TOWARDS A COLREGS COMPLIANT AUTONOMOUS SURFACE VESSEL

A Thesis Presented

By

James Connor Meyers

To

The Department of Electrical and Computer Engineering

in partial fulfillment of the requirements for the degree of

Master of Science

in the field of

Electrical and Computer Engineering

at

Northeastern University

Boston, Massachusetts

May 2019
Abstract

Bringing about Autonomous Surface Vessel (ASV) autonomy faces two major challenges in the standard robot navigation pipeline – mapping the world and making decisions based upon that map; for the most part, ASV state estimation and vehicle control may be assumed due to the reliable presence of GPS on the ocean and the scale on which decisions are currently made. Numerous approaches have been taken at solving these problems, yet few full systems have been demonstrated in complex live harbor environments. In this paper we describe a full system for navigating these environments using the WHOI Jetyak.

The Jetyak is equipped with a broadband marine radar and Inertial Navigation System (INS) in order to perform static and dynamic obstacle detection. Once detected, individual measurements are combined over time using data association to identify and track specific vehicles. Most radar manufacturers come with this capability through automatic radar plotting aids (ARPA), however, these are typically not exposed to the user for work in robotics. We use a common pipeline of data association between scans using a Joint Probabilistic Data Association (JPDA) algorithm to give unique ids to boats, and an Unscented Kalman Filter (UKF) to track and predict their motion.

With a projection of where other vehicles in the environment are and a prediction of where they are going, the ASV must make decisions on what actions to take to avoid other boats. While this problem is similar to the common 2D+Time obstacle avoidance problem often seen in terrestrial robotics, it has the added element that all boats must act under the constraints of COLREGs. Similar to the rules of driving, COLREGs defines which boat has the right-of-way, and what actions the non-right-of-way boat must perform. We account for these rules using a reactive path planning algorithm based on the one proposed by Kuwata et al., that incorporates the rule based constraints of COLREGS into a velocity obstacles algorithm. In addition we test an implementation of incorporating COLREGS into a global motion planner to provide better performance in complex environments.

Thesis Supervisor: Hanumant Singh
Professor, Northeastern University
Acknowledgements

I would like to express my sincere gratitude to my advisor, Professor Hanumant Singh, for the continuous support of my Master’s study and research, his patience, his motivation, his enthusiasm, and his immense knowledge. I could not have asked for a better advisor and mentor for my studies. I would also like to thank the rest of my thesis committee: Professor Taskin Padir and Professor Lawson Wong, for their insightful comments and hard questions.

For their constant help, particularly in hours spent transporting and testing the Jetyak, I’d like to thank the members of the Field Robotics Lab - Mithun Diddi, Vikrant Shah, Jagatpreet Singh, Pushyami Kaveti, Tarun Uday, Yang Liu, and Sahil Ghangas. Thanks to Rick Schroeder for his willingness to house our Jetyak at the Northeastern crew house and for all his help and to Kevin Manganini at WHOI for his Jetyak troubleshooting expertise.

A very special thanks to Zhiyong Zhang for helping me tackle the obstacle detection portion of this thesis, the project would not be complete if not for his hard work and dedication.

Lastly, I would like to thank my family and friends who have been part of my journey and supported me throughout my academic career. In particular I’d like to thank my wife Nicole for always loving me and supporting my goals no matter what.
Towards A COLREGS Compliant Autonomous Surface Vessel

Contents

List of Figures 4

List of Tables 6

1. Introduction 7
   1.1. Navigation Rules (COLREGS) 7
   1.2. Autonomous Surface Vessels (ASVs) 8
   1.3. Sensing for COLREGS 9
   1.4 Closest Point of Approach (CPA) 12
   1.5 Current Solutions 16

2. Test Surface Vessel 19
   2.1. WHOI Jetyak 19
   2.2. Radar 21
   2.3. Inertial Navigation System 22
   2.4. Software 23

3. Simulation Environment 25
   3.1. Matlab Mobile Robotics Simulation 25
   3.2. Binary Occupancy Grids 27
   3.3. Target Tracking 29
   3.4. COLREGS Situations 30
   3.5. Velocity Obstacles 31
   3.6 Probabilistic Roadmaps 34
   3.7 Results 36

4. Data Collection and Analysis 40
   4.1. Environmental Conditions 40
   4.2. Shore Exclusion 42
   4.3. Target Identification and Tracking 43
   4.4. COLREGS Velocity Obstacles 46
   4.5 Path Planning 47
   4.6 Results 51

5. Conclusions and Future Work 56

6. References 58
## List of Figures

<table>
<thead>
<tr>
<th>Fig. No</th>
<th>Title</th>
<th>pg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fig. 1</td>
<td>COLREGS situations between power driven vessels insight of each other.</td>
<td>8</td>
</tr>
<tr>
<td>Fig. 2</td>
<td>NOAA ENC Chart for Woods Hole, MA US5MA28M</td>
<td>10</td>
</tr>
<tr>
<td>Fig. 3</td>
<td>Lowrance Broadband 3G Radar</td>
<td>11</td>
</tr>
<tr>
<td>Fig. 4</td>
<td>COLREGS situations based on difference between USV heading $\theta_a$ and Target Vessel heading $\theta_b$. ( \text{diff} = \theta_a - \theta_b )</td>
<td>14</td>
</tr>
<tr>
<td>Fig. 5</td>
<td>WHOI Jetyak equipped with Lowrance Broadband Radar and Vectornav VN-300 INS</td>
<td>21</td>
</tr>
<tr>
<td>Fig. 6</td>
<td>Vectornav VN-300 Dual GPS Inertial Navigation System</td>
<td>23</td>
</tr>
<tr>
<td>Fig. 7</td>
<td>Poor radar chart overlay with inaccurate heading vs. Radar Chart Overlay with extremely accurate heading via VN-300 INS</td>
<td>23</td>
</tr>
<tr>
<td>Fig. 8</td>
<td>WHOI Jetyak equipped with Lowrance Broadband Radar and Mobile Robotics Simulation Toolbox example simulation</td>
<td>26</td>
</tr>
<tr>
<td>Fig. 9</td>
<td>Simple Binary Occupancy Grid (BOG)</td>
<td>28</td>
</tr>
<tr>
<td>Fig. 10</td>
<td>Satellite Imagery of Charles River generated using Google Earth</td>
<td>28</td>
</tr>
<tr>
<td>Fig. 11</td>
<td>Binary Occupancy Grid (BOG) as a .png file of Charles River created using satellite imagery</td>
<td>29</td>
</tr>
<tr>
<td>Fig. 12</td>
<td>Velocity Obstacles for Target Vessels with added uncertainty</td>
<td>32</td>
</tr>
<tr>
<td>Fig. 13</td>
<td>Velocity Obstacles for applicable COLREGS situations</td>
<td>33</td>
</tr>
<tr>
<td>Fig. 14</td>
<td>Simple Example of Probabilistic Roadmap computing an obstacle free path</td>
<td>35</td>
</tr>
<tr>
<td>Fig. 15</td>
<td>Probabilistic Roadmap for Jetyak Simulation based on Charles River Map</td>
<td>36</td>
</tr>
<tr>
<td>Fig. 16</td>
<td>Simulation Results of COLREGS compliant navigation. Left: Targets in map (1 is Jetyak) with waypoints via PRM Right: VOs + COLREGS VOs where applicable at same time as image on left.</td>
<td>38</td>
</tr>
<tr>
<td>Fig. 17</td>
<td>Google Earth imagery of 8km (4.2 NM) transit from Northeastern boathouse to MIT boathouse on Charles River, Boston, MA</td>
<td>40</td>
</tr>
<tr>
<td>Fig. 18</td>
<td>Chase Boat used for following Jetyak and for Target Identification and Tracking and COLREGS avoidance testing.</td>
<td>40</td>
</tr>
<tr>
<td>Fig. 19</td>
<td>Google Earth imagery of Boston University Bridge on Charles River (Left) GoPro image taken from Jetyak navigating autonomously through Eliott Bridge on Charles River (Right)</td>
<td>44</td>
</tr>
</tbody>
</table>
Fig. 20 Joint Probabilistic Data Association (JPDA) overview
Fig. 21 Visibility Graph Inspired Path Planning Stages
Fig. 22 Visibility Graph Inspired Path Planning applied to a Head On situation with COLREGS VO
Fig. 23 Jetyak Being Overtaken (No Risk Of Collision or COLREGS VO)
Fig. 24 Jetyak Overtaking Target Vessel (Risk of Collision Exists, Small VO applied, no trackline modification necessary)
Fig. 24 Tracking Performance of Target Vessel via UKF and JPDA
Fig. 26 Jetyak Entering First Bridge. Green represents “navigable lane” through center of bridge.
Fig. 27 Jetyak in Head On Situation with target vessel (Risk Of Collision Exists, Head On VO applied, path appropriately modified to avoid collision)
Fig. 28 Jetyak Entering Second Bridge. Green represents “navigable lane” through center of bridge.
Fig. 29 Jetyak Entering Third Bridge. Green represents “navigable lane” through center of bridge.
List of Tables

