Risk Vector-based Near miss Obstacle Avoidance for Autonomous Surface Vehicles

Mingi Jeong and Alberto Quattrini Li

***Abstract*— This paper presents a novel risk vector-based near miss prediction and obstacle avoidance method. The proposed method uses the sensor readings about the pose of the other obstacles to infer their motion model (velocity and heading) and, accordingly, adapt the risk assessment and take corrective actions if necessary. Relative vector calculations allow the method to perform in real-time. The algorithm has 1.68 times faster computation performance with less change of motion than other methods and it enables a robot to avoid 25 obstacles in a congested area. Fallback behaviors are also proposed in case of faulty sensors or situation changes. Simulation experiments with parameters inferred from experiments in the ocean with our custom-made robotic boat show the flexibility and adaptability of the proposed method to many obstacles present in the environment. Results highlight more efficient trajectories and comparable safety as other state-of-the-art methods, as well as robustness to failures.**

I. Introduction

The goal of this paper is to enable safe navigation of marine autonomous surface vehicles (ASVs) via an adaptive real-time static and dynamic obstacle avoidance method. The proposed method presents more flexibility compared to other methods – see Fig. 1 for an illustration of trajectories followed by an ASV using a state-of-the-art method compared to the one followed using our proposed method.

Current methods for marine vessel obstacle avoidance (e.g., [1]–[3]) have shown safe behavior and ability to avoid static and dynamic obstacles, following COLREGs, Convention on the International Rules for Preventing Collisions at Sea [4]. One of the main criteria taken into account includes the Closest Point of Approach (CPA) – the positions at which two dynamically moving objects get to their closest possible distance. Such a factor does not encode velocity or direction of the controlled or other vehicles, resulting in the algorithm working in a conservative manner that requires increased computational and operational time. Some methods have been developed to address this problem by taking into account velocity and direction of other obstacles [5]–[7]. By analyzing the Velocity Obstacle (VO) cone and choosing an action, the corresponding vector is determined to lie outside of that cone. The velocity relationship between the controlled ASV and an obstacle is not explicitly considered, resulting in potentially ineffective trajectories, e.g., passing ahead of the obstacle.

To address the above mentioned problems, in this paper, we propose a novel risk vector-based near miss prediction and obstacle avoidance method. The proposed methodology is based on calculating the risk according to, in addition to the CPA, the encountering angle and velocities of the vehicles to foster an adaptive avoidance behavior. The methodology is based on relative vector calculations to make the proposed method perform in real-time for risk assessment and action decision. We also include a sequence of fallback algorithms to account for noise in the sensors and ensure the safety of the vehicles. This allows the trajectory to be as safe as the previous methods but with improved performance in terms of traveled distance and computation time. After calibrating parameters of a simulated ASV with experiments with a real ASV, numerous simulations under different encountering scenarios, number of obstacles, and failures validate the proposed approach.

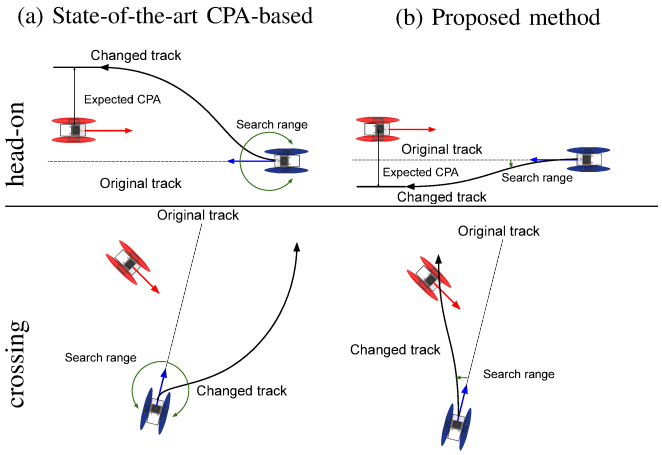


Fig. 1: Obstacle avoidance behaviors from a state-of-the-art COLREG-constrained method keeping a fixed CPA (left) vs. our adaptive method (right) able to achieve the same CPA but with less traveled distance, under head-on and crossing scenarios. **Red vehicle:** obstacle. **Blue:** controlled ASV.

This paper provides the following main contributions:

- A flexible risk vector-based near miss prediction method[1], which evaluates dynamic encounters (e.g., head-on, overtaking, and crossing) adapting to the specific condition based on different size, velocity, and compliance with COLREGs;
- Relative vector-based predictions for real-time control, resulting in assessment of situation and corresponding action that can be computed in a few milliseconds;
- Fallback behaviors to react to unexpected behaviours

The authors are with Department of Computer Science, Dartmouth College, Hanover, NH USA {mingi.jeong.gr, alberto.quattrini.li}@dartmouth.edu

[1]The code will be made opensource on our lab website https://rlab.cs.dartmouth.edu

 DOI: 10.1109/IROS45743.2020.9341105

of obstacles, such as close-quarter situations after an action, or sensor failure; and
- Experimental evaluation and comparison with CPA- or VO-based methods that highlights the feasibility of our approach in avoiding multiple obstacles.

This work represents a first effort towards a context-aware collision avoidance method that can be applicable to ASVs and Maritime Autonomous Surface Ships (MASS), to ensure safer navigation. This will result in the advancement of many important applications, including environmental monitoring, military missions, search and rescue, and oil spill response [8].

This paper is structured as follows: the next section shows the related work in maritime obstacle avoidance. Section III describes the proposed approach including the framework of risk vector-based concept and near miss computation. Section IV presents results and demonstrations of the proposed method with different encounter situations and the number of obstacles. Finally, Section V highlights strengths and potential extensions.

II. Related work

Collision avoidance and planning for robots have been studied for decades in literature, with methods that use techniques spanning from control theory to algorithms, and that search a collision-free path in a global domain vs. a local domain [9]. In the robotics literature, example of methods include RRT-based [10], PRM-based [11] for global methods; and Vector Field Histogram [12], Dynamic Window Approach [13], velocity based [5] for local methods.

For maritime navigation, the literature has focused on the collision avoidance from obstacles happening in the local domain [14], as many of the obstacles are not known until discovered with the on-board sensors. Recently, [15] proposed metrics to evaluate how compliant an algorithm is to COLREGs. Most of the research on maritime collision avoidance for autonomous vessels assess the situation and accordingly decide on the actions considering different criteria: operability in diverse encounter situations and mobility (static or dynamic obstacle) [2], [16], complexity of decision-making on an evasive action [17]–[19], safety metric represented by Closest Point of Approach (CPA) (or variations of CPA) [3], [20], compliance with COLREG [4], [21], vector-based velocity obstacle [6], [22], the number of obstacles in the corresponding circumstances [1], [21], and feasibility of contingency maneuver in case of unexpected close-quarter situations [18], [23]. An important concept is the concept of *near miss*, defined as a situation where a collision did not happen, but a slight change in motion could have resulted into collision, typically based on Distance to CPA or Time to CPA [24].

