Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 58-20 (2024) 147–152

COLSim, a simulator for Hybrid Navigation Acceptability and Safety

Benoit Clement *,** **Thomas Chaffre** ** **Pouria Sarhadi** *** **Marie Dubromel** *

* *ENSTA Bretagne, CROSSING IRL CNRS 2010, Adelaide, Australia*
** *Flinders University, Adelaide, Australia*
*** *University of Hertfordshire, Hatfield, UK*

Abstract: Autonomous vessels have emerged as a prominent and accepted solution. However, achieving full autonomy for marine vessels requires the development of robust and reliable mission planning and control systems that can handle various encounters with manned and unmanned vessels while operating effectively in various weather and sea conditions. These algorithms need to account for various aspects of a mission like global/local planning, manoeuvrability limitations, external disturbances, Collision Avoidance (COLAV), motion regulations. A significant challenge in this pursuit is ensuring the autonomous vessels' compliance with the International Regulations for Preventing Collisions at Sea (COLREGs). This paper proposes a (prototypical) realisation of an open-source simulator including replay of AIS Data. The aim of this simulator is to provide fast simulation feeded with real data and including autonomous agents.

Copyright © 2024 The Authors. This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0/)

Keywords: COLREGs, Simulation, Situation Awareness, DRL, Human Factors.

1. INTRODUCTION

Autonomous vessels are gaining acceptance, as they offer an obvious means of removing human personnel from risks, but they also need the acceptance of the classical vessels (Wilkinson et al. (2024)). A critical requirement for 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 and 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 (1972); Cockcroft and Lameijer (2012)). COLREGs evolved from a set of practices that were originally designed in the mid-19th century for human interpretation, and the classical process is given in Fig. 2. Therefore, these rules are written in qualitative and sometimes ambiguous or qualitative prose, assuming that their interpretation and execution were carried out by highly experienced sailors and not by an autonomous system even if the IMO (International Maritime Organisation) is about to provide an automonous ship code (IMO (2021)). Here is a list of the relevant COLREGs rules (13 and 14 are illustrated in Fig. 1) that are considered in our work:

- R8 - *Actions to avoid collision*: if there is sufficient sea-room, alteration of course alone may be most effective. Reduce speed, stop or reverse if necessary.
- R13 - *Overtaking*: Any vessel overtaking any other shall keep out of the way of the vessel being overtaken.
- R14 - *Head-on*: Each head-on vessel shall alter her course to starboard so that each shall pass on the port side of the other.

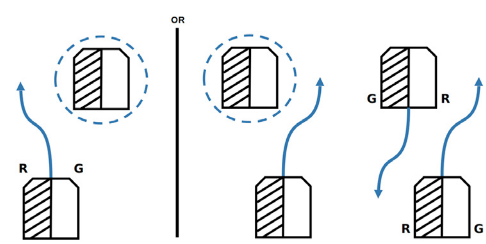

Fig. 1. Illustration of Rule 13 (left) and Rule 14 (right) - the dotted circle represent who has the priority and **G** for green (port) and **R** for red (starboard)

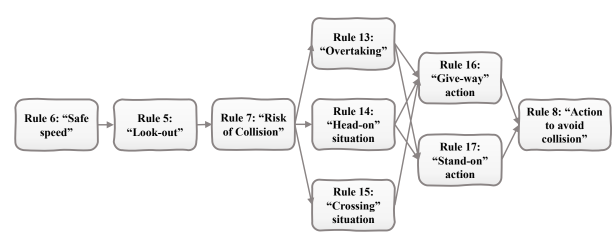

Fig. 2. Decision-making process based on COLREGs rules from Sarhadi et al. (2022b)

- R15 - *Crossing*: The vessel which has the other on her own starboard side shall keep out of the way.
- R16 - *Actions by give-way vessel*: Take early and substantial action to keep things clear.
- R17 - *Actions by stand-on vessel*: Keep her course and speed, but may take action to avoid collision if the other vessel is not COLREGs-compliant.