<table>
<thead>
<tr>
<th>Table No</th>
<th>Title</th>
<th>Pg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1</td>
<td>COLREGS avoidance methods compared.</td>
<td>18</td>
</tr>
<tr>
<td>Table 2</td>
<td>COLREGS Velocity Obstacle parameters</td>
<td>33</td>
</tr>
</tbody>
</table>

1. Introduction

1.1. Navigation Rules (COLREGS)

The International Regulations for Preventing Collisions at Sea (COLREGS) [1] were first promulgated in 1972. They are published by the International Maritime Organization (IMO) and set out, among other things, the "rules of the road" or navigation rules to be followed by ships and other vessels at sea to prevent collisions between two or more vessels.

Currently, there is very little definitive regulation on proper operating procedures for Autonomous Surface Vessels (ASVs), but there has been some guidance put out in the form of Navigation Safety Advisory Council (NAVSAC) Resolution 16-01 [2]. It does not directly mention COLREGs, but leaves it up to the operator to “Operate the UMS as responsible members of the maritime community.” and states the importance for Unmanned Maritime Systems (UMS) users to “Develop industry standards amongst similar types of UMS.” This leaves it up to the operator to develop procedures to safely operate the vessel.

One way to attempt safe navigation for USVs is to emulate human behavior as closely as possible through use of COLREGS. The rules of the road contain sections regarding responsibilities of vessels in restricted visibility (fog, rain, etc.), and hierarchy of vessels to determine right-of-way (i.e. Sailing Vessels have the right of way over Power Driven Vessels), for the purposes of this thesis
we only tackle the situations represented in Fig. 1. These represent the responsibilities of power driven vessels in sight of one another to prevent collision. See Ref. [1] for a more detailed description of responsibilities between vessels.

Fig. 1. COLREGS situations between power driven vessels insight of each other.

1.2. Autonomous Surface Vessels (ASVs)

While there are many issues inherent with ASVs that prevent them from being fully compliant with the rules of the road, COLREGs can serve as a guideline for ASVs to avoid collision. Using nautical charts and reliable positioning and control systems, ASVs can safely avoid static targets or running aground. For the dynamic aspects of the maritime environment such as vessels, debris, drift of buoys, etc., an obstacle detection method must be in place. These could be laser
scanner or computer vision systems, but discussed in more detail below, these systems generally suffer under the often encountered foggy conditions on the water. More commonly used and explicitly discussed in COLREGs for collision avoidance is the use of radar and Automatic Radar Plotting Aids (ARPA).

1.3. Sensing for COLREGS

The first element of our mapping system is the use of marine charts, which provide accurate descriptions of land, water depths, navigational aids, and features useful to the safe navigation of a vessel. An example of an Electronic Navigation Chart (ENC) of Woods Hole, MA generated by the National Oceanic and Atmospheric Administration (NOAA) [3] is shown in Fig. 2 below. Because marine charts provide a good description of the large scale static environment, the challenge for ASV mapping systems is detecting other boats in the environment and predicting their future motion. Various detection systems have been proposed based upon computer vision or laser scanners (mimicking approaches commonly used in terrestrial robotics) or radar (mimicking the sensors commonly used on human controlled surface vessels). While computer vision and laser scanners are well suited for robotic obstacle detection, they suffer in the foggy conditions which often come up unexpectedly on the ocean.
Standard radar systems are high powered and have a large dead-zone making them ill suited for small ASV operation. Instead, we use the Lowrance 3G Broadband radar system [4] (Fig. 3) which is low power (18 W) and has a dead-zone of only a few meters around the vehicle. Thanks to the work of [5] the raw data from the radar may be accessed, and has been demonstrated in [6]. Radars provide range and bearing measurements of other obstacles, and are well studied in a marine environment, working reliably in weather and in darkness.
Once detected, individual measurements must be combined over time using data association to identify and track specific vehicles. Most radar manufacturers come with this capability through automatic radar plotting aids (ARPA), however, these are typically not exposed to the user for work in robotics. We use a common pipeline of data association between scans using a Joint Probabilistic Data Association (JPDA) algorithm to give unique ids to boats, and an Unscented Kalman Filter (UKF) to track and predict their motion.

With a projection of where other vehicles in the environment are and a prediction of where they are going, the ASV must make decisions on what actions to take to avoid other boats. While this problem is similar to the common 2D+Time obstacle avoidance problem often seen in terrestrial robotics, it has the added element that all boats must act under the constraints of COLREGs. Similar to the
rules of driving, COLREGs defines which boat has the right-of-way, and what actions the non-right-of-way boat must perform. We account for these rules using a reactive motion planning algorithm based on the one proposed by Kuwata et al. [7], that incorporates the rule based constraints of COLREGs into a velocity obstacles algorithm. In addition we test an implementation of incorporating COLREGS into a global motion planner to provide better performance in complex environments.