Such metrics do not account for the specific encountering situation, such as the current velocity of the obstacles, the vector's encounter relationship with the controlled vehicle, or obstacle size, resulting sometimes in an analysis that does not reflect the actual near miss. We address the problem of "how to assess the risk in a more adaptive way, according to the specific encountering scenario, so that appropriate maneuvers are taken by the ASV?".

III. Approach

Our approach assesses the risk and determines the actions to avoid collisions from obstacles that are within the ASV sensor range. We assume that any obstacle within that range can be detected. We also relax this assumption in our experiments, simulating a sensor failure to evaluate the responsiveness of our algorithm. While obstacle detection from, e.g., cameras, LIDAR, RADAR, is an interesting research direction, it is out of the scope of this paper.

In addition, we assume that our ASV is a *Give-way* vehicle, following the COLREG definition [4]: our vehicle takes a preemptive action to avoid a collision, while other dynamic obstacles are considered as *Stand-on* vehicles, which means they have the right of way [4]. The reason for this choice is to robustly consider the worst case for the controlled vehicle, thus taking a conservative behavior, as in many cases, people do not follow the rules even if they shall act as a *Give-way* vehicle by the rules.

To simplify our presentation, we describe the concept and method of risk vector and near miss calculation in a single obstacle situation. The multi-obstacle situation can be extended and handled by our method as shown in Section IV.

A. Relative perspective

Computations necessary for algorithms – e.g., motions, poses, tangents to a boundary of an obstacle – are modeled in two ways: a relative perspective with respect to the ASV's reference frame $\{R\}$, and a relative perspective with respect to the obstacle's reference frame $\{O\}$. For efficiency and computational loads, the proposed method has a significant advantage over other methods that reason in the global reference frame $\{W\}$, e.g., [14]. In other words, between two variables changing simultaneously, the relative approach reduces complexity by considering only one variable from the fixed perspective (with respect to $\{O\}$) as shown in Fig. 2. Specifically, the relative motion vector of own ASV

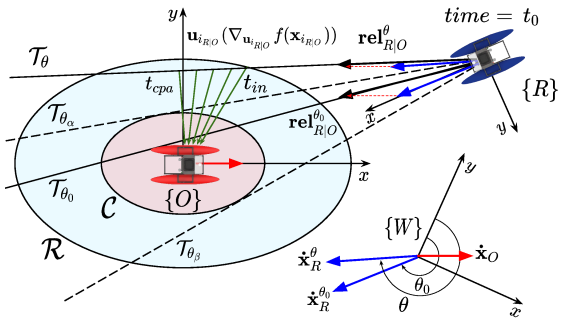


Fig. 2: Conceptual diagram for boundary setup, risk vector, predicted track, near miss, and evasive action computation when Case 1 with respect to $\{O\}$ – $\exists \mathcal{T} \cap \mathcal{C}$. **Red vehicle:** obstacle. **Blue vehicle:** own ASV.

with respect to the obstacle can be represented by

$$\mathbf{rel}_{R|O} = \mathbf{T}_W^O(\dot{\mathbf{x}}_R - \dot{\mathbf{x}}_O) \tag{1}$$

where $\mathbf{T}_W^O$ is the homogeneous transformation matrix of $\{O\}$ with respect to $\{W\}$. In addition, own ASV's pose with respect to the obstacle is defined as

$$\mathbf{x}_{R|O} = \mathbf{T}_W^O\,\mathbf{x}_R \tag{2}$$

where $\mathbf{x}_{R|O}$ is a relative pose of the ASV with respect to $\{O\}$, and $\mathbf{x}_R$ is a relative pose of the ASV with respect to $\{W\}$. In the same way, $\mathbf{rel}_{O|R}$ and $\mathbf{x}_{O|R}$ are derived.

B. Boundary and Risk vector

The concept of risk vector for collision assessment is based on a virtual boundary – so called *ship domain* – of an obstacle [3], [20], where the controlled ASV shouldn't enter. The boundary is divided into two areas: *risky boundary* ($\mathcal{R}$), where evasive maneuvers can be evaluated "safely"; and *collision boundary* ($\mathcal{C}$), when inside, it is regarded as collision – see Fig. 2. Here, the risky boundary follows a linear model for the collision risk [14], [20]. When the obstacle is within the ASV *sensible range* ($\mathcal{L}$), the method evaluates the situation depending on whether it will go through the collision boundary, the risky boundary, or neither of them. Based on the adaptive modeling of the ship domain [18], [20], $\mathcal{R}$ and $\mathcal{C}$ with respect to the obstacle's reference frame are derived as:

$$\mathcal{C}_x = L_R + L_O + v_O * t_u + v_R * t_u \tag{3}$$

$$\mathcal{C}_y = \mathcal{C}_x * k_{\mathcal{C}} \tag{4}$$

where $\mathcal{C}_x$ is a semi-major axis of $\mathcal{C}$, $\mathcal{C}_y$ is semi-minor axis of $\mathcal{C}$, L_R, L_O is the length of the ASV and the obstacle, v_R, v_O is the speed of the ASV and the obstacle (m/s), t_u is a unit time (s) and k_C is ratio between major and minor axis of the domain [25], and

$$\mathcal{R}_x = \mathcal{C}_x * k_{\mathcal{R}} \tag{5}$$

$$\mathcal{R}_y = \mathcal{C}_y * k_{\mathcal{R}} \tag{6}$$

where $\mathcal{R}_x$ is a semi-major axis of $\mathcal{R}$, $\mathcal{R}_y$ is semi-minor axis of $\mathcal{R}$, and k_R is abort distance coefficient from $\mathcal{C}$ based on a requirement of a tactical diameter: the diameter of a turning circle requiring the minimum maneuverability [26].

While the ASV passes the risky boundary, the risk vector is defined as:

$$\mathbf{risk}(\mathbf{x}_{R|O}) = \mathbf{u}_{R|O}(\nabla f(\mathbf{x}_{R|O}) \cdot \mathbf{u}_{R|O}) \tag{7}$$

$$\nabla f(\mathbf{x}_{R|O}) \cdot \mathbf{u}_{R|O} = \lambda \tan\psi \tag{8}$$

where $\nabla_{\mathbf{u}_{R|O}}$ is directional derivative based on a unit vector $\mathbf{u}_{R|O}$, $\mathbf{u}_{R|O}$ is a vector with a direction from the controlled ASV position to the center position of the obstacle on the obstacle reference frame, $f(\mathbf{x}_{R|O})$ is a linear cost function determining the risk vector toward the center position of the obstacle [14] on ASV position $\mathbf{x}_{R|O}$, λ is a level of danger based on types of vehicles [3], and ψ is slope angle with direction to the center of the obstacle in 3D.

