. So the same repulsive field will by applied to the considered USV, around the same circle of radius **R** and center **c**. So the final equation (10) for the repulsive vector field will be

$$\boxed{\phi_{rep}(p) = \begin{bmatrix} k \cdot (x_p - c_x) \\ k \cdot (y_p - c_y) \end{bmatrix}} \tag{10}$$

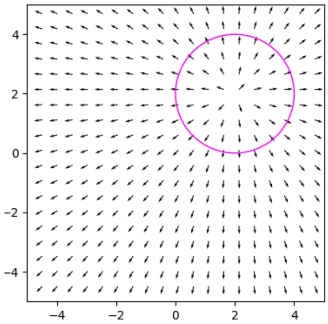


Figure 7 – Illustration of the repulsive artificial vector field applied

- If $\mathbf{v_q} \cdot \mathbf{v_p} > 0$: The two USVs are going in opposite direction. This can lead to a situation of R to R (rule 14).

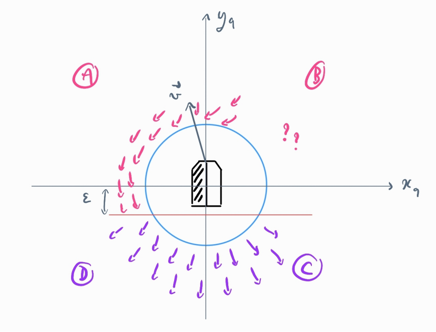


Figure 8 – Detailed breakdown of the various vector fields applied in the avoidance of a collision when USV have opposite headings

- **Zone A** : $\mathbf{p} \in \begin{bmatrix}]-\infty; x_q] \\ ]y_q - \epsilon; +\infty[\end{bmatrix}$
  The Red to Red rule needs to be applied. Therefore, a counterclockwise field $\phi_{\mathbf{ccw}}$ is chosen for this zone.

- **Zone B** : $\mathbf{p} \in \begin{bmatrix}]x_q; +\infty] \\ ]y_q - \epsilon; +\infty[\end{bmatrix}$
  This zone can have two different outcomes depending on the heading of the two USV. If the two USV have similar headings (calculate through a scalar product) and are not too close to each other, then there is no risk of collision, so no vector field will be applied. But if their headings are too similar, then the Red to Red rule (rule 14) must be applied, by using a counterclockwise field $\phi_{\mathbf{ccw}}$.

- **Zone C and D** : $\mathbf{p} \in \begin{bmatrix}]-\infty; +\infty[\\ ]y_p - \epsilon; -\infty[\end{bmatrix}$
  Final step of the avoiding collision manoeuvre, a repulsive field $\phi_{\mathbf{rep}}$ will be applied.

These are the different artificial vector fields which are going to be used in the simulator depending on the situation simulated. Nevertheless, the actual behaviour of the USV implemented is slightly different. Different special cases will be developed in the following part 4 in order to obtain a more realistic simulation.

4 Avoiding collision system

4.1 Overall functioning

The main function of the simulator, is *avoid_collision* in the *sea_object.py*. This function will be called when there is a risk of collision, and will start an avoiding collision process. The condition to start the process is if the distance between two USV is inferior to a quantity named **r**. This quantity represent the DCPA Yaseen Adnan Ahmed [2021], or Distance at Closest Point of Approach. It's a crucial concept defined by the International Regulations for Preventing Collisions at Sea (COLREGs) to ensure the safe navigation of vessels. It represents the minimum separation between two vessels on a collision course, indicating the closest distance they will approach each other. It is a fundamental parameter used in the determination of potential collision situations. And as it helps mariners to assess the risk of collision and to take appropriate actions to maintain a safe separation, this quantity is essential to be implemented in the simulator. It is represented by a red circle around every USV. A second magenta circle has been created. Once a USV will enter into this circle, the anti-collision system will be activated. And the zone between the red and magenta circle is the **manoeuvring zone** where the boat will proceed to the avoidance following the COLREGs.

Depending on the position of the USV in the obstacle repository, one of the artificial potential vector field described in 3.2.2 will be applied. Then, once the obstacle is avoided, the USV will go back to its initial consign ruled by the *equation of control* (6) to follow the path leading to its final destination. With this new function, situations with several collision scenarios can be run. Here are a couple of examples of simulations tested and the result of the corrected path established by the avoiding collision system.

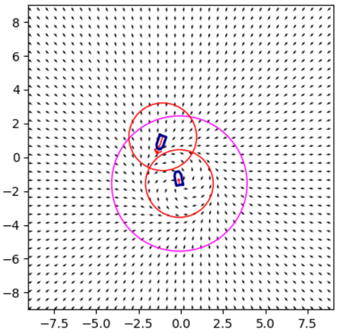


Figure 9 – Illustration of USV with their DCPA in red and following an artificial vector field

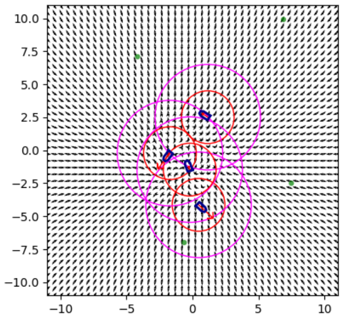


Figure 10 – Illustration of four USV avoiding collisions with each other

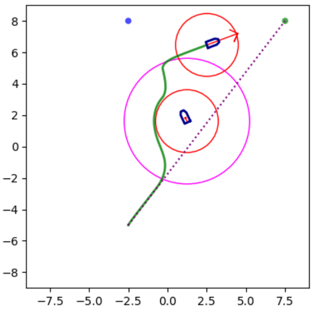
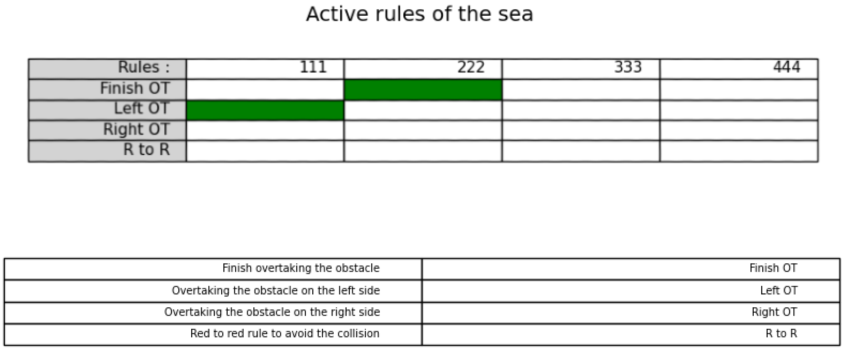


Figure 11 – Illustration of two USV avoiding a collision through a corrected path

The little green or blue dots represent the final destination of each USV. Those three illustrations have been extracted from the results of some tests run with the simulator.

In the developed simulator, a real-time monitoring system was implemented to track the application of International Regulations for Preventing Collisions at Sea (COLREGs) during collision avoidance scenarios. Matplotlib was employed to create a dynamic table illustrating the rules in use at any given moment. This table offers an immediate snapshot of the specific COLREG rules being applied as vessels navigate through potential collision scenarios by coloring in green the corresponding cells. The USV are identified by their MMSI number, more detailed in the paragraph 5.1. The real-time feedback enhances situational awareness and facilitates an assessment of the effectiveness of the implemented collision avoidance strategies to facilitate potential future works develop in the 7.4 paragraph.

It is implemented directly in the *avoid_collision* function after being initialised through the function *init_table* defined in *draw.py* and called with all the initialisation in the *simulation_runner.py* file.