Classical rule-based and model-based approaches to automated COLREG compliance have proven to be too complicated to accommodate all possible encounters, envi-

2405-8963 Copyright © 2024 The Authors. This is an open access article under the CC BY-NC-ND license.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2024.10.046

ronment scenarios, and human behaviours (Statheros et al. (2008)). Modern Machine Learning (ML) methods, such as Deep Reinforcement Learning (DRL), could provide a flexible and adaptable model-free guidance and obstacle avoidance system, by which multiple possible interactions and scenarios can be abstracted from previous observations (Burmeister and Constapel (2021)). Nevertheless, the deployment of high-fidelity simulation tools to develop and test these algorithms will be a key enabler. This work is a preparation for the development of an AI-based COLREG-compliant model-free collision avoidance system that will be trained using historical AIS (Automated Identification System) of real-world scenarios, while the possible human interpretations of the rules will take a central point in the development presented in Clement et al. (2023).

For evaluation of our future work, we propose an open-source custom-developed ship simulator that models ship movement constraints such as engine force and drag and reaction of the ship to environmental changes. The expected simulator is implemented in Python, light and open-source with a collision avoidance system on every simulated Unmanned Surface Vessel (USV), that adheres to the COLREGs. The developed simulator can be used to train ML techniques to deploy historical AIS-based simulations of real-world scenarios. In this paper, the anticollision algorithm is based on the Artificial Potential Field (APF).

2. COLSIM SIMULATOR

The proposed simulator is conceptually similar to two existing simulation packages. The *UTSeaSim simulator* (Agmon et al. (2011)) 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 simulator also encompasses a graphical user interface (GUI) to visualise the simulation, control the simulation parameters, and monitor the vehicle behaviour (Crase et al. (2013)). *COLAV* (Hybrid Collision Avoidance) (Eriksen et al. (2020)) represents a three-layered collision avoidance system adapted for USV, designed to adhere to rules 8 and 13-17 outlined in COLREGs. It integrates several crucial components: a high-level planner for generating energy-optimised trajectories, a mid-level COLAV algorithm founded on model predictive control, which takes into account dynamic obstacles and navigational regulations, and a branching-course model predictive control algorithm. The latter ensures short-term collision avoidance manoeuvres in emergency scenarios while adhering to the COLREGs.

Our objective is to develop an easy-to-use simulator (contrary to very high fidelity ones like Lechene et al. (2024)) with low computational needs that is capable of generating large amounts of simulated data. The ability to generate high-quality and quantity data is ultimately the key to the development of flexible and adaptable maritime autopilots, and the proposed simulator aims to address this challenge.

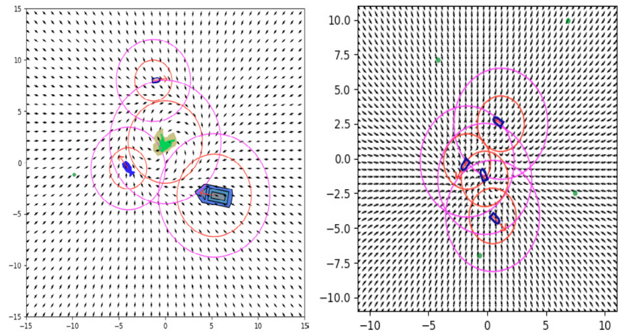

Fig. 3. Illustrations of different simulation runs with different types of agents.

2.1 Description

To measure the success rate of the development of a collision avoidance system for autonomous, semiautonomous, and manual ships, a new light simulator is under development. The main objective of this simulator is to replay past scenarios based on collected AIS data and to add multiple USVs to the scene without causing any disturbance.

