Toward a COLREGs Compliant Autonomous Surface Vessel in a Constrained Channel

JAMES CONNOR MEYERS, THOMAS SAYRE-MCCORD, ZHIYONG ZHANG, AND HANUMANT SINGH (Fellow, IEEE)

1 Field Robotics Laboratory, Northeastern University, United States Coast Guard, Boston, MA 02115 USA
2 MIT-WHOI Joint Program, Boston, MA 02139 USA
3 Field Robotics Laboratory, Northeastern University, Boston, MA 02115 USA
CORRESPONDING AUTHOR: HANUMANT SINGH (ha.singh@northeastern.edu)

ABSTRACT In this article, we look at the role of autonomous navigation in the maritime domain. Specifically, we examine how an autonomous surface vessel (ASV) can achieve obstacle avoidance based on the Convention on the International Regulations for Preventing Collisions at Sea (1972), or COLREGs, in real-world environments. Our ASV is equipped with a broadband marine radar, and an inertial navigation system (INS), and uses official electronic navigation charts (ENCs). These sensors are used to provide situational awareness, and, in a series of well-defined steps, we can exclude land objects from the radar data, extract tracks associated with moving vessels within range of the radar, and then use a Kalman filter to track and predict the motion of other moving vessels in the vicinity. A constant velocity model for the Kalman filter allows us to solve the data association to build a consistent model between successive radar scans. We account for multiple COLREGs situations based on the predicted relative motion. Finally, an efficient path planning algorithm is presented to find a path and publish waypoints to perform real-time COLREGs compliant autonomous navigation within highly constrained environments. We demonstrate the results of our framework with operational results collected over the course of a 3.4 nautical mile mission on the Charles River in Boston in which the ASV encountered and successfully navigated multiple scenarios and encounters with other moving vessels at close quarters.

INDEX TERMS Autonomous driving, marine robots, simultaneous localization and mapping.

I. INTRODUCTION
A. NAVIGATION RULES (COLREGs)

The International Maritime Organization (IMO) promulgated the International Regulations for Preventing Collisions at Sea (COLREGs) [11] in 1972. These are considered the “rules of the road” or the navigation rules to be followed by ships and other vessels at sea to prevent collisions between two or more vessels.

One way to attempt safe navigation for unmanned surface vessels (USVs) is to emulate human behavior as closely as possible through COLREGs. In this article, we consider the situations represented in Fig. 1. These represent the responsibilities of power-driven vessels in sight of each other to prevent collision while passing close to each other. A far more detailed description of responsibilities between vessels is provided in [11].

B. RELATED WORK

There has been a lot of research on collision avoidance that is COLREGs compliant. García Maza and Argüelles [23] and Wróbel et al. [32] conducted analyses on current COLREGs compliant solutions and discussed challenges in the field. Zhou et al. [37] studied the application barriers presented by COLREGs. Woerner et al. [31] worked on quantifying and evaluating collision avoidance protocols, such as COLREGs.

For current solutions, there are expensive military applications, such as the common unmanned surface vehicle (CUSV) created by U.S. Navy that operate under dynamic environments, but the research is highly proprietary, and the sensor suites utilized are extremely expensive. Other methods use lower cost sensors and vessels, and some of them perform well. Blaich et al. [3] propose a collision avoidance
method without COLREGs constraints. Benjamin et al. [2] follow COLREGs by sharing the action and position of multiple autonomous surface vessels (ASVs). Liu and Bucknall [19] simulate a path planning algorithm in a static environment. Dubey and Louis [6] and Naeem et al. [24] simulate the collision avoidance strategies based on COLREGs. Benjamin and Curcio [1] address the problem of safe navigation under COLREGs in open waters. Johansen et al. [13] and Zhang et al. [35] outline methods for avoiding multiple dynamic obstacles in simulation. Kuwata et al. [18] provide a velocity obstacles-based solution for open water.