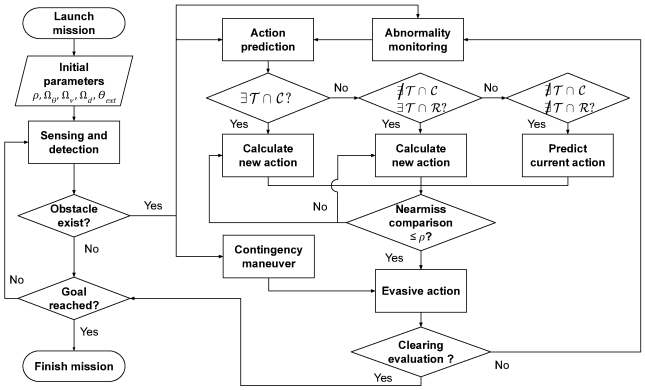


Fig. 3: System architecture.

C. Near miss definition

Assume that the vehicles – obstacles and controlled robots – are following the set course at a constant speed. The state vector for the pose is expressed in $\mathbf{x}_R = [\,x_R\ y_R\,]^{\mathbf{T}}$ and $\mathbf{x}_O = [\,x_O\ y_O\,]^{\mathbf{T}}$ where x_R, y_R, x_O, y_O are global GPS coordinates of an ASV and obstacle on a Cartesian coordinate system. Hence, the true motion vectors on the global frame can be expressed in $\dot{\mathbf{x}}_R = [\,\dot{x}_R\ \dot{y}_R\,]^{\mathbf{T}} = [\,v_R\cos\theta_R\ \ v_R\sin\theta_R\,]^{\mathbf{T}}$ and $\dot{\mathbf{x}}_O = [\,\dot{x}_O\ \dot{y}_O\,]^{\mathbf{T}} = [\,v_O\cos\theta_O\ \ v_O\sin\theta_O\,]^{\mathbf{T}}$ where v_R and v_O is the velocity of the speed for the robot and obstacle, and θ_R and θ_O is the heading angle based on clock-wise direction from 0° as North.

We propose a novel near miss calculation considering factors such as movements of vehicles, encounter situations, and time and distance during encounters. For an expected trajectory $\mathcal{T}$ followed by the ASV, which enters the risky boundary $\mathcal{R}$ of an obstacle at time t_{in} and gets to the CPA at time t_{cpa}, , the near miss $\Gamma(\mathcal{T})$ is defined as:

$$\Gamma(\mathcal{T}) = \int_{\mathcal{T}_R} \mathbf{rel}_{R|O} \cdot \mathbf{risk}(\mathbf{x}_{R|O})\, ds_{R|O} \tag{9}$$

where $\mathcal{T}$ is the expected trajectory of the ASV, $\mathbf{rel}_{R|O}$ and $s_{R|O}$ are a relative motion vector and the trajectory of the robot with respect to the obstacle reference frame. The computation indicates that, intuitively, the higher relative velocity, the higher collision near miss.

Considering the sensor data coming at discrete time intervals, the near miss computation becomes [27], [28]:

$$\Gamma(\mathcal{T}) = \sum_{i=1}^{n} \mathbf{rel}_{i_{R|O}} \cdot \mathbf{risk}_i(\mathbf{x}_{R|O})\, \Delta s_{R|O} \tag{10}$$

where n is the number of sensing data points that depends on the sensor frequency.

D. Near miss-based obstacle avoidance

An action prediction step in Fig. 3 is based on predictions starting at time t_0 when an ASV detects/tracks an obstacle – see pose of own ASV in Fig. 2.

The proposed method outputs an evasive action depending on a relationship between $\mathcal{T}$ and $\mathcal{R}$, $\mathcal{C}$.

$$\mathcal{T} = \mathbf{x}_{R|O} + \mathbf{s} \quad \text{s.t.} \forall \mathbf{s} \in \mathcal{S}(\mathbf{rel}_{R|O}) \tag{11}$$

$$\mathcal{S}(\mathbf{rel}_{R|O}) = \left\{ t\,\mathbf{rel}_{R|O} \mid t \in \mathcal{F} \cap [t_0, t_{\max}] \right\} \tag{12}$$

where $\mathcal{S}(\mathbf{rel}_{R|O})$ is the linear span of vector $\mathbf{rel}_{R|O}$, $t_{\max}$ is a maximum monitoring time limit, and $\mathcal{F}$ is a set of time stamps according to the sensor frequency. In general, $t_{\max}$ can be determined by the time that can be derived from maximum sensor range $\mathcal{L}$ divided by $\left\|\mathbf{rel}_{R|O}\right\|$.

1) Case 1 – $\exists\,\mathcal{T} \cap \mathcal{C}$: The ASV is expected to collide with the obstacle by entering $\mathcal{C}$ as shown in Fig. 2. The proposed method makes the ASV take an action in comparison with tangent lines to $\mathcal{C}$ by changing ${\mathbf{rel}^{\theta}}_{R|O}$, a relative motion vector of the ASV with respect to $\{O\}$ when the ASV's heading is changed to θ. According to Equations (1) and (2), ${\mathbf{rel}^{\theta}}_{R|O}$ can be derived by changing the heading of own ASV from initial $\dot{\mathbf{x}}_R$ while $\dot{\mathbf{x}}_O$ is fixed.

As shown in Fig. 2, there exist two tangent lines except for a case that ASV is on the edge of $\mathcal{C}$. The ASV's vector matching the tangent line can be derived by the following system of linear equations:

$$\begin{cases} \mathbf{rel}^{\theta}_{R|O} \times \mathbf{p}(\theta) = \mathbf{0} \\ \left\|\dot{\mathbf{x}}^{\theta}_{R}\right\| = \left\|\dot{\mathbf{x}}^{\theta_0}_{R}\right\| \end{cases} \tag{13}$$

where $\mathbf{p}(\theta)$ is tangent point on $\mathcal{C}$ when the heading is θ, $\dot{\mathbf{x}}^{\theta}_{R|O}$ and $\dot{\mathbf{x}}^{\theta_0}_{R|O}$ is velocity of ASV when the heading is θ and the initial heading θ_0, respectively.