Active rules of the sea

Rules :	111	222	333	444
Finish OT		🟩		
Left OT	🟩			
Right OT				
R to R				

Finish overtaking the obstacle	Finish OT
Overtaking the obstacle on the left side	Left OT
Overtaking the obstacle on the right side	Right OT
Red to red rule to avoid the collision	R to R

Figure 12 – The rules grid is displayed alongside the main simulation to visualise which COLREGs'rules are applied

4.2 Cross-path situation

After running couple of simulations, specific scenarios tested don't look very realistic. Indeed, in some scenarios, a USV can enter into the manoeuvring zone, but its orientation and speed won't lead to an actual collision with the second USV. In reality, an experienced captain won't start a full process of avoiding collision if it's not necessary and if there is no danger. Therefore, the simulator can be improved by implementing a special process for the **cross path situation**.

To do so, a new instance variable has been created in the SeaObject class. It is a boolean called *cross_path*, set at **False**. The cross path situation can only be encountered when a USV is trying to overtake another one. More specifically, if one of these two conditions is verified, the boolean **cross_path** will be set to **True** :

— **USV in the left lower zone heading to the right lower zone :** $(y_p < y_q + \epsilon)$ and $(x_p < x_q)$ and $(\hat{p_y} < y_q + \epsilon)$ and $(\hat{p_x} > x_q)$
Therefore a counterclockwise artificial vector ϕ_{ccw} field will be applied instead of a clockwise as there is no risk of collision.

— **USV in the right lower zone heading to the left lower zone :** $(y_p < y_q + \epsilon)$ and $(x_p > x_q)$ and $(\hat{p_y} < y_q + \epsilon)$ and $(\hat{p_x} < x_q)$
And a clockwise artificial potential field ϕ_{cw} will be applied instead of a counterclockwise.

5 The Simulator A

Now, zooming out to the broader scope of the entire simulator, this crafted system seamlessly integrates various functions, including *avoiding_collision*, to emulate realistic maritime scenarios. The simulator serves as a comprehensive platform, not only incorporating complex collision avoidance strategies but also providing a holistic environment for testing and refining autonomous navigation algorithms in diverse maritime contexts. This first Simulator A can be run in two different ways :

— The simulation with the scene with the different USV displayed

— The simulation without any display, but with the USV's following information : MMSI number, x, y, theta, v saved in a .csv file

5.1 The different types of agents

The aim of this simulator is to evaluate an avoidance collision system while adhering to the COLREGs rules. Therefore, the agents simulated in the environment should emulate various types of boats commonly encountered in maritime scenarios.
In the context of maritime agents, an essential parameter must be integrated into our study: the concept of privilege. This parameter is introduced to establish a hierarchy among the diverse maritime entities. When two vessels approach each other dangerously at sea, the ease of executing an avoidance manoeuvre depends on their size and characteristics. For instance, it is inherently logical for a small pleasure craft to yield to a large cruise ship. Therefore, the degree of privilege is formulated such that higher degrees signify greater priority, hence the need for an avoidance manoeuvre.
To streamline the analysis process, we categorised the diverse range of agents into four distinct categories:

- **Boat**: Representing a pleasure boat with a hull length less than or equal to 24 metres. Its privilege degree is set to **0** because it is the most mobile agent.
- **Ship**: Representing an ocean liner (couple of hundreds metres), to simulate bigger agents, slower with less manoeuvrability. Therefore, its privilege degree is set to **30**.
- **Whale**: Representing all the agents for which no information are known, so with a complete uncertainty about their future trajectory. It can represent marine animal, jet-skis, or any another agent too small to be registered or detected. Due to its unknown behaviour, its privilege degree will be set to **500**.
- **Island**: Representing any sort of non-mobile agent for which it is impossible to move and avoid a collision. It could be a natural obstacle part of the environment like an island, a rock, a reef, or an anchored agent. Therefore, its privilege degree is set to the maximum value **1000**.

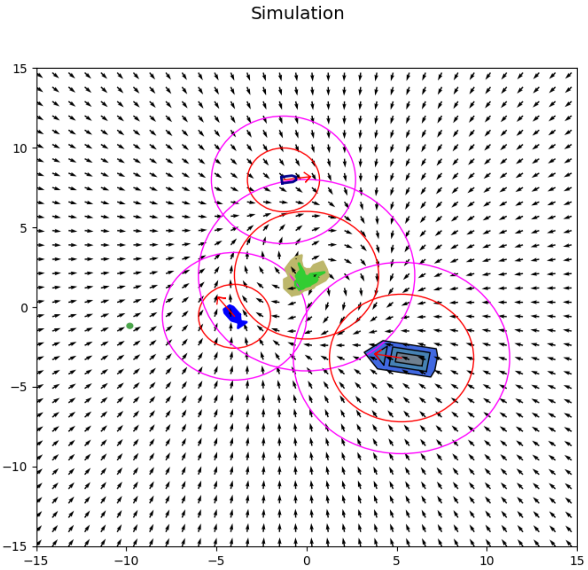


Figure 13 – Illustration of a boat (on top), island (in the middle), whale (bottom left), and a ship (bottom right)

5.2 The structure

This simulator is accessible through a GitHub Marie Dubromel [2023] and can be found in the *main_program* file.

This directory contains the most up-to-date and comprehensive version of the USV simulator. It introduces a more sophisticated structure for the simulation of sea objects. The main components are:

- **SimulationRunner**: This class is responsible for initialising all objects and constants for the simulation, such as *s, dt, k, num_steps*. Those parameters allow the user to change the speed of the simulation (dt), its duration (num_step), the size of the display (s), or the intensity of the artificial vector field used (k). It is the file where the simulator user will run simulations, as explained in the user guide in paragraph 5.3.
  A function called *initialize_sea_objects()* creates various sea objects (*Boat, Ship, Whale, Island*, etc.). These objects are initialised with parameters such as x, y, v, and θ, and then added to the *sea_objects* vector.
  A separate function *initialize_data* is also created to display the rules of collision avoidance at sea (COLREGs). This function returns a list of the different COLREGs rules and a list of the MMSI numbers of the defined sea objects.
  The *run()* function then takes this *sea_objects* vector and the list of rules and MMSI numbers, passing them to the *Simulation* class.

- **Simulation**: This class is responsible for running the simulation. It accepts the *sea_objects* vector and runs the simulation. It contains two functions *run()* and *run_with_data()*. The user will be able to chose between displaying the simulation with the first one, or saving the data in a .csv file. During each iteration, it calls the *move()* methods for each object in the *sea_objects* vector, and the and *draw()* method if the user chose to run the simulation with the display.

- **SeaObject**: This is the parent class for all sea objects detailed in 5.1 (Boat, Ship, Whale, Island, etc.). Each time the *move()* method is called for a *SeaObject*, the *in_collision* variable is set to **False**, indicating that the object is currently not in collision with another.
  - The *move()* method then loops over every other object in the *sea_objects* vector and calculates the distance between the current object and every other object. If the distance is smaller than the maximum radius of the two objects (this is to account for the fact that different objects might have different collision radius for safety reasons), the privilege of the two objects is compared 5.1. If the current object has a lower privilege, it needs to avoid collision. In this case, the *avoid_collision()* function is called and *in_collision* is set to **True**. If the current object has a higher privilege, it does not need to do anything and ignores the potential collision. If the distance is larger than the collision radius, the current object also does not need to do anything. If the degree of privilege is the same between two USV, one of the two USV, arbitrarily chosen will start to avoid the collision.
  - The *move_straight()* function is called when the *SeaObject* either is not in collision with any other objects, or is in collision but has higher privilege. This function instructs the object to continue on its initial path.

  Whether an object moves straight or avoids collision, both actions return a control vector u_p, which is then passed to the *update()* function to update the object's position. The *draw()* function is then called by *Simulation* to draw the new position of the object. This process repeats in each iteration of the simulation.
  In this manner, each sea object is responsible for its own actions, deciding whether to avoid collision or not, based on the rules defined in its methods. The object does not care about the reactions of others, ensuring each object makes decisions autonomously.