Many collision avoidance and path planning systems are also proposed. For example, [10] and [25] designed an adaptive fuzzy ASV inference system. References [5], [8], [9], [33], [34], and [36] use deep reinforcement learning to perform obstacle avoidance. Serigstad et al. [28] use a hybrid dynamic window (HDW) algorithm to perform collision avoidance. Kang et al. [15] use particle swarm optimization (PSO) algorithm to plan ship paths. Eriksen et al. [7] present a three-layered hybrid collision avoidance compliant with COLREGs rules 8 and 13–17. Lyu and Yin [20] use modified artificial potential fields to perform COLREGS-constrained collision avoidance in simulation.

C. OUR APPROACH

The method proposed in this article tackles all these issues as illustrated in Fig. 2. By using the WHOI Jetyak [17] and open-source hardware and software, costs are kept low compared with Naval and professional-grade sensor suites without significantly degraded performance. By using freely available marine charts to generate a binary occupancy grid of the environment, the Jetyak can identify static and dynamic obstacles and avoid other vessels while following COLREGs. It can do so in an environment constrained by land areas, shoal water, and other navigational hazards. All the results presented in this article are based on real data collected during on water trials with real vessels on the Charles River in Boston.

II. JETYAK ASV AND ITS SENSOR SUITE

The WHOI Jetyak was used as the ASV for tests presented in this article [17]. A Navico BR24 Radar was mounted on top of the Jetyak providing environmental sensing of dynamic and stationary obstacles, while a Vectornav VN-300 dual antennal internal navigation system was used to determine the Jetyak’s own position and yaw. The data acquisition is accomplished with robot operating system (ROS), and the use of the BR24 library [4] to collect radar data, MavRos [22] to collect odometry information, including speed, and to broadcast MavLink [21] messages, including waypoints and vehicle information, and the Vectornav library [30] to collect heading and GPS information from the VN-300. ROS messages received include approximately 800 radar scanlines per second. Each scanline contains 512 ranges with intensity values from 0 to 255. The scan rate of the radar is one full rotation every 2.5 s (one full 360° scan). Highly accurate heading information is delivered at 100 Hz from the VN-300 inertial navigation system (INS) with errors of the order of approximately 0.2° rms. All Mavlink messages are updated at 10 Hz, with GPS updates at ~2 Hz.

Marine charts are required for our algorithm, and they typically provide accurate descriptions of the land, water depths, navigational aids, and features useful to safe nautical navigation. The electronic navigation chart (ENC) provides a good description of the large-scale static environment. In our work, ENCs are used to overlay radar data and perform avoidance of the shoreline, other navigational hazards, and ultimately help us to extract target vessels in the water. Here, National Oceanic and Atmospheric Administration (NOAA) does not provide ENCs; as in our case on the Charles River, an open source geographic information system (QGIS) software [26] was used to create approximations of the land area using satellite imagery.

III. DATA PREPROCESSING

A. BINARY OCCUPANCY GRID

A binary occupancy grid is used to represent space occupied by known navigational hazards, such as coastlines and
bridges as well as target vessels. Any black areas are considered static obstacles on the map, and any white area is considered open water space. A path planner can determine an obstacle-free path based on this map. See Fig. 3 for a representation of the Charles River binary occupancy grid.

**B. RADAR PROCESSING**

We overlay the radar data on the binary occupancy grid map of the environment. It is completed by offsetting the radar scanline angle from the Jetyak’s heading and global position. The global position data are computed in Universal Transverse Mercator (UTM) coordinates. After the data overlay, we can extract target vessels on the river and exclude land area in the next step, as illustrated in Fig. 4.

There is a dead zone of returns near the radar, which appear as occupied grids, but, in reality, are artifacts of the radar. The dead zone occupied grids can sometimes connect to real obstacles near the Jetyak. To differentiate between these two cases, we find the lowest intensity point at the junction of the radar intensities, and radar signal returns before the junction point are considered the dead zone. Other signals after the junction are treated as obstacles. We filter out all the signals that are considered to be in the dead zone (see Fig. 5).

**C. TARGET EXTRACTION**

As the first step of target tracking, the components of the radar scan corresponding to known land in the ENC map are removed. In contrast to Schuster et al. [27], which removes objects based on whether the center point lies on land or water, we use a polygon-based approach. As an offline processing step, the ENC (or map from QGIS for the Charles River) is converted to polygons for fast online processing.