In such a case, Equation (13) returns two heading angles. Let θ_α and θ_β be the headings corresponding to the tangent lines. The algorithm compares values between $\Gamma(\mathcal{T}_{\theta_\alpha})$ and $\Gamma(\mathcal{T}_{\theta_\beta})$ and takes an action based on a safety threshold:

$$\frac{\Gamma(\mathcal{T}_\theta)}{\min(\Gamma(\mathcal{T}_{\theta_\alpha}), \Gamma(\mathcal{T}_{\theta_\beta}))} \leq \rho \tag{14}$$

where $\Gamma(\mathcal{T}_\theta)$ is the near miss as per the changed heading, $\Gamma(\mathcal{T}_{\theta_\alpha})$ and $\Gamma(\mathcal{T}_{\theta_\beta})$ are defined as the upper-bound near miss as per the heading θ_α, θ_β, and ρ is the safety threshold for the evasive action depending on operator's preference.

Since ρ is determined adaptively in a same way as preferred CPA [3], [20] by an operator, the proposed method can react on the encounter situation in a more flexible way than other methods [16], [22]. For instance, the lower threshold, the safer action as the ASV will clear out of the obstacle with less $\Gamma(\mathcal{T}_\theta)$ as well as more safe distance.

Moreover, the final heading is found by a relative relationship with either θ_α or θ_β. For example, if θ_α is located on the left side of θ_0, θ can be found by turning the ASV's heading more to the left from θ_α, i.e., the upper-bound angle in this case. The obstacle avoidance control action is initiated immediately according to the calculated θ.

2) Case 2 – $\nexists\,\mathcal{T} \cap \mathcal{C}$ and $\exists\,\mathcal{T} \cap \mathcal{R}$: The ASV is expected not to collide with the obstacle, but to pass within $\mathcal{R}$. In this case, from the same method to get aforementioned tangency, let θ' be either θ_α or θ_β satisfying the condition $\min(|\theta_\alpha - \theta_0|, |\theta_\beta - \theta_0|)$. There are two possible scenarios:

- $\frac{\Gamma(\mathcal{T}_{\theta_0})}{\Gamma(\mathcal{T}_{\theta'})} \leq \rho$ where the current heading (θ_0) already meets the requirement of the safety threshold based on the close tangent bound from θ_0. Thus, the ASV is not required to change the current track to avoid the obstacle.
- $\frac{\Gamma(\mathcal{T}_{\theta_0})}{\Gamma(\mathcal{T}_{\theta'})} > \rho$ where the heading should be changed from θ_0. In such a case, θ is found based on the upper bound $\Gamma(\mathcal{T}_{\theta'})$ in the same way as *Case 1* to make $\frac{\Gamma(\mathcal{T}_\theta)}{\Gamma(\mathcal{T}_{\theta'})} \leq \rho$.

3) Case 3 - $\nexists\,\mathcal{T} \cap \mathcal{C}$ and $\nexists\,\mathcal{T} \cap \mathcal{R}$: The ASV is expected to enter neither $\mathcal{C}$ nor $\mathcal{R}$. Therefore, the ASV can safely pass the obstacle without changing action.

4) Additional case - velocity: Based on the same method we introduced above, the robot can also change velocity only or velocity and heading together for a new $\mathbf{rel}_{R|O}$. This is mostly done in case of restricted visibility or requirement of situational awareness. The velocity and the heading change option will be effective considering advantages of heading change and velocity change. The comparison will be discussed in Section IV-A.

E. Abnormality monitoring

In the previous sections, we assumed that the obstacle keeps the same velocity and heading. In general, this is not true and we propose a method – detailed in Algorithm 1 – to detect changes to recalculate the action to avoid the obstacle.

Algorithm 1 Abnormality monitoring

Input: $\dot{\mathbf{x}}_O$, $\dot{\mathbf{x}}'_O$, Ω_θ, Ω_v, t, t_{freq}
Output: θ

```
if (t / t_freq) remainder = 0 then
    if (‖ẋ_O − ẋ′_O‖ ≥ Ω_v) or
    (arccos(ẋ_O · ẋ′_O / (‖ẋ′_O‖‖ẋ_O‖)) ≥ Ω_θ) then
        θ ← recalculate near miss-based action
        ẋ_O ← ẋ′_O
return θ
```

$\dot{\mathbf{x}}_O$ and $\dot{\mathbf{x}}'_O$ are a current and a previous motion vector, Ω_θ and Ω_v are monitoring thresholds for the heading and the velocity of the obstacle, t is current time, and t_{freq} is the monitoring period for the change. For instance, if the obstacle changes motion over Ω_v, Ω_θ from the previous monitoring time, the ASV finds it to be a new risky situation and initiate calculation. Thus, the algorithm can prevent the obstacle from approaching the ASV abruptly.

F. Contingency maneuver

The proactive measures described in Sections III-D and III-E might not be enough when the controlled ASV is suddenly in a close-quarter situation, e.g., in case of intermittently lost signals from sensors (RADAR, LiDAR, AIS, GPS), or a suspicious approach for the sake of piracy or fishing [29], [30]. We propose Algorithm 2 to determine a contingency maneuver based on the ASV's abort distance Ω_d, with θ_{ext} hard-over angle for the change of heading. Ω_d is calculated by $\max(\mathcal{C}_x * k_d, L_R * k_\mathcal{R})$ as per the literature [31] where k_d is an abort threshold set by an operator.

Typically, θ_{ext} is considered to be over 35° [15]. Without iterative finding for θ described in Section III-D and Section III-E, the algorithm finds it immediately by reflection of θ_{ext}. Thus, the ASV can take a hard-over action to avoid an imminent collision.

Algorithm 2 Contingency maneuver

Input: $\mathbf{x}_O$, $\mathbf{x}_R$, θ_0, θ_{ext}, Ω_d
Output: θ
1: **if** $(\|\mathbf{x}_O - \mathbf{x}_R\| \leq \Omega_d)$ **then**
2: recalculate tangent heading θ_α, θ_β
3: $\theta' \leftarrow \theta_\alpha$ **or** θ_β making $\min(|\theta_\alpha - \theta_0|, |\theta_\beta - \theta_0|)$
4: $\theta \leftarrow \theta' - \theta_{ext}$ **or** $\theta \leftarrow \theta' + \theta_{ext}$
5: **return** θ

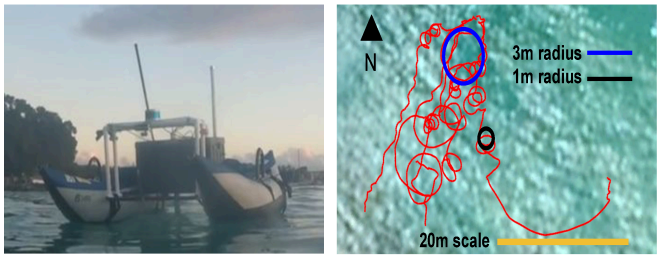


Fig. 4: Tested ASV model, *Catabot* (left) and its motion (right) in the Caribbean Sea.