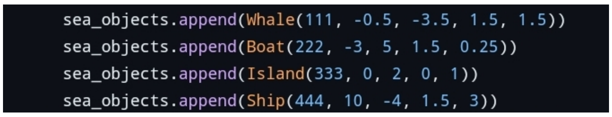
5.3 The user manual : the initialisation of a simulation

To conduct tests using this simulator, users only need to modify the *simulation_runner.py* file.

- **Constants :** The simulation's constants can be initialised as needed. Below is an example of the selected settings for these constants: *s=15, dt = 0.1, k = 0.5,* $\epsilon = 2$*, and num_steps = 1000.*

- **Type of simulation :** If the user wishes to save the data from the simulation, the class parameter *record_data* needs to be set to **True**. Consequently, the MMSI number, x, y, theta, and v will be saved in a .csv file, and the simulation won't be displayed. Conversely, if *record_data* is set to **False**, the user will be able to observe the simulation through a Matplotlib window. Therefore, if the simulation is run twice with the same parameters, except for *record_data*, the user will be able to visualise the scene and analyse its data through the .csv file.

- **Creation of USV :** To create a USV, the user needs to add the new agent in the *initialize_sea_objects* function and set its parameters in the following order: MMSI number x, y, v, and *theta*. Several examples are provided in the comments at the end of the Python file.



```
sea_objects.append(Whale(111, -0.5, -3.5, 1.5, 1.5))
sea_objects.append(Boat(222, -3, 5, 1.5, 0.25))
sea_objects.append(Island(333, 0, 2, 0, 1))
sea_objects.append(Ship(444, 10, -4, 1.5, 3))
```

Figure 14 – Initialisation of different USV

- **Further modifications** If the user wishes to improve or changes some characteristics of the different USV, it can be done in the file of each USV like *boat.py* or *ship.py*...

6 The Simulator B : AIS data

The second simulator introduced offers the capability to recreate real-life scenarios using AIS data from various registered boats. The datasets are sourced from the AISHub website ais, providing an authentic and dynamic backdrop for simulation scenarios. With this simulator, users can immerse themselves in maritime environments, and understand the actual vessel movements and interactions. Therefore, it would be a good base for the final simulator which will mix real USV from AIS data, and fictive USV added by the user of the simulator.

6.1 The AIS data extraction

The Python file used to extract the data is *dataExtractor.py* The code demonstrates how to extract and manipulate AIS (Automatic Identification System) data using SQLite database with Python to store the data. Here's a breakdown of the main functionalities:

- **Adding Data to Database** : The *add_data_to_db* function reads data from CSV files in chunks and appends it to the specified table in the SQLite database.
- **Querying Static Data** : The *query_data_static* function retrieves data from the specified table in the database based on a given MMSI (Maritime Mobile Service Identity) number. It calculates the center points of latitude and longitude, centralises coordinates, converts headings to radians, computes average speed, and scales speeds accordingly. This querying process allows to analyse and extract useful information from the dataset. Therefore, it enables to visualise the data in a meaningful way and communicate the insights effectively.
- **Querying Dynamic Data** : The *query_data_dynamic* function performs similar operations to the static data query but for dynamic AIS data. It also includes additional processing steps such as dividing speed values by 10.
- **Deleting Table** : The *delete_table* function deletes the specified table from the database.
- **Main Functionality** : The main function connects to the SQLite database, adds data to it, queries dynamic data for a specific MMSI number, and closes the connection afterwards.

To run this simulation for the first time, the user needs to uncomment the *add_data_to_db* function with the chosen .csv file, and the *query_data_static* and *query_data_dynamic*. If the same simulation has to be rerun, those function can be commented. And to use a different set of data, the user can simply uncomment the **delete_table** function to reset the SQL strorage.

After doing this process, tables will be created. The SQL table created from the AIS data would include several columns corresponding to different attributes of the AIS data. Here's an estimation of the columns based on the operations performed in the code:

For static data:

1. **sourcemmsi** : MMSI identifier of the maritime entity
2. **LAT**: Latitude of the maritime entity
3. **LON**: Longitude of the maritime entity
4. **SPEED**: Speed of the maritime entity
5. **HEADING**: Heading of the maritime entity

   Other columns may be present depending on the original AIS data, such as vessel name, vessel type, etc.

For dynamic data:

1. **sourcemmsi**: MMSI identifier of the maritime entity
2. **LAT**: Latitude of the maritime entity
3. **LON**: Longitude of the maritime entity
4. **SPEEDOVERGROUND**: Speed of the maritime entity over the ground
5. **TRUEHEADING**: True heading of the maritime entity

The exact structure of the table would depend on the specific input data and the application's requirements, but it would generally include these basic attributes to represent AIS information.

6.2 AIS data processing

After extracting the AIS data from a .csv file, it's going to be processed in order to create a USV and simulate its trajectory in the simulator B. This work is done the the *learningShip.py* Python file where a class ‘LearningShip‘ is defined, to represent an USV in a simulation. Here's how it works:

- **Initialization:** The *__init__* method initialises the USV's attributes such as its position **x and y**, velocity **v**, orientation (‘theta‘), and a **DataFrame df** containing data about the USV's movements. It also initialises an index variable to keep track of the ship's current position in the DataFrame and sets a privilege value.
- **Move Method:** The *move* method is responsible for updating the ship's position based on the data in the DataFrame. It checks if the index is within the bounds of the DataFrame, retrieves the ship's speed and heading from the DataFrame at the current index, and then updates the ship's position accordingly using basic kinematic equations.

  $$\begin{cases} x+ = dt \cdot v \cdot \cos(\theta) \\ y+ = dt \cdot v \cdot \sin(\theta) \end{cases} \tag{11}$$

  The method also increments the index to move to the next row of data in the DataFrame.
- **Get State Vector Method:** The *get_state_vector* method returns a state vector representing the ship's current state, including its position, velocity, and orientation.
- **Draw Method:** The *draw* method is responsible for visualising the ship and its surroundings. It calls a function to draw the ship itself (*draw_boat_and_vector*) and draws two circles around the ship representing different zones: the **DCPA** (closest point of approach) zone to avoid collisions (r), and an extended DCPA zone for safety ($r + \epsilon$). These circles are drawn on the provided Matplotlib axis (*ax*).

Therefore, this class encapsulates the behaviour of an USV from a set of AIS data in the simulation, including its movement, visualisation, and representation of its state.

7 Further updates

At the conclusion of this internship, two simulators (A and B) were developed and are now accessible on GitHub Marie Dubromel [2023]. However, there remains ample room for enhancement and refinement, considering the significance of addressing pertinent issues in a world where autonomous vessels are rapidly gaining traction. This simulator holds promise in resolving security and safety concerns by devising strategies to prevent collisions among diverse types of USVs prevalent in contemporary maritime settings. Looking ahead, numerous ideas for improvement have been identified, laying the groundwork for future projects.

7.1 The merger of Simulator A and Simulator B

A first pretty obvious update would be to try to merge the two simulators A and B. Indeed, a merged simulator will give the possibility to the user to run a simulation where old existing USV and their scenarios would be simulated from AIS data, as well as fictive USV created by the user. Then two solutions can be implemented. One would be to implement the avoiding collision system of the simulator A only in the fictive USV. This will lead to have a simulator where the added USV are completely integrated to the maritime traffic without impacting it and the AIS data USV won't have their trajectory modified. The other solution would be to implement this avoiding collision system to all the USV. As a result, the AIS data USV will have to adapt and modify their former trajectory to potentially avoid a collision with a fictive USV.
In practice, to simplify the implementation of the two possible solutions, and also a simple way to switch between those two, the level of privilege can be used. Indeed, if the user don't want the AIS data USV to have to change their initial trajectory, their level of privilege can be set as the highest one among all the other implemented type of USV (at 2000 for instance). Therefore, the AIS data USV will always have the priority and their trajectory will correspond to the AIS at all time.