1.4 Closest Point of Approach (CPA)

One of the most important decision making factors for collision avoidance is Closest Point of Approach (CPA). In its simplest form, CPA can be determined by two or more relative radar bearings and ranges to a target over time. The difference between these two bearings and ranges, in addition to our vessel’s course and speed, can be used to determine the target’s course and speed, and the time, bearing, and range of CPA. By reducing the radar data into four variables, $v_a$ as the velocity vector of our vessel, $v_b$ as the velocity vector of the target vessel, $p_a$ as the position vector of our vessel, and $p_b$ as the position vector of the target vessel, the time until CPA or $t_{CPA}$ can be calculated as follows.

$$t_{CPA} = \begin{cases} 0, & \text{if } \|v_a - v_b\| \leq \varepsilon \\ \frac{(p_b - p_a) \cdot (v_a - v_b)}{\|v_a - v_b\|^2}, & \text{otherwise} \end{cases}$$
Range of CPA ($r_{CPA}$) can also be computed to determine how close the target vessel will be at CPA. Using $t_{CPA}$ and our variables $v_a, v_b, p_a$, and $p_b$ we compute $r_{CPA}$ as follows.

$$r_{CPA} = \| (p_a + v_a t_{CPA}) - (p_b + v_b t_{CPA}) \|$$

Range of CPA ($r_{CPA}$) determines how close the vessel will be at its closest point, bearing of CPA ($b_{CPA}$) determines where the vessel will be relative to our own vessel at CPA, and time of CPA ($t_{CPA}$) determines when the closest point of approach is predicted to be. A negative $t_{CPA}$ indicates that CPA has already occurred and the target vessel is moving away from our vessel. Combining target courses and speeds with calculated $r_{CPA}$, $t_{CPA}$, and $b_{CPA}$, we can make decisions to maneuver in accordance with the rules of the road. See Fig. 1 above for simple COLREGS situations and [8,11] and [9,14] for further discussion of the implications of COLREGS for autonomous vehicles.

The main situations we are examining are those depicted above in Fig. 1, which represent actions to be taken by power driven vessels insight of each other. These rules do not apply to vessels of different type (sail boats, fishing vessels, etc) or under restricted visibility. It should be noted that situations regarding hierarchy (i.e. sailboats have the right of way over power driven vessels) are not considered. This would require an augmentation of the detection and tracking system to include vessel type identification which is beyond the scope of this project.
Towards A COLREGS Compliant Autonomous Surface Vessel

In order to determine what actions need to be taken, the applicable COLREGS situation must first be determined. This is done by using the \( r_{CPA} \) and \( t_{CPA} \) previously computed and the course of our vessel and the target vessel \( \theta_a \) and \( \theta_b \). See the matlab function below for an understanding of how these situations are computed. Note that the rangeThreshold and timeThreshold parameters are tuneable to best fit any given USV, operating environment, level of comfort with CPA, etc. The threshold values set below are used in our Matlab simulation environment, threshold values for real world obstacle avoidance are discussed in more detail later. See Fig. 4 below as well for a representation of how a given COLREGS situation is applied based on the Jetyak’s course \( \theta_a \) and the target vessel’s course \( \theta_b \).
Towards A COLREGS Compliant Autonomous Surface Vessel

Fig. 4. COLREGS situations based on difference between USV heading $\theta_a$ and Target Vessel heading $\theta_b$. $\text{diff} = \theta_a - \theta_b$

function [situation] = colregsSituation(theta,thetab,rcpa,tcpa)

%Set Risk Of Collision (R.O.C.) thresholds
rangeThreshold = 50; %any vessel with rcpa < x meters has possible R.O.C.
timeThreshold = 60; %any vessel with tcpa < x seconds has possible R.O.C.

%we use both rcpa and tcpa as we could have a rcpa of 10m, but it
%doesn't occur for another hour, therefore R.O.C. doesn't exist yet

%Evaluate if Risk Of Collision exists
if rcpa < rangeThreshold && tcpa < timeThreshold && tcpa > 0
    riskOfCollision = 1;
else
    riskOfCollision = 0;
end

%Evaluate relative courses between our vessel and target vessel
diff = wrapTo360(rad2deg(theta-thetab));

if riskOfCollision == 1 %Only necessary to check COLREGS if R.O.C. possible
    %Evaluate which situation applies
    if 165 <= diff && diff <= 195
        situation = 1; %Head-On
end
elseif 195 <= diff && diff <= 292.5
    situation = 2; %Crossing situation (we are give-way vessel)
elseif 292.5 <= diff || diff <= 67.5
    situation = 3; %Overtaking situation
elseif 67.5 <= diff && diff <= 165
    situation = 4; %Crossing situation (we are stand-on vessel)
else
    situation = -1; %no situation applies
end
else
    situation = 0;
end

1.5 Current Solutions

While many currently existing solutions for COLREGS compliant autonomous surface vessels exist, many of them are missing one element that the others contain. There are no solutions available that use open source hardware and software with relatively low cost sensors that perform COLREGS avoidance for both stationary and dynamic obstacles in congested waterways. There are expensive military applications such as the CUSV created by the Navy that operate under dynamic environments, but the research is highly proprietary and
the sensor suites utilized are extremely expensive [10]. Other methods using lower cost sensors and vessels and some perform well in constrained environments [11, 12, 13]. However, they typically navigate in open waters or directly share the positions of other vessels using GPS broadcasting of all vessel positions. While using an Automatic Identification System (AIS) is often proposed as an option for determining vessel position, course, and speed, most small or recreational vessels are not equipped with AIS or methods to broadcast or receive their position between other vessels.

The method proposed here tackles all of these problems. By using the WHOI Jetyak [15] and open source hardware and software, costs are kept low compared to Navy and professional grade sensor suites without significantly degraded performance. By using both freely available ENC charts and generating Velocity Obstacles in a Binary Occupancy Grid of the environment, the Jetyak is able to not only avoid other vessels in accordance with COLREGS, but it able to do so in an environment constrained by land areas, shoal water, bridges, and other navigational hazards. The added benefit of incorporating ENC charts into the path planning pipeline is the ease of varying which navigational hazards to avoid based on vessel requirements. As an example, a deep draft vessel entering a port will have significantly less navigable area. By marking areas below a certain depth as “occupied” in the binary occupancy grid in conjunction with the COLREG velocity obstacles, a USV can follow COLREGS while navigating in constrained environments.
Towards A COLREGS Compliant Autonomous Surface Vessel

Often the papers representing COLREGS avoidance methods primarily utilize simulation as a method of demonstrating capabilities. While this is an important step in determining optimal COLREGS path planning tools, and was used extensively in this thesis, the true benchmark should be real world performance of USVs.

Particularly successful methods of obstacle avoidance to date have used velocity obstacles with rule based constraints [7] and used interval programming to optimize over a multi-objective function with both goal points and COLREGs compliant objectives [9, 16]. Both of these methods take advantage of COLREGs natural enforcement in velocity space, working in a reactive manner to adjust a globally computed path with local deviations to avoid other vessels according to COLREGs. While there has been numerous other efforts to incorporate COLREGs within a path planner, see [12, 13], a planner that can account for COLREGs in a multi-vehicle environment with static obstacles has yet to be effectively demonstrated until now.