G. Clearing evaluation

Let's say that obstacle avoidance is performed including trivial and non-trivial situations described in Section III-D to III-F. We proposed an evaluation process to confirm that the ASV is cleared from an obstacle. Then, the state model changes the ASV to the normal mission mode so that it can reach a goal point and perform an original task such as environmental monitoring.

The clearing status is calculated by two factors: distance between the ASV and the obstacle and angle between the relative motion vector and risk gradient vector. In other words, $\|\mathbf{x}_O - \mathbf{x}_R\|$ and $\arccos\left(\frac{\mathbf{rel}_{R|O} \cdot \mathbf{u}}{\|rel_{R|O}\| \|\mathbf{u}\|}\right)$ makes the situation clear or not. For the angle, it is based on the concept that the dot product of the two vectors when they each other becomes 0 – *orthogonal* – and turns into the negative – *cleared*. The thresholds can be determined by the operation. In general, the distance can be the same as DCPA setting and the angle can be the $112.5°$ [4].

IV. Results

Our proposed method has been tested using simulations, where vehicle model parameters have been extracted performing real experiments with our custom-made ASV, *Catabot*, a catamaran-hull type ASV, which uses a differential drive mechanism – see Fig. 4. First, we consider one-obstacle situations, to show the method ideal behavior. Second, we validate the robustness under more complex scenarios: sensor signal lost and abnormal behaviors of multiple obstacles. Lastly, we show a comparison with previous studies.

A. One-obstacle scenarios

1) Case 1: typical situations, as per [4], are considered *Overtaking*, *Head-on*, and *Crossing* – with a single dynamic obstacle, as shown in Fig. 5. In this case, the velocity v is constant and our algorithm changes only the heading θ of

TABLE I: Comparison of obstacle avoidance per each case. t_c is the computation time by averaging 10 iterations of the simulations. θ_0 is $45°$ in all cases.

Situation	$\lvert\theta$ - $\theta_0\rvert$ [°]	$\Gamma(\mathcal{T}_{\theta_0})$	$\Gamma(\mathcal{T}_\theta)$	CPA [m]	$\mathcal{C}_x$[m]	t_c [s]
Overtaking	16	120.73	38.49	33.12	15	0.006077
Head-on	37	119.83	55.39	31.93	15	0.010876
*Crossing*1	17	45.46	19.46	18.18	15	0.007202
*Crossing*2	14	49.48	24.44	18.72	15	0.006336

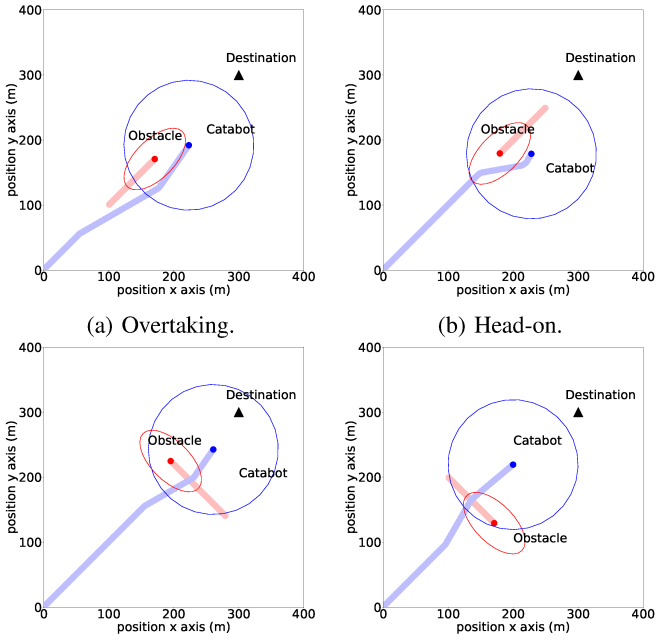
Catabot. The obstacle has the same principal dimension as *Catabot* – L_O as $2.5\,\mathrm{m}$. We set its velocity v_o as $1\,\mathrm{m/s}$ to allow the analysis of the overtaking scenario. The experiment was designed by placing the initial position of the obstacle outside of $\mathcal{L}$. The quantitative results are shown in Table I. To evaluate an action, we used the following metrics. $|\theta - \theta_0|$ is the angular change from the initial heading. Larger $|\theta - \theta_0|$ stands for large offset from the intended trajectory. We compare $\Gamma(\mathcal{T}_\theta)$ with $\Gamma(\mathcal{T}_{\theta_0})$. This represents how the evasive action is effective in lowering the potential risk of collision. The relationship between CPA and $\mathcal{C}_x$ represents the safety clearance from the obstacle. Last, computation time t_c for finding the action.

Comparing *Head-on* (Fig. 5a) and *Overtaking* (Fig. 5b), the first one has a relatively higher nearmiss than the second one due to the opposite heading of the obstacle with respect to *Catabot*, despite a short encounter time. This derives from the fact that near miss is calculated considering relative vectors (Equation (10)). The resulting action is a hard turn-over for *Head-on*. The computation time is also slightly higher compared to the *Overtaking situation*, as the algorithm needs to iterate for more angles to find an action that is safe.

Figures 5c and 5d show two *Crossing* situations, with the obstacle moving in two opposite directions. In both cases, our algorithm finds the collision-avoidance action by turning the heading to the stern of the obstacle. Note that, according to COLREG rule 17, in Fig. 5d the obstacle was supposed to be the *Give-way*. However, because we assumed that the obstacles will always be the *Stand-on*, our ASV becomes the *Give-way*. The COLREG rules 15 and 17 would also mandate to always take an action to our right, which our vehicle did not follow: our algorithm determined more efficient to go to the left and pass behind the vehicle, keeping the required safe CPA.

In all cases, using our definition of near miss $\Gamma(\mathcal{T}_{\theta_0})$ allows for (1) optimized action taken by the ASV, and (2) sufficient clearance with the obstacle, as shown comparing $\mathcal{C}_x$ with CPA in Table I.

2) Case 2: Our algorithm can also choose an action changing either velocity or velocity and heading together. The results are shown in Table II and Fig. 6 according to the three possible actions in the same situation as of *Crossing* 1. It is noted that H (Heading-based) method has advantage of efficiency without decreasing v_R and has a reasonable CPA despite the smallest. V (velocity-based) method waits for the obstacle to pass by ahead of *Catabot* thus, it has the largest CPA and time margin to assess the situation while it maintains θ. Finally, H-V (Heading- and Velocity-based)



(a) Overtaking. (b) Head-on.

(c) Crossing 1. (d) Crossing 2.