7.2 Interpolation on the AIS data

Concerning the AIS data simulator, an important future enhancement for our simulator would involve the implementation of an **interpolation function** to improve the accuracy and consistency of AIS data. Currently, technical issues on vessels or errors in data transmission during the recording of the AIS data can result in gaps or missing data points in our AIS data files. To address this, a possible solution would be to create individual .csv files for each vessel based on their unique MMSI numbers and applying interpolation techniques to estimate the missing values. Methods such as linear interpolation, spline interpolation, or nearest neighbour interpolation can be utilised depending on the data characteristics and desired accuracy level. By filling in these gaps, we can ensure a continuous and smooth representation of vessel trajectories. This process has already begun, and once completed, the interpolated AIS data files will be seamlessly integrated into the simulator B, enhancing the accuracy and realism of the simulations by providing more precise and consistent vessel movements.

7.3 Integration of Maritime Chart Sources

Another enhancement for our simulator would involve integrating a maritime chart into the simulation background, corresponding to the location of the simulated scenario. However, the users would have to choose and import their own maritime map into the simulator directory as it would be a very complex problem to implement a solution uploading the maps itself on the internet depending on the coordinates present in the .cvs files. Those maritime chart could be found on websites like ope *Open CPN*.

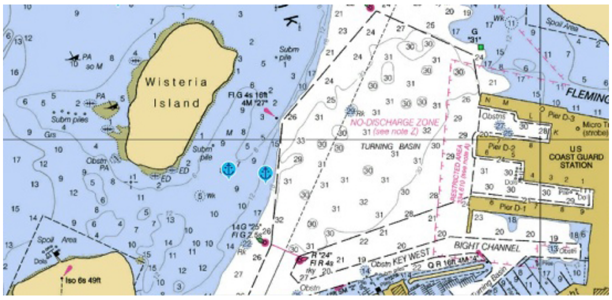


Figure 15 – Maritime Chart Sample

To determine the dimensions of the map to be imported, one approach would be to analyse the x and y coordinates of the USVs in the AIS data file and extract their minimum and maximum values. These values would then define the dimensions of the map and the scale chosen, ensuring that the USVs are accurately positioned on the map.

7.4 AI

In certain scenarios, sailors may encounter mixed-motive situations where adhering to COLREGs conflicts with other objectives or motivations, such as minimising time to reach their destination or reducing fuel consumption. Understanding these conflicting motivations and how they impact the decision-making of human navigators can aid in designing more effective training programs. This is why integrating an AI to the simulator could be an interesting enhancement. AI could be one of the last step leading to the final product for the simulator. Our ultimate objective for the simulator is to validate or not realistic scenarios where the collision avoidance procedures implemented by our system could feasibly occur in real-world situations. To achieve this, experienced sailors could provide training for the implemented AI, assessing the plausibility of the simulations. Initially, a labelling and reviewing technique could be employed to train the AI.

In the end, before displaying the solution found by the anti-collision system, the AI will analyse the solution and send a message to the user to communicate about the plausibility of the solution found by the simulator.

7.5 Uncertainties

An interesting aspect which hasn't been exploited yet, would be to add uncertainties on the trajectories of the boats. To do so, the notion of intervals could be use, more specifically in the case of the study, **tubes**. Simon Rohou Rohou, a French researcher is writing a Codac library to help implementing intervals through boxes and tube in robotics.

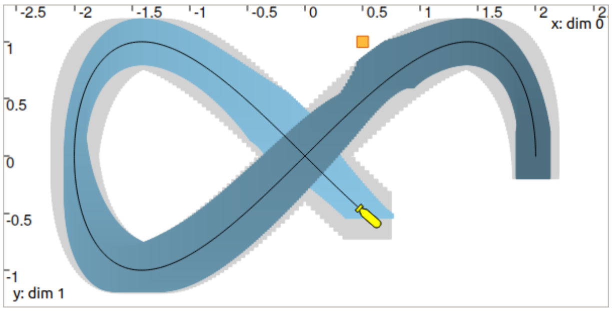


Figure 16 – Exemple of a contracted tube aplied on the trajectory of a yellow robot following a Lissajou curve

8 Conclusion

To conclude, the development of two simulators marks progress in the pursuit of building an anti-collision system that adheres to COLREGS regulations for manned, unmanned, and hybrid vessels. Simulator A, utilising artificial potential fields, offers a comprehensive approach by identifying various scenarios, calculating corresponding vector fields, and implementing adapted controllers. The inclusion of a real-time monitoring system ensures adherence to COLREGs during simulated collision avoidance scenarios, facilitating analysis by users and experienced sailors alike. This will lead to a better understanding of humans motivations and concerns and adapt the COLREGs rules to autonomous vessels. Nevertheless, the option to select from four distinct types of agents representing different categories of USVs (from small boats, to big commercial vessels or any object with an unpredictable behaviour), offers versatility and insight into collision avoidance strategies. Each one of the different types of USV has a level of privilege, in order to know which agent will have to start an avoiding-collision procedure first. Simulator B, on the other hand, leverages real AIS data from past USV trips, laying the foundation for a unified simulator capable of blending fictive and real-world USVs seamlessly. In the broader context, the convergence of these simulators promises to address the evolving landscape of maritime traffic, particularly with the emergence of autonomous and hybrid USVs. By ensuring compliance with COLREGs in novel operational scenarios, these advancements limit risks to human life while unlocking new possibilities for ambitious missions. Looking ahead, the continued refinement and integration of AI-driven enhancements will be pivotal in fostering safe and efficient collaboration between manned and unmanned vessels, ultimately ushering in a new era of maritime exploration and innovation. This could also be helped by the collaboration of private companies in order to compare the different work and products created, as this research focuses on finding a potential solution of an anti-collision system integrating the COLREGs rule to ensure safety on the sea.

9 Acknowledgements

This internship was conducted at the IRL Crossing Laboratory in Adelaide, Australia, under the guidance of Jean-Philippe Diguet and Anna Ma-Wyatt. I extend my sincere gratitude to my supervisor, Benoit Clement, for his invaluable support throughout my internship. He not only provided guidance and advice on my projects but also offered valuable insights into career development and helped me navigate life in Australia. I am also grateful to Jean-Philippe Diguet for welcoming me to the lab and providing me with the opportunity to work on such an engaging project. Additionally, I am indebted to Cédric Buche for his mentorship and who took time with me to discuss about scientific research, which greatly contributed to my understanding of the field. Lastly, I would like to express my appreciation to Peter Wu for his collaboration on optimising the algorithmic structure of the simulator, enabling it to efficiently handle a large number of USVs and AIS data.

References

Aishub aisdata exchange. URL `https://www.aishub.net/api`.

Irl crossing. URL `https://crossing.cnrs.fr/`.

Open cpn- carte sources. URL `https://opencpn.org/OpenCPN/info/chartsource.html`.

Paulo E. Santos Karl Sammut Michelle Oppert Feras Dayoub Benoit Clement, Marie Dubromel. Hybrid navigation acceptability and safety. Technical report, 2024. URL `https://ojs.aaai.org/index.php/AAAI-SS/article/view/27643`.

A.N. Cockcroft and J.N.F. Lameijer. *A Guide to the Collision Avoidance Rules (Seventh Edition)*. Butterworth-Heinemann, Oxford, 2012.

COLREGs. *Convention on the International Regulations for Preventing Collisions at Sea (COLREGs)*. International Maritime Organization (IMO) London, UK, 1972.