To include a USV in maritime traffic without any disturbance, any agent (USV or vessel) coming from the AIS data is considered as an obstacle. Therefore, all added USVs must correct their initial trajectory to avoid collision with the USV AIS data. To solve this issue, we use a privilege scale (from 0 to 1000). Before running the simulation, multiple USV can be initialised, with a position (x and y), a speed (v), a heading (θ), and a type of USV. The latter are listed below and illustrated in Figure 3:

- **Boat**: a small vessel (size can be adjusted). Its privilege degree is set to **0** because it is the most mobile agent.
- **Ship**: a large boat to simulate bigger agents, slower with less manoeuvrability. Its privilege degree is set to **30**.
- **Whale**: all agents for whom no information is known, with uncertainty about their future trajectory (marine animals or any other agent too small to be registered or detected). Due to its unknown behaviour, its privilege degree is set to **500**.
- **Island**: any sort of static agent. It could be a natural obstacle part of the environment, such as an island or a mooring agent. The privilege degree is set to the maximum value **1000**.

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

2.2 Structure

This simulator is accessible through GitHub `https://github.com/clemenbe/COLSim` and is under development. This directory contains the most up-to-date and comprehensive version of the USV simulator, intoducing a more sophisticated structure for the simulation of sea objects. The main components are:

- **Simulation Runner**: This class initialises all objects and constants for the simulation, such as *s, dt, k, num_steps*. A function called *initialize_sea_objects()*

creates various sea objects (*Boat, Ship, Whale, Island*, etc.). A separate function *initialize_data* displays the rules of collision avoidance at sea (COLREGs). The *run()* function then takes this *sea_objects* vector and the list of rules and MMSI (Maritime Mobile Service Identity) numbers, passing them to the *Simulation* class.

- **Simulation**: This class runs 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 choose 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 (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 not currently in collision with another.
  - The *move()* method 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 , the privilege of the two objects is compared. 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**.
  - The *move_straight()* function is called when the *SeaObject* is either 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.

Regardless of the actions taken, a control vector u_p is generated, which is then used by the *update()* function to update the position of the object. 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. Following this paradigm, each sea object is responsible for its actions, deciding to avoid collision or not based on the rules defined in its associated methods. Therefore, the actions of an object are not correlated with the actions of other objects, resulting in a fully autonomous decision-making process based on the association of internal rules and the perception of the environment.

3. COLLISION AVOIDANCE SYSTEM

3.1 Artificial Potential Field controller

The dynamics of the USV are represented by the following state equations:

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

where $\mathbf{p} = [x, y]^T$ is the USV position in the global reference frame. In addition, an obstacle is denoted $\mathbf{q}$ and its target position is denoted as $\hat{\mathbf{q}}$. USVs navigate through a crowded environment, including mobile and static obstacles such as other boats (manned and unmanned) or rocks. As a first step, the chosen approach is based on the APF method proposed in Khatib (1986). 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 with the context and environment of the study. The following relation is based on classical physics:

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

where $\mathbf{p}$ is the position of the point particle in space, $\mathbf{V}$ is the potential and $\mathbf{f}$ is the force applied to 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 $\mathbf{V}$ that includes 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. By deriving the potential Eq. (3) and incorporating it in the formulation of the force Eq. (2), we can determine the force applied to the USV as a function of its position and speed:

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

To complete the controller, the commands of the vector $\mathbf{u}$ need to be calculated as shown in Fig. 4.

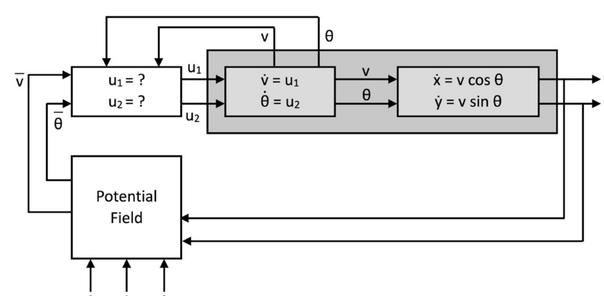

Fig. 4. Structure of the potential field based controller.