<table>
<thead>
<tr>
<th>Solution</th>
<th>Obstacle Detection</th>
<th>Cost</th>
<th>Navigation</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kuwata [7]</td>
<td>JPL Sensor Suite</td>
<td>Medium</td>
<td>Optimal Trajectory</td>
<td>Open Water Only</td>
</tr>
<tr>
<td>Michael Benjamin [12]</td>
<td>Position Sharing via GPS broadcast</td>
<td>Very Low</td>
<td>Optimal trajectory</td>
<td>No account for unshared vessel positions</td>
</tr>
<tr>
<td>Navy CUSV [10]</td>
<td>Advanced RADAR, camera, and/or LIDAR systems</td>
<td>Very High</td>
<td>N/A</td>
<td>Expensive and Proprietary</td>
</tr>
<tr>
<td>Mobile</td>
<td>Radar Detection and Tracking via low cost sensor, open source software</td>
<td>Low</td>
<td>Optimal Trajectory</td>
<td>Limited Testing, Open Water, Single Target</td>
</tr>
<tr>
<td>--------</td>
<td>-------------------------------------------------------------------</td>
<td>-----</td>
<td>--------------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>WHOI Jetyak [15]</td>
<td>Radar Detection and Tracking via low cost sensor, open source software</td>
<td>Very Low</td>
<td>ENC, waypoints</td>
<td>None of the above limitations</td>
</tr>
</tbody>
</table>

Table 1. COLREGS avoidance methods compared. [7, 10, 11, 12, 15]

As a disclaimer before the presented work, we can attempt to create a vessel that mimics human behavior according to the rules of the road, but as they are written now, and until further guidance is promulgated, it is not possible for a fully autonomous ASV to navigate waters in accordance with COLREGS.

2. Test Surface Vessel

2.1. WHOI Jetyak

The WHOI Jetyak (Fig. 5) is a Mokai gasoline powered kayak [17] propelled and steered by water-stream. It is 11 feet long, weighs 165 pounds, draws 3 inches, and has a payload of 360 pounds. It utilizes two servos connected to a flight controller PixHawk Ardupilot [18] board to control thrust and steering. The Pixhawk is a modified Arduino board with professional grade open-source unmanned vehicle software. Communication capabilities include a long range RFD900x [19]
for telemetry between the Pixhawk and Mission Planner software [20] and a FrSky receiver [21] connected to the Pixhawk that wirelessly connects with a Tarranis [22] transmitter to drive the Jetyak remotely, switch between auto and manual, and utilize remote starter and emergency kill switches. There is also a Ubiquiti Rocket Long Range Wifi antenna [23] on the Jetyak that enables us to access the Jetyak computer wirelessly via Secure Shell (SSH) [24] to monitor data acquisition and ROS. A GPS module is mounted on the bow of the Jetyak and connected to the Pixhawk, which has an on-board IMU/compass. Both are used in determining the vessel’s position, course, and speed.

The Jetyak is configured to allow for a reconfigurable science payload, in this case, it is equipped with a Broadband Lowrance 3G Radar [4] connected via Ethernet to a small on-board Intel NUC computer [37] that automatically starts upon powering on the Jetyak batteries and immediately begins collecting radar data, Micro Aerial Vehicle Link (MAVLink) [38] data via USB connection to the Pixhawk. A VectorNav VN-300 dual GPS Inertial Navigation System (INS) is also used in order to provide extremely accurate heading information at 100Hz. This is necessary in order to offset the Jetyak’s position and heading from radar scanlines with a degree of accuracy necessary to perform chart overlay. A small amount of heading error can cause a large amount of radar chart overlay error.
Data acquisition is accomplished with the Robot Operating System (ROS) discussed in more detail later. Without obstacle detection via radar, the Jetyak cannot recognize or avoid hazards or obstacles. For this reason, it must be closely supervised while on a mission, particularly when other vessels are in the area. The radar however can be used to identify potential hazards and integrate their relative positions and velocities into the mission planning and decision making process.

2.2. Radar

While Automatic Radar Plotting Aids (ARPA) aren’t commonly available for use with robotics, a significant amount of work has been done on radar tracking [16], forming the underpinnings of ARPA, which may be implemented by robotic vehicles. Schuster et. al. [6] show an effective pipeline using the Joint Probabilistic Data Association (JPDA) technique to associate measurements with specific tracks, and an interacting multiple model (IMM) Kalman filter to track and predict
the motion of each vessel. Elements of this method are incorporated into the target detection and tracking including JPDA and use of an Unscented Kalman Filter (UKF). The output is a list of target vessels, providing target positions and velocities both relative to the Jetyak and represented globally in a map of the environment.

The Broadband Lowrance 3G radar system utilized is an imaging sensor which provides a 360 degree echo image of the environment every 2.5s. The image consists of 2048 range scanlines with each scanline covering approximately 0.2 degrees, but with an opening beam width of 5.2 degrees (3dB). Each scanline is split in 512 resolution cells, whose occupancy is indicated by a value from 0-255. The maximum range is configurable from 50m up to 24NM. For the collision avoidance in a harbor, we focused our work on a maximum range of about 100m, and 100m-250m for operation on the Charles River in Boston, MA.

2.3. Inertial Navigation System

Early in the data collection process, it was determined that the available heading sensor of the Pixhawk was not accurate enough to offset the Jetyak’s heading and position from the radar to accurately overlay radar data on a nautical chart. In order to rectify this, the Vectornav VN-300 [25] was utilized. The VN-300 shown in Fig. 6 is a dual-GPS Inertial Navigation System (INS) that delivers heading and position data at 100 Hz with a heading accuracy of < 0.2° RMS. The comparative results between the two are shown in the images below (Fig. 7). In
the future this INS system will be used to more robustly navigate under bridges and areas with temporarily poor GPS reception.

Fig 6. Vectornav VN-300 Dual GPS Inertial Navigation System [25]

Fig 7. Poor radar chart overlay with inaccurate heading vs. Radar Chart Overlay with extremely accurate heading via VN-300 INS

2.4. Software

The Robotic Operating System or ROS is a flexible framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms [26]. In this case, it is used for data acquisition and is used to make high level decisions based on position, heading, course, speed, and
radar data, to alter course and speed to avoid collision in accordance with COLREGs while maintaining a desired trackline within a certain safety threshold.

The data acquisition is accomplished with ROS and use of the BR24 library [27] to collect radar data, MavRos [28] to collect and broadcast MavLink messages including waypoints and vehicle information, and the VectorNav library [29] to collect heading information from the VN-300 INS. Currently “rosbags” or files of data are split into one minute intervals to be analyzed in post-processing via Matlab and python. ROS “topics” and “messages” received include approximately 800 radar scanlines per second, each scanline containing 512 ranges with intensity values from 0-255. The scan rate of the radar is one full rotation every 2.5 seconds (one full 360 degree scan). Extremely accurate heading information is delivered at 100Hz from the VN-300 INS with errors in the order of approximately 0.2 degrees RMS. All Mavlink messages are updated at 10Hz, with GPS updating at ~5Hz.

Electronic Navigation Charts (ENC) [3] are used to overlay radar data and perform avoidance of the shoreline, other navigational hazards, and ultimately for avoiding dynamic and static obstacles. This is done by importing the ENC chart data into Python and overlaying received radar data with the ENC chart. Where ENC charts are not produced by NOAA, open source QGIS [30] software was used to manually create approximations of land area using satellite imagery.