E. Crase, C. Wideman, M. Noble, and A. Tarantola. Utseasim documentation. page 10, 2013. URL `https://www.cs.utexas.edu/~UTSeaSim/download/1.0/Oct2013Documentation.pdf`.

Thor I. Fossen. Python vehicule simulator. 2013. URL `https://www.cs.utexas.edu/~UTSeaSim/`.

Thor I. Fossen. Python vehicule simulator. 2021. URL `https://www.fossen.biz/pythonVehicleSim/`.

Luc Jaulin. Mobile robotics: Guidance. Technical report, ENSTA Bretagne, 2023. URL `https://www.ensta-bretagne.fr/jaulin/guidage.html`.

Peter Marie Dubromel. Mobile robotics: Guidance. Technical report, 2023. URL `https://github.com/wuzikang961215/Simulator`.

Simon Rohou. Codac tutorial. URL `https://codac.io/tutorial/05-tubes/index.html`.

Mahmoud Yasser Oraby Adi Maimun Yaseen Adnan Ahmed, Mohammed Abdul Hannan. Colregs compliant fuzzy-based collision avoidance system for multiple ship encounters. *Journal of Maritime Sciences and Engineering*, 2021. URL `https://www.mdpi.com/2077-1312/9/8/790`.

A Appendice

During this internship and under the direction of Benoit Clement, a scientific article has been written for the Vol. 2 No. 1 of Proceedings of the 2023 AAAI Fall Symposia. It is accessible through this website Benoit Clement [2024] or in the appendice.

Hybrid Navigation Acceptability and Safety

Benoit Clement[1,2,3], **Marie Dubromel**[1], **Paulo E. Santos**[1,3], **Karl Sammut**[1,3], **Michelle Oppert**[1,4], **Feras Dayoub**[1,5]
[1] CNRS IRL 2010 CROSSING, Adelaide, South Australia
[2] ENSTA Bretagne, Brest, France
[3] Flinders University, Adelaide, South Australia
[4] University of South Australia, Adelaide, South Australia
[5] University of Adelaide, Adelaide, South Australia
benoit.clement@ensta-bretagne.fr

Abstract

Autonomous vessels have emerged as a prominent and accepted solution, particularly in the naval defence sector. However, achieving full autonomy for marine vessels demands the development of robust and reliable control and guidance systems that can handle various encounters with manned and unmanned vessels while operating effectively under diverse weather and sea conditions. A significant challenge in this pursuit is ensuring the autonomous vessels' compliance with the International Regulations for Preventing Collisions at Sea (COLREGs). These regulations present a formidable hurdle for the human-level understanding by autonomous systems as they were originally designed from common navigation practices created since the mid-19th century. Their ambiguous language assumes experienced sailors' interpretation and execution, and therefore demands a high-level (cognitive) understanding of language and agent intentions. These capabilities surpass the current state-of-the-art in intelligent systems. This position paper highlights the critical requirements for a trustworthy control and guidance system, exploring the complexity of adapting COLREGs for safe vessel-on-vessel encounters considering autonomous maritime technology competing and/or cooperating with manned vessels.

Introduction

Autonomous vessels are rapidly gaining acceptance, particularly within the naval defence sector, as they offer an obvious means of removing human personnel from risks originating from conflict or environmental threats. A critical requirement of achieving full autonomy for a marine vessel is the development of a robust and trustworthy control and guidance system that accommodates different approach encounters (considering manned as well as unmanned vessels), while operating under a wide range of weather and sea state conditions. To safely accommodate vessel-on-vessel encounters, the autonomous vessel must comply with the International Regulations for Preventing Collisions at Sea (COLREGs) (COLREGs 1972; Cockcroft and Lameijer 2012). COLREGs evolved from a set of practises that were originally designed in the mid-19th century for human interpretation. Therefore, these rules are written in ambiguous prose, assuming that their interpretation and execution were carried out by highly experienced sailors and not by an autonomous system. Here is a list of the relevant COLREGs rules that are considered in our work:

- Rule 8 - *Actions to avoid collision*: if there is sufficient sea-room, alteration of course alone may be most effective. Reduce speed, stop or reverse only if necessary.
- Rule 13 - *Overtaking*: Any vessel overtaking any other shall keep out of the way of the vessel being overtaken.
- Rule 14 - *Head-on*: Each head-on vessel shall alter her course to starboard so that each shall pass on the port side of the other.
- Rule 15 - *Crossing*: The vessel which has the other on her own starboard side shall keep out of the way.
- Rule 16 - *Actions by give-way vessel*: Take early and substantial action to keep well clear.
- Rule 17 - *Actions by stand-on vessel*: Keep her course and speed but may take action to avoid collision if the other vessel is not taking a COLREGs-compliant action.

Moreover, while implementing COLREGs in an environment exclusively inhabited by unmanned vessels is relatively straightforward, its implementation becomes significantly more complex in an environment where unmanned vessels interact with manned ones. This complexity arises from the often unpredictable behavior of human navigators, who may occasionally deviate from the rules in their efforts to avoid potentially hazardous situations. In contrast, current COLREG-compliant methods strictly adhere to the rules, regardless of the navigator's intentions. Consequently, in a potential future scenario, fleets of autonomous vessels could be diverted from their intended course or even hijacked by malicious actors manipulating these machines' nearly explicit knowledge states. Developing systems capable of such advanced epistemic reasoning, particularly in mixed-motive situations, is one of the primary objectives of this research.

Classical model-based approaches to automated COLREG compliance have proven to be too complicated to accommodate all possible encounters, environment scenarios, and human behaviours (Statheros, Howells, and Maier 2008). Modern Machine Learning (ML) methods, such as

---

Copyright © 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Deep Reinforcement Learning (DRL), could provide a flexible and adaptable model-free guidance and obstacle avoidance system, whereby the multiple possible interactions and scenarios can be abstracted from previous observations (Burmeister and Constapel 2021). However, ML methods do not provide the semantics of rules, or possible ways of breaking them, given the situation perceived. This work proposes the development of an AI-based COLREG-compliant model-free collision avoidance system that will be trained using historical AIS-based simulations of real-world scenarios, whereas the possible human interpretations of the rules will take a centre point in the development. This will be accomplished according to the following four sub-modules:

- Module 1: Autonomous Situation Awareness (ASA) aims to classify obstacles, other vessels, and intentions, defining vessel-on-vessel encounters from the information provided by multiple sensors (e.g. AIS and radar) .
- Module 2: Readability of Human Rules (RHR) will take into account how to create algorithms capable of presenting a *human-like* understanding of the COLREG rules, which should take into account the multiple possible space-time histories consistent with the observations and inferences provided by the ASA subsystem.
- Module 3: Path Planning and Control (PPC) aims at the implementation of guidance and control algorithms to ensure acceptability and safety based on human understanding of the COLREGs provided by the RHR subsystem.
- Module 4: Human Acceptability (HA) is the task of measuring the acceptability of the behaviour of autonomous systems based on studies of human operators.

Module 1: Autonomous Situation Awareness

The Autonomous Situation Awareness (ASA) module in this work has the task of integrating data from multiple sensors, including radar, Automatic Identification System (AIS) and possibly cameras and Synthetic Aperture Radar (SAR) imagery. Using machine learning techniques, the ASA system's task is to accurately identify and track vessels, obstacles, and navigational hazards to provide a comprehensive interpretation of the situation in which the vessel is emerged.

Developing an ASA system is crucial to ensure the safe operation of Autonomous Surface Vessels (ASV) in open-water environments. By providing real-time situational awareness of the vessel's surroundings and predicting potential risks, the ASA system will significantly enhance the safety and reliability of autonomous navigation. This information will be used by the PPC system (module 3) to generate safe and COLREG-compliant trajectories and vessel behaviour.