Fig. 5: Collision avoidance for diverse encounters in single vehicle scenario. **Red vehicle:** obstacle represented with $\mathcal{R}$ ellipse at the last position. **Blue vehicle:** *Catabot* represented with $\mathcal{L}$ circle at the last position. The more transparent dots represent the trajectory of the two vehicles. *Catabot* initial position (0,0) and destination (300, 300). Obstacle initial position and heading (a) (100, 100), 45° (b) (250, 250), 225° (c) (280, 140), 315° – crossing from *Catabot* right side (d) (100, 200), 135° – crossing from its left side.

	H	V	H-V
θ	62	45	52
v_R	3	1.5	1.98
Γ	19.46	13.48	21.40
CPA	18.18	21.22	18.84
t_c	0.0073	0.0040	0.0055

TABLE II: Comparisons of actions under the same situation: a method for changing H (Heading), V (Velocity), H-V(Heading and Velocity).

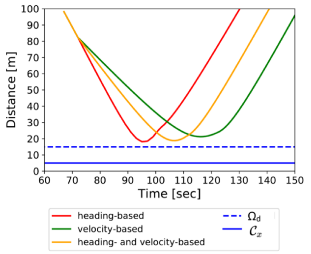


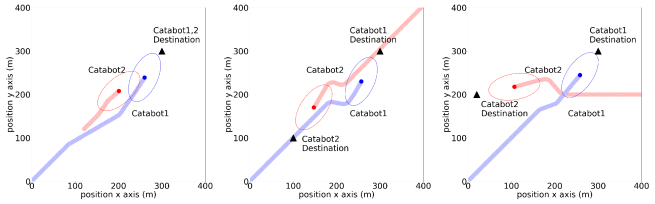
Fig. 6: Distance comparison.

method consolidates the advantage of the H and V. Since the passing time of $\mathcal{R}$ will be the longest, γ is the largest. However, it has a relatively small change of θ by decreasing v_R and still has a reasonable CPA.

3) Case 3: We tested scenarios when one ASV encounters another ASV with the same obstacle avoidance procedure. As shown in Fig. 7, the algorithm makes both vehicles take relevant maneuvers to safely pass, e.g., *Catabot*2 taking an action despite being overtaken (*Give-way* vehicle) in Fig. 7a.

B. Multi-obstacle and Sensor failure scenarios

1) Case 1: Fig. 8 shows a scenario of random static and dynamic obstacles with a traffic congestion (total 25 obstacles) within $400\,\mathrm{m} \times 400\,\mathrm{m}$ area, similar to a heavy



(a) Overtaking. (b) Head-on. (c) Crossing.

Fig. 7: Our method applied to two identical ASVs (Red – *Catabot*1, Blue – *Catabot*2, both represented with $\mathcal{R}$ ellipse at the last position). *Catabot*1 speed $3\,\mathrm{m/s}$ and original heading 45°. *Catabot*2 speed and original heading (a) $1\,\mathrm{m/s}$, 45°, (b) $3\,\mathrm{m/s}$, 225°, (c) $3\,\mathrm{m/s}$, 270°.

traffic environment full of fishing boats, large commercial vessels, and fishing buoys in areas such as Singapore strait and South or East China Sea [32]. The size, velocity, and heading are different for each obstacle and set randomly from a range of realistic values. In all experiments, *Catabot* could avoid collisions and reach the destination. Fig. 9a shows when obstacle detection happened and how distance changed. After the nearmiss-based action was initiated, the obstacle was cleared even if it tried to approach *Catabot*. We found out that the minimum $\mathcal{C}_x$ was $15\,\mathrm{m}$ and the obstacles entered in the collision area. The action taken to avoid collision over time is shown in Fig. 9b and is derived from the calculated nearmiss value, shown in Fig. 9c. Note that, while only 6 obstacles are displayed in Fig. 9c, our algorithm considers all obstacles that are in the sensor range. Others do not contribute to an actual risk, and thus do not make the ASV change action. In such a scenario, the computation time for assessing the risk and determining the action for obstacles detected was $0.010\,08\,\mathrm{s}$ on average; whereas $0.001\,01\,\mathrm{s}$ for the other obstacles which *Catabot* assessed that it could safely pass without an action. These computational time results show that our method can perform real-time with many obstacles.

Our proposed algorithm also ensures obstacle avoidance in case of unexpected behaviors (Section III-E), such as sudden change by an obstacle of heading and/or velocity. An example in our scenario is from obstacle 5. *Catabot* analyzed obstacle 5 at time step $12\,\mathrm{s}$ and found out it is safe to pass without an action. However, as the obstacle 5 is following a zig-zag path (Fig. 8), *Catabot* performed a recalculation according to the periodic monitoring. As *Catabot* is expected to pass $\mathcal{R}$ over the nearmiss threshold ρ, it initiated an action to avoid obstacle 5.

2) Case 2: Despite the very high traffic in the previous scenario, the contingency maneuver (Section III-F) did not trigger. To test such a maneuver, we performed another robustness test: the obstacle is detected after a while inside the sensor range, simulating a false negative – scenario that can happen with, e.g., a RADAR and wooden fishing boats or buoys [33]. We configured the experiment by making the obstacle detectable only when very close to *Catabot*The spawned obstacle was static and located in (155, 155) at time step $60\,\mathrm{s}$. *Catabot* sensor detects the obstacle only after $60\,\mathrm{s}$, at $36.2\,\mathrm{m}$ from the obstacle (the sensor range is

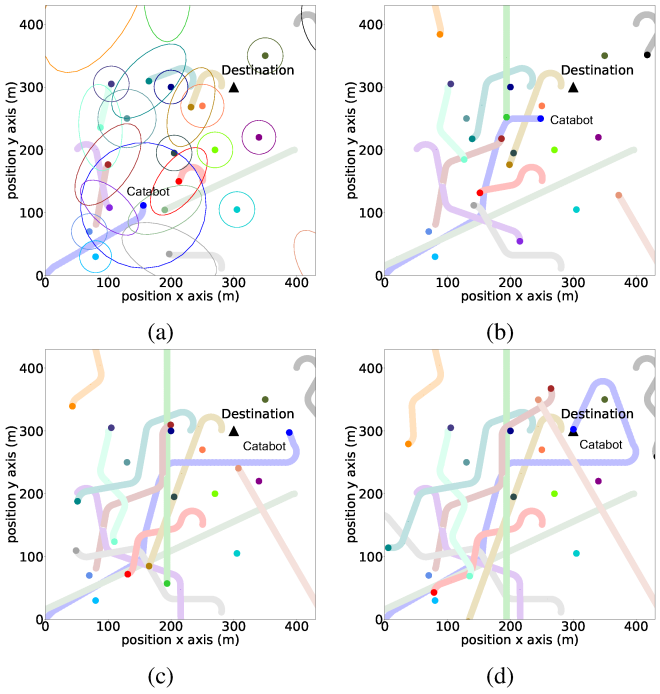


Fig. 8: Collision avoidance for diverse encounters in multi vehicle scenario. The more transparent dots represent the trajectory of the vehicles. Time steps (a) 65 s illustrated with $\mathcal{R}$ ellipse of obstacles and $\mathcal{L}$ circle of *Catabot* (b) 130 s, (c) 195 s, (d) 258 s ASV's arrival at destination.