For each radar scan, nearby radar intensities are first grouped by connected component labeling. These connected components become separate polygons in the binary occupancy grid map. If the radar polygons are verified to intersect the land polygons that we extracted, we consider them to be stationary land objects and merge these radar polygons with the land area polygons. Otherwise, the radar polygon is considered a potential target in the water. All the potential targets are kept for further processing. Fig. 6 shows an example with the red target polygons we extracted as a candidate target and
IV. TARGET TRACKING

Tracking targets by automatic identification systems (AIS) is available on some marine radars, and has been used for ASV applications by Kazimierski and Stateczny [16] and Tetreault [29]. However, in narrow channels, such as the Charles River, it is not able to differentiate between targets in the water area and targets on land. Our tracking algorithm integrated with ENC can achieve this and is simpler, open source, and available for any radar platform.

A. KALMAN FILTER

A Kalman filter [14] is implemented to determine a given target’s position in \(x\) and \(y\), and its velocity along \(x\) and \(y\). In our case, only \(x\) and \(y\) can be measured for the target, which is derived in the previous step described above. A constant velocity model is applied to track the target’s motion.

The state vector is given by

\[
\begin{bmatrix}
    p_x \\
    p_y \\
    v_x \\
    v_y
\end{bmatrix}
\]

where

- \(p_x\) distance to the East in UTM;
- \(p_y\) distance to the North in UTM;
- \(v_x\) speed along the \(x\) (East) axis;
- \(v_y\) speed along the \(y\) (North) axis.
FIGURE 5. Excluding the dead zone. We find the lowest intensity point at the junction of the radar intensities and remove all the occupied grid cells from the start of the transmission to the junction point.

FIGURE 6. Processed radar data. The red polygon is the potential target, and the gray polygons on land are considered stationary land objects. The green rectangle is the Jetyak itself. We note that the radar returns often encompass parts of the river that we conservatively classify as land. (a) Binary occupancy grid. (b) Satellite view.

The initial state noise covariance matrix $P$ is defined as follows:

$$ P = \begin{bmatrix} \sigma_{px}^2 & 0 & 0 & 0 \\ 0 & \sigma_{py}^2 & 0 & 0 \\ 0 & 0 & \sigma_{vx}^2 & 0 \\ 0 & 0 & 0 & \sigma_{vy}^2 \end{bmatrix} $$

The predicted state $x'$ is based on a constant velocity model where the process model is given by

$$ x' = Fx + v. $$

Additional acceleration:

$$ \begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{v}_x \\ \dot{v}_y \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ v_x \\ v_y \end{bmatrix} + \begin{bmatrix} \Delta t^2 \\ \frac{2}{\Delta t^2} \\ \frac{2}{\Delta t} \end{bmatrix} \begin{bmatrix} a_x \\ a_y \end{bmatrix}. $$

We can decompose the noise term $v$ as

$$ v = \begin{bmatrix} \Delta t^2 \\ \frac{2}{\Delta t^2} \\ \frac{2}{\Delta t} \end{bmatrix} \begin{bmatrix} a_x \\ a_y \end{bmatrix} = \begin{bmatrix} \Delta t^2 \\ \frac{2}{\Delta t} \end{bmatrix} \begin{bmatrix} a_x \\ a_y \end{bmatrix} = G a. $$
$Q$ is the process noise covariance matrix, which is the expectation of $v^T v$

$$Q = E[v^T v] = E[Gaa^T G^T].$$

(6)

As $G$ does not contain any random variable we can rewrite this as

$$Q = GE[a a^T] G^T = G \begin{bmatrix} \sigma_{ax}^2 & \sigma_{ax} \sigma_{ay} \\ \sigma_{ax} \sigma_{ay} & \sigma_{ay}^2 \end{bmatrix} G^T = GQvG^T.$$  

(7)

As $a_x$ and $a_y$ have no correlation, which means $\sigma_{ax} \sigma_{ay} = 0$, we can simplify $Qv$

$$Qv = \begin{bmatrix} \sigma_{ax}^2 & 0 \\ 0 & \sigma_{ay}^2 \end{bmatrix}. $$

(8)