The proposed ASA system should have the following features:

- Object detection and tracking: using a combination of sensors, a system will be implemented to detect and track maritime objects, including other vessels, buoys, and obstacles. This information will be used to adjust the vessel's course and speed to avoid collisions.
- Real-time situational awareness: the proposed ASA system will process data from various sensors and provide real-time situational awareness to the vessel control system. This should allow the vessel to make informed decisions and take appropriate actions in a dynamic and changing environment.
- Adaptive decision-making: the proposed ASA system will use machine learning algorithms to fuse sensor data and make adaptive decisions based on the vessel's goals, objectives, and COLREGs.

The main research questions that should be considered during this development are the following:

1. What are the requirements for an ASA system for autonomous navigation of ASVs in busy open waters?
2. How can machine learning techniques be used to accurately identify and track other vessels, obstacles, and navigational hazards in real-time?
3. How can the ASA system be integrated with the ASV's collision avoidance planner to generate safe and compliant trajectories that comply with the Convention on the International Regulations for Preventing Collisions at Sea (COLREGs)?
4. How can the performance of the ASA system be tested and validated in various scenarios, including heavy traffic and adverse weather conditions?

Module 2: Readability of Human Rules

Laws and regulations, such as COLREGs, are inherently rule-based. They invariably state constraints that must be followed or activities for which permission or obligations are given. These rule-based conventions that govern the behaviour of entities in the world need to be captured so that robotic and autonomous systems do not violate them. While machine learning (ML) approaches can likely capture some intended constraints on behaviour, given enough effort on creating training examples, the uncertainly in the quality of a ML output does not support an acceptable level of trust in any ML-based system. For instance, having an autonomous vehicle that only takes action to avoid a crash 90% of the time, while acceptable for a machine learning academic work, it is unacceptable in real situations. In many cases, it may also be impractical or at least inefficient to generate a statistically significant set of training examples for every possible relevant scenario. In this context, we propose a combination of ML algorithms with approaches for representing reasoning about COLREGS, such as employing the Suggested Upper Merged Ontology (SUMO) (Niles and Pease 2001), which embodies two decades of work on a reusable inventory of common concepts.

In general, existing collision avoidance algorithms translate COLREGS as a set of hard navigational constraints (e.g. (Statheros, Howells, and Maier 2008)), ignoring the navigator's intentions and potential misunderstanding (or misuse) of the rules intended semantics. Past research has been devoted to the formal representation of a limited set of COLREG rules in terms of ontologies (Kreutzmann et al. 2013; Dylla 2009). However, this previous research has used a very limited notion of high-level formalisation (akin to a taxonomy rather than to a well-defined, formalised, body of knowledge), and the rules defined were written independently from the formalisation of other important concepts (such as those related to meteorology, or sea conditions). Moreover, little attention has been given to the human understanding of these rules and, to the best of our knowledge, there was no mention in the current scientific literature about how to interpret a ship's behaviour, considering the navigator's intentions. Therefore, the present proposal aims at bridging the gap between the human interpretation of COLREG rules, situation awareness (provided by ML algorithms) and real-time path planning systems. As ever larger commercial ships are provided leaner crews, the necessity of ship automation becomes more pressing. This proposal has the potential to accelerate the transfer of autonomy from at least some human-guided vehicles to machine-guided scenarios.

Module 3: Path Planning and Control

This part of the project aims to develop and explore multi-constraint optimisation-based planners (Lan-Xuan et al. 2020; Tsolakis et al. 2022) that can efficiently identify long-term trajectories for Autonomous Surface Vehicles (ASV) while ensuring compliance with the human-understanding of COLREGs. To achieve these objectives, the planners will utilise multi-constraint optimisation techniques. This approach involves finding the best trajectory for the ASV, considering multiple constraints that may be related to the vehicle's dynamics, environmental conditions, and most importantly, adherence to the COLREGs formalisation obtained in Module 2. In order to anticipate future collisions and plan safe trajectories, the planners will conduct joint forward simulations of both the ASV and other manned/unmanned vessels operating in the region. By simulating the movements of all vessels together, potential collision scenarios can be identified and avoided. Virtual obstacles will be constructed to represent the constraints imposed by COLREGs during the optimisation process. These obstacles will effectively encode the navigational rules and regulations specified in the COLREGs, ensuring that the trajectories generated by the planner adhere to international maritime regulations. The evaluation of the proposed planners will be carried out in two parts. First, planners will be tested in single-ship encounters to demonstrate their ability to produce COLREG-compliant trajectories when encountering a single other vessel. Second, the planners will be compared against state-of-the-art methods in more complex scenarios involving multi-ship encounters, whereas the simulation will use real descriptions of ship-on-ship encounters (including edge cases in which distinct interpretations of the rules could be applied). These multi-ship encounters pose greater challenges, as the planners must navigate through potentially crowded and dynamic environments, with autonomous, semi-autonomous and manual ships, while avoiding collisions and strictly adhering to the COLREGs.

The success rates of the proposed planners will be defined as key performance metrics. These metrics should measure the planners' efficiency and effectiveness in generating trajectories that comply with the COLREGs, according to the general human understanding of these rules. The success rates may include the percentage of successful COLREGs-compliant trajectories generated in different scenarios, the average time taken to find a feasible trajectory, or the overall safety and collision avoidance performance in both single and multi-ship encounters.

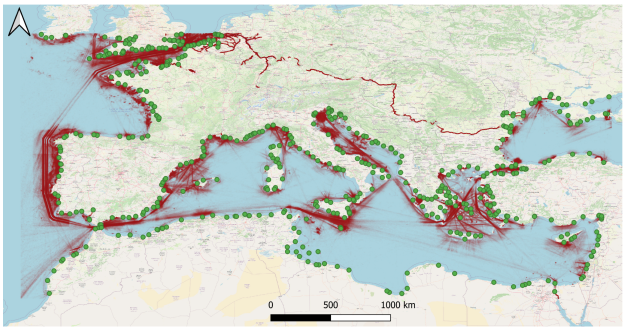


Figure 1: Illustration of maritime traffic based on AIS data in Europe (from (Elayam, Ray, and Claramunt 2022)).

To facilitate proactive collision avoidance, autonomous vessels will need to be able to make long-term trajectory predictions, taking into account the situation awareness inferences output by the algorithms developed in Module 1, and also by artificially representing the navigation experience while emulating the human mental models that facilitate these functions. We suggest taking advantage of Machine Learning to emulate the development of mental models that are constrained by (and consistent with respect to) the COLREG formalisations provided in Module 2.

We list here the steps that can feed our methodologies to go from data to models.

1. **AIS Data**. We propose to exploit historical Automatic Identification System (AIS) data to serve as a synthetic form of navigation experience. AIS relays information on ship behaviour, such as position, heading, speed, and ship type. By examining AIS histories, we can get an idea of the historical behaviour of the vessel. This can be seen as analogous to a navigator's experience of historical ship behaviour for a given geographical region. For the European zone, the AIS data recovered in 2016 give the map shown in Figure 1. Current work at the Ecole Navale (Elayam, Ray, and Claramunt 2022) can be used as a basis for the study. To facilitate long-term trajectory predictions, it is desirable to develop methods to emulate the development of mental models. Thus, methods must first be developed to classify data into categories of specific ship behaviour. Next, methods to facilitate the matching of trajectory segments to a category of ship behaviour should be investigated. Finally, methods to model the dynamics within each category of ship behaviour need to be developed. In this way, a new trajectory can be assigned to a given category, and its future trajectory can be predicted based on the unique behaviour of that category.