TABLE III: Comparison of computation time with other methods [3]: VI as volume integration, MCS as Monte Carlo simulation, PF as probability flow, and our method (unit: s).

	VI	MCS	PF	Our method
Overtaking	0.31310	0.56081	0.01907	0.01138
Head-on	0.30385	0.59596	0.01809	0.00745
Crossing	0.32662	0.56002	0.01779	0.00746

100 m). After assessing the motion of the dynamic obstacle, our method calculates normal nearmiss-based action at 65 s toward 87°. Because the obstacle is inside Ω_d as the sensor failed to detect it beforehand, *Catabot* did a contingency maneuver to 110° and cleared the obstacle at 76 s. In this case, the computation time was 0.013 85 s for the nearmiss-based action and 0.000 81 s for the contingency maneuver. Note, a contingency maneuver overrides the normal obstacle avoidance to ensure the ASV safety.

C. Comparison

We qualitatively and quantitatively evaluate the proposed algorithm with other papers:

- Overall performance of the computational time is shown in Table III using experimental setup and result as [3]. In the worst case – overtaking, our nearmiss-based method performs 1.68 times faster than other methods.
- The proposed method considers an effective and efficient maneuver than the state of the art VO method [5]. Fig. 10a shows that our method passes the stern of the obstacle with $|\theta - \theta_0|$ as 14° and maintained speed as 3.0 m/s, whereas VO methods passed the ahead of the the obstacle, which can cause a navigational burden

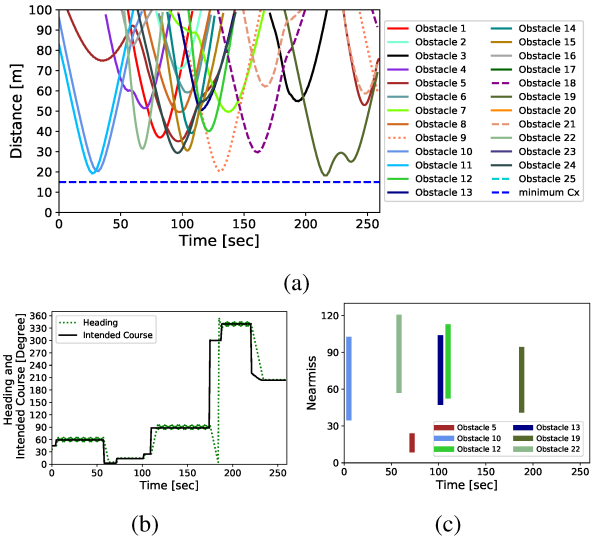


Fig. 9: Distance to obstacle and heading changes as per intended course decided by nearmiss-based obstacle avoidance action. (a) Distance obstacle comparison: Obstacle 17, 20, 23, 25 were not detected being out of $\mathcal{L}$. (b) Intended course input and heading follow-up. (c) Obstacle's nearmiss change triggering action. After 230 s and clearance of obstacles, the intended course is back to the mission destination.

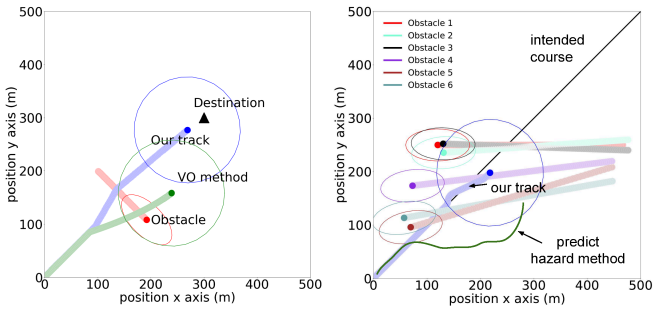


Fig. 10: Comparison with the literature (a) Crossing 2 situation in Fig. 5d compared with VO method [5] (b) multiple obstacle crossing scenario in [1].

until it fully cleared, with $|\theta - \theta_0|$ as 26° and reduced speed as 1.8 m/s. In addition, if the state-of-the-art method were only constrained by COLREG, the *stand-on* vehicle would not take a proper action under this scenario in contrast to our method's preemptive action.

- The proposed method avoids obstacles while attaining smaller angle changes. Based on the same configuration of the paper [1], we compare the trajectory of the controlled ASV as shown in Fig. 10b. Our method avoided the obstacles with the worst CPA as 19.26 m.
- Differently from previous methods on single or a couple of multiple encounters, our algorithm could make the ASV navigate within heavy traffics. By far, no other maritime collision avoidance method tested over 20 obstacles – the most was 16 [18].
- Our method has a motion model which means it follows the actual rate of turn. This outperforms previous works

which do not cover smooth turning [27], [28].

V. Conclusion and Future Work

This paper presented a novel obstacle avoidance method based on risk vectors and near miss computation in real-time. The proposed method in simulation complying with actual robot motion models validates applications to successful collision avoidance in robust environments with multiple number of obstacles and their arbitrary motions. In a single obstacle scenario, our algorithm outperforms a state-of-art methods by performing 1.68 times faster computation as well as more efficient action resulting in similar clearnace, e.g., $14°$ heading alteration at same speed by our method under a crossing situation, while $26°$ heading alteration with 40% speed reduction by the other methods. In addition, an ASV based on our algorithm is able to avoid 25 obstacles in a congested traffic scenario with fast computation time as $0.001\,01\,\mathrm{s}$ on average. This method can be applied regardless of the geometric size of the vehicle, including not only small low-cost vehicles such as ASVs but also large high-cost vessels such as MASS. In addition to the application in 2D contexts, potential expansion into 3D space with risk vectors and fast algorithms is expected to work effectively.

Our future work is to test the proposed algorithm in the field with a real ASV. In addition, we are investigating obstacle detection with the sensors to develop a full pipeline.

ACKNOWLEDGMENT

This work is supported in part by the Burke Research Initiation Award and NSF CNS-1919647, OIA1923004.

References

[1] 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 Transactions on Intelligent Transportation Systems*, vol. 17, no. 12, pp. 3407–3422, Dec 2016.