Thus, the $Q$ matrix becomes

$$Q = GQvG^T = \begin{bmatrix}
\frac{\Delta t^4}{4}\sigma_{ax}^2 & 0 & \frac{\Delta t^3}{2}\sigma_{ax} & 0 \\
0 & \frac{\Delta t^4}{4}\sigma_{ay}^2 & 0 & \frac{\Delta t^3}{2}\sigma_{ay} \\
\frac{\Delta t^3}{2}\sigma_{ax} & 0 & \Delta t^2\sigma_{ax}^2 & 0 \\
0 & \frac{\Delta t^3}{2}\sigma_{ay} & 0 & \Delta t^2\sigma_{ay}^2
\end{bmatrix}.$$ 

(9)

So that we can compute the prediction state covariance as $P' = FPF^T + Q$.

**Update Stage:** We update the vessel’s position and velocity based on the radar measurement.

The radar measurement $z$ includes position $p_x$ and $p_y$

$$z = \begin{bmatrix} p_x \\ p_y \end{bmatrix}. $$

(10)

The measurement noise covariance matrix $R$ can be defined as follows:

$$R = \begin{bmatrix} \sigma_{px}^2 & 0 \\ 0 & \sigma_{py}^2 \end{bmatrix}. $$

(11)

We also define matrix $H$, which is used to transform state $x$ to measurement $z$

$$\begin{bmatrix} p_x \\ p_y \end{bmatrix} = H \begin{bmatrix} p'_x \\ p'_y \\ v'_x \\ v'_y \end{bmatrix} $$

(12)

where

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. $$

(13)

**B. DATA ASSOCIATION**

Data association is a crucial step in target tracking. When there are multiple targets in consecutive frames, the Kalman filter needs to know how these targets are associated to perform the update step. On the other hand, data association also needs to know the target’s prediction state to maximize the probability of associations across different combinations.

Radar scans may contain noise that can adversely affect target extraction and tracking. To simplify the problem, we make two assumptions. We assume that any object creates only one single measurement (one polygon in the binary occupancy grid), and one measurement can be created only by a single object. These assumptions ignore the case in which one target vessel appears as separate polygons on the radar scan, or multiple close vessels are grouped as one target. These cases do not happen in our dataset but can cause a bad estimation in target tracking.

We run a separate Kalman filter for each target that is identified in the radar scans. The Kalman filter calculates a predicted state estimation for each target for the next time step, including a predicted position and its covariance. We use the elliptical shape to represent the covariance as a gate. If a target in a succeeding frame is in the gate associated with a target’s predicted state in the current frame, then these two targets are considered to be the same. If there are multiple targets.
in the same gate in the next frame, we compute the likelihood of every target in this gate associated with the corresponding target in the original frame. The probability density function (pdf) given by the Kalman filter can be applied to get the probability, which corresponds to the likelihood in this case and can be used for data association. We aim to maximize the sum of all the likelihoods for all the associations. The associations with the highest probability are selected. In practice, this method works extremely well, as illustrated in Fig. 7.

If a new target shows up in a frame, then it would be associated with a “null” target in the previous frame, which leads to a likelihood of zero, so that the new target can be recognized and initialized. The predicted velocity of the new target would be zero, since there are no other measurements available. If a target disappears in a succeeding frame, the target in the current frame would be associated with a “null” target in the next frame.

Associations of targets are checked in two ways to prevent highly improbable results. As a first check, we use the
maximum distance we believe any given target could travel in the time between measurements, range \( = v \cdot t_{\text{meas}} \). In our case, since \( t_{\text{meas}} \approx 2.5 \) s for each radar scan and we are highly unlikely to encounter any vessels on the Charles with \( v > 10 \) m/s, we use range = 25 m. In environments where we do expect to see faster vessels, the range can be increased accordingly. Another check is implemented by restricting the size of the gate. The likelihood computed by the pdf has a threshold that restricts the size of the covariance ellipse. In our case, we use a threshold of \( 10^{-6} \).

Fig. 8 shows the results of our data association algorithm for multiple targets across multiple frames. The red lines in the figure are the trajectories of those targets.