   The development of this part of the project should follow the sequence of steps listed below:

2. **Clustering**. The first step in mimicking the development of mental models is to facilitate categorisation functions. To predict future ship trajectories, historical ship behaviour needs to be decomposed into groups of specific behaviours. This is analogous to *clustering*, where similar data are grouped into clusters.
3. **Classification** The second step is to facilitate the mapping of a new trajectory to one of the existing categories of ship behaviour. Model matching will facilitate the selection of the appropriate category used to understand the situation, as well as the use of the corresponding behaviour model to project future dynamics. This is therefore classification in the sense of Machine Learning.
4. **Prediction**. To facilitate the prediction of long-term trajectories, each cluster must have a behavioural model. This mirrors the same functionality of mental models, where each category has a transition model capable of predicting future dynamics. Machine Learning is also capable of facilitating such regression functions.
5. **Deep Learning to emulate high-level *situation awareness*.** In the longer term, Deep Learning and AIS for ship trajectory prediction seem to be a promising development path. It could be proposed that historical AIS data for a given region be aggregated to reflect specific historical behaviour.

In this study, we will attempt to show that proactive collision avoidance actions taken some time before the nearest point of approach can make the cohabitation of autonomous and non-autonomous vessels on a given stretch of water safer and more fluid. The aim of taking into account the behaviours recorded by the AIS is to allow conventional vessels and autonomous vehicles to co-habit. We will seek to develop models through a classification of ship behaviour, where each class will have a specific transition function that models future dynamics. When the models are matched, a new trajectory will be able to fall into one of the existing classes, and the appropriate model will be applied to predict the future behaviour of a given vessel.

Module 4: Human Acceptability

This module aims to investigate human factors associated with the human acceptance of autonomous maritime vessels in following a set of rules of navigation / interaction (COLREGS in the case of a maritime application). These salient human factors include situational awareness, decision-making, workload changes, skills and training requirements, trust and reliance, and ethical considerations. Understanding stakeholders (currently humans who adhere to COLREGs) and their acceptance of autonomous systems of their vessels is of interest for future research and will inform the development of guidelines and integration of advanced autonomous systems and agents in maritime environments.

In the context of humans and AI, adoption and adaption are related but distinct concepts. Adoption refers to the decision to implement or use AI technologies. In other words, if the human takes on AI. This involves deciding to implement AI technologies in processes and could include purchasing of developing AI systems. Adoption is considered a strategic decision that considers factors such as cost, benefits, and risk, but can also consider perceptions by potential users. Adaption refers to the practical actions and strategies taken to integrate and interact with these technologies effectively. In other words, how the human uses AI. This involves practical steps to integrate and interact with AI systems, including developing data management protocols, training other users, and monitoring the performance of the system. Adaption is an ongoing process that requires continuous evaluation and adjustment to ensure that AI systems remain effective and aligned with the users' objectives. Acceptance underpins the adoption and adaption of AI technologies. Without acceptance, users may resist or be hesitant to use AI technologies, which can limit their effectiveness and potential benefits. It could be considered the foundation for any effort to implement and integrate AI technologies effectively. Promoting the acceptance of AI technologies can also help address any concerns or fears users may have about AI's impact on employment, privacy, and society (for example). Building trust and understanding of AI technologies means that users can better leverage the benefits of these technologies while also mitigating any potential risks or concerns. Understanding acceptance may be iterative depending on adoption and adaption and needs to be further understood.

The use of autonomous systems in the context of COLREGs introduces new challenges related to legal and regulatory frameworks, liability and responsibility, cybersecurity, and human factors. For example:

- clear guidelines and regulations to govern the use, and consequences, of COLREGs in the context of autonomous systems;
- considerations of liability and responsibility in the event of an incident or accident. Who would be held responsible – particularly if there was zero crew?
- Autonomous systems rely on software and communication technologies, so, this can make them vulnerable to cyber-attacks. Cybersecurity is an increasingly important issue in all realms, not least maritime;
- An autonomous system could help improve safety and efficiency, but they can also introduce new risks related to human factors such as situational awareness or monitoring of systems and performance.

Human factors are important when considering Adoption, Adaption, and Acceptance (AAA). Some of these reasons include perceived challenges or changes in situational awareness, familiarity (or unfamiliarity) in the operational context, changes in workload, skills and training requirements, trust and reliance, and ethical considerations. Autonomous systems could have a meaningful impact on human factors in maritime operations, and it is important to address these factors in design and implementation of autonomous systems to ensure safe and effective integration into maritime operations, in other words, to increase adoption, adaption, and overall acceptance:

- Adopting autonomous systems aboard maritime vessels may be interpreted by users in opposing ways. Initial research should aim to understand barriers and drivers for the perceived implementation of these systems. This process should feedback on design and development;
- Adaption should be tested and observed with users and considered with a feedback system between users and designers/developers;
- Acceptance should be understood to inform on policy change and other associated regulatory policies and procedures.

These assessments are recommended to include a blend of methodologies to collect the most salient data, as these systems will directly impact stakeholders. It is recommended that qualitative data such as interviews and focus groups are conducted throughout the AAA process. Quantitative data can be collected simultaneously to ensure congruence between what is said and observed, and what is thought. Validated instruments such as the Technology Acceptance Model (TAM) (Davis 1989; Sohn and Kwon 2020) could be adapted to autonomous systems throughout the AAA process to map changes in user perceptions. Paired with this, users are encouraged to participate in experiments from initial simulations to real-world observations across the AAA process. These experiments could include replications of real-world scenarios where decision-making, situational awareness, and workload changes are experienced with the use of the autonomous system (such as realistic scenarios, for instance, fatigue of crew due to illness, extreme weather, or high-stakes service-related demands). This would be useful for developers and stakeholders across the AAA process to encourage the final acceptance of autonomous systems. This process would highlight ethical issues and provide insights into the future skills and training required to continue to sail vessels but in combination with autonomous systems.

First Simulations

In order to measure the success rate of the development of an avoiding collision system for autonomous, semi-autonomous, and manual ships, a new light simulator has been developed. The main objective of this simulator is to both replay a past scenario based on collected AIS data, and add multiple ASV in the scene without causing any disturbance. This simulator also takes into account the rules of the COLREGs to avoid any kind of collision.

To include an ASV in maritime traffic without any disturbance, any ASV coming from the AIS data are considered as obstacles. Therefore, all added ASV must correct their initial trajectory to avoid collision with the AIS data AVS. To solve this issue, a privilege scale is introduced (from 0 to 1000). Before running the simulation, multiple ASV can be initialised, with a position (x and y), a speed (v), a heading (θ), but also with the type of ASV it is needed to create. Here are the different types of ASV that have been implemented as illustrated in Figure 2:

- **Boat**: Representing a pleasure boat with a hull length less than or equal to 24 metres. Its privilege degree is set to **0** because it is the most mobile agent.
- **Ship**: Representing an ocean liner (couple of hundreds metres), to simulate bigger agents, slower with less manoeuvrability. Therefore, its privilege degree is set to **30**.
- **Whale**: Representing all the agents for which no information are known, so with a complete uncertainty about their future trajectory. It can represent marine animal, jet-skis, or any another agent too small to be registered or detected. Due to its unknown behaviour, its privilege degree will be set to **500**.
- **Island**: Representing any sort of non-mobile agent for which it is impossible to move and avoid a collision. It could be a natural obstacle part of the environment like an island, a rock, a reef, or an anchored agent. Therefore, its privilege degree is set to the maximum value **1000**.

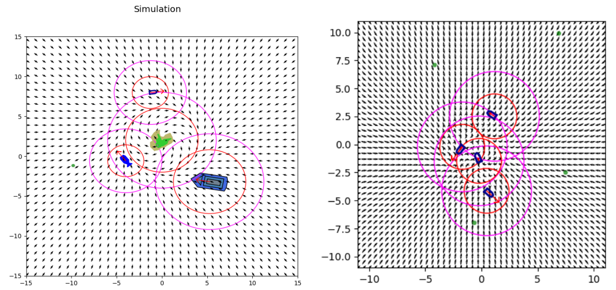


Figure 2: Illustrations of different simulation runned with different types of agents.