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

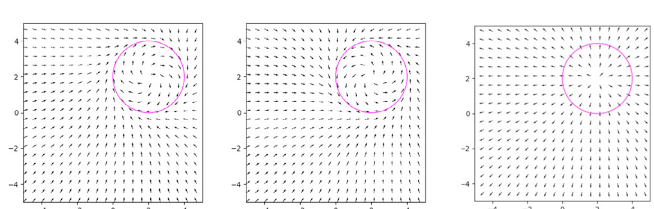

Fig. 5. Clockwise (left), Counterclockwise (middle) and Repulsive (right) APF.

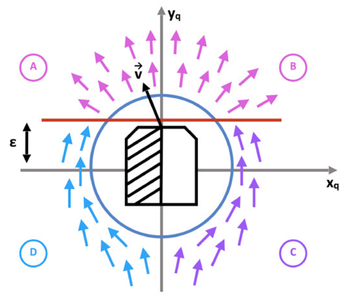

Fig. 6. Illustration of the various vector fields applied in the avoidance of a collision with similar heading USVs.

$$\begin{cases} y = x_3 - \text{atan2}(\psi_2, \psi_1), \\ \dot{y} = \dot{x_3} + \dfrac{\psi_1 \cdot \dot{\psi_2} - \psi_2 \cdot \dot{\psi_1}}{\psi_1^2 + \psi_2^2} = u + \dfrac{\psi_2 \cdot \dot{\psi_1} - \psi_1 \cdot \dot{\psi_2}}{\psi_1^2 + \psi_2^2}. \end{cases} \quad (5)$$

The command u is obtained using feedback linearisation:

$$\begin{cases} u_1 = 0, \\ u_2 = u + \dfrac{\psi_2 \cdot \dot{\psi_1} - \psi_1 \cdot \dot{\psi_2}}{\psi_1^2 + \psi_2^2}. \end{cases} \quad (6)$$

Three different situations can be identified and will therefore need an adapted potential field each. The potential field applied to the USV will depend on the following parameters: the position of the USV involved in a possible collision, and also on their orientation with each other. The evaluation of orientation will help determine whether it is a situation of overtaking, or Red to Red i.e. rule 13 or 14 according to COLREGs (1972). Those fields will be applied according to certain conditions developed in Section 3. With p being the USV considered and its state vector $[x_p, y_p, v_p, \theta_p]^T$, and q the USV obstacle, a case disjunction can be established. Let us set the matrix D as:

$$D = \begin{bmatrix} \mathbf{R} & 0 \\ 0 & \mathbf{R} \end{bmatrix}, \quad (7)$$

where $\mathbf{R}$ is the radius of the circle from the vector field. Then, we can defined the vector field as:

$$\phi_0(p) = \begin{bmatrix} -x_p^3 - x_p \cdot y_p^2 + x_p - y_p \\ -y_p^3 - y_p \cdot x_p^2 + x_p - y_p \end{bmatrix}. \quad (8)$$

The coordinates of the centre of the circle of the potential field is defined as $C = [c_x \; c_y]^T$. Finally, ϵ is a quantity created to induce time delays in the launch of the collision avoidance manoeuvre, allowing the USV to evade dangers more smoothly with more realistic moves.

Case 1: If $\mathbf{v_q} \cdot \mathbf{v_p} > 0$ (geometrical scalar product): The USVs are approximately heading towards the same direction, requiring an overtaking (rule 13). This case is pictured in Fig. 6, resulting in different operating zones.