After the data association, measurements can be input into the Kalman filter to run the update functions. Then, the posterior probability of each target is retrieved as the current state of the Kalman filter, including UTM coordinates \( p_x \) and \( p_y \), velocity along each axis \( v_x \) and \( v_y \), and corresponding state covariance \( P \).
Path planning algorithm. We predefine a safe path (step 1) independent of any obstacles. We then set the shorter contour of obstacle polygons as the temporary safe path (step 2). We continue to refine the path by checking each point from the endpoint to the start point, to see whether this point can directly connect to the start point. If a shorter path is detected, we take that as the new path. After checking all the points after the start point, we repeat the procedure with the second point as the start point and check all the points after it. After all the points are checked in this way, the final path is selected (steps 3 and 4).

Sometimes, the target disappears for a single frame but then reappears. Typically, this happens when the target is too close to the coastline and gets lumped as a land object. Two frames are cached to avoid losing track of such targets in such a situation. As there is no measurement in the frame where the target disappears, only the Kalman filter’s prediction step is run.

VI. PATH PLANNING ALGORITHM

Finally, we can account for the COLREGs projections by using a reactive motion planning algorithm based on aspects of the one proposed by Kuwata et al. [18]. The velocity obstacles are represented as polygons in the binary occupancy grid of the environment. To avoid the risk of collision, a set of waypoints is created through the use of visibility graph inspired path planning (VGIPP). These waypoints modify the global path for the Jetyak and are executed to avoid collisions related to COLREGs as illustrated in Fig. 12. Using VGIPP is computationally very efficient for our application.

VGIPP is based on the binary occupancy grid map. All the stationary and dynamic obstacles with their COLREGs projections are represented as occupied polygons on the map. The part of the grid that is not occupied is considered safe for traversal.

First, we predefine a global path for the entire trip. As our river transit involves sections that are relatively narrow in parts, we choose a path that keeps us in the center of the river. This is not the shortest path, but avoids getting the vessel very close to shore.

While transiting down the river, we are continually running VGIPP, as illustrated in Fig. 11. The predefined path may intersect with some obstacle polygons, and we find the contours of these polygons. These contours can, in turn, be split into two sets of polylines. We use the shorter polyline as our temporary optimal path. We then check each point from the endpoint to the start point, to see whether the point under consideration can directly connect to the start point. If a shorter path is detected, we use the path as our new optimal path. After checking all the points after the start point, we repeat the procedure using the second point as our start point and checking all the points after it. After all the points have been checked in this way, the final path emerges.

Fig. 11 shows how VGIPP is implemented.
FIGURE 12. Path planning algorithm results. (a) We predefine a safe path in the middle of the river. (b) We check to see if the path intersects with any obstacles between the start point (Jetyak) and the endpoint (destination). (c) We then get the shorter contour of obstacle polygons as the temporary safe path. (d) Finally, we optimize the temporary path by checking to see whether there is a shorter path in the route.

If the Jetyak is already in an unsafe area, such as the COLREGs projection of a target, it tries to escape that area first. The algorithm finds the nearest safe point and creates a path to it and then continues to find a safe path to the destination.

This algorithm is not guaranteed to yield an optimal result, but it is very efficient in the binary occupancy grid. In our case, the size of the binary occupancy grid map is 4000².

A traditional algorithm could take orders of magnitude longer to calculate the shortest path in such a huge and detailed map for each scan. The VGIPP works efficiently in this case, which takes only around 0.2 s to get the result with Python. In most cases, VGIPP does yield the optimal path. In complex environments, it also guarantees that the safe path can be detected if one exists. We note that if no acceptable path is found, the Jetyak can slow down and come to a complete stop.

VII. CONCLUSION AND FUTURE WORK
In this article, we have presented an end-to-end system that is capable of navigating in constrained waters while
respecting the constraints associated with COLREGs. Our minimal sensor suite consists of GPS, inertial measurement unit (IMU), and radar measurements.

Currently, we can collect data from radar, GPS, IMU, and compass sensors with ROS and replay the data to calculate obstacle-free waypoints that comply with COLREGs. While those waypoints are not executed live on the Jetyak due to safety and permitting concerns, they are calculated in real time with real data and can be used to avoid collision with both dynamic and static obstacles in a complicated environment. We have made all our data freely available as rosbags for researchers to duplicate our results as well as to use the data as a baseline for comparing different algorithms. The dataset is at [12], which also includes the source code for running the algorithms associated with this article.