The radar chart overlay is completed by offsetting the radar scanline angle from the vehicles own heading and range offset from the vessels current position.
All position data is computed in Universal Transverse Mercator coordinates (UTM) \[31\] in order to consistently use meters instead of degrees of latitude and longitude. Details about shore exclusion and target identification and tracking are described in more detail later. Based on the target tracking results, Velocity Obstacles are generated from target vessel positions, courses, and speeds. These Velocity Obstacles incorporate the applicable COLREGS situation in order to occupy space within the ENC chart. All occupied space is determined to be unnavigable, which leaves our path planning algorithm to detect the optimal path between the vessel’s current position and desired end goal as a set of waypoints. Finally, this set of waypoints can be published via ROS to the Pixhawk and executed by the Jetyak. This enables the Jetyak to perform real-time obstacle detection and avoidance in accordance with COLREGS in a congested waterway.

3. Simulation Environment

3.1. Matlab Mobile Robotics Simulation

Real world results of any COLREGS compliant path planner should be the true benchmark of its performance, but simulation environments offer low cost opportunities to test path planning algorithms. Our simulation environment is built in MATLAB and allows us to create a range of scenarios, from the simplest stationary obstacle avoidance, to the most complex environment based on real
Towards A COLREGS Compliant Autonomous Surface Vessel

world ENC charts converted to binary occupancy grids with multiple dynamic and static obstacles to test robustness.

The core of this simulation was based on the the Mobile Robotics Simulation Toolbox [32] created by Mathworks. This toolbox provides utilities for single-robot and multi-robot simulation and algorithm development including 2D kinematic models for robot geometries, configurable lidar and object detector simulators, visualization of robotic vehicles and sensors in occupancy grid maps, and MATLAB and Simulink examples and documentation. A simple example of a multi-robot environment is shown in Fig. 8 below.

![Multi-Robot Visualization](image)

Fig. 8. Mobile Robotics Simulation Toolbox example simulation [32]

We used this environment to emulate both the Jetyak and other target vessels on the water. The radar target identification and tracking was modeled through use of a simulated lidar and robot detector, and binary occupancy grids
were created to scale based on satellite imagery of the intended real-world environments such as Woods Hole Harbor and the Charles River in Boston, MA.

By using this simulation environment, we successfully implemented a path planning system for a USV that performed target identification and tracking in order to perform obstacle avoidance in accordance with COLREGS. This was done in a dynamic environment with multiple static and moving targets in a map representing a real-world operating environment with vessel dynamics comparable to real world vessels. These results were used to create the path planning system implemented on the WHOI Jetyak to perform COLREGS compliant navigation on the Charles River.

3.2 Binary Occupancy Grids

Binary Occupancy Grids (BOGs) were used to represent space occupied by known navigational hazards (i.e. land, bridges, etc) as well as target vessels. Any area that is black is considered occupied in the map, and any area that is white is considered open space. Based on the occupied and unoccupied space, a path planner can be used to determine an obstacle free path.

In addition to using simple BOGs, satellite imagery of the Charles River was used to produce a map of our intended real-world operating environment. This was done by simply creating a .png image with areas marked as black to represent navigational hazards and white as navigable area scaled appropriately. See Fig. 9 below for a representation of a simple BOG and Fig. 10 and Fig. 11 for
the Charles River BOG and associated satellite imagery. For more details on Occupancy Grids, see Ref. [33]

![Fig. 9. Simple Binary Occupancy Grid (BOG) [33]](image)

![Fig. 10. Satellite Imagery of Charles River generated using Google Earth [34]](image)
3.3 Target Tracking

In order to simulate the radar used on the Jetyak, a simulated lidar was used for obstacle detection. There are inherent differences between lidar and radar that make this an imperfect representation of the radar data, but the lidar can be used as an approximation of how we can perform tracking with the radar. We also utilized a “robot detector” within the simulation environment to provide precise pose information about nearby robots within the simulation. This was used to perform target tracking and serves as ground truth position data for each target detected.
Towards A COLREGS Compliant Autonomous Surface Vessel

Given two positions over time, we are able to compute the target’s course and speed as well as $r_{CPA}$, $t_{CPA}$, and $b_{CPA}$ for each target. This information can be used later in determining if Risk of Collision exists, and if so, which COLREGS situations apply. Note that Risk of Collision does not mean that a collision is certain or imminent, merely that two vessels will pass within a certain threshold that requires further examination. More often than not, ROC may exist, but no action is required so long as COLREGS is followed and the end result is both vessels maintaining a safe distance from each other.

Due to the inherent nature of using a simulation environment, the target tracking is very clean and accurate, which is not the case when using real radar data in a noisy and dynamic environment.

3.4 COLREGS Situations

Given the calculated $r_{CPA}$ and $t_{CPA}$ for each target tracked, we determine if Risk of Collision exists by comparing $r_{CPA}$ and $t_{CPA}$ to thresholds which we set. See the matlab function function [situation] = colregsSituation( theta, thetab, rcpa, tcpa ) from section 1.4 for a more detailed description of how COLREGS situations are determined.

If Risk of Collision is deemed to exist due to $r_{CPA}$ and $t_{CPA}$ exceeding our rangeThreshold and timeThreshold, we compute the COLREGS situation based on the difference between the Jetyak’s heading $\theta_a$ and the target’s heading $\theta_b$. 
where \( \text{diff} = \theta_a - \theta_b \). The applicable COLREGS situation is then applied as a Velocity Obstacle to display in the Binary Occupancy Grid.

### 3.5 Velocity Obstacles

Velocity obstacles (VOs) are a set of velocities that will result in a collision between one vessel and another at some moment in time, assuming that the other vessel maintains their current velocity. If an ASV chooses a velocity inside the velocity obstacle, then collision will eventually occur, but if a velocity outside the velocity obstacle is selected, such a collision is guaranteed not to occur. Kuwata, Michael Benjamin, Michael Schuster and others have used VOs applied with applicable COLREGS rules in order to perform obstacle avoidance successfully [6, 7, 12]. The velocity obstacles are generally projected from the perspective of the ASV which works well for performing optimal trajectory selection.

In my method, I seek to handle not only moving obstacles, but stationary ones as well as land and other navigational hazards within a map. In order to do this, I project those velocity obstacles with COLREGS situations added as applicable into the environment. This results in creating a map of all navigational hazards and stationary obstacles with dynamic obstacles represented as cones of area projecting their future motion. The basic layout for the VOs I generate are shown in Fig. 12. Based a given target’s position, I project where that vessel will be over a given time \( t_{vo} \) where the distance traveled along the current trajectory is \( d_{vo} = t_{vo} \times v_b \) with \( v_b \) as the velocity of the target vessel. Uncertainty in the course
of the target is added in the VO by making a cone with angle $\theta_{vo}$ as shown. The more certain we are about a target’s course, the smaller $\theta_{vo}$ can be. The last step is to pad the VO with the estimated size of the target vessel and the known size of the Jetyak in order to avoid collision.