- **Zone D** : The USV is approaching the position of the obstacle on its left side. The shortest way to avoid the obstacle is to go clockwise, **e** is set to -1. Therefore, $g(\mathbf{p}) = D \cdot \begin{bmatrix} 1 & 0 \\ 0 & e \end{bmatrix} \cdot \mathbf{p} + \mathbf{c}$. The equation for the clockwise vector field is the following:

$$\phi_{cw}(p) = \mathbf{g} \circ \phi_\mathbf{0}(\mathbf{p}) = \begin{bmatrix} R & 0 \\ 0 & -R \end{bmatrix} \cdot \phi_\mathbf{0}(\begin{bmatrix} R & 0 \\ 0 & -R \end{bmatrix}^{-1} (p - c)). \quad (9)$$

- **Zone C** : The USV is approaching the position of the obstacle on its right side. The shortest way to avoid the obstacle is to turn counterclockwise; therefore **e** is set to 1. The equation for the clockwise vector field is the following:

$$\phi_{ccw}(p) = \mathbf{D} \cdot \phi_\mathbf{0}(\mathbf{D^{-1}} \cdot (\mathbf{p} - \mathbf{c})). \quad (10)$$

- **Zone B** : The USV is finishing to overtake the obstacle from the right side. To complete the collision avoidance manoeuvre, a repulsive field will be applied to the USV considered, around the same circle of radius **R** and centre **c**. Its point of repulsion will be the centre of the obstacle. The equation for the repulsive vector field is the following:

$$\phi_{rep}(p) = \begin{bmatrix} k \cdot (x_p - c_x) \\ k \cdot (y_p - c_y) \end{bmatrix}, \quad (11)$$

with here **k** fixed to **0.5**.

- **Zone A** : The USV finishes to overtake the obstacle from the left side. The same repulsive field will be applied to the USV considered, around the same circle of radius **R** and centre **c**. The equation for the repulsive vector field is the following:

$$\phi_{rep}(p) = \begin{bmatrix} k \cdot (x_p - c_x) \\ k \cdot (y_p - c_y) \end{bmatrix}. \quad (12)$$

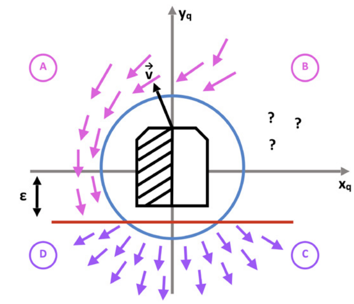

Fig. 7. Breakdown of the various vector fields applied in the avoidance of a collision when USV have opposite headings.

Case 2: If $\mathbf{v_q} \cdot \mathbf{v_p} > 0$: The two USVs are going in opposite directions. This can lead to a situation of Red-to-Red (rule 14). This case is defined in Fig. 7

- **Zone A**. The Red-to-Red rule needs to be applied. Therefore, a counterclockwise field $\phi_{\mathbf{ccw}}$ is chosen for this zone.
- **Zone B** : This zone can have two different outcomes depending on the heading of the two USV. If the two USVs have a similar heading (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, using a counterclockwise field $\phi_{\mathbf{ccw}}$.
- **Zone C and D** : The final step of the collision avoidance manoeuvre will apply a repulsive field $\phi_{\mathbf{rep}}$ will be applied.

These are the different APFs that will be used in the simulator depending on the situation considered.

3.2 Overall operating

The main function of the simulator, *avoid_collision*, is called when there is a risk of collision and will start a

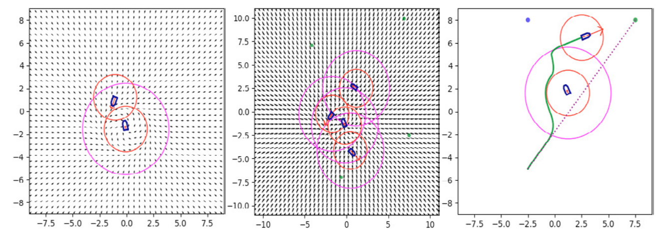

Fig. 8. (left) USV with DCPA in red following an APF, (middle) 4 USVs collision avoidance, and (right) 2 USVs collision avoidance with corrected path.

collision avoidance process. The condition for starting the process is that the distance between two USVs is less than a predefined threshold **r**. It represents the DCPA (Distance at Closest Point of Approach). Note that TCPA (Time to CPA) is also used as a threshold to determine whether a collision will occur soon or not. For more details on the CPA concept and calculation, Sarhadi et al. (2022a) provides the CPA calculation method in a simple form. The concept of CPA and the experience of a sailor are the key tools to determine the actions to take; the future human factors studies will try to determine new efficient and measurable indicators for this.