The processing time of the whole algorithm (including data preprocessing, target extraction, Kalman filtering, data association, etc.) is at [12], which also includes the source code for running the algorithms associated with this article.

**REFERENCES**

[1] M. R. Benjamin and J. A. Curcio, “COLREGs-based navigation of autonomous marine vehicles,” in *Proc. IEEE/OES Auto. Underwater Vehicles*, Jun. 2004, pp. 32–39.

[2] M. R. Benjamin, J. J. Leonard, J. A. Curcio, and P. M. Newman, “A method for protocol-based collision avoidance between autonomous marine surface craft,” *J. Field Robot.*, vol. 23, no. 5, pp. 333–346, May 2006.

[3] M. Blaich, S. Köhler, M. Schuster, T. Schuchhardt, J. Reuter, and T. Tietz, “Mission integrated collision avoidance for USVs using laser range finder,” in *Proc. OCEANS*, May 2015, pp. 1–6.

[4] *Bv24*. Accessed: Apr. 15, 2022. [Online]. Available: https://github.com/juancamilog/bv24

[5] Y. Cheng and W. Zhang, “Concise deep reinforcement learning obstacle avoidance for underactuated unmanned marine vessels,” *Neurocomputing*, vol. 272, pp. 63–73, Jan. 2018.

[6] R. Dubey and S. J. Louis, “VORF-COLREGs: A hybrid velocity obstacles and RRT based COLREGs-compliant path planner for autonomous surface vessels,” in *Proc. OCEANS*, Sep. 2021, pp. 1–8.

[7] B.-O.-H. Eriksen, G. Bitar, M. Breivik, and A. M. Lekkas, “Hybrid collision avoidance for ASVs compliant with COLREGs rules 8 and 13–17,” *Frontiers Robot. AI*, vol. 7, pp. 11, Feb. 2020.

[8] Y. Fan, Z. Sun, and G. Wang, “A novel reinforcement learning collision avoidance algorithm for USVs based on maneuvering characteristics and COLREGs,” *Sensors*, vol. 22, no. 6, p. 2000, Mar. 2022.

[9] *GYM-AUV*. Accessed: Apr. 15, 2022. [Online]. Available: https://github.com/ThomasNLarsen/gym-auv

[10] Y. Hu, X. Meng, Q. Zhang, and G.-K. Park, “A real-time collision avoidance system for autonomous surface vessel using fuzzy logic,” *IEEE Access*, vol. 8, pp. 108835–108846, 2020.

[11] IMO, *COLREG: Convention on the International Regulations for Preventing Collisions at Sea*, Int. Maritime Org., London, U.K., 2003.

[12] *Jetyak Dataset*. (2023). The Jetyak Colreg Dataset and Associated Code. Accessed: Jul. 30, 2023. [Online]. Available: https://github.com/study-is-happy/Jetyak

[13] T. A. Johansen, T. Perez, and A. Cristofaro, “Ship collision avoidance and COLREGs compliance using simulation-based control behavior selection with predictive hazard assessment,” *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 12, pp. 3407–3422, Dec. 2016.

[14] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, Mar. 1960.

[15] Y. T. Kang, W. J. Chen, D. Q. Zhu, J. H. Wang, and Q. M. Xie, “Collision avoidance path planning for ships by particle swarm optimization,” *J. Mar. Sci. Technol.*, vol. 26, no. 6, p. 3, 2018.

[16] W. Kazimierski and A. Stateczny, “Radar and automatic identification system track fusion in an electronic chart display and information system,” *J. Navigat.*, vol. 68, no. 6, pp. 1141–1154, Nov. 2015.

[17] P. Kimball et al., “The WHOI jetyak: An autonomous surface vehicle for oceanographic research in shallow or dangerous waters,” in *Proc. IEEE/OES Auto. Underwater Vehicles (AUV)*, Oct. 2014, pp. 1–7.

[18] Y. Kuwata, M. T. Wolf, D. Zarzhitsky, and T. L. Huntsberger, “Safe maritime autonomous navigation with COLREGs, using velocity obstacles,” *IEEE J. Ocean. Eng.*, vol. 39, no. 1, pp. 110–119, Jan. 2014.