![Initial VO with Uncertainty](image1.png)

![Padding with USV and Target Size](image2.png)

Fig. 12. Velocity Obstacles for Target Vessels with added uncertainty

These velocity obstacles allow us to project where a vessel will be within the map over a given time $t_{vo}$, but they do not account for COLREGS situations. In order to do this, we add an additional velocity obstacle if Risk Of Collision (ROC) is deemed to exist ($r_{cpa} < \text{rangeThreshold}$ and $t_{cpa} < \text{timeThreshold}$). Table 2 and Fig. 13 below shows how these additional VOs are created within the environment. As with our other parameters, these values can be tuned to adjust to different navigating environments, different USVs, and different levels of comfort in rangeThreshold, timeThreshold, $t_{vo}$, and prediction uncertainties such as $\theta_{vo}$.

Note that $t_{vo}$ should be large enough to account for at a minimum, several measurement steps. Doing so when combined with a path planner will always ensure the USV’s path will avoid potential obstacles. We use a $t_{vo}$ of 10 seconds.
as our radar retrieves one full 360 degree scan every 2.5 seconds. That ensures VOs project out far enough to predict where a vessel will be over the time 4 measurements are taken. With each measurement, the VO is updated so there is never any chance of predicting a course that causes collision.

<table>
<thead>
<tr>
<th>Situation</th>
<th>Velocity Obstacle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Head On</td>
<td>$d_{\text{headon}} = d_{\text{vo}} \times 2 (r.h.s)$</td>
</tr>
<tr>
<td>Crossing (Give Way)</td>
<td>$d_{\text{giveaway}} = d_{\text{vo}} \times 2$</td>
</tr>
<tr>
<td>Overtaking</td>
<td>$\theta_{\text{overtaking}} = \theta_{\text{vo}} \times 2$</td>
</tr>
<tr>
<td>Crossing (Stand On)</td>
<td>No additional VO</td>
</tr>
</tbody>
</table>

Table 2. COLREGS Velocity Obstacle parameters

Fig 13. Velocity Obstacles for applicable COLREGS situations
Using VOs in this method removes some of the practicality of the methods employed by Kuwata etc, as they allow space in the map to be occupied that has an undefined time associated with it, it simply shows the entire space that will be occupied over a certain time interval $t_{vo}$. It uses a worst case scenario prediction by starting from the origin of the targets position, padded by uncertainty in that measurement, projected out along the predicted course with a a level of uncertainty padded into that. That VO is then padded by the ship’s predicted size and the Jetyak’s size. The final result is a set of points representing anywhere the target could be over the course of time $t_{vo}$. The VOs are projected into the map as a set of rays, then inflated to occupy space in the map as a cone. Once the map is occupied with these VOs, the next step is to determine an optimal path that avoids all of these obstacles.

### 3.6 Probabilistic Roadmaps

Probabilistic Roadmaps (PRMs) are network graphs of potential paths in a map based on both free and occupied spaces [35]. Within the Matlab Robotics Toolbox, we use the `robotics.PRM` class to randomly generate nodes and connections between these nodes based on the PRM input parameters. Nodes are connected based on the obstacle locations specified in a Binary Occupancy Grid, and on the specified connection distance. The number of nodes can be modified to fit the complexity of a given map and efficiency of path required. The more complex the map, the more nodes required. Increasing nodes will provide a
more efficient path, but will increase required processing time. The PRM algorithm uses a network of connected nodes to find a path free of obstacles from a start to an end location. Effectively planning a path through an environment requires tuning of the number of nodes and connection distance properties. Fig. 14 below shows an example of a simple PRM generated and Fig. 15 shows a PRM computed for our simulation environment.

![Probabilistic Roadmap](image)

Fig. 14. Simple Example of Probabilistic Roadmap computing an obstacle free path [35]
3.7 Results

Many different situations were tested, but results of the most easily interpreted example are shown below in Fig 16. Vessel 1 is the Jetyak and we can see it detect other vessels, determine their COLREGS situation, and react accordingly to avoid collision.

Images on the left represent the target positions at a given time. Targets are blue dots with a directional vector and are represented by number. The red “X” marks represent the Jetyak’s current waypoints. Images on the right represent the generated velocity obstacles that match the frame to the left. These VOs are generated as a set of rays and inflated to account for target size, Jetyak size, and simulated uncertainty. Note as COLREGS situations occur and VOs occupy the

Fig. 15. Probabilistic Roadmap for Jetyak Simulation based on Charles River Map
map how the waypoints are modified. The map is produced to emulate an actual bend in the Charles River.

In this simulation, the Jetyak is initialized as “Target 1” and there are 4 other targets generated within the map. Target 2 is on a reciprocal path, course, and speed to the Jetyak’s and causes a head on situation ~35 seconds into the simulation. When this head on situation is encountered, a COLREGS VO for a head on situation is generated within the map. The PRM path planner accounts for all the generated VOs at each timestep and output an optimal path to avoid all obstacles.

Target 3 is a slow moving vessel that is crossing the river. When ROC parameters are exceed with target 3, no VO is necessary because the two vessels are in a crossing situation and the Jetyak is the Stand On vessel. Target 4 is moving in roughly the same direction as the Jetyak up the river, but stays to the right side. The Jetyak has to overtake Target 4 while simultaneously dealing with the head on situation with Target 2. Because Target 4 is moving slowly and the COLREGS VOs are proportional to velocities of the targets, the COLREGS VO for overtaking is relatively small. The remaining unoccupied space in the map is used to compute a set of waypoints to avoid obstacles. The result is a varying set of waypoints that prevent collision between the Jetyak and other vessels while following COLREGS.

Target 5 can be effectively ignored as it is beyond the end waypoint for the Jetyak. It is another slow moving vessel that is crossing the river from right to left.
Towards A COLREGS Compliant Autonomous Surface Vessel

Should the Jetyak need to maneuver around this type of vessel, the Jetyak would be the Give Way vessel in a crossing situation and a velocity obstacle projecting out in the direction of the Stand On vessel would force the PRM to deduce a set of waypoints that cause the Jetyak to come to starboard and safely pass from the other target’s stern. Because Target 5 is slow moving, it is possible that the generated VO would be small enough for the Jetyak to cross in front of the target vessel. This may be the best option if there is little room on the right hand side of the river to navigate around the target’s stern.
Fig. 16. Simulation Results of COLREGS compliant navigation. Left: Targets in map (1 is Jetyak) with waypoints via PRM Right: VOs + COLREGS VOs where applicable at same time as image on left.
4. Data collection and Analysis

4.1. Environmental Conditions

The Jetyak was taken out on the water many times, both at the Woods Hole Oceanographic Institute (WHOI) in Woods Hole, MA and on the Charles River in Boston, MA. The main dataset demonstrating the Jetyak’s collision avoidance capabilities was collected in one of the 8 km (4.2 NM) transits from the Northeastern Henderson boathouse to the MIT sailhouse (Fig. 17). The transit included many tight winding turns on a narrow river as narrow as 50 m across and only as wide as 600 m at its widest throughout this transit. As the data was collected during the Winter and early Spring, there were very few other vessels on the water other than the Jetyak. A chase boat shown in Fig. 18 was used to both follow the Jetyak and create artificial COLREGS situations to test both the Target Identification and Tracking performance and ultimately the ability for the Jetyak to generate obstacle free paths while following COLREGS.
Fig 17. Google Earth imagery of 8km (4.2 NM) transit from Northeastern boathouse to MIT boathouse on Charles River, Boston, MA [34]

Fig 18. Chase Boat used for following Jetyak and for Target Identification and Tracking and COLREGS avoidance testing.