It is a crucial operational concept defined to deal with COLREGs to ensure 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 to determine potential collision situations. Since it helps sailors assess the risk of collision and 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 each USV. A second magenta circle has been created which, once crossed by another USV, activates the anti-collision system. Finally, the zone between the red and magenta circle is denoted as the **manoeuvring zone** where the boat performs avoidance manoeuvre based on COLREGs.

Depending on the position of the USV in the obstacle repository, one of the APF described in 3.1 will be applied. Then, once the obstacle is avoided, the USV will go back to its initial reference trajectory by Eq. (6) to follow the path to its final destination. Figure 8 shows simulation examples. The three examples represented in Figure 8 have been generated. A real-time monitoring system was implemented to track the application of COLREGs during collision avoidance scenarios. A dynamic table illustrates the rules used at any given time. This table offers a snapshot of the specific rules that are being applied as vessels navigate through potential collision scenarios by coloring in green the corresponding cells. The USV are identified by their MMSI number. Real-time feedback improves situational awareness and facilitates an assessment of the effectiveness of collision avoidance strategies implemented to facilitate future work.

When a collision situation occurs, a table, as shown in Fig. 9, is set up to make clear which kind of situation the simulator is dealing with.

Active rules of the sea

Rule	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

Fig. 9. The grid is displayed alongside the main simulation to visualise which COLREG's rules are applied.

Table 1. Different sources for AIS Data

Source	Area	Time	Fees
Marine Traffic	World	All	Not Free
AIS Hub	World	All	Free
Searates	World	All	Not Free
Datalastic	World	All	Not Free
VT Explorer	World	All	Not Free
My Ship Tracking	World	All	Not Free
My Ship Tracking	World	All	Free
MarineCadastre US	USA	2015-2023	Free
AMSA	Australia	from 1999	Free
Spire Maritime	World	All	Not Free

3.3 *Cross-path situation*

In some scenarios, a USV can enter the manoeuvring zone, but its orientation and speed will not lead to an actual collision with the second USV. In reality, an experienced sailor will not start a full collision avoidance process if it is not necessary and if there is no danger. Therefore, the simulator can be improved by implementing a special process for the **cross path situation**. For this purpose, a new instance variable has been created in the SeaObject class, a boolean *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 lower right 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 APF ϕ_{ccw} is applied instead of clockwise, as there is no risk of collision.
- **USV in the right lower zone heading to the lower left 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 APF ϕ_{cw} will be applied instead of a counterclockwise one.

4. AIS REPLAY

AIS is a protocol used by ships, drones, navigation aids, and some aircraft to identify themselves and report their position and trajectory. Analysing AIS data makes it possible to determine the behaviour of the vessel, including speed, direction, and adherence to the priority rules outlined by COLREG. AIS data allows us to obtain information such as position, speed, heading, and type of ship. Table 1 gives different sources for the AIS data.

4.1 Data Quality

The database is made up of records acquired from the National AIS, which includes approximately 58 ports and 11 coastal areas up to 24 nautical miles offshore. For further information on vessel reporting requirements and participation in the AIS. The data were acquired directly from primary sources, meaning that no additional testing was performed to cross-reference or confirm either the field or the geometry values from other sources. There are a significant number of null values within the dataset due mainly to vessel operator practices. The *position accuracy* of each vessel position report depends on the positioning device used by the vessel. In most cases, GPS and differential GPS positional devices have been used, ensuring an accuracy of +/- 10 metres horizontally.

4.2 Build a Database

To provide detailed information about the structure, content, and meaning of data within a dataset, a data dictionary was built as a reference guide to understand the various attributes or variables present in the dataset.