[19] Y. Liu and R. Bucknall, “Path planning algorithm for unmanned surface vehicle formations in a practical maritime environment,” *Ocean Eng.*, vol. 97, pp. 126–144, Mar. 2015.

[20] H. Lyu and Y. Yin, “COLREGs-constrained real-time path planning for autonomous ships using modified artificial potential fields,” *J. Navigat.*, vol. 72, no. 3, pp. 588–608, May 2019.

[21] *Mavlink*. Accessed: Apr. 15, 2022. [Online]. Available: https://github.com/mavlink/mavlink

[22] *Mavros*. Accessed: Apr. 15, 2022. [Online]. Available: https://github.com/mavlink/mavros

[23] J. A. García Mazaa and R. P. Argüelles, “COLREGs and their application in collision avoidance algorithms: A critical analysis,” *Ocean Eng.*, vol. 261, Oct. 2022, Art. no. 11029.

[24] W. Næem, G. W. Irwin, and A. Yang, “COLREGs-based collision avoidance strategies for unmanned surface vehicles,” *Mechatronics*, vol. 22, no. 6, pp. 669–678, Sep. 2012.

[25] H. Namgung and J.-S. Kim, “Collision risk inference system for maritime autonomous surface ships using COLREGs rules compliant collision avoidance,” *IEEE Access*, vol. 9, pp. 7823–7835, 2021.

[26] *QGIS*. Accessed: Apr. 15, 2022. [Online]. Available: https://qgis.org/en/site

[27] M. Schuster, M. Blaich, and J. Reuter, “Collision avoidance for vessels using a low-cost radar sensor,” *IFAC Proc. Volatiles*, vol. 47, no. 3, pp. 9673–9678, 2014.

[28] E. Serigstad, B.-O.-H. Eriksen, and M. Breivik, “Hybrid collision avoidance for autonomous surface vessels,” *IFAC-PapersOnLine*, vol. 51, no. 29, pp. 1–7, 2018.

[29] B. Tetreault, “Use of the automatic identification system (AIS) for maritime domain awareness (MDA),” in *Proc. OCEANS*, vol. 2, 2005, pp. 1590–1594.

[30] *Vectornav*. Accessed: Apr. 15, 2022. [Online]. Available: https://github.com/dawonn/vectornav

[31] K. Woerner, M. R. Benjamin, M. Novitzky, and J. J. Leonard, “Quantifying protocol evaluation for autonomous collision avoidance,” *Auto. Robots*, vol. 43, no. 4, pp. 967–991, Apr. 2019.

[32] K. Wröbel, M. Gil, Y. Huang, and R. Wawruch, “The vagueness of COLREG versus collision avoidance techniques—A discussion on the current state and future challenges concerning the operation of autonomous ships,” *Sustainability*, vol. 14, no. 24, p. 16515, Dec. 2022.

[33] X. Xu, P. Cai, Z. Ahmed, V. S. Yellapu, and W. Zhang, “Path planning and dynamic collision avoidance algorithm under COLREGs via deep reinforcement learning,” *Neurocomputing*, vol. 468, pp. 181–197, Jan. 2022.

[34] X. Xu, Y. Lu, X. Liu, and W. Zhang, “Intelligent collision avoidance algorithms for USVs via deep reinforcement learning under COLREGs,” *Ocean Eng.*, vol. 217, Dec. 2020, Art. no. 107704.

[35] J. Zhang, D. Zhang, X. Yan, S. Haugen, and C. G. Soares, “A distributed anti-collision decision support formulation in multi-ship encounter situations under COLREGs,” *Ocean Eng.*, vol. 191, pp. 336–348, Sep. 2015.

[36] L. Zhao and M.-I. Roh, “COLREGs-compliant multiship collision avoidance based on deep reinforcement learning,” *Ocean Eng.*, vol. 191, Nov. 2019, Art. no. 106436.

[37] Y.-Y. Zhou, J.-J. Huang, F.-W. Wang, Z.-L. Wu, and Z.-J. Liu, “A study of the application barriers to the use of autonomous ships posed by the good seamanship requirement of COLREGs,” *J. Navigat.*, vol. 73, no. 3, pp. 710–725, May 2020.