Rules :	111	222	333	444
Finish OT		▨		
Port OT	▨			
Starboard OT				
R to R				

Finish overtaking the obstacle	Finish OT
Overtaking the obstacle on the port side	Port OT
Overtaking the obstacle on the starboard side	Starboard OT
Red to Red rule to avoid collision	R to R

Table 1: The rules grid is displayed alongside the main simulation to visualise which COLREGs'rules are applied. The upper numbers are the agent MMSI numbers.

The structure of the simulator offers the option of adding new types of maritime objects.

The ASV which will have to operate the collision avoidance manoeuvre will be the one with the lowest level of privilege, as it will be easier for it to move than the other one, which will be considered as an obstacle. The rules that are applied during the scenario (and a reminder of full names) are highlighted as shown in Table 1.

The very first collision avoidance system consists of a potential based method and an associated controller. The collision zone is divided into four parts, and an adapted artificial potential field for each sector is applied. This potential field is updated every iteration depending on the new situation. This system is coupled with a path planning method. These two algorithms will be updated with the output of the re-

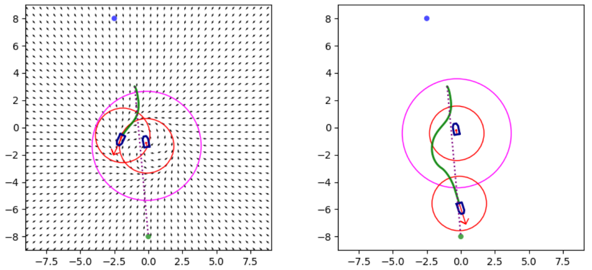


Figure 3: Illustration of Rule 14 - Head-on situation from the COLREGs implemented in the simulator.

search project according to module 3.

The AIS data ASV will move according to their AIS data located in .csv files, and the added ASV will have a final destination automatically calculated depending on their initial heading.

Multiple investigations can be considered using the simulator such as:

- *Testing acceptability*: different kind of situations can be run, which will each illustrate a situation where a collision is possible without any sort of intervention to correct it. The system will be acceptable if the potential collisions are avoided, and if the chosen new trajectory is realistic and validated by the AAA process. In Figure 3, the initial path is represented by a purple dotted line. The red circle represents the distance to the closest point of approach (DCPA); in other words, it is the area which must not be overcome by any sort of obstacle. For the magenta circle, it represents the manoeuvring zone, where boats will have to start the avoiding collision manoeuvres also in respect of the COLREGs rules. The result is the trace of a new green line, which represents the corrected chosen path.
- *Testing feasibility*, any new collision avoidance or path planning algorithm can be tested. As an example, the current chosen path planning algorithm could easily be replaced by another one, like the Bézier polynomials method or any AI-based algorithm, to study an other situation where obstacles would be registered. As a light simulator, it has been built to be suitable for long learning/training.

Conclusion

The Convention on the International Regulations for Preventing Collisions at Sea, or COLREGs, primarily focus on rules for safe navigation at sea, assuming that all vessels share the common goal of avoiding collisions. However, it is important to consider the context in which these regulations operate, especially in relation to human navigators and their behaviour.

- While COLREGs provide a framework for safe navigation, understanding the motivations and decision-making processes of human navigators can be crucial in ensuring compliance with these regulations. There may be cases where better understanding and modelling of human navigators can lead to improved outcomes, such as fewer collisions and near misses.
- In some situations, sailors may find themselves in mixed-motive scenarios where compliance with COLREGs conflicts with other goals or motivations, such as the desire to minimise time to destination or fuel consumption. Understanding these competing motives and how they influence human navigators' decisions can help design better training programmes, develop more effective autonomous systems.
- Understanding human motivations and concerns is also relevant in the context of autonomous navigation. Many stakeholders, including shipowners, captains, and crew members, may have reservations about adopting autonomous navigation systems. By considering and addressing these concerns, such as trust in technology, it may be possible to increase the acceptance and adoption of autonomous navigation solutions, which in turn could contribute to safer and more efficient maritime operations.

The objective is to get a deeper understanding of human behaviour and motives and to find out how it can contribute to safer and more effective maritime operations. It can also bridge the gap between regulatory frameworks such as COLREGs and the practical challenges of ensuring compliance and safety in real-world navigational scenarios.

This paper also proposed a combined approach to advance autonomous maritime navigation through the development of multi-constraint optimisation-based planners involving the formalisation of the human understanding of COLREG rules. The primary objective is to identify long-term COLREGs-compliant trajectories with a high navigation success rate for autonomous surface vessels (ASVs) while ensuring safe encounters with both manned and unmanned vessels within the region.

This research focuses on demonstrating the effectiveness of the proposed methods in single-ship encounters and benchmarking its performance against state-of-the-art techniques in multi-ship scenarios. The success rates obtained from extensive simulations serve as crucial indicators of the efficacy of the approach, but they are not the final aim.

References

Burmeister, H.-C.; and Constapel, M. 2021. Autonomous Collision Avoidance at Sea: A Survey. *Frontiers in Robotics and AI*, 8.

Cockcroft, A.; and Lameijer, J. 2012. *A Guide to the Collision Avoidance Rules (Seventh Edition)*. Oxford: Butterworth-Heinemann.

COLREGs. 1972. *Convention on the International Regulations for Preventing Collisions at Sea (COLREGs)*. International Maritime Organization (IMO) London, UK.

Davis, F. D. 1989. Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly*, 13(3): 319–340.

Dylla, F. 2009. Qualitative Spatial Reasoning for Navigating Agents - Behavior Formalization with Qualitative Representations. In Gottfried, B.; and Aghajan, H. K., eds., *Behaviour Monitoring and Interpretation - BMI - Smart Environments [an outgrow of BMI workshops]*, volume 3 of *Ambient Intelligence and Smart Environments*, 98–128. IOS Press.

Elayam, M. M.; Ray, C.; and Claramunt, C. 2022. A hierarchical graph-based model for mobility data representation and analysis. *Data & Knowledge Engineering*, 141.

Kreutzmann, A.; Wolter, D.; Dylla, F.; and Lee, J. H. 2013. Towards safe navigation by formalizing navigation rules. *TransNav: International Journal on Marine Navigation and Safety of Sea Transportation*, 7(2): 161–168.

Lan-Xuan, A.; Xin-Yu, Z.; Zhi-Guo, D.; and Cheng-Bo, W. 2020. Research on ontology-based situation understanding and decision-making approach for MASS. In *IOP Conference Series: Materials Science and Engineering*, volume 929, 012029. IOP Publishing.

Niles, I.; and Pease, A. 2001. Towards a Standard Upper Ontology. In *Proceedings of the International Conference on Formal Ontology in Information Systems - Volume 2001*, FOIS '01, 2–9. New York, NY, USA: Association for Computing Machinery. ISBN 1581133774.

Sohn, K.; and Kwon, O. 2020. Technology acceptance theories and factors influencing artificial Intelligence-based intelligent products. *Telematics and Informatics*, 47: 101324.

Statheros, T.; Howells, G.; and Maier, K. M. 2008. Autonomous Ship Collision Avoidance Navigation Concepts, Technologies and Techniques. *The Journal of Navigation*, 61(1): 129–142.

Tsolakis, A.; Benders, D.; de Groot, O.; Negenborn, R.; Reppa, V.; and Ferranti, L. 2022. COLREGs-aware Trajectory Optimization for Autonomous Surface Vessels. *IFAC-PapersOnLine*, 55(31): 269–274. 14th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles CAMS 2022.