Data was collected from all sensors via ROS with the radar range set to 100m with the Jetyak transiting up the Charles River. Data from rosbags are then replayed using ROS and Python so radar data can be displayed visually. Using the
vehicles heading and azimuth of each radar scan with respect to the the vessel, ego motion compensation is performed. Offsets are also performed using the vessels GPS position converted from degrees and minutes of latitude to meters in UTM coordinates in order to overlap radar imagery with chart imagery. We are able to accurately overlay radar data with imagery from either Google maps or ENC charts while the Jetyak is stationary or moving.

In order to test the algorithms derived through simulation, the Jetyak was given a set of waypoints to navigate autonomously from the Northeastern Henderson boathouse 8 km to the MIT boathouse up the narrow and shallow Charles River. An ENC chart for this area was generated using Google Maps and QGIS as NOAA doesn’t generate charts for this area. While this practice is highly inadvisable for general navigation, it meets our standard for determining what radar data constitutes targets on the land or the water.

The initial set of waypoints provided for the Jetyak were slightly to the right of the center of the river. This data was collected with the Jetyak navigating in autonomous waypoint following mode, but without live obstacle detection and avoidance in effect. All results shown use data collected from this transit with the ability to detect obstacles, create and display COLREGS applicable VOs, and create an optimal set of waypoints to avoid obstacles in real-time.

The radar detection and tracking system determines when there are obstacles on the water and generates velocity obstacles based on their determined positions, courses, and speeds through COLREGS, UKF, and JPDA.
These velocity obstacles are then projected onto the existing binary occupancy grid based on the generated ENC chart. If Risk Of Collision is deemed to exist, Visibility Graph Inspired Path Planning (VGIPP) is used to find an optimal path to avoid collision in accordance with COLREGS. In the future, these waypoints will be provided to the Pixhawk to change the current path to the new one determined to minimize risk while still making way towards the end goal.

4.2 Shore Exclusion

Based on work by [6], targets are first identified and connected component labeling is used to group nearby occupied cells into a single target. These targets are represented as outlines of their entire shape and displayed as complex polygons in the map. If the target is verified to be on the water and not on land, they are considered as potential targets, otherwise they are eliminated as potential targets as the ENC chart will operate in the binary occupancy grid.

Because the transit includes passing under 7 different bridges such as the ones shown in Fig. 19 below, a separate layer for our ENC chart was created to indicate a navigable path through the center of each bridge. In our testing, the Jetyak was able to successfully navigate under the bridges most times, but suffered from magnetic interference on several occasions causing the Pixhawk autopilot to malfunction. This can be rectified in the future through temporary use of dead-reckoning for a short enough period of time to pass under the bridge.
It should also be noted that while near and under the bridges, radar data near the Jetyak is unreliable and not sufficient for obstacle detection and avoidance. If another vessel were to pass under the bridge at the same time as the Jetyak, we would have no way of detecting and avoiding it. This could be potentially solved through use of a temporarily activated vision system, and possibly LIDAR scanning. This is outside the scope of this project, but an important consideration for creating a robust USV able to handle a diverse set of situations.

4.3 Target Identification and Tracking

An Unscented Kalman Filter (UKF) is implemented to determine a given target’s position in x and y, course, speed, yaw rate, acceleration, and yaw acceleration. Standard Kalman Filters (KF) utilize the “Predict, Measure, and Update” steps that give it great performance for linear systems and serve as a generality of Bayesian Estimation. For non-linear systems such as ours, an
Extended Kalman Filter (EKF) can be used to linearize the system via Taylor Expansion to make estimates. This works well in many cases, but can be unstable and take a long time to properly converge. The method we utilize is the Unscented Kalman Filter (UKF), which is essentially an EKF, but rather than using a single point for estimation, multiple Sigma points are selected, transformed through a nonlinear function, and used to compute a multivariate Gaussian based on the transformed and weighted points. The number of Sigma Points is typically $2N+1$ points where $N$ is the dimensionality of the state-vector. In our case, our state-vector has 7 dimensions, so we use 15 sigma points ($2(7)+1=15$). For more detailed discussions on the merits of KF, EKF, and UKF filters, see Ref. [36].

Initial target predictions tend have a large covariance, but as the number of measurements increase, that covariance decreases. We always assume a “worst case scenario” and underestimate the accuracy of our tracking in order to prevent any poor collision avoidance recommendations. Estimates and measurements are padded with an additional uncertainty and the final state is then used in the obstacle avoidance algorithm.

We assume that any object creates only a single measurement and that target paths can be in close proximity. Therefore, we can use a Joint Probabilistic Data Association (JPDA) [16]. The JPDA associates all target measurements within a certain gate in order to track a given target to all other tracks available within the gate. Fig. 20 below demonstrates the method by which we track targets over time frames. Each target has a given state vector including its position with an
associated covariance which is used to determine the likelihood that another target in the next frame is the same target. Every target in frame 1 is checked with every target in frame 2 and the target correlation with the highest probability is selected. In order to prevent highly improbable results, targets are only checked in a range of $r_{JPDA} = v_b \cdot t_{meas}$ that represents the maximum distance we believe any given target could travel in the time between measurements. In our case, since $t_{meas} = 2.5s$ for each radar scan and we are highly unlikely to encounter any vessels on the Charles with a $v_b > 10 \text{ m/s}$, we use $r_{JPDA} = 25\text{m}$. Notice that the red target is not checked with the green target between frames as they are too far apart to be possible matches. In environments where we do expect to see faster vessels, $r_{JPDA}$ can simply be increased accordingly. For a more in depth discussion of the target identification and tracking via UKF and JPDA see Ref. [6].

Fig. 20. Joint Probabilistic Data Association (JPDA) overview
4.4 COLREGS Velocity Obstacles

As described in the sections 3.4 and 3.4, when the $r_{CPA}$ and $t_{CPA}$ pass a certain threshold, we deem Risk Of Collision to exist between our vessel and a target vessel. Based on the Jetyak’s heading and the target’s estimated heading, the applicable COLREGS situation is determined and applied as a Velocity Obstacle. The equations for creating COLREGS VOs remains the same as the equations from the simulation environment, however they are displayed directly as polygons rather than a set of rays projected into the binary occupancy grid. The end result is the same, which is target vessels which the Jetyak has Risk Of Collision with are shown as occupied space via VOs according to the applicable COLREGS situation. The map is updated at every measurement and an optimal set of waypoints is computed to navigate around potential obstacles.