[2] M. R. Benjamin, M. Defilippo, P. Robinette, and M. Novitzky, "Obstacle avoidance using multiobjective optimization and a dynamic obstacle manager," *IEEE Journal of Oceanic Engineering*, vol. 44, no. 2, pp. 331–342, April 2019.

[3] J. Park and J. Kim, "Predictive evaluation of ship collision risk using the concept of probability flow," *IEEE Journal of Oceanic Engineering*, vol. 42, no. 4, pp. 836–845, Oct 2017.

[4] International Maritime Organization, "Convention on the international regulations for preventing collisions at sea, 1972 (COLREGs)," 1972.

[5] J. Snape, J. Van Den Berg, S. J. Guy, and D. Manocha, "The hybrid reciprocal velocity obstacle," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 696–706, 2011.

[6] Y. Kuwata, M. T. Wolf, D. Zarzhitsky, and T. L. Huntsberger, "Safe maritime autonomous navigation with colregs, using velocity obstacles," *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 110–119, 2014.

[7] S. Samavati, M. Zarei, and M. T. Masouleh, "An optimal motion planning and obstacle avoidance algorithm based on the finite time velocity obstacle approach," in *2017 Artificial Intelligence and Signal Processing Conference (AISP)*, 2017, pp. 250–255.

[8] M. Dunbabin and L. Marques, "Robots for environmental monitoring: Significant advancements and applications," *IEEE Robot. Autom. Mag.*, vol. 19, no. 1, pp. 24–39, 2012.

[9] M. Hoy, A. S. Matveev, and A. V. Savkin, "Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey," *Robotica*, vol. 33, no. 3, pp. 463–497, 2015.

[10] M. Otte and E. Frazzoli, "Rrt^x: Real-time motion planning/replanning for environments with unpredictable obstacles," in *Algorithmic Foundations of Robotics XI*. Springer, 2015, pp. 461–478.

[11] L. Kavraki, P. Svestka, and M. H. Overmars, *Probabilistic roadmaps for path planning in high-dimensional configuration spaces*. Unknown Publisher, 1994, vol. 1994.

[12] D. Panagou, "Motion planning and collision avoidance using navigation vector fields," in *Proc. ICRA*. IEEE, 2014, pp. 2513–2518.

[13] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robot. Autom. Mag.*, vol. 4, no. 1, pp. 23–33, 1997.

[14] M. R. Benjamin, "Capturing velocity function plateaus for efficient marine vehicle collision avoidance calculations," in *2018 OCEANS - MTS/IEEE Kobe Techno-Oceans (OTO)*, May 2018, pp. 1–10.

[15] K. Woerner, M. R. Benjamin, M. Novitzky, and J. J. Leonard, "Quantifying protocol evaluation for autonomous collision avoidance," *Auton. Robot.*, vol. 43, no. 4, pp. 967–991, 2019.

[16] R. Polvara, S. Sharma, J. Wan, A. Manning, and R. Sutton, "Obstacle avoidance approaches for autonomous navigation of unmanned surface vehicles," *Journal of Navigation*, vol. 71, no. 1, p. 241–256, 2018.

[17] T. Statheros, G. Howells, and K. M. Maier, "Autonomous ship collision avoidance navigation concepts, technologies and techniques," *Journal of Navigation*, vol. 61, no. 1, p. 129–142, 2008.

[18] H. Lyu and Y. Yin, "Colregs-constrained real-time path planning for autonomous ships using modified artificial potential fields," *Journal of Navigation*, vol. 72, no. 3, p. 588–608, 2019.

[19] L. Zhao and M.-I. Roh, "Colregs-compliant multiship collision avoidance based on deep reinforcement learning," *Ocean Engineering*, vol. 191, p. 106436, 2019.

[20] Q. Xu and N. Wang, "A survey on ship collision risk evaluation," *Promet - Traffic & Transportation*, vol. 26, no. 6, pp. 475–486, Dec. 2014.

[21] 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 Engineering*, vol. 105, pp. 336 – 348, 2015.

[22] P. Chen, Y. Huang, J. Mou, and P. van Gelder, "Ship collision candidate detection method: A velocity obstacle approach," *Ocean Engineering*, vol. 170, pp. 186 – 198, 2018.

[23] L. P. Perera, V. Ferrari, F. P. Santos, M. A. Hinostroza, and C. Guedes Soares, "Experimental evaluations on ship autonomous navigation and collision avoidance by intelligent guidance," *IEEE Journal of Oceanic Engineering*, vol. 40, no. 2, pp. 374–387, April 2015.

[24] W. Zhang, F. Goerlandt, P. Kujala, and Y. Wang, "An advanced method for detecting possible near miss ship collisions from ais data," *Ocean Engineering*, vol. 124, pp. 141 – 156, 2016.

[25] Y. Fujii, H. Yamanouchi, and T. Matui, *Survey on Vessel Traffic Management Systems and Brief Introduction to Marine Traffic Studies*. Electronic Navigation Research Institute, 1984.

[26] International Maritime Organization, *Standards for Ship Maneuverability Resolution MSC.137(76)*, 2002.

[27] K. Jo, K. Chu, and M. Sunwoo, "Interacting multiple model filter-based sensor fusion of gps with in-vehicle sensors for real-time vehicle positioning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 1, pp. 329–343, March 2012.

[28] J. Larson, M. Bruch, and J. Ebken, "Autonomous navigation and obstacle avoidance for unmanned surface vehicles," in *Unmanned Systems Technology VIII*, G. R. Gerhart, C. M. Shoemaker, and D. W. Gage, Eds., vol. 6230, International Society for Optics and Photonics. SPIE, 2006, pp. 53 – 64.

[29] S. Pristrom, Z. Yang, J. Wang, and X. Yan, "A novel flexible model for piracy and robbery assessment of merchant ship operations," *Reliability Engineering & System Safety*, vol. 155, pp. 196 – 211, 2016.

[30] X. Wang, Z. Zhou, Y. Liu, X. Wang, and H. Song, "Discussions about the best methods of collision between merchant ships and fishing boats during fishing seasons near chinese coast," in *International Conference on Computer Information Systems and Industrial Applications*. Atlantis Press, 2015/06.

[31] K. Kwik, "Calculation of ship collision avoidance manoeuvres: A simplified approach," *Ocean Engineering*, vol. 16, no. 5, pp. 475 – 491, 1989.

[32] P. Silveira, A. Teixeira, and C. G. Soares, "Use of ais data to characterise marine traffic patterns and ship collision risk off the coast of portugal," *Journal of Navigation*, vol. 66, no. 6, p. 879–898, 2013.

[33] H. Leung, N. Dubash, and N. Xie, "Detection of small objects in clutter using a ga-rbf neural network," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 1, pp. 98–118, Jan 2002.