Our data dictionary includes the following:

- **Variable Names** or column headers in the dataset.
- **Description** or definition of each variable.
- **Data Type**.
- If applicable, **scale of measurement** for numerical variables (e.g., nominal, ordinal, interval, ratio).
- If applicable, **units** in which the variable is measured.
- For categorical variables, a list of **possible categories** or values the variable can take.
- Information about how **missing values** are represented in the dataset.
- Additional details about the data **format**.
- Information about the **source**, including where it was obtained and any relevant citations.
- For datasets with multiple tables, information about how different tables are related to each other.

Data dictionaries are essential for data understanding, data cleaning, data engineering, and analysis. They help ensure consistency in data interpretation and facilitate collaboration among team members working with the same dataset. The dataset can used to replay a situation, add a new agent in traffic or for an analysis to understand how the rules are implemented in real life.

5. CONCLUSION

The development of a robust and trustworthy control and guidance system for autonomous vessels is essential to achieve full autonomy and ensure safe navigation in various maritime environments. As a first step, COLSim simulator simulates maritime traffic scenarios and collision avoidance manoeuvres in accordance with COLREGs. The simulator allows replay of historical AIS-based simulations and training of ML methods using real-world scenarios. It incorporates an Artificial Potential Field (APF) controller for collision avoidance, which dynamically influences the vessel's path based on the surrounding environment. Additionally, the simulator includes a real-time monitoring system to track COLREGs compliance during collision avoidance scenarios. In addition, we have discussed the importance of AIS data in understanding the behaviour of the vessel and predicting potential collision scenarios. By analysing AIS data, we can determine vessel speed, direction, and adherence to COLREG rules, crucial to anticipate hazardous situations and ensure safe navigation.

REFERENCES

Agmon, N., Urieli, D., and Stone, P. (2011). Multiagent patrol generalized to complex environmental conditions. In *Proceedings of the 25th Conference on Artificial Intelligence (AAAI)*.

Burmeister, H.C. and Constapel, M. (2021). Autonomous collision avoidance at sea: A survey. *Frontiers in Robotics and AI*.

Clement, B., Dubromel, M., Santos, P., Sammut, K., Oppert, M., and Dayoub, F. (2023). Hybrid Navigation Acceptability and Safety. In *AAAI 2023 Fall Symposium Series*. Arlington.

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

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

Crase, E., Wideman, C., Noble, M., and Tarantola, A. (2013). UTSeaSim Documentation. Https://www.cs.utexas.edu/ UT-SeaSim/download/1.0/Oct2013Documentation.pdf.

Eriksen, B., Bitar, G., Breivik, M., and Lekkas, A. (2020). Hybrid collision avoidance for asvs compliant with colregs rules 8 and 13–17. *Frontiers in Robotics and AI*.

IMO (2021). Outcome of the regulatory Scoping Exercise for the use of Maritime Autonomous Surface Ships (MASS). Technical Report MSC.1/Circ.1638.

Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*.

Lechene, H., Clement, B., Sammut, K., Santos, P., Cunningham, A., Coppin, G., and Buche, C. (2024). LOTUS: Learning from Operational Teaming with Unmanned Systems. In *IEEE Oceans Conference*. Singapore.

Sarhadi, P., Naeem, W., and Athanasopoulos, N. (2022a). An integrated risk assessment and collision avoidance methodology for an autonomous catamaran with fuzzy weighting functions. In *UKACC 13th International Conference on Control (CONTROL)*.

Sarhadi, P., Naeem, W., and Athanasopoulos, N. (2022b). A survey of recent machine learning solutions for ship collision avoidance and mission planning. In *14th IFAC Conference on Control Applications in Marine Systems, Robotics and Vehicles*. Denmark.

Statheros, T., Howells, G., and Maier, K.M. (2008). Autonomous ship collision avoidance navigation concepts, technologies and techniques. *The Journal of Navigation*.

Wilkinson, C., Grosser, L., Oppert, M., Banks, S., and Clement, B. (2024). Automation at sea and human factors. In *15th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles*. Blacksburg.