4.5 Path Planning

Next, we can account for the COLREGS VOs by using a reactive motion planning algorithm based on elements of the one proposed by Kuwata et. al. [7]. Rather than evaluating an optimal trajectory for the USV as Kuwata does, the velocity obstacles are represented in the Binary Occupancy Grid of the environment. When Risk Of Collision is deemed to exist on the Jetyak’s current set of waypoints, a new set of waypoints is created through use of Visibility Graph Inspired Path Planning (VGIPP). See Fig. 21 below for a general depiction of how
VGIPP is conducted and Fig. 22 for an application of VGIPP with a head on situation COLREGS VO. The new set of waypoints can be broadcast to the Jetyak and executed in order to avoid areas that would cause a risk of collision. Using VGIPP is computationally less expensive than using Probabilistic Roadmaps and has proven to work well on our datasets. In the future, other path planning methods such as A* may be considered.

In the worst case scenario when no acceptable path is found, the Jetyak can slow down to bare steerageway, rotate in place, or stop completely. The key difference between our method and that of Kuwata’s and other velocity obstacle based collision avoidance systems is that while they provide an optimal trajectory, our method provides a new set of waypoints. This is an important distinction because it allows us to visually display our projected path over time based on the current environment. Both dynamic and stationary obstacle velocity obstacles are projected into the binary occupancy grid of our environment and if Risk Of Collision is deemed to exist, a new path in unoccupied space is determined and executed.
Towards A COLREGS Compliant Autonomous Surface Vessel

Visibility Graph Inspired Path Planning

Fig. 21. Visibility Graph Inspired Path Planning Stages

Fig. 22. Visibility Graph Inspired Path Planning applied to a Head On situation with COLREGS VO
4.6 Results

Currently we are able to 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 aren’t executed live on the Jetyak, they are calculated in real-time with real data and can be used to avoid collision with both dynamic and static obstacles in a highly constrained environment. The ultimate goal was to identify, track, and avoid all targets in a dynamic environment according to COLREGS and the methods to do this have been successfully created. Figures below demonstrate highlights from the 8 km transit on the Charles River from Northeastern’s boathouse to MIT’s boathouse.

Fig. 23 shows the target vessel overtaking the Jetyak. The applicable COLREGS rule requires the vessel overtaking the other to stay clear, which allows the Jetyak to maintain its current course and speed. Appropriately, no COLREGS VOs are added to the map and there is no change in the Jetyak’s trajectory. Fig. 24 shows the Jetyak overtaking the target vessel with an overtaking COLREGS VO generated. There Jetyak’s current set of waypoints does not cross through the VOs though so there is no need to determine a new set of waypoints. Fig. 25 shows the tracking performance of the UKF and JPDA as the target vessel performs erratic movements in front of the Jetyak. There was no Risk Of Collision despite these apparent erratic patterns, so no COLREGS VO was generated and the Jetyak maintained its current trajectory. Had the target vessel had ROC, the
applicable COLREGS VO would have been generated. If that COLREGS VO blocked the current set of waypoints, the VGIPP would have computed an optimal set of waypoints to avoid collision.

Fig. 26 shows the first of seven bridges encountered throughout the transit. The green patch represents a navigable path through the center of the bridge. This is necessary to discern as the radar shows this area as occupied, but prior knowledge of the environment tells us that the area is actually navigable. The downside of this method is that while the Jetyak is navigating through the bridge, there is no ability for the radar to detect and avoid obstacles. Methods to rectify this in the future are discussed in more detail later.

Fig. 27 is likely the most important of these results as it demonstrates the detection of a head on situation with the target vessel, computes and displays the head on COLREGS VO, and the subsequent waypoint output signaling the need for the Jetyak to come to starboard (turn right) to not only avoid the obstacle, but to do so in accordance with the rules of the road. There are several other head on situations encountered throughout this transit and all of them were accurately detected and the appropriate set of waypoints was computed to avoid all obstacles while following COLREGS.

While crossing situations were created for brief instances of time, the river was too narrow to generate a crossing situation that lasted long enough to generate a crossing situation COLREGS VO. Most crossing situations lasted no more than 1 or 2 radar frames (”2.5-5 seconds). For a narrow river like the Charles,
the most likely situation to encounter would be a head on situation or an overtaking situation and all of these were appropriately handled. In the future, more open water testing may be useful in better testing the performance of all possible COLREGS situations with multiple vessels involved.

The next two figures, Fig. 28 and Fig. 29 are two more examples of path planning performed through bridges. The last four bridges have been omitted due to their similarity in appearance to the first three.

![Fig. 23. Jetyak Being Overtaken (No Risk Of Collision or COLREGS VO)](image-url)
Towards A COLREGS Compliant Autonomous Surface Vessel

Fig. 24. Jetyak Overtaking Target Vessel (Risk of Collision Exists, Small VO applied, no trackline modification necessary)

Fig. 25. Tracking Performance of Target Vessel via UKF and JPDA
Towards A COLREGS Compliant Autonomous Surface Vessel

Fig. 26. Jetyak Entering First Bridge. Green represents “navigable lane” through center of bridge.

Fig. 27. Jetyak in Head On Situation with target vessel (Risk Of Collision Exists, Head On VO applied, path appropriately modified to avoid collision)
Towards A COLREGS Compliant Autonomous Surface Vessel

Fig. 28. Jetyak Entering Second Bridge. Green represents “navigable lane” through center of bridge.

Fig. 29. Jetyak Entering Third Bridge. Green represents “navigable lane” through center of bridge.
5. Conclusions and Future Work

The work presented represents a COLREGS compliant USV that is able to generate waypoints in real-time that avoid both static and dynamic obstacles in a constrained environment. It is able to do so on data collected in a live environment with a relatively low-cost vessel utilizing open source hardware and software. Promising results were attained in full simulation, path planning using real world data, and real world COLREGS obstacle avoidance. Using JPDA and UKF, we are able to determine a target’s course, speed, and CPA information required to make COLREGs based maneuvers and can then project them as Velocity Obstacles into our map. The resulting map is a binary occupancy grid that contains all navigational hazards, stationary objects, dynamic object VOs, and COLREGS VOs representing occupied space. Using Visibility Graph Inspired Path Planning we successfully calculate an optimal set of waypoints that avoids collision with all obstacles.

The next steps include utilizing the generated obstacle free waypoints for navigation in order to perform real-time obstacle detection and avoidance on the water. A more robust obstacle avoidance system specifically for transiting under bridges should also be considered as obstacle detection via radar fails in close proximity to bridges. Further tuning of parameters with the possibility of dynamic updating depending on the environment could further improve performance. The
principles applied here could be transferred to other vessels relatively easily and could be tested on datasets collected by larger vessels with greater constraints.
6. References


[18] Pixhawk Arudupilot Board.


[22] Taranis X9D Plus Transmitter

[23] Ubiquiti Long Range Wifi
https://store.ui.com/collections/wireless/products/rocket-m2, 2019


[26] Robot Operating System (ROS) http://www.ros.org/about-ros/, 2018


[34] Google Earth, https://www.google.com/earth/, 2019

[36] Robot Mapping, Unscented Kalman Filter, AIS,

[37] Intel NUC https://www.Intel.com, 2018

[38] MAVLink https://mavlink.io/en/